IEEE Signal Processing MAGAZINE
Volume 36 | Number 1 | January 2019

RECENT ADVANCES IN MUSIC SIGNAL PROCESSING

Audio and Beyond

Autonomous UAV Filming in Dynamic Unstructured Outdoor Environments

Paraconsistent Feature Engineering

Efficient Floating-Point Division for DSP Application

IEEE VIP Cup 2018 Student Competition Highlights

IEEE Signal Processing Society
IEEE

CALL FOR PAPERS
52nd Annual Asilomar Conference on Signals, Systems, and Computers

Asilomar Hotel and Conference Grounds
Pacific Grove, California
October 28 - 31, 2018
www.asilomarssc.org

Authors are invited to submit papers before **May 1st, 2018**, in the following areas:

A. Communications Systems: 1. Modulation and Coding, 2. Cognitive Radio and Spectrum Sharing, 3. Ultra-Low Latency, 4. Optimization, 5. Physical Layer Security and Privacy, 6. mmWave, 7. Underwater, 8. Wireline and Optical Communications, 9. Satellite, 10. IoT, 11. V2V, 12. 5G and Beyond

B. MIMO Communications and Signal Processing: 1. Multiuser and Massive MIMO, 2. Channel Estimation & Equalization, 3. Full-Duplex, 4. Cooperation & Relaying, 5. Interference Management & Awareness, 6. mmWave, 7. Optimization

C. Networks: 1. Wireless Networks, 2. IoT, 3. Network Info. Theory, 4. Optimization, 5. Graph Signal Processing, 6. Social Networks, 7. Distributed Algorithms, 8. Security, 9. Computational Offloading, 10. Self-Organizing Networks, 11. UAV & V2V Netw., 12. Smart Grid

D. Adaptive Systems, Machine Learning, Data Analytics: 1. Compressive Sensing, 2. Machine Learning, 3. Estimation, Inference and Learning, 4. Adaptive and Cognitive Systems, 5. Adaptive Filtering, 6. Fast and Scalable Algorithms, 7. High-Dimensional Large-Scale Data, 8. Distributed Computation and Storage, 9. Deep Learning

E. Array Processing and Multisensor Systems: 1. Source Localization and Separation, 2. Beamforming, 3. Robust Methods 4. MIMO and Cognitive Radars, 5. Tensor Models and Processing, 6. Sparse Sensor Arrays, 7. Optimization, 8. Applications (Imaging, Sonar, Radar, Microphone Arrays, etc.).

F. Biomedical Signal and Image Processing: 1. Molecular and Medical Imaging, 2. Computational Imaging, 3. Neuroengineering, 4. Processing of Physiological Signals, 5. Bioinformatics and Computational Biology, 6. Image Registration and Multimodal Imaging, 7. Functional Imaging, 8. Brain Machine Interfaces, 9. Neural Signal Processing, 10. Visualization

G. Architectures and Implementation: 1. Energy Efficiency, 2. Accelerators, 3. Reconfigurable Processing, 4. Non-idealities, 5. Multicore, Many-core & Distributed Systems, 6. Algorithm & Architecture Co-optimization, 7. Architectures for Big Data, 8. Architectures for Machine Learning 9. Cyber-Physical System, 10. Test-beds, 11. Mixed-Signal Processors 12. Arithm. & Algorithms

H. Speech, Image and Video Processing: 1. Speech Processing, 2. Speech Coding, 3. Speech Recognition, 4. Audio Coding, 5. Document Processing, 6. Models for Signal and Image Processing, 7. Image and Video Coding and Communication, 8. Learning and Autonomous Systems, 9. Computer Vision, Image and Video Analysis, 10. Image/Video Forensics, 12. Biometrics and Security, 13. Hybrid Imaging Systems

Submissions should include a 50 to 100 word abstract and an extended summary (500 to 1000 words, plus figures). Submissions must include the title of the paper, authors' names and affiliations, technical area, and topic designation from the above list. Check the conference website (www.asilomarssc.org) for specific information on the electronic submission process. Submissions will be accepted starting February 1, 2018. **No more than FOUR submissions are allowed** per contributor, as author or co-author. **All submissions must be received by May 1st, 2018**. Notifications of acceptance will be mailed by mid July 2018, and author information will be available on the conference website by late July 2018. Full papers are due shortly after the conference and published in early 2018. **To publish a paper the author must register and present the paper at the conference.** All technical questions should be directed to Technical Program Chair **Prof. Martin Haardt**, e-mail martin.haardt@tu-ilmenau.de or General Chair **Prof. Visa Koivunen**, email visa.koivunen@aalto.fi. Prospective organizers of special sessions are invited to submit proposals to the General or Technical Chair by January 15, 2018. Proposals must include title, topic, rationale, session outline, contact information, and a description of how the session will be organized.

CONFERENCE COMMITTEE

General Chair:	Visa Koivunen, *Aalto University, Finland*
Technical Program Chair:	Martin Haardt, *TU Ilmenau, Germany*
Conference Coordinator:	Monique P. Fargues*, *Naval Postgraduate School*
Publication Chair:	Michael Matthews, *NorthWest Research Associates, Inc.*
Publicity Chair:	Linda S. DeBrunner, *Florida State University*
Finance Chair:	John D. Roth*, *Naval Postgraduate School*
Electronic Media Chair:	Marios S. Pattichis, *University of New Mexico*

The site for the 2018 Conference is at the Asilomar Conference Grounds in Pacific Grove, CA. The grounds border the Pacific Ocean and are close to Monterey, Carmel, and the scenic Seventeen Mile Drive in Pebble Beach. The Conference is organized by the non-profit Signals, Systems and Computers Conference Corporation. *Serving in his/her personal capacity.

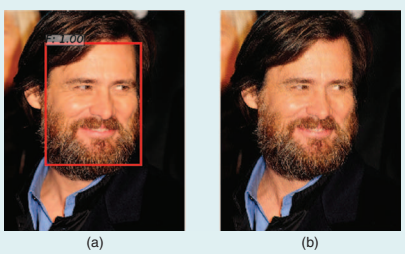
Digital Object Identifier 10.1109/MSP.2018.2884746

Contents

Volume 36 | Number 1 | January 2019

SPECIAL SECTION

MUSIC SIGNAL PROCESSING



(a) (b)

PG. 13

ON THE COVER

This issue of *IEEE Signal Processing Magazine* surveys recent advances in music processing with a focus on audio signals. Eleven articles cover topics including music analysis, retrieval, source separation, singing-voice processing, musical sound synthesis, and user interfaces, among others.

COVER IMAGE: ©ISTOCKPHOTO.COM/MOORSKY

COLUMNS

PG. 164

IEEE SIGNAL PROCESSING MAGAZINE (ISSN 1053-5888) (ISPREG) is published bimonthly by the Institute of Electrical and Electronics Engineers, Inc., 3 Park Avenue, 17th Floor, New York, NY 10016-5997 USA (+1 212 419 7900). Responsibility for the contents rests upon the authors and not the IEEE, the Society, or its members. Annual member subscriptions included in Society fee. Nonmember subscriptions available upon request. **Individual copies:** IEEE Members US$20.00 (first copy only), nonmembers US$241.00 per copy. Copyright and Reprint Permissions: Abstracting is permitted with credit to the source. Libraries are permitted to photocopy beyond the limits of U.S. Copyright Law for private use of patrons: 1) those post-1977 articles that carry a code at the bottom of the first page, provided the per-copy fee indicated in the code is paid through the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923 USA; 2) pre-1978 articles without fee. Instructors are permitted to photocopy isolated articles for noncommercial classroom use without fee. **For all other copying, reprint, or republication permission,** write to IEEE Service Center, 445 Hoes Lane, Piscataway, NJ 08854 USA. Copyright © 2019 by the Institute of Electrical and Electronics Engineers, Inc. All rights reserved. Periodicals postage paid at New York, NY, and at additional mailing offices. **Postmaster:** Send address changes to IEEE Signal Processing Magazine, IEEE, 445 Hoes Lane, Piscataway, NJ 08854 USA. Canadian GST #125634188 **Printed in the U.S.A.**

Digital Object Identifier 10.1109/MSP.2018.2877379

DEPARTMENTS

©ISTOCKPHOTO.COM/GORANQ

The 26th IEEE International Conference on Image Processing will be held at the Taipei International Convention Center, Taipei, Taiwan, 22–25 September 2019.

SUSTAINABLE FORESTRY INITIATIVE Certified Chain of Custody Promoting Sustainable Forestry www.sfiprogram.org SFI-01681

IEEE prohibits discrimination, harassment, and bullying. For more information, visit http://www.ieee.org/web/aboutus/whatis/policies/p9-26.html.

IEEE Signal Processing Magazine

EDITOR-IN-CHIEF
Robert W. Heath, Jr.—The University of Texas at Austin, U.S.A.

AREA EDITORS

Feature Articles
Matthew McKay—Hong Kong University of Science and Technology, Hong Kong SAR of China

Special Issues
Namrata Vaswani—Iowa State University, U.S.A.

Columns and Forum
Roberto Togneri—The University of Western Australia

e-Newsletter
Ervin Sejdic—University of Pittsburgh, U.S.A.

Social Media and Outreach
Tiago Henrique Falk—INRS, Canada

Special Initiatives
Andres Kwasinski—Rochester Institute of Technology, U.S.A.

EDITORIAL BOARD
Daniel Bliss—Arizona State University, USA
Danijela Cabric—University of California, Los Angeles
Volkan Cevher—École polytechnique fédérale de Lausanne, Switzerland
Mrityunjoy Chakraborty—Indian Institute of Technology, Kharagpur, India
George Chrisikos—Qualcomm, Inc., U.S.A.
Elza Erkip—New York University, U.S.A.
Alfonso Farina—Leonardo S.p.A., Italy
Clem Karl—Boston University, U.S.A.
C.-C. Jay Kuo—University of Southern California, U.S.A.
Erik Larsson—Linköping University, Sweden
David Love—Purdue University, USA
Maria G. Martini—Kingston University, U.K.
Helen Meng—City University of Hong Kong, Hong Kong SAR of China
Meinard Mueller—Friedrich-Alexander Universität Erlangen-Nürnberg, Germany
Alejandro Ribeiro—University of Pennsylvania, U.S.A.
Douglas O'Shaughnessy—INRS Universite de Recherche, Canada
Osvaldo Simeone—Kings College London, U.K.
Milica Stojanovic—Northeastern University, USA
Ananthram Swami, Army Research Labs, U.S.A.
Jong Chul Ye—KAIST, South Korea
Qing Zhao—Cornell University, USA
Josiane Zerubia—INRIA Sophia-Antipolis Mediterranee, France

ASSOCIATE EDITORS—COLUMNS AND FORUM
Ivan Bajic—Simon Fraser University, Canada
Balázs Bank—Budapest University of Technology and Economics, Hungary
Panayiotis (Panos) Georgiou—University of Southern California, U.S.A.
Hana Godrich—Rutgers University, U.S.A.
Rodrigo Capobianco Guido—São Paulo State University, Brazil
Yuan-Hao Huang—National Tsing Hua University, Taiwan
Euee Seon Jang—Hanyang University, Republic of Korea
Vishal Patel—Rutgers University, U.S.A.
Christian Ritz—University of Wollongong, Australia
Changshui Zhang—Tsinghua University, China
H. Vicky Zhao—Tsinghua University, China

ASSOCIATE EDITORS—e-NEWSLETTER
Csaba Benedek—Hungarian Academy of Sciences, Hungary
Yuhong Liu—Penn State University at Altoona, U.S.A.
Andreas Merentitis—University of Athens, Greece
Michael Muma—TU Darmstadt, Germany
Le Yang—Harbin Institute of Technology, China
Xiaorong Zhang—San Francisco State University, U.S.A.

ASSOCIATE EDITOR—SOCIAL MEDIA/OUTREACH
Guijin Wang—Tsinghua University, China

IEEE SIGNAL PROCESSING SOCIETY
Ali H. Sayed—President
Ahmed Tewfik—President-Elect
Fernando Pereira—Vice President, Conferences
Nikos D. Sidiropoulos—Vice President, Membership
Sergio Theodoridis—Vice President, Publications
Walter Kellerman—Vice President, Technical Directions

IEEE SIGNAL PROCESSING SOCIETY STAFF
William Colacchio—Senior Manager, Publications and Education Strategy and Services
Rebecca Wollman—Publications Administrator

IEEE PERIODICALS MAGAZINES DEPARTMENT
Jessica Welsh, *Managing Editor*
Geraldine Krolin-Taylor, *Senior Managing Editor*
Janet Dudar, *Senior Art Director*
Gail A. Schnitzer, *Associate Art Director*
Theresa L. Smith, *Production Coordinator*
Mark David, *Director, Business Development - Media & Advertising*
Felicia Spagnoli, *Advertising Production Manager*
Peter M. Tuohy, *Production Director*
Kevin Lisankie, *Editorial Services Director*
Dawn M. Melley, *Staff Director, Publishing Operations*

Digital Object Identifier 10.1109/MSP.2018.2877380

SCOPE: *IEEE Signal Processing Magazine* publishes tutorial-style articles on signal processing research and applications as well as columns and forums on issues of interest. Its coverage ranges from fundamental principles to practical implementation, reflecting the multidimensional facets of interests and concerns of the community. Its mission is to bring up-to-date, emerging and active technical developments, issues, and events to the research, educational, and professional communities. It is also the main Society communication platform addressing important issues concerning all members.

Robert W. Heath, Jr. | Editor-in-Chief | rheath@utexas.edu

Vehicular Applications of Signal Processing

Signal processing is what's behind many developments in vehicle automation. As of the writing of this editorial, I am preparing for a talk at the University of California, Los Angeles, and a distinguished symposium talk at IEEE GlobalSIP on "Signal Processing for Automated Driving." To prepare, I have been reading beyond my core areas of expertise and have gained an appreciation for the wide range of signal processing opportunities in vehicular systems.

Vehicles are becoming automated thanks, in part, to diverse kinds of sensors, not to mention communication modalities. Higher levels of automation remove control from the driver, eventually replacing the need for human drivers entirely. Radars and cameras are found in most vehicles with any level of automation, while lidar is used only at higher levels due to cost. Various types of communication, including 4G cellular, wireless local area networks, and dedicated short-range communication, are also found in different combinations. Signal processing is everywhere.

In this editorial, I review some applications of signal processing in vehicular systems. I focus specifically on applications that are pertinent to automated driving (the so-called higher levels of automation). Many technologies also support driver assistance applications. In a future issue of IEEE *Signal Processing Magazine* (*SPM*), a call for papers will be announced for a special issue on autonomous driving.

Digital Object Identifier 10.1109/MSP.2018.2880687
Date of publication: 24 December 2018

Radar is widely used at all levels of automation. Ultrasonic radars are implemented for short-range positioning, while millimeter-wave radars are applied during driving for ranging, velocity estimation, mapping, and localization. Most automated vehicle prototypes feature at least a forward- and backward-facing radar, but some have six or more radars. The advantages of radar over light-based sensing modalities are that it works equally well during either day or night, and it is able to penetrate fog and smog. Statistical signal processing is used extensively in radar. Current research challenges include designing hardware and algorithms for mass-produced millimeter-wave radars with large arrays, high bandwidth, and distortions due to radio-frequency components, as well as managing coexistence and interferences. *SPM* has an upcoming two-part special issue on "Advances in Radar Systems for Modern Civilian and Commercial Applications."

Lidar is used at the highest levels of automation. Automated vehicle prototypes often have a single larger lidar or two side-mounted lidar units. In principle, lidar is the light-based equivalent of radar, although, practically, the way the signals are generated and processed is vastly different. Lidar implements laser beams with mechanical scanning, while most radars have beams that are electronically steered. The system is used primarily to generate 3D point clouds that allow accurate mapping and object detection. Types of signal processing include image processing and machine learning. One of the most significant impediments to the wider deployment of lidar is the high cost of the mechanical solutions. Companies are currently working on solid-state versions where there will likely be many more signal processing applications to compensate for hardware imperfections.

Cameras, making use of multiple frames at the visible-light level, are widely used sensors in vehicle automation. This development is not surprising, as human drivers primarily rely on their eyes. Most automated vehicle prototypes have several cameras, both forward- and rear-facing. Cameras are used in combination with machine-learning algorithms as a core part of computer vision. Generally, the cameras are used to mimic functions performed by the human driver. They have a particular application for object detection, especially bicycles and pedestrians, which are otherwise hard to see with radar or lidar. Different camera configurations are possible, including stereo and multiview. There are many research challenges related to exploiting cameras for automated driving, including better algorithms for object recognition, combining the views of multiple cameras, enhancing performance in low-light conditions, and the use of the infrared spectrum.

Sensor fusion is a key signal processing application in automated driving. Each of the aforementioned sensors provides information about the shared environment from a different physical and spectral perspective. Sensor fusion is the

(continued on page 6)

Ali H. Sayed | IEEE Signal Processing Society President | ali.sayed@epfl.ch

Dynamite, Electricity, and Nobel's World

It has become a tradition to expect the announcement of the Nobel Prizes in the last quarter of the year. This past October, we were pleased to witness the rarity of two women scientists, Donna Strickland from Canada and Frances Arnold from the United States, be accorded the Nobel Prizes in Physics and Chemistry, respectively. Strickland is only the third woman to receive the Nobel Prize in Physics (the last time this happened was more than 50 years ago in 1963). What a fitting answer these announcements were to the statement that "physics was invented and built by men," which was proclaimed by an Italian physicist at the European Center for Nuclear Research (CERN) in September. He was attending a workshop, the goal of which was to highlight gender issues in physics, and it did!

This does not mean that the Nobel Prize institution has not been remiss in its duty to acknowledge the contributions of outstanding women scientists. In fact, the Nobel Prizes provide one vivid example of the problematic gender gap that plagues the sciences, as highlighted by the statistics in Table 1 [1]. Only 5.7% of all Nobel laureates are women—and Marie Currie is counted twice! To compound the issue, only a single woman has received the Nobel Prize in Economic Sciences. These dismal numbers are not for lack of excellent candidates. For example, in 1944, the Austrian-Swedish physicist Lise Meitner (1878–1968) was denied sharing the Nobel Prize in Chemistry, which went to her male collaborator, in a decision that many considered unjust. Other similar incidents have occurred [2].

The Nobel Prize was a stroke of genius by the Swedish chemist Alfred Nobel (1833–1896), who was the inventor of dynamite and profited immensely from the sales of arms, gunpowder, and explosives. Many believe that to ensure that his legacy was not associated with these instruments of death, he bequeathed his wealth to the establishment of five Nobel Prizes in Physics, Chemistry, Physiology or Medicine, Literature, and Peace. The first prizes were given in 1901, while the Economics prize was added later in 1969.

Given their long history and unwavering quality, the Nobel Prizes have risen to become the most eminent prize in science and are watched across the globe with intense media coverage. Their prestige is undisputed and their laureates' achievements are among the finest. At the same time, the Nobel Foundation is held accountable to higher standards and its decisions are subject to scrutiny.

For example, there was a controversy this past year when the 2018 Nobel Prize in Literature was not awarded due to sexual assault allegations by 18 women against the husband of one of the Swedish Academy members. In earlier years, the award did not recognize several deserving scientists or pacifists. One glaring omission is Mahatma Gandhi, who was apparently nominated five times but was never awarded the Nobel Peace Prize! Other examples are Thomas Edison and Nikola Tesla who were bypassed despite the incredible revolution that electricity has brought to our world, certainly

Table 1. Gender statistics for the Nobel Prizes.

Nobel Prize	Number of Prizes	Number of Laureates	Female Laureates	Women (%)
Physics	112	210	3	1.4
Chemistry	110	181	5	2.8
Medicine	109	216	12	5.6
Literature	110	114	14	12.3
Peace	99	107	17	15.9
Economy	50	81	1	1.2
Total	590	909	52	5.7

Digital Object Identifier 10.1109/MSP.2018.2880688
Date of publication: 24 December 2018

more than Nobel's dynamite and gunpowder. If you were to choose between turning off all electricity in the world or destroying all piles of dynamite and gunpowder, which choice would you make? In other instances, the prizes have been controversially awarded to some scientists while ignoring legitimate contributions by others. This even happened as recently as 2017 when the Nobel Prize in Physics was awarded to three deserving scientists for the discovery of gravitational waves. This discovery involved the efforts of literally hundreds of other individuals from more than 20 countries. The three scientists were awarded the prize for "their decisive contributions to the LIGO detector and the observation of gravitational waves." Notice the use of the word "decisive." It has a purpose. It was perhaps meant to ensure that only three individuals share the award, which is the limit that the Nobel Foundation follows. This rule is likely spreading scientific injustices, regardless of intention. The scientific community has always been strict about the practice of proper citation to the work of others and we, as scientists, are expected to properly acknowledge prior contributions. Why should the Nobel Foundation be allowed to apply a different standard?

Even the Nobel Peace Prize has generated controversies. It is reported that Alfred Nobel once stated, "I intend to leave after my death a large fund for the promotion of the peace idea, but I am skeptical of its results" [3]. Indeed, the Nobel Peace Prize has been awarded 99 times since its launch. Is our world more peaceful today? There have always been conflicts brewing in different parts of the world with innocent people and children falling victim to violence. Who was not touched by the image of three-year-old Alan Kurdi lying lifeless on a beach in September 2015 after drowning in the Mediterranean Sea, or the painful sight of seven-year-old Amal Hussain who starved to death this past October 2018 in the midst of a tragic war? Despite modern advances in our world, children do still starve to death. Yasser Arafat, Shimon Peres, and Yitzhak Rabin shared the 1994 Nobel Peace Prize for "their efforts to create peace in the Middle East." Is the Middle East a more peaceful place today? Many others are calling for the 1991 Nobel Peace Prize to be revoked citing the apparent indifference of Aung San Suu Kyi to the calamity befalling the Rohingya people in her country. I have always wondered, since my younger years, how could the origins of a Peace Prize of this magnitude be associated with dynamite and gunpowder!

Is the Nobel Prize doing enough to stimulate diversity in the STEM fields?

I recently watched a documentary about the life of King Edward VII who ascended to the throne of England in January 1901, following the death of his mother Queen Victoria. This is the same year in which the Nobel Prize was launched. What I found interesting about the two-part documentary was not his adventures as a prince, but rather the video footage showing how life was at that era when Nobel penned his will. Alfred Nobel (1833–1896) lived in a different time with its own technological limitations. Imagine if we were to switch off electricity today, ground our planes, park our cars, disconnect our communications infrastructure, and disable all phones, radios, TVs, and the Internet. In the minds of many, we would be returning to the Stone Age. But that was, to a good extent, how the world looked like during Nobel's lifetime. Nobel did not witness any of the wonders we take for granted today. His interests and thinking were framed by the experiences of his time. While the STEM fields (science, technology, engineering, and mathematics) are recognized today as indispensable and strategic drivers for the economic growth of nations, Nobel himself ignored the "TEM" fields altogether and focused mainly on "S" alone. At a time when we need to popularize STEM fields among younger students, and especially among female students, it is fair to question whether the Nobel Prizes of today are helping or hindering this effort. I am of the opinion that these prizes could and should do more to support STEM outreach for several reasons.

First, the Nobel Foundation is hardly awarding sufficient prizes to female scientists. This in itself sends a distorted message to the younger generation of female STEM students who are eager for role models.

Second, there was no place for technology, engineering, and mathematics in Nobel's plan following his "mature deliberation," as he refers to it in the opening line of his will. Many have criticized him for leaving out mathematics. Does not much of the work by laureates in the economic sciences, for example, rely heavily on sophisticated mathematical and statistical models?

Third, although he was an engineer, one can perhaps forgive Nobel's oversight of technology and engineering since; at his time, these disciplines did not have the significant influence they have on our lives today. There are today other prestigious prizes in these domains, including the Turing Award, the Kyoto Prize, and Queen Elizabeth's Prize for Engineering. Despite their prominence, these prizes do not attract the same level of global and media attention as the Nobel Prize. Back in 1986, a proposal was made to the Nobel Foundation by the American Association of Engineering Societies to create a Nobel Prize in Engineering. The proposal was rejected [4]. But that was more than 30 years ago and our world has changed dramatically since then. A step like this would immediately raise awareness of the critical role that technology and engineering play in modern times in the public's mind, as well as in the minds of the younger generation of students whom we wish to attract to the STEM fields.

Fourth, in some cases the Nobel Prize is taking recognition away from technology and engineering and marginalizing their role. This is because many engineering innovations such as the radio, the transistor, the LED, and fiber optics have been recognized under the Nobel Physics Prize and, moreover, many Nobel laureates have been well-accomplished engineering researchers. For example, Dr. Frances Arnold (this year's laureate in Chemistry) is a professor of chemical engineering and a member of the U.S. National Academy of Engineering. Her undergraduate degree was in mechanical and aerospace engineering, and her Ph.D. degree was in chemical engineering. Likewise, Dr. Shuji Nakamura (Physics laureate, 2014) is a

professor of materials science engineering at the University of California in Santa Barbara. His undergraduate degree was in electronic engineering in Japan. Also, Dr. Charles K. Kao (Physics laureate 2009) studied electrical engineering and received a Ph.D. degree in the same field in 1965. Closer to our discipline, Jack Kilby (1923–2005) was awarded the 2000 Nobel Prize in Physics for "basic work on information and communication technology." That is squarely in the field of interest of our professional society. Kilby was an electrical engineer. He worked on the first integrated circuit at Texas Instruments. The technology was pivotal in launching the digital signal processor revolution, and in embedding signal processing intelligence into billions of electronic devices and gadgets including your cell phones.

Nobel's intention has been to honor "inventions or discoveries" of the greatest practical benefit to mankind. It is difficult for anyone to argue that inventions like electricity; cellular communications, personal computing, and the Internet have not had such an impact. Besides, engineering today is a discipline where real discoveries and not just inventions happen, which is why the term "engineering sciences" is also common. It is not true anymore that scientific discovery alone drives engineering design. On the contrary, it is also true that engineering ideas help motivate and discover new science to enable them. And many Nobel Prize winning works would not have been possible without creative and amazing engineering and technological advances and discoveries. Einstein postulated the existence of gravitational waves around 100 years ago. Why did it take until 2016 to detect them?

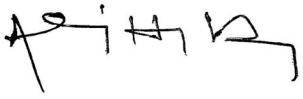
References

[1] The Nobel Foundation, "The Nobel Prize." Accessed on: Nov. 4, 2018. [Online]. Available: https://www.nobelprize.org

[2] A. Paul. (2018, Oct. 7). Five women who missed out on the Nobel prize. The Guardian. [Online]. Available: https://www.theguardian.com/science/2018/oct/07/five-women-the-nobel-prize-missed

[3] Brainy Quote. (2001). Alfred Nobel quotes. [Online]. Available: https://www.brainyquote.com/authors/alfred_nobel

[4] H. Petroski, "Engineering and the Nobel prizes," *Issues Sci. Technol.*, vol. 4, no. 1, pp. 56–60, 1987.



SP

FROM THE EDITOR *(continued from page 3)*

task of combining all of those observations to make meaningful decisions that account for different levels of uncertainty. Many research challenges remain in sensor fusion, with many specifically related to the combination of sensors under consideration and the level of preprocessing applied prior to fusion.

Communication is not required for automation, but it makes it more efficient. Most of the previous work at high levels of automation does not leverage the potential for low-latency and/or high data rate communication between vehicles. With communication, vehicles can share information over a much longer range than humans, making support for communication a departure from development over mirroring human drivers. Communication can be used to coordinate vehicles at lower levels of automation, as in platooning, for example, which leads to efficiency improvements. At higher levels of automation, communication facilities exchange-sensor data. This allows, in essence, vehicles to make use of the sensors on other vehicles or their infrastructure to expand the sensing range. New research is focused on the application of 5G communication systems to vehicles, especially high data rate millimeter-wave communications. For example, high data rates permit lower layers of sensor information to be shared and fused jointly. Signal processing research challenges include methods for making high data rate low-latency communication resilient in highly mobile channels, including tasks such as adaptive channel estimation and tracking in high-dimensional millimeter-wave communication systems.

While everything I have outlined targets ground vehicles, many of the research directions also apply to aerial vehicles. There are additional challenges due to the limited payloads in aerial vehicles, especially in small, unmanned vehicles. As a result, there are new tradeoffs related to the weight and energy consumption of sensors and signal processing hardware. For example, it may be possible to support only a limited number of sensors, and the data may be processed outboard at a ground-based processing center. An *SPM* special issue on signal processing for aerial vehicles will appear in the near future.

Vehicles are an exciting new application of signal processing. The types of signal processing associated with sensors and communication, however, have even broader applications. An example would be the similar challenges faced in robotics and factory automation. The recently launched Autonomous Systems Initiative (for more information, see http://asi.politecnica.unige.it) will become a focal point for signal processing research related to automation. In parallel, in *SPM*, we are working to include more content on these and other new advances of signal processing.

SP

2018 Member-at-Large and Regional Director-at-Large Election Results

Three new members-at-large will take their seats on the IEEE Signal Processing Society (SPS) Board of Governors (BoG) beginning 1 January 2019 and will serve until 31 December 2021. Eight candidates competed for the three positions. The successful candidates represent a broad spectrum of the IEEE SPS. The successful candidates are

- Douglas O'Shaughnessy
- Ana Isabel Pérez-Neira
- Zhi (Gerry) Tian.

Completing their terms as members-at-large on 31 December 2018 are Robert W. Heath, Jr., The University of Texas at Austin; Lina J. Karam, Arizona State University; and Min Wu, University of Maryland.

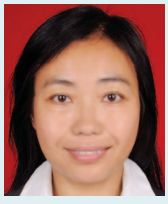
The BoG, the governing body that oversees the activities of the SPS, is responsible for establishing and implementing policy and receiving reports from its standing boards and committees. Members-at-large represent the member viewpoint in the Board's decision making. They typically review, discuss, and act upon a wide range of items affecting the actions, activities, and health of the Society. More information can be found at http://www.signalprocessingsociety.org/.

Digital Object Identifier 10.1109/MSP.2018.2876964
Date of publication: 24 December 2018

The new members-at-large

Douglas O'Shaughnessy

Ana Isabel Pérez-Neira



Zhi (Gerry) Tian

Two new regional directors-at-large will take their seats on the SPS BoG and Membership Board beginning 1 January 2019 and will serve until 31 December 2020. Regional directors-at-large promote and foster local activities (such as conferences, meetings, and social networking) and encourage new Chapter development; represent their Regions to the core of the SPS; offer advice to improve membership relations, recruiting, and service to their Regions; guide and work with their corresponding Chapters to serve their members; and assist the vice president-membership in conducting Chapter reviews. The new regional directors-at-large are

- Bhuvana Ramabhadran (Regions 1–6)
- Cédric Richard (Region 8).

The new regional directors-at-large

Regions 1–6

Bhuvana Ramabhadran

Region 8

Cédric Richard

Completing their terms as regional directors-at-large on 31 December 2018 are Zhengdao Wang, Iowa State University, for Regions 1–6 and Ana Isabel Pérez-Neira, Technical University of Catalonia, for Region 8.

SPS Announces 2019 Class of Distinguished Lecturers and Distinguished Industry Speakers

The IEEE Signal Processing Society (SPS) Distinguished Lecturer (DL) Program provides a means for Chapters to have access to well-known educators and authors in the field of signal processing to lecture at Chapter meetings. While many IEEE Societies have similar programs, the SPS provides a substantial amount of financial support for the Chapters to take advantage of this service.

The Distinguished Industry Speaker (DIS) Program provides means for Chapters to have access to individuals who are recognized experts with a background in industrial applications in the signal processing area and are well versed in the ongoing issues/activities in the industry to lecture at Chapter meetings. The main difference and goal of the DIS Program is to educate and interact with Society members about topics that are of primary importance to the industry and the signal processing community-at-large. The DIS Program will supplement and closely mirror the regular DL Program.

Five colleagues were honored with the appointment of DL: Petros Boufounos (Mitsubishi Electric Research Laboratories), Israel Cohen (Technion–Israel Institute of Technology), Janusz Konrad (Boston University), Anna Scaglione (Arizona State University), and Rui Zhang (National University of Singapore).

Five colleagues were honored with the appointment of DIS: Sergio Goma (Technion–Israel Institute of Technology), Neil Gordon (Defence Science and Technology), Pedro Moreno (Google), Ashish Pandharipande (Signify), and Tao Zhang (Starkey Hearing Technologies).

Chapters interested in arranging lectures by a DL or a DIS can visit the following websites:

- DL Program: http://signalprocessingsociety.org/professional-development/distinguished-lecturer-program
- DIS Program: http://signalprocessingsociety.org/professional-development/distinguished-industry-speaker-program.

Lecturers can also be arranged by sending an e-mail to sp.info@ieee.org.

Digital Object Identifier 10.1109/MSP.2018.2877377
Date of publication: 24 December 2018

Petros T. Boufounos

Petros T. Boufounos is a senior principal research scientist and the Computational Sensing Team leader at Mitsubishi Electric Research Laboratories (MERL) and a visiting scholar in the Electrical and Computer Engineering Department at Rice University, Houston, Texas. Dr. Boufounos completed his undergraduate and graduate studies at the Massachusetts Institute of Technology, Cambridge. He received his S.B. degree in economics in 2000, his S.B. and M.Eng. degrees in electrical engineering and computer science (EECS) in 2002, and his Sc.D. degree in EECS in 2006. Between September 2006 and December 2008, he was a postdoctoral associate with the Digital Signal Processing Group at Rice University. Dr. Boufounos joined MERL in January 2009, where he has been heading the Computational Sensing Team since 2016.

Dr. Boufounos is a Senior Member of the IEEE. He has served as area editor (2012–2014) and senior area editor (2014–2018), *IEEE Signal Processing Letters*; member, SigPort Editorial Board (2015–2017); and member, IEEE Signal Processing Theory and Methods Technical Committee (2016–present). He received the SPS Best Paper Award (2015) and the Geoscience and Remote Sensing Society Symposium Paper Award (2014).

Dr. Boufounos' immediate research focus includes signal acquisition and processing, inverse problems, frame theory, quantization, and data representations with applications in compression, sensing, array processing, and lidar, among others. He is also interested in how signal acquisition interacts with other fields that use sensing extensively, such as machine learning, robotics, and dynamical system theory.

His lecture topics include embeddings and information representation, inverse problems in array and multichannel signal processing, depth sensing, and inverse problems in multisensor fusion.

Israel Cohen

Israel Cohen is a professor of electrical engineering at the Technion–Israel Institute of Technology, Haifa. He received his B.Sc. (summa cum laude), M.Sc., and Ph.D. degrees in electrical engineering from the Technion–Israel Institute of Technology in 1990, 1993, and 1998, respectively.

Dr. Cohen served as associate editor, *IEEE Transactions on Audio, Speech, and Language Processing* (2004–2007) and *IEEE Signal Processing Letters* (2004–2008); member, Audio and Acoustic Signal Processing Technical Committee (2012–2017) and Speech and Language Processing Technical Committee (2013–2015).

He is a Fellow of the IEEE "for contributions to the theory and application of speech enhancement." He was awarded the Norman Seiden Prize for Academic Excellence (2017), the SPS Signal Processing Letters Best Paper

Award (2014), the Alexander Goldberg Prize for Excellence in Research (2010), and the Muriel and David Jacknow Award for Excellence in Teaching (2009). He is a coauthor of *Fundamentals of Signal Enhancement and Array Signal Processing* (Wiley-IEEE Press, 2018).

Dr. Cohen's research interests are in the broad area of signal processing, with a specific focus on array processing, statistical signal processing, analysis and modeling of acoustic signals, speech enhancement, noise estimation, microphone arrays, source localization, blind source separation, system identification, and adaptive filtering.

His lecture topics include multichannel signal enhancement, optimal array processing, differential beamforming, robust design of beam patterns, canonical correlation analysis in speech enhancement, Kronecker product beamforming, and array processing in the time domain.

Sergio Goma

Sergio Goma received his B.S., M.S., and Ph.D. degrees from the Universitatea Politehnica Timisoara, Romania, in 1994, 1995, and 1998, respectively, all in computer engineering.

Since 2008, Dr. Goma has been the senior director of technology at Qualcomm Technologies, where he leads the research and development (R&D) and standardization group for imaging. Previously, Dr. Goma was the principal member of the technical staff at AMD and, prior to that, was the architect of the imaging processing solution present in the Imageon series of chips at ATI.

Dr. Goma was a technical subgroup chair of Camera Phone Image Quality (currently IEEE P1858), building and standardizing image quality test metrics and methodologies across the industry to correlate objective test results with human perception and combining these data into a meaningful consumer rating system.

Dr. Goma was the technical lead for the AMD and ATI contributions to the standardization of the serial camera interface MIPI CSI-2SM, which is the most widely used camera interface in the mobile industry. It has achieved widespread adoption for its ease of use and ability to support a broad range of high-performance applications, including 1080p, 4K, 8K, and beyond video and high-resolution photography. His technical solutions were accepted as the core technologies for both the MIPI-CSI2 and MIPI-DSI (e.g., MIPI-specific error-correcting code, system definition, and example implementation of the MIPI-CSI2, and so on), becoming the foundation of the MIPI-CSI2. Also, he drove the interoperability implementations of the MIPI-CSI2 across the industry, as a coauthor of the AMD/ATI CSI2 implementation.

Before ATI, Dr. Goma was designing high-performance charge-coupled device and complementary metal–oxide–semiconductor cameras and imaging systems used in metrology and industrial automation in companies such as Photon Dynamics (currently Orbotech), Taymer Industries, and others.

Dr. Goma is a Senior Member of the IEEE. He serves as an associate editor of *IEEE Transactions on Image Processing* (2013–present), *IEEE Transactions on Computational Imaging* (2015–present), and *Springer Journal of Real-Time Image Processing* (2009–present) and a committee member of the Society for Imaging Science and Technology's (IS&T) Human Vision and Electronic Imaging Conference (2008–present) and Photography, Mobile, and Immersive Imaging Conference (2007–present). He has also performed as vice president of the IS&T (2014–2018) and committee member of the Computational Imaging Special Interest Group since its inception in 2015 until it became the Computational Imaging Technical Committee, where he continues to serve as an associate member.

Dr. Goma received the 2014 IS&T Service Award for operating as the 2014 Electronic Imaging Symposium chair and for leading the efforts for IS&T sole management of the symposium beginning in 2016.

His research interests include programmable hardware architectures and hardware accelerators for imaging, computational photography and integrated imaging sensors, image quality, image processing, and computer vision. Dr. Goma's lecture topics include integrated image sensors with compute elements and programmable architectures for image processing.

Neil Gordon

Neil Gordon received a B.Sc. degree in mathematics and physics from the University of Nottingham, United Kingdom, in 1988 and a Ph.D. degree in statistics from Imperial College, University of London, United Kingdom, in 1993. He was with the Defence Evaluation and Research Agency in the United Kingdom from 1988 to 2002, working on statistical signal processing for a range of defense applications. In 2002, he moved to Defence Science and Technology (DST), part of the Department of Defence in Australia, where he currently leads the Intelligence Analytics Branch. Dr. Gordon's team of scientists and engineers conducts R&D to improve the situation awareness of intelligence analysts by extracting, fusing, and disseminating meaningful content from a wide range of diverse data sources and types. He provides expert scientific and technical advice to the Department of Defence and other commonwealth agencies. He led the DST team by providing scientific advice to the Australian Transport Safety Bureau MH370 search.

Dr. Gordon is an IEEE Senior Member and a fellow of the Royal Statistical Society. He was an associate editor of *IEEE Signal Processing Letters* (2015–2017) and has given invited plenary lectures at the ICASSP (2015) and the International Fusion Conference (2004 and 2018). He has coauthored three books, four book chapters, and more than 90 peer-reviewed journal articles and conference papers.

Dr. Gordon's main areas of research are particle filters and Bayesian methods for nonlinear estimation, data, and information fusion in distributed sensor networks. His lecture topics include beyond the Kalman filter: 25 years of particles and other random points and signal processing and the search for MH370.

Janusz Konrad

Janusz Konrad received a master's degree from the Technical University of Szczecin, Poland, in 1980 and a Ph.D. degree from McGill University, Montréal, Canada, in 1989, both in electrical engineering. He joined INRS-Télécommunications, Montréal, Canada, as a postdoctoral fellow (1989–1991) and then as a faculty member (1991–2000). Subsequently, he moved to Boston University, Massachusetts, where he is currently a professor in the Department of Electrical and Computer Engineering.

Dr. Konrad is an IEEE Fellow. He was awarded the IEEE Signal Processing Magazine Best Paper Award (2001) and the EURASIP Image Communication Best Paper Award (2004–2005). He was also a corecipient of the Best Paper Award at the IEEE International Conference on Advanced Video and Signal-Based Surveillance (2010) and a member of the winning team in the Aerial View Activity Classification Challenge at the International Conference on Pattern Recognition (2010).

Dr. Konrad served as a member-at-large on the conference board of the IEEE SPS (2015–2016) and a member of the IEEE Image, Video, and Multidimensional Signal Processing Technical Committee (2000–2006). He is currently a member of the Steering Committee of the IEEE International Conference on Advanced Video and Signal Based Surveillance (2014–present). His service on editorial boards includes *IEEE Transactions on Image Processing* (1996–2000 and 2013–2016 as associate editor and since 2017 as a senior associate editor), *EURASIP Signal Processing: Image Communication* (2011–present), *IEEE Communications Magazine* (1998–2012), *EURASIP Journal on Image and Video Processing* (2006–2010), and *IEEE Signal Processing Letters* (2002–2004). He was the general chair of the IEEE International Conference on Advanced Video and Signal Based Surveillance (2015) and served on organizing committees of many IEEE conferences.

Dr. Konrad's interests include video processing and computer vision, stereoscopic and three-dimensional imaging and displays, visual sensor networks, human–computer interfaces, and cybersecurity. His lecture topics include privacy-preserving localization and recognition of human activities, user authentication for natural user interfaces, and toward autonomous video surveillance.

Pedro J. Moreno

Pedro J. Moreno received his Ph.D. degree in electrical and computer engineering at Carnegie Mellon University, Pittsburgh, Pennsylvania, in 1996 and his telecommunications engineering degree at the Universidad Politecnica de Madrid, Spain, in 1986. Prior to joining Google, Dr. Moreno held a research scientist position at HP Labs, where he led research in audio mining and searching.

Dr. Moreno leads a team of 50 engineers and scientists in the languages modeling group, part of the Speech team at Google. His team is in charge of deploying speech-recognition services in all supported languages and improving their quality. His team has pioneered the use of context signals in speech-recognition systems. He has been involved in building the Google technology behind every voice-activated application that is used by billions of users in more than 100 languages every day.

He is a Member of the IEEE. His research interests include signal processing, machine learning, and statistical modeling with applications to speech processing. He has published more than 100 well-cited publications in several conferences and journals and holds several patents to his name. Dr. Moreno's lecture topics include speech recognition, language modeling, and voice assistants.

Ashish Pandharipande

Ashish Pandharipande received his B.E. degree in electronics and communications engineering from Osmania University, Hyderabad, India, in 1998 and his M.S. and Ph.D. degrees in electrical and computer engineering from the University of Iowa, Iowa City, in 2000, 2001, and 2002, respectively.

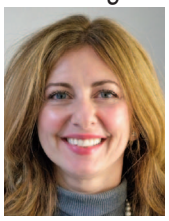
Dr. Pandharipande has since been a postdoctoral researcher at the University of Florida, Gainsville; a senior researcher at the Samsung Advanced Institute of Technology; and a senior scientist at Philips Research. He has held visiting positions at AT&T Laboratories, New Jersey, and the Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore. He is currently the lead R&D engineer at Signify (the new company name of Philips Lighting) in Eindhoven, The Netherlands.

Dr. Pandharipande is a Senior Member of the IEEE. He has served as associate editor, *IEEE Transactions on Signal Processing* (2012–2015), *IEEE Sensors Journal* (2012–present), *IEEE Signal Processing Letters* (2016–present), and *IEEE Journal of Biomedical and Health Informatics* (2014–present); and member, International Advisory Board, *Lighting Research & Technology Journal* (2010–present).

His research interests are in sensing, wireless communications, controls, data analytics, and signal processing applications in domains like smart lighting, energy monitoring and control, and cognitive spectrum sharing. He has more than 160 scientific publications and approximately 90 patents/filings in these areas.

Dr. Pandharipande's lecture topics include sensing technologies and applications in smart lighting and beyond; sensor-driven smart lighting controls; machine learning in connected lighting: applications and opportunities; and sensing, visible light communication, and illumination control in LED lighting systems.

Anna Scaglione



Anna Scaglione received her M.Sc. degree from the University of Rome "La Sapienza," Italy, in 1995 and her Ph.D. degree in 1999. She is currently a professor in electrical and computer engineering at Arizona State University, Phoenix. She was previously a professor of electrical engineering at the University of California at Davis (2008–2014) and Cornell University, New York (2001–2008). Prior to joining the engineering faculty at Cornell, Dr. Scaglione was an assistant professor at the University of New Mexico, Albuquerque (2000–2001).

Dr. Scaglione was elected an IEEE Fellow in 2011. She served as editor-in-chief, *IEEE Signal Processing Letters* (2012–2013); associate editor, *IEEE Transactions on Wireless Communications* (2002–2005); Editorial Board member, *IEEE Transactions on Signal Processing* (2008–2010); area editor, *IEEE Transactions on Signal Processing* (2010–2011); senior editor, *IEEE Transactions on Control of Networked Systems*; general chair, IEEE International Workshop on Signal Processing Advances in Wireless Communications (2005); member, Signal Processing for Communications and Networking Technical Committee (2004–2009); Steering Committee member, IEEE SmartGridComm Conference (2010–2015); and member-at-large, IEEE SPS Board of Governors (2012–2014).

Dr. Scaglione received the IEEE Signal Processing Best Paper Award (2000) and the IEEE Donald G. Fink Prize Paper Award (2013). Her research with her students was also honored with the IEEE SPS Young Author Best Paper Award (Lin Li) (2013) and three conference best paper awards: the Ellersick Best Paper Award (2005) at the International Conference for Military Communications, the Student Best Paper Award at SmartGridComm (2014), and the Student Best Paper Award (2017) ICASSP. She was also a recipient of the National Science Foundation CAREER Award (2002).

Dr. Scaglione's expertise is in the broad area of statistical signal processing for communication, electric power systems, and information and social networks. Her current research focuses on studying and enabling decentralized learning and signal processing in networked systems.

Her lecture topics include distributed signal processing, opinion dynamics in social networks, networked system identification, cooperative transmission in networked systems, distributed synchronization and scheduling via pulse-coupled oscillators model, and signal processing in energy systems.

Rui Zhang

Rui Zhang received his B.Eng. and M.Eng. degrees from the National University of Singapore in 1999 and 2001, respectively, and his Ph.D. degree from Stanford University, California, in 2007, all in electrical engineering. From 2007 to 2009, he worked as a research scientist at the Institute for Infocomm Research, ASTAR, Singapore. In 2010, he joined the Department of Electrical and Computer Engineering at the National University of Singapore, where he is now a Dean's chair associate professor with the faculty of engineering.

Dr. Zhang is a Fellow of the IEEE. He was the recipient of the IEEE Communications Society Asia-Pacific Region Best Young Researcher Award (2011) and the Young Researcher Award of National University of Singapore (2015). He was a corecipient of the IEEE Marconi Prize Paper Award in Wireless Communications (2015), the IEEE Communications Society Asia-Pacific Region Best Paper Award (2016), the IEEE SPS Best Paper Award (2016), the IEEE Communications Society Heinrich Hertz Prize Paper Award (2017), the IEEE SPS Donald G. Fink Overview Paper Award (2017), and the IEEE Technical Committee on Green Communications & Computing Best Journal Paper Award (2017). His coauthored paper received the IEEE SPS Young Author Best Paper Award (2017).

Dr. Zhang served as the guest editor for three special issues in *IEEE Journal of Selected Topics in Signal Processing* and *IEEE Journal on Selected Areas in Communications*. He also served as member, IEEE Signal Processing for Communications and Networking Technical Committee (2012–2017); and IEEE Sensor Array and Multichannel Technical Committee (2013–2015); vice chair, IEEE Communications Society Asia-Pacific Board Technical Affairs Committee (2014–2015); and editor, *IEEE Transactions on Wireless Communications* (2012–2016), *IEEE Journal on Selected Areas in Communications—Green Communications and Networking Series* (2015–2016), and *IEEE Transactions on Signal Processing* (2013–2017). He is now an editor of *IEEE Transactions on Communications* and *IEEE Transactions on Green Communications and Networking*. He also serves as a member of the Steering Committee for *IEEE Wireless Communications Letters*.

Dr. Zhang's current research interests include wireless information and power transfer; drone communication, wireless eavesdropping and spoofing; energy-efficient and energy-harvesting-enabled wireless communication; multiuser multiple-input, multiple-output (MIMO); cognitive radio; and optimization methods.

His lecture topics include signal processing and optimization in unmanned aerial vehicle communication and trajectory design, MIMO communication and signal processing with passive intelligent surface, and signal and system design for wireless information and power transfer.

Tao Zhang

Tao Zhang attended Nanjing University, China, from 1982 to 1986 and received his B.S. degree in physics in 1986; attended Peking University, Beijing, China, from 1986 to 1989 and received his M.S. degree in electrical engineering in 1989; and attended The Ohio State University, Columbus, from 1991 to 1995 and received his Ph.D. degree in speech and hearing science in 1995. He joined the Advanced Research Department at Starkey Laboratories, Inc. as a senior research scientist in 2001, managed the Digital Signal Processing (DSP) Department at Starkey Laboratories, Inc. from 2004 to 2008, and managed the Signal Processing Research Department at Starkey Laboratories, Inc. from 2008 to 2014. He has been the director of the Signal Processing Research Department at Starkey Hearing Technologies since 2014.

Dr. Zhang is a Senior Member of the IEEE and a member of the IEEE Engineering in Medicine and Biology Society. He is also a member of the IEEE Audio and Acoustics Signal Processing Technical Committee (2014–present), IEEE Industrial Relationship Committee (2014–present), IEEE Communications Society North America Region Board (2018–present), and the IEEE Industry Convoy for the United States, Regions 1–6 (2017–present). He has been the chair of the IEEE Twin-Cities Signal Processing and Communication Chapters since 2013.

Since 2001, Dr. Zhang has been actively promoting education and research in the field of hearing research and technology in the global research community. He has hosted many IEEE distinguished lectures and organized many joint IEEE and Starkey research seminars in hearing research and technology. He has been a sponsor for best paper awards and other student awards for ICASSP European Signal Processing Conference, the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, the Joint Workshop on Hands-free Speech Communication and Microphone Arrays, and the International Engineering in Medicine and Biology Conference. He was the organizer of global research reception for audio, acoustic, and speech signal processing researchers for hearing instruments at ICASSP 2013, 2015, and 2016. Dr. Zhang has received many prestigious awards from Starkey Hearing Technologies, including the Outstanding Technical Leadership Award (2003), the Engineering Service Award (2007), the Most Valuable Idea Award (2009), the Mount Rainier Best Research Team Award (2016), and the Inventor of the Year Award (2018).

Dr. Zhang's current research interests include audio, acoustic, speech, and music signal processing; multimodal signal processing and machine learning for hearing enhancement and health and wellness monitoring; psychoacoustics; room and ear canal acoustics; ultralow-power-embedded system design; and real-time fixed-point (DSP) algorithm design. He has authored more than 120 presentations and publications, received over 20 approved patents, and has approximately 30 more patents pending.

His lecture topics include tackling the cocktail party problem for hearing devices; intelligent hearing aids: the next revolution; robust and practical acoustic feedback control for hearing devices; practical challenges, current solutions, and future directions for hearing devices; and multimodal signal processing and machine learning for multipurpose ear-level devices.

SP

John Edwards

Growing Security and Privacy Threats Inspire New Approaches

Signal processing is playing a crucial role in the development of next-generation protection technologies

As threats to data security and privacy multiply, researchers worldwide are investigating novel approaches to protect critical assets, including financial accounts, medical histories, and personal identities. In this effort, signal processing is playing an important role across virtually all security technologies, helping to make new protection tools function efficiently and flawlessly.

At the Massachusetts Institute of Technology (MIT), researchers are focusing on securing the data transmitted across online neural networks without dramatically hindering runtimes. The approach, dubbed *Gazelle*, holds promise for using cloud-based neural networks in medical-image analysis and other applications that require secure data, says Chiraag Juvekar, a Ph.D. student in MIT's Department of Electrical Engineering and Computer Science (EECS) and the first author of a paper on the technology.

Outsourcing machine learning is a growing industry trend. Many firms have deployed cloud platforms designed to handle computation-heavy tasks, such as running data through a convolutional neural network (CNN) for image classification. Users can upload data to those services for a fee and get back results in only a few hours.

The growing popularity of cloud-based machine learning raises a natural question about what privacy guarantees can be provided in such a setting. "Our work tackles this problem in the context where a client wishes to classify private images using a CNN trained by a server," Juvekar explains. "Our goal is to build efficient protocols whereby the client can acquire the classification result without revealing their input to the server, while guaranteeing the privacy of the server's neural network."

Gazelle was built with a focus on building efficient and fast protocols. Juvekar explains that one reason the name *Gazelle* was chosen was to underscore the speed of the proposed protocols, which is more than an order of magnitude faster than the state of art.

Juvekar envisions several potential applications for Gazelle. A prime use would be for protecting extremely private information, such as medical data. "Being able to run classifications on patient data without requiring access to the plain text would be very helpful," he notes. An additional advantage is protection against adversarial third-party breaches the cloud's security because the cloud service provider never needs plain-text access to the data. In such a scenario, the attacker can only get access to encryptions of user data, since only the user has access to the plain text, he explains.

The system blends two techniques—homomorphic encryption and garbled circuits—in a way that helps networks run much faster than they do with conventional approaches. Juvekar believes the work is unique from a signal processing point of view in two ways. At the lower level, the encryption scheme involves a lot of finite field algebra and lattices, which traditionally fall in the domain of coding theory, he states. "At a higher level, we require a deep understanding of neural networks themselves, which involve convolutions and matrix algebra."

The homomorphic encryption schemes require the computation of number theoretic transforms (NTTs). These are essentially a finite field analog of the classic Fourier transform. "Traditional parameters for these schemes used a prime-sized NTT, which necessitated a slower Bluestein algorithm for computing the NTT," Juvekar says. "Our work shows how to implement the required encrypted computation using a power-of-two-sized NTT." This approach enabled researchers to take advantage of the classic Cooley–Tukey algorithm for the NTT, resulting in a substantial speed boost.

Previous approaches to evaluating encrypted neural networks have typically changed the network structure substantially. "One of our main contributions is a technique to evaluate existing neural networks in the encrypted domain without having to retrain them," Juvekar says.

Juvekar believes the main advantage stemming from the research is the ability to run machine-learning inference, particularly neural network inference for image classification, while simultaneously maintaining the privacy of both the

Digital Object Identifier 10.1109/MSP.2018.2873823
Date of publication: 24 December 2018

user's input data and the server's model (Figure 1). "At the lowest level of the hierarchy, we realized that a major source of inefficiency in implementing homomorphic encryption schemes was the need for fast modular arithmetic in order to implement all the finite field algebra," Juvekar says. To circumvent that issue, he explains, researchers proposed new concrete parameters for selection of efficient finite fields compatible with homomorphic encryption algorithms where the modular reduction operation can be computed using just multiplications and additions, resulting in faster modular arithmetic.

Given that more and more private data are being uploaded to the cloud, maintaining privacy is an increasingly pressing concern. Concerted effort on multiple fronts will be required to protect privacy, Juvekar predicts. "On the technical front, we will definitely need more research both from a theoretical and practical standpoint." Yet he believes there's social and legal angles that need to be considered, with access to such tools as homomorphic encryption being equally important.

Juvekar notes that the project's original goal was to build a system for secure neural network inference. "While we have completed that phase of the project, there is definite interest in continuing this work further," he says. Potential areas of further research may include ways to scale up the current system, develop new algorithms, and create hardware better able to handle higher speeds.

The project's coresearchers include Vinod Vaikuntanathan, an associate professor in EECS and a member of the MIT Computer Science and Artificial Intelligence Laboratory, and Anantha Chandrakasan, dean of the MIT School of Engineering and the Vannevar Bush professor of electrical engineering and computer science.

Healthy and secure

Ensuring the privacy and security of personal medical data collected and stored by wearable devices is the goal of a team of Arizona State University researchers. The team, led by Jae-sun Seo, an assistant professor in the School of Electrical, Computer, and Energy Engineering, has developed a prototype chip that takes advantage of the unique characteristics of each individual's electrical heartbeat signals to enable secure biometric authentications using electrocardiographic (ECG) information.

A growing number of wearable health-monitoring devices incorporate sensors designed to detect physiological signals. The new technology exploits the ECG sensors already present on wearable devices, Seo says.

Seo notes that although ECG signals look quite similar from person to person, they are never the same. The researchers' new chip uses signal processing and relatively simple neural networks to extract features that differ the most between individuals (Figure 2).

Seo reports that the research addresses an important need. According to data from the U.S. Federal Bureau of Investigation, medical information available on the black market is worth approximately ten times more than credit card data. Health insurance information, for instance, can be used to purchase drugs or medical equipment, which are then resold illegally, or even to obtain medical care.

For feature extraction and authentication using ECG signal processing to be seamlessly integrated into wearable products, an accurate algorithm must be implemented within a small device that consumes very little power. The 65-nm complementary metal–oxide–semiconductor prototype chip consumes 1.06 μW at 0.55 V for real-time ECG authentication.

The technology is designed to authenticate continuously, ensuring that the person wearing the device is the legitimate owner and not someone who may have stolen it. To perform authentication using an individual's ECG signal, a representative ECG beat needs to be

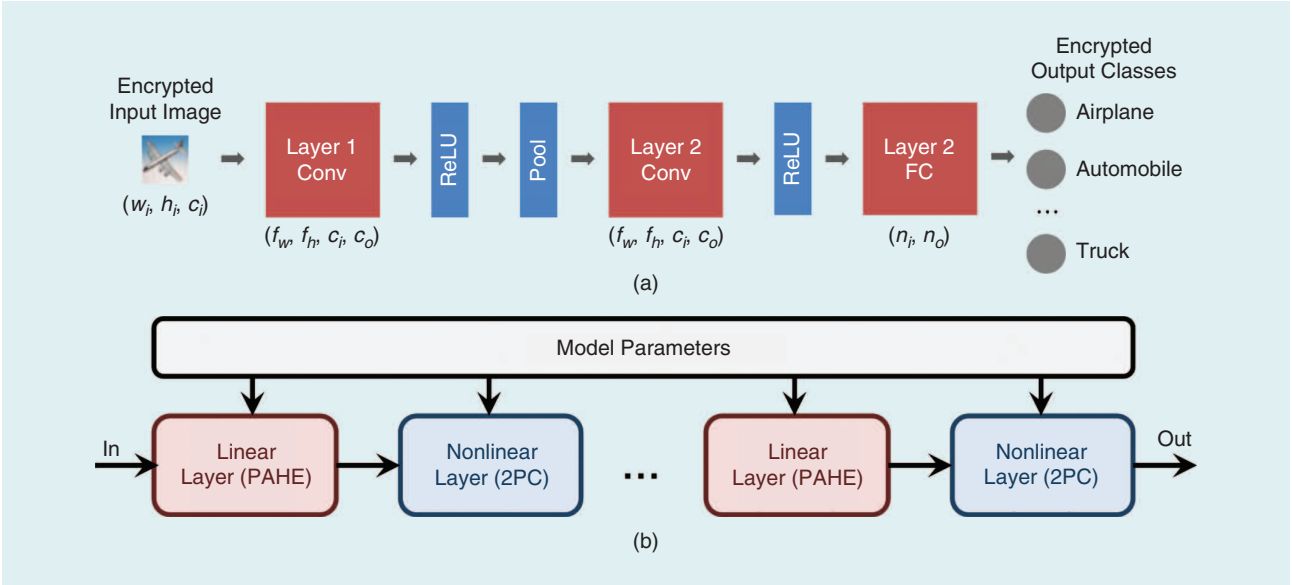


FIGURE 1. The (a) neural network view and (b) Gazelle view. ReLU: rectified linear unit. Conv: convolution; PAHE: packed additively homomorphic encryption; 2PC: two-party computation.

computed and extracted for each individual, Seo says. Yet raw ECG signals include a lot of noise, tend to exhibit dc wandering, and can be affected by motion artifacts, he notes.

The researchers extract various ECG features by filtering the raw signals with different cutoff frequencies and aligning the signals differently around the detected R-peak of the ECG signal. After that, the filtered and aligned ECG data go through four parallel neural networks that are pretrained with a 645-subject in-house database. The concatenated output of four neural networks are used as the final features for ECG authentication, Seo explains.

Since the project's primary objective was to develop low-power hardware for ECG authentication, the researchers worked to lower the precision of the various finite-impulse response (FIR) filters used. "To prevent degradation in the error rate, we found out that FIR filter signals and R-peak detection need to maintain relatively high precision of 13 bits," Seo notes. "On the other hand, the compressed neural network weights could be reduced to 6-bit precision."

Seo states that the current prototype, verified with the data set of 645 users, provides real-time ECG authentication with very low power, enabling seamless integration into wearable devices. Developing a commercial version of the technology, however, will require additional validation and verification of quality assessment, Seo says.

The research was conducted in collaboration with Samsung Advanced Institute of Technology in South Korea.

Fighting facial recognition

Facial recognition is widely recognized as a valuable security technology capable of protecting everything from smartphones, cashpoints/ATMs, to military and industrial facilities. Yet the technology also has its dark side, such as when it's used by governments to identify and punish peaceful political protestors or by businesses to invade customers' personal privacy.

With the aim of blocking the misuse of such technology, University of Toronto researchers, led by Parham Aarabi, an associate professor of communications and computer engineering, and Avishek Bose, a graduate researcher, have developed an algorithm that's specifically designed to dynamically disrupt facial recognition systems.

The team's approach is based on adversarial training, a deep-learning technique that pits two artificial-intelligence algorithms against each other, with each learning from the other as well as opposing the other. "Essentially, there are two networks," Bose says. A face detector perceives faces and localizes them with boxes in images. A generator network takes the facial images and adds a small amount of noise crafted to fool the face detector into misclassifying the face as the background. "The generator network is trained against the face detector with the objective of creating these perturbations to fool the face detector but, at the same time, ensuring that the perturbations are small enough that humans can still recognize the face with ease," he explains (Figure 3).

Unlike conventional signal processing, where the use of Kalman filters and other techniques requires explicit knowledge of the task and, to a certain extent, custom features, the researchers focused on a deep-learning-based approach. For the Faster Regions with CNN (R-CNN) detector, this means first extracting relevant image features through a pretrained CNN, which has already learned features, such as edges, shapes, and textures, by being trained on a very large data set. Using a pretrained CNN as a feature extractor makes it possible to focus on the relevant features needed for face detection, Bose says.

In the next stage of Faster R-CNN, a region proposal network (RPN) is employed by convolving small filters over the extracted features. The RPN, through training, identifies regions in

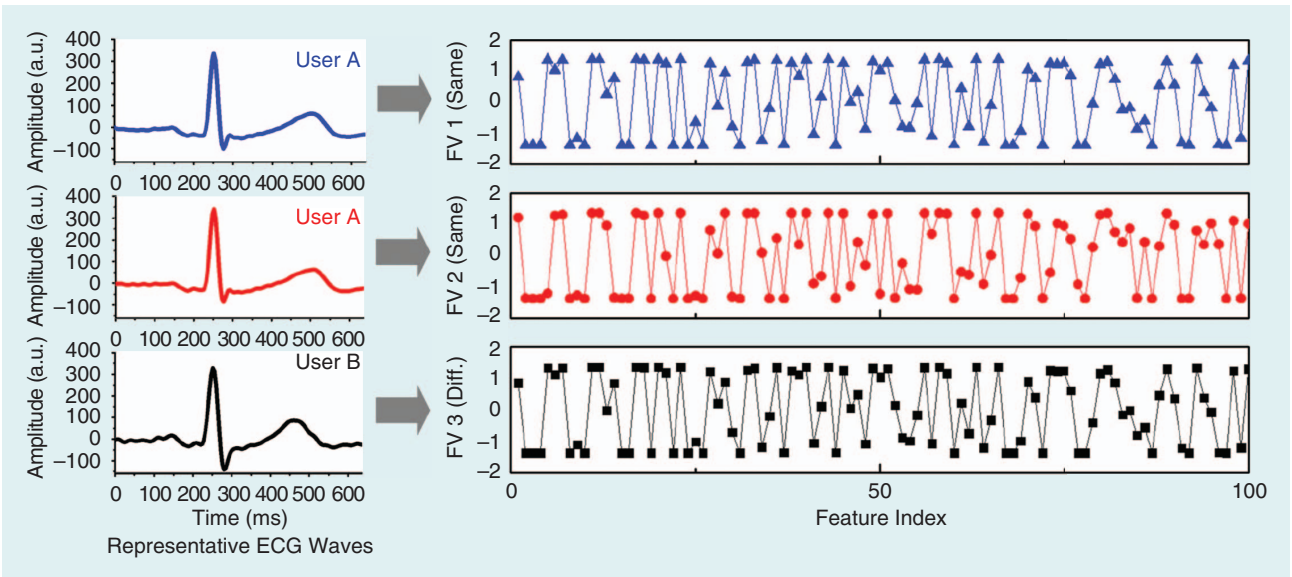
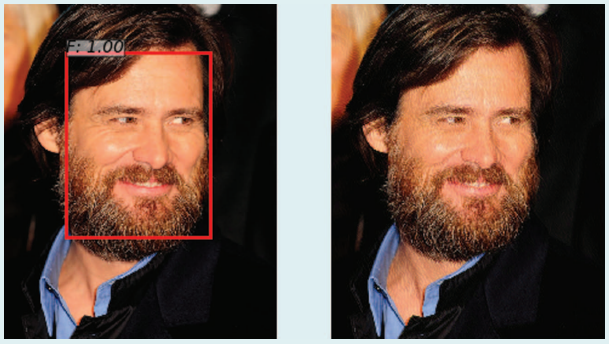


FIGURE 2. The new chip uses signal processing and relatively simple neural networks to extract ECG features that differ the most between individuals. FV: feature vectors.



(a) (b)

FIGURE 3. (a) A face detector perceives faces and localizes them with boxes in images. (b) A generator network adds enough noise to the image to fool the face detector while still maintaining the image as recognizably a face to the human eye. (Photo courtesy of the University of Toronto.)

the original image that correspond to the abstract notion of an object and its location. The RPN then proposes many of these regions—often overlapping—to the detection network, which is another CNN that's geared to identify actual faces from the proposed regions. "The filters learned by this detection network are only for the face-detection task and it is learned automatically through training via stochastic gradient descent," Bose says. "Similarly, the generator network is also a CNN, but is much smaller in comparison." The filters "learned" by the generator are primarily based on the mistakes made by the face-detector network. "The generator tries to amplify the mistakes made by the detector through the perturbations it generates," he notes.

"Signal processing is at the core of what we do," Aarabi observes. "What is especially interesting is the combination of machine learning and signal processing." That combination can be applied in such areas as convolutional neural networks, technologies for encoding and decoding deep neural network architectures, and analyses of signals and images, Aarabi says.

"One of the main benefits of this research is that we can come up with compelling privacy filters that can be applied to face images and be effective against face detectors," Bose says. He suggests the technology might be useful, for example, in an online setting to protect people from malicious efforts to automatically mine personal data.

Bose says that the team's research represents a step toward technology that can enable the average user to take back ownership of the digital data they produce. "In terms of breaking face detectors, we are the first ones to do so, to the best of our knowledge," Bose reports. He also notes, however, that the general field of adversarial attacks is vast and calls for scientists to follow any number of interesting research paths.

Aarabi and Bose tested their system using an industry-standard pool of more than 600 faces that encompassed a wide range of ethnicities, lighting conditions, and environments. The results showed that their system could reduce the proportion of faces that were originally detectable from nearly 100% down to 0.5%.

The researchers have demonstrated that they can break the class of face detectors that they have access to. However, Bose states, there are many other classes of face detectors that the researchers can't access. "Future research in this area is coming up with ways to attack these 'black box' face detectors," he says. "Furthermore, for our attack to be effective in a commercial setting, we must train it to simultaneously attack multiple detectors."

The current research should be considered more of a first step rather than a finished product, Bose notes. "We are the first to show attacks against a certain class of face detectors in a white-box setting," he says. "To have the most impact, future research needs to be done against detectors in a black-box setting."

Author

John Edwards (jedwards@johnedwardsmedia.com) is a technology writer based in the Phoenix, Arizona, area. Follow him on Twitter @TechJohnEdwards.

IEEE connects you to a universe of information!

As the world's largest professional association dedicated to advancing technological innovation and excellence for the benefit of humanity, the IEEE and its Members inspire a global community through its highly cited publications, conferences, technology standards, and professional and educational activities.

Visit www.ieee.org.

IEEE

Publications / IEEE *Xplore*® / Standards / Membership / Conferences / Education

IMAGE LICENSED BY INGRAM PUBLISHING

Meinard Müller, Bryan A. Pardo, Gautham J. Mysore,
and Vesa Välimäki

Recent Advances in Music Signal Processing

Music is a ubiquitous and vital part of our lives. Thanks to the digital revolution in music distribution and storage, music has become one of the most popular categories of multimedia content. In general terms, music processing research contributes concepts, models, and algorithms that extend our capabilities of accessing, analyzing, manipulating, and creating music. Given the complexity and diversity of music, researchers must account for various aspects, such as the genre, instrumentation, dynamics, tempo, rhythm, and timbre. Music signals typically comprise a wide range and large number of different sound sources. Postprocessing and the use of audio effects in the mixing and mastering stages may further complicate the analysis of recorded musical material. Furthermore, music is inherently multimodal, incorporating speech-like signals (singing), video (of live performances), and still images (scanned music scores). This wealth of data makes music processing a challenging field of research and closely connected to areas such as audio and acoustic signal processing, multimedia signal processing, and machine learning.

Compared with speech processing, a research field with a long tradition, music processing is still a relatively young discipline, but it is rapidly growing. In recent decades, the music processing community has come together by organizing major annual conferences on topics including music information retrieval, sound and music computing, audio-effects processing, computer music, and applications in audio engineering. Although computer-based music research has traditionally been conducted using symbolic representations, the research focus has shifted to other types of music-related data including audio recordings, digitized images, music videos, and other types of sensor data. As a consequence, digital signal processing has found its way into many research communities dealing with music-related data.

In this special issue of *IEEE Signal Processing Magazine (SPM)*, we survey recent advances in music processing with a focus on audio signals. Eleven articles cover topics including music analysis, retrieval, source separation, singing-voice processing, musical sound synthesis, and user interfaces, to name a few. The tutorial-style articles provide an overview of theory and applications and discuss main advances. Although music processing has benefited a lot from traditional fields, such as signal processing, we hope to convince the reader that the rich and challenging problem domain of music also has many things to offer to signal processing and other research disciplines.

We start the special issue with the classic problem of music transcription, where the objective is to extract note events, key signature, time signature, instrumentation, and other score parameters from a given music recording. "Automatic Music Transcription" by Benetos et al. gives an overview of computational algorithms for converting a music signal to written music notation, with an emphasis on approaches for transcribing polyphonic music produced by pitched instruments and voice. The article details the methodology used in the two main families of approaches: those based on deep learning and those based on nonnegative matrix factorization (NMF). It also links automatic music transcription to other problems found in the broader field of digital signal processing, including multiple-F0 estimation, instrument recognition, and source separation.

Music signals contain complex mixtures of different yet highly correlated sound sources, which creates a challenge for processing. For example, a singer, a guitarist, a keyboard player, and a drummer may be active at the same time, following the same rhythmic pattern, and playing notes that are harmonically related. In "Musical Source Separation," Cano et al. review the problem of recovering the individual tracks as if they had been played in isolation. The article discusses sound characteristics of music signals and how these characteristics can be exploited to develop appropriate source-separation algorithms. Furthermore, it covers various key techniques including kernel additive models, NMF, sinusoidal models, and deep neural networks. Although music source separation has significantly progressed over the last decade,

Digital Object Identifier 10.1109/MSP.2018.2876190
Date of publication: 24 December 2018

there are still numerous open problems, including the quality assessment of separated sources.

In the last ten years, music streaming has become the predominant way of accessing and consuming music. Along with providing content, music streaming services also give personalized recommendations based on play history or collaborative filtering. In this context, song descriptors, such as genre, mood, instrumentation, and vocal quality, are needed. Obtaining these descriptors by manual annotation is a costly and time-consuming process that does not scale to huge music collections. Therefore, computational approaches for music content understanding—including music genre classification, music mood classification, and music autotagging—have become a major strand of research. The article "Deep Learning for Audio-Based Music Classification and Tagging" by Nam et al. provides an up-to-date survey of the deep network designs tailored for music classification and tagging tasks. It covers best practices and applications to music services and discusses the limitations of current approaches and open issues in this area of research.

Compared with speech processing, a research field with a long tradition, music processing is still a relatively young discipline, but it is rapidly growing.

The rapid growth of digitally available music data goes beyond audio recordings and includes other modalities, such as digitized images of sheet music, album covers, liner notes, and video clips. The following three articles are all concerned with exploiting, linking, and jointly analyzing this wealth of data. "Cross-Modal Music Retrieval and Applications" by Müller et al. reviews several cross-modal retrieval scenarios, with a particular focus on sheet music (visual domain) and audio (acoustic domain). Given a query in one modality, the task is to retrieve semantically corresponding documents in some other modality. By bridging the gap between various music representations, this technology enables music navigation and browsing applications, including the classic problem of automated score following. Besides traditional approaches based on musically motivated features, this article also discusses generalized audio fingerprinting and recent data embedding techniques based on deep learning.

The article "Audiovisual Analysis of Music Performances" by Duan et al. shows the growing significance of jointly analyzing audio and visual data for music, with a specific focus on music performance. This cross-modal approach is of particular relevance for tasks, such as tracking a musician's fingering or a conductor's gestures, for which the video provides information that is complementary to the audio signal. The article provides an overview of analyzing performances in which the audio and video have both static correspondences (such as musicians in an orchestra whose relative positions do not change) and dynamic correspondences (such as vibrato analysis).

Goto and Dannenberg's article, "Music Interfaces Based on Automatic Music Signal Analysis," shows how music processing techniques open up new possibilities for developing interactive music systems and Web services. Such interfaces include audiovisual elements based on structural segments, beats, melody line, and chords to simplify audio navigation or enrich listening experience. As for music production, intelligent audio editors allow users to identify and rearrange drums patterns and other score-based sound events.

The human voice plays an integral part in nearly all music cultures. Therefore, it is not surprising that the analysis, synthesis, and classification of music signals that involve the human voice is at the core of music signal processing. Singing not only differs fundamentally from spoken language but often occurs in a polyphonic context and in combination with other instruments. Thus, it requires computational approaches that are different from those used in speech processing. The article "An Introduction to Signal Processing for Singing-Voice Analysis" by Humphrey et al. covers fundamentals of the human voice and vocalization techniques. Based on time–frequency patterns specific to singing, various music processing tasks, such as singer activity detection, melody estimation, genre classification, and intonation estimation, are covered. Furthermore, the article highlights the unique difficulties that are faced when dealing with music, where tasks such as language identification, audio–lyrics alignment, and lyrics transcription (tasks that are also well known in speech processing) become extremely challenging.

While the article by Humphrey et al. treats singing-voice processing from an analysis perspective, the contribution "Speech-to-Singing Voice Conversion" by Vijayan et al. approaches this topic from a synthesis perspective. It covers the problem of converting a speaking voice into a singing one while preserving the linguistic content and the speaker's vocal identity. While discussing template- and model-based key techniques for singing synthesis, the article highlights the differences between singing and speaking in terms of dynamic range, pitch variations, and duration of the linguistic content.

Continuing with music synthesis, the article "Model-Based Digital Pianos" by Bank and Chabassier addresses the task of developing digital sounds that mimic the sound of an acoustic piano. The authors review the physics of the piano and then introduce a comprehensive physical model considering hammer motions, string vibrations, soundboard, and sound radiation. Such models can become prohibitively expensive to compute. Therefore, when developing real-time sound-synthesis applications, one requires model simplifications that do not sacrifice perceptual quality. The article reviews optimization approaches that neglect inaudible phenomena and enhance the physical models with perceptually appropriate modifications.

Nogueira et al. shine another light on music signals and their properties in their article "Making Music More Accessible for Cochlear Implant Listeners" by adopting the perspective of

hearing-impaired listeners. Although cochlear implants (CIs) enable users to understand continuously spoken speech to a high degree, music features, such as pitch and timbre, are still poorly transmitted. The contribution discusses various signal processing methods (e.g., the amplification of a song's melody or the simplification of the music content) that make music more accessible and enjoyable for CI users. Finally, the authors address the problem of objective and subjective evaluation measures used to assess the quality of the transmitted music signal.

In applied research areas, such as music signal processing, theoretical concepts need to be implemented and evaluated using real-world data in concrete application scenarios. Implementation details can have a substantial impact on the overall performance of a given system. In practice, it is often nearly impossible to specify all relevant factors of an experiment, including design choices in signal processing and machine-learning approaches, the exact composition of the data collection used, and implicit assumptions in the evaluation process. In "Open-Source Practices for Music Signal Processing Research," McFee et al. give recommendations on best practices for open-source software development in the context of music information retrieval applications. Furthermore, they lay out future directions for incorporating open-source and open-science methodology—a topic that pervades all data-driven areas of signal processing.

Many people contributed to compiling this special issue, which is the result of a team effort throughout the whole research community. First, we thank *SPM*'s Special Issues Area Editor Douglas O'Shaughnessy, IEEE Signal Processing Society Publications Administrator Rebecca Wollman, and Managing Editor Jessica Welsh for helping with the organization and production of this issue. Furthermore, we thank the many reviewers for their detailed and constructive comments throughout several rounds of revision. Last but not least, we thank the authors for writing tutorial-style articles and finding a way to be comprehensive and compact, instructive, and entertaining, while covering basic principles and state-of-the-art techniques.

Music signal processing is an exciting and challenging area of research. Music not only connects people but also relates to many different research disciplines. This interdisciplinary perspective is becoming commonplace in music technology research labs and enabling novel algorithm development. We hope that this special issue provides examples of such work and inspires new ideas that cross boundaries of the IEEE Signal Processing Society and other fields.

Meet the guest editors

Meinard Müller (meinard.mueller@audiolabs-erlangen.de) received his diploma degree in mathematics in 1997 and his Ph.D. degree in computer science in 2001, both from the University of Bonn, Germany. Since 2012, he has held a professorship for semantic audio signal processing at the International Audio Laboratories Erlangen, Germany. His recent research interests include music processing, music information retrieval, and audio signal processing. He has coauthored more than 100 peer-reviewed scientific articles and has written a monograph, *Information Retrieval for Music and Motion* (Springer, 2007), and a textbook, *Fundamentals of Music Processing* (Springer, 2015).

Bryan A. Pardo (pardo@northwestern.edu) received his M.Mus. degree in jazz studies in 2001 and his Ph.D. degree in computer science in 2005, both from the University of Michigan, Ann Arbor. He is an associate professor of computer science and music at Northwestern University, Evanston, Illinois. He has authored more than 100 peer-reviewed publications in the areas of machine learning, signal processing, and music information retrieval. When he is not programming, writing, or teaching, he performs throughout North America on saxophone and clarinet.

Gautham J. Mysore (gmysore@adobe.com) received his M.A. degree in music, science, and technology in 2005, his M.S. degree in electrical engineering in 2008, and his Ph.D. degree in computer-based music theory and acoustics in 2010, all from Stanford University, California. He was previously a visiting researcher at the Gatsby Computational Neuroscience Unit at University College London, United Kingdom. He is a principal scientist and head of the Audio Research Group at Adobe Research in San Francisco, California, and an adjunct professor in the Center for Computer Research in Music and Acoustics at Stanford University. His research involves developing new machine-learning and signal processing algorithms for a wide variety of real-world audio applications.

Vesa Välimäki (vesa.valimaki@aalto.fi) received his M.Sc. degree in 1992 and his doctor of science in technology degree in 1995, both in electrical engineering from the Helsinki University of Technology, Espoo, Finland. In 2008–2009, he was a visiting scholar at Stanford University, California. He was the chair of the 2017 International Conference on Sound and Music Computing. He is a professor of audio signal processing and vice dean for research in electrical engineering at the Aalto University, Espoo, Finland. His research interests include headphone and loudspeaker signal processing, audio-effects processing, and sound synthesis. He is a fellow of the IEEE Audio Engineering Society.

SP

Emmanouil Benetos, Simon Dixon,
Zhiyao Duan, and Sebastian Ewert

Automatic Music Transcription

An overview

©ISTOCKPHOTO.COM/TRAFFIC_ANALYZER

Digital Object Identifier 10.1109/MSP.2018.2869928
Date of publication: 24 December 2018

The capability of transcribing music audio into music notation is a fascinating example of human intelligence. It involves perception (analyzing complex auditory scenes), cognition (recognizing musical objects), knowledge representation (forming musical structures), and inference (testing alternative hypotheses). Automatic music transcription (AMT), i.e., the design of computational algorithms to convert acoustic music signals into some form of music notation, is a challenging task in signal processing and artificial intelligence. It comprises several subtasks, including multipitch estimation (MPE), onset and offset detection, instrument recognition, beat and rhythm tracking, interpretation of expressive timing and dynamics, and score typesetting.

Given the number of subtasks it comprises and its wide application range, it is considered a fundamental problem in the fields of music signal processing and music information retrieval [1], [2]. Because of the very nature of music signals, which often contain several sound sources (e.g., musical instruments and voice) that produce one or more concurrent sound events (e.g., notes and percussive sounds) that are meant to be highly correlated over both time and frequency, AMT is still considered a challenging and open problem in the literature, particularly for music containing multiple instruments and many simultaneous notes (called *polyphonic music* in the music signal processing literature) [2].

The typical data representations used in an AMT system are illustrated in Figure 1. Usually, an AMT system takes an audio waveform as input [Figure 1(a)], computes a time–frequency representation [Figure 1(b)], and outputs a representation of pitches over time [also called a *piano-roll* representation, Figure 1(c)] or a typeset music score [Figure 1(d)].

In this article, we provide a high-level overview of AMT, emphasizing the intellectual merits and broader impacts of this topic and linking AMT to other problems found in the wider field of digital signal processing. We give an overview of approaches to AMT, detailing the methodology used in the two main families of methods, based respectively on deep learning and nonnegative matrix factorization (NMF). Finally, we provide an

 1053-5888/19©2019IEEE

extensive discussion of open challenges for AMT. Regarding the scope of the article, we emphasize approaches for transcribing polyphonic music produced by pitched instruments and voice. Outside the scope of the article are methods for transcribing nonpitched sounds, such as drums, for which a brief overview is given in the "Percussion and Unpitched Sounds" section, as well as methods for transcribing specific sources within a polyphonic mixture, such as melody and bass lines.

Applications and impact

A successful AMT system would enable a broad range of interactions between people and music, including music education (e.g., through systems for automatic instrument tutoring), music creation (e.g., dictating improvised musical ideas and automatic music accompaniment), music production (e.g., music content visualization and intelligent content-based editing), music search (e.g., indexing and recommendation of music by melody, bass, rhythm, or chord progression), and musicology (e.g., analyzing jazz improvisations and other nonnotated music). As such, AMT is an enabling technology with clear potential for both economic and societal impact.

AMT is closely related to other music signal processing tasks [3], such as audio source separation, which also involves the estimation and inference of source signals from mixture observations. It is also useful for many high-level tasks in music information retrieval [4], such as structural segmentation, cover-song detection, and assessment of music similarity, since these tasks are much easier to address once the musical notes are known. Thus, AMT provides the main link between the fields of music signal processing and symbolic music processing (i.e., the processing of music notation and music language modeling). The integration of the two aforementioned fields through AMT will be discussed in the section "Further Extensions and Future Work."

Given the potential impact of AMT, the problem has attracted commercial interest in addition to academic research. While it is outside the scope of this article to provide a comprehensive list of commercial AMT software, commonly used applications include Melodyne (http://www.celemony.com/en/melodyne), AudioScore (http://www.sibelius.com/products/audioscore/), ScoreCloud (http://scorecloud.com/), AnthemScore (https://www.lunaverus.com/), and Transcribe! (https://www.seventhstring.com/xscribe/). It is worth noting that AMT papers in the literature have refrained from making explicit comparisons with commercially available music transcription software, possibly because of the difference in scope and target application between commercial and academic tools.

AMT is an enabling technology with clear potential for both economic and societal impact.

Analogies to other fields

AMT has close relations with other signal processing problems. With respect to the field of speech processing, AMT is widely considered to be the musical equivalent of automatic speech recognition (ASR), in the sense that both tasks involve converting acoustic signals to symbolic sequences. Like the cocktail party problem in speech, music usually involves multiple simultaneous voices, but, unlike speech, these voices are highly correlated in time and in frequency (see challenges 2 and 3 in the "Key Challenges" section). In addition, both AMT and ASR systems benefit from language modeling components that are combined with acoustic components to produce plausible results. Thus, there are also clear links between AMT and the wider field of natural language processing (NLP), with music having its own grammatical rules or statistical regularities, in a way similar to natural language [5]. The use of language models for AMT is detailed in the section "Further Extensions and Future Work."

Within the emerging field of sound scene analysis, there is a direct analogy between AMT and sound event detection (SED) [6], in particular with polyphonic SED, which involves detecting and classifying multiple overlapping events from audio. While everyday and natural sounds do not exhibit the same degree of temporal regularity and intersource frequency dependence as found in music signals, there are close interactions

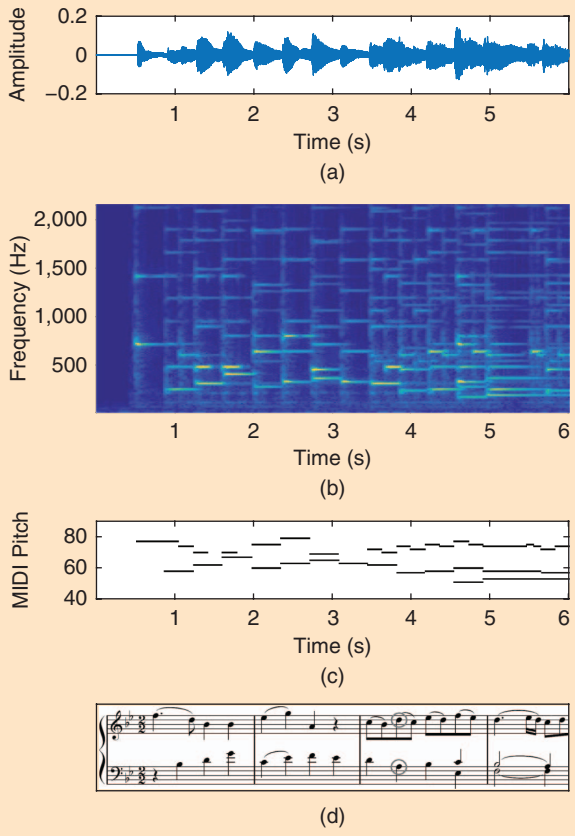


FIGURE 1. The data represented in an AMT system: the (a) input waveform, (b) internal time–frequency representation, (c) output piano-roll representation, and (d) output music score, with notes A and D marked in gray circles. The example corresponds to the first 6 s of W.A. Mozart's Piano Sonata no. 13, third movement. (Images courtesy of the MIDI Aligned Piano Sounds database.) MIDI: Musical Instrument Digital Interface.

between the two problems in terms of the methodologies used, as observed in the literature [6].

Furthermore, AMT is related to image processing and computer vision, as musical objects, such as notes, can be recognized as two-dimensional patterns in time–frequency representations. Compared with image processing and computer vision, where occlusion is a common issue, AMT systems are often affected by musical objects occupying the same time–frequency regions (this is detailed in the "Key Challenges" section).

Key challenges

Compared to other problems in the music signal processing field or the wider signal processing discipline, there are several factors that make AMT particularly challenging:

1) Polyphonic music contains a mixture of multiple simultaneous sources (e.g., instruments and vocals) with different pitch, loudness, and timbre (sound quality), with each source producing one or more musical voices. Inferring musical attributes (e.g., pitch) from the mixture signal is an extremely underdetermined problem.
2) Overlapping sound events often exhibit harmonic relations with each other. For any consonant musical interval, the fundamental frequencies form small integer ratios, so that their harmonics overlap in frequency, making the separation of the voices even more difficult. Taking a C major chord as an example, the fundamental frequency ratio of its three notes C:E:G is 4:5:6, and the percentages of harmonic positions that are overlapped by the other notes are 46.7%, 33.3%, and 60% for C, E, and G, respectively.
3) The timing of musical voices is governed by the regular metrical structure of the music. In particular, musicians pay close attention to the synchronization of onsets and offsets between different voices, which violates the common assumption of statistical independence between sources, which otherwise facilitates separation.
4) The annotation of ground-truth transcriptions for polyphonic music is very time consuming and requires high expertise. The lack of such annotations has limited the use of powerful supervised-learning techniques to specific AMT subproblems, such as piano transcription, where the annotation can be automated because of certain piano models that can automatically capture performance data. An approach to circumvent this problem was proposed in [7], but it requires professional music performers and thorough score pre- and postprocessing. We note that sheet music does not generally provide good ground-truth annotations for AMT; it is not time-aligned to the audio signal, nor does it usually provide an accurate representation of a performance. Even when accurate transcriptions exist, it is not trivial to identify corresponding pairs of audio files and musical scores because of the multitude of versions of any given musical work that are available from music distributors. At best, musical scores can be viewed as weak labels.

AMT provides the main link between the fields of music signal processing and symbolic music processing.

These key challenges are often not fully addressed in current AMT systems, leading to common issues in the AMT outputs, such as octave errors, semitone errors, missed notes (in particular, in the presence of dense chords), extra notes (often manifested as harmonic errors in the presence of unseen timbres), merged or fragmented notes, incorrect onsets/offsets, or misassigned streams [1], [2]. The remainder of this article will focus on ways to address the previously mentioned challenges as well as on discussion of additional open problems for the creation of robust AMT systems.

An overview of AMT methods

In the past four decades, many approaches have been developed for AMT for polyphonic music. While the end goal of AMT is to convert an acoustic music recording to some form of music notation, most approaches were designed to achieve a certain intermediate goal. Depending on the level of abstraction and the structures that need to be modeled for achieving such goals, AMT approaches can be generally organized into four categories: frame level, note level, stream level, and notation level.

Frame-level transcription, or MPE, is the estimation of the number and pitch of notes that are simultaneously present in each time frame (on the order of 10 ms). This is usually performed independently in each frame, although contextual information is sometimes considered through filtering frame-level pitch estimates in a postprocessing stage. Figure 2(a) shows an example of a frame-level transcription, where each black dot is a pitch estimate. Methods in this category do not form the concept of musical notes and rarely model any high-level musical structures.

A large portion of existing AMT techniques operate at this level. Recent approaches include traditional signal processing methods [11], [12], probabilistic modeling [8], Bayesian approaches [13], NMF [14]–[17], and neural networks (NNs) [18], [19]. All of these methods have pros and cons, and the research has not converged on a single approach. For example, traditional signal processing methods are simple and fast and generalize better to different instruments, while deep NN methods generally achieve higher accuracy on specific instruments (e.g., piano). Bayesian approaches provide a comprehensive modeling of the sound generation process, but the models can be very complex and slow. Readers interested in a comparison of the performance of different approaches are referred to the Multiple Fundamental Frequency Estimation and Tracking task of the annual Music Information Retrieval Evaluation eXchange (MIREX) (http://www.music-ir.org/mirex). However, readers are reminded that evaluation results may be biased by the limitations of data sets and evaluation metrics (see the sections "Key Challenges" and "Evaluation Metrics").

Note-level transcription, or note tracking, is one level higher than MPE in terms of the richness of structures of the estimates. It not only estimates the pitches in each time frame but also connects pitch estimates over time into notes. In the AMT literature,

a musical note is usually characterized by three elements: pitch, onset time, and offset time [1]. As note offsets can be ambiguous, they are sometimes neglected in the evaluation of note-tracking approaches, and, as such, some note-tracking approaches only estimate pitch and onset times of notes. Figure 2(b) shows an example of a note-level transcription, where each note is shown as a red circle (onset) followed by a black line (pitch contour). Many note-tracking approaches form notes by postprocessing MPE outputs (i.e., pitch estimates in individual frames). Techniques that have been used in this context include median filtering [12], hidden Markov models (HMMs) [20], and NNs [5]. This postprocessing is frequently performed for each Musical Instrument Digital Interface (MIDI) pitch independently without considering the interactions among simultaneous notes. This often leads to spurious or missing notes that share harmonics with correctly estimated notes.

From frame level to note level to stream level, the transcription task becomes more complex as more musical structures and cues need to be modeled.

Some approaches have been proposed to consider note interactions through a spectral likelihood model [9] or a music language model [5], [18] (see the "MLMs" section). Another subset of approaches estimates notes directly from the audio signal instead of building upon MPE outputs. Some approaches first detect onsets and then estimate pitches within each interonset interval [21], while others estimate pitch, onset, and sometimes offset in the same framework [22]–[24].

Stream-level transcription, also called *multipitch streaming* (*MPS*), targets the grouping of estimated pitches or notes into streams, where each stream typically corresponds to one instrument or musical voice and is closely related to instrument source separation. Figure 2(c) shows an example of a stream-level transcription, where pitch streams of different instruments have different colors. Compared to note-level transcription, the pitch contour of each stream is much longer than a single note and contains multiple discontinuities that are caused by silence, nonpitched sounds, and abrupt frequency changes. Therefore, techniques that are ordinarily used in note-level transcription are generally not sufficient for grouping pitches with a long and discontinuous contour. One important cue for MPS that is not explored in MPE and note tracking is timbre. Notes of the same stream (source) generally show similar timbral characteristics compared to those in different streams. Therefore, stream-level transcription is also called *timbre tracking* or *instrument tracking* in the literature. Existing works at this level are few, with [10], [16], and [25] as examples.

From frame level to note level to stream level, the transcription task becomes more complex as more musical structures and cues need to be modeled. However, the transcription outputs at these three levels are all parametric transcriptions, which are parametric descriptions of the audio content. The MIDI piano roll shown in Figure 1(c) is a good example of such a transcription. It is, indeed, an abstraction of music audio, but it has not yet reached the level of abstraction of music notation. Time is still measured in the unit of seconds instead of beats; pitch is measured in MIDI numbers instead of spelled note names that are compatible with the key (e.g., C♯ versus D♭); and the concepts of beat, bar, meter, key, harmony, and stream are lacking.

Notation-level transcription aims to transcribe the music audio into a human-readable musical score, such as the staff notation widely used in Western classical music. Transcription at this level requires a deeper understanding of musical structures, including harmonic, rhythmic, and stream structures. Harmonic structures, such as keys and chords, influence the note spelling of each MIDI pitch; rhythmic structures, such as beats and bars, help to quantize the length of notes; and stream structures aid the assignment of notes to different staffs. There has been some work on the estimation of musical structures from audio or MIDI

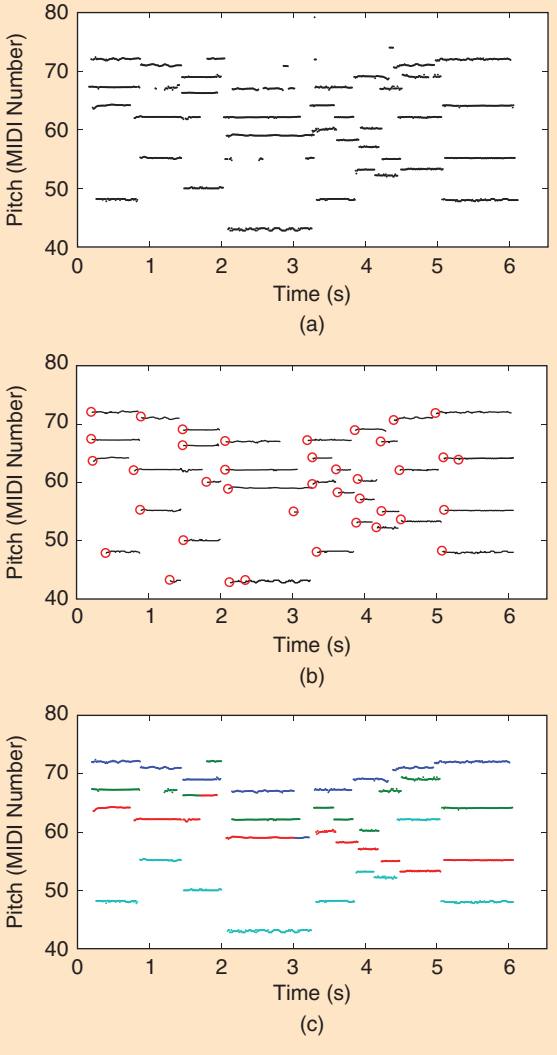


FIGURE 2. Examples of (a) frame-level, (b) note-level, and (c) stream-level transcriptions, produced by running the methods proposed in [8], [9], and [10], respectively, of the first phrase of J.S. Bach's chorale *Ach Gott und Herr* from the Bach10 data set. All three levels are parametric descriptions of the music performance.

representations of a performance. For example, methods for pitch spelling [26], timing quantization [27], and voice separation [28] from performed MIDI files have been proposed. However, little work has been done on integrating these structures into a complete music notation transcription, especially for polyphonic music.

Several software packages, including Finale, GarageBand, and MuseScore, provide the functionality of converting a MIDI file into music notation, but the results are typically not satisfying, and it is not clear what musical structures have been estimated and integrated during the transcription process. Cogliati et al. [29] proposed a method to convert a MIDI performance into music notation, based on a systematic comparison of the transcription performance with the aforementioned software. In terms of audio-to-notation transcription, a proof-of-concept work using end-to-end NNs was proposed by Carvalho and Smaragdis [30] to directly map music audio into music notation without explicitly modeling musical structures.

State of the art

While there is a wide range of applicable methods, AMT has been dominated during the last decade by two algorithmic families: NMF and NNs. Both families have been used for a variety of tasks, from speech and image processing to recommender systems and NLP. Despite this wide applicability, both approaches offer a range of properties that make them particularly suitable for modeling music recordings at the note level.

While there is a wide range of applicable methods, AMT has been dominated during the last decade by two algorithmic families: NMF and NNs.

NMF for AMT

The basic idea behind NMF and its variants is to represent a given nonnegative time–frequency representation $\mathbf{V} \in \mathbb{R}_{\geq 0}^{M \times N}$, e.g., a magnitude spectrogram, as a product of two nonnegative matrices: a dictionary $\mathbf{D} \in \mathbb{R}_{\geq 0}^{M \times K}$ and an activation matrix $\mathbf{A} \in \mathbb{R}_{\geq 0}^{K \times N}$ (see Figure 3). Computationally, the goal is to minimize a distance (or divergence) between $\mathbf{V}$ and $\mathbf{DA}$ with respect to $\mathbf{D}$ and $\mathbf{A}$. As a straightforward approach to solving this minimization problem, multiplicative update rules have been central to the success of NMF. For example, the generalized Kullback–Leibler divergence between $\mathbf{V}$ and $\mathbf{DA}$ is nonincreasing under the following updates and guarantees the nonnegativity of both $\mathbf{D}$ and $\mathbf{A}$ as long as both are initialized with positive real values [31]:

$$\mathbf{A} \leftarrow \mathbf{A} \odot \frac{\mathbf{D}^\top \left(\frac{\mathbf{V}}{\mathbf{DA}}\right)}{\mathbf{D}^\top \mathbf{J}} \quad \text{and} \quad \mathbf{D} \leftarrow \mathbf{D} \odot \frac{\left(\frac{\mathbf{V}}{\mathbf{DA}}\right)\mathbf{A}^\top}{\mathbf{J}\mathbf{A}^\top},$$

where the $\odot$ operator denotes pointwise multiplication, $\mathbf{J} \in \mathbb{R}^{M \times N}$ denotes the matrix of ones, and the division is pointwise. Intuitively, the update rules can be derived by choosing a

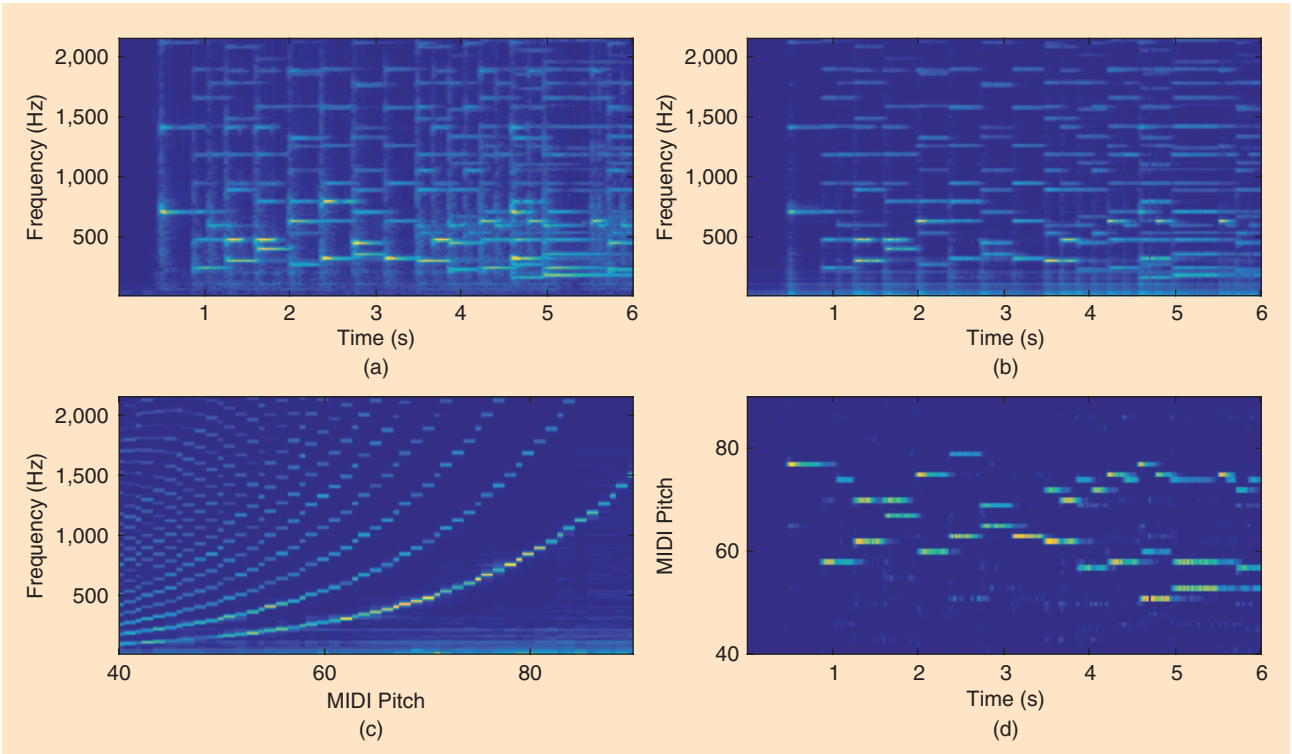


FIGURE 3. An example of NMF, using the same audio recording as in Figure 1: the (a) input spectrogram **V**, (b) approximated spectrogram **DA**, (c) dictionary **D** (preextracted), and (d) activation matrix **A**.

specific step size in a gradient (or, rather, coordinate) descent-based minimization of the divergence [31].

In an AMT context, both unknown matrices have an intuitive interpretation. The nth column of $\mathbf{V}$, i.e., the spectrum at time point n, is modeled in NMF as a linear combination of the K columns of $\mathbf{D}$, and the corresponding K coefficients are given by the nth column of $\mathbf{A}$. Given this point of view, each column of $\mathbf{D}$ is generally referred to as a *spectral template* and usually represents the expected spectral energy distribution associated with a specific note played on a specific instrument. For each template, the corresponding row in $\mathbf{A}$ is referred to as the associated *activation* and encodes when and how intensely that note is played over time. Given the nonnegativity constraints, NMF yields a purely constructive representation in the sense that the spectral energy modeled by one template cannot be canceled by another. This property is often seen as instrumental in identifying a parts-based and interpretable representation of the input [31].

In Figure 3, an NMF-based decomposition is illustrated. The magnitude spectrogram $\mathbf{V}$ shown in Figure 3(a) is modeled as a product of the dictionary $\mathbf{D}$ and activation matrix $\mathbf{A}$, shown in Figure 3(c) and (d), respectively. The product $\mathbf{DA}$ is given in Figure 3(b). In this case, the templates correspond to individual pitches, with clearly visible fundamental frequencies and harmonics. Additionally, comparing $\mathbf{A}$ with the piano-roll representation shown in Figure 1(c) indicates the correlation between NMF activations and the underlying musical score.

While Figure 3 illustrates the principles behind NMF, it also indicates why AMT is difficult. Indeed, a regular NMF decomposition would rarely look as clean as in Figure 3. Compared to speech analysis, sound objects in music are highly correlated. For example, even in a simple piece as in Figure 1, most pairs of simultaneous notes are separated by musically consonant intervals, which acoustically means that many of their partials overlap [e.g., the A and D notes around 4 s, marked with gray circles in Figure 1(d), share a high number of partials]. In this case, it can be difficult to disentangle how much energy belongs to which note. The task is further complicated by the fact that the spectrotemporal properties of notes vary considerably between different pitches, playing styles, dynamics, and recording conditions. Furthermore, the stiffness property of the strings affects the travel speed of transverse waves based on their frequency. As a result, the partials of instruments like the piano are not found at perfect integer multiples of the fundamental frequency. Because of this property, called *inharmonicity*, the positions of partials differ between individual pianos (see Figure 4).

To address these challenges, the basic NMF model has been extended by encouraging additional structure in the dictionary and the activations. For example, an important principle is to enforce sparsity in $\mathbf{A}$ to obtain a solution dominated by activations that are few but substantial; the success of sparsity paved the way for a whole range of sparse-coding approaches, in which the dictionary size K can considerably exceed the input dimension M [32]. Other extensions focus on the dictionary design. In the case of supervised NMF, the dictionary is precomputed and fixed using additionally available training material. For example, given K recordings, each containing only a single note, the dictionary shown in Figure 3(b) was constructed by extracting one template from each recording. This way, the templates are guaranteed to be free of interference from other notes and also to have a clear interpretation. As another example, Figure 5 illustrates an extension in which each NMF template is represented as a linear combination of fixed narrow-band subtemplates [15], which enforces a harmonic structure for all NMF templates. This way, a dictionary can be adapted to the recording to be transcribed, while maintaining its clean, interpretable structure.

In shift-invariant dictionaries, a single template can be used to represent a range of different fundamental frequencies. In particular, using a logarithmic frequency axis, the distances between individual partials of a harmonic sound are fixed, and thus shifting a template in frequency allows the modeling of sounds of varying pitch. Sharing parameters between different pitches in this way has turned out to be effective for increasing model capacity (see, e.g., [16] and [17]). Furthermore, spectrotemporal dictionaries alleviate a specific weakness of NMF models. In NMF, it is difficult to express that notes often have a specific temporal evolution. For example, the beginning of a note (or attack phase) might have entirely different spectral properties than the central part (decay phase). Such relationships are modeled in spectrotemporal dictionaries using a Markov process, which governs the sequencing of templates across frames so that different subsets of templates can be used for the attack and decay parts, respectively [16], [23].

NNs for AMT

As for many tasks relating to pattern recognition, NNs have, in recent years, had a considerable impact on the problem of music transcription and on music signal processing in general. NNs are able to learn a nonlinear function (or a composition of functions) from input to output via an optimization algorithm, such as stochastic gradient descent [33]. Compared to other fields, including image processing, progress on NNs for music transcription has been slower, and we will discuss a few of the underlying reasons.

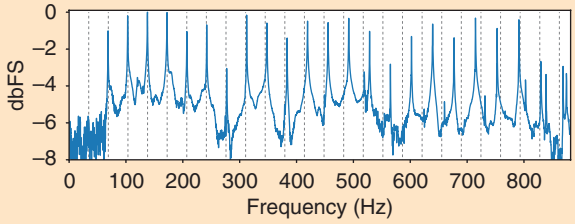


FIGURE 4. An illustration of inharmonicity: the spectrum of a C♯1 note played on a piano. The stiffness of the strings causes partials to be shifted from perfect integer multiples of the fundamental frequency (shown as vertical dotted lines). Here, the 23rd partial is at the position where the 24th harmonic would be expected. Note that the fundamental frequency of 34.65 Hz is missing, as piano soundboards typically do not resonate for modes with a frequency smaller than ≈ 50 Hz.

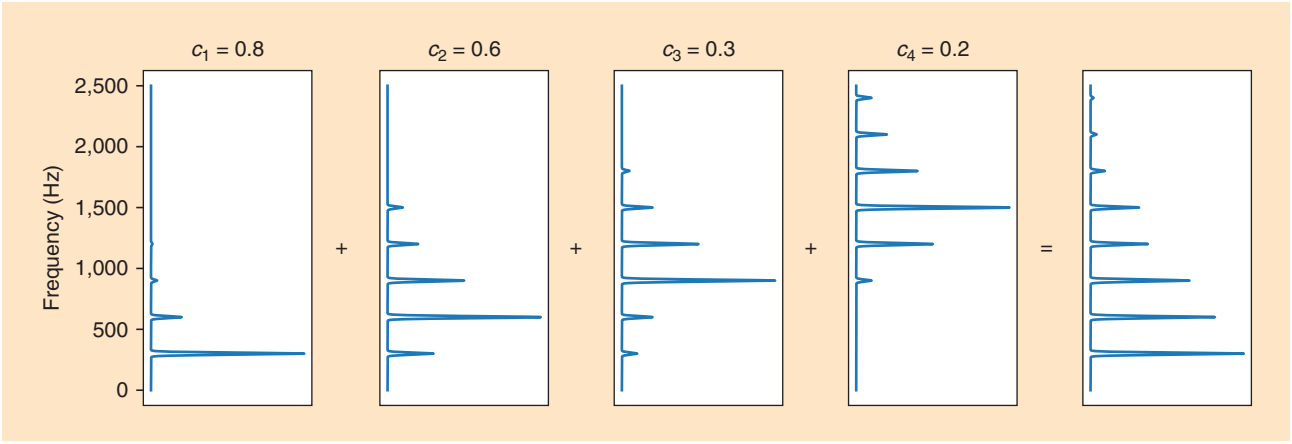


FIGURE 5. An illustration of harmonic NMF [15]. Each NMF template (far right) is represented as a linear combination of fixed narrow-band subtemplates. The resulting template is constrained to represent harmonic sounds by construction.

One of the earliest approaches based on NNs was Marolt's Sonic system [21]. A central component in this approach was the use of time-delay networks, which resemble convolutional networks in the time direction [33] and were employed to analyze the output of adaptive oscillators to track and group partials in the output of a gammatone filterbank. Although it was initially published in 2001, the approach remains competitive and still appears in comparisons in more recent publications [23].

In the context of the more recent revival of NNs, a first successful system was presented by Böck and Schedl [34]. One of the core ideas was to use two spectrograms as input to enable the network to exploit both a high time accuracy (when estimating the note onset position) and a high frequency resolution (when disentangling notes in the lower frequency range). This input is processed using one (or more) long short-term memory (LSTM) layers [33]. The potential benefit of using LSTM layers is twofold. First, the spectral properties of a note evolve across input frames, and LSTM networks have the capability to compactly model such sequences. Second, medium- and long-range dependencies between notes can potentially be captured. For example, based on a popular chord sequence, after hearing C and G major chords followed by A minor, a likely successor is an F major chord. An investigation of whether such long-range dependencies are indeed modeled, however, was not within the scope of this work.

Compared to other fields, including image processing, progress on NNs for music transcription has been slower.

Sigtia et al. [18] focused on long-range dependencies in music by combining an acoustic front end with a symbolic-level module resembling a language model, as used in speech processing. Using information obtained from MIDI files, a recurrent NN (RNN) was trained to predict the active notes in the next time frame, given those in the past. This approach needed to learn and represent a very large joint probability distribution, i.e., a probability for every possible combination of active and inactive notes across time. Note that, even in a single frame, there are 2^{88} possible combinations of notes on a piano. To render the problem of modeling such an enormous probability space tractable, the approach employed a specific NN architecture (the Neural Autoregressive Distribution Estimator, also known as *NADE*), which represented a large joint probability as a long product of conditional probabilities, an approach quite similar to the idea popularized recently by the well-known WaveNet architecture. Despite the use of a dedicated music language model, which was trained on relatively large MIDI-based data sets, only modest improvements over an HMM baseline could be observed, and thus the question remains open regarding to which degree long-range dependencies were indeed captured.

To further disentangle the influence of the acoustic front end from the language model on potential improvement in performance, Kelz et al. [19] focused on the acoustic modeling, reporting on the results of a larger-scale hyperparameter search and describing the influence of individual system components. Trained using this careful and extensive procedure, the resulting model outperformed existing models by a reasonable margin. In other words, while in speech processing, language models have led to a drastic improvement in performance, the same effect is still to be demonstrated in an AMT system, a challenge we will discuss in more detail hereafter.

The development of NN-based AMT approaches continues. The current state-of-the-art method for general-purpose piano transcription was proposed by Google Brain [24]. Combining and extending ideas from existing methods, this approach combines two networks (Figure 6). One detects note onsets, and its output is used to inform a second network, which focuses on perceiving note lengths. This can be interpreted from a probabilistic point of view. Note onsets are rare events compared to framewise note activity detections. The split into two network branches can thus be interpreted as splitting the representation of a relatively complex joint probability distribution over onsets and frame activity into a probability over onsets and a probability over frame activities, conditioned on the onset

distribution. Since the temporal dynamics of onsets and frame activities are quite different, this can lead to improved learning behavior for the entire network when trained jointly.

A comparison of NMF and NN models

Given the popularity of NMF and NN-based methods for AMT, it is interesting to discuss their differences. In particular, neglecting the nonnegativity constraints, NMF is a linear, generative model. Given that NMF-based methods are increasingly being replaced by NN-based ones, the question arises in which way linearity could be a limitation for an AMT model.

To look into this, assume we are given an NMF dictionary with two spectral templates for each musical pitch. To represent an observed spectrum of a single pitch C4, we can linearly combine the two templates associated with C4. The set (or manifold) of valid spectra for C4 notes, however, is complex, and thus, in most cases, our linear interpolation will not correspond to a real-world recording of a C4. We could increase the number of templates such that their interpolation could potentially get closer to a real C4—but the number of invalid spectra we can represent increases much more quickly compared to the number of valid spectra. Deep networks have shown considerable potential in recent years to implicitly represent such complex manifolds in a robust and comparatively efficient way [33]. An additional benefit over generative models like NMF is that NNs can be trained in an end-to-end fashion, i.e., note detections can be a direct output of a network without the need for additional postprocessing of model parameters (such as NMF activations).

Yet, despite these quite principled limitations, NMF-based methods remain competitive or even exceed the results achieved using NNs. Currently, there are two main challenges for NN-based approaches. First, there are only a few, relatively small annotated data sets available, and these are often subject to severe biases [7]. The largest publicly available data set [11] contains several hours of piano music—but all recorded on only seven different synthesizer-based and real pianos. While typical data augmentation strategies, such as pitch shifting or simulating different room acoustics, might mitigate some of the effects, there is still a considerable risk that a network overfits the acoustic properties of these specific instruments. For many types of instruments, even small data sets are not available. Other biases include musical style as well as the distribution over central musical concepts, such as key, harmony, tempo, and rhythm.

A second considerable challenge is the adaptability to new acoustic conditions. Providing just a few examples of isolated notes of the instrument to be transcribed, considerable improvements are observed in the performance of NMF-based models. There is currently no corresponding, equally effective mechanism to retrain or adapt an NN-based AMT system on a few seconds of audio. Thus, the error rate for nonadapted networks can be an order of magnitude higher than that of an adapted NMF system [23], [24]. Overall, as both of these challenges cannot easily be overcome, NMF-based methods are likely to remain relevant in specific use cases.

In Figure 7, we qualitatively illustrate some differences in the behavior of systems based on supervised NMF and NNs. Both systems were specifically trained for transcribing piano recordings, and we expose the approaches to a recording of an organ. Like the piano, the organ is played with a keyboard, but its acoustic properties are quite different. The harmonics of

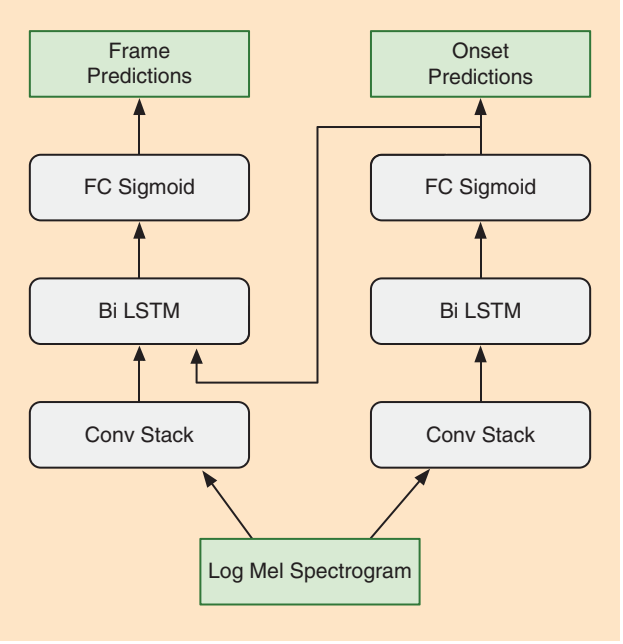


FIGURE 6. Google Brain's Onset and Frames Network. The input is processed by an initial network detecting note onsets. The result is used as side information for a second network focused on estimating note lengths. Bi LSTM: bidirectional LSTM layers; FC Sigmoid: fully connected sigmoid layer; Conv Stack: a series of convolutional layers. (Image adapted with permission from [24].)

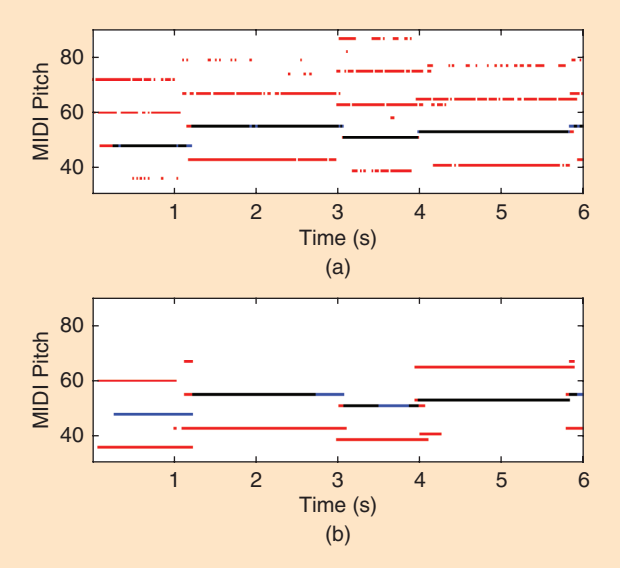


FIGURE 7. Piano-roll representations of the first 6 s of a recording of a Bach piece (BWV 582) for the organ. The black corresponds to correctly detected pitches, red to false positives, and blue to false negatives. (a) The output of an NMF model trained on piano templates. (b) The output of the piano-music-trained NN model of [24].

the organ are rich in energy and cover the entire spectrum; the energy of the notes does not decay over time, and the onsets are less pronounced. With this experiment, we want to find out how gracefully the systems fail when they encounter a sound that is outside the piano-sound manifold but still musically valid.

Comparing the NMF output in Figure 7(a) and the NN output in Figure 7(b) with the ground truth, we find that both methods detect additional notes (shown in red), mostly at octaves above and below the correct fundamental. Given the rich energy distribution, this behavior is expected. While we use a simple baseline model for NMF—and thus some errors could be attributed to that choice—the NN fails more gracefully. That is, fewer octave errors and fewer spurious short note detections are observed. (Yet, in terms of recall, the NMF-based approach identifies additional correct notes.)

It is difficult to argue why the acoustic model within the network should be better prepared for such a situation. However, the results suggest that the network learned something additional: the LSTM layers as used in the network (compare Figure 6) seem to have learned how typical piano notes evolve in time, and thus most note lengths look reasonable and less spurious. Similarly, the bandwidth in which octave errors occur is narrower for the NN, which could potentially indicate that the network models the likelihood of co-occurring notes or, in other words, a simple music language model (MLM). This leads us to our discussion of important remaining challenges in AMT.

As might be evident by surveying the AMT literature, the vast majority of approaches target only Western (or Eurogenetic) music.

Further extensions and future work

MLMs

As outlined in the "Analogies to Other Fields" section, AMT is closely related to ASR. In the same way that a typical ASR system consists of an acoustic component and a language component, an AMT system can model both the acoustic sequences and the underlying sequence of notes and other music cues over time. AMT systems have thus incorporated MLMs for modeling sequences of notes in a polyphonic context, with the aim of improving transcription performance. The capabilities of deep-learning methods toward modeling high-dimensional sequences have recently made polyphonic music sequence prediction possible. Boulanger-Lewandowski et al. [5] combined a restricted Bolzmann machine (RBM) with an RNN for polyphonic music prediction, which was used to postprocess the acoustic output of an AMT system.

Sigtia et al. [18] also used the aforementioned RNN–RBM as an MLM and combined the acoustic and language predictions using a probabilistic graphical model. While these initial works showed promising results, there are several directions for future research in MLMs. These include creating unified acoustic and language models (as opposed to using MLMs as postprocessing steps) and modeling other musical cues, such as chords, key, and meter (as opposed to simply modeling note sequences).

Score-informed transcription

If a known piece is performed, the musical score provides a strong prior for the transcription. In many cases, there are discrepancies between the score and a given music performance, which may be due to a specific interpretation by a performer or to performance mistakes. For applications like music education, it is useful to identify such discrepancies, by incorporating the musical score as additional prior information to simplify the transcription process (score-informed music transcription [35]). Typically, systems for such transcription use a score-to-audio alignment method as a preprocessing step to align the music score with the input music audio prior to performing transcription, as in [35]. While specific instances of such systems have been developed for certain instruments (piano and violin), the problem is still relatively unexplored, as is the related and more challenging problem of lead-sheet-informed transcription and the eventual integration of these methods toward the development of automatic music tutoring systems.

Context-specific transcription

While the creation of a blind multi-instrument AMT system without specific knowledge of the music style, instruments, and recording conditions is yet to be achieved, considerable progress has been reported on the problem of context-specific transcription, where prior knowledge of the sound of the specific instrument model or manufacturer and the recording environment is available. For context-specific piano transcription, multipitch detection accuracy can exceed 90% [22], [23], making such systems appropriate for user-facing applications. Open work in this topic includes the creation of context-specific AMT systems for multiple instruments.

Non-Western music

As might be evident by surveying the AMT literature, the vast majority of approaches target only Western (or Eurogenetic) music. This allows several assumptions, regarding both the instruments used and also the way that music is represented and produced. Typical assumptions include octaves containing 12 equally spaced pitches; two modes, major and minor; and a standard tuning frequency of A4 = 440 Hz.

However, these assumptions do not hold true for other music styles from around the world, where an octave is often divided into microtones (e.g., Arabic music theory is based on quartertones) or where there are modes not used in Western music (e.g., classical Indian music recognizes hundreds of modes, called *ragas*). Therefore, automatically transcribing non-Western music still remains an open problem with several challenges, including the design of appropriate signal and music notation representations while avoiding a so-called Western bias [36]. Another major issue is the lack of annotated data sets for non-Western music,

rendering the application of data-intensive machine-learning methods difficult.

Expressive pitch and timing

Western notation conceptualizes music as sequences of unchanging pitches being maintained for regular durations and has little scope for representing expressive use of microtonality and microtiming or for the detailed recording of timbre and dynamics. Research on automatic transcription has followed this narrow view, describing notes in terms of discrete pitches plus onset and offset times. For example, no suitable notation exists for performed singing, the most universal form of music making. Likewise, for other instruments without fixed pitch or with other expressive techniques, better representations are required. These richer representations can then be reduced to Western score notation, if required, by modeling musical knowledge and stylistic conventions.

Automatically transcribing non-Western music still remains an open problem with several challenges.

Percussion and unpitched sounds

An active problem in the music signal processing literature is that of detecting and classifying nonpitched sounds in music signals [1, Ch. 5]. In most cases, this is expressed as the problem of drum transcription, since the vast majority of contemporary music contains mixtures of pitched sounds and unpitched sounds produced by a drum set. Drum set components typically include the bass drum, snare drum, hi-hat, cymbals, and toms. The problem in this case is to detect and classify percussive sounds into one of the aforementioned sound classes. Elements of the drum transcription problem that make it particularly challenging are the concurrent presence of several harmonic, inharmonic, and nonharmonic sounds in the music signal, as well as the requirement of an increased temporal resolution for drum transcription systems compared to typical multipitch detection systems. Approaches for pitched instrument transcription and drum transcription have largely been developed independently, and the creation of a robust music transcription system that supports both pitched and unpitched sounds still remains an open problem.

Evaluation metrics

Most AMT approaches are evaluated using the set of metrics proposed for the MIREX Multiple-F0 Estimation and Note Tracking public evaluation tasks (http://www.music-ir.org/mirex/). Three types of metrics are included: frame based, note based, and stream based, mirroring the frame-level, note-level, and stream-level transcription categories presented in the "State of the Art" section. While the above sets of metrics all have their merits, it could be argued that they do not correspond with human perception of music transcription accuracy, where, e.g., an extra note might be considered as a more severe error than a missed note, or where out-of-key note errors might be penalized more compared with in-key ones. Therefore, the creation of perceptually relevant evaluation metrics for AMT and the creation of evaluation metrics for notation-level transcription remain open problems.

Conclusions

AMT has remained an active area of research in the fields of music signal processing and music information retrieval for several decades, with several potential benefits in other areas and fields extending beyond music. As outlined in this article, there remain several hurdles to be overcome, i.e., on modeling music signals and on the availability of data, as described in the "Key Challenges" section; with respect to the limitations of state-of-the-art methodologies, as described in the section "A Comparison of NMF and NN-Models"; and, finally, on extensions beyond the current area of existing tasks, as presented in the "Further Extensions and Future Work" section. We believe that addressing these challenges will lead toward the creation of a complete music transcription system and unlock the full potential of music signal processing technologies. Supplementary audio material related to this article can be found on the companion website (http://c4dm.eecs.qmul.ac.uk/spm-amt-overview/).

Acknowledgment

Emmanouil Benetos is supported by U.K. RAEng Research Fellowship RF/128. The authors are listed alphabetically.

Authors

Emmanouil Benetos (emmanouil.benetos@qmul.ac.uk) received his B.Sc. and M.Sc. degrees in informatics from Aristotle University of Thessaloniki, Greece, in 2005 and 2007, respectively, and his Ph.D. degree in electronic engineering from Queen Mary University of London, in 2012. He is a lecturer and Royal Academy of Engineering research fellow with the Centre for Digital Music, Queen Mary University of London, and a Turing Fellow with the Alan Turing Institute. From 2013 to 2015, he was a university research fellow with the Department of Computer Science, City University of London. He has published more than 80 peer-reviewed papers spanning several topics in audio and music signal processing. His research focuses on signal processing and machine learning for music and audio analysis as well as applications to music information retrieval, acoustic scene analysis, and computational musicology. He is a Member of the IEEE.

Simon Dixon (s.e.dixon@qmul.ac.uk) received his B.Sc. (with honors) and Ph.D. degrees in computer science from the University of Sydney, Australia, in 1989 and 1994, respectively. He studied classical guitar at the New South Wales Conservatorium of Music, Sydney, where he obtained his A.Mus.A and L.Mus.A degrees in 1987 and and 1988, respectively. He is the deputy director of the Centre for Digital Music, Queen Mary University of London, where he is also a professor. His research is in music informatics, including high-level music signal analysis, computational modeling of musical knowledge, and the study of musical performance. Particular areas of focus include automatic music transcription, beat tracking, audio alignment, and analysis of intonation and temperament. He was president (2014–2015) of the International Society for

Music Information Retrieval (ISMIR), is founding editor of the *Transactions of ISMIR*, and has published more than 200 refereed papers in the area of music informatics.

Zhiyao Duan (zhiyao.duan@rochester.edu) received his B.S. degree in automation and his M.S. degree in control science and engineering from Tsinghua University, China, in 2004 and 2008, respectively, and his Ph.D. degree in computer science from Northwestern University, Evanston, Illinois, in 2013. He is an assistant professor in the Electrical and Computer Engineering Department, University of Rochester, New York. His research interest is in the broad area of computer audition, i.e., designing computational systems that are capable of understanding sounds, including music, speech, and environmental sounds. He copresented a tutorial on automatic music transcription at the International Society for Music Information Retrieval (ISMIR) Conference in 2015. He received a best paper award at the 2017 Sound and Music Computing Conference and a best paper nomination at ISMIR 2017. He is a Member of the IEEE.

Sebastian Ewert (sewert@spotify.com) received his M.Sc./Diplom and Ph.D. degrees (summa cum laude) in computer science from the University of Bonn, Germany, in 2007 and 2012, respectively. In 2012, he was awarded a German Exchange Academic Service fellowship and joined the Centre for Digital Music, Queen Mary University of London. While there, he became a lecturer in signal processing in 2015 and was one of the founding members of the Machine Listening Lab, which focuses on the development of machine learning and signal processing methods for audio and music applications. He is now a senior research scientist at Spotify and is a Member of the IEEE.

References

[1] A. Klapuri and M. Davy, Eds., *Signal Processing Methods for Music Transcription*. New York: Springer-Verlag, 2006.

[2] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri, "Automatic music transcription: Challenges and future directions," *J. Intelligent Inform. Syst.*, vol. 41, no. 3, pp. 407–434, 2013.

[3] M. Müller, D. P. Ellis, A. Klapuri, and G. Richard, "Signal processing for music analysis," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 6, pp. 1088–1110, 2011.

[4] M. Schedl, E. Gómez, and J. Urbano, "Music information retrieval: Recent developments and applications," *Foundations Trends Inform. Retrieval*, vol. 8, pp. 127–261, 2014. doi: 10.1561/1500000042.

[5] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, "Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription," in *Proc. Int. Conf. Machine Learning*, 2012, pp. 1159–1166.

[6] T. Virtanen, M. D. Plumbley, and D. P. W. Ellis, Eds., *Computational Analysis of Sound Scenes and Events*. New York: Springer-Verlag, 2018.

[7] L. Su and Y.-H. Yang, "Escaping from the abyss of manual annotation: New methodology of building polyphonic datasets for automatic music transcription," in *Proc. Int. Symp. Computer Music Multidisciplinary Research*, 2015, pp. 309–321.

[8] Z. Duan, B. Pardo, and C. Zhang, "Multiple fundamental frequency estimation by modeling spectral peaks and non-peak regions," *IEEE Trans. Audio, Speech, Language Process. (2006–2013)*, vol. 18, no. 8, pp. 2121–2133, 2010.

[9] Z. Duan and D. Temperley, "Note-level music transcription by maximum likelihood sampling," in *Proc. 15th Int. Society Music Information Retrieval Conf.*, 2014, pp. 181–186.

[10] Z. Duan, J. Han, and B. Pardo, "Multi-pitch streaming of harmonic sound mixtures," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 22, no. 1, pp. 138–150, 2014.

[11] V. Emiya, R. Badeau, and B. David, "Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle," *IEEE Trans. Audio, Speech, Language Process. (2006–2013)*, vol. 18, no. 6, pp. 1643–1654, 2010.

[12] L. Su and Y.-H. Yang, "Combining spectral and temporal representations for multipitch estimation of polyphonic music," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 23, no. 10, pp. 1600–1612, 2015.

[13] P. H. Peeling, A. T. Cemgil, and S. J. Godsill, "Generative spectrogram factorization models for polyphonic piano transcription," *IEEE Trans. Audio, Speech, Language Process. (2006–2013)*, vol. 18, no. 3, pp. 519–527, 2010.

[14] P. Smaragdis and J. C. Brown, "Non-negative matrix factorization for polyphonic music transcription," in *Proc. IEEE Workshop Applications Signal Processing Audio and Acoustics*, 2003, pp. 177–180.

[15] E. Vincent, N. Bertin, and R. Badeau, "Adaptive harmonic spectral decomposition for multiple pitch estimation," *IEEE Trans. Audio, Speech, Language Process. (2006–2013)*, vol. 18, no. 3, pp. 528–537, 2010.

[16] E. Benetos and S. Dixon, "Multiple-instrument polyphonic music transcription using a temporally-constrained shift-invariant model," *J. Acoust. Soc. Amer.*, vol. 133, no. 3, pp. 1727–1741, 2013.

[17] B. Fuentes, R. Badeau, and G. Richard, "Harmonic adaptive latent component analysis of audio and application to music transcription," *IEEE Trans. Audio, Speech, Language Process. (2006–2013)*, vol. 21, no. 9, pp. 1854–1866, 2013.

[18] S. Sigtia, E. Benetos, and S. Dixon, "An end-to-end neural network for polyphonic piano music transcription," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 24, no. 5, pp. 927–939, 2016.

[19] R. Kelz, M. Dorfer, F. Korzeniowski, S. Böck, A. Arzt, and G. Widmer, "On the potential of simple framewise approaches to piano transcription," in *Proc. Int. Society Music Information Retrieval Conf.*, 2016, pp. 475–481.

[20] J. Nam, J. Ngiam, H. Lee, and M. Slaney, "A classification-based polyphonic piano transcription approach using learned feature representations," in *Proc.* Int. Society Music Information Retrieval Conf., 2011, pp. 175–180.

[21] M. Marolt, "A connectionist approach to automatic transcription of polyphonic piano music," *IEEE Trans. Multimedia*, vol. 6, no. 3, pp. 439–449, 2004.

[22] A. Cogliati, Z. Duan, and B. Wohlberg, "Context-dependent piano music transcription with convolutional sparse coding," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 24, no. 12, pp. 2218–2230, 2016.

[23] S. Ewert and M. B. Sandler, "Piano transcription in the studio using an extensible alternating directions framework," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 24, no. 11, pp. 1983–1997, 2016.

[24] C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. S. C. Raffel, J. Engel, S. Oore, and D. Eck, "Onsets and frames: Dual-objective piano transcription," in *Proc. Int. Society Music Information Retrieval Conf.*, 2018, pp. 50–57.

[25] V. Arora and L. Behera, "Multiple F0 estimation and source clustering of polyphonic music audio using PLCA and HMRFs," *IEEE/ACM Trans. Audio, Speech, Language Processing*, vol. 23, no. 2, pp. 278–287, 2015.

[26] E. Cambouropoulos, "Pitch spelling: A computational model," *Music Perception*, vol. 20, no. 4, pp. 411–429, 2003.

[27] H. Grohganz, M. Clausen, and M. Mueller, "Estimating musical time information from performed MIDI files," in *Proc. Int. Society Music Information Retrieval Conf.*, 2014, pp. 35–40.

[28] I. Karydis, A. Nanopoulos, A. Papadopoulos, E. Cambouropoulos, and Y. Manolopoulos, "Horizontal and vertical integration/segregation in auditory streaming: A voice separation algorithm for symbolic musical data," in *Proc. Sound and Music Computing Conf.*, 2007, pp. 299–306.

[29] A. Cogliati, D. Temperley, and Z. Duan, "Transcribing human piano performances into music notation," in *Proc. Int. Society Music Information Retrieval Conf.*, 2016, pp. 758–764.

[30] R. G. C. Carvalho and P. Smaragdis, "Towards end-to-end polyphonic music transcription: Transforming music audio directly to a score," in *2017 IEEE Workshop Applications Signal Processing Audio and Acoustics*, 2017, pp. 151–155.

[31] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. Advances Neural Information Processing Systems*, 2001, pp. 556–562.

[32] S. A. Abdallah and M. D. Plumbley, "Unsupervised analysis of polyphonic music by sparse coding," *IEEE Trans. Neural Netw. (1990–2011)*, vol. 17, no. 1, pp. 179–196, 2006.

[33] I. Goodfellow, A. Courville, and Y. Bengio, *Deep Learning.* Cambridge, MA: MIT Press, 2016.

[34] S. Böck and M. Schedl, "Polyphonic piano note transcription with recurrent neural networks," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, 2012, pp. 121–124.

[35] S. Wang, S. Ewert, and S. Dixon, "Identifying missing and extra notes in piano recordings using score-informed dictionary learning," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 25, no. 10, pp. 1877–1889, 2017.

[36] X. Serra, "A multicultural approach in music information research," in *Proc. 12th Int. Society Music Information Retrieval Conf.*, 2011, pp. 151–156.

SP

Estefanía Cano, Derry FitzGerald, Antoine Liutkus,
Mark D. Plumbley, and Fabian-Robert Stöter

Musical Source Separation

An introduction

©ISTOCKPHOTO.COM/TRAFFIC_ANALYZER

Many people listen to recorded music as part of their everyday lives, e.g., from radio or TV programs, compact discs, downloads, or, increasingly, online streaming services. Sometimes we might want to remix the balance within the music, perhaps to make the vocals louder or to suppress an unwanted sound, or we might want to upmix a two-channel stereo recording to a 5.1-channel surround sound system. We might also want to change the spatial location of a musical instrument within the mix. All of these applications are relatively straightforward, provided we have access to separate sound channels (stems) for each musical audio object.

However, if we only have access to the final recording mix, which is usually the case, this is much more challenging. To estimate the original musical sources, which would allow us to remix, suppress, or upmix the sources, we need to perform musical source separation (MSS).

In the general source separation problem, we are given one or more mixture signals that contain different combinations of some original source signals. This is illustrated in Figure 1, where four sources, i.e., vocals, drums, bass, and guitar, are all present in the mixture. The task is to recover one or more of the source signals given the mixtures. In some cases, this is relatively straightforward, e.g., if there are at least as many mixtures as there are sources and if the mixing process is fixed, with no delays, filters, or nonlinear mastering [1].

However, MSS is normally more challenging. Typically, there may be many musical instruments and voices in a two-channel recording, and the sources have often been processed with the addition of filters and reverberation (sometimes nonlinear) in the recording and mixing process. In some cases, the sources may move or the production parameters may change, meaning that the mixture is time varying.

Nevertheless, musical sound sources have particular properties and structures that can help us. For example, musical source signals often have a regular harmonic structure of frequencies at regular intervals and can have frequency contours characteristic of each musical instrument. They may also repeat, in particular, temporal patterns based on the musical structure.

Digital Object Identifier 10.1109/MSP.2018.2874719
Date of publication: 24 December 2018

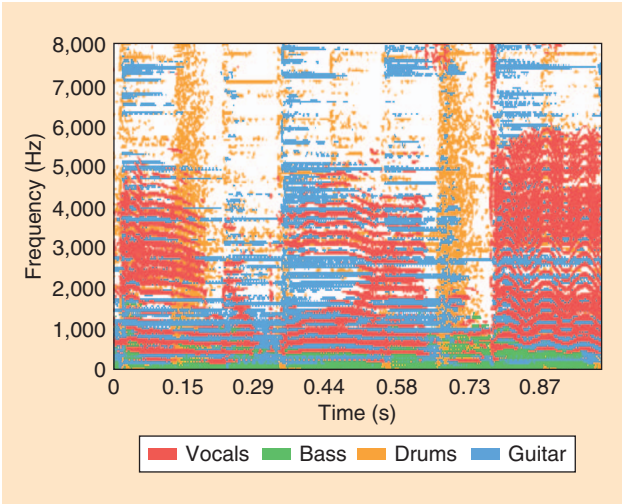


FIGURE 1. A representation of a music mixture in the TF domain. The dominant musical source in each TF bin is displayed with a different color.

In this article, we explore the MSS problem and introduce approaches to tackle it. We begin by presenting characteristics of music signals; we then introduce MSS and, finally, consider a range of MSS models. We also discuss how to evaluate the MSS approaches and discuss limitations and directions for future research. (A dedicated website with complementary information about MSS including sound examples, an extended bibliography, data set information, and accompanying code can be accessed at https://soundseparation.songquito.com/.)

Characteristics of music signals

Music signals have distinct characteristics that clearly differentiate them from other types of audio signals, such as speech or environmental sounds. These unique properties are often exploited when designing MSS methods, and so an understanding of these characteristics is crucial.

All music separation problems start with the definition of the desired musical source to be separated, often referred to as the *target source*. In principle, a musical source refers to a particular musical instrument, such as a saxophone or a guitar, that we wish to extract from the audio mixture. In practice, the definition of musical source is often more relaxed and can refer to a group of musical instruments with similar characteristics that we want to separate. This is the case, e.g., in singing voice separation, where the goal often includes the separation of both the main and background vocals from the mixture. In some cases, the definition of musical source can be even looser, as is the case in harmonic–percussion separation, where the aim is to separate the pitched instruments from the percussive ones.

In a general sense, musical sources are often categorized as predominantly harmonic, predominantly percussive, or as singing voice. Harmonic music sources mainly contain tonal components and, as such, are characterized by the pitch or pitches they produce over time. Harmonic signals exhibit a clear structure composed of a fundamental frequency F0 and a harmonic series. For most instruments, the harmonics appear at multiple integers of the fundamental frequency: for a given F0 at 300 Hz, a harmonic component can be expected close to 600 Hz, the next harmonic component around 900 Hz, and so on. Nonetheless, a certain degree of inharmonicity, i.e., deviation of harmonics from multiple integers of F0, should be expected and accounted for. Harmonic sources exhibit a relatively stable behavior over time and can typically be identified in the spectrogram as horizontal components. This can be observed in Figure 2, where a series of notes played by an acoustic guitar are displayed. Additionally, the trajectories in time of the F0 and the harmonics are usually very similar, a phenomenon referred to as *common fate* [2]. This can be clearly seen in the spectrogram of the vocals in Figure 2, where it can be observed that the vocal harmonics have common trajectories over time. The relative strengths of the harmonics, and the way that the harmonics evolve over time, contribute to the characteristic sound, or timbre, which allows the listener to differentiate one instrument from another. Furthermore, and particularly in Western music, the sources often play in harmony, where the ratios of the F0s of the notes are close to integer ratios. While harmony can result in homogeneous and pleasing sounds, it most often also implies a large degree of overlap in the frequency content of the sources, which makes separation more challenging.

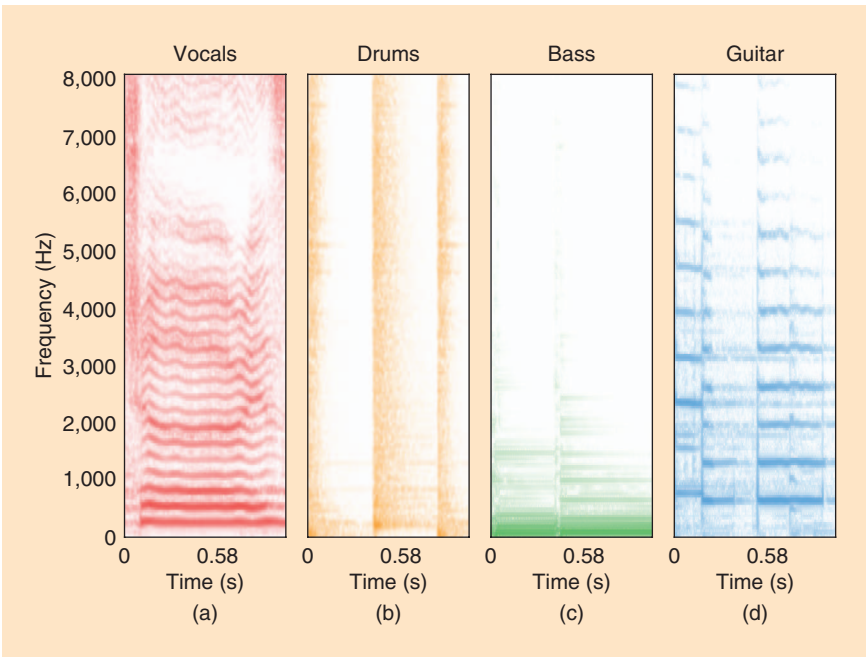


FIGURE 2. A magnitude spectrogram of four example music signals: (a) vocals, (b) drums, (c) bass, and (d) acoustic guitar.

In contrast to harmonic signals, which contain only a selected number of harmonic components, percussive signals contain energy in a wide range of frequencies. Percussive signals exhibit a much flatter spectrum and are highly localized in time with transient-like characteristics. This is apparent in the drums spectrogram in Figure 2, which shows clear vertical structures produced by the drums signal. Percussive instruments play a very important role of conveying rhythmic information in music, giving a sense of speed or tempo in a musical piece.

In reality, most music signals contain both harmonic and percussive elements. For example, a note produced by a piano is considered predominantly harmonic, but it also contains a percussive attack produced by the hammer hitting the strings. Similarly, singing voice is an intricate combination of harmonic (voiced) components, produced by the vibrations of the vocal chords and percussive-like (plosive) components including consonant sounds such as "k" or "p," where no vocal fold vibration occurs. These components are, in turn, filtered by the vocal cavity, with different formant frequencies created by changing the shape of the vocal cavity. As shown in Figure 2, singing voice typically exhibits a higher rate of pitch fluctuation compared with other musical instruments.

A notable property of musical sources is that they are typically sparse in the sense that for the majority of points in time and frequency, the sources have very little energy present. This is commonly exploited in MSS and can be clearly seen for each of the sources in Figure 2. Another characteristic of music signals is the fact that they typically exhibit repeating structures over different time scales, e.g., a repeating percussion pattern over a couple of seconds, to larger structures such as the verse-chorus structures found in pop songs. As explained in the section "Musical Source Models," these repetitions can be leveraged when performing MSS.

The bulk of research on MSS to date has focused on Western pop music as the main genre to be separated. It should be noted that other types of music present their own unique problems for MSS, e.g., many instruments playing in unison in some types of traditional/folk music. These cases are typically not covered by existing MSS techniques.

Once the target source has been defined, the characteristics of the audio mixture should be carefully considered when developing MSS methods. Modern music production techniques offer innumerable possibilities for transforming and shaping an audio mix. Most music signals today are created using audio content from a great diversity of origins, usually combined and mixed using a digital audio workstation (DAW), a software system for transforming and manipulating audio tracks. As depicted in Figure 3 for the trumpet, some musical sources can be recorded in a traditional manner using a microphone or a set of microphones. Very often, hardware devices that shape and color the sound are introduced into the signal chain; these may include, e.g., guitar tube amplifiers or distortion pedals that can impart very particular sound qualities to the signal. In other cases, musical sources are not captured using a microphone. Instead, the digital signal they produce is directly fed into the DAW, or alternatively created within the system itself, using, e.g., a keyboard as an interface as shown in Figure 3. Most frequently, an audio interface is used to facilitate the process of capturing input signals from different origins and delivering them to the DAW. The DAW itself offers many additional possibilities to further enhance and modify the signal.

Once all the audio content is in the DAW, the final step is the creation of the audio mixture. Most commercial music today is in stereo format (two channels). Other multichannel formats such as 5.1 (five main channels plus a low-frequency channel) are available but less common. The number of channels available to perform music separation is a key factor that can be exploited when designing MSS models. Multichannel mixtures allow spatial positioning of the music sources in the sound field. This means that a certain source can be perceived as coming from the left, the center, the right, or somewhere in between. The spatial positioning of the sources is usually achieved with a pan pot that regulates the contribution of each musical source in each of the available channels. This artificial creation of spatial positioning ignores delay as a spatial cue, and so interchannel delay is much less important in MSS than in speech source separation. In contrast, monophonic (single-channel) recordings offer no information about spatial positioning and are often the most challenging separation problem.

A final but fundamental aspect to be considered when designing MSS systems is the fact that music quality, and audio quality in general, is inherently defined and measured

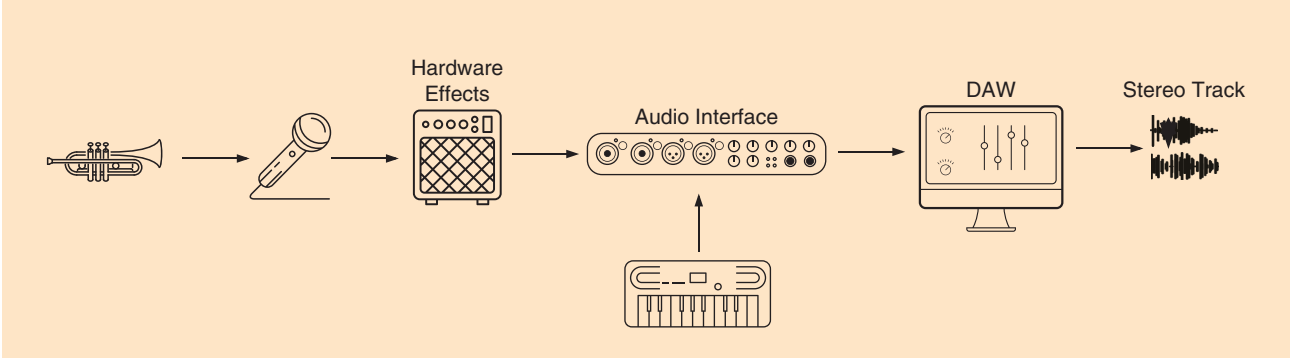


FIGURE 3. The common music recording and mixing setup. In most cases, musical content is combined and mixed into a stereo signal using a DAW.

by human perception. This sets an additional challenge to MSS methods: regardless of the mathematical soundness of the model, all systems are expected to result in perceptually pleasing music signals. Aside from the task of truthfully capturing the target source, we must also minimize the impact on perceptual quality of the distortions introduced in the separation process.

A typical MSS workflow

A high-level overview of the steps involved in most MSS systems is illustrated in Figure 4. First, the input mixture signal is transformed to the time–frequency (TF) domain. The TF representation of the signal is then manipulated to obtain parameters that model the individual sources in the mixture. These are then used to create filters to yield TF estimates of the sources. This is typically done in an iterative manner before the final estimated time-domain signals are recovered via an inverse TF transform. To describe these steps in greater detail, we first introduce the mathematical notation used.

Notation

MSS involves decomposing a time-domain audio mixture signal x into its constituent musical sources y_j. Both x and y_j are vectors of samples in time. Here, the index j denotes the musical source, with $j \in \{1 \cdots J\}$ and J the total number of sources in the mixture. TF representations are denoted in uppercase, with X denoting the complex spectrogram of the mixture x and Y_j denoting the complex spectrogram of the source y_j. Round brackets are used to denote individual elements in a TF representation, with frequency bins denoted with k and time frames denoted with n. The magnitude spectrogram of source j is defined as $S_j = |Y_j|$, whereas $\hat{Y}_j$, $\hat{S}_j$, and y_j denote estimates of the source TF representation, magnitude spectrogram, and source signal, respectively.

Interchannel delay is much less important in MSS than in speech source separation.

TF transformation

Most research in MSS has focused on the use of the short-time Fourier transform (STFT), $X(k, n) \in \mathbb{C}$. The complex STFT has the advantage of being computationally efficient, invertible, and linear: the mixture equals the sum of the sources in the transformed domain, $X = \Sigma_j Y_j$. Other transform alternatives like the constant Q transform have been proposed but have not, as yet, found widespread use in MSS.

Source modeling

Most MSS methods focus solely on analyzing the magnitude spectrogram $M = |X|$ of the mix. The goal at this stage is to estimate either a model of the spectrogram of the target source or a model of the location of the target source in the sound field. As explained in the "MSS Models" section, source and spatial models are the most common approaches used for MSS. Figure 4 shows an example where, starting with the mixture X, estimates of the magnitude spectrograms of the sources $\hat{S}_j$ are obtained.

Filtering

The goal at this stage is to estimate the separated music source signals given the source models. This is typically done using a soft-masking approach, the most common form of which is the generalized Wiener filter (GWF) [3], although other soft-masking approaches have been used [4]. Given X and $\hat{S}_j$, this allows recovery of the separated sources provided their characteristics are well estimated. The STFT of source $j = 1$ can be estimated elementwise using the GWF as $\hat{Y}_1(k, n) = X(k, n)\hat{S}_1(k, n)/\Sigma_j \hat{S}_j(k, n)$. The same procedure is applied for all sources in the mix. Essentially, each TF point in the original mixture is weighted with the ratio of the source magnitude to the sum of the magnitudes of all sources. This can be understood as a multiband equalizer of hundreds of bands, changed dynamically every few milliseconds to attenuate or let pass the required frequencies for the desired source. A special case of the GWF is the process of binary masking, where it is assumed that only one source has energy at a given TF bin, so that mask values are either 0 or 1.

Estimating source parameters from the mixture is not trivial, and it can be difficult to obtain good source parameters to enable successful filtering at the first try. For this reason, it is common to proceed iteratively. First, separation is achieved with the current estimated source models. These models are then updated from the separated signals, and the process is repeated as necessary. This approach is illustrated in Figure 4 and rooted in the expectation–maximization algorithm. Most of the models presented in the "MSS Models" section may be used in this iterative methodology.

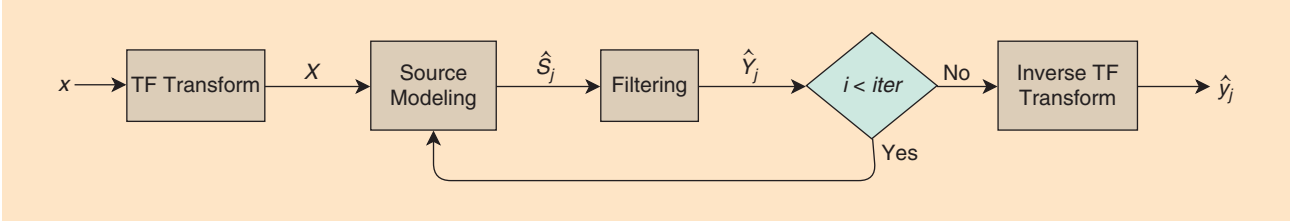


FIGURE 4. The common MSS workflow: source models are obtained from the spectrogram of the audio mix. This is often done in an iterative manner where *iter* represents the total number of iterations and i is the iteration index.

Inverse transform

The final stage in the MSS work flow is to obtain the time-domain source waveforms y_j using the inverse transform of the chosen TF representation.

MSS models

Having described the necessary steps for MSS, we now focus on how the unique characteristics of musical signals are used to perform MSS. While numerous categorizations of MSS algorithms are possible, such as categorization by source type, here we take the approach of dividing the algorithms into two broad categories: algorithms that model the musical sources and those that model the position of the sources in multichannel or stereo audio signals. The key distinction between these two categories is that algorithms in the first category model aspects of the mixture intrinsic to the sources, while those in the second category model aspects intrinsic to the recording/mixing process. Models in the two categories exploit distinct but complementary information that can readily be combined to yield more powerful MSS models.

Musical source position models

In the case of multichannel music signals, the spatial position of the sources has often been exploited to perform music source separation. In this section, we assume that we are dealing with a stereo (two-channel) mixture signal and that the spatial positioning of source j has been achieved using a constant power panning law, defined by a single parameter, the panning angle $\phi_j \in [0, \pi/2]$. For a given source q_j, its stereo representation (or stereo image) is given by $y_{1j} = q_j \cos\theta_j$ and $y_{2j} = q_j \sin\theta_j$, with the subscripts 1 and 2 explicitly denoting the first and second channels, respectively. Figure 5 illustrates the spatial positioning of three sources. The singing voice is positioned in the center and, hence, its stereo image is obtained with an angle of $\pi/4$.

One of the earliest techniques used to exploit spatial position for MSS was independent component analysis (ICA) [1], which estimates an unmixing matrix for the mixture signals based on statistical independence of the sources. However, ICA requires mixtures that contain the same number of channels as musical sources in the mix. This is not typically the case for music signals, where there are usually more sources than channels.

As a result of the shortcomings of ICA, algorithms that worked when there were more sources than channels were developed. Several techniques utilizing spatial position for separation, such as the Degenerate Unmixing Estimation Technique (DUET) [5], Azimuth Discrimination Resynthesis (ADRess) [6], and the PROJection Estimation Technique (PROJET) [7], assume that the TF representations of the sources have very little overlap. This assumption, which holds to a surprising degree for speech, does not hold entirely for music, where the use of harmony and percussion instruments ensures there is overlap. Nonetheless, this assumption has proved to be useful in many circumstances and often results in fast algorithms capable of real-time separation. The degree of overlap between sources can be seen in the spectrogram shown in Figure 1, which only shows the dominant musical source in each TF bin of the mixture.

To illustrate the usefulness of assuming very little overlap between sources, consider the elementwise ratio of the individual mixture channels in the TF domain $R(k, n) = |X_1(k, n)/X_2(k, n)|$, where the subscript numbers indicate the channel number. If there is little TF overlap between the sources, then a single source j will contribute most of the energy at a single point in the TF representations, and so $R(k, n) \approx \cos\phi_j / \sin\phi_j$. Given that $R(k, n)$ only depends on ϕ_j under this assumption, it can therefore be used to estimate a panning angle for each TF point. By plotting an energy-weighted histogram of these angle estimates, a mixture spatial histogram as shown in Figure 5 can be obtained. A peak in this histogram then provides an estimate of the panning angle ϕ_j of a given source. All TF points with an angle close to that of the jth peak are assigned to source j. Then, recovery of an estimate of source j can be done by means of binary masking. DUET, ADRess, and PROJET all estimate histograms of energy versus angle. The main difference between the techniques lies in how these histograms are generated and used and in the type of masking used. Both DUET and ADRess require peak picking from a mixture spatial histogram and use binary masking. PROJET estimates individual source position histograms, the superposition of which results in the mixture spatial position histogram (as shown in Figure 5). Furthermore, PROJET utilizes the GWF for masking. It should also be noted that both DUET and PROJET can also incorporate and deal with interchannel delays, extending their range of applicability.

The aforementioned separation methods directly model sound-engineering techniques such as panning and delay for creating multichannel mixtures. Another line of research models the

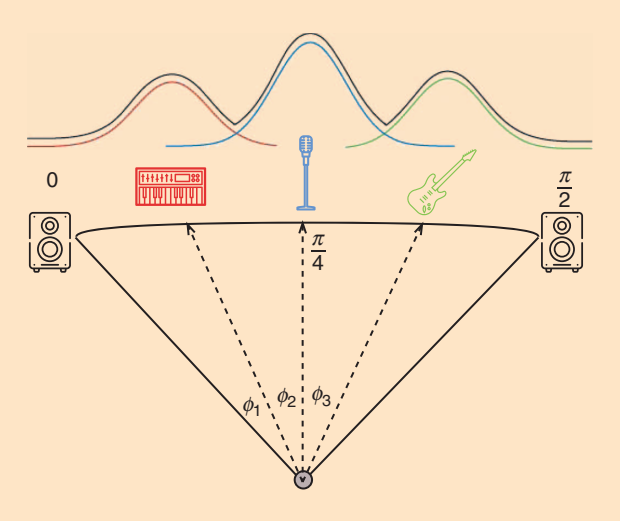


FIGURE 5. An illustration of standard Pan law. The position of the source $j = 1$ (keyboard), source $j = 2$ (voice), and source $j = 3$ (guitar) is defined by the angle ϕ_j, which is always measured with respect to the first channel. Also shown are individual (colored) source position histograms and the mixture spatial histogram (black).

spatial configuration of a source directly through interchannel correlations: at each frequency k, the correlation between the STFT coefficients of the different channels is calculated and encoded in a matrix called the *spatial covariance matrix*. The core idea of such methods is to leverage the correlations between channels to design better filters than those obtained by considering each channel in isolation. This approach is termed *local Gaussian modeling* (*LGM*) [3], [8]. It can give good separation whenever the spatial covariance matrices are well estimated for all sources. It is also able to handle highly reverberated signals, for which no single direction of arrival may be identifiable. This strength of covariance-based methods in dealing with reverberated signals comes at the price of difficult parameter inference. LGM algorithms are often very sensitive to initialization, and the estimated spatial covariance matrices alone are often not sufficient to allow separation. Successful LGM methods need to incorporate musical source models to further guide the separation to obtain acceptable solutions. Musical source models are discussed in the following section.

Musical source models

While spatial positioning can give good results if each source occupies a unique stereo position, it is common for multiple sources to be located in the same stereo position, or for the mixture signal to consist of a single channel. In these cases, model-based approaches that attempt to capture the spectral characteristics of the target source can be used. In the following sections, a range of MSS model-based approaches are described.

Kernel models

Similarly to the idea that the definition of the target sources can be relatively loose in an MSS scenario (see the "Characteristics of Music Signals" section), source models can also incorporate different degrees of specificity when it comes to describing the sources in the mix. Consider the harmonic–percussive separation task: harmonic sources are characterized as horizontal components in the spectrogram, while percussive sources are characterized as time-localized vertical ones. These particularities of the sources can also be understood as harmonic sources exhibiting continuity over time and percussive sources exhibiting continuity over frequency. Music separation models such as kernel additive models (KAMs) particularly exploit local features observable in music spectrograms, e.g., continuity, repetition (at both short and longer timescales such as repeating verses and choruses), and common fate [9]. To estimate a music source at a given TF point, the KAM involves selecting a set of TF bins, which, given the nature of the target source, e.g., percussive, harmonic, or vocals, are likely to be similar in value. This set of TF bins is termed a *proximity kernel*. An example of a vertical proximity kernel used to extract percussive sounds is shown in Figure 6(a). Here, a set of adjacent frequency bins are chosen since percussion instruments tend to exhibit stable ridges across frequency. The vertical kernel is also positioned on a percussive hit seen in the spectrogram of the mixture in Figure 6, where the middle TF bin (k, n) is the one to be estimated. In the case of sustained pitched sounds that tend to have similar values in adjacent time frames, a suitable proximity kernel consists of adjacent time frames across a single frequency [see the horizontal kernel in Figure 6(a)]. KAM approaches assume that interference due to other sources than the target source is sparse, so TF bins with interference are regarded as outliers. To remove these outliers and to obtain an estimate of the target source, the median amplitude of the bins in the proximity kernel is taken as an estimate of the target source at a given TF bin. The median acts as a statistical estimator robust to outliers in energy. Once the

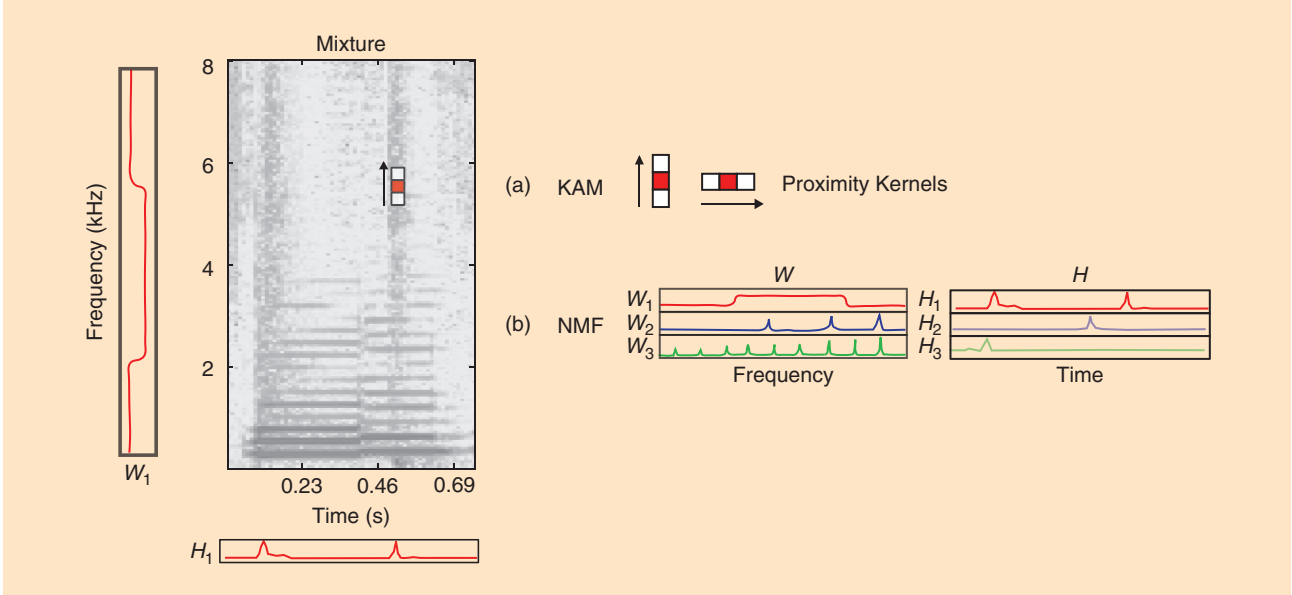


FIGURE 6. An example of different models used in MSS. (a) The proximity kernels used for harmonic–percussive separation within a KAM approach. (b) The spectral templates W and time activations H within an NMF approach.

proximity kernels have been chosen for each of the sources to be separated, separation proceeds iteratively in the manner described in the "A Typical MSS Workflow" section. The KAM is a generalization of previous work on vocal separation and harmonic–percussive separation [4], [10] and has demonstrated considerable utility for these purposes [9].

Spectrogram factorization models

Another group of musical source models is based on spectrogram factorization techniques. The most common of those is nonnegative matrix factorization (NMF). NMF attempts to factorize a given nonnegative matrix into two nonnegative matrices [11]. For the purpose of MSS, NMF can be applied to the nonnegative magnitude spectrogram of the mix M. The goal is to factorize M into a product $M \approx \mathrm{WH}$ of a matrix of basis vectors W, which is a dictionary of spectral templates modeling spectral characteristics of the sources, and a matrix of time activations H. Figure 6(b) shows a series of spectral templates and their corresponding time activations. One of the spectral templates W_1 and its corresponding time activations H_1 are also displayed next to the mixture spectrogram. Peaks in the time activations represent the time instances where a given spectral template has been identified within the mix. Note that the peaks in the time activation H_1 coincide with the percussive hits (vertical structures) in the spectrogram. The factorization task is solved as an optimization problem where the divergence (or reconstruction error) between M and WH is minimized using common divergence measures D such Kullback–Leibler or Itakura–Saito: $\min_{W,H \geq 0} D(M \| WH)$. Many variants of NMF algorithms have been proposed for the purpose of MSS, including methods with temporal continuity constraints [12], multichannel separation models [8], score-informed separation [13], among others [14].

The NMF-based methods described typically assume that the spectrogram for all sources may be well approximated through low-rank assumptions, i.e., as the combination of only a few spectral templates. While this assumption is often sufficient for instrumental sounds, it generally falls short on modeling vocals, which typically exhibit great complexity and require more sophisticated models. In this respect, an NMF variant that has been particularly successful in separating the singing voice uses a source-filter model to represent the target vocals [15]. The idea behind such models is that the voice signal is produced by an excitation that depends on a fundamental frequency (the source), while the excitation is then filtered by the vocal tract or by spectral shapes related to timbre (the filter). A dictionary of source spectral templates and a matrix of filter spectral shapes are used in this model within an expectation–maximization framework.

Some models for the singing voice are based on the observation that spectrograms of vocals are usually sparse, composed of strong and well-defined harmonics, and mostly zero elsewhere (as seen in Figure 2). In this setting, the observed mixture is assumed to be equal to the accompaniment for a large portion of the mixture spectrogram entries. This is the case of robust principal component analysis [16], which does not rely on overconstraining low-rank assumptions on the vocals and, in turn, uses the factorization only for the accompaniment, leaving the vocals unconstrained. Many elaborations on this technique have been proposed. For instance, [17] incorporates voice activity detection in the separation process, allowing the vocals to be inactive in segments of the signal and thus strongly boosting performance.

Sinusoidal models

Another strand of research for MSS models focuses on sinusoidal modeling. This method works under the premise that any music signal can be approximated by a number of sinusoids with time-varying frequencies and amplitudes [18]. Intuitively, sinusoidal modeling offers a clear representation of a music signal, which in most cases is composed of a set of fundamental frequencies and their associated harmonic series. If the pitches present in the target source, as well as the spectral characteristics of the associated harmonics of each pitch, are known or can be estimated, sinusoidal modeling techniques can be very effective for separation of harmonic sources. However, given the complexity of the model and the very detailed knowledge of the target source required to successfully create a realistic representation, the use of sinusoidal modeling techniques for MSS has been limited. Sinusoidal modeling techniques have been proposed for harmonic sound separation [19] and harmonic–percussive separation [20].

Deep neural network models

Historically, MSS research has focused heavily on the use of model-based estimation that enforced desired properties on the source spectrograms. However, if the properties required by the models are not present, separation quality can rapidly degrade. More recently, the use of deep neural networks (DNNs) in MSS has rapidly increased.

In contrast to the approaches described previously, which require explicit models of the source for processing, methods based on DNNs take advantage of optimization techniques to train source models in a supervised manner [21]–[23], i.e., using data sets where both the mix and the isolated sources are available. As depicted in Figure 7, most supervised DNN-based methods take magnitude spectrograms of the audio mix as inputs, optionally also incorporating some further context cues. The targets are set either as the magnitude spectrograms *Sj* of the desired sources (also shown in Figure 7) or as their separating masks (either soft masks or binary masks as described in the section "A Typical MSS Workflow"). Regardless of the inputs and targets used, DNN methods work by training the parameters of nonlinear functions to minimize the reconstruction error of the chosen outputs (spectrograms or masks) based on the inputs (audio mixes).

The models obtained from a DNN depend on two core aspects. First, the type and quantity of the data used for training is of primary importance. To a large extent, representative training data overcome the need for explicitly modeling the underlying spectral characteristics of the musical sources: they

are directly inferred by the network. Second, the DNN topology is of great significance, both for the training capabilities of the network and as a means of incorporating prior knowledge in the system.

The earliest DNN-based approaches for MSS consisted of taking a given frame of the spectrogram, as well as additional context frames as input, and outputting the corresponding frame for each of the targets. These systems mostly consisted of fully connected networks (FCNs). However, given the large size of music spectrograms, the resulting FCNs contained a large number of parameters. This restricted the use of temporal context in such networks to less than 1 s [22]. Therefore, these networks were typically applied on sliding windows of the mixture spectrogram. To overcome this limitation, both recurrent NNs (RNNs) and convolutional NNs were investigated as they offered a principled way to learn dynamic models. RNNs are similar to FCNs, except that they apply their weights recursively over an input sequence and can process sequential data of arbitrary length. This makes such dynamical models ideally suited for the processing of spectrograms of different tracks with different durations, while still modeling long-term temporal dependencies. The most commonly used setting for DNN-based separation today is to fix the number and the nature of the sources to separate beforehand. For example, we may learn a network able to separate drums, vocals, and bass from a mixture. However, having MSS systems that can dynamically detect and separate an arbitrary number of sources is an open challenge, and deep clustering [24] methods represent a possible approach to designing such systems.

A limitation of many DNN models is that they often use mean squared error (MSE) as a cost function. While MSE results in a well-behaved stochastic gradient optimization problem, it also poorly correlates with perceived audio quality. This is the reason why the design of more appropriate cost functions is also an active research topic in MSS [23].

Finally, a crucial limitation of current research for DNN-based MSS is the need for large amounts of training data. The largest MSS multitrack public data set today is MUSDB (https://doi.org/10.5281/zenodo.1117372), which comprises 10 h of data (refer to the accompanying website for more information about available data sets: https://soundseparation.songquito.com/evaluationOf.htm). However, this is still small compared with existing speech corpora comprising hundreds or thousands of hours. The main difficulty in creating realistic multitrack data sets for MSS comes from the fact that individual recordings for each instrument in a mixture are rarely available. Conversely, if individual recordings of instruments are available, the process of creating a realistic music mixture with them is time consuming and very costly (as outlined in the "Characteristics of Music Signals" section). As a result, designing DNN architectures for MSS still requires fundamental knowledge about the sources to be separated, their input and output representations, as well as the use of suitable signal processing techniques and postprocessing operations to further improve the recovered sources.

Evaluation of MSS models

Once an MSS system has been developed, its performance needs to be evaluated. All MSS approaches invariably introduce unwanted artifacts in the separated sources. These artifacts can be caused by mismatches between the models and the sources, musical noise that appears whenever rapid phase or spectral changes are introduced in the estimates (e.g., when using binary masking), reconstruction errors in resynthesis, as well as phase errors in the estimates. Quality evaluation in MSS systems is, however, nontrivial. As musical signals are intended to be heard by human listeners, it is therefore reasonable that evaluation should be based on subjective listening tests such as the Multiple Stimulus with Hidden Reference and Anchors test [25]. However, listening tests are time-consuming and costly, requiring human volunteers to take the tests, and need certain

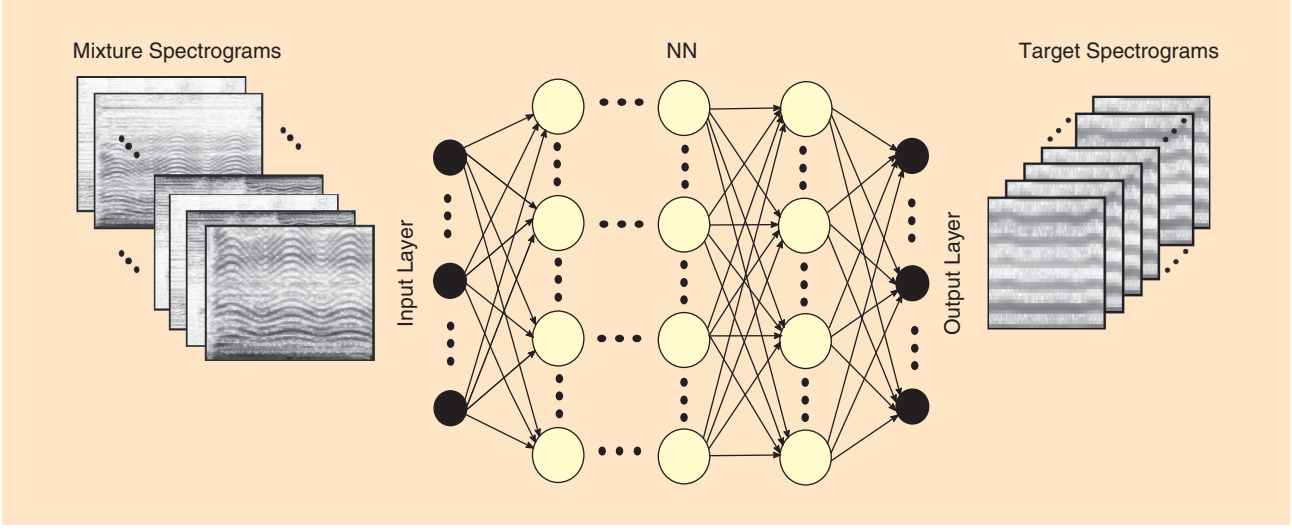


FIGURE 7. The DNN architecture for MSS. The mixture magnitude spectrograms are set as inputs, and source magnitude spectrograms of the desired source S_j are set as the targets.

expertise to be conducted properly. This has made them an infrequent choice for separation quality evaluation.

In an attempt to reduce the efforts of evaluating MSS systems, objective quality metrics have been proposed in the literature. These include the Blind Source Separation Evaluation (BSS Eval) toolbox [26], based on nonperceptual energy ratios, and the Perceptual Evaluation Methods for Audio Source Separation toolkit [27], which aimed to map results obtained via listening tests to create metrics. However, the validity of these metrics has been questioned in recent years, as the values obtained with them do not seem to correlate with perceptual measures obtained via listening tests [28].

DNNs have introduced a paradigm shift in MSS research, with an increasing focus on data-driven models.

Today, given the lack of a unified and perceptually valid strategy for MSS quality evaluation, most algorithm development is still conducted using BSS Eval; however, it is highly recommended to conduct a final listening test to verify the validity of separation results. As a final note, the source separation community runs a regular Signal Separation Evaluation Campaign (SiSEC) [29] including musical audio source separation tasks. SiSEC raises the visibility and importance of evaluation and acts as a focus for discussions on evaluation methodologies.

Future research directions

MSS is a challenging research area with numerous real-world applications. Due to both the nature of the musical sources and the very particular processes used to create music recordings, MSS has many unique features and problems that make it distinct from other types of source separation problems. This is further complicated by the need to achieve separations that sound good in a perceptual sense.

While the quality of MSS has greatly improved in the last decade, several critical challenges remain. First, audible artifacts are still produced by most algorithms. Possible research directions to reduce artifacts include the use of phase retrieval techniques to estimate the phase of the target source, the use of feature representations that better match human perception, allowing models to concentrate on the parts of the sounds that are most relevant for human listeners, and the exploration of MSS systems that model the signal directly in the time domain as waveforms.

Many remaining issues in MSS come from the fact that systems are often not flexible enough to deal with the richness of musical data. For example, it is typically assumed that the actual number of musical sources in a given recording is known. However, this assumption can lead to problems when the number of sources changes over the course of the training procedure. Another issue comes with the separation of sources from the same or similar instrument families, such as the separation of multiple singing voices or violin ensembles.

As previously mentioned, a unified, robust, and perceptually valid MSS quality evaluation procedure does not yet exist. Even while new alternatives for evaluation have been explored in recent years [30], listening tests remain the only reliable quality evaluation method to date. The design of new MSS quality evaluation procedures that are applicable for a wide range of algorithms and musical content will require large research efforts such as large-scale listening experiments, common data sets, and the availability of a wide range of MSS algorithms for use in development.

Additionally, a better understanding of how DNN-based techniques can be exploited for music separation is still needed. In particular, we need better training schemes to avoid overfitting and architectures suitable for music separation. The inclusion of perceptually based optimization schemes and availability of training data are also current challenges in the field.

Recent developments in the area of DNNs have introduced a paradigm shift in MSS research, with an increasing focus on data-driven models. Nonetheless, previous techniques have achieved considerable success in tackling MSS problems. We believe that combining the insights gained from previous approaches with data-driven DNN approaches will allow future researchers to overcome current limitations and challenges in MSS.

Acknowledgments

Mark D. Plumbley was partly supported by grants EP/L027119/2 and EP/N014111/1 from the U.K. Engineering and Physical Sciences Research Council and European Commission H2020 "AudioCommons" research and innovation grant 688382. Antoine Liutkus and Fabian-Robert Stöter were partly supported by the research program "KAMoulox" (ANR-15-CE38-0003-01) funded by the L'Agence Nationale de la Recherche, the French state agency for research.

Authors

Estefanía Cano (cano@idmt.frauhofer.de) received her B.Sc. degree in electronic engineering from the Universidad Potificia Bolivariana, Medellín-Colombia, in 2005, her B.A. degree in music–saxophone performance from the Universidad de Antioquia. Medellín-Colombia, in 2007, her M.Sc. degree in music engineering from the University of Miami, Florida, in 2009, and her Ph.D. degree in media technology from the Ilmenau University of Technology, Germany, in 2014. In 2009, she joined the Semantic Music Technologies group at the Fraunhofer Institute for Digital Media Technology, Germany, as a research scientist. In 2018, she joined the Music Cognition Group at the Agency for Science, Technology, and Research in Singapore. Her research interests include sound source separation, analysis and modeling of musical instrument sounds, and the use of music information retrieval techniques in musicological analysis.

Derry FitzGerald (derry.fitzgerald@cit.ie) received his B.Eng. degree in chemical engineering from the Cork Institute of Technology, Ireland, in 1995 and his M.A. degree in music technology and Ph.D. degree in digital signal processing both from the Dublin Institute of Technology, Ireland, in 2000 and 2004, respectively. He has worked as a research fellow at both Cork and Dublin Institutes of Technology and is currently the chief technology officer at AudioSourceRE, an Irish-based

start-up company developing sound source separation technologies. His research interests are in the areas of sound source separation and tensor factorizations.

Antoine Liutkus (antoine.liutkus@inria.fr) received his state engineering degree from Telecom ParisTech, France, in 2005, his M.Sc. degree in acoustics, computer science, and signal processing applied to music (Acoustique, Traitement du Signal, Informatique, Appliqués à la Musique) from the Université Pierre et Marie Curie (Paris VI), France, in 2005, and his Ph.D. degree in electrical engineering from Telecom ParisTech in 2012. He worked as a research engineer on source separation at Audionamix, Paris, France, from 2007 to 2010. He is currently a researcher in the speech processing team at Inria Nancy Grand Est located in Villers-lès-Nancy, France. His research interests include audio source separation and machine learning.

Mark D. Plumbley (m.plumbley@surrey.ac.uk) received his Ph.D. degree in neural networks from the Engineering Department at Cambridge University, United Kingdom, in 1991 and became a lecturer at King's College London, United Kingdom. He moved to Queen Mary University, London, United Kingdom, in 2002, later becoming a professor of machine learning and signal processing and the director of the Centre for Digital Music. In 2015, he joined the University of Surrey, United Kingdom, as a professor of signal processing in the Centre for Vision, Speech, and Signal Processing. His research interests include the analysis and processing of audio and music and using a wide range of signal processing techniques, including independent component analysis and sparse representations.

Fabian-Robert Stöter (fabian-robert.stoter@inria.fr) received his diploma degree in electrical engineering from the Leibniz University of Hanover, Germany, in 2012 and worked toward his Ph.D. degree in audio signal processing in the research group of B. Edler at the International Audio Laboratories Erlangen, Germany. He is currently a researcher at Inria/Laboratory of Computer Science, Robotics, and Microelectronics in Montpellier, France. His research interests include supervised and unsupervised methods for audio source separation and signal analysis of highly overlapped sources.

References

[1] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*. New York: Wiley, 2001.

[2] A. S. Bregman, *Auditory Scene Analysis: The Perceptual Organization of Sound*. Cambridge, MA: MIT Press, 1990.

[3] N. Q. K. Duong, E. Vincent, and R. Gribonval, "Under-determined reverberant audio source separation using a full-rank spatial covariance model," *IEEE Trans. Audio Speech Language Processing*, vol. 18, no. 7, pp. 1830–1840, Sept. 2010.

[4] Z. Rafii and B. Pardo, "Repeating pattern extraction technique (REPET): A simple method for music/voice separation," *IEEE Trans. Audio Speech Language Processing*, vol. 21, no. 1, pp. 73–84, Jan. 2013.

[5] S. Rickard, "The DUET blind source separation algorithm," in *Blind Speech Separation, Signals and Communication Technology*, S. Makino, H. Sawada, T. W. Lee, Eds. Dordrecht, Netherlands: Springer, 2007, pp. 217–241.

[6] D. Barry, B. Lawlor, and E. Coyle, "Real-time sound source separation using azimuth discrimination and resynthesis," in *Proc. 117th Audio Engineering Society (AES) Conv.*, San Francisco, CA, 2004, pp. 1–7.

[7] D. FitzGerald, A. Liutkus, and R. Badeau, "Projection-based demixing of spatial audio," *IEEE/ACM Trans. Audio Speech, Language Processing*, vol. 24, no. 9, pp. 1560–1572, Sept. 2016.

[8] A. Ozerov and C. Févotte, "Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation," *IEEE Trans. Audio Speech Language Processing*, vol. 18, no. 3, pp. 550–563, Mar. 2010.

[9] A. Liutkus, D. Fitzgerald, Z. Rafii, B. Pardo, and L. Daudet, "Kernel additive models for source separation," *IEEE Trans. Signal Processing*, vol. 62, no. 16, pp. 4298–4310, Aug. 2014.

[10] D. FitzGerald, "Harmonic/percusssive separation using median filtering," in *Proc. 13th Int. Conf. Digital Audio Effects (DAFx)*, Graz, Austria, 2010, pp. 1–4.

[11] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances Neural Inform. Processing Syst.*, vol. 13, pp. 556–562, Apr. 2001.

[12] T. Virtanen, "Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria," *IEEE Trans. Audio Speech Language Processing*, vol. 15, no. 3, pp. 1066–1074, Mar. 2007.

[13] S. Ewert, B. Pardo, M. Müller, and M. D. Plumbley, "Score-informed source separation for musical audio recordings: An overview," *IEEE Signal Processing Mag.*, vol. 31, no. 3, pp. 116–124, May 2014.

[14] P. Smaragdis, C. Fevotte, G. J. Mysore, N. Mohammadiha, and M. Hoffman, "Static and dynamic source separation using nonnegative factorizations: A unified view," *IEEE Signal Processing Mag.*, vol. 31, no. 3, pp. 66–75, May 2014.

[15] J. L. Durrieu, G. Richard, B. David, and C. Fevotte, "Source/filter model for unsupervised main melody extraction from polyphonic audio signals," *IEEE Trans. Audio Speech Language Processing*, vol. 18, no. 3, pp. 564–575, Mar. 2010.

[16] P.-S. Huang, S. D. Chen, P. Smaragdis, and M. Hasegawa-Johnson, "Singing-voice separation from monaural recordings using robust principal component analysis," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, Kyoto, Japan, 2012, pp. 57–60.

[17] T.-S. Chan, T.-C. Yeh, Z.-C. Fan, H.-W. Chen, L. Su, Y.-H. Yang, and R. Jang, "Vocal activity informed singing voice separation with the ikala dataset," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, Brisbane, Australia, 2015, pp. 718–722.

[18] X. Serra, "Musical sound modeling with sinusoids plus noise," in *Studies on New Music Research*, M. Leman and P. Berg, Eds. Swets & Zeitlinger, 1997, pp. 91–122.

[19] T. Virtanen and A. Klapuri, "Separation of harmonic sound sources using sinusoidal modeling," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, Budapest, Hungary, 2000, pp. II765–II768.

[20] E. Cano, M. Plumbley, and C. Dittmar, "Phase-based harmonic/percussive separation," in *Proc. 15th Annu. Conf. Int. Speech Communication Association Interspeech*, Singapore, 2014, pp. 1628–1632.

[21] P. S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Joint optimization of masks and deep recurrent neural networks for monaural source separation," *IEEE/ACM Trans. Audio Speech Language Processing*, vol. 23, no. 12, pp. 2136–2147, 2015.

[22] S. Uhlich, F. Giron, and Y. Mitsufuji, "Deep neural network based instrument extraction from music," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, Brisbane, Australia, 2015, pp. 2135–2139.

[23] A. A. Nugraha, A. Liutkus, and E. Vincent, "Multichannel audio source separation with deep neural networks," *IEEE/ACM Trans. Audio Speech Language Processing*, vol. 24, no. 9, pp. 1652–1664, Sept. 2016.

[24] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, Shanghai, China, 2016, pp. 31–35.

[25] International Telecommunication Union. (2015, Oct.). Recommendation BS.1534-3: Method for the subjective assessment of intermediate quality levels of coding systems. [Online]. Available: https://www.itu.int/rec/R-REC-BS.1534-3-201510-I/en

[26] E. Vincent, R. Gribonval, and C. Fevotte, "Performance measurement in blind audio source separation," *IEEE Trans. Audio Speech Language Processing*, vol. 14, no. 4, pp. 1462–1469, July 2006.

[27] V. Emiya, E. Vincent, N. Harlander, and V. Hohmann, "Subjective and objective quality assessment of audio source separation," *IEEE Trans. Audio Speech Language Processing*, vol. 19, no. 7, pp. 2046–2057, Sept. 2011.

[28] E. Cano, D. FitzGerald, and K. Brandenburg, "Evaluation of quality of sound source separation algorithms: Human perception vs quantitative metrics," in *Proc. European Signal Processing Conf. (EUSIPCO)*, Budapest, Hungary, 2016, pp. 1758–1762.

[29] A. Liutkus, F.-R. Stöter, Z. Rafii, D. Kitamura, B. Rivet, N. Ito, N. Ono, and J. Fontecave, "The 2016 signal separation evaluation campaign," in *Proc. Int. Conf. Latent Variable Analysis and Signal Separation*, 2017, pp. 323–332.

[30] D. Ward, H. Wierstorf, R.D. Mason, E. M. Grais, and M.D. Plumbley, "BSS Eval or PEASS? Predicting the perception of singing-voice separation," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, Calgary, Canada, 2018.

SP

Juhan Nam, Keunwoo Choi, Jongpil Lee,
Szu-Yu Chou, and Yi-Hsuan Yang

Deep Learning for Audio-Based Music Classification and Tagging

Teaching computers to distinguish rock from Bach

©ISTOCKPHOTO.COM/TRAFFIC_ANALYZER

Over the last decade, music-streaming services have grown dramatically. Pandora, one company in the field, has pioneered and popularized streaming music by successfully deploying the Music Genome Project [1] (https://www.pandora.com/about/mgp) based on human-annotated content analysis. Another company, Spotify, has a catalog of over 40 million songs and over 180 million users as of mid-2018 (https://press.spotify.com/us/about/), making it a leading music service provider worldwide. Giant technology companies such as Apple, Google, and Amazon have also been strengthening their music service platforms. Furthermore, artificial intelligence speakers, such as Amazon Echo, are gaining popularity, providing listeners with a new and easily accessible way to listen to music.

While music-streaming services have made a huge volume of music accessible to users, the enormous size of the service catalogs has created the challenge of finding among so many choices the songs that fit users' tastes. A general approach to this issue has been collaborative filtering, which predicts songs of potential interest based on previous usage data, such as play history and song rating. Although collaborative filtering effectively retrieves songs and accommodates personalized recommendations, its performance is hampered by such issues as popularity bias and the cold-start problem, the challenge of recommending new music to users [2]. The content-based approach is often regarded as a supplementary solution to those problems. Pandora radio is a representative example as it retrieves songs by exploiting the similarities of song descriptors, such as genre, mood, instruments, and vocal quality. However, high-quality manual annotation is costly and not scalable, suggesting a need for better ways to automate classification of music content. As a result, much attention in the field of music information retrieval (MIR) over the last few years has centered on finding ways to automate the process of classifying music genre and mood and tagging music. Hereafter, this article will use the term *music classification and tagging* as a general expression for tasks that involve taking music audio data as input and automatically annotating them with a certain form of semantic label.

Digital Object Identifier 10.1109/MSP.2018.2874383
Date of publication: 24 December 2018

The focus of a survey paper on music classification and tagging in 2011 [3] revealed the previous trends in the field. Most of the 149 papers surveyed therein were based on the "conventional" machine-learning framework, which involves a pipeline of feature extraction and classifier learning. The features were mostly manually designed to succinctly represent acoustic or musical characteristics given the task. However, recent breakthroughs using deep neural networks have shifted the paradigm to learning representations in an end-to-end manner, which has opened the era of deep learning [4], [5]. This method has been applied to various tasks in MIR as well [6]. For several reasons, researchers have been especially active in exploring the problems of music classification and tagging. First, music classification and tagging tasks annotate audio clips at a track level (i.e., segments lasting several seconds or longer), and the audio clips are typically represented as two-dimensional (2-D) image-like data, such as mel-spectrograms. This is similar to the way images are classified, which means the technique may be borrowed and applied in the field of music classification. Second, an essential ingredient of deep learning is the availability of large data sets. One is the Million Song Data Set (MSD), which was introduced in 2011 [7]. MSD has facilitated large-scale training of deep neural networks for music classification and tagging tasks. Last, successful efforts to automate music classification have drawn interest from the music-streaming service industry, leading to investment in research resources to develop advanced content-based approaches [1], [8].

While the latest developments in other domains have inspired parallel developments in music, it is still necessary to take into consideration the specific properties of music signals when developing deep-learning methods for music classification and tagging. This article, an up-to-date tutorial-like survey, reviews the representative deep-network designs tailored for music classification and tagging, the best practices found thus far, the applications to music services and other MIR tasks, and, finally, the limitations and open issues that still need to be addressed.

It is still necessary to take into consideration the specific properties of music signals when developing deep-learning methods for music classification and tagging.

From feature engineering to end-to-end learning

Humans classify or annotate music based on diverse characteristics extracted from the audio signals. For example, a heavily distorted electric guitar sound with growling vocals is a good indication of metal music. Swing rhythms, syncopation, and chromatic comping by polyphonic instruments (e.g., piano or guitars) are obvious cues that the music is jazz. Translating these acoustic and musical features into numerical representations that computers can interpret is the essence of music classification and tagging. This usually involves a series of computation steps that convert audio content into a time–frequency representation, extract discriminative features, summarize them over time, and repeat the feature extraction and summarization until the proper category for the music can be determined. The way of improving each feature extraction step to achieve the best performance has evolved with advances in learning algorithms from hand engineering based on domain knowledge to end-to-end learning. Humphrey et al. [9] explained the transition in a unified deep architecture model where multiple blocks of affine transformation, nonlinear function, and optional pooling operation are pipelined. Figure 1 illustrates four different feature representation approaches in their framework. In reviewing the evolution of such approaches,

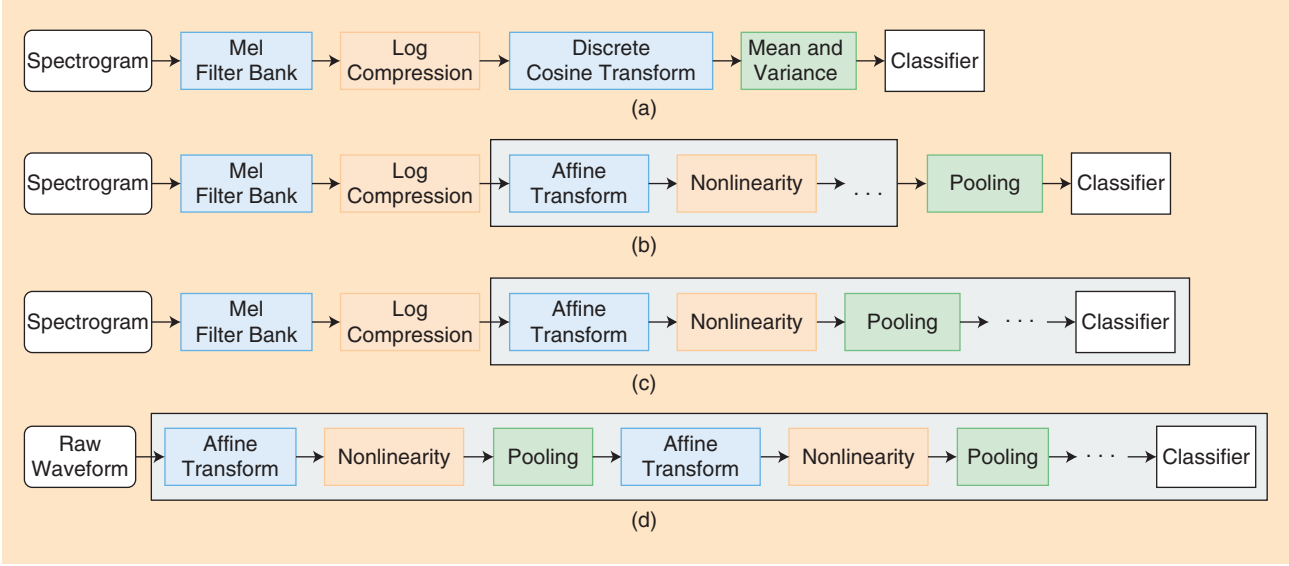


FIGURE 1. The transition of feature representation for music classification: (a) feature engineering [mel-frequency cepstral coefficients (MFCCs)], (b) low-level feature learning, (c) convolutional neural networks, and (d) end-to-end learning. The blocks inside the black lines indicate that they are learned by the algorithms.

we first separate them into two classes: feature engineering and feature learning.

Feature engineering

A single line of melody can be arranged and performed in any of a variety of genres or moods of music, depending on the choice of instruments, chord progressions, rhythms, dynamics, and other musical elements. Considering the generative process in creating music, an intuitive approach to music classification and tagging would require features on each axis of the musical elements to be distilled and their distributions to be modeled. The traditional approach attempted to craft a variety of audio features under this principle. A representative example is the seminal work by Tzanetakis and Cook [10]. They tackled the automatic music genre classification problem by using three groups of audio features: timbre, pitch, and rhythm. The timbre feature was formed by summarizing the zero-crossing rate, low-level spectral features, low-energy feature, and MFCCs within a texture window. The pitch feature was extracted by encapsulating the pitch content from a multipitch estimator into two types of histograms, one that contains harmony information and one that contains pitch-range information. The rhythmic feature was represented by a beat histogram that explains temporal regularity by counting intervals of periodic energy fluctuation via a subband analysis. Finally, they combined all features and applied them to classifiers, such as the *k*-nearest neighbors and Gaussian mixture models. Since this study laid a foundation for music classification and tagging, numerous research studies have developed new or better-tuned audio features and have followed the two-stage framework, where the hand-engineered features are used as input of a standard classifier.

This feature-engineering approach designs each computation step manually based on the domain knowledge. For example, Figure 1(a) shows the computation pipeline of MFCCs. The mel filter bank and discrete cosine transform are tailored based on psychoacoustics and signal processing knowledge, respectively. These hand-engineered features have advantages in that they are interpretable and usually expressed in a compact form. However, most hand-engineered audio features are based on short-time analysis and may not capture high-level information in music. In addition, the engineering process is separated from the data-driven optimization in the classifier. Currently, this two-stage approach seems to lead to an imperfect solution.

Feature learning

The gist of deep learning is that the feature representations of input data can be learned by the algorithm via the deep neural networks. That is, learning is achieved layer by layer, with higher-level features learned in the deeper layers. This contrasts with the feature-engineering approach in that the domain knowledge is much less involved in finding the features and the input data are processed at a minimum level before they are fed into the algorithm. In the tide of deep learning, various feature-learning algorithms have been introduced and applied to music classification and tagging. We categorize them into the following three classes: low-level feature learning, convolutional neural networks (CNNs), and end-to-end learning models.

Low-level feature learning

Early studies focused on learning low-level audio features to replace the handcrafted features in the two-stage framework, as illustrated in Figure 1(b). One kind of research focused on learning a meaningful dictionary of spectrograms using unsupervised learning algorithms, such as the restricted Boltzmann machine, *K*-means, and sparse coding (e.g., [11]). These shallow feature-learning algorithms are usually trained to encode multiple frames of spectrograms into a high-dimensional sparse feature vector. They capture a variety of musically interpretable time–frequency patterns. The other kind of research focused on supervised feature learning that maps a short-term spectrum to genre or mood labels with a pretrained multilayer perceptron or deep belief networks (e.g., [12]). The hidden layer activations are used as learned features. While both groups deliver better performance than that using hand-engineered features in many music classification and tagging tasks, they are still limited to low-level feature learning, and the adopted framework still has two stages.

CNNs

Lately, CNNs have been the most widely used learning model in music classification and tagging tasks [13], [14]. Based on several seconds of audio as an input, CNNs can be improved in an end-to-end fashion to learn hierarchical features. However, as shown in Figure 1(c), most successful CNN models used the spectrogram (particularly, mel-spectrogram) as the input representation, indicating that domain knowledge is still helpful. Under different assumptions of locality and translation invariance on the time–frequency representation, several configurations of CNN models have been suggested [13], [14]. We describe more details about such models in the next section.

End-to-end learning models

More recently, a few attempts have been made to directly use raw waveforms as the input of CNNs [8], [13], [15]. As illustrated in Figure 1(d), no single step requires a hand-designed representation, thus realizing a complete end-to-end feature learning. Lee et al. proposed a successful model in music classification and tagging [15], [16]. They found that the model performs better when the bottom convolutional layer takes a small grain of samples (e.g., two or three samples) rather than a typical window size (e.g., 256 or 512 samples). However, as the filter size in the convolutional layer is smaller, the model becomes progressively deeper, and, as a result, it takes longer to train. More details about this type of model are described in the next section.

Deep-learning models

In this section, we review three representative CNN models for music classification. The first two models are one-dimensional (1-D) [13] and 2-D CNNs [14], each of which has been applied in efforts to make networks more flexible. This trend toward

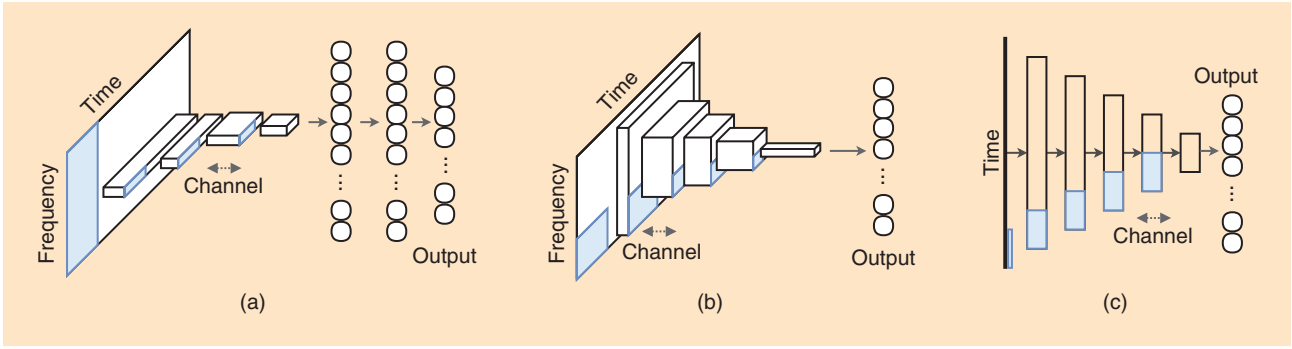


FIGURE 2. Block diagrams of (a) 1-D, (b) 2-D, and (c) sample-level CNNs. (a) and (b) are based on 2-D time–frequency representation inputs (e.g., mel-spectrograms or short-time Fourier transforms), and (c) is based on a time-series input.

greater flexibility continues with the most recent and most successful approach, the sample-level CNN [15], where a time-series audio signal is used as input. Additionally, we will introduce a few advanced methods that can improve performance. While there are many other kinds of architectures, we focus on CNN-based ones in this article, as they are more widely used.

For the sake of clarity, in this section we specify layers using Keras-style grammar (https://keras.io). A 2-D kernel is specified by its lengths in the frequency (f) and time (t) axes, e.g.,(f, t). A convolutional layer with 2-D kernels measuring (f, t), N channels, (s_1, s_2) strides, and "valid" (or "same") padding is denoted as Conv2D[filters $= N$, kernel_size $= (f, t)$, strides $= (s_1, s_2)$, padding $=$ "valid"] with some of the parameters omitted if they follow the aforementioned default values. In addition, the parameter names can be omitted while keeping their order (i.e., like Python syntax). Conv1D is defined similarly, but the kernel size and stride are 1-D. Max-pooling layers are defined as MP1D(pool_size) and MP2D(pool_size). Finally, we specify the size of a feature map with (F, T, N) for lengths of F in the frequency axis, T in the time axis, and N channels.

1-D CNNs

Dieleman et al. [13] initiated some of the earliest advancements in the area of deep learning in music classification and tagging. Dieleman also made a significant early contribution with his blog post about his internship with Spotify (http://benanne.github.io/2014/08/05/spotify-cnns.html). The network structure is illustrated in Figure 2(a), and we call it *1-D CNN* in this article. Here, "1-D" refers to the dimensionality of the first layer's convolution operation and should not be confused with that of the kernel.

The assumed behavior of 1-D CNNs with respect to music signal input is straightforward. As mentioned previously, 1-D CNNs take a time–frequency representation, such as mel-spectrogram, as input. With the kernel height of F, the first convolutional layer "sees" the entire frequency range at once. That is to say, during training, the network finds some patterns that cover the entire frequency range. For example, the size of the first convolutional layer's kernel in [13] is (128, 4) with the number of output channels as 256, i.e., Conv2D[256, (128, 4), "valid"], resulting in (1, 599, 256)-sized feature maps. More convolutional layers and densely connected layers are shown as in Figure 2(a). This structure is musically plausible in some sense, as it puts a strong prior to the network design at the same time. To elaborate, we know that in images an object can appear in any location, making 2-D CNNs a popular design choice, as 2-D CNNs can deal with such spatial variants. However, this may not be the case for musical audio. In a time–frequency representation, a musical object or pattern can appear anytime, but not in any frequency band. This is because different musical components can exist in different frequency ranges with a minor shift. In other words, the invariance property we want to have may be mostly along the time axis. This characteristic enables us to see and interpret what is learned at the first convolutional layer. Because the learned kernels operate directly on the spectrogram input, we can visualize the kernels using the learned weights and see which genres of songs maximally activate them. For example, in Figure 3, we show the top four relevant tags for a few selected kernels. The tags are sorted (in descending order from top to bottom) based on the tag activation score of each kernel. We can see that the corresponding tags somehow explain each of the learned kernels.

One-dimensional CNNs are computationally efficient. Its first convolutional layer takes the entire frequency range, makes the feature maps of the subsequent layers much smaller (the length of the frequency axis becomes 1), and, accordingly, drastically reduces the total number of network parameters. However, this is actually a double-edged aspect of 1-D CNNs. A small number of parameters means it is easier to train the network with relatively small data sets. At the same time, it means that 1-D CNNs will not fully benefit from the development of hardware resources and large-scale data sets due to their limited representation power.

The aforementioned assumption, or the strong prior, of 1-D CNNs introduces a clear limitation: a complete lack of frequency-axis shift-invariance. In the first layer, the 128-dimensional frequency components are assumed to have their own meaning; therefore, a slight change along the frequency axis (i.e., pitch transposition) results in a significantly different activation. Using a slightly smaller kernel [e.g., (126, 4)] has been proposed as an alternative (while making it technically 2-D convolution), but it only provides a "global" shift-invariance. In other words, assuming a max-pooling of $(3, x)$ follows, this alternative approach is invariant to a global transposition by two

semitones. However, it is not invariant to local changes, e.g., a combination of an ϵ_1 frequency shift in the bass guitar component, an ϵ_2 shift in the vocal component, and an ϵ_3 shift in the piano component, where ϵs are different (unlike in the case of 2-D CNNs). As a result, the representations that 1-D CNNs learn in the first layer are limited to some common patterns of the entire frequency ranges.

2-D CNNs

With larger data sets and better hardware resources becoming available, a natural step is to increase network flexibility to improve representation learning, as in [14]. The network structure is illustrated in Figure 2(b). We call it *2-D CNNs* in contrast to 1-D CNNs, as they focus on the contiguous 2-D convolutional layers including the first one. The five-layer structure in [14], for example, gradually combines smaller time–frequency patterns to create larger ones with 2-D convolutional layers, e.g., Conv2D[32, (3, 3)], which allow for small shifts by the following max-pooling layers, e.g., MP2D[(2, 2)]. Since the kernel sizes are small, the padding strategy ("valid" or "same") is not of very much interest.

Two-dimensional CNNs assume that more flexibility will be helpful in finding the time–frequency patterns. The flexibility can have several aspects: the shift (or location) invariance along both axes, the size of the patterns, and small distortions within the patterns. They are realized by 2-D convolutional layers with small kernels (typically three-by-three) and 2-D max-pooling layers. Although this may contradict the different meanings of time and frequency axes mentioned in the previous section, 2-D CNNs have, in fact, performed better than 1-D CNNs. Thanks to their simple structure and good performance, 2-D CNNs may now be the most popular approach for music audio classification.

Two-dimensional CNNs usually demand better hardware than 1-D CNNs for two reasons. First, the parameters easily outnumber those of 1-D CNNs due to the use of contiguous 2-D kernels, which then require more memory. Second, the training and use of 2-D CNNs add a significant computation burden due to the large size of the feature maps, along which the kernel should be convolved. For example, with a 1-D CNN, all of the feature maps are of size $(1, x, N)$. The frequency axis is always of length 1, which makes the feature maps 1-D with channels. In contrast, with a 2-D CNN, the feature maps would be of size (F, x, N), i.e., 2-D with channels. This significantly increases the computation in both the forward and backward passes of the model training process.

So far, we have reviewed the advantages and disadvantages of 2-D CNNs as compared with 1-D CNNs. In practice, 2-D CNNs offer some practical advantages. For example, improvements in hardware have enabled researchers and practitioners to use 2-D CNNs when they have sufficient data. Once the bottlenecks of the data size and hardware resource are resolved, the flexibility of 2-D CNNs may bring about better performance. Empirical evidence provided in [17] compares various CNN architectures according to number of parameters, computation use, and performance.

Sample-level CNNs

As explained in the previous subsection, 2-D CNNs may lead to better results in music classification and tagging, as they provide more flexibility. Sample-level CNNs go further in the same direction by discarding the 2-D time–frequency input preprocessing stage and learning directly from the audio waveforms in an extremely granular way [15]. Although it was not the first approach that directly learns representations from the raw audio, it is the first architecture that has achieved a state-of-the-art performance with a significantly shorter kernel size than the regular window size in short-time analysis with a deep network.

Among the variations in [15], we explain the details of the 3^9 model structure. As illustrated in Figure 2(c), the

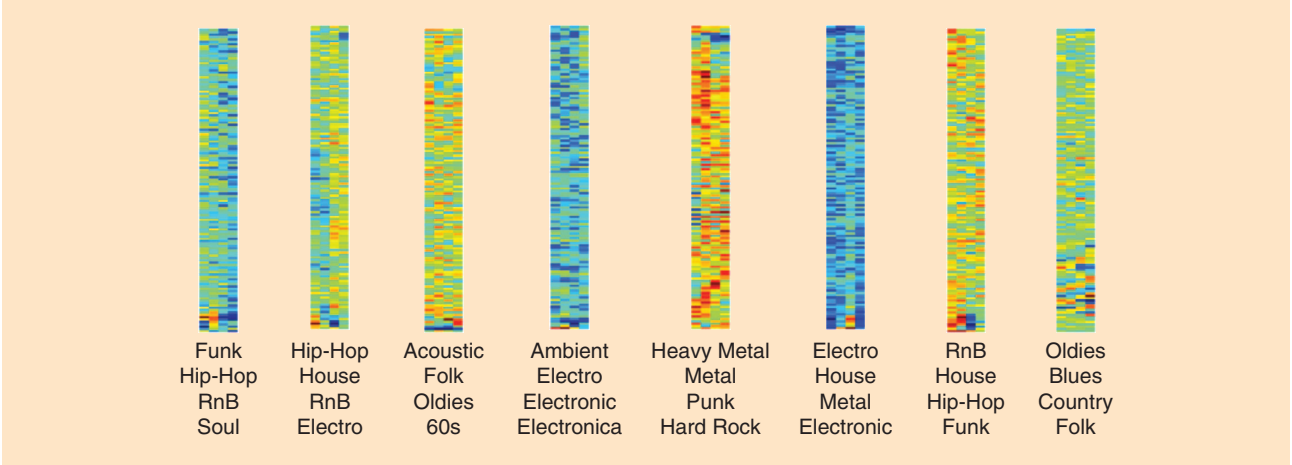


FIGURE 3. Visualization of the first convolution kernels in a trained 1-D CNN with relevant tags. For each kernel, an activation score for each tag is calculated by computing the average activation of the kernel for all songs with the tag. The four tags are those with the highest activation scores for the selected kernel. The kernels are of size 128 × 4, where 128 is the number of mel bins and 4 is the number of frames in time. The learned kernels can be interpreted as spectrotemporal patterns associated with acoustic characteristics of music with the tags. RnB: rhythm and blues.

model consists of one [Conv1D(filters = 128, kernel_size = 3, strides = 3)], 9 × [Conv1D(128, 3, 1) + MP1D(pool_size = 3)], and the output layer. The base (3) of the model name indicates the kernel size and stride of the layers while the exponent (9) means the number of Conv1D + MP1D modules. The first layer learns 128 1-D kernels, with which the layer can extract certain 1-D patterns at each time step. The activation of the first convolutional layer is based on size (time step, channels), and we can understand it as a 2-D time–frequency representation where each frequency component is not necessarily a pure sinusoid and the frequency axis is not sorted. Afterward, those basic nonsinusoid components are combined with convolutional layers. The effective operation in the subsequent convolutional layer is equivalent to that of 1-D CNNs.

The following three properties of sample-level CNNs, all of which are related to the extra flexibility of the model, may contribute to their strong performance. First, one of the motivations underlying sample-level CNNs is to learn "phase-invariant" representations. The time-domain kernels involve learning all the possible time shifts within the kernel window. Therefore, a large kernel may require even more filters to cover the variations. The deep stack of the short kernels and max-pooling layers in sample-level CNNs effectively takes care of the phase variation. Second, by learning kernels that are directly applied to the audio signal, sample-level CNNs improve the spectral bandwidth assigned for the input signal analysis. Finally, as previously mentioned, the kernels in the first convolutional layer of sample-level CNNs can be chosen to represent harmonic components rather than pure sinusoids, which form usual 2-D time–frequency representations, such as the spectrogram. This flexibility also improves the discriminative power of the learned features.

A downside of sample-level CNNs is their computation complexity. The authors of [15] informally reported that it took about three to seven times longer to train sample-level CNN models as compared with 1-D CNN models. A way to accelerate the training is to down-sample the waveform input [16], but researchers need to develop more efficient models.

Advanced models

This section summarizes several advanced methods that have addressed various aspects of deep learning-based models. We note that these methods are designed to achieve different goals and that they are not mutually exclusive but can be combined in a model.

Convolutional recurrent neural networks

A convolutional recurrent neural network (CRNN) is a variant of the CNN structure that uses recurrent layers to replace the final convolutional layers [17]. The CRNN model assumes that the long-term patterns are better encoded with recurrent layers than with convolutional layers. This is probably because the important patterns are shorter than the input duration. Therefore, the temporal dynamics of the patterns is a sequence of some short-term patterns rather than a whole, single pattern. The use of recurrent layers also makes the model flexible with respect to the input length, which can be useful for music classification. The network structure in [17] is based on 2-D CNNs, but we note that the recurrent layers can be added to other types of CNNs as well.

Residual networks and squeeze-and-excitation networks

These network architectures have achieved state-of-the-art performance on ImageNet challenges in 2015 and 2017, respectively [18], [19]. Unlike the usual network structures, some layers in a residual network share skip connections, with which the layers are directly connected without any operation. Researchers have enthusiastically adopted this idea because it enables very deep networks (e.g., with more than 100 layers) to be trained. The squeeze-and-excitation network, by applying a trained channel-wise weighting, provides another way to enhance the representation of a layer. It was successfully applied for music autotagging in [20].

Pairwise data

Finally, a more macroscopic modification of a network can be done with a different supervised learning scheme. When the label consists of pairwise similarities or ranking, it is possible to achieve metric learning by using a triplet loss function. A network using this function takes three data samples: an anchor, a positive item, and a negative item. The network learns respective representations, or embeddings, in a way that the embeddings of the anchor and the positive item are close to each other while those of the anchor and the negative item are not. In MIR, music content embeddings were used to predict music similarity in [21].

Data sets and tasks

In this section, we describe four public data sets that have been widely used for music classification and tagging. One of the crucial elements in the success of deep learning is the availability of large-scale public data sets that are used not only for the training of deep-learning models but also for benchmark evaluation. The MIR community has organized an annual algorithm evaluation exchange called *Music Information Retrieval Evaluation eXchange* (*MIREX*) which includes several music classification and tagging tasks; see http://www.music-ir.org/mirex/wiki/MIREX_HOME for more information. However, the development of deep learning has not benefited much from this exchange chiefly because the MIREX data sets are not open to the public and both the training and testing are conducted by the MIREX committee. Also, the volumes of the hidden data sets are not sufficient to fully evaluate the deep models. Presumably, this may be attributed to the serious copyright issues related to music content because the commercial music is released through professional sound producers and the license is more restricted. The four public data sets presented below circumvent the issue by using trimmed or degraded audio clips, e.g., 30 s with 16- or 22.05-kHz sample rate, or copyright-free music tracks. We note that this is not a comprehensive list of available data sets but a selection

Table 1. Selected data sets for music classification.

Data Sets	Number of Clips	Number of Artists	Main Task	Annotation	Audio	Year
GTZAN [10]	1,000	~300	Genre classification	Author's labeling	Yes	2002
MTAT [22]	25,863	230	Autotagging	Crowdsourced	Yes	2009
MSD [7]	1 million	44,745	Autotagging	Crowdsourced	No	2011
FMA [23]	106,574	16,341	Genre classification	Artist's labeling	Yes	2017

of those that have been used mainly to evaluate deep-learning models (Table 1).

GTZAN

Despite its small size, GTZAN (its name derived from the name of George Tzanetakis, who assembled the data set) is one of the most widely used data sets for music genre classification [10]. It contains 1,000 pieces of 30-s audio clips (ten genres and 100 songs for each genre). The up-to-date version uses an artist-stratified split of 443, 197, and 290 audio clips for training, validation, and testing, respectively, with no repeated artists across these sets. The artist-stratified split is unique in the music domain because artists are likely to have similar styles of music across their own songs. We note that GTZAN has also been used for conducting a target task with a small volume of data in the context of transfer learning [16], [24].

MagnaTagATune

MagnaTagATune (MTAT) is one of the most widely used benchmark data sets for music autotagging. It is a multilabel music classification task that annotates genre, mood, instruments, and other song descriptions heterogeneously [22]. The data set comes with tags and similarity annotations. The autotagging benchmark has been conducted using a different number of tags, including 188 tags (the original version), 160 tags (the MIREX 2009 version), and the most frequently used 50 tags. The 50-tag version is currently the most benchmarked. From the 16 predefined partitions of the data set, a common practice is to use the first 12 for training, the 13th for validation, and the remaining three for testing. This data set contains 25,863 30-s audio clips. Its midsize volume is appropriate for training a deep neural network. However, the data set has drawbacks. For example, some clips are cut from the same song, and the music styles are slightly different from popular chart music, as the music tracks are mainly obtained from independent musicians.

MSD

The MSD is a cluster of complementary data sets created from contributions by the MIR community [7]. The original MSD contains artist-level metadata along with the Echo Nest (hand-engineered) audio features without access to the original audio. However, the MSD has been augmented by other metadata by matching the identification data (IDs), including the song-level tags, similarity, lyrics, cover songs, user listening history, and genre labels. To train deep neural networks that take spectrograms or waveforms, researchers have used 30-s preview audio clips downloaded from 7digital (https://www.7digital.com/). Also, the Last.fm tag annotations (https://www.last.fm/) have been widely used in benchmarking for music autotagging.

Free Music Archive

The Free Music Archive (FMA) is the most recently published large-scale data set under the Creative Commons license [23] (http://freemusicarchive.org/). The data set provides the rich track-level, album-level, and artist-level metadata, including the genres, the number of listens, and tags. It is mainly oriented for genre classification, and there are four subsets for benchmarking: small, medium, large, and full. The small and medium subsets are for single-label genre classification, whereas the large and full subsets are for multilabel genre classification. Although the main task is genre classification, tag annotations are also included in the metadata.

Evaluation

As mentioned previously, we set up the problems as either a multiclass (e.g., genre or mood classification) or a multilabel (e.g., autotagging) task. In the multiclass task, the models are primarily evaluated using the accuracy score. In the multilabel task, the predictions are regarded as independent binary outputs. Each of these outputs is evaluated in both annotation and retrieval (or ranking) contexts. The main metrics for annotation are precision, recall, and F score. They are computed for each word label and are averaged. The metrics for retrieval include the area under the receiver–operator curve (AUC), the mean average precision, and the precision at (or up to) rank K (P@K). Among them, AUC has been primarily used to compare different deep-learning models. Table 2 lists the performance comparison reported so far, showing how the AUC obtained for MTAT and MSD has improved over the years due to a cumulative effort from the MIR community.

We note that each of the metrics has slightly different characteristics. For example, the P@K metric is important when we develop recommendation services because users tend to be interested in the top K-ranked results rather than all of them. Therefore, depending on the target application, one may choose different performance metrics when evaluating the results.

Practical guide

This section describes several practical issues when applying a deep-learning model for music classification and tagging tasks.

Table 2. A selection of results for music auto-tagging task. The AUC metric is used for the evaluation.

Models	Published Year	End to End	MTAT	MSD
1-D CNN [13]	2014	No	0.8815	–
		Yes	0.8487	–
Multiscale CNN [25]	2016	No	0.8960	–
2-D CNN [14]	2016	No	0.8940	0.8510
Multi-D CNN [26]	2017	No	0.8930	–
CRNN [17]	2017	No	–	0.8620
Sample-level CNN [15]	2017	Yes	0.9055	0.8812
ReSE-2-multisample CNN [20]	2018	Yes	0.9113	0.8847

Data preprocessing

The first parameter to check is the sample rate. While 44.1 kHz is the standard for commercial music tracks, most data sets are down-sampled to 16 or 22.05 kHz. Researchers often reduce the sample rate even further (e.g., 8 or 12 kHz) as they observe that by reducing data, training is quicker without significantly affecting performance [14], [16]. The 1-D and 2-D CNNs take spectrograms as audio input. In particular, the mel-spectrogram or other log-frequency spectrograms are commonly used. This requires selecting short-term analysis parameters (e.g., window function, hop size), a mel-band size, and the log compression strength. Librosa (https://librosa.github.io/librosa/) is a widely used audio-processing library for this purpose. In contrast, sample-level CNNs do not require any preprocessing other than sample-rate conversion. The waveform input is already zero-centered, and the amplitude of commercial music clips is normalized well by postprocessing (e.g., audio mastering).

Data augmentation

Data augmentation is a technique that regularizes the model by increasing the volume of data. For audio signals, the digital audio effects, such as pitch-shifting or time-stretching, are effective means to this end. However, this should be done within a range where the nature of music labels is not distorted. Data augmentation is not found in the music classification literature yet, but it may be useful when the data set of the target task is small. Musical Data Augmentation (http://muda.readthedocs.io/en/latest/) is a useful audio-processing library for this purpose.

Input length

Semantic labels are usually annotated to each song at a track level, and the audio length is typically several minutes. Therefore, to use audio tracks to train the models, they need to be chopped into a fixed length of segments. This causes a tradeoff between model complexity and label noisiness. If the segment is shortened, a more compact model can be trained with greater input data. However, the labels inherited from the track level tend to be noisier due to the dynamic nature of music within a track, and the compact model can miss learning high-level musical features. On the other hand, if the segment is lengthened, the label noisiness will be mitigated and a more long-term structure can be learned. However, this requires having more complex models along with more data. The common practice is using segments between 3 and 6 s as input. Some complex models take up to 30 s [14], [17]. The assumption that the segments of a track share the same labels as the track has also been referred to as a *weakly supervised learning problem* [25], as when the segment-level supervision is noisy. An advanced method to deal with such an issue might be adding the so-called attention module to the neural network, as demonstrated by [27] for music mood classification.

Applications

In this section, we explain how a neural network model pretrained on larger-scale labeled data for music classification and tagging can be applied to other tasks, such as classifying across data sets, making recommendations, thumbnailing music, and predicting hit songs, as illustrated in Figure 4.

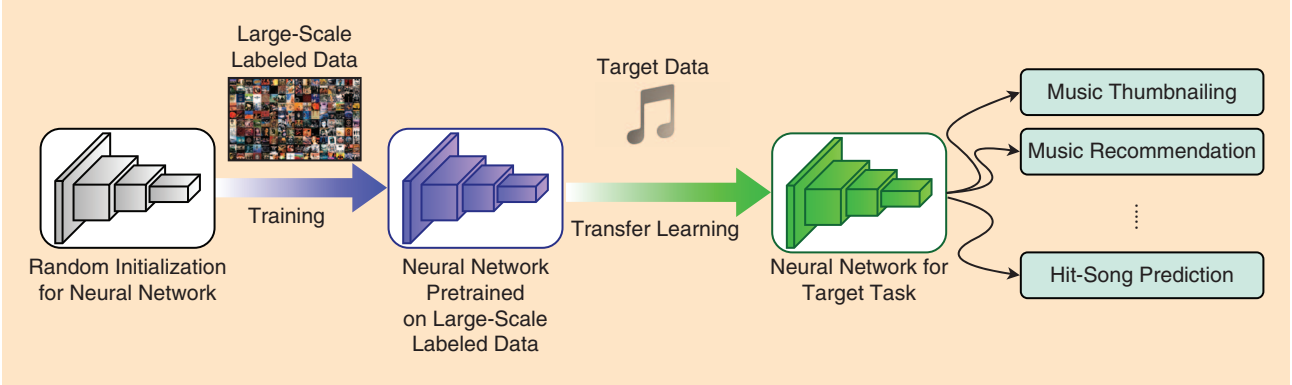


FIGURE 4. Transferring the knowledge of a neural network pretrained on larger-scale labeled data to other music applications. Refer to the "Applications" section for details.

Findings show that pretrained models using large-scale labeled data can provide a good estimate of the similarities among audio content (Figure 5). Therefore, with the so-called transfer-learning techniques, we can build classifiers for problems with sparsely labeled data on top of such pretrained models. For example, Choi et al. [24] used approximately 250,000 MSD preview clips to train a 2-D CNN to classify 50 music tags. They then showed that a concatenated feature vector using the activations of the feature maps of the CNN can serve as a nice general-purpose music representation. This is useful for a variety of other tasks, such as classifying ballroom dancing and other subgenres, predicting the emotions the music might stimulate, distinguishing between vocal and nonvocal sounds, and sorting various sound events, such as car horns and dog barks.

Pretrained models can also contribute to addressing the challenge of making content-based music recommendations. For instance, Pandora, powered by the Music Genome Project, can create various personalized playlists for each of its users by combining traditional collaborating filtering algorithms with the classified attributes of music [34]. Compared to the purely collaborating filtering methods, adding content filtering by means of pretrained models for music classification and tagging helps ensure acoustic consistency (e.g., similarities in genre/style, rhythmic patterns, vocal timbre, or expressed emotions) in the recommended list of music, which in turn improves the user experience. With content filtering, the (acoustic) diversity of the recommended music can also be controlled [2].

An interesting application of pretrained models is music thumbnailing, i.e., detecting the highlight of a song. Huang et al. [27] employed a pretrained model for music-mood classification and an attention module to learn to weigh the contribution of a song's different segments in deciding the overall mood of the song. Then, a moving window was used to aggregate the per-segment attention scores over time to pick the song's peak, assuming that the highlight is usually the most emotional part of a song. They achieved a promising result in highlight detection without using any labeled data related to music highlights.

Another interesting application is audio-based hit-song prediction. Yu et al. [28] used a pretrained 1-D CNN model for music classification as part of a bigger CNN model for predicting song popularity. The experiments showed that deep structures are indeed more accurate than shallow structures in predicting song popularity and that the use of the pretrained music classification model further improves the accuracy by a large margin. We believe that similarly pretrained models can also be applied to other problems.

Limitations and future challenges

In this section, we discuss some major limitations of the existing methods for music classification and tagging, and we outline some directions for future research.

Share of audio data

The problem of copyright infringement may limit widespread research on music classification and tagging. People cannot freely distribute the audio files from the data sets. Common approaches for getting around this issue include sharing precomputed features instead of the audio files, providing a list of IDs with which people may find the audio previews on the web, or using copyright-free music. The last approach provides more options as people can get the audio files for the entire songs. However, to work with popular music that people are familiar with (which are usually copyright protected), some other solutions are still needed. A possible approach is to automatically generate music that is similar to the popular music by using deep-generative models, such as generative adversarial networks [30].

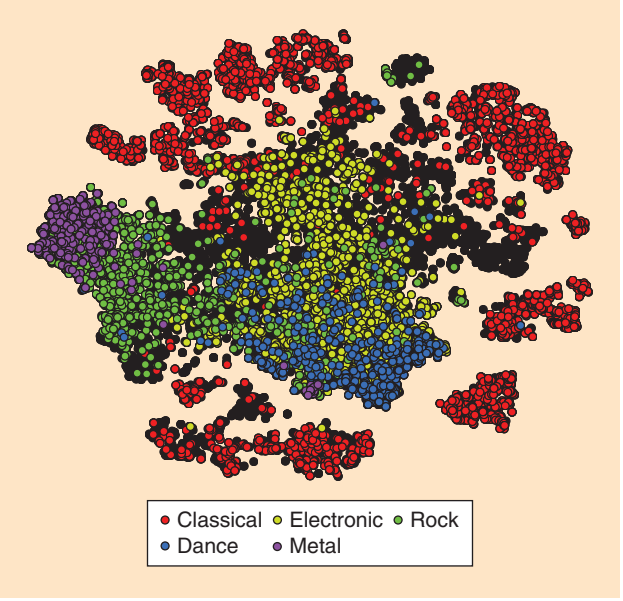


FIGURE 5. We generate T-distributed stochastic neighbor embedding visualization (best viewed in color) of the distribution of song embeddings by using a CNN model [29] trained on MTAT. The colors represent five different tags of music (provided by MTAT). The black dots denote songs that are not labeled by any of these tags. We see that songs with similar genres cluster together in the learned feature space.

Musically meaningful network design

We also expect developers to make more use of peculiar characteristics of music in the design of deep neural networks. In the past few years, deep learning-based approaches to many MIR problems have established new state-of-the-art benchmarks. However, to explain the network and for better performance, future work is needed to bring back music-domain knowledge to the loop of network design. For instance, instead of expecting that the network can learn abstract representations of music in different hierarchies from the bottom up on its own, it might be better to inform the network (in a top-down fashion) of the midlevel features, such as the presence of syncopation, the extensive use of diminished chords, and the use of synthesizer. Then, classifiers could be built on top of these midlevel features. This requires a joint effort from the research community to put together resources and labeled data to model different layers of music knowledge and to conduct experiments to

find out the best ways to use them in a neural network, with all the layers possibly trained in an end-to-end fashion.

Another way to incorporate music knowledge is to use not only the audio files but also the corresponding musical scores, if available. Musical scores contain rich information about the music piece, such as the melody line and the chord sequence. Score-informed approaches have been shown to greatly improve the performance of source separation [31]. By aligning a score with the audio recording of its actual performance, we can also extract performance-related features (e.g., stylistic changes in velocity, note duration, and the use of different playing techniques) that characterize how the performer interprets the piece of music. However, to date, little work has been done to use the audio and musical score jointly in a neural network. Future work can build, for example, a two-stream network that takes as input the audio file as well as its score or other symbolic representations.

We know that a music piece is usually composed of several elements, such as melody, chords, percussion, and baseline, and each of them is often played by different instruments [29]. However, most neural networks for music classification process audio inputs as a whole without distinguishing among the component sources of sound. While deep-learning approaches have led to the state-of-the-art results in sound-source separation and music classification, little work has been done to jointly tackle the two problems under a unified network. Requiring the neural network to learn to separate the musical sources that compose an audio mixture while performing feature learning can, therefore, be an important future direction.

Vocabulary and personalization

The diversity and coverage of labels considered in classification and tagging models can also be increased. Ideally, it is better to have a granular vocabulary as fine as that of Pandora's Music Genome Project [1], which claims to have around 450 musical attributes. One possible solution is to leverage the abundant user-provided tags from social platforms, such as last.fm, Twitter, or SoundCloud. However, how to get rid of the social tags' noises and ambiguity while learning an effective music classifier remains an open issue. In imagining extreme possibilities, we foresee technologies that would enable end users to use arbitrary natural language as input (e.g., via voice commands) to query for music. For example, "Hey Google, I need music to make me feel better" and "Alexa, I cannot fall asleep. Maybe some music?" Such scenarios may be important given the ever-increasing popularity of artificial intelligence speakers. To support such retrieval applications, we need to collaborate with researchers from the speech community to better understand natural language. The vocabulary considered by our machines in describing music also has to be expanded and adapted to cope with the richness of natural language.

Moreover, the associations between music and some types of labels such as moods (e.g., "happy," "aggressive," "sad," "relaxing") and usages (e.g., "for exercising," "for reading") are known to be subjective. Therefore, it is more difficult to computationally model them. However, such labels are important, for example, if we want to automatically create playlists that fit a user's mood or activity. We surmise that the assignment of such labels has to be personalized, taking into account the listener's preference as well as the "personal definition" of those labels to the listener. Although much research has been done for music mood classification, it remains to date a challenge to effectively personalize such systems.

Cross-modality approach

We see a lot of cross-modality research in the neighboring field of computer vision, which aims to combine the visual world with the textual world. Notable applications include image captioning, conditional image generation from visual attributes, and cross-modality retrieval. We expect similar attempts to flourish in the MIR community as well, not only for classification and tagging tasks (e.g., [33]) but also for generative tasks, such as tag-conditioned music generation, melody-conditioned lyrics generation, album cover generation, and music video generation. To facilitate research on these tasks, sharing pretrained models or knowledge (e.g., best practices in model training) can play an important role.

Music is important in our daily lives and there are many ways machine learning can improve or change the way we experience and create music. By summarizing what has been known thus far, we hope this article can encourage follow-up research to further enhance our modeling and understanding of music.

Authors

Juhan Nam (juhannam@kaist.ac.kr) received his B.S. degree in electrical engineering from Seoul National University, South Korea, in 1998 and his Ph.D. degree in music from Stanford University, California, in 2013 studying at the Center for Computer Research in Music and Acoustics. He is an assistant professor at the Graduate School of Culture Technology at the Korea Advanced Institute of Science and Technology (KAIST), South Korea. Before joining KAIST, he was a staff research engineer at Qualcomm, San Diego, California, from 2012 to 2014. He was also a software/digital signal processing engineer at Young Chang (Kurzweil), South Korea, from 2001 to 2006. He is interested in various topics at the intersection of music, audio signal processing, and machine learning. He is a Member of the IEEE.

Keunwoo Choi (keunwoo.choi@qmul.ac.uk) received his B.S. degree in electric engineering and studied applied acoustics for his M.S. degree at Seoul National University, South Korea, in 2009 and 2011, respectively. He worked as a researcher at the Electronic and Telecommunications Research Institute, South Korea. He received his Ph.D. degree in 2018 from the Centre for Digital Music, School of Electronic Engineering and Computer Science, Queen Mary University of London, United Kingdom. In 2017, he received the Best Paper Award at the 18th International Society of Music Information Retrieval Conference. He is currently with Spotify Inc., New York. His research interests include music information retrieval and machine learning.

Jongpil Lee (richter@kaist.ac.kr) received his B.S. degree in electrical engineering from Hanyang University, South Korea, in 2015. He received his M.S. degree in 2017 from the Graduate School of Culture Technology at the Korea Advanced Institute of Science and Technology, South Korea, and is currently pursuing his Ph.D. degree from the same institution. From July to September 2017, he was an intern at Naver Clova Artificial Intelligence Research. His current research interests include machine learning and signal processing applied to audio and music applications.

Szu-Yu Chou (fearofchou@citi.sinica.edu.tw) received his bachelor's degree from National Formosa University, Huwei, Taiwan, in 2010. He is currently working toward his Ph.D. degree at National Taiwan University, Taipei. He is a research assistant in the Research Center for Information Technology Innovation at Academia Sinica. His research interests include music recommendation and music information retrieval. In 2015, he received the Best Paper Award at the IEEE International Conference on Multimedia and Expo.

Yi-Hsuan Yang (yang@citi.sinica.edu.tw) received his bachelor's and Ph.D. degrees from National Taiwan University, Taipei, in 2006 and 2010, respectively. He is an associate research fellow at the Research Center for Information Technology Innovation, Academia Sinica. He is also a joint-appointment associate professor with National Cheng Kung University, Taiwan. His research interests include music information retrieval, affective computing, multimedia, and machine learning. He is an author of the book *Music Emotion Recognition*. He has been an associate editor of *IEEE Transactions on Affective Computing* and *IEEE Transactions on Multimedia* since 2016. He is a Senior Member of the IEEE.

References

[1] M. Prockup, A. F. Ehmann, F. Gouyon, E. M. Schmidt, Ò. Celma, and Y. E. Kim, "Modeling genre with the Music Genome Project: Comparing human-labeled attributes and audio features," in *Proc. Int. Society for Music Information Retrieval Conf.*, Málaga, Spain, 2015, pp. 31–37.

[2] S.-Y. Chou, L.-C. Yang, Y.-H. Yang, and J.-S. Jang, "Conditional preference nets for user and item cold start problems in music recommendation," in *Proc. IEEE Int. Conf. Multimedia and Expo*, Hong Kong, 2017, pp. 1147–1152.

[3] Z. Fu, G. Lu, K. M. Ting, and D. Zhang, "A survey of audio-based music classification and annotation," *IEEE Trans. Multimedia*, vol. 13, no. 2, pp. 303–319, Apr. 2011.

[4] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Mag.*, vol. 29, no. 6, pp. 82–97, 2012.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Advances in Neural Information Processing Systems*, Lake Tahoe, NV, 2012, pp. 1097–1105.

[6] K. Choi, G. Fazekas, K. Cho, and M. Sandler. (2017). A tutorial on deep learning for music information retrieval. arXiv. [Online]. Available: https://arxiv.org/abs/1709.04396

[7] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, "The Million Song Dataset," in *Proc. Int. Society Music Information Retrieval Conf.*, Miami, FL, 2011, pp. 591–596.

[8] J. Pons, O. Nieto, M. Prockup, E. M. Schmidt, A. F. Ehmann, and X. Serra, "End-to-end learning for music audio tagging at scale," in *Proc. Machine Learning Audio Signal Processing Workshop, Advances Neural Information Processing Systems*, Tokyo, Japan, 2017.

[9] E. J. Humphrey, J. P. Bello, and Y. LeCun, "Feature learning and deep architectures: New directions for music informatics," *J. Intell. Inform. Syst.*, vol. 41, no. 3, pp. 461–481, 2013.

[10] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Trans. Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002.

[11] J. Nam, J. Herrera, M. Slaney, and J. O. Smith, "Learning sparse feature representations for music annotation and retrieval," in *Proc. Int. Society for Music Information Retrieval Conf.*, Porto, Portugal, 2012, pp. 565–570.

[12] P. Hamel and D. Eck, "Learning features from music audio with deep belief networks," in *Proc. Int. Society for Music Information Retrieval Conf.*, Utrecht, The Netherlands, 2010, pp. 339–344.

[13] S. Dieleman and B. Schrauwen, "End-to-end learning for music audio," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Florence, Italy, 2014, pp. 6964–6968.

[14] K. Choi, G. Fazekas, and M. Sandler, "Automatic tagging using deep convolutional neural networks," in *Proc. Int. Society for Music Information Retrieval Conf.*, New York, NY, 2016, pp. 805–811.

[15] J. Lee, J. Park, K. L. Kim, and J. Nam, "Sample-level deep convolutional neural networks for music auto-tagging using raw waveforms," in *Proc. Sound and Music Computing Conf.*, Espoo, Finland, 2017, pp. 220–226.

[16] J. Lee, J. Park, K. L. Kim, and J. Nam, "SampleCNN: End-to-end deep convolutional neural networks using very small filters for music classification," *Appl. Sci.*, vol. 8, no. 1, p. 150, 2018.

[17] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Convolutional recurrent neural networks for music classification," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, New Orleans, LA, 2017, pp. 2392–2396.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Honolulu, HI, 2016, pp. 770–778.

[19] J. Hu, L. Shen, and G. Sun. (2017) Squeeze-and-excitation networks. arXiv. [Online]. Available: https://arxiv.org/abs/1709.01507

[20] T. Kim, J. Lee, and J. Nam, "Sample-level CNN architectures for music auto-tagging using raw waveforms," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Aalborg, Denmark, 2018, pp. 366–370.

[21] R. Lu, K. Wu, Z. Duan, and C. Zhang, "Deep ranking: Triplet matchnet for music metric learning," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, New Orleans, LA, 2017, pp. 121–125.

[22] E. Law, K. West, M. I. Mandel, M. Bay, and J. S. Downie, "Evaluation of algorithms using games: The case of music tagging," in *Proc. Int. Society for Music Information Retrieval Conf.*, Kobe, Japan, 2009, pp. 387–392.

[23] M. Defferrard, K. Benzi, P. Vandergheynst, and X. Bresson, "FMA: A dataset for music analysis," in *Proc. Int. Society for Music Information Retrieval Conf.*, Suzhou, China, 2017, pp. 316–323.

[24] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Transfer learning for music classification and regression tasks," in *Proc. Int. Society for Music Information Retrieval Conf.*, Suzhou, China, 2017, pp. 141–149.

[25] J.-Y. Liu and Y.-H. Yang, "Event localization in music auto-tagging," in *Proc. ACM on Multimedia Conf.*, Amsterdam, The Netherlands, 2016, pp. 1048–1057.

[26] J. Pons, O. Slizovskaia, R. Gong, E. Gómez, and X. Serra, "Timbre analysis of music audio signals with convolutional neural networks," in *Proc. European Signal Processing Conf.*, Kos, Greece, 2017, pp. 2744–2748.

[27] Y.-S. Huang, S.-Y. Chou, and Y.-H. Yang, "Music thumbnailing via neural attention modeling of music emotion," in *Proc. Asia Pacific Signal and Information Processing Assoc. Annu. Summit and Conf.*, Kuala Lumpur, Malaysia, 2017, pp. 347–350.

[28] L.-C. Yu, Y.-H. Yang, Y.-N. Hung, and Y.-A. Chen. (2017) Hit song prediction for pop music by Siamese CNN with ranking loss. arXiv. [Online]. Available: https://arxiv.org/abs/1710.10814

[29] S.-Y. Chou, J.-S. R. Jang, and Y.-H. Yang, "Learning to recognize transient sound events using attentional supervision," in *Proc. Int. Joint Conf. Artificial Intelligence*, Stockholm, Sweden, 2018, pp. 3336–3342.

[30] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning.* Cambridge, MA: MIT Press, 2016.

[31] S. Ewert, B. Pardo, M. Mueller, and M. D. Plumbley, "Score-informed source separation for musical audio recordings: An overview," *IEEE Signal Process. Mag.*, vol. 31, no. 3, pp. 116–124, May 2014.

[32] J. Bosch, J. Janer, F. Fuhrmann, and P. Herrera, "A comparison of sound segregation techniques for predominant instrument recognition in musical audio signals," in *Proc. Int. Society Music Information Retrieval Conf.*, Porto, Portugal, 2012, pp. 559–564.

[33] S. Oramas, O. Nieto, F. Barbieri, and X. Serra, "Multi-label music genre classification from audio, text and images using deep features," in *Proc. Int. Society for Music Information Retrieval Conf.*, Suzhou, China, 2017, pp. 23–30.

[34] S. Perez. (2018, Mar.). Pandora takes on Spotify with dozens of personalized playlists built using its Music Genome. TechCrunch. [Online]. Available: https://techcrunch.com/2018/03/28/pandora-takes-on-spotify-with-dozens-of-personalized-playlists-built-using-its-music-genome/

SP

Meinard Müller, Andreas Arzt, Stefan Balke,
Matthias Dorfer, and Gerhard Widmer

Cross-Modal Music Retrieval and Applications

An overview of key methodologies

©ISTOCKPHOTO.COM/TRAFFIC_ANALYZER

Digital Object Identifier 10.1109/MSP.2018.2868887
Date of publication: 24 December 2018

There has been a rapid growth of digitally available music data, including audio recordings, digitized images of sheet music, album covers and liner notes, and video clips. This huge amount of data calls for retrieval strategies that allow users to explore large music collections in a convenient way. More precisely, there is a need for cross-modal retrieval algorithms that, given a query in one modality (e.g., a short audio excerpt), find corresponding information and entities in other modalities (e.g., the name of the piece and the sheet music). This goes beyond exact audio identification and subsequent retrieval of metainformation as performed by commercial applications like Shazam [1].

In this article, we review several cross-modal retrieval scenarios, with a particular focus on sheet music (visual domain) and audio (acoustic domain). First, we discuss a traditional approach where the sheet music and audio representations are converted into common midlevel feature representations that capture musical properties related to pitches and harmony. The resulting feature sequences can then be compared using standard alignment algorithms [2], [3].

Second, we review an approach based on symbolic fingerprinting techniques. Originally, audio fingerprinting referred to a procedure that allows for a robust identification of exact replicas of audio recordings [4]. In our cross-modal scenario, however, we discuss tempo- and transposition-invariant symbolic fingerprinting methods based on note parameters extracted via audio transcription techniques [5], [6].

Third, employing deep-learning methods, we describe an end-to-end, cross-modal retrieval strategy that works without the need for manually crafted feature representations [7]. Given snippets of sheet music (in the form of pixel images) and corresponding audio excerpts (in spectrograms), a neural network learns a joint embedding space on which cross-modal retrieval can be performed using simple distance measures and nearest-neighbor search.

Using these three approaches as illustrative examples, the primary objective of this article is to discuss the principles and challenges encountered in general music processing, such as designing musically motivated features and similarity measures to cope with semantic data variability. Furthermore, to

illustrate the potential of cross-modal retrieval techniques, we describe some navigation and browsing applications, including a prototype system called the *Piano Music Companion*, while indicating future research directions.

Music representations

Before we delve into the various cross-modal retrieval approaches, we first introduce some basic notions, following [3, Ch. 1]. As indicated by Figure 1, music can be represented in many different ways and formats. For example, a composer may write a composition in the form of a musical score, where musical symbols are used to visually encode which notes are to be played and how. The printed form of a musical score is also referred to as *sheet music*. The original medium of this representation is paper, although it is now also accessible on computer screens in the form of digital images.

In electronic instruments and computers, music can be communicated by means of standard protocols—such as the widely used Musical Instrument Digital Interface protocol (https://www.midi.org/)—where event messages specify note pitches, note intensities (velocities), and other parameters to generate the intended sounds. Often, the term *symbolic* is used to refer to any data format that explicitly represents musical entities. The musical entities may range from timed note events, as is the case in MIDI files, to graphical shapes with attached musical meaning, as in music engraving systems. In contrast to such symbolic representations, the musical events are not given explicitly in audio representations, such as WAV or MP3 files. The latter formats encode acoustic waves that are generated when, e.g., playing an instrument and travel from the sound source to the human ear as air-pressure oscillations.

At this point, it is important to note that each of these representations reflects certain aspects of a musical entity but that no single representation encompasses all of the properties. For example, rather than giving strict specifications, a musical score serves only as a guide for performing a piece of music, leaving room for different interpretations. Reading the instructions in the score, a musician shapes the music by varying the tempo, dynamics, articulation, and other parameters, thus creating a personal interpretation of the piece. Furthermore, while sheet music visually encodes the musical notes, such information is hidden in an audio recording, which is basically a time series of samples. In summary, even if various formats refer to the same piece of music, there may be a significant gap—technically as well as semantically—between different representations, such as sheet music and audio.

The boundaries between the diverse music representations are not sharp. As illustrated by Figure 1, symbolic representations—depending on their specific format and intended application—may be closer to sheet music or audio representations. For example, symbolic representations such as MusicXML (https://www.musicxml.com/) are used for rendering sheet music, where the shape of the note objects and their arrangement on a page are determined. Optical music recognition (OMR) can be seen as the inverse process, with the objective to transform sheet music into a symbolic representation.

Furthermore, symbolic representations such as MIDI are used for synthesizing audio, where the note objects are transformed into musical tones and real sounds. The inverse process is known as *automatic music transcription* (*AMT*) and aims at extracting note events, key signature, time signature, instrumentation, and other score parameters from a given music recording [3]. Both transformations, OMR and AMT, are far from straightforward. For example, correctly recognizing and interpreting the meaning of all of the musical symbols in complex sheet music is easy for a trained human but hard for a computer. Even though current OMR software is reported to yield highly accurate results, manual postprocessing is necessary to obtain a high-quality symbolic representation [8]. Similarly, converting a music recording into a note-based representation is a largely unsolved problem—in particular, for multivoiced music involving different instruments [9].

For relating different types of data (e.g., sheet music and audio data) to each other, traditional methods are often based

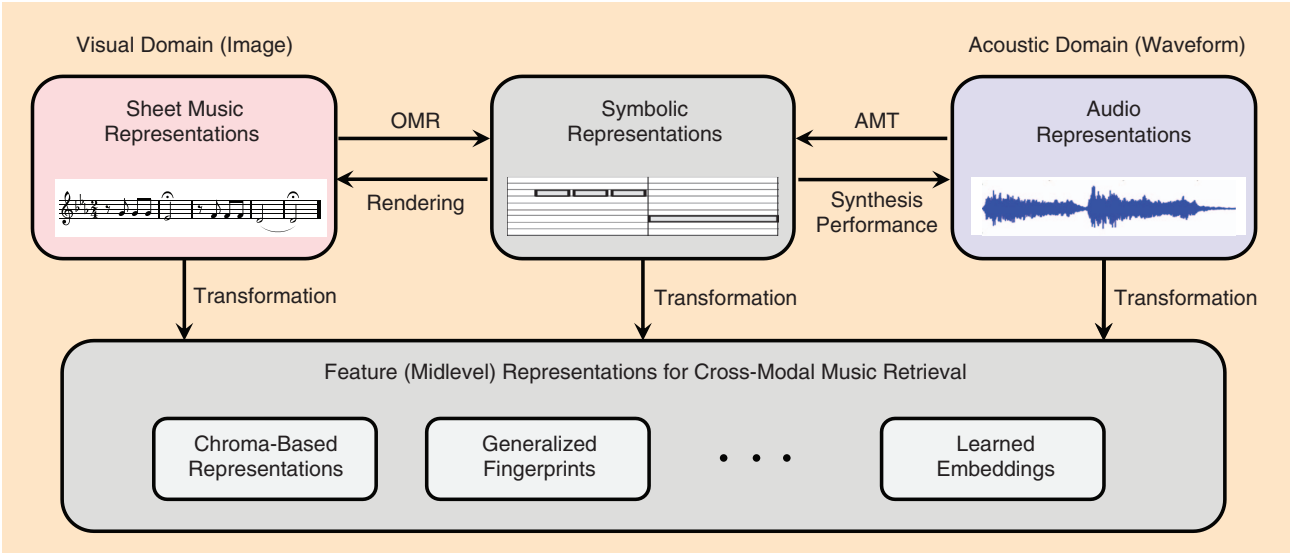


FIGURE 1. The different representations for music data and data transformations relevant for cross-modal music retrieval.

on midlevel representations that exploit specific domain knowledge. As an important example, we first consider midlevel representations that capture musical properties related to pitches and harmony. We then discuss symbolic fingerprints that are based on note-level descriptors. Both of these approaches require expert knowledge in the transformation process. As an alternative, we present an end-to-end learning approach based on deep neural networks, where the idea is to circumvent the explicit definition of a midlevel representation. In the following sections, we address the benefits and limitations of these conceptually different approaches in the context of cross-modal music retrieval.

Chroma-based approach

To make music data algorithmically accessible, traditional music processing tries to extract suitable features that capture relevant key aspects while suppressing irrelevant details. For music-related retrieval and analysis tasks, chroma features have turned out to be a powerful midlevel representation [3], [10].

Because of their central importance in music processing, we give a short introduction to the basics of chroma features, following [3, Ch. 1]. Recall that playing a note on an instrument results in a more or less periodic sound of a certain fundamental frequency. This frequency is closely related to what is called the *pitch* of a note. This notion allows us to order pitched sounds from lower to higher—similar to the keys of a piano keyboard ordered from left to right.

Two notes with fundamental frequencies in a ratio equal to any power of two (e.g., half, twice, or four times) are perceived as very similar or musically/harmonically equivalent, in some sense. This observation leads to the fundamental notion of an octave, which is defined as the interval between one musical note and another with half or double its fundamental frequency. In Western music, the space within one octave is generally subdivided into 12 scale steps with fundamental frequencies equally spaced on a logarithmic frequency axis, resulting in what is known as the *12-tone, equal-tempered scale*. In this scale, each pitch can be separated into two components, which are referred to as the *tone height* (or *octave number*) and the *chroma* (or *pitch spelling attribute*, denoted by $C, C^{\sharp}, D, \ldots, B$ in Western music notation).

Chroma features rely on this perception of octave equivalence and map absolute pitch into 12 octave-independent pitch classes, where a pitch class consists of all of the pitches that share the same chroma. Thus, a chroma feature is represented by a 12-dimensional vector $x = (x(1), \ldots, x(12))^{\top}$, where $x(1)$ corresponds to chroma C, $x(2)$ to $C^{\sharp}$, and so on. In the feature extraction step, a given audio signal is converted into a sequence of chroma vectors, also called *chromagrams*, where each vector expresses how the short-time energy of the signal is spread over the 12 chroma bands. A chromagram closely correlates to the melodic and harmonic progression of the music, while exhibiting a high degree of robustness to variations in instrumentation and dynamics.

There are many ways to compute chroma-based features from audio recordings, e.g., by using short-time Fourier transforms (STFTs) in combination with binning strategies [10] or employing suitable multirate filter banks [11]. Furthermore, the properties of chroma features can be significantly changed by introducing suitable pre- and postprocessing steps modifying spectral, temporal, and dynamical aspects. As an example, Figure 2(b) shows two different chromagram variants extracted from a piano audio recording. While the first is a traditional chromagram, the second version is enhanced, such that certain important frequencies that relate to melody notes, as specified by the upper staff, are emphasized—which can be important, e.g., for melody-based retrieval. When given a symbolic music representation, such as MIDI or MusicXML files, it is straightforward to derive chromagrams from the explicitly encoded note parameters (pitches, note onsets, and note durations).

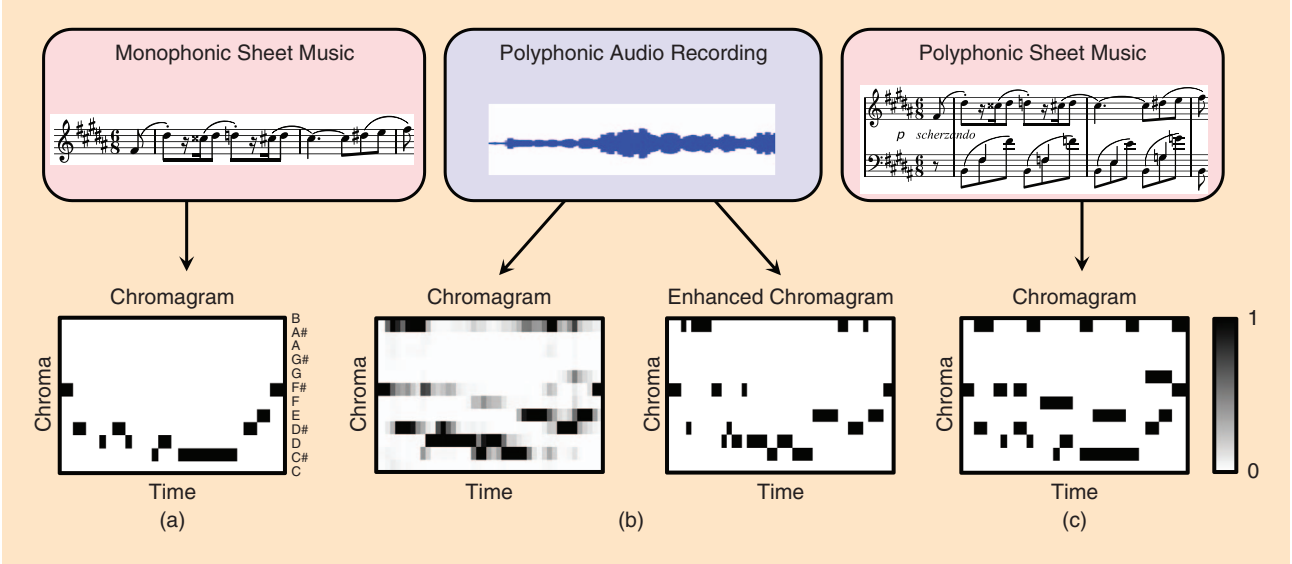


FIGURE 2. Some chromagrams obtained from (a) monophonic sheet music, (b) polyphonic audio representations, and (c) polyphonic sheet music for the beginning of Frédéric Chopin's Nocturne in B Major, op. 9, no. 3.

Figure 2 also shows a symbolic chromagram obtained from a monophonic [Figure 2(a)] and a polyphonic [Figure 2(c)] sheet music representation. While symbolic chromagrams are based on pure note information, audio-based chromagrams tend to be noisy, reflecting the full range of the signal's acoustic properties, including partials, transients, and room acoustics. Still, as also demonstrated by Figure 2, chroma features mainly capture melodic and harmonic properties and are suited to serve as a midlevel feature representation for comparing and relating acoustical and symbolic music.

To demonstrate the applicability and potential of chroma-based features, we consider a cross-modal retrieval scenario motivated by Barlow and Morgenstern's book *A Dictionary of Musical Themes*, published in 1949 [12]. This book contains about 10,000 musical themes of well-known instrumental pieces from the corpus of Western classical music. These monophonic themes (usually four bars long) are typically the most memorable parts of a piece of music. This motivates the retrieval scenario as considered in [13] and [14], where the objective is to retrieve all audio recordings from a music collection that contain a specified musical theme. More formally, let $\mathcal{Q}$ be the collection of musical themes where each element $Q \in \mathcal{Q}$ is regarded as a query. Furthermore, let $\mathcal{D}$ be a set of audio recordings, which we regard as a database collection consisting of documents $D \in \mathcal{D}$. Given a query $Q \in \mathcal{Q}$, the retrieval task is to identify the semantically corresponding documents $D \in \mathcal{D}$.

One approach, as illustrated by Figure 3(a), is to first transform a query Q (possibly using OMR as an intermediate step) and each of the documents D into chromagrams. Based on these midlevel representations, one computes a matching function Δ_D^Q by locally comparing the query chromagram to the audio chromagram using a subsequence variant of dynamic time warping (DTW) [11, Ch. 4]. For each position of the audio recording D, such a matching function indicates the local cost of aligning the query chromagram with a segment ending at that position of the audio chromagram. In other words, each local minimum of Δ_D^Q that is close to the value zero points to a location where the query (musical theme) is similar to a local segment of the document (audio recording). Thus, for a given query, the retrieval task can be solved by computing matching curves for all documents and screening for local minima that are below a certain threshold in these curves. The costs of the local minima yield a natural ranking of the retrieved documents and their relevant sections, which can then be presented in the form of a ranked list [Figure 3(b)].

As detailed in [13] and [14], there are various challenges that need to be addressed, including tempo deviations, OMR extraction errors, musical tunings, key transpositions, and differences in the degree of polyphony between the symbolic query and the audio recordings. For some of these challenges, there already exist reliable compensation strategies. For example, key transpositions are simulated by a cyclic shift of the query's chromagram, or local and global tempo deviations are compensated for by using sequence alignment techniques, such as DTW. Handling differences in the degree of polyphony is still subject to ongoing research. One strategy to bridge the polyphony gap is to first extract the predominant melody of the audio recording using harmonic summation [15] and source-filter models [16]. From the resulting salience representations, enhanced audio chromagrams that better match the monophonic theme may be derived [see Figure 2(b) for an illustration].

Obviously, computing matching curves for each database document results in a retrieval procedure that does not scale to large music collections. Indexing techniques based on short audio excerpts (so-called audio shingles) can help speed up the retrieval procedure [17], [18]. In the next section, we discuss an alternative approach that is based on symbolic fingerprints and permits extremely efficient retrieval.

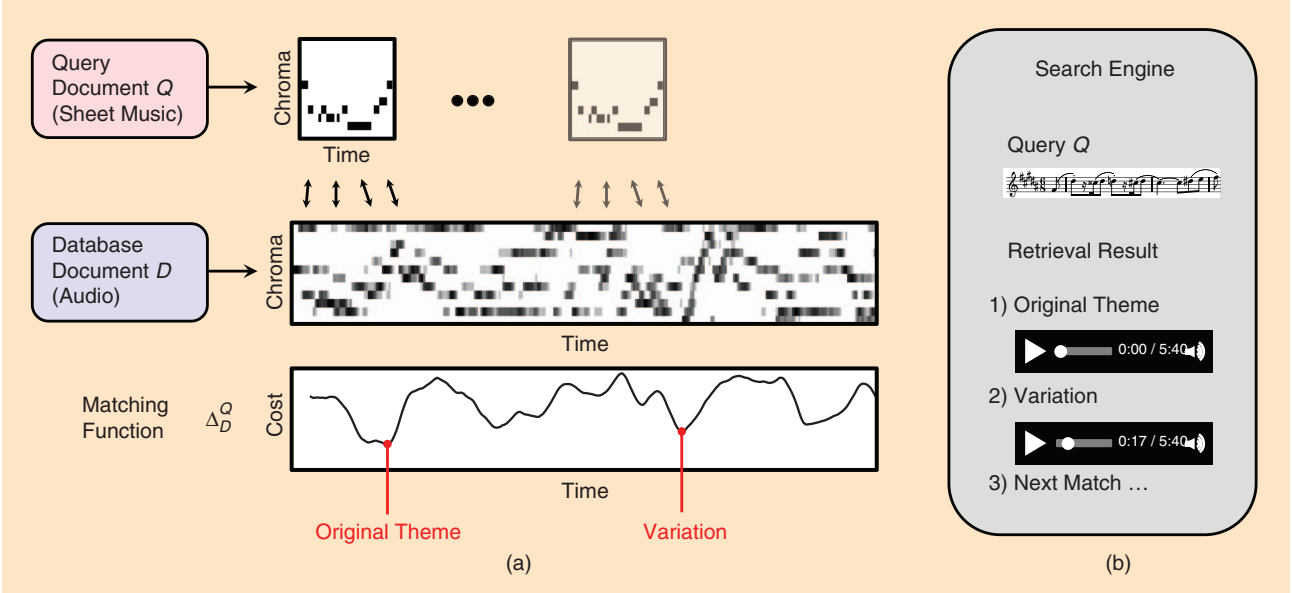


FIGURE 3. (a) An illustration of the matching procedure with chroma-based representations. (b) The costs of the local minima yield a natural ranking of the retrieved documents and their relevant sections, which are shown in the form of a ranked list.

Symbolic fingerprinting approach

We have seen that chroma features are a very convenient mid-level representation for comparing music data of different modalities. One main benefit is that both symbolic and audio data can be easily converted into chromagrams. Furthermore, capturing only the coarse harmonic/melodic progression, chromagrams are highly robust to musical and acoustic variations. However, the reduction onto the chroma level also leads to a loss of valuable information that may be contained in the input data, such as accurate timing and pitch parameters, as encoded by sheet music. As a consequence, chroma-based retrieval strategies often become problematic for short input sequences (e.g., covering only a couple of notes). Furthermore, reducing pitch information to the 12 chroma bands renders the comparison of monophonic and polyphonic versions difficult. An alternative to using chroma-based features is to exploit the high specificity of note parameters and of the resulting time–pitch patterns of occurring notes. To this end, both the visual and acoustic data need to be transformed into the symbolic music domain. In the following, we discuss such an approach, based on symbolic fingerprints, and highlight the resulting benefits and limitations.

Traditionally, in music processing, *audio fingerprinting* refers to methods for identifying exact replicas of audio recordings, which are possibly distorted in some way (e.g., compression artifacts or background noise). For this problem, also known as *audio identification*, powerful algorithms exist and are in everyday use in commercial applications (see, e.g., [1], [3], [4], and [19]). In the identification process, the audio material is compared by means of so-called audio fingerprints, which are compact and discriminative audio features. There are many different ways of designing and computing audio fingerprints, and the suitability of a specific type of fingerprint very much depends on the requirements imposed by the intended application. For example, in the pioneering work by Wang [1], a fingerprinting approach is described that operates on spectral peaks extracted from a time–frequency representation.

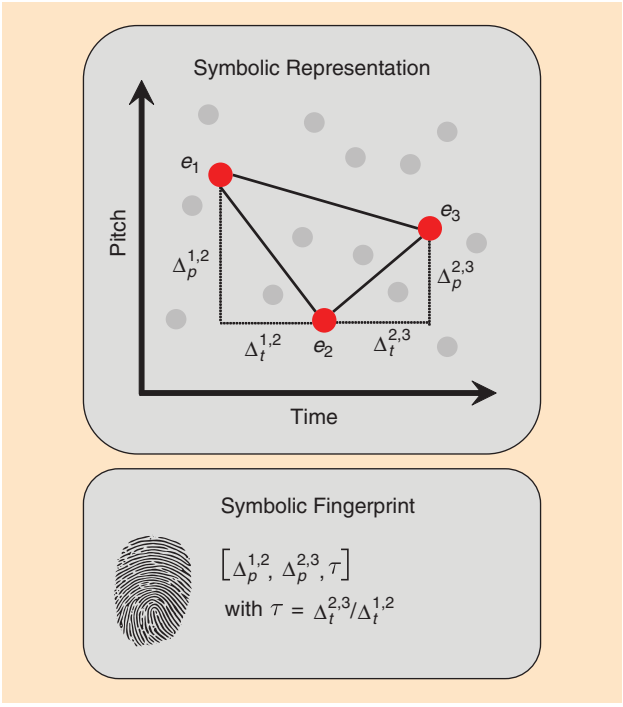


FIGURE 4. An illustration of symbolic fingerprints.

Recent work, such as [19] and [20], has focused on making fingerprinting algorithms more robust to transformation in the time scale (the playback speed of the audio) and the frequency scale (transpositions). Classical fingerprinting approaches, combined with indexing techniques, allow for an identification of audio material that is extremely efficient (scalable to huge fingerprint data sets) and effective (with high precision even for short queries). However, being based on audio-specific spectro-temporal patterns, these techniques are not suited for handling music-specific variations, as required for cross-modal music retrieval or related tasks, such as cover song retrieval [3], [21].

Inspired by classical fingerprinting techniques, Arzt et al. [5], [6] introduced a symbolic fingerprinting approach that allows not only for the identification of exact replicas of recordings but also for fast retrieval of different versions of the same piece of music, including differently performed audio recordings and score representations. In the following, we summarize the main idea of this approach. We start with a symbolic music representation where all note events are encoded explicitly. As illustrated by Figure 4, we assume that each note event $e = (t, p)$ is specified by an onset time t and a pitch p. To obtain fingerprints, we consider triples consisting of three events, $e_1 = (t_1, p_1)$, $e_2 = (t_2, p_2)$, and $e_3 = (t_3, p_3)$, with $t_1 < t_2 < t_3$. For each such triple, we define the time differences $\Delta_t^{1,2} := t_2 - t_1$ and $\Delta_t^{2,3} := t_3 - t_2$ as well as the pitch differences $\Delta_p^{1,2} := p_2 - p_1$ and $\Delta_p^{2,3} := p_3 - p_2$. Furthermore, we set $\tau := \Delta_t^{2,3} \big/ \Delta_t^{1,2}$. Finally, a symbolic fingerprint is defined to be a list of the following numbers:

$$[\Delta_p^{1,2}, \Delta_p^{2,3}, \tau]. \tag{1}$$

Considering time and pitch relations in a relative fashion, each fingerprint is invariant with regard to musical transpositions (pitch shifts) and tempo changes. To obtain local descriptors, fingerprints are computed only from note events within a certain neighborhood, typically a few seconds. This not only facilitates short query lengths, but also reduces the number of fingerprints to be stored in the database. Also, observe that, since each individual fingerprint encodes relative timing information, we need to assume that the onset times of a triple are distinct. As a result, simultaneous note events (as occurring in a chord) may not be encoded by a single fingerprint. However, such cooccurring events can be captured by considering several fingerprints. In summary, being discriminative yet compact descriptors of fixed length, such fingerprints have turned out to be suitable for indexing symbolically encoded music data.

We now discuss how the symbolic fingerprints can be used for cross-modal music retrieval. As a challenging example scenario, we consider a combined sheet music identification and score-following application tailored to piano music [6]. Given a short excerpt of an audio recording (used as a query), the task is to identify the underlying sheet music document as well as the exact score position (see Figure 5). Accordingly, the database

$\mathcal{D}$ for this task consists of sheet music representations of all of the pieces to be potentially identified. In a preprocessing step, all sheet music documents $D \in \mathcal{D}$ are first transformed into a suitable symbolic format (e.g., by applying OMR or by extracting note parameters from a MusicXML file). From this encoding, symbolic fingerprints are extracted for each document by considering all of the possible triples of note events that obey certain constraints. For example, to avoid a combinatorial explosion, one typically imposes constraints in the form of minimum and maximum values for the time differences $\Delta_t^{1,2}$ and $\Delta_t^{2,3}$. The resulting fingerprints, along with links to suitable metadata (e.g., corresponding piece and sheet music positions), are stored in a fingerprint database that is equipped with efficient search structures based on indexing techniques.

Similarly, an incoming audio query is also transformed into a set of symbolic fingerprints. This, however, involves a nontrivial transcription step to convert the recording into a symbolic representation. In general, automatic music transcription is still an unsolved problem—in particular, for polyphonic music recordings with many different instruments (e.g., orchestral music) (see [9], [22], and [23]). In the case of single-instrument polyphonic music (such as piano music), state-of-the-art algorithms provide reasonable, albeit far from perfect, transcriptions. In our scenario, we employ a recent transcription algorithm based on a recurrent neural network [22]. The use of bidirectional hidden layers enables the system to better model the context of the notes, which exhibit a very characteristic envelope during their decay phase, particularly for piano music. The network was trained on a collection of several hundred piano pieces recorded on various pianos, virtual and real (see [22] for further details).

To make the transcriber also applicable in online scenarios, instead of preprocessing the whole piece of audio at one time, the signal is split into blocks that consist of several subsequent frames centered around the current frame. Using such blocks, each covering roughly 210 ms of audio, is a tradeoff between maintaining the system's ability to model the context of the notes and keeping the introduced delay to a minimum. The network outputs a transcription of the audio query consisting of a list of note onsets and pitches, which can be further transformed into a set of audio fingerprints. Finally, the score fingerprint database is searched for subsets that approximately fit the query's set of audio fingerprints. The best matching subset in the database yields the sheet music document, along with the score position (Figure 5).

In contrast to chroma-based midlevel representations, symbolic fingerprints are compact, possess a high discriminative power, and are well suited for indexing techniques. As a result, these techniques scale well to large amounts of data in terms of memory requirements, accuracy, and efficiency. However, there is also a price to be paid. The necessary transcription from audio signals into the symbolic domain is a hard problem that is solvable well enough only for certain classes of music (e.g., piano music recorded under reasonable acoustic conditions). Even though a small proportion of the fingerprints extracted from the query may suffice to identify the correct piece, symbolic fingerprinting may fail if the input representation becomes too noisy.

For general music recordings, including those with many instruments (e.g., an orchestra), there is still a long way to go. Here, one requires strategies that better adapt to the multitude of musical aspects, including harmony, melody, rhythm, dynamics, and instrumentation. In this context, recent advances in deep

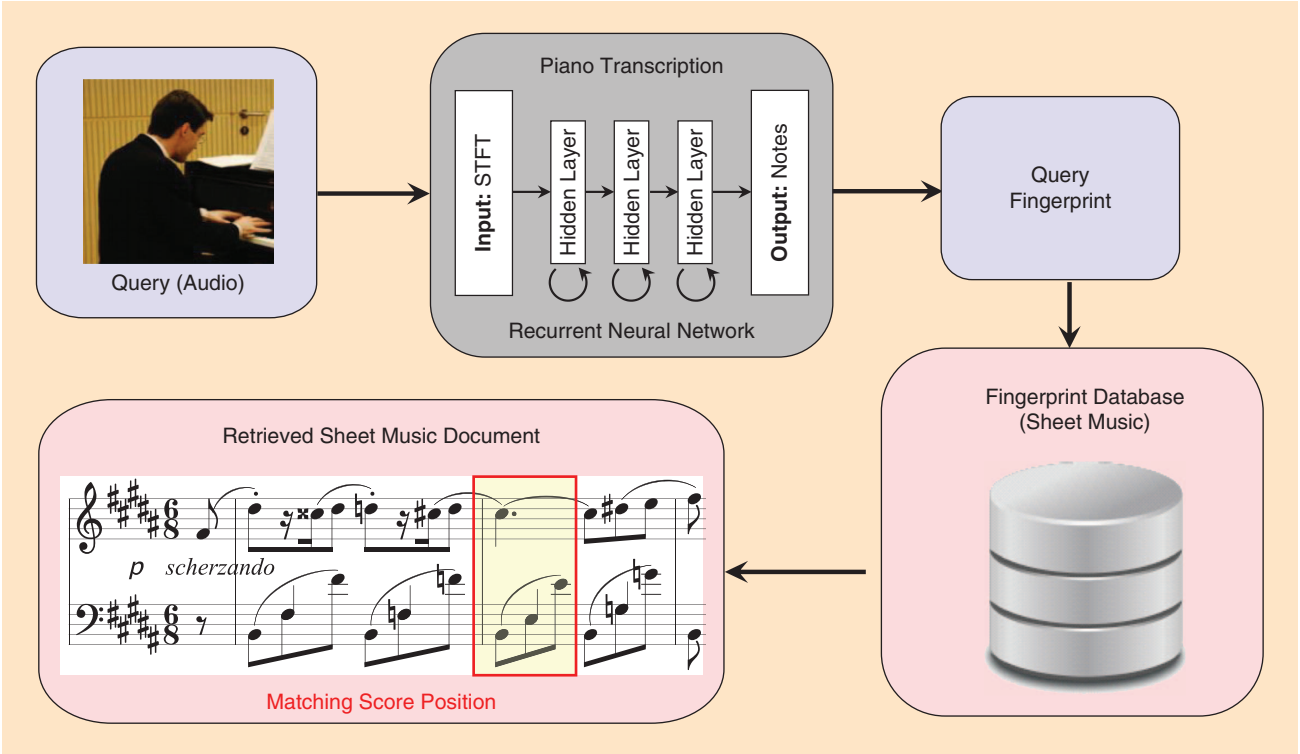


FIGURE 5. An illustration of cross-modal retrieval via piano transcription and symbolic fingerprinting. (Photo of Werner Goebl courtesy of Clemens Chmelar.)

learning may help to achieve further progress in this area. In the following section, we discuss such a deep-learning approach that tries to learn sheet music and audio correspondences directly from raw input representations, without the need for midlevel representations that explicitly exploit musical knowledge.

Deep-learning approach

In the previous sections, we have seen two more-traditional approaches for linking audio and sheet music data using musically informed midlevel representations—one using chroma features and one symbolic fingerprints. Such representations not only require expert knowledge at the design stage but are also problematic when relying on error-prone preprocessing steps, such as automatic music transcription on the audio side or optical music recognition on the sheet music side. As an alternative, we now present a methodology to directly learn correspondences between audio data and sheet music images from a set of training observations, thus circumventing the explicit definition of a midlevel representation.

This approach builds on the current success of artificial neural networks, today often referred to as *deep learning*, which have proven to be powerful tools for automatic feature learning [24]. Given snippets of sheet music images and corresponding audio excerpts, we introduce a cross-modal neural network that learns an embedding space in which both modalities are represented as low-dimensional vectors [7]. In this embedding space, cross-modal music retrieval can then be easily performed by using a simple similarity measure.

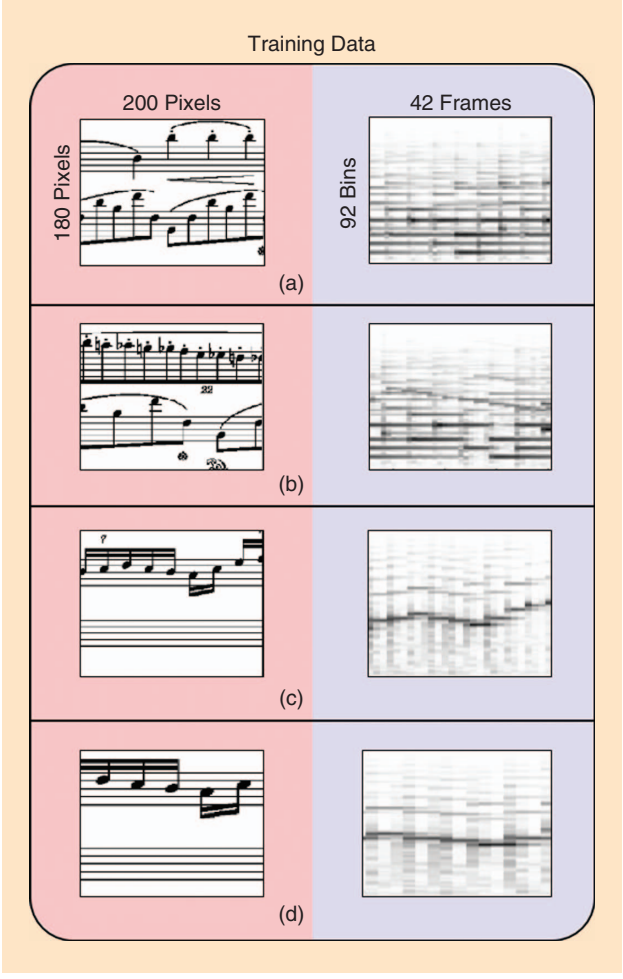


FIGURE 6. (a)–(d) Four training pairs, each consisting of a sheet music snippet and an audio excerpt. The pair in (d) was obtained from the pair in (c) by applying data augmentation techniques.

The general principle of supervised feature learning is to learn latent representations in an end-to-end fashion from a set of raw training observations. Such approaches are not only generally applicable but also have the advantage of automatically adapting the learned representations to the given problem. One limitation, however, is that supervised learning requires a sufficiently large set of training data to arrive at models that generalize well to unseen data.

In our scenario, we need training pairs that consist of sheet music snippets and corresponding audio excerpts. Typical examples as used in our system are shown in Figure 6(a)–(d). Note that, for creating such training pairs, we need to first establish correspondences between individual pixel locations of the note heads in a score and their respective counterparts (note onset events) in the corresponding audio recording. Establishing the correspondences can be done either in a manual annotation process or by relying on synthetic training data generated from digital sheet music formats, such as Musescore (https://musescore.com/) or Lilypond (http://lilypond.org/). Based on these relationships, one can generate corresponding snippets of sheet music images (in our case, 180×200 pixels) and short excerpts of audio (in our case, represented by log-frequency spectrograms with 92 bins $\times$ 42 frames). These are the pairs presented to the multimodal network for training.

To improve the generalization ability of the resulting network, one can further apply data augmentation techniques to synthetically increase the effective size of the training set and better account for relevant data variability. In this setting, different transformations for sheet music augmentation (e.g., image scaling and translation) and audio augmentation (e.g., using different sound fonts and tempo scaling) are applied. At this point, we emphasize that data augmentation is a crucial component for learning cross-modal representations that generalize to unseen music, especially when limited data are available.

In this process, augmenting the data set using data transformations is conceptually different from and more promising than automatically generating random scores. First, rendering (synthesizing) sheet music typically results in images with strong regularities (e.g., the same scale or perfectly centered staff lines). By applying image transformations, these regularities are disturbed, thus making the embedding networks robust to small distortions, like those that occur in realistic scenarios, e.g., images of printed sheet music scanned under different conditions and sheet music originating from different publishers using varying type settings. Second, note that music (and, hence, also sheet music) follows musical rules. Therefore, augmentation by adding randomly generated music may distort the inherent data distribution of realistic music, which may have a negative impact on embedding space learning.

Based on such training pairs, the retrieval task is formulated as an embedding problem, with the aim of learning a joint

embedding space of the two different modalities [7]. This approach is inspired by a similar text-to-image retrieval problem, where a pairwise ranking loss is introduced as an optimization target [25]. In the following, let $(\mathbf{x}, \mathbf{y})$ denote a training pair consisting of a sheet image snippet $\mathbf{x}$ and an audio excerpt $\mathbf{y}$. As shown in Figure 7, the network consists of two separate pathways. One processes $\mathbf{x}$ and is represented by the function f_α, where α is the network parameters to be trained. The other pathway, which is represented by the function g_β, with parameters β, is responsible for $\mathbf{y}$. The two functions map $\mathbf{x}$ and $\mathbf{y}$, respectively, to a k-dimensional vector, where $k \in \mathbb{N}$ denotes the embedding dimension.

To define the loss function, we need a scoring function $s : \mathbb{R}^k \times \mathbb{R}^k \to \mathbb{R}$ to measure similarity in the embedding space. In our scenario, s is chosen to be the cosine measure, i.e., the cosine of the angle between two vectors. Furthermore, for each given training pair $(\mathbf{x}, \mathbf{y})$, we assume that there are $L \in \mathbb{N}$ additional contrasting examples $\mathbf{y}_\ell$ for $\ell \in \{1, 2, \ldots L\}$. Then, the pairwise ranking loss (also known as the *max-margin hinge loss* [25]) is defined as follows:

$$\mathcal{L}_{\text{rank}} = \sum_{(\mathbf{x},\mathbf{y})} \sum_{\ell=1}^{L} \max\{0, \gamma - s(f_\alpha(\mathbf{x}), g_\beta(\mathbf{y})) + s(f_\alpha(\mathbf{x}), g_\beta(\mathbf{y}_\ell))\}. \tag{2}$$

In this formula, the first sum is taken over a set of training pairs $(\mathbf{x}, \mathbf{y})$ called a *training batch*, where each such pair comes with a separate set of contrasting examples (in practice, all remaining audio samples of the current training batch). The purpose of this loss function is to encourage an embedding where the distance between matching samples $(\mathbf{x}, \mathbf{y})$ is lower than the distance between mismatching samples $(\mathbf{x}, \mathbf{y}_\ell)$. The parameter $\gamma \in \mathbb{R}_+$ is the margin parameter of the hinge loss and, in combination with the maximum function, imposes a penalty on poorly embedded training pairs. More precisely, if the elements of a matching pair $(\mathbf{x}, \mathbf{y})$ are already close in the learned embedding space and, in addition, the elements of the mismatching pairs $(\mathbf{x}, \mathbf{y}_\ell)$ are embedded far enough apart, the second term in the max operator goes below zero, and the respective pairs do not contribute to the overall loss. On the contrary, if the embedded elements of a matching pair are still far apart, the second term is usually above zero and will yield a substantial contribution to the overall loss.

In the training stage, the pairwise ranking loss in (2) is minimized via stochastic gradient descent with respect to the network parameters α and β. Once the networks represented by the functions f_α and f_β are learned, the elements of the matching pairs are close in the embedding space, while those of contrasting pairs are far apart (in the ideal case). For further details concerning the network topology and the training procedure, we refer to [7] and [26].

Given this learned embedding space, cross-modal retrieval can be performed based on a retrieval-by-embedding paradigm (Figure 7). It is important to note that, although the network pathways are trained simultaneously on pairs of sheet music snippets and audio excerpts, both modalities are required only at training time. At application time, the two network pathways operate independently of each other.

This has huge benefits in view of the cross-modal retrieval applications discussed in the previous sections. For example, in sheet music identification and score-following applications, one can first compute an embedding of an entire collection of sheet music snippets using the image embedding function f_α. The resulting k-dimensional embedding vectors can be further processed

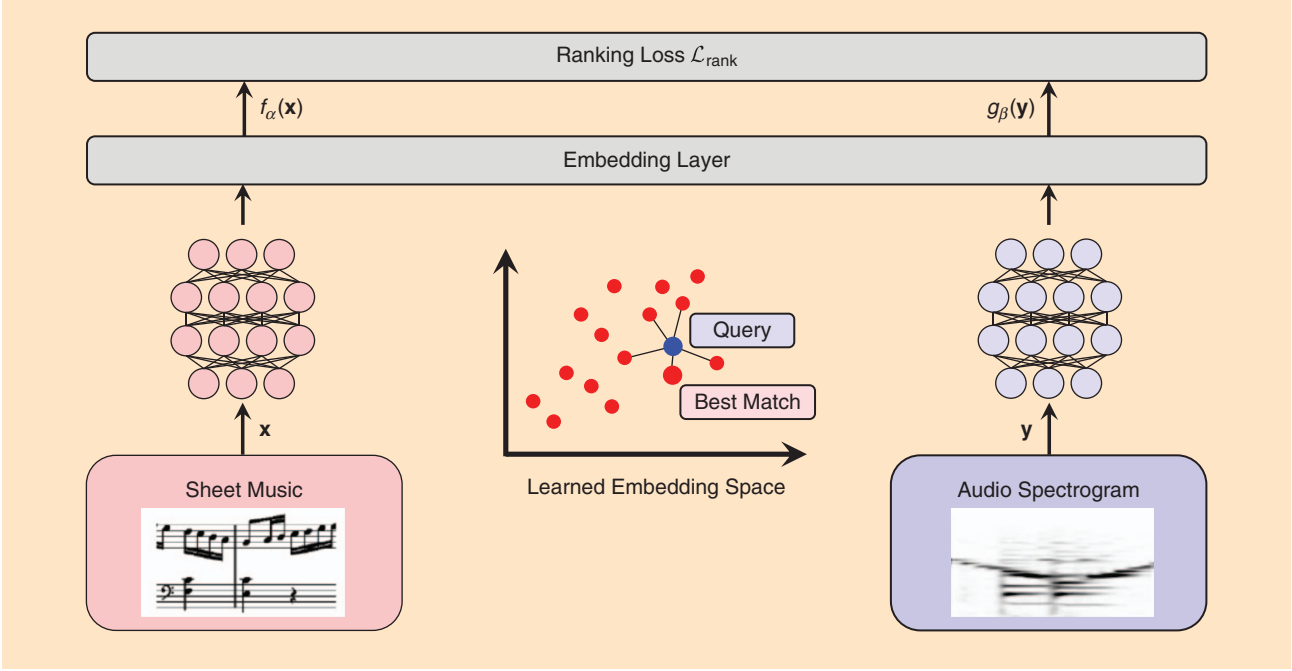


FIGURE 7. An illustration of the network used for learning a cross-modal embedding space. At application time, the learned functions f_α and g_β are used to project the sheet music snippets and audio excerpts, respectively, to the joint embedding space.

and stored using suitable index structures that allow for an efficient neighborhood search. Then, given an audio excerpt as a query, the search can be performed by first projecting the query into the joint embedding space using the audio embedding function g_β of the network and then performing a nearest-neighbor search.

The experiments reported in [7], which are based on 26 classical piano pieces (including the composers Bach, Haydn, Beethoven, and Chopin) and roughly 20,000 training pairs, demonstrate that the end-to-end learning approach yields reasonable retrieval results for sheet music of medium complexity (e.g., piano scores) and synthesized audio (used for evaluation to establish the ground truth). In particular, combining retrieval based on snippets/excerpts with a subsequent majority voting step, the approach is capable of correctly relating sheet music and audio recordings on the piece level with high accuracy. However, on the level of sheet music snippets (consisting of one or two bars) and audio excerpts (lasting a couple of seconds), the proposed system is not yet competitive with engineered approaches that exploit musical knowledge or are based on symbolic representations (see the approaches presented in the two previous sections).

At this stage, one may conclude that, even when comparatively scant training data are available, it is still possible to use deep-learning models by designing appropriate, task-specific data augmentation strategies. Initial experiments showed that, when trained on only one composer, the model started to generalize to unseen scores by other composers. Therefore, we may expect that the described model will develop its full potential when provided a comprehensive data set that consists of millions of training pairs comprising different editions and layouts of sheet music and different recorded performances.

Applications and future directions

In this article, we have introduced different approaches for cross-modal music retrieval aiming to bridge the gap between various music representations. Despite the remaining challenges, current technology enables a variety of music navigation and browsing applications that have educational and commercial relevance. For example, in the context of modern digital music libraries, cross-modal retrieval strategies have become an important component for content-based analysis, synchronization, indexing, and navigation in heterogeneous music collections [27]. Other cross-modal applications are often subsumed under the umbrella of score following, where the computer listens to a live performance and tries to read along in the sheet music. The output of a score-following algorithm can be used for highlighting the current measure in a digital score, automatic page turning (a page turner being a person who turns sheet music pages for a soloist during a performance), or automatic accompaniment (see, e.g., [28]).

In the following, we describe one specific example of a prototype system to give a concrete impression of what is already possible. The Piano Music Companion is a versatile system focused on piano music, intended to be useful for both pianists and music lovers [29]. The system is able to identify, follow, and synchronize live performances of classical piano music in real time. The Piano Music Companion is a permanent listener. Whenever the pianist starts playing (regardless of which piece or where within the piece), the companion identifies the piece, the position within the score, and continues to follow along. This allows triggering various actions synchronized to the performed music—for instance, the current position in the sheet music is highlighted.

While this is helpful for the performer and listener, further information about important themes, musical structures, and chords can be provided. In a concert setting, the system may also give hints to the listener about what to focus on at specific moments. The system may also give additional background information on the piece or composer, while telling the user where to acquire additional recordings of the current or related pieces.

Technically, the Piano Music Companion is based on two main components that run in parallel. The first is responsible for identifying the piece being played. To this end, symbolic fingerprinting, as described earlier, is used to continuously match the most recently detected notes of the live performance to a database of symbolically encoded sheet music (Figure 5). Currently, the database includes the complete solo piano works by Chopin and the complete Beethoven piano sonatas, and consists of roughly 1 million notes in total (about 330 pieces). Once the piece and the rough position within the sheet music representation have been identified, the actual score following is conducted using a separate chroma-based tracking procedure, which is realized as an online variant of the matching procedure shown in Figure 3. In this way, the system combines the strengths of the respective components. The fingerprinting component is flexible, it works globally across different pieces, and it scales over large data sets. However, since the fingerprinter's transcription step is in general faulty, the component often leads to outliers and local misalignments. This weakness is compensated for by the separate chroma-based tracking component, which is less efficient but introduces a high degree of robustness (due to the chroma features). This second component is applied only locally, for tracking the score once the piece and the rough position are known.

By combining these two components, the Piano Music Companion continuously reevaluates its hypothesis and tries to match the current input stream to the complete database. Thus, even if the musician suddenly jumps to a different score position or starts playing a completely different piece, the system is able to follow as long as the piece is part of the database. The Piano Music Companion is also highly tolerant of deviations from the notated score (due to performance errors, transcription errors, or intentional variations) and to tempo changes. A video demonstration of our system can be found at https://www.youtube.com/watch?v=SUBtND_MJZs.

Our vision is to extend this scenario toward a Complete Classical Music Companion. Such a system would be at one's fingertips anytime and anywhere, possibly as an application on a mobile device. Whatever source of music—be it a live concert, a digital video disc, a video stream, or a radio program—whatever piece of classical music, whatever instrumentation, and whoever the performers, the companion would detect what it is listening to and inform about the music, the historical context of the piece, famous interpretations, and so forth, thus guiding the user in the listening process.

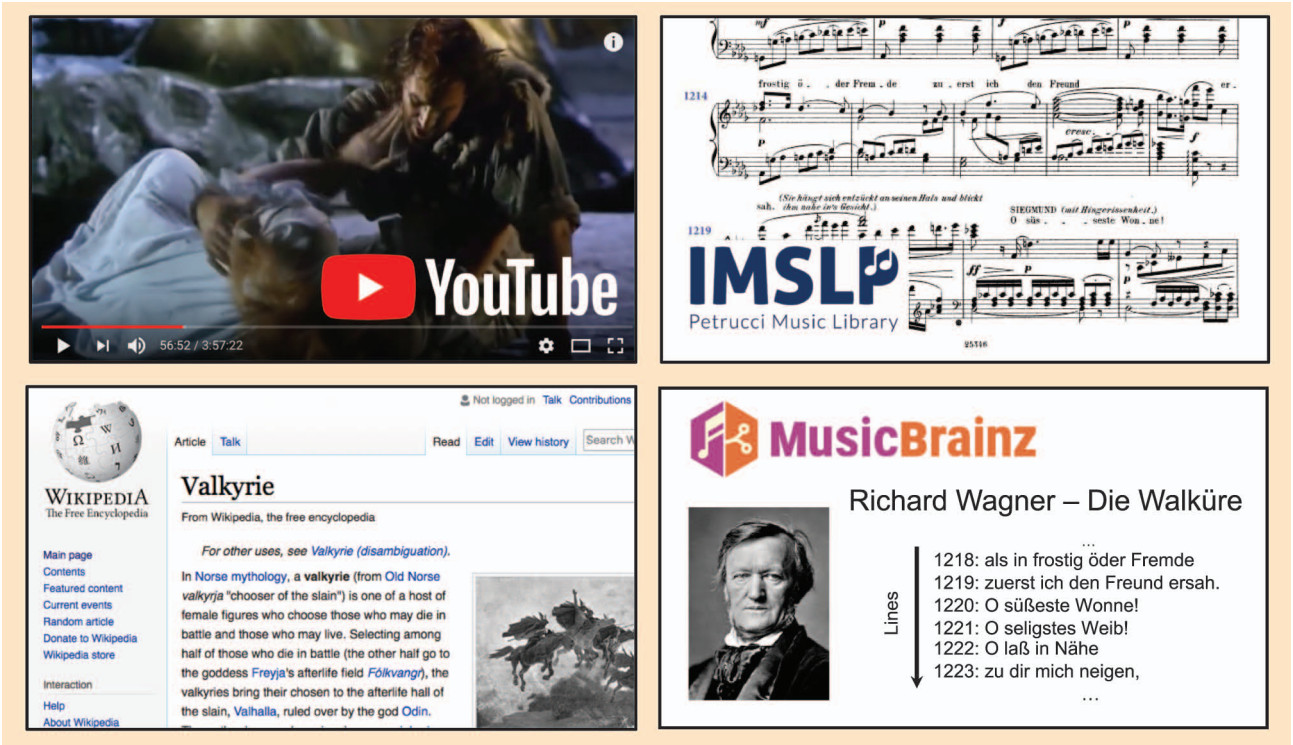


FIGURE 8. Some sources of freely accessible music data distributed via the Internet. (MusicBrainz image courtesy of the Warner Music Group.)

Beyond this specific music companion scenario, cross-modal music processing techniques are essential for organizing and searching information distributed via the Internet (Figure 8). For example, there are millions of digitized pages of sheet music publicly available on sites such as the International Music Score Library Project (IMSLP) Petrucci Music Library (http://imslp.org/). On the audio side, widely accessible music and video platforms, such as YouTube, offer a vast and rapidly growing corpus of music recordings. Furthermore, music-related websites, as available at *Wikipedia*, contain information of various types, including text, score, images, and audio. Finally, community-driven encyclopedias, such as MusicBrainz (https://musicbrainz.org/), collect and provide music-related metadata in a systematic fashion. For example, structured websites can be used to automatically derive text-, score-, and audio-based queries to look for other musically related documents on the Web [13], [30]. Furthermore, YouTube videos may be automatically enriched with manually or automatically generated musical annotations, as recently demonstrated in [31].

This rich application potential, demonstrated in concrete application scenarios, makes cross-modal music retrieval a very active research field that also drives research on other music processing tasks. For instance, one key challenge is to improve transformation techniques, such as OMR and AMT, which are a bottleneck in many of the current approaches. Also, deep neural networks that directly learn to relate different data modalities are a very promising alternative that is currently receiving a lot of attention. We hope that these prospects will serve as an inspiration for the signal processing community to pay even more attention to music as a promising (and beautiful) object of study.

Acknowledgments

The International Audio Laboratories Erlangen is a joint institution of the Friedrich-Alexander-Universität Erlangen-Nürnberg and the Fraunhofer-Institut für Integrierte Schaltungen IIS. Meinard Müller and Stefan Balke are supported by the German Research Foundation (DFG MU 2686/11-1). Matthias Dorfer acknowledges financial support for early versions of this work by the Austrian Ministries Federal Ministry of Transport, Innovation, and Technology and the Federal Ministry of Science, Research, and Economy and the Province of Upper Austria via the Competence Centers for Excellent Technologies, Software Competence Center Hagenberg. The work of Andreas Arzt and Gerhard Widmer was supported by the European Research Council (ERC) under the European Union's Horizon 2020 Framework Programme (H2020, 2014–2020)/ERC Advanced Grant Agreement 670035, project Con Espressione.

Authors

Meinard Müller (meinard.mueller@audiolabs-erlangen.de) received his diploma degree in mathematics and his Ph.D. degree in computer science from the University of Bonn, Germany, in 1997 and 2001, respectively. Since 2012, he has held a professorship in semantic audio signal processing at the International Audio Laboratories Erlangen, Germany. His recent research interests include music processing, music information retrieval, and audio signal processing. He has coauthored more than 100 peer-reviewed scientific papers and is the author of a monograph, *Information Retrieval for Music and Motion* (Springer, 2007), and textbook, *Fundamentals of Music Processing* (Springer, 2015).

Andreas Arzt (andreas.arzt@jku.at) received his B.S. degree in software and information engineering and his M.S. degree in computational intelligence from the Vienna University of Technology, Austria, in 2006 and 2008, respectively. In 2016, he completed his Ph.D. degree on the topic "Flexible and Robust Music Tracking" in the Department of Computational Perception, Johannes Kepler University, Linz, Austria, where he is currently an assistant professor. His research interests include real-time music tracking, music synchronization, and music identification.

Stefan Balke (stefan.balke@audiolabs-erlangen.de) received his diploma in electrical engineering from Leibniz Universität, Hanover, Germany, in 2013. In early 2018, he completed his Ph.D. degree in the Semantic Audio Signal Processing Group headed by Prof. Meinard Müller at the International Audio Laboratories Erlangen, Germany. Afterward, he joined the Institute of Computational Perception led by Gerhard Widmer at Johannes Kepler University, Linz, Austria. His research interests include music information retrieval, machine learning, and multimedia retrieval.

Matthias Dorfer (matthias.dorfer@jku.at) studied medical informatics at Vienna University of Technology, Austria, with a focus on computational image analysis and machine learning. After finishing his master's degree studies, he worked for two years as an industrial researcher in the field of medical image analysis. Since April 2015, he has been a Ph.D. student in the Department of Computational Perception, Johannes Kepler University, Linz, Austria, under the supervision of Prof. Gerhard Widmer. His research interests are artificial neural networks, especially multimodality deep learning, and audiovisual representation learning.

Gerhard Widmer (gerhard.widmer@jku.at) received his Ph.D. degree in computer science in 1989 from Technische Universität Wien, Austria. He is a professor and the head of the Department of Computational Perception, Johannes Kepler University, Linz, Austria, and head of the Intelligent Music Processing and Machine Learning Group, Austrian Research Institute for Artificial Intelligence, Vienna. His research interests include artificial intelligence, machine learning, and intelligent music processing. He is a fellow of the European Association for Artificial Intelligence, has been awarded Austria's highest research awards [the START Prize (1998) and the Wittgenstein Award (2009)], and currently holds a European Research Council Advanced Grant for research on computational models of expressivity in music.

References

[1] A. Wang, "An industrial strength audio search algorithm," in *Proc. Int. Society Music Information Retrieval Conf.*, 2003, pp. 7–13.

[2] F. Kurth, M. Müller, C. Fremerey, Y. Chang, and M. Clausen, "Automated synchronization of scanned sheet music with audio recordings," in *Proc. Int. Conf. Music Information Retrieval*, 2007, pp. 261–266.

[3] M. Müller, *Fundamentals of Music Processing*. Berlin: Springer-Verlag, 2015.

[4] P. Cano, E. Batlle, T. Kalker, and J. Haitsma. (2005). A review of audio fingerprinting. J. VLSI Signal Process. [Online]. 41(3), pp. 271–284. Available: http://dx.doi.org/10.1007/s11265-005-4151-3

[5] A. Arzt, S. Böck, and G. Widmer, "Fast identification of piece and score position via symbolic fingerprinting," in *Proc. Int. Society Music Information Retrieval Conf.*, 2012, pp. 433–438.

[6] A. Arzt, G. Widmer, and R. Sonnleitner, "Tempo- and transposition-invariant identification of piece and score position," in *Proc. Int. Society Music Information Retrieval Conf.*, 2014, pp. 549–554.

[7] M. Dorfer, A. Arzt, and G. Widmer, "Learning audio-sheet music correspondences for score identification and offline alignment," in *Proc. Int. Society Music Information Retrieval Conf.*, 2017, pp. 115–122.

[8] D. Byrd and J. G. Simonsen, "Towards a standard testbed for optical music recognition: Definitions, metrics, and page images," *J. New Music Res.*, vol. 44, no. 3, pp. 169–195, 2015.

[9] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri. (2013). Automatic music transcription: challenges and future directions. *J. Intelligent Inform. Syst.* [Online]. *41(3)*, pp. 407–434. Available: http://dx.doi.org/10.1007/s10844-013-0258-3

[10] E. Gómez, "Tonal description of music audio signals," Ph.D. dissertation, Universitat Pompeu Fabra, Barcelona, Spain, 2006.

[11] M. Müller, *Information Retrieval for Music and Motion*. Berlin: Springer-Verlag, 2007.

[12] H. Barlow and S. Morgenstern, *A Dictionary of Musical Themes*, rev. 3rd ed. New York: Crown, 1975.

[13] S. Balke, S. P. Achankunju, and M. Müller, "Matching musical themes based on noisy OCR and OMR input," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, 2015, pp. 703–707.

[14] S. Balke, V. Arifi-Müller, L. Lamprecht, and M. Müller, "Retrieving audio recordings using musical themes," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, 2016, pp. 281–285.

[15] J. Salamon, J. Serrà, and E. Gómez. (2013). Tonal representations for music retrieval: From version identification to query-by-humming. Int. J. Multimedia Inform. Retrieval. [Online]. 2(1), pp. 45–58. Available: http://dx.doi.org/10.1007/s13735-012-0026-0

[16] J. S. Juan J. Bosch, R M. Bittner and E. Gómez, "A comparison of melody extraction methods based on source-filter modelling," in *Proc. Int. Conf. Music Information Retrieval*, 2016, pp. 571–577.

[17] M. A. Casey, C. Rhodes, and M. Slaney, "Analysis of minimum distances in high-dimensional musical spaces," *IEEE Trans. Audio, Speech, Language Process. (2006–2013)*, vol. 16, no. 5, pp. 1015–1028, 2008.

[18] P. Grosche and M. Müller, "Toward characteristic audio shingles for efficient cross-version music retrieval," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, 2012, pp. 473–476.

[19] R. Sonnleitner and G. Widmer, "Robust quad-based audio fingerprinting," *IEEE Trans. Audio, Speech, Language Process. (2006–2013)*, vol. 24, no. 3, pp. 409–421, 2016.

[20] J. Six and M. Leman, "Panako: A scalable acoustic fingerprinting system handling time-scale and pitch modification," in *Proc. Int. Conf. Music Information Retrieval*, 2014, pp. 259–264.

[21] J. Serrà, E. Gómez, and P. Herrera, "Audio cover song identification and similarity: Background, approaches, evaluation and beyond," in *Advances in Music Information Retrieval* (Studies in Computational Intelligence Series), Z. W. Ras and A. A. Wieczorkowska, Eds. Berlin: Springer-Verlag, 2010, vol. 274, pp. 307–332.

[22] S. Böck and M. Schedl, "Polyphonic piano note transcription with recurrent neural networks," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, 2012, pp. 121–124.

[23] S. Sigtia, E. Benetos, and S. Dixon, "An end-to-end neural network for polyphonic piano music transcription," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 24, no. 5, pp. 927–939, 2016.

[24] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.

[25] R. Kiros, R. Salakhutdinov, and R. S. Zemel. (2014). Unifying visual-semantic embeddings with multimodal neural language models. arXiv. [Online]. Available: https://arxiv.org/abs/411.2539

[26] M. Dorfer, J. Schlüter, A. Vall, F. Korzeniowski, and G. Widmer, "End-to-end cross-modality retrieval with CCA projections and pairwise ranking loss," *Int. J. Multimedia Inform. Retrieval*, vol. 7, no. 2, pp. 117–128, 2017.

[27] D. Damm, C. Fremerey, V. Thomas, M. Clausen, F. Kurth, and M. Müller, "A digital library framework for heterogeneous music collections: From document acquisition to cross-modal interaction," *Int. J. Digital Libraries*, vol. 12, no. 2–3, pp. 53–71, 2012.

[28] R. B. Dannenberg and C. Raphael, "Music score alignment and computer accompaniment," *Commun. ACM*, vol. 49, no. 8, pp. 38–43, 2006.

[29] A. Arzt, S. Böck, S. Flossmann, H. Frostel, M. Gasser, C. C. S. Liem, and G. Widmer, "The piano music companion," in *Proc. European Conf. Artificial Intelligence*, 2014, pp. 1221–1222.

[30] M. Gasser, A. Arzt, T. Gadermaier, M. Grachten, and G. Widmer, "Classical music on the web: User interfaces and data representations," in *Proc. Int. Conf. Music Information Retrieval*, 2015, pp. 571–577.

[31] S. Balke, C. Dittmar, J. Abeßer, K. Frieler, M. Pfleiderer, and M. Müller, "Bridging the gap: Enriching YouTube videos with jazz music annotations," *Front. Dig. Humanities*, vol. 5, Feb. 2018. doi: 10.3389/fdigh.2018.00001.

SP

Zhiyao Duan, Slim Essid, Cynthia C.S. Liem,
Gaël Richard, and Gaurav Sharma

Audiovisual Analysis of Music Performances

Overview of an emerging field

©ISTOCKPHOTO.COM/TRAFFIC_ANALYZER

Digital Object Identifier 10.1109/MSP.2018.2875511
Date of publication: 24 December 2018

In the physical sciences and engineering domains, music has traditionally been considered an acoustic phenomenon. From a perceptual viewpoint, music is naturally associated with hearing, i.e., the audio modality. Moreover, for a long time, the majority of music recordings were distributed through audio-only media, such as vinyl records, cassettes, compact discs, and mp3 files. As a consequence, existing automated music analysis approaches predominantly focus on audio signals that represent information from the acoustic rendering of music.

Music performances, however, are typically multimodal [1], [2]: while sound plays a key role, other modalities are also critical to enhancing the musical experience. In particular, the visual aspects of music—be they disc cover art, videos of live performances, or abstract music videos—play an important role in expressing musicians' ideas and emotions. With the popularization of video-streaming services over the past decade, such visual representations also are increasingly available with distributed music recordings. In fact, video-streaming platforms have become one of the preferred music distribution channels, especially among the younger generation of music consumers.

Simultaneously seeing and listening to a musical performance often provides a richer experience than pure listening. Researchers have found that "the visual component is not a marginal phenomenon in music perception, but an important factor in the communication of meanings" [3]. Even for prestigious classical music competitions, studies have revealed that visually perceived elements of the performance, such as the musician's gestures, motions, and facial expressions, affect the evaluations of judges (experts or novices alike) even more significantly than the sound [4].

Symphonic music provides another example of visible communicated information where large groups of orchestra musicians play simultaneously in close coordination. For expert audiences familiar with the genre, both the visible coordination between musicians and the ability to closely watch individuals within the group add to the attendee's emotional experience of a concert [5]. Attendees unfamiliar with the genre can also be

better engaged via enrichment, i.e., offering supporting information in various modalities (e.g., visualizations or textual explanations) beyond the stimuli that the event naturally triggers in the physical world.

In addition to the audiences at music presentations, others also gain from information obtained through audiovisual rather than audio-only analysis. In educational settings, instrument learners benefit significantly from watching demonstrations by professional musicians, where the visual presentation provides deeper insight into specific instrument-technical aspects of the performance (e.g., fingering or choice of strings). Generally, when broadcasting audiovisual productions involving large ensembles captured with multiple recording cameras, it is also useful for the producer to be aware of which musicians are visible in which camera stream at each point in time. For such analyses to be done, relevant information needs to be extracted from the recorded video signals and coordinated with recorded audio. As a consequence, there has recently been growing interest in the visual analysis of musical performances, even though such analysis was largely overlooked in the past.

Aim and focus

In this article, we aim to introduce this emerging area to the music signal processing community and the broader signal processing community. To our knowledge, this article is the first overview of research in this area. For conciseness, we restrict our attention to the analysis of audiovisual music performances, which is an important subset of audiovisual music productions that is also representative of the main challenges and techniques of this field of study. Other specific applications, such as the analysis of music video clips or other types of multimodal recordings not involving audio and visuals (e.g., lyrics or music score sheets), although important in their own right, are not covered here to maintain a clear focus and a reasonable length.

Significance and challenges

Significance

Figure 1 illustrates some examples of how visual and aural information in a musical presentation complement each other, and how they offer more information on the performance than what can be obtained by considering only the audio channel and a musical score. In fact, while the musical score is often considered to be the ground truth of a musical presentation, significant performance-specific expressive information, such as the use of vibrato, is not indicated in the score and is instead evidenced in the audiovisual performance signals.

Compared to audio-only music performance analysis, the visual modality offers extra opportunities to extract musically meaningful cues out of recorded performance signals.

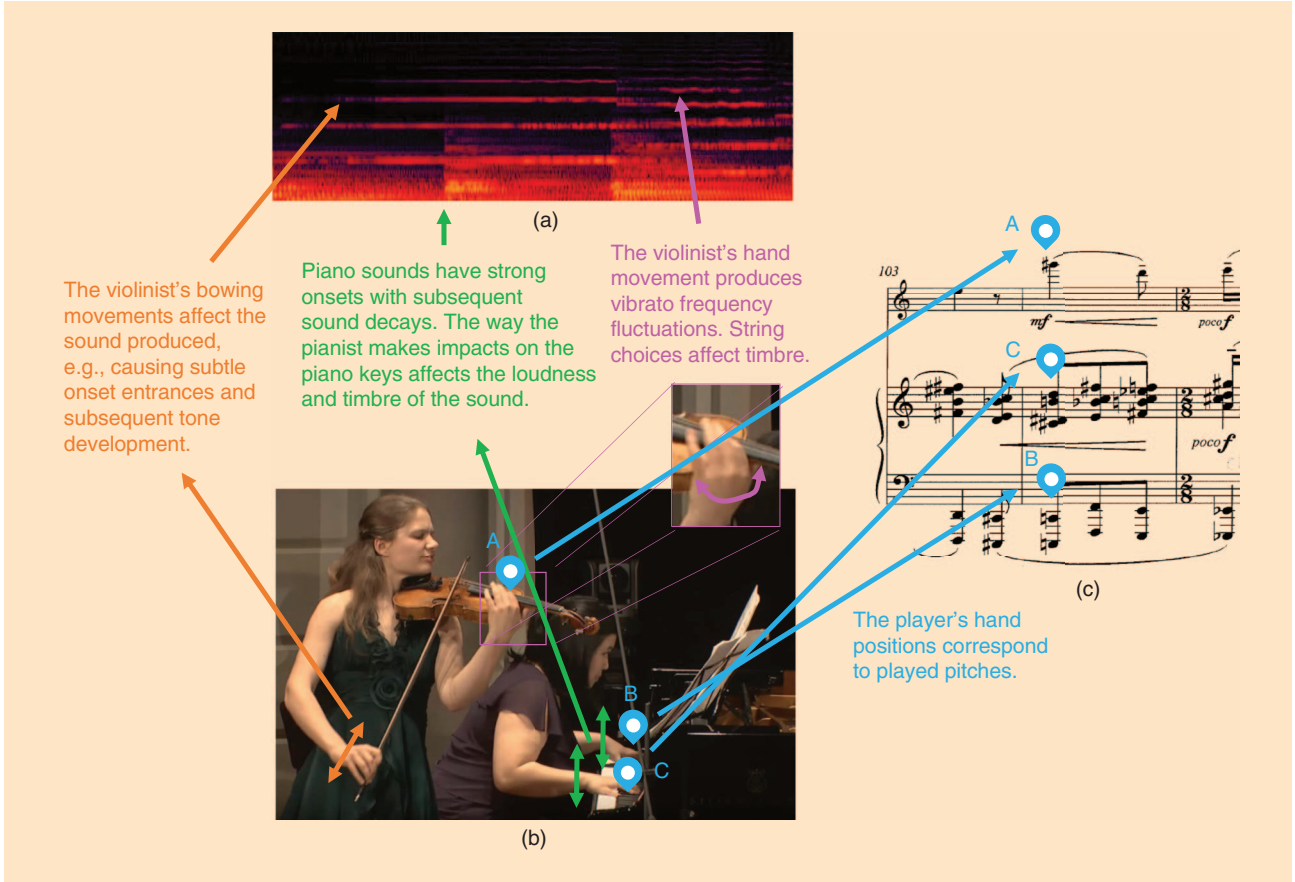


FIGURE 1. Examples of the information present in three parallel representations of a music performance excerpt: (a) a spectrogram of a recorded audio signal, (b) a video recording of performing musicians, and (c) a score of the performed music.

In some cases, the visual modality allows for addressing tasks that would not be possible in audio-only analysis, e.g., tracking a musician's fingerings or a conductor's gestures and analyzing individual players in the same instrumental section of an orchestra. In other cases, the visual modality provides significant help in task solving, e.g., in source separation and in the characterization of expressive playing styles. In the "Overview of Existing Research" section, we discuss several representative tasks along these lines.

Audiovisual analysis of musical performances broadens the scope of music signal processing research, connecting the audio signal processing area with other areas, i.e., image processing, computer vision, and multimedia. The integration of the audio and visual modalities also naturally creates a connection with emerging research areas, such as virtual reality and augmented reality, and extends music-related human–computer interaction. It serves as a controlled test bed for research on multimodal data analysis, which is critical for building robust and universal intelligent systems.

Challenges

The multimodal nature of audiovisual analysis of music poses new research challenges. First, the visual scenes of music presentations present new problems for image processing and computer vision. Indeed, the visual scene is generally cluttered, especially when multiple musicians are involved, who additionally may be occluded by each other and by music stands. Also, musically meaningful motions may be subtle (e.g., fingering and vibrato motion), and camera views may be complex (e.g., musicians not facing toward cameras, zoom-in/out, and changes of views).

Second, the way to integrate audio and visual processing in the modeling stage of musical scene analysis is a key challenge. In fact, independently tackling the audio and visual modalities to merely fuse the output of the corresponding (unimodal) analysis modules at a later stage is generally not an optimal approach. To take advantage of potential cross-modal dependencies, it is better to combine low-level audiovisual representations as early as possible in the data analysis pipeline. This is, however, not always straightforward. Certain visual signals (e.g., the bowing motion of string instruments) and audio signals (e.g., note onsets) of a sound source are often highly correlated, yet some performer movements (e.g., head nodding) are not directly related to sound [6]. How to discover and exploit audiovisual correspondence in a complex audiovisual scene of music performances is thus a key question.

Third, the lack of annotated data is yet another challenge. While commercial recordings are abundant, they are usually not annotated and are also subject to copyright restrictions that limit their distribution and use. Annotated audio data sets of musical performances are already scarce because of the complexities of recording and ground-truth annotation. Audiovisual data sets are even scarcer, and their creation requires more effort. The lack of large-scale annotated data sets limits the application of many supervised learning techniques that have proven successful for data-rich problems. We note that available music data sets were surveyed in a recent paper [7] that detailed the creation of a new multitrack audiovisual classical music data set. The data set provided in [7] was relatively small, with only 44 short pieces, but was richly annotated, providing individual instrument tracks to allow the assessment of source separation methods and associated music score information in a machine-readable format.

At the other end of the data spectrum, the YouTube-8M data set [8] provides a large-scale labeled video data set (with embedded audio) that also includes many music videos. However, the YouTube-8M data set is currently annotated only with overall video labels and therefore is suited primarily for video/audio classification tasks.

Overview of existing research

It is not an easy task to give a well-structured overview of an emerging field, yet here we make a first attempt from two perspectives. The following section categorizes the existing work into different analysis tasks for different instruments, while the section after that provides a perspective on the type of audiovisual correspondence that is exploited during the analysis.

Categorization of audiovisual analysis tasks

Table 1 organizes existing work on audiovisual analysis of musical presentations along two dimensions: 1) the type of musical instrument and 2) the analysis task.

The first dimension is not only a natural categorization of musicians in a music performance but is also indicative of the types of audiovisual information revealed during the performance. For example, percussionists show large-scale motions that are almost all related to sound articulation. Pianists' hand and finger motions are also related to sound articulation, but they are much subtler and also indicative of the notes being played (i.e., the musical content). For guitars and strings, the left-hand motions are indicative of the notes being played, while the right-hand motions tell us how the notes are articulated (e.g.,

Table 1. A categorization of existing research on audiovisual analysis of music performances according to the type of instrument and the analysis task.

Visual	Is Critical		Is Significant				
Tasks	**Fingering**	**Association**	**Play/Nonplay**	**Onset**	**Vibrato**	**Transcription**	**Separation**
Percussion	N/A	–	[9]	–	N/A	[10]	–
Piano	[11], [12]	–	–	–	N/A	–	–
Guitar	[13]–[16]	–	–	–	–	[16]	–
Strings	[17]	[18], [19]	[9], [20]	[19]	[21]	[17], [20]	[22]
Wind	–	–	[9]	[23]	–	–	–
Singing	N/A	–	–	–	–	–	–

Certain combinations of instruments and tasks do not make sense, and are marked N/A. Various techniques and their combinations have been employed, including support vector machines, hidden Markov models, nonnegative matrix factorization, and deep neural networks.

legato or staccato). For wind instruments, note articulations are difficult to see, and almost all visible motions (e.g., the fingering of a clarinet or the hand positioning of a trombone) are about notes. Finally, singers' mouth shapes reveal only the syllables being sung but not the pitch; also, their body movements can be correlated with the musical content but are not predictive enough for the details.

The second dimension is about tasks or aspects that the audiovisual analysis focuses on. The seven tasks/aspects are further classified into two categories: tasks in which visual analysis is critical and those in which visual analysis provides significant help. In the first category, there are the following tasks:

- *Fingering analysis*: It is very difficult to infer the fingering purely from audio, while it becomes possible by observing the finger positions. There has been research on fingering analysis from visual analysis for guitar [13]–[16], violin [17], and piano [11], [12]. Fingering patterns are mostly instrument specific, but the common idea is to track hand and finger positions relative to the instrument body.
- *Audiovisual source association*: This is a task that determines which player in the visual scene corresponds to which sound source in the audio mixture. The problem is addressed for string instruments by modeling the correlation between visual features and audio features, such as the association between bowing motions and note onsets [18] and that between vibrato motions and pitch fluctuations [19].

The second category contains more tasks. They can be listed as follows:

- *Playing/nonplaying (P/NP) activity detection*: In an ensemble or orchestral setting, it is extremely difficult to detect from the audio mixture whether a certain instrument is being played, yet the visual modality, if not occluded, offers a direct observation of the playing activities of each musician. Approaches based on image classification and motion analysis [9], [20] have been proposed.
- *Vibrato analysis*: This is for string instruments. The periodic movement of the fingering hand detected from visual analysis has been shown to correlate well with the pitch fluctuation of vibrato notes and has been used to detect vibrato notes and analyze the vibrato rate and depth [21].
- *Automatic music transcription*: This and its subtasks, such as multipitch analysis, are very challenging if only audio signals are available. Studies have found that audiovisual analysis is beneficial for monophonic instruments like the violin [17], polyphonic instruments like the guitar [16] and drums [10], and musical bodies like string ensembles [20]. The common underlying idea is to improve audio-based transcription results with play/nonplay activity detection and fingering analysis.
- *Audio source separation*: This is a task that can be significantly improved by audiovisual analysis. The motions of players are often highly correlated with the characteristics of the sound sources [6]. There has been work on modeling such correlations for audio source separation [22].

Besides instrumental players, conductor gesture analysis has also been investigated in audiovisual music performance analysis. Indeed, conductors do not directly produce sounds (besides occasional noises), but they are critical in musical presentations. Under the direction of different conductors, the same orchestra can produce significantly different renditions of the same musical piece. One musically interesting research problem is comparing the conducting behaviors of different conductors and analyzing their influences on the sound production of the orchestra. There has been work on conductor baton tracking [24] and gesture analysis [25] using visual analysis.

Different levels of audiovisual correspondence

Despite the various forms of music performances and analysis tasks, the common underlying idea of audiovisual analysis is to find and model the correspondence between audio and visual modalities. This correspondence can be static, i.e., between a fixed image and a short time frame of audio. For example, a certain posture of a flutist is indicative of whether the musician is playing or not; a static image of a fingering hand is informative regarding the notes being played.

This correspondence can also be dynamic, i.e., between a dynamic movement observed in the video and the fluctuation of audio characteristics. For example, a strumming motion of a guitar player's right hand is a strong indicator of the rhythmic pattern of the music passage; the periodic rolling motion of a violin player's left hand corresponds well to the pitch fluctuation of vibrato notes. Because of the large variety of instruments and their unique playing techniques, this dynamic correspondence is often instrument specific. The underlying idea of dynamic correspondence, however, is universal among different instruments. Therefore, it is appealing to build a unified framework for capturing this dynamic correspondence. If such correspondence can be captured robustly, the visual information can be better exploited to stream the corresponding audio components into sources, leading to visually informed source separation.

In the following three sections, we further elaborate upon these different levels of audiovisual correspondence by summarizing existing works and presenting concrete examples.

Static audiovisual correspondence

In this section, we first discuss works focusing on the modeling of static audiovisual correspondence in musical performances. *Static* here refers to correspondences between sonic realizations and their originating sources that remain stable over the course of a performance and for which the correspondence analysis does not rely on short-time dynamic variations. After giving a short overview with more concrete examples, a more extended case study discussion will be given on P/NP detection in instrument ensembles.

Overview

Typical static audiovisual correspondences have to do with positions and poses: which musician sits where, at what parts of the instrument the interaction occurs that leads to sound production, and how the interaction with the instrument can be characterized.

Regarding musicians' positions, when considering large ensemble situations, it is too laborious for a human to annotate

every person in every shot, especially when multiple cameras record the performance at once. At the same time, because of the typically uniform concert attire worn by ensemble members and musicians being part of large player groups that will actively move and occlude one another, recognizing individual players purely by computer vision methods is again a nontrivial problem, for which it also would be unrealistic to acquire large amounts of training data. However, within the same piece, orchestra musicians will not change positions relative to one another. Therefore, the orchestra setup can be considered as a quasi-static scene.

The work in [26] proposed to identify each musician in each camera over a full-recording timeline by combining partial visual recognition with knowledge of the scene's configuration and a human-in-the-loop approach in which humans were strategically asked to indicate the identities of performers in visually similar clusters. With minimal human interaction, a scene map was built up, and the spatial relations within this scene map assisted face clustering in crowded quasi-static scenes.

Regarding positions of interest on an instrument, work has been performed on the analysis of fingering. This can be seen as static information, as the same pressure action on the same position of the instrument will always yield the same pitch realization. Visual analysis has been performed to analyze fingering actions on pianos [11], [12], guitars [13]–[16], and violins [16], [17]. The main challenges involve the detection of the fingers in unconstrained situations and without the need to add markers to the fingers.

Case study: P/NP detection in orchestras

Whether individual musicians in large ensembles are playing their instrument or not seems to be unimportant; however, this information can be significant to critical in audiovisual analysis. Within the same instrument group, not all players may be playing at once. If this occurs in a multichannel audio recording, it is not trivial to distinguish which subset of individuals is playing, while this will visually be obvious. Furthermore, having a global overview of which instruments are active and visible in performance recordings provides useful information for audiovisual source separation.

In [9], a method was proposed to detect P/NP information in multicamera recordings of symphonic concert performances in which unconstrained camera movements and varying shooting perspectives occur. As a consequence, performance-related movement may not always be easily observed from the video, although coarser P/NP information can still be inferred through face and pose clustering.

A hierarchical method was proposed, which is illustrated in Figure 2 and that focuses on employing clustering techniques rather than learning sophisticated human–object interaction models. First, musician diarization is performed to annotate which musician appears when and where in a video. For this, keyframes are extracted at regular time intervals. In each keyframe, face detection is performed, including an estimation of the head pose angle and an inference of bounding boxes for the hair and upper body of the player. Subsequently, segmentation is performed on the estimated upper body of the musician, taking into account the gaze direction of the musician, as the instrument is expected to be present in the same direction.

After this segmentation step, face clustering methods are applied, including several degrees of contextual information (e.g., on the scene and upper body) and different feature sets, the richest ones consisting of a pyramid histogram of oriented gradients, the Joint Composite Descriptor, Gabor texture, edge histogram, and auto color correlogram.

Upon obtaining per-musician clusters, a renewed clustering is performed per musician, aiming to generate subclusters that contain images of only the same musician, performing one particular type of object interaction, recorded from one particular camera viewpoint. Finally, a human annotator action completes the labeling step: an annotator has to indicate who the musician is and whether a certain subcluster contains a playing or nonplaying action. As the work in [9] investigated various experimental settings (e.g., clustering techniques and feature sets), yielding thousands of clusters, the expected annotator action at various levels of strictness is simulated by setting various thresholds on how dominant a class within a cluster should be.

An extensive discussion of evaluation outcomes per framework module is given in [9]. Several takeaway messages can be derived from this work. First of all, the face and upper body regions are most informative for clustering. Furthermore,

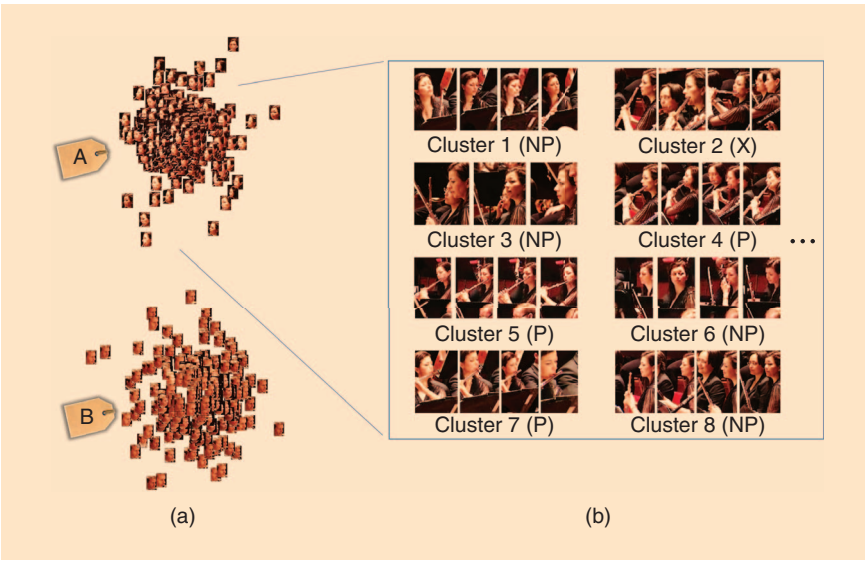


FIGURE 2. An example of hierarchical clustering steps for P/NP detection: (a) Diarization is performed on global face clustering results to identify a musician's identity. (b) Then, within each global artist cluster, subclusters are assigned with a P/NP label.

the proposed method can effectively discriminate playing versus nonplaying action, while generating a reasonable number of subclusters (i.e., enough to yield informative subclusters, but not too many, which would cause a high annotator workload). Face information alone may already be informative, as it indirectly reveals pose. However, in some cases, clustering cannot yield detailed, relevant visual analyses (e.g., subtle mouth movements for a wind player), and the method has a bias toward false positives, which can be caused by playing-anticipation movement. The application of merging strategies per instrumental part helps in increasing timeline coverage, even if a musician is not always detected. Finally, high annotator rejection thresholds (demanding clear majority classes within clusters) effectively filter out nonpure clusters.

One direct application of P/NP activity detection is in automatic music transcription. In particular, for multipitch estimation (MPE), P/NP information can be used to improve the estimation of instantaneous polyphony (i.e., the number of pitches at a particular time) of an ensemble performance, assuming that each active instrument produces only one pitch at a time. Instantaneous polyphony estimation is a difficult task from the audio modality itself, and its errors constitute a large proportion of music transcription errors. Furthermore, P/NP is also helpful for multipitch streaming (MPS), i.e., assigning pitch estimates to pitch streams corresponding to instruments: a pitch estimate should be assigned only to an active source. This idea has been explored in [20], and it was shown that both MPE and MPS accuracies are significantly improved by P/NP activity detection for ensemble performances.

Dynamic audiovisual correspondence

In a music performance, a musician makes many movements [6]. Some of these (e.g., bowing and fingering) are the articulation sources of sound while others (e.g., head shaking) are responses to the performance. In both cases, the movements show a strong correspondence with certain feature fluctuations in the music audio. Capturing this dynamic correspondence is important for the analysis of musical presentations.

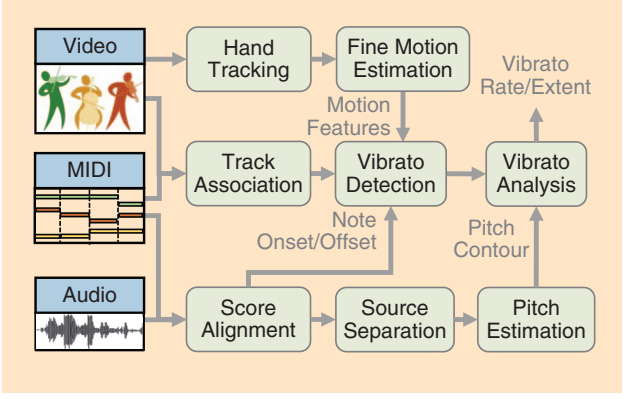


FIGURE 3. An overview of an audiovisual vibrato detection and analysis system for string instruments in ensemble performances that was proposed in [21].

Overview

Because of the large variety of musical instruments and their playing techniques, dynamic audiovisual correspondence displays different forms. In the literature, researchers have investigated the correspondence between bowing motions and note onsets of string instruments [18], between hitting actions and drum sounds of percussion instruments [10], and between left-hand rolling motions and pitch fluctuations of string vibrato to notes [19], [21]. On the visual modality, object tracking and optical flow techniques have been adopted to track relevant motions, while on the audio modality, different audio features have been considered.

The main challenge lies in determining where to look for the dynamic correspondence and what to look for. This is challenging not only because the correspondence is dependent on the instrument and playing technique, but also because there are many irrelevant motions in the visual scene [6] and interferences from multiple, simultaneous sound sources in the audio signal. Almost all existing methods rely on prior knowledge of the instrument type and playing techniques to attend to relevant motions and sound features. For example, for the association between string players and score tracks, [18] captured the correspondence between bowing motions and some note onsets. This is informed by the fact that many string instrument notes are started with a new bow stroke and that different tracks often show different onset patterns. For the association of wind instruments, the onset cue is still useful, but the motion capture module would need to be revised to capture the more subtle and diverse finger movements.

Case study: Vibrato analysis of string instruments

Vibrato is an important musical expression, and vibrato analysis is important for musicological studies, music education, and music synthesis. Acoustically, vibrato is characterized by a periodic fluctuation of pitch, with a rate between 5 and 10 Hz. Audio-based vibrato analysis methods rely on the estimation of the pitch contour. In an ensemble setting, however, multipitch estimation is very challenging because of the interference of other sound sources. For string instruments, vibrato is the result of periodic change of the length of the vibrating string, which is effectuated by the rolling motion of the left hand. If the rolling motion is observable, then vibrato notes can be detected and studied with the help of visual analysis. Because such analysis does not suffer from the presence of other sound sources (barring occlusion), it offers a tremendous advantage for vibrato analysis of string instruments in ensemble settings.

In [21], an audiovisual vibrato detection and analysis system was proposed. As shown in Figure 3, this approach integrates audio, visual, and score information and contains several modules to capture the dynamic correspondence among these modalities.

The first step is to detect and track the left hand for each player using the Kanade–Lucas–Tomasi tracker. This results in a dynamic region of the tracked hand, shown as the green box in Figure 4(a). Optical flow analysis is then performed to

calculate motion velocity vectors for each pixel in this region in each video frame. Motion vectors in frame t are spatially averaged as $\mathbf{u}(t) = [u_x(t), u_y(t)]$, where u_x and u_y represent the mean motion velocities in the x and y directions, respectively. It is noted that these motion vectors may also contain the slower large-scale body movements that are not associated with vibrato. Therefore, to eliminate the body movement effects, the moving average of the signal $\mathbf{u}(t)$ is subtracted from itself to obtain a refined motion estimation $\mathbf{v}(t)$. Figure 4(c) shows the distribution of all $\mathbf{v}(t)$ across time, from which the principal motion direction can be inferred through principal component analysis, which aligns well along the fingerboard. The projection of the motion vector $\mathbf{v}(t)$ onto the principal direction is defined as the one-dimensional (1-D) motion velocity curve $V(t)$. Taking an integration over time, one obtains a 1-D hand displacement curve $X(t) = \int_0^t V(\tau)d\tau$, which corresponds directly to the pitch fluctuation.

To use the motion information to detect and analyze vibrato notes, one needs to know to which note the hand motion corresponds. This is solved by audiovisual source association and audio–score alignment. In this work, audiovisual source association is performed through the correlation between bowing motions and note onsets, as described in [18]. Audio–score alignment [27] synchronizes the audiovisual performance (assuming perfect audiovisual synchronization) with the score, from which onset and offset times of each note are estimated. This can be done by comparing the harmonic content of the audio and the score and dynamic time warping. Score-informed source separation is then performed, and the pitch contour of each note is estimated from the separated source signal.

Given the correspondence between the motion vectors and the sound features (pitch fluctuations) of each note, vibrato detection is performed with two methods. The first uses a support vector machine to classify each note as vibrato or nonvibrato using features extracted from the motion vectors. The second technique simply sets a threshold on the autocorrelation of the 1-D hand displacement curve $X(t)$.

For vibrato notes, the vibrato rate can also be calculated from the autocorrelation of the hand displacement curve $X(t)$. However, the vibrato extent (i.e., the dynamic range of the pitch contour) cannot be estimated by capturing the motion extent. This is because it varies based upon the camera distance and angle as well as the vibrato articulation style, hand position, and instrument type. To address this issue, the hand displacement curve is scaled to match the estimated noisy pitch contour from score-informed audio analysis. Specifically, assuming $F(t)$ is the estimated pitch contour [in a Musical Instrument Digital Interface (MIDI) number] of the detected vibrato note from audio analysis after subtracting its dc component, the vibrato extent v_e (in musical cents) is estimated as $\hat{v}_e$, with

$$\hat{v}_e = \arg\min_{v_e} \sum_{t=t^{\text{on}}}^{t^{\text{off}}} \left| 100 \cdot F(t) - v_e \frac{X(t)}{\hat{w}_e} \right|^2, \tag{1}$$

where $100 \cdot F(t)$ is the pitch contour in musical cents and $\hat{w}_e$ is the dynamic range of $X(t)$.

Music source separation using dynamic correspondence

Audio source separation in music recordings is a particularly interesting task, where audiovisual matching between the visual events of a performer's actions and their audio rendering can be of great value. Notably, such an approach enables addressing audio separation tasks that could not be performed in a unimodal fashion (solely analyzing the audio signal), as when considering two or more instances of the same instruments, say, a duet of guitars or violins, as done in the work of Parekh et al. [22]. Knowing whether a musician is playing or not at a particular point in time gives important cues for source allocation. Seeing the hand and finger movements of a cellist

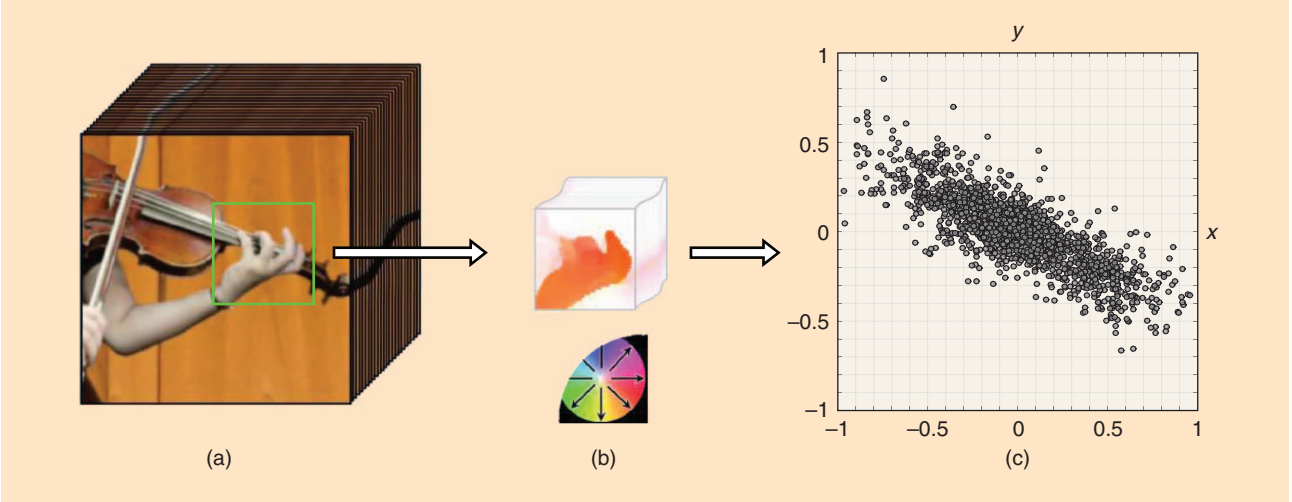


FIGURE 4. The motion capture results from (a) left-hand tracking, (b) color-encoded pixel velocities, and (c) a scatter plot of frame-wise refined motion velocities.

helps us attend to the cello's section sound in an orchestral performance. The same idea applies to visually informed audio source separation.

Overview

There is a large body of work in multimodal (especially audiovisual) source separation for speech signals, but much less effort has been dedicated to audiovisual music performance analysis for source separation. It was shown in the work of Godoy et al. [6], however, that there are certain players' motions that are highly correlated to the sound characteristics of audio sources. In particular, by analyzing a solo piano performance, the authors highlighted the correlation that may exist between music and hand movements or the sway in the upper body. An earlier work by Barzelay and Shechner [28] exploited such a correlation in introducing an audiovisual system for individual musical source enhancement in violin–guitar duets. The authors isolated audio-associated visual objects by searching for cross-modal temporal incidences of events and then used these to perform musical source separation.

Case study: Motion-driven source separation in a string quartet

The idea that motion characteristics obtained from visual analysis encode information about the physical excitation of a sounding object is also exploited in more recent studies. As an illustration, we detail a model in which it is assumed that the characteristics of a sound event (e.g., a musical note) is highly correlated with the speed of sound-producing motion [22]. More precisely, the proposed approach extends the popular nonnegative matrix factorization (NMF) framework using visual information about objects' motion. Applied to string quartets, the motion of interest is mostly carried by the bow speed. The main steps of this method are the following (see Figure 5):

1) Gather motion features, i.e., average motion speeds (further described later), in a data matrix $\mathbf{M} \in \mathbb{R}_+^{N \times C}$ that summarizes the speed information of the coherent motion trajectories within predefined regions. In the simplest case, there is one region per musician (i.e., per source). $C = \sum_j C_j$ is the number of motion clusters, where C_j is the number of clusters per source j, and N is the frame size of the short-time Fourier transform (STFT) used for computing the audio signal's spectrogram.
2) Ensure that the typical motion speeds (such as the bow speed) are active synchronously with the typical audio events. This is done by constraining the audio spectrogram decomposition obtained by NMF $\mathbf{V} \approx \mathbf{WH}$ and the motion data decomposition $\mathbf{M} \approx \mathbf{H}^\top \mathbf{A}$ to share the same activity matrix $\mathbf{H} \in \mathbb{R}_+^{K \times N}$, where $\mathbf{W} \in \mathbb{R}_+^{F \times K}$ is the matrix collecting the so-called nonnegative audio spectral patterns (column-wise), and where $\mathbf{A} = [\alpha_1, \ldots, \alpha_c]$ gathers nonnegative linear regression coefficients for each motion cluster, with $\alpha_c = [\alpha_{1c}, \ldots, \alpha_{Kc}]^T$.
3) Ensure that only a limited number of motion clusters is active at a given time. This can be done by imposing a sparsity constraint on $\mathbf{A}$.
4) Assign an audio pattern to each source for separation and reconstruction. This is done by assigning the kth basis vector (column of $\mathbf{W}$) to the jth source, if $\text{argmax}_c \alpha_{kc}$ belongs to the jth source cluster. The different sources are then synthesized by element-wise multiplication between the soft mask, given by $(\mathbf{W}_j \mathbf{H}_j)./(\mathbf{WH})$, and the mixture spectrogram, followed by an inverse STFT, where ./ stands for element-wise division, and $\mathbf{W}_j$ and $\mathbf{H}_j$ are the submatrices of spectral patterns $\mathbf{w}_k$ and their activations $\mathbf{h}_k$ assigned to the jth source (see Figure 6).

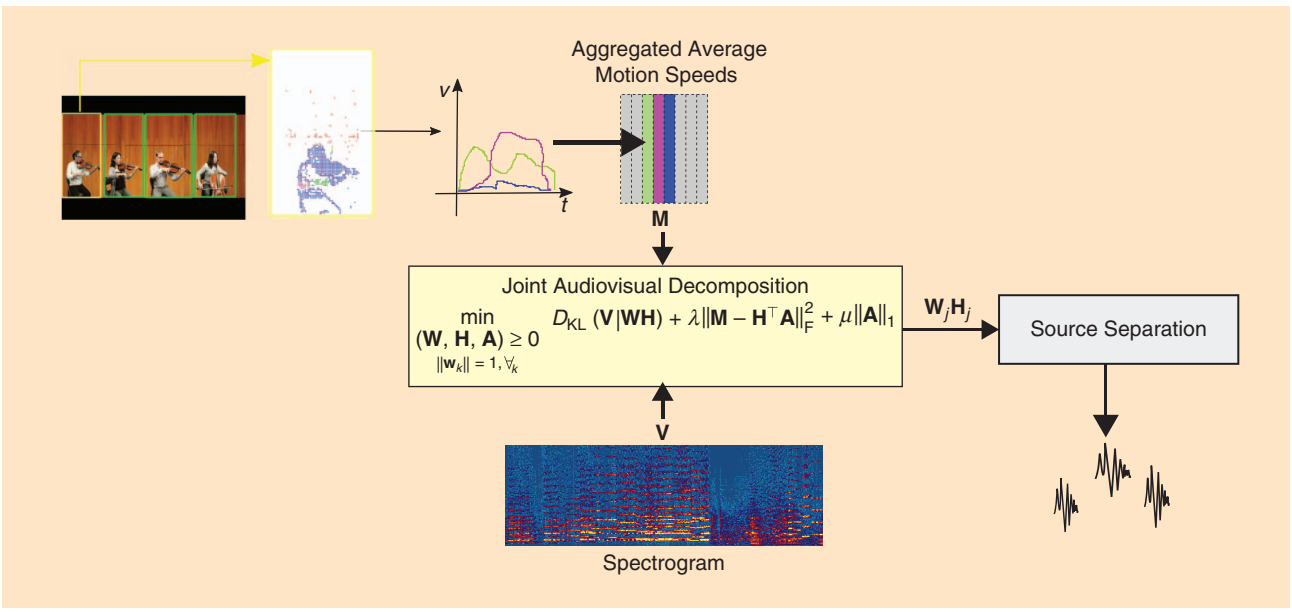


FIGURE 5. A joint audiovisual music source separation system.

A possible formulation for the complete model can then be written as the following optimization problem:

$$\underset{\substack{(\mathbf{W},\mathbf{H},\mathbf{A})\geq 0\\ \|\boldsymbol{w}_k\|=1,\ \forall k}}{\text{minimize}} \quad D_{KL}(\mathbf{V}|\mathbf{WH}) + \lambda\left\|\mathbf{M}-\mathbf{H}^{\top}\mathbf{A}\right\|_F^2 + \mu\left\|\mathbf{A}\right\|_1, \quad (2)$$

where D_{KL} is the Kullback–Leibler divergence, λ and μ are positive hyperparameters (to be tuned), and $\|\,.\,\|_F$ is the Frobenius norm.

More details can be found in [22], but for most situations, this joint audiovisual approach significantly outperformed the corresponding sequential approach proposed by the same authors and the audio-only approach introduced in [29]. For example, for a subset of the University of Rochester Multimodal Music Performance data set [7], the joint approach obtained a signal-to-distortion ratio of 7.14 dB for duets and 5.14 dB for trios, while the unimodal approach of [29] obtained signal-to-distortion ratios of 5.11 dB and 2.18 dB, respectively. It is worth mentioning that, in source separation, a difference of +1 dB is usually acknowledged as significant.

The correlation between motion in the visual modality and audio is also at the core of some other recent approaches. While bearing some similarities to the system detailed previously, the approach explained in [18] further exploits the knowledge of the MIDI score to well align the audio recording (e.g., onsets) and video (e.g., bow speeds). An extension of this work is presented in [19], where the audiovisual source association is performed through a multimodal analysis of vibrato notes. It is in particular shown that the fine-grained motion of the left hand is strongly correlated with the pitch fluctuation of vibrato notes and that this correlation can be used for audiovisual music source separation in a score-informed scenario.

Current trends and future work

This article provides an overview of the emerging field of audiovisual music performance analysis. We used specific case studies to highlight how techniques from signal processing, computer vision, and machine learning can jointly exploit the information contained in the audio and visual modalities to effectively address a number of music analysis tasks.

Current work in audiovisual music analysis has been constrained by the availability of data. Specifically, the relatively small size of current annotated audiovisual data sets has precluded the extensive use of data-driven machine-learning approaches, such as deep learning. Recently, deep learning has been utilized for vision-based detection of acoustic timed music events [23]. Specifically, the detection of onsets performed by clarinet players was addressed in this work by using a three-dimensional convolutional neural network (CNN) that relied on multiple streams, each based on a dedicated region of interest (ROI) from the video frames that was relevant to sound production. For each ROI, a reference frame was examined in the context of a short surrounding frame sequence, and the desired target was labeled as either an *onset* or *not an onset*. Although state-of-the-art audio-based onset detection methods outperform the model proposed in [23], the data set, task setup, and architecture setup gave rise to interesting research questions, especially on how to deal with significant events in temporal multimedia streams that occur at fine temporal and spatial resolutions.

Interesting ideas exploiting deep-learning models can also be found in related fields. For example, in [30] a promising strategy in the context of emotional analysis of music videos was introduced. The approach consisted in fusing learned audiovisual midlevel representations using CNNs. Another important promising research direction is transfer learning, which could better cope with the limited size of annotated

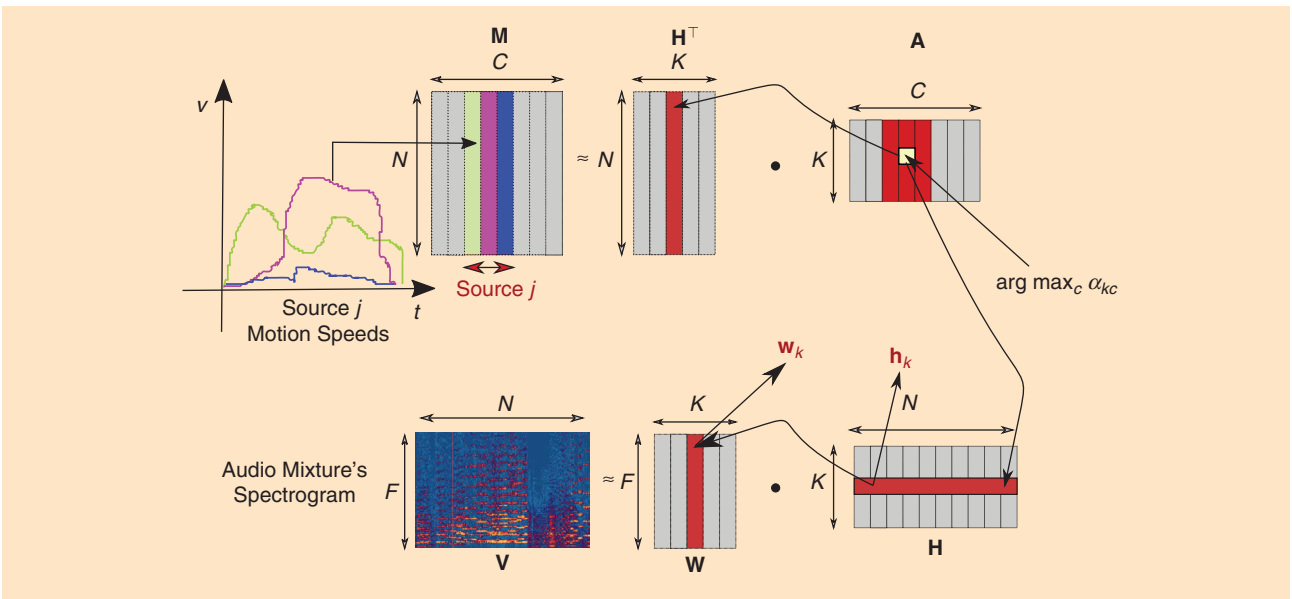


FIGURE 6. A joint audiovisual source separation—an illustration of the audio pattern assignment to source *j* (an example for the *k*th basis vector).

audiovisual musical performance data sets. As highlighted in [31], it is possible to learn an efficient audio feature representation for an audio-only application, specifically audio event recognition, by using a generic audiovisual database.

The inherent mismatch between the audio content and the corresponding image frames in a large majority of video recordings remains a key challenge for audiovisual music analysis. For instance, at a given point in time, edited videos of live performances often show only part of the performers' actions (think of live orchestra recordings). In such situations, the audiovisual analysis systems need to be flexible enough to effectively exploit the partial and intermittent correspondences between the audio and visual streams. Multiple-instance learning techniques already used for multimodal event detection in the computer vision community may offer an attractive option for addressing this challenge.

As new network architectures are developed for dealing with multimodal temporal signals and as significantly larger annotated data sets become available, we expect that deep learning-based data-driven approaches will lead to rapid progress in audiovisual music analysis, mirroring the deep-learning revolution in computer vision, natural-language processing, and audio analysis.

Beyond the immediate examples included in the case studies presented in this article, audiovisual music analysis can be extended toward other music genres, including pop, jazz, and world music. It can also help improve a number of applications in various musical contexts. Video-based tutoring for music lessons is already popular (e.g., guitar lessons on YouTube). The use of audiovisual music analysis can make such lessons richer by better highlighting the relations between the player's actions and the resulting musical effects. Audiovisual music analysis can similarly be used to enhance other music understanding/learning activities, including score-following, auto-accompaniment, and active listening.

Better tools for modeling the correlation between visual and audio modalities can also enable novel applications beyond the analysis of music performances. For example, in recent work on cross-modal audiovisual generation [32], sound-to-image sequence generation and video-to-sound spectrogram generation have been demonstrated using deep generative adversarial networks. Furthermore, the underlying tools and techniques can also help address other performing arts that involve music. Examples of such work include dance movement classification [33] and alignment of different dancers' movements within a single piece [34] by using (visual) gesture tracking and (audio) identification of stepping sounds.

Acknowledgments

Zhiyao Duan is partially supported by National Science Foundation grant 1741472.

Authors

Zhiyao Duan (zhiyao.duan@rochester.edu) received his B.S. degree in automation and his M.S. degree in control science and engineering from Tsinghua University, Beijing, in 2004 and 2008, respectively, and his Ph.D. degree in computer science from Northwestern University, Evanston, Illinois, in 2013. He is an assistant professor in the Department of Electrical and Computer Engineering and the Department of Computer Science, University of Rochester, New York. His research interest is in the broad area of computer audition, i.e., designing computational systems that are capable of understanding sounds, including music, speech, and environmental sounds. He copresented a tutorial on automatic music transcription at the International Conference on Music Information Retrieval 2015. He received a best paper award at the 2017 Sound and Music Computing Conference and a best paper nomination at the International Conference on Music Information Retrieval 2017. He is a Member of the IEEE.

Slim Essid (slim.essid@telecom-paristech.fr) received his state engineering degree from the École Nationale d'Ingénieurs de Tunis, Tunisia, in 2001, his M.Sc. (D.E.A.) degree in digital communication systems from the École Nationale Supérieure des Télécommunications, Paris, France, in 2002, his Ph.D. degree from the Université Pierre et Marie Curie (UPMC), Paris, France, in 2005, and his Habilitation à Diriger des Recherches degree from UPMC in 2015. He is a professor in Telecom ParisTech's Department of Images, Data, and Signals and the head of the Audio Data Analysis and Signal Processing team. His research interests are machine learning for audio and multimodal data analysis. He has been involved in various collaborative French and European research projects, among them Quaero, Networks of Excellence FP6-Kspace, FP7-3DLife, FP7-REVERIE, and FP-7 LASIE. He has published over 100 peer-reviewed conference and journal papers, with more than 100 distinct coauthors. On a regular basis, he serves as a reviewer for various machine-learning, signal processing, audio, and multimedia conferences and journals, e.g., a number of IEEE transactions, and as an expert for research funding agencies.

Cynthia C.S. Liem (c.c.s.liem@tudelft.nl) received her B.Sc., M.Sc., and Ph.D. degrees in computer science from Delft University of Technology, The Netherlands, in 2007, 2009, and 2015, respectively, and her B.Mus. and M.Mus. degrees in classical piano performance from the Royal Conservatoire, The Hague, The Netherlands, in 2009, and 2011, respectively. Currently, she is an assistant professor in the Multimedia Computing Group of Delft University of Technology. Her research focuses on music and multimedia content discovery through search and recommendation techniques. She gained industrial experience at Bell Labs Netherlands, Philips Research, and Google and is a recipient of several major grants and awards, including the Lucent Global Science Scholarship, Google European Doctoral Fellowship, and Netherlands Organisation for Scientific Research Veni grant. She is a Member of the IEEE.

Gaël Richard (gael.richard@telecom-paristech.fr) received his State Engineering degree from Télécom ParisTech, France, in 1990, his Ph.D. degree from the University of Paris XI, France, in 1994 in speech synthesis, and his Habilitation à

Diriger des Recherches degree from the University of Paris XI in 2001. After receiving his Ph.D. degree, he spent two years at Rutgers University, Piscataway, New Jersey, in the Speech Processing Group of Prof. J. Flanagan, where he explored innovative approaches for speech production. In 2001, he joined Télécom ParisTech, where he is now a professor in audio signal processing and the head of the Image, Data, and Signal Department. He is coauthor of more than 200 papers, His research interests are mainly in the field of speech and audio signal processing and include topics such as signal representations and signal models, source separation, machine-learning methods for audio/music signals, music information retrieval, and multimodal audio processing. He is a Fellow of the IEEE.

Gaurav Sharma (gaurav.sharma@rochester.edu) received his B.S. degree in electronics and communication engineering from the Indian Institute of Technology, Roorkee (formerly the University of Roorkee), in 1990 and his Ph.D. degree in electrical and computer engineering from North Carolina State University, Raleigh, in 1996. He is a professor in the Department of Electrical and Computer Engineering, Department of Computer Science, and Department of Biostatistics and Computational Biology, University of Rochester, New York. From 1996 to 2003, he was with Xerox Research and Technology, Webster, New York, initially as a member of the research staff and subsequently in the position of principal scientist. His research interests include multimedia signal processing, media security, image processing, computer vision, and bioinformatics. He is the editor-in-chief of *IEEE Transactions on Image Processing*. From 2011 to 2015, he was the editor-in-chief of *Journal of Electronic Imaging* and is the editor of *Color Imaging Handbook* (CRC Press, 2003). He is a fellow of SPIE and of the Society of Imaging Science and Technology and a member of Sigma Xi. He is a Fellow of the IEEE.

References

[1] C. C. S. Liem, M. Müller, D. Eck, G. Tzanetakis, and A. Hanjalic, "The need for music information retrieval with user-centered and multimodal strategies," in *Proc. Int. ACM Workshop Music Information Retrieval User-Centered and Multimodal Strategies at ACM Multimedia*, 2011, pp. 1–6.

[2] S. Essid and G. Richard. (2012). Fusion of multimodal information in music content analysis. Multimodal Music Processing. Dagstuhl, Germany: Schloss Dagstuhl-Leibniz-Zentrum für Informatik. [Online]. *3*, pp. 37–52. Available: http://drops.dagstuhl.de/opus/volltexte/2012/3465

[3] F. Platz and R. Kopiez, "When the eye listens: A meta-analysis of how audio-visual presentation enhances the appreciation of music performance," *Music Perception: Interdisc. J.*, vol. 30, no. 1, pp. 71–83, 2012.

[4] C.-J. Tsay, "Sight over sound in the judgment of music performance," *Nat. Acad. Sci.*, vol. 110, no. 36, pp. 14,580–14,585, 2013.

[5] M. S. Melenhorst and C. C. S. Liem, "Put the concert attendee in the spotlight. a user-centered design and development approach for classical concert applications," in *Proc. Int. Society Music Information Retrieval Conf.*, 2015, pp. 800–806.

[6] R. I. Godøy and A. R. Jensenius, "Body movement in music information retrieval," in *Proc. Int. Society Music Information Retrieval Conf.*, 2009, pp. 45–50.

[7] B. Li, X. Liu, K. Dinesh, Z. Duan, and G. Sharma, "Creating a multi-track classical music performance dataset for multi-modal music analysis: Challenges, insights, and applications," *IEEE Trans. Multimedia*, 2018. doi: 10.1109/TMM.2018.2856090.

[8] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan. (2016). YouTube-8M: A large-scale video classification benchmark. arXiv. [Online]. Available: http://arxiv.org/abs/1609.08675

[9] A. Bazzica, C. C. S. Liem, and A. Hanjalic, "On detecting the playing/non-playing activity of musicians in symphonic music videos," *Comput. Vision Image Understanding*, vol. 144, pp. 188–204, Mar. 2016.

[10] K. McGuinness, O. Gillet, N. E. O'Connor, and G. Richard, "Visual analysis for drum sequence transcription," in *Proc. IEEE European Signal Processing Conf.*, 2007, pp. 312–316.

[11] D. Gorodnichy and A. Yogeswaran, "Detection and tracking of pianist hands and fingers," in *Proc. Canadian Conf. Computer and Robot Vision*, 2006. doi: 10.1109/CRV.2006.26.

[12] A. Oka and M. Hashimoto, "Marker-less piano fingering recognition using sequential depth images," in *Proc. Korea-Japan Joint Workshop Frontiers Computer Vision*, 2013. doi: 10.1109/FCV.2013.6485449.

[13] A.-M. Burns and M. M. Wanderley, "Visual methods for the retrieval of guitarist fingering," in *Proc. Int. Conf. New Interfaces Musical Expression*, 2006, pp. 196–199.

[14] C. Kerdvibulvech and H. Saito, "Vision-based guitarist fingering tracking using a Bayesian classifier and particle filters," in *Proc. Pacific Rim Conf. Advances in Image and Video Technology*, 2007, pp. 625–638.

[15] J. Scarr and R. Green, "Retrieval of guitarist fingering information using computer vision," in *Proc. Int. Conf. Image and Vision Computing New Zealand*, 2010. doi: 10.1109/IVCNZ.2010.6148852.

[16] M. Paleari, B. Huet, A. Schutz, and D. Slock, "A multimodal approach to music transcription," in *Proc. Int. Conf. Image Processing*, 2008, pp. 93–96.

[17] B. Zhang and Y. Wang, "Automatic music transcription using audio-visual fusion for violin practice in home environment," Nat. Univ. Singapore, Tech. Rep. TRA7/09, 2009.

[18] B. Li, K. Dinesh, Z. Duan, and G. Sharma, "See and listen: Score-informed association of sound tracks to players in chamber music performance videos," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, 2017, pp. 2906–2910.

[19] B. Li, C. Xu, and Z. Duan, "Audiovisual source association for string ensembles through multi-modal vibrato analysis," in *Proc. Sound and Music Computing*, 2017, pp. 159–166.

[20] K. Dinesh, B. Li, X. Liu, Z. Duan, and G. Sharma, "Visually informed multi-pitch analysis of string ensembles," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, 2017, pp. 3021–3025.

[21] B. Li, K. Dinesh, G. Sharma, and Z. Duan, "Video-based vibrato detection and analysis for polyphonic string music," in *Proc. Int. Society Music Information Retrieval Conf.*, 2017, pp. 123–130.

[22] S. Parekh, S. Essid, A. Ozerov, N. Q. K. Duong, P. Perez, and G. Richard, "Guiding audio source separation by video object information," in *Proc. IEEE Workshop Applications Signal Processing Audio and Acoustics*, 2017, pp. 61–65.

[23] A. Bazzica, J. C. van Gemert, C. C. S. Liem, and A. Hanjalic. (2017). Vision-based detection of acoustic timed events: A case study on clarinet note onsets. arXiv. [Online]. Available: http://arxiv.org/abs/1706.09556

[24] D. Murphy, "Tracking a conductor's baton," in *Proc. Danish Conf. Pattern Recognition and Image Analysis*, 2003, pp. 1–8.

[25] Á. Sarasúa and E. Guaus, "Beat tracking from conducting gestural data: A multi-subject study," in *Proc. ACM Int. Workshop Movement and Computing*, 2014, pp. 118–123.

[26] A. Bazzica, C. C. S. Liem, and A. Hanjalic, "Exploiting scene maps and spatial relationships in quasi-static scenes for video face clustering," *Image Vision Comput.*, vol. 57, pp. 25–43, Jan. 2017.

[27] R. B. Dannenberg and C. Raphael, "Music score alignment and computer accompaniment," *Commun. ACM*, vol. 49, no. 8, pp. 38–43, 2006.

[28] Z. Barzelay and Y. Y. Schechner, "Harmony in motion," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[29] M. Spiertz and V. Gnann, "Source-filter based clustering for monaural blind source separation," in *Proc. Int. Conf. Digital Audio Effects*, 2009, pp. 1–7.

[30] E. Acar, F. Hopfgartner, and S. Albayrak, "Fusion of learned multi-modal representations and dense trajectories for emotional analysis in videos," in *Proc. 13th Int. Workshop Content-Based Multimedia Indexing*, 2015, pp. 1–6.

[31] Y. Aytar, C. Vondrick, and A. Torralba, "SoundNet: Learning sound representations from unlabeled video," in *Proc. Advances Neural Information Processing*, 2016, pp. 1–9.

[32] L. Chen, S. Srivastava, Z. Duan, and C. Xu, "Deep cross-modal audio-visual generation," in *Proc. Thematic Workshops ACM Multimedia*, 2017, pp. 349–357.

[33] A. Masurelle, S. Essid, and G. Richard, "Multimodal classification of dance movements using body joint trajectories and step sounds," in *Proc. IEEE Int. Workshop Image Analysis Multimedia Interactive Services*, 2013, pp. 1–4.

[34] A. Drémeau and S. Essid, "Probabilistic dance performance alignment by fusion of multimodal features," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, 2013, pp. 3642–3646.

SP

Masataka Goto and Roger B. Dannenberg

Music Interfaces Based on Automatic Music Signal Analysis

New ways to create and listen to music

©ISTOCKPHOTO.COM/TRAFFIC_ANALYZER

Digital Object Identifier 10.1109/MSP.2018.2874360
Date of publication: 24 December 2018

Music analysis based on signal processing offers new ways of creating and listening to music. This article focuses on applications and interfaces that are enabled by advances in automatic music analysis. By using signal processing, some of these applications provide nonexperts the chance to enjoy music in their daily lives, while other applications apply signal processing to enhance professional music production and create new opportunities for composers and performers. Described in this article are the history and state of the art of music interfaces as well as its future directions that emphasize interactive music applications based on automatic music signal analysis.

Applications of music signal processing

Music understanding and music analysis is part of the human experience; whether the listener is a nonmusician casually enjoying music and tapping along with the beat or a professional making a formal analysis or transcription. As with many other human-oriented tasks, engineers and scientists have been inspired to formalize and automate aspects of human music perception such as identifying tempo, chords, melody, and repetition. Automatic music analysis capabilities have inspired research into new interfaces that take advantage of these novel possibilities. At the same time, applications have inspired new developments in signal processing for music listening and understanding. We have seen an explosion of new and exciting applications and interfaces. In this article, we explore some of the recent and emerging possibilities for music signal processing in music software.

Music signal processing goes by many names. Among these are *machine listening* (which also includes nonmusic signals) and *music understanding* (which emphasizes deep musical abstractions, e.g., patterns and structures, in contrast to shallower features such as pitch, loudness, and note-onset times). *Music content analysis* emphasizes the processing of signals (content) as opposed to metadata (often machine-readable text, such as file names, tags, web pages, or catalog entries). In this article, *music analysis* (i.e., music signal analysis) is used to refer to virtually any type of automatic (computational) music

 1053-5888/19©2019IEEE

recognition, detection, decomposition, classification, or understanding. Music analysis can identify music structure (chorus, section, and repetition) [1], [2], melody lines, chords, beat structure (beat and bar), drums [3], and so on.

Music interfaces may focus on a single musical piece or on collections of music, such as playlists, personal libraries or online catalogs. By means of a number of representative examples, this article explains how automatic music analysis can augment music interfaces; however, it considers only interfaces that focus on a single musical piece. The following three sections present different types of music interfaces based on playback navigation, customization, and music production, respectively.

Music interfaces for content-aware playback navigation

The traditional way of listening to music is hearing the piece from beginning to end. In the past, before it became possible to record music audio, one could only hear music at a live performance. When recording music became a reality, one could play a specific musical passage on demand, although manually controlling phonograph and tape players was often time consuming and inconvenient. The listener's ability to change the playback position almost instantly, with just the push of a button, only began recently with digital music on compact discs and the personal computer.

Interactive control of music playback is also a relatively recent development. Although digital music makes it easy for a listener to quickly jump from one song to another, only the fast-forward and rewind buttons can change the playback position within a musical piece. Even after media-player software on computers and portable digital audio players (e.g., MP3 players) appeared in the 1990s, music-listening interfaces remained unchanged except for a continuous playback slider. The total length of the slider corresponds to the length of a piece, and listeners can manipulate the slider to jump to any position in a song. However, listeners must use trial and error to search for a specific playback position.

Automatic music analysis based on signal processing addresses this problem by adding content-based navigation to conventional interfaces. Music interfaces that visualize music structure allow the listener to change the playback to logical positions. This approach is introduced in the "Music-Listening Interfaces Based on Automatic Music Structure Analysis" section of this article. Furthermore, when lyrics and music notation are aligned with audio signals, music interfaces display the lyrics or score in synchronization with the audio playback of a musical piece. As a result, new visual information about music content offers the listener a way to specify the playback position based on either lyrical or musical content. This approach is introduced in the "Music-Listening Interfaces Based on Automatic Music Synchronization" section.

Music-listening interfaces based on automatic music structure analysis

Automatic analysis of music structure improves conventional music-listening interfaces by using content-based playback navigation. The earliest of these works, introduced in 2003, is SmartMusicKIOSK [4], an intelligent music-listening station. In addition to the standard playback control buttons, SmartMusicKIOSK added a "jump to chorus" button and "jump to next/previous section" buttons, as shown in the lower window of Figure 1. SmartMusicKIOSK also extended the playback slider by visualizing the detected sections as the music structure. This visual representation, shown in the upper window of Figure 1, is called the *music map* and consists of chorus sections (the top orange row) and repeated sections (the five, lower green rows). In each row, colored sections indicate similar (repeated) sections. The music map helps a user decide where to jump to next, while each visualized section acts as a button to listen from the section's beginning.

The chorus and repeated sections are automatically determined by a signal processing method (RefraiD) [4] used for chorus-section detection, with a focus on popular music. First, a 12-dimensional feature vector, called a *chroma vector* [2], [4], is extracted from each frame of an input audio signal. Each element of the chroma vector corresponds to one of the 12 pitch classes (C, C#, D, D#, E, F, F#, G, G#, A, A#, and B) and the value of each element is the sum of magnitudes at frequencies of the pitch class over six octaves. In practice, this representation has been found to be robust with respect to changes in accompaniments, largely because its low dimensionality is enough to capture aspects of harmony and melody but not spectral details. The whole song is thus represented as a sequence of chroma vectors, i.e., a chromagram, and a pair of repeated sections is expected to have similar sequences of chroma vectors. RefraiD then calculates the similarity between all of the chroma vectors within the song and finds pairs of repeated sections whose similarity is higher than a certain threshold. This threshold is determined by an automatic threshold-selection method based on a discriminant criterion since the appropriate threshold varies for each song. To organize commonly repeated sections into groups and to identify both ends of each section, the pairs of repeated sections are integrated (grouped) by analyzing their relationships throughout the entire song. For example, three pairs of repeated sections, A and A', A' and A", and A and A", can be grouped on the basis of their relationships, even if one of the pairs is missing. Accordingly,

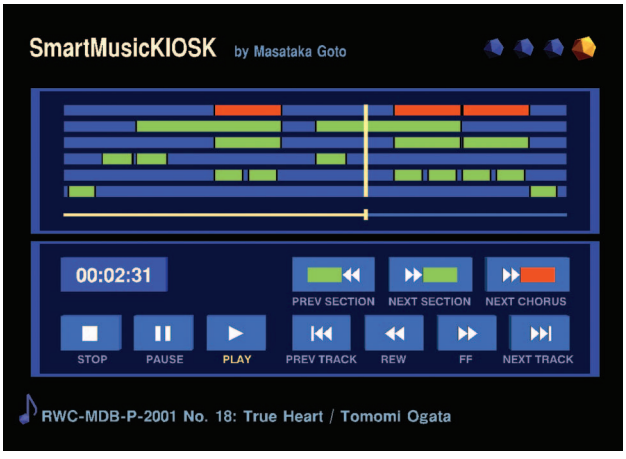


FIGURE 1. The SmartMusicKIOSK interface [4]. A user can actively listen to various parts of a song, guided by the visualized music structure ("music map") in the upper window.

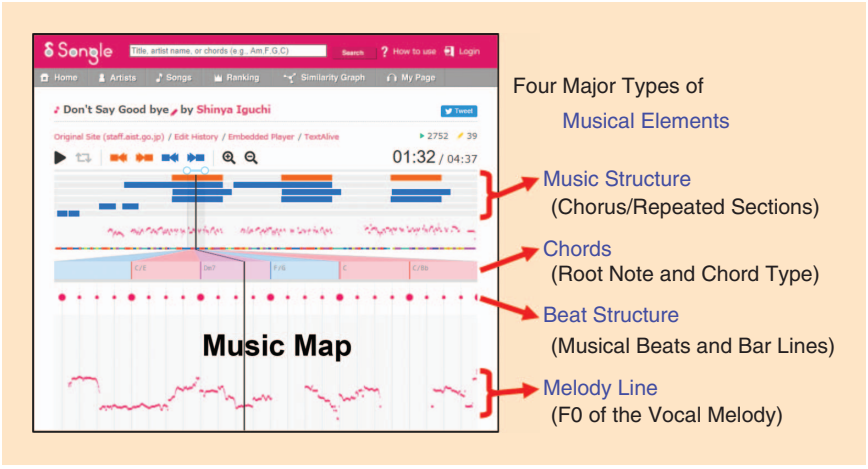


FIGURE 2. An interface of the Songle web service [7]. Songle automatically analyzes songs publicly available on the web and visualizes them with an informative "music map," including four types of musical elements. It is also equipped with the SmartMusicKIOSK interface.

RefraiD obtains several groups of repeated sections as intermediate results (the five lower green rows in Figure 1). Finally, it selects the chorus sections from them by evaluating the possibility of being chorus sections for each group. This possibility is greater when its sections are repeated more frequently with higher similarity, are longer and more appropriately positioned.

The RefraiD method was sufficiently useful for detecting and playing back the chorus sections (the final output) during trial listening without any interface or visualization, but it was even more useful for visualizing the chorus sections along the playback slider, as shown in the top orange row on the music map of the SmartMusicKIOSK in Figure 1. We also found this method to be informative when visualizing the repeated sections in the five lower green rows in Figure 1, even when those sections were not final but intermediate results.

SmartMusicKIOSK thus augments within-song browsing and trial listening. A user can skip sections of a song that are of no interest by interactively changing the playback position while viewing the music map. This is an example of active-music-listening interfaces [5], which allow a user to enjoy music in more active ways than conventional passive music playback. Moreover, this interface can draw attention to music structures that are unknown to users. By enabling the user to listen to the chorus sections of a song in succession, the user can more accurately understand how lyrics and the arrangement change for each repetition of the chorus (as a reflection of the musicians' intent). SmartMusicKIOSK is not only an active interface, but also it is considered an example of augmented music-understanding interfaces [6] that facilitate a deeper understanding of music.

The interface concept of SmartMusicKIOSK is universal and can be used with other methods for music-structure analysis [1], [2]. Its interface is also versatile enough to be used with music structures annotated by humans even though the manual-annotation process is not scalable to a large music collection.

In fact, the SmartMusicKIOSK interface has already been implemented and made available for more than 1,200,000 songs on Songle [7], a public web service that was launched in 2011 and available at http://songle.jp free of charge. Songle enriches the music experience by providing an active, augmented music-listening interface. Through signal processing, Songle estimates not only the music structure but also the beat structure (beat and downbeat), melody line, and chords of songs available on the web and visualizes all of them (Figure 2). Given the wide variety of music available, one drawback of automatic analysis is that errors are inevitable. To overcome this, Songle provides a crowdsourcing interface that encourages users to correct errors in the estimated results by selecting from a list of alternatives or by providing an alternative annotation. The supplied corrections are then shared and used to immediately improve the user experience. Since Songle also provides an application programming interface (commonly known as API), the results of music analysis and human annotation can be used to develop music-driven applications such as robot dancing, stage lighting, and computer animation [8]. Songle therefore serves as an open showcase that demonstrates how people can benefit from signal processing-based music analysis and how interfaces can contribute to better music-listening experiences.

As previously mentioned in this section, a visual representation of music analysis results is key to changing traditional music interfaces into advanced interfaces with content-aware playback navigation. Another example is Dunya, a web-based application [9] that visualizes the pitch [fundamental frequency (F0)] contour of the melody line and its histogram as well as the waveform and spectrogram, with a focus on Carnatic music. By showing related recordings based on culturally specific similarity, it also allows a user to discover musically relevant relationships between different pieces.

Music-listening interfaces based on automatic music synchronization

Automatic synchronization of different representations of music, such as audio signals, lyrics, Musical Instrument Digital Interface (MIDI), and music scores, may also improve conventional interfaces with multifaceted, content-based music navigation and browsing. SyncPlayer [10], which is based on semiautomatic music synchronization procedures, is an early example of a music interface that provides users the opportunity to discover and explore multimodal representations of music. SyncPlayer's alignments between various music representations are computed in a preprocessing step and stored using suitable data structures. During the playback of a musical piece, it synchronously displays lyrics and a MIDI-based piano-roll view along with audio waveform and spectrogram. Time-aligned lyrics are shown in a karaoke-like display as the phrase currently being sung is highlighted. SyncPlayer has a lyrics search function that enables a user to submit a text-based query for lyrics that finds the corresponding

audio. Time-aligned MIDI is generated by an automatic score-to-audio synchronization (alignment) method [11] and visualized in a piano-roll display. SyncPlayer first detects note onsets in the audio signal of a musical piece to obtain a score-like representation. This representation is then aligned with musical notes of a MIDI file by using a dynamic time-warping (DTW) algorithm.

This concept is further extended to the Score Viewer and Interpretation Switcher interfaces [11], [12], as shown in Figure 3. Score Viewer displays a time-aligned music score (scanned sheet music) that highlights the current bar. With a focus on Western classical music, spatial regions of the scanned sheet music are automatically synchronized with musically corresponding temporal sections within the audio recording. Score Viewer first extracts chromagrams (temporal sequences of the chroma vectors, as described in the previous section) from the results of optical music recognition of the sheet music. It then uses DTW to align those representations with chromagrams of the audio recordings. In classical music, recordings of different performers playing the same piece are often available. Interpretation Switcher automatically synchronizes those recordings and allows a user to seamlessly switch from one performance to another while continuing playback.

Score Viewer does not synchronize real-time audio input with the sheet music. To enrich the audience's experience of classical music concerts, however, real-time input is necessary. Another project, EU FP7 PHENICX (http://phenicx.upf.edu), developed and used an automatic, real-time audio-to-sheet-music synchronization method to track a live public performance of the Royal Concertgebouw Orchestra. During the performance, time-aligned sheet music was displayed for an audience in a concert hall in Amsterdam [13].

While Interpretation Switcher synchronizes different performances and allows comparisons by ear, a web-based interface [14] facilitates a more objective comparison of features of loudness (using dynagrams) and tempo (using tempograms) in two performances. Music performances can also be shown as two-dimensional (2-D) tempo-loudness trajectories called *performance worms*. The alignment between the waveform displays of two performances is visualized as line patterns connecting the corresponding bar lines. This visualization also includes an interactive musical-score display based on automatic alignment.

LyricSynchronizer [15], another interface that synchronizes symbolic text displays with music playback, is lyrics oriented and displays scrolling time-aligned lyrics by using an automatic lyrics-to-audio synchronization method. Because lyrics are automatically highlighted, a user can easily follow the current playback position. Additionally, the user can click on a word that interests them and listen to a song from that word forward.

Music interfaces for customization and personalization

Traditional music players often include graphic equalizers or tone controls for bass and treble. Listeners can therefore customize/personalize music playback in a simple way by adjusting the overall frequency response. However, listeners cannot change the volume or timbre of each individual instrument in existing recordings unless individual tracks, called *stems* (separate recordings before mixing, corresponding to different instruments), are provided.

Sound source separation of musical audio signals can overcome this limitation and enable new types of music interfaces that allow a listener to customize music by changing the volume or timbre of instrument sounds in existing music recordings or by altering notes and styles. These kinds of creative customization represent music personalization for a user.

Music-customization interfaces based on sound source separation

Drumix [16] is an early example of a music-customization interface that allows a user to edit the drum part of an existing recording during music playback the same as if another drummer was performing different drum patterns. With this interface, a user can change the volume or timbre of the sounds of

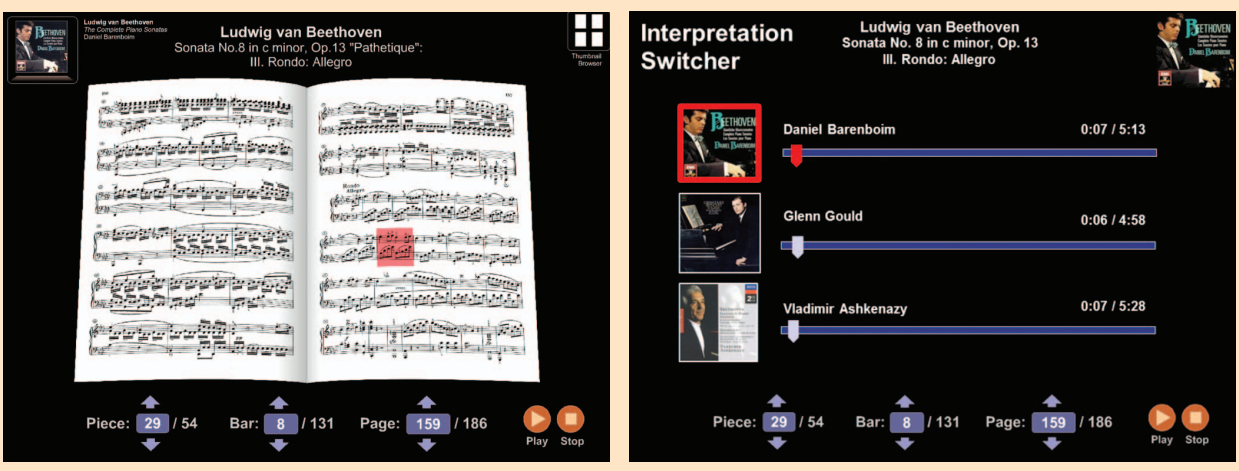


FIGURE 3. (a) The Score Viewer and (b) Interpretation Switcher interfaces. The Score Viewer displays interactive scanned sheet music synchronized with music playback. The Interpretation Switcher enables a user to seamlessly switch to different recordings of the same piece of music [11].

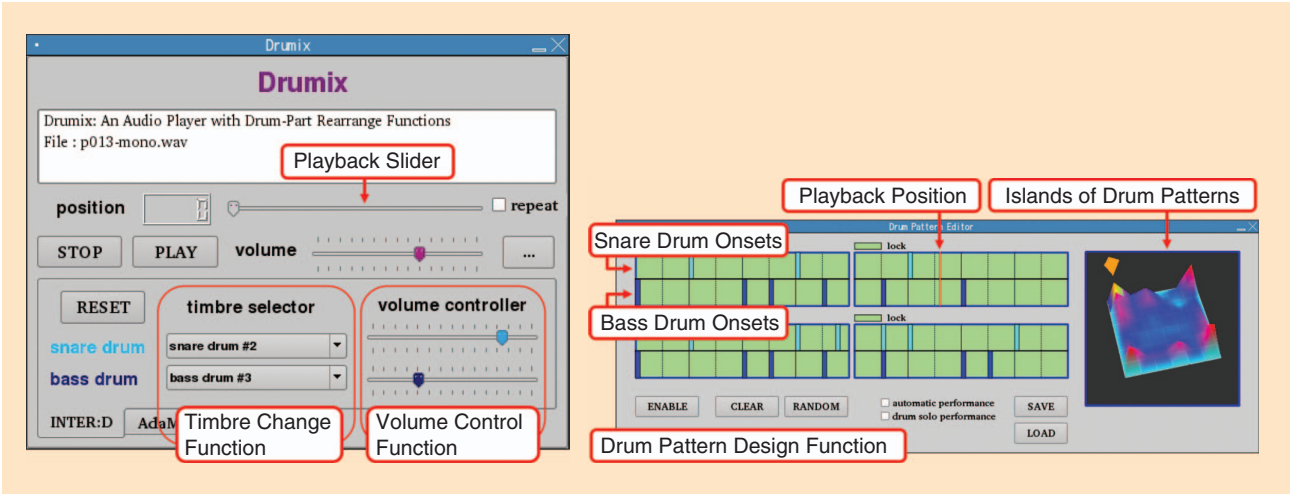


FIGURE 4. The Drumix interface. A user can actively change the volume or timbre of drum sounds and rearrange drum patterns during music playback [16].

bass and snare drums, as shown in the left window of Figure 4. In the right window, the user can also rearrange drum patterns of bass and snare drums by dragging a pattern from "islands of drum patterns" in which each island represents a different cluster of similar drum patterns. The larger an island is, the more popular its drum patterns are in a corpus of drum patterns. A user who is usually unaware of the drum pattern or timbre of drum sounds can use Drumix to edit them, which helps a user develop an appreciation of the musical choices of performers and producers. Drumix thus enhances the ability of the user to understand the role of drums in songs.

The onset times and spectrograms of drum sounds are automatically estimated by AdaMast, a drum-sound-recognition method [16]. It first prepares a seed template that is the spectrogram of a typical bass or snare drum sound and then detects onset times of drum-sound candidates in an existing music recording (polyphonic sound mixture) by using a template-matching technique. Since the seed template is different from the actual drum sound in the recording, AdaMast uses the median of the detected spectrograms to update its template. It then uses the updated template to repeat this iterative-matching and adaptation process. After several iterations, AdaMast obtains the template (i.e., spectrogram) of the actual drum sound, which can then be used to separate, change, remove, and add drum sounds. To deal with drum patterns in units of bar (measure), Drumix also uses a beat-tracking method.

The concept of Drumix can be used not only with other drum-sound recognition methods [3] but also with any instrument or voice if sound source separation for them can be achieved. Given polyphonic sound mixtures of popular music, however, it is well known to be extremely difficult to decompose them into all of the original stems because musical audio signals often combine more than ten simultaneous sounds with overlapping frequency, content, and reverberation. An ongoing, unsolved challenge for signal processing researchers is to achieve better source separation [17] and enable higher-quality audio manipulation of arbitrary music mixtures [11], [16].

Despite these challenges, this concept has been further investigated by different research groups. For example, a music-manipulation method [18] can change the timbre and phrases of a pitched instrument part. Because it is difficult to segregate an arbitrary instrument part from polyphonic sound mixtures, this method is based on score-informed source separation [19] that leverages a musical score of the target part to help isolate its sound and change its timbre. This method also changes the original phrase into a phrase specified by another score provided by the user.

By decomposing an existing recording of the input song into the vocal track (singing voice) and the karaoke track (the rest of the input sound mixture), a vocal-editing interface was proposed in [20]. This interface allows a user to manipulate vocal F0 by adding vocal expressions (e.g., vibrato and glissando) and changing the melodic contour (i.e., the pitches of musical notes).

Even if users are not musicians, music signal processing enables easy-to-use customization of existing music that allows for enjoying music in active ways and facilitates a deeper understanding of music. The interfaces discussed in this section are considered good examples of active music-listening interfaces [5] and augmented music-understanding interfaces [6].

Music interfaces for production and performance

Music analysis presents new capabilities for computer-assisted music creation and performance. In the "Music-Production Interfaces Based on Score-to-Audio Alignment" section, we examine how music analysis enables audio-editing software to "adjust" music recordings automatically based on models of pitch and rhythm, perhaps with guidance from machine-readable music notation, envisioned as early as 1982 [21]. In the "Real-Time Signal Analysis in Interactive Music Performance" section, we see examples of how new modes of music performance are enabled by real-time machine listening.

Music-production interfaces based on score-to-audio alignment

Audio editors typically use visual representations of waveforms and spectrograms, but these are difficult to comprehend and navigate. As an alternative, music-editing software can display symbolic representations of music alongside waveforms,

as shown in the Figure 5 mockup. With this style of interface, the music audio is actively labeled with a human-readable notation, which facilitates search and navigation. Constructing such an interface requires some form of symbolic notation and a method to align audio to it.

Obtaining symbolic notation directly from audio requires automatic music transcription [17]. Since this is extremely difficult to achieve (especially for recordings of multiple instruments), full, automatic transcription from arbitrary music signals is not likely to be practical for building notation-based interfaces in the near future. On the other hand, since many composers already use music-notation software, their music exists in a machine-readable form. Rather than transcribing audio, interfaces can align existing notation to music audio. For example, an experimental version of the Audacity (https://www.audacityteam.org/) audio editor can display MIDI files in a piano-roll view that is automatically aligned to a corresponding audio track.

Score-to-audio alignment (synchronization) generally works by converting music to feature sequences, such as the chromagram described in the "Music-Listening Interfaces Based on Automatic Music Structure Analysis" section, and using DTW or hidden Markov models to align them [22]. In this audio-editing system, DTW was used, as in SyncPlayer, described in the "Music-Listening Interfaces Based on Automatic Music Synchronization" section.

Navigating a digital audio track is a common facet of audio editing. Digital-editing software allows recording engineers and music producers to apply advanced signal processing techniques to make timing, pitch, and loudness adjustments on a note-by-note basis. Sophisticated interfaces have evolved to support this work, but actual edits are nearly always specified manually. One exception is the Antares Audio Technologies Auto-Tune product (https://www.antarestech.com/), which has become a standard tool in music production for correcting off-key pitch. Auto-Tune works mainly by shifting pitch to the nearest musical scale degree as specified by the user, so it can automatically calculate a target pitch and apply pitch corrections. Apple's Flex Time (https://support.apple.com/kb/PH13083) processing interface enables automatic timing adjustments in one track to be guided by audio transients in another track, which is much easier than manually performing "microsurgery" to achieve the same result.

Rather than simply quantizing to pitches or beats, score-to-audio alignment provides an audio editor the ability to automatically determine the intended timing, pitch, and loudness of every note by reading the score, compare that to every performed note based on the score-to-audio alignment, and then use signal processing techniques to adjust audio recordings [23]. In this article, multitrack audio is assumed, and each instrument is recorded on a separate track. Each track is then aligned separately with music notation for a specific instrument. Since monophonic instruments are assumed, alignment is based on DTW to match pitch sequences obtained from onset-detection-based note segmentation and F0 estimation. Next, the interface produces a list of edits, applying small pitch adjustments (through resampling) and timing adjustments (by cutting, splicing and cross-fading) on a note-by-note basis. Finally, tracks are mixed to balance the average root mean square. This article shows that "intelligent" editors can automate and simplify many routine edits made in music production.

Of course, forcing audio to meet precise specifications can remove important musical nuance. Rather than "fixing" everything, an audio editor might present an interface to act as a spell checker, in which the human engineer decides to accept or reject each of the computer's suggested changes. We see this as a promising direction for future audio-editing interfaces and a logical extension of some of the automated tools and interfaces that exist for commercial editors today.

Beyond editing to correct mistakes or polish recorded performances, music producers use equalization, gain control, reverberation, stereo placement, and many other techniques to creatively enhance their work. There is growing interest in computational music production, and there are many automated mastering services online, already claiming millions of mastering sessions in total. As signal processing becomes more complex [24], interfaces are needed to operate at higher levels of abstraction. A machine-learning approach [25], for instance, was proposed to describe filter-transfer functions with user-oriented terms such as *warm* or *bright*.

Real-time signal analysis in interactive music performance

Some of the earliest work in music audio analysis was motivated by composers and performers exploring real-time sensing and computation to create interactive musical works. These works often used F0 estimation to obtain pitches from monophonic instruments because the simple hardware needed for this purpose was readily available. For example, Voyager [26] is a pioneering interactive system that uses note-level analysis. Monophonic audio input is analyzed for pitch (F0) and dynamic (signal-amplitude) information, which is processed to form pitch histograms, note density, and other features. These, in turn, influence music-generation algorithms that control a music synthesizer, thereby producing something akin to a collectively improvising ensemble. In [26], Lewis describes his system in terms of improvisation: "Improvisation must be

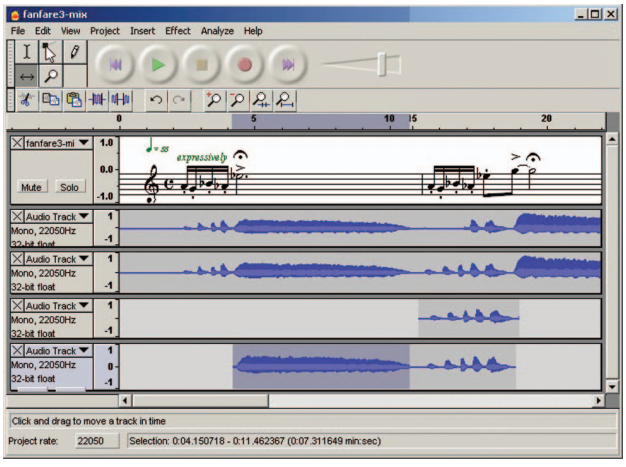


FIGURE 5. The concept design for an intelligent editor that stretches and aligns music notations and audio, which enables users to quickly navigate to, select, and splice together the best "takes" from a recording session [23].

open—i.e., open to input, open to contingency—a real-time and (often enough) a real-world mode of production." In this sense, music analysis is critical to his work.

Live electronic compositions are not as well known as commercial popular music and Western classical music, but there are many international festivals that feature interactive computer music, and music analysis is playing an increasingly important role by enabling composers to incorporate more sophisticated "listening" into their works. These works illustrate and explore the possibilities of nongraphical user interfaces. One example is CataRT [27], which slices input sound into short grains and organizes those grains according to composer-/performer-selected features. These features can have high dimension, drawing from spectral, perceptual, and harmonic descriptors; or, when applied another way, grains are projected onto a 2-D display that can be navigated with mouse or trackpad input. The performer then uses this control space to create sonic textures by summing from a few to thousands of grains per second. The navigation can also be controlled by features obtained from a musician's audio in real time, offering a sort of analysis/resynthesis system in which the representation is a highly abstract feature space.

Another interesting development is that of Wekinator [28] (http://www.wekinator.org/), a machine-learning software package developed especially for musicians and interactive music performances. Wekinator uses a visual interface to simplify the capture of input-to-output examples that are used to train the response of interactive systems. A typical application is mapping audio features or even a simple fast Fourier transform to the multidimensional control space of a music synthesizer or music-composition algorithm. A variety of classifiers are then used for supervised training of the input-to-output mapping.

Yet another class of interactive music performance systems is occasioned by computer accompaniment, which models the familiar scenario of a soloist and accompanist, such as a flute accompanied by piano, except that the accompanist's part is played by a computer system that "listens" to the soloist, follows along in a machine-readable score, and synchronizes the accompaniment part to the live soloist [22], [29]. In this model, both the solo and accompaniment parts are composed and played note for note; therefore, the task performed by the computer is primarily that of synchronization. Computer accompaniment systems use various algorithms for "score following," including DTW and hidden/semihidden Markov models. The signal processing challenges associated with these systems include dealing with the presence of accompaniment audio in a live performance (even with a microphone in close to the soloist). These systems also implement various strategies for musically adjusting tempo to maintain synchronization. In addition to performance, score-following technology allows for rich-performance interfaces that feature automatic music page turning and automatically generated feedback to student musicians.

Discussion and future directions

Music signal processing continues to encourage exciting new ways of working with music. From the listener's perspective, we have seen how music interfaces can help to visualize music information and assist users in music playback, navigation, and multifaceted browsing. For creative amateurs, interfaces can harness sophisticated signal processing for customization or personalization of music, while for professionals, applications automate high-level editing and production tasks, allowing composers and performers to use music analysis to creatively control music generation and create new "instruments."

In the future, advances in automatic music analysis will inspire and provide more advanced music interfaces. Conversely, the invention of novel music interfaces will require more advanced signal processing for music analysis. This interdependency drives the development and improvement of both novel music interfaces and state-of-the-art signal processing methods. Although significant research progress has been made in the past 30 years, music-analysis technologies have not yet matured and remain far from human levels of music understanding and analysis. Further progress is important for advanced music interfaces. For example, active music-listening and augmented music-understanding interfaces may benefit from advances in automatic music analysis. As discussed in the "Music Interfaces for Customization and Personalization" section, music-customization interfaces would benefit from better source separation and audio-manipulation techniques; and music-production interfaces would benefit from automatic music transcription of arbitrary polyphonic sound mixtures.

In addition to the efforts of advancing music signal processing, another important future direction is to research and develop a variety of music interfaces that involve human intelligence (i.e., human in the loop). For example, the Songle service discussed in the "Music-Listening Interfaces Based on Automatic Music Structure Analysis" section features an error-correction interface. Tarsos [30], a system used for pitch analysis in Western and non-Western music, employs F0-estimation algorithms such as the standard YIN and McLeod Pitch Method (https://github.com/JorenSix/TarsosDSP) but offers a graphical interface to guide the analysis. Similarly, the Interactive Source Separation Editor (ISSE) [31] (http://isse.sourceforge.net/) uses a sophisticated interface for source separation based on probabilistic latent-component analysis, including machine learning from manual corrections. Interfaces that integrate human control and knowledge with automatic music analysis are advancing rapidly, and we expect to see increasingly sophisticated interaction in future intelligent systems used for music editing and production.

Another theme in emerging research is a consistent drive toward more active listening. If computers bring interactive and "smart" capabilities, and if music is now mediated by computers, it seems only natural to pursue greater interactivity and intelligence in interfaces for music. We see this trend in many experimental interfaces for music listening, and there are hundreds of interactive music games, tablet-based electronic instruments, and composing programs. More active music-listening interfaces such as SmartMusicKIOSK, Songle, and Drumix have the potential to blur the boundaries between music listening, music creation, games, and entertainment. Perhaps the extreme form of active listening is music performance, where interactive software such as SmartMusic (https://www.smartmusic.com/) provides always-available music instruction and accompaniment.

We have seen a revolution in music storage, processing, and distribution brought about by digital signal processing. The digitization of music has progressed from an initial, quantitative phase in which costs came down and the number of recordings in music collections went up. Today, we are in a second, qualitative phase that is changing the nature of musical experiences. We believe this phase will reveal the true value of digitization. The key to change is automatic music analysis, which enables music interfaces to move from just storing music to offering high-level musical interactions. Music interfaces based on music analysis produce qualitative changes in music experiences for professional and casual listeners alike.

Authors

Masataka Goto (m.goto@aist.go.jp) received his B.E. and Ph.D. degrees in engineering from Waseda University, Tokyo, Japan, in 1993 and 1998, respectively. He is currently a prime senior researcher at the National Institute of Advanced Industrial Science and Technology (AIST). In 1992, he worked on automatic music understanding based on signal processing and has since contributed to the research and development of music technologies and music interfaces based on those technologies. He has published more than 300 papers in refereed journals and international conferences and has received 47 awards, including several best paper awards, best presentation awards, the Tenth Japan Academy Medal, and the Tenth Japan Society for the Promotion of Science Prize.

Roger B. Dannenberg (rbd@cs.cmu.edu) received his B.S. degree in electrical engineering from Rice University in 1977 and his Ph.D. degree in computer science in 1982 from Carnegie Mellon University, Pittsburgh, Pennsylvania, where he is currently a professor of computer science, art, and music. His pioneering work in computer accompaniment led to the awarding of three patents and the advent of the SmartMusic system now used by tens of thousands of music students. He is the chief science officer for Music Prodigy and a cocreator of Audacity, an open-source audio editor that has been downloaded more than 300 million times. As a trumpet player, he is active in performing jazz, classical music, and new works. His compositions include many interactive computer works, and, in 2017, he premiered La Mare dels Peixos, an opera cocomposed with Jorge Sastre.

References

[1] R. Dannenberg and M. Goto, "Music structure analysis from acoustic signals," in *Handbook of Signal Processing in Acoustics*, D. Havelock, S. Kuwano, and M. Vorländer, Eds. Berlin: Springer-Verlag, 2009, pp. 305–331.

[2] J. Paulus, M. Müller, and A. Klapuri, "State of the art report: Audio-based music structure analysis," in *Proc. 11th Int. Society Music Information Retrieval Conf. (ISMIR)*, 2010, pp. 625–636.

[3] C. W. Wu, C. Dittmar, C. Southall, R. Vogl, G. Widmer, J. Hockman, M. Müller, and A. Lerch, "A review of automatic drum transcription," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 26, no. 9, pp. 1457–1483, 2018.

[4] M. Goto, "A chorus-section detection method for musical audio signals and its application to a music listening station," *IEEE Trans. Audio, Speech, Language Process.*, vol. 14, no. 5, pp. 1783–1794, 2006.

[5] M. Goto, "Active music listening interfaces based on signal processing," in *Proc. IEEE Int. Conf. Acoustics Speech and Signal Processing (ICASSP)*, 2007, pp. 1441–1444.

[6] M. Goto, "Frontiers of music information research based on signal processing," in *Proc. 12th IEEE Int. Conf. Signal Processing*, 2014, pp. 7–14.

[7] M. Goto, K. Yoshii, H. Fujihara, M. Mauch, and T. Nakano, "Songle: A web service for active music listening improved by user contributions," in *Proc. 12th Int. Society Music Information Retrieval Conf. (ISMIR)*, 2011, pp. 311–316.

[8] J. Kato, M. Ogata, T. Inoue, and M. Goto, "Songle Sync: A large-scale web-based platform for controlling various devices in synchronization with music," in *Proc. ACM Multimedia*, 2018, pp.1697–1705.

[9] A. Porter, M. Sordo, and X. Serra, "Dunya: A system to browse audio music collections exploiting cultural context," in *Proc. 14th Int. Society Music Information Retrieval Conf. (ISMIR)*, 2013, pp. 101–106.

[10] F. Kurth, M. Müller, D. Damm, C. Fremerey, A. Ribbrock, and M. Clausen, "SyncPlayer—an advanced system for multimodal music access," in *Proc. 6th Int. Society Music Information Retrieval Conf. (ISMIR)*, 2005, pp. 381–388.

[11] M. Müller, "*Fundamentals of Music Processing—Audio, Analysis, Algorithms, Applications*," Berlin: Springer-Verlag, 2015.

[12] D. Damm, C. Fremerey, V. Thomas, M. Clausen, F. Kurth, and M. Müller, "A digital library framework for heterogeneous music collections: from document acquisition to cross-modal interaction," *Int. J. Digit. Libraries*, vol. 12, no. 12, pp. 1726–1737, 2014.

[13] A. Arzt, H. Frostel, T. Gadermaier, M. Gasser, M. Grachten, and G. Widmer, "Artificial intelligence in the concertgebouw," in *Proc. 24th Int. Joint Conf. Artificial Intelligence (IJCAI)*, 2015, pp. 2424–2430.

[14] M. Gasser, A. Arzt, T. Gadermaier, M. Grachten, and G. Widmer, "Classical music on the web—user interfaces and data representations," in *Proc. 16th Int. Society Music Information Retrieval Conf. (ISMIR)*, 2015, pp. 571–577.

[15] H. Fujihara, M. Goto, J. Ogata, and H. G. Okuno, "LyricSynchronizer: Automatic synchronization system between musical audio signals and lyrics," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 6, pp. 1252–1261, 2011.

[16] K. Yoshii, M. Goto, K. Komatani, T. Ogata, and H. Okuno, "Drumix: An audio player with real-time drum-part rearrangement functions for active music listening," *Info. Proc. Soc. Japan (IPSJ) Journal*, vol. 48, no. 3, pp. 1229–1239, 2007.

[17] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri, "Automatic music transcription: Challenges and future directions," *J. Intell. Inform. Syst.*, vol. 41, no. 3, pp. 407–434, 2013.

[18] N. Yasuraoka, T. Abe, K. Itoyama, T. Takahashi, T. Ogata, and H. G. Okuno, "Changing timbre and phrase in existing musical performances as you like: Manipulations of single part using harmonic and inharmonic models," in *Proc. ACM Multimedia*, 2009, pp. 203–212.

[19] S. Ewert, B. Pardo, M. Müller, and M. D. Plumbley, "Score-informed source separation for musical audio recordings: An overview," *IEEE Signal Process. Mag.*, vol. 31, no. 3, pp. 116–124, 2014.

[20] Y. Ikemiya, K. Yoshii, and K. Itoyama, "Singing voice analysis and editing based on mutually dependent F0 estimation and source separation," in *Proc. IEEE Int. Conf. Acoustics Speech and Signal Processing (ICASSP)*, 2015, pp. 574–578.

[21] C. Chafe, B. Mont-Reynaud, and L. Rush, "Toward an intelligent editor of digital audio: Recognition of musical constructs," *Comput. Music J.*, vol. 6, no. 1, pp. 30–41, 1982.

[22] R. Dannenberg and C. Raphael, "Music score alignment and computer accompaniment," *Commun. ACM*, vol. 49, no. 8, pp. 38–43, 2006.

[23] R. Dannenberg and N. Hu. "Polyphonic audio matching for score following and intelligent audio editors," in *Proc. Int. Computer Music Conf. (ICMC)*, 2003, pp. 27–34.

[24] E. P. Gonzalez and J. D. Reiss, "Automatic equalization of multi-channel audio using cross-adaptive methods," in *Proc. 127th Audio Engineering Society Conv. (AES)*, 2009.

[25] B. Pardo, D. Little, and D. Gergle, "Building a personalized audio equalizer interface with transfer learning and active learning," in *Proc. 2nd Int. ACM Workshop Music Information Retrieval with User-Centered and Multimodal Strategies (MIRUM)*, 2012, pp. 13–18.

[26] G. Lewis, "Too many notes: Computers, complexity and culture in Voyager," *Leonardo Music J.*, vol. 10, pp. 33–39, Dec. 2000.

[27] D. Schwarz, "Corpus-based concatenative synthesis: Assembling sounds by content-based selection of units from large sound databases," *IEEE Signal Process. Mag.*, vol. 24, no. 2, pp. 92–104, 2007.

[28] R. Fiebrink, "Real-time human interaction with supervised learning algorithms for music composition and performance," Ph.D. dissertation, Comput. Sci. Dept. Princeton Univ., New Jersey, 2011.

[29] A. Maezawa and K. Yamamoto, "MuEns: A multimodal human-machine music ensemble for live concert performance," in *Proc. ACM Conf. Human Factors in Computing Systems (CHI)*, 2017, pp. 4290–4301.

[30] J. Six, O. Cornelis, and M. Leman, "Tarsos, a modular platform for precise pitch analysis of western and non-western music," *J. New Music Res.*, vol. 42, no. 2, pp. 113–129, 2013.

[31] N. Bryan, G. Mysore, and G. Wang, "ISSE: An interactive source separation editor," in *Proc. ACM Conf. Human Factors in Computing Systems (CHI)*, 2014, pp. 257–266.

SP

Eric J. Humphrey, Sravana Reddy, Prem Seetharaman, Aparna Kumar, Rachel M. Bittner, Andrew Demetriou, Sankalp Gulati, Andreas Jansson, Tristan Jehan, Bernhard Lehner, Anna Kruspe, and Luwei Yang

An Introduction to Signal Processing for Singing-Voice Analysis

High notes in the effort to automate the understanding of vocals in music

©ISTOCKPHOTO.COM/TRAFFIC_ANALYZER

Digital Object Identifier 10.1109/MSP.2018.2875133
Date of publication: 24 December 2018

Humans have devised a vast array of musical instruments, but the most prevalent instrument remains the human voice. Thus, techniques for applying audio signal processing methods to the singing voice are receiving much attention as the world continues to move toward music-streaming services and as researchers seek to unlock the deep content understanding necessary to enable personalized listening experiences on a large scale. This article provides an introduction to the topic of singing-voice analysis. It surveys the foundations and state of the art in computational modeling across three main categories of singing: general vocalizations, the musical function of voice, and the singing of lyrics. We aim to establish a starting point for practitioners new to this field and frame near-field opportunities and challenges on the horizon.

Power of the human voice

The human voice dominates nearly all music cultures. The voice, through singing, can function as a musical instrument and at the same time convey semantic meaning. Theory from the field of psychology suggests that people generally find the human voice especially salient and powerful and that the human voice is a meaningful factor, perhaps the most meaningful factor, in affecting our music-listening behavior. Research has suggested that music exists because of the complex system that enables humans to communicate, interpret, and feel emotions via vocal sounds [1]. Given such strong anthropological links between music and voice, it is unsurprising that singing plays a prominent role in modern music culture; karaoke, for example, is a billion-dollar worldwide industry.

Thus, digital signal processing research has long focused on methods and techniques for modeling the human voice. Early progress in efforts to encode and transmit speech for telecommunication systems [2] paved the way for singing-information processing, the study of signal processing techniques on the human voice in musical contexts [3]. Singing information processing can be represented as a cyclic system where, under ideal conditions, an audio signal is transformed, via analysis, into high-level descriptors or symbols, such as pitch or lyrics; rich symbolic information can then be transformed, via synthesis, into audio signals of singing; and, falling between analysis

and synthesis, effects can be applied to either audio or symbolic information by manipulating intermediary representations between the two domains. A popular vocal effect, for example, is that of pitch correction ("autotune"), where a vocal audio signal is analyzed, the estimated pitch over time is quantized to a given key, and the voice signal is resynthesized.

Starting around the end of the 20th century, the field of music information retrieval (MIR) has developed techniques and methods for various applications of singing-information processing. While many researchers have made contributions to this field, the work of two groups in particular stands out: the Music Technology Group (MTG) at Universitat Pompeu Fabra in Spain, under the direction of Xavier Serra, and the National Institute of Advanced Industrial Science and Technology (AIST) of Japan, under the direction of Masataka Goto. Researchers at the MTG have a long history of advancing the state of the art in singing-voice synthesis, resulting in both commercial products and published studies [4]. Meanwhile, the efforts of AIST are noteworthy for their novelty and breadth, spanning use cases in music production, education, and consumption [5]. One of the more comprehensive reviews of singing-information processing research to date appeared as a tutorial at the 16th International Society for Music Information Retrieval Conference in Málaga, Spain, in 2015 [43]. This tutorial provided an exhaustive list of methods, data sets, tools, and applications, including real-world examples of different singing styles.

Given the pervasiveness of voice in music, demand is keen for improvements in singing-information processing. Now that music-streaming services are the de facto way for people across the world to not only listen to music but also to discover new songs, personalized recommendation is a very promising application. A recent study confirms that music-streaming listeners are especially attuned to the perception of singing [6]. Of several hundred users surveyed (1.2% response rate), listeners indicated that vocals (29.7%), lyrics (55.6%), or both (16.1%) are among the salient attributes they notice in music. Additionally, the four most important "broad" content categories were found to be emotion/mood, voice, lyrics, and beat/rhythm. Meanwhile, listeners said and the seven most important vocal semantic categories are skill, "vocal fit" (to the music), lyricism, the meaning of lyrics, authenticity, uniqueness, and vocal emotion. High-level content attributes like these can be combined with traditional recommendation approaches (e.g., collaborative filtering, factorization machines, or deep networks) to reach a level of nuance that would be difficult to achieve with user-interaction signals alone (e.g., explicit feedback or curated playlists). Furthermore, content-informed methods are necessary for cold-start recommendation (i.e., discovery), an inherent problem for algorithms that rely solely on user signals. Though expert-backed approaches, like the one taken by the Music Genome Project (https://www.pandora.com/about/mgp), have made considerable progress over the last decade, the demand for further improvements is rising along with the seemingly limitless growth in the amount of digital music content and in the number of listeners. Only through automation of music-content description will it be possible to match so much content to so many listeners.

In this article, we focus specifically on the challenge of automatically characterizing attributes of the voice in music as a self-contained and independently testable problem. A holistic view of singing analysis is diagramed in Figure 1, which provides the basic structure of this article. We first outline the fundamentals of the human voice and singing, provide notation to represent singing in recorded music, and introduce common computational models of the voice. Different applications of singing analysis are then grouped by their relationships to music and natural language: vocalized sound in general, voice in musical contexts, and the singing of lyrics. Having outlined approaches to automatically characterizing the voice, we offer

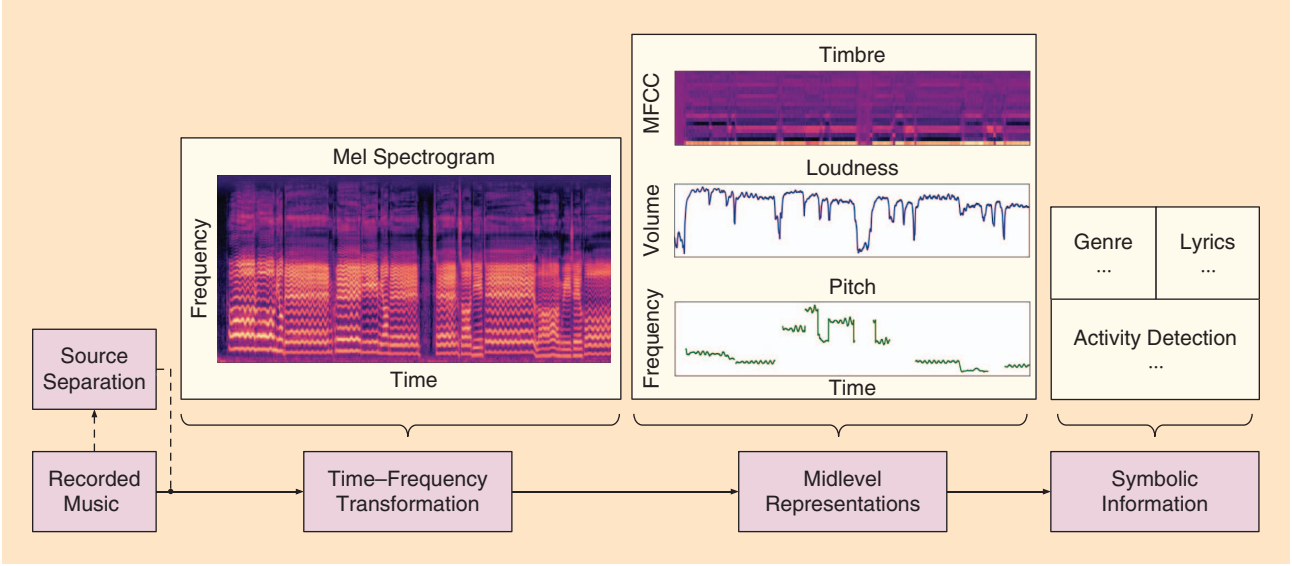


FIGURE 1. A high-level overview of singing-analysis systems: recorded music is optionally preprocessed by a source-separation algorithm before undergoing feature transformations to extract descriptors or symbolic information. Depending on the task, machine learning may be applied between these operations or in an "end-to-end" fashion.

some concrete next steps in this research lineage, and we conclude with an assessment of potential challenges and opportunities facing singing analysis research.

Fundamentals of singing

The compression and expansion, or rarefaction, of air molecules causes a propagation of oscillations known as an *acoustic wave*. These fluctuations can be expressed as a combination of pure sinusoids such that frequencies in the range of 20 to 20,000 Hz are perceived by humans as sound. Classified as an aerophone in the Hornbostel–Sachs taxonomy, the human voice produces sound by moving air, forced from the diaphragm, across the vocal cords, causing them to resonate. This harmonic sound is then shaped via the mouth, with varieties of sibilance added from the teeth, lips, and tongue. The physiological formation of different sounds in the vocal cords and glottis is known as *phonation*, which is how humans convey different phonemes in speech and different voicing styles in singing.

Computational approaches to modeling the human voice fall into either physical or spectral categories [4]. Much is understood about the human vocal organs, and so physical models can be used to demonstrate how the voice produces sound. Source–filter theory, an approach that applies to a variety of string and wind instruments as well, represents sound production as a two-stage process, where a source signal is convolved with the impulse response of a filter. The source can be either voiced (e.g., periodic vowels like *[a]*) or unvoiced (e.g., aperiodic fricatives like *[f]*). In the case of a voiced source signal, the vocal folds vibrate and generate a signal similar to that of a vibrating string. The pitch or fundamental frequency (f_0) of a voiced sound is determined by the rate at which the vocal folds vibrate, and subsequent peaks created at multiples of f_0 are called *harmonics*. Higher frequencies are damped, sloping downwards at approximately −12 dB per octave. In the case of an unvoiced source signal, turbulent noise is created with the teeth, lips, tongue, and, in case of whispering, the glottis. The vocal tract, a tube-shaped acoustic resonator that acts as a filter, is assumed to be independent of the source signal. The resonance frequencies are the direct consequence of the vocal tract, causing what are known as *formants*. They are the main contributor to the spectral envelope of the voice (i.e., the relative amplitudes of the harmonic series) and change along with the length and shape of the vocal tract. Compared to the vibrating vocal folds (source), the vocal tract (filter) can only exhibit relatively slow alternations. Formants allow for the articulation of different vowels and a wealth of different timbres.

Due to the independence of source and filter, it is possible by estimating one component to reconstruct the second. Thus in vocal-signal analysis, the spectral envelope is of specific interest, since it determines the timbre—everything that is not pitch or loudness—to a large degree. One prominent method to estimate the filter/spectral envelope is linear prediction, and its results are the linear predictive coefficients (LPCs) [2]. The basic idea is that the current amplitude of a time-varying digital signal is predictable (approximately) from a linear combination of its past values. The error of this linear model equals the source signal relating to vocal fold characteristics, thus making the source and filter separable.

In contrast, spectral approaches measure the relative contributions of sinusoidal components in signals, often through short-time analysis under assumptions of local stationarity. One of the earliest approaches used sinusoidal modeling, which fits the frequencies and amplitudes of a number of time-varying oscillators to a signal. This method was later extended to model the residual signal as either noise alone or both noise and transients [4]. Though it has the properties of being both compact and complete, sinusoidal modeling can be computationally expensive and quite sensitive to the presence of other signals. As a result, it is more common to model vocal-tract characteristics via mel-frequency cepstral coefficients (MFCCs). MFCCs have been used specifically for music analysis since being introduced by [7] and, until the recent popularization of deep learning, served as one of the standard features in speech and music timbre analysis.

MFCCs are computed through a two-stage process. First, a mel filter bank is applied to the audio signals, typically via the fast Fourier transform for efficiency, such that frequency components are collapsed into 30–120 half-overlapping triangular-shaped filters along a frequency scale grounded in psychoacoustics. Next, the signals are transformed into the cepstral domain by computing and applying a discrete cosine transform (DCT) to the log-magnitude spectra, thus decorrelating the mel-filter bank coefficients. Discarding some of the higher-order coefficients of the DCT results in the representation of a low-pass-filtered spectral envelope, which can be reconstructed by applying the inverse DCT. More recently, the "fluctogram" has been proposed as an alternative time–frequency representation specific to the singing voice. Designed to encode the temporal evolution of the fundamental frequency and its harmonics [8], the fluctogram is computed for several frequency bands based on the cross-correlation of a log-scaled spectrum to the succeeding spectrum, exploiting the characteristic of the voice as a continuous pitched source.

Importantly, the motivation for these models is based on the assumption that the signal of interest contains only a single voice recorded in isolation. However, most recordings in consumer music settings are the result of professional sound production, also referred to as "mixing," an artistic process that combines a number of audio signals arranged in time, subject to any number of complex effects processors (e.g., compression, equalization, reverb, and distortion). For clarity, this process can be expressed as the summation of N digital audio signals, notated as $x[t] = \sum_{n=0}^{N} \alpha_n[t] * f(x_n[t] \,|\, \phi_n[t])$, where α defines a time-varying gain and f an arbitrary, often nonlinear, effects chain with its composite parameters $\phi_n[t]$. In this article, we use "recorded music" to mean the resulting signal $x[t]$, and "voice" as all K signals, x_k, $K \leq N$, that were produced by human voices (note, however, that the true number of voice signals, K, in a recording will not necessarily correspond to the number of distinct voices a listener perceives).

Often in music, one or more of these voice signals will emerge as the "lead" voice, whereby a typical listener perceives a single voice as being particularly salient. Robust, human-level

understanding of singing in recorded music therefore presents the additional complex task of first identifying the voice amid multiple sounds before extracting some desired high-level information.

When creating the architecture for vocal analysis systems to operate on recorded music, any one of three basic approaches can be taken. First, a system could be designed to only consider parts of the music signal where the voice is naturally isolated (i.e., points at which all nonvocal signals are silent). This approach is conceptually straightforward, but has three major drawbacks. The system is limited by its ability to discriminate a solo voice from all other conditions, and any errors will propagate through the system. There are no guarantees that isolated vocals will occur with sufficient frequency in a recording to perform some task. Even so, occasional views of the signal will be inadequate for applications that require comprehensive information regardless of interference (e.g., transcription of melody or lyrics).

Another approach, described in a large body of work in source separation of music, attempts to isolate a sound source of interest given a mix of other signals [9]. Source-separation algorithms generally fall into one of two categories: those that exploit domain knowledge of music in the application of signal decomposition algorithms (e.g., independent components analysis, nonnegative matrix factorization, robust principal components analysis) or those that use data-driven methods that act as filters to directly produce the voice signal in isolation. To the former, the singing voice is often sparse and nonrepetitive in a musical mixture, and algorithms can exploit these properties to perform singing-voice separation [10]. Accompaniment is often considered "low rank," in that it consists of instruments (e.g., drums or guitars playing repetitive patterns), whereas the voice is monophonic and irregular. In a complementary fashion, audio decomposition techniques can be applied in a cascaded fashion to disassemble the music recording into a set of midlevel components that are fine enough to model various characteristics of the singing voice, while coarse enough to keep an explicit semantic meaning of the components [11]. More recently, deep neural networks have emerged in singing-voice separation as powerful nonlinear filters. These algorithms are trained on existing pairs of aligned mixture and isolated voice signals, with the objective of minimizing the error between the true and estimated vocal signals. Modern deep-learning approaches show particular promise, and various works continue to explore different architectures, objective functions, and data sources [12]. To chart progress in this area, the Signal Separation Evaluation Campaign is an annual community-led event organized to systematically and reproducibly compare source-separation algorithms [13].

The third, and most direct, approach is to develop models or features that can characterize the voice despite the presence of interfering signals. In practice, MFCCs or LPCs have proven to be reasonably useful as a consequence of standard practice in sound production; typically, though by no means always, lead vocals are the predominant signal in the mix, and thus vocal information also tends to dominate these representations. For some tasks, feature engineering has proven rather effective, but there are obvious limitations to this approach. More generally, given advances in machine learning, and particularly deep learning, generic time–frequency representations (e.g., MFCCs or spectrograms) or raw time-domain waveforms may be used as inputs to deep neural networks. Data-driven methods enable the system to tease apart signal attributes relevant to voice given an objective, but present their own challenges with respect to data collection, training, and computation. We will see how these three approaches are applied as a function of the task, model, and data.

Singing analysis applications

From the perspective of web-scale music listening, singing-voice analysis aims to extract high-level information from audio signals to enable systems to address some user need (e.g., find instrumental music or songs without expletives). This application space is broad, given the range of sounds the human voice can produce, and so it is helpful to distinguish between the different categories of sound within this space. Musicality and natural language can be represented as two partially overlapping subsets (Figure 2), whose union lies within a larger space of vocalization: for example, one can sing without adhering to the rules of any natural language (e.g., humming or scat), communicate via speech amusically, or produce a variety of sounds that qualify as neither. The ability of humans to comprehend information in musical or linguistic contexts is achieved through high-level cognition built upon lower-level perceptual faculties.

Noting that significant time and attention has been paid to the computational analysis of speech [2], we focus our attention here on three types of singing, each with an eye toward the corresponding musical applications:

- *Vocalization*: acoustic primitives of voice that are common to both musical and linguistic contexts, contributing to such tasks as vocal activity, technique classification, and vocalist identification
- *Vocal music*: singing in musical contexts, which give rise to intonation, melody, and genre by establishing or reinforcing the elements of harmony, rhythm, and timbre

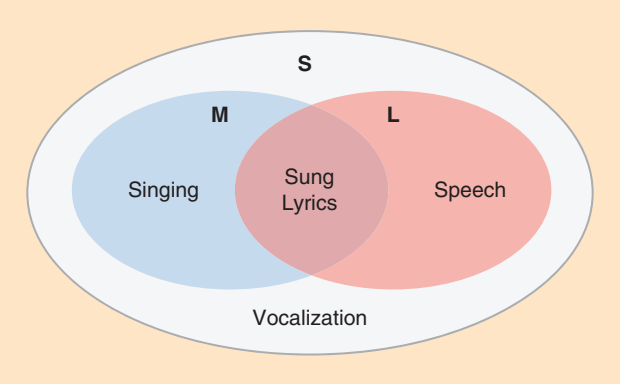


FIGURE 2. An illustration of the set relationship between musicality across the space of sound produced by the human voice (S) such that "singing" comprises vocalizations in a musical grammar (M), "speech" as vocalizations in a linguistic grammar (L), and "sung lyrics" as the intersection of the two, $M \cap L$.

- *Sung lyrics*: the intersection of musicality and language, with applications similar to those in speech recognition, such as language identification, audio–text alignment, and transcription.

Before proceeding, we offer a few notes for consideration. First, these domains are ordered by level of abstraction, which serves as an approximate guide of computational difficulty (e.g., vocal activity is simpler than melody estimation, and both are simpler than lyrics transcription; this is not to say, however, that any of these tasks are trivial, as all are open research areas). Related tasks typically employ similar approaches, and lower-level tasks or representations are often reused in higher-level ones. Finally, the applications presented here are connected to salient dimensions reported by listeners when relevant, both to motivate and identify opportunities for future work.

Vocalization

As described previously, vocalization encompasses the superset of sounds produced by the human voice. Given that listeners are particularly sensitive to the presence of voice generally, the first stage in singing analysis aims to characterize the acoustic primitives of voice. These systems focus on the human voice as a sound source and thus share the common properties that they are not inherently constrained to musical applications. As a result, these systems find additional application in higher-level voice-analysis systems (e.g., only apply lyrics transcription when the voice is present to reduce errors).

Activity detection

The automatic detection of singing voice in recorded music finds immediate use in recommendation contexts (e.g., identifying "focus" music). Referred to as *vocal activity detection* (*VAD*), such systems typically predict the likelihood of vocal activity on short time-scales (i.e., 1 s to dozens of seconds) and can be applied convolutionally over longer signals to produce time-varying estimates; others aim to make predictions over a complete recording. Continuous-valued likelihoods may be simply thresholded at some bias point to produce binary decisions between vocal or instrumental states. Alternatively, in time-varying estimates, postprocessing [e.g., hidden Markov models (HMMs) or median filtering] may be used to prevent spurious or brief detection intervals.

At a high level, two basic approaches may be taken to detect the presence of a singing voice from an observation. The traditional approach involves feature engineering in combination with such classifiers as random forests, support vector machines (SVMs), or neural networks. The current state of the art with this approach uses fluctogram and delta-MFCC features (i.e., first-order difference) that are fed to a long short-term memory recurrent neural network [8]. Alternative approaches use deep neural networks in an end-to-end fashion. The current state of the art with this approach produces results similar to those of its feature-engineered counterpart when trained without data augmentation [14]. With data augmentation, the results seem to be superior, but it is still not clear how previous approaches would also benefit from data augmentation.

One particular challenge faced in VAD systems is a heightened sensitivity to data-set composition and domain transfer for training and evaluation. Both prior discussed approaches yield models that appear to distinguish even highly harmonic instruments producing voice-like pitch trajectories from actual singing voices, as demonstrated by extremely low false-positive rates on specifically curated tests. However, it is especially important to make use of instrumental music to better assess performance [8]. Training with instrumental music helps decrease false-positive rates, while evaluating on instrumental music can reveal certain weaknesses in a given model. Algorithms insensitive to variations of the level of loudness may allow for meaningful comparison. Otherwise, a performance gap between two methods—one loudness-invariant, the other not—could possibly be caused by a convenient level of loudness for the loudness-sensitive method. To give an example, for a loudness-sensitive method the number of false positives will often decrease along with the level of loudness, contrary to the output of a loudness-invariant method, where the number of false positives stays constant.

Technique classification

Machine perception of vocal technique, a burgeoning area of research in singing-voice analysis, relates to a listener's affinity or aversion to a music recording. Phonation modes are important building blocks of more advanced vocal techniques and corresponding analysis systems, such as genre recognition or lyrics transcription. Technique modeling can be seen as a more granular form of general vocal-activity detection, where short-time observations are classified into the kind of vocal activity present. To these ends, the Phonation Modes data set consists of sung vowels in one of the four main phonation modes: breathy, pressed, flow, and neutral [15]. By using a model of singing voice that simulates airflow and pressure through the vocal folds, the authors of the data set achieve an accuracy of 65% with a four-way classifier.

VocalSet is a singing voice data set that consists of these more advanced vocal techniques [16]. These vocal techniques include vibrato, straight, breathy, vocal fry, lip trill, trill, trillo, inhaled singing, belting, and spoken. Some of these techniques are found in a basic vocal repertoire, such as vibrato or trill, while others, like inhaled singing or vocal fry, are found in more advanced repertoires. Figure 3 shows spectrograms of each of these techniques for a male singer in the data set. The spectrograms of each technique are visually different, despite coming from the same singer with the same musical intention (e.g., singing scales, arpeggios, and long tones). VocalSet was collected by recruiting professional singers to sing examples of each of these techniques. The data set consists of 20 singers (11 female), each singing these ten techniques on scales, arpeggios, and long tones. VocalSet contains 10.1 h of recordings. Using deep convolutional neural networks, the authors of the data set achieved a precision of 0.676 and a recall of 0.619 in a ten-way classification setup.

Notably, the role of phonation in performance varies across musical cultures. Computational and quantitative techniques have been used to study variations of singing technique in the

Beijing Opera as a result of educational influence [17]; founded by different instructors, the students of different schools inherit the corresponding vocal production characteristics. Going beyond the subjective description of singing style (e.g., sweet, clear, fragile), the authors take into account a diverse set of audio features common in music-signal analysis, and experimental results support previous findings in the musicology literature.

Singer identification

The automatic identification of vocalists in music audio can help address metadata errors and identify collaborations in

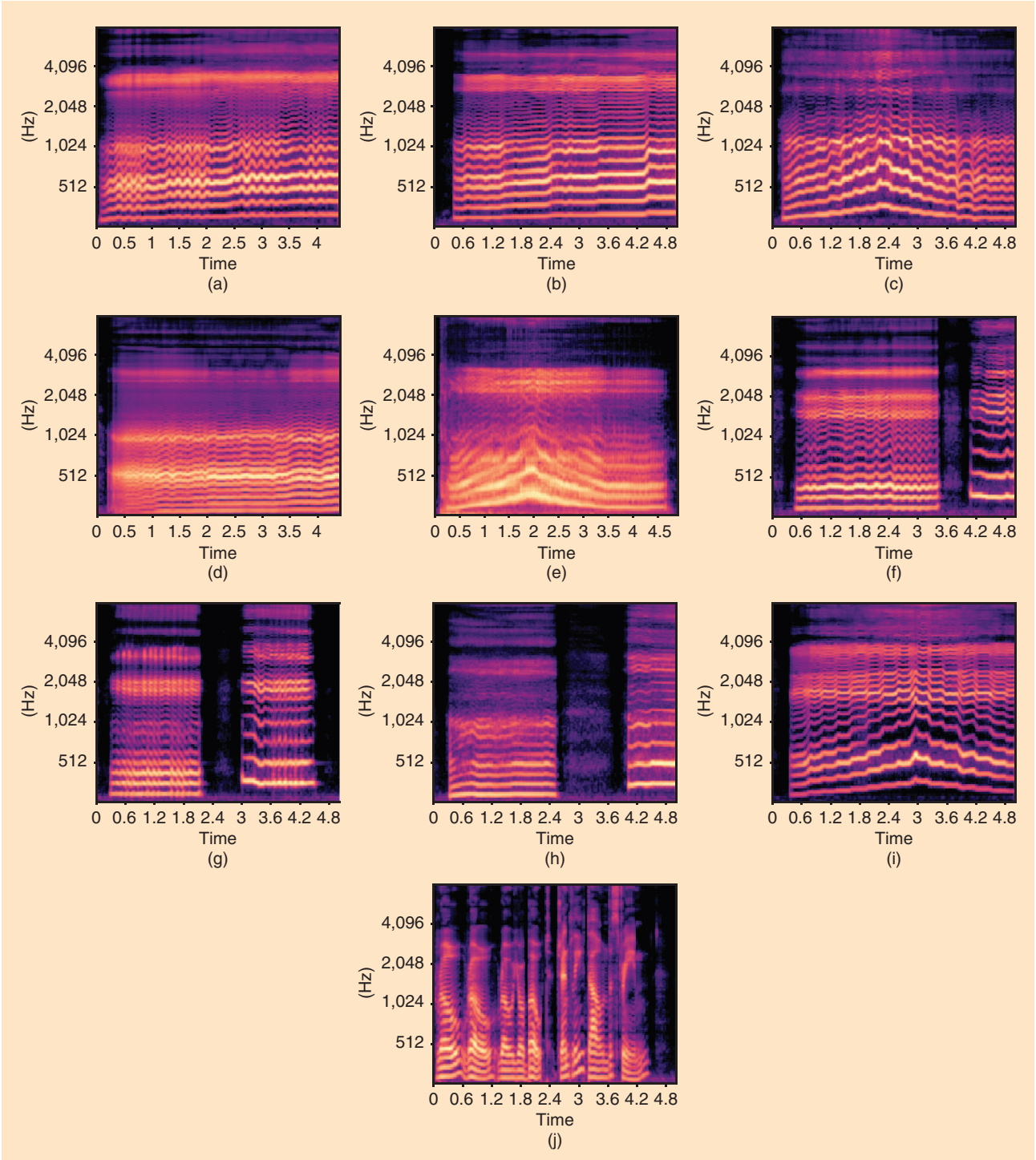


FIGURE 3. Mel spectrograms of the ten vocal techniques contained in the VocalSet data set: (a) vibrato, (b) straight, (c) breathy, (d) vocal fry, (e) lip trill, (f) trill, (g) trillo, (h) inhaled singing, (i) belting, and (j) speaking. Each is a performance of a specific vocal technique by the same male singer. Different vocal techniques produce characteristic spectrograms.

recordings, two commonly recurring challenges. As yet another degree of specificity beyond technique modeling, the problem of vocalist identification is one that stands to benefit greatly from data-driven methods. While efforts in singer identification (singer-ID) have produced few results, one system of note proceeds by extracting vocal segments from songs, computing some engineered feature representation, and classifying with a machine-learning model of choice (e.g., SVMs or Gaussian mixture models) [18]. Singer-ID is distinct from the recognition of vocal technique alone in two ways: 1) longer time scales may be necessary to distinguish among different vocalists; and 2) it remains unclear what the preceptual or computational limits of singer-ID might be in terms of accuracy or performance. However, given that music collections typically provide artist labels on recordings, singer-ID presents an interesting opportunity due to the availability of data for supervised machine learning.

Vocal music

Building upon general vocalizations, we now focus on the analysis of the singing voice in musical contexts specifically. While singing may also convey natural language, "vocal music" is defined as the musical compositions or performances that feature one or more human voices. This entails an understanding that singing conforms to the basic dimensions of music: harmony (pitch), rhythm (timing), and timbre (source discrimination). However, while timbre encompasses the distinguishing traits of a particular sound source—here, the human voice—a singer is considered a monophonic instrument, i.e., of a single pitch. While the human voice is capable of producing multiple pitched sounds simultaneously, the practice is uncommon and not considered here. As a result of emphasis placed on harmony in traditional music theory practice, the analysis of singing often, though not exclusively, focuses on pitch.

Intonation

The harmonic basis on which a piece of music is built is known as *intonation*. In popular Western music, the common tuning system is known as *12-tone equal temperament* and has standardized by convention on A4 = 440 Hz. While some popular instruments produce sound in quantized pitch intervals (e.g., piano), the human voice is capable of producing arbitrary pitch. Some non-Western music traditions, such as Indian art music (IAM), take other approaches to intonation that complicate the design of signal processing systems, making intonation a relevant research topic. For context, IAM refers to two art music traditions of the Indian subcontinent, Hindustani music (also known as *North Indian music*) and Carnatic music (also known as *South Indian music*). Both Hindustani and Carnatic music are singing-centric traditions, and therefore the voice effectively dictates the intonation used in a piece. Rāga is defined as the melodic framework in IAM and serves as the core musical concept used in composition, performance, music organization, and pedagogy. Hindustani and Carnatic music is characterized by different melodic attributes, such as svaras (roughly speaking, notes), intonation of the svaras, and characteristic melodic phrases.

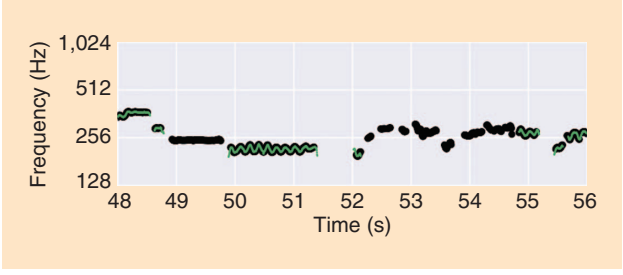


FIGURE 4. A vocal melody estimated by an algorithm (green) against the ground truth (human labeled) vocal melody (black).

Due to the importance and variation inherent to pitched singing, the lack of simplifying assumptions around tuning complicates the automatic analysis of these kinds of music. Carnatic music, for example, does not make use of an equal-tempered tuning schema, being closer to five-limit just intonation, whereas Hindustani music can be explained by a mixture of equal-tempered tuning and five-limit just intonation (a five-limit tuning system uses powers of two, three, and five to compute notes relative to a reference frequency). The intonation of svaras is an important characteristic of a rāga, and so detailed pitch distributions are informative as a result. It has been shown, for example, that the shape of the pitch histogram for different svaras can assist in automatic identification of rāgas [19]. Since there exists subtle intonation differences across rāgas, the frequency resolution chosen for intonation analysis in IAM is much higher than that for many other music traditions.

Melody estimation

The task of determining the pitch, or fundamental frequency, of the singing voice in music over time is generally referred to as *vocal melody estimation*. Estimated melodies are typically represented in the form of time series (time, pitch), where the interval between time steps is small (e.g., 10 ms), and pitch values are continuous (measured in hertz) values rather than as discrete note values. Figure 4 shows an example of a vocal melody estimated by an algorithm (green) plotted against the ground truth vocal melody (black) for a short excerpt. Note how by representing the pitch values on a continuous rather than discrete frequency grid, information, such as vibrato, is captured between 50 and 51 s in the figure. Additionally, note that part of the task is also to determine where no vocal melody is present.

There are three common types of approaches to vocal melody estimation [20]: salience, source separation, and machine-learning based. Salience based methods leverage the assumption that vocals exhibit a known harmonic series. To exploit this information, these approaches first estimate a vocal salience representation, a time–frequency representation derived from a short-time Fourier transform, realized by reweighting the amplitude of each time–frequency bin based on the presence or absence of related harmonics. The purpose of this is twofold: 1) to de-emphasize content that is not part of the vocal melody and 2) to emphasize content that is likely part of the vocal melody (i.e., content with many related harmonics). Salience representations are computed, for example, via harmonic summation, harmonic percussive

source separation, or filtering/equalization. After computing a salience representation, these methods often apply heuristics-based rules for selecting the most likely vocal melodies from the computed representation. Source separation-based methods first isolate the singing voice and subsequently apply a pitch tracker in order to compute the melody, or conversely they jointly estimate the singing-voice audio signal and the vocal melody. More recently, machine-learning methods have been used to turn the task into a classification problem by discretizing the frequency space with at least one class per semitone and predicting the most likely class over time [21], [22]. Alternatively, machine learning can be used to learn robust salience representations [23].

Vocal melody estimation has a number of applications in musical indexing and retrieval. A long-standing goal of MIR is known as *query-by-humming*, where a listener can search a collection of content by vocalizing a given melody. The ability to find specific recordings by melody would likely result in related results and similarity-based retrieval. Additionally, melody is a predominant feature of music and would further inform higher-level analysis, such as pattern discovery and structural segmentation (e.g., thumbnailing or chorus detection).

Estimation of the predominant melody is also at the core of singing-voice analysis in IAM [24]. In a typical performance, the main vocalist is accompanied by another melodic instrument, almost like a lagging imitation of the lead. There are approaches that exploit this convention by tracking the two melodic contours simultaneously, one of which being that of the lead vocalist. Attempts have been made to automate the selection of pitch contour corresponding to the lead artist by using temporal instability of the voice harmonics. Due to the subtle nuances in the temporal evolution of the melodies (specifically in the transitory regions between two svaras), the entire pitch contour is often used as a midlevel feature for singing-voice analysis. Often, steady-state regions and transitory regions in a melody are segmented for better characterization of the melodies.

Genre

Among the more abstract concepts in music, genre is used to describe the musical categories that emerge naturally from a culture's influence on itself. A genre is established through the use or reuse of certain musical aspects, such as structural form, instrumentation, or melodic patterns, which leads to shared understanding across groups of people. Various forms of rock prominently feature distorted guitars, for example, while blues is known for dominant chords and 12-bar phrasing.

While there are numerous, often inscrutable characteristics that may contribute to the boundaries of a genre, it is relevant here to consider those that place a specific emphasis on the singing voice. One instance is that of subgenres of metal music, which are characterized by extreme vocal effects [25]. One of the primary motivations behind singing-voice analysis in IAM is for automatic rāga identification. Recently, a technique called *time-delayed melody surfaces* has been shown to capture continuous tonal and temporal characteristics of these melodies, resulting in a significant improvement in rāga recognition accuracy [26]. Rap is another notable instance of a genre identified in large part by distinctive rhythmic voice delivery characteristics. It has been demonstrated that only 11 perception-inspired features lead to 91% classification accuracy between rapping and singing with only 3-s isolated vocal segments [27]. The most salient feature was found to be the ratio of voiced frames to nonsilent frames, confirming the prominent role of rhythm and lack of melodic characteristics of rapping, in contrast with the more melodic nature of traditional singing found in contemporary rhythm and blues music.

Genre can also serve as a suitable proxy for singing style, a musically appealing but difficult to define characterization of vocal performance (e.g., theatrical, aggressive, or powerful). Vocal-specific features, such as statistics computed over fundamental frequency (f_0) contours, are useful for discriminating between different singing styles in both supervised and unsupervised approaches [28]. Clustering these features has enabled the semantically meaningful organization of a collection of 50,000 excerpts of folk music from around the world, while large-scale embeddings for vocal style are also a promising avenue of research [29].

Sung lyrics

Viewed from the perspective of linguistics, human vocal communication with language has four dimensions [30]:

- *Phonemes*: the building blocks of vocalized language, representing discrete units of sound
- *Prosody*: the articulation of phonemes over time, including aspects of inflection, duration, rate, or intonation
- *Vocabulary*: the combination of phonemes into words as higher-level sound objects
- *Grammar*: the sequential, structural composition of words.

At the intersection of music and natural language, the singing of lyrics presents unique difficulties beyond those typically faced in speech processing alone [31]. Often the rules of grammar are bent or ignored for artistic reasons (e.g., rhyme). Prosodic elements are constrained by the melodic and rhythmic dimensions of a musical work and not necessarily by the language in which the lyrics are performed. For example, the typical fundamental frequency for female speech lies between 165 and 200 Hz, while in singing it can reach more than 1,000 Hz. This is further complicated in a tonal language like Chinese, where the inflection of pitch is also used to convey semantic meaning. As a result, traditional speech corpora are insufficient for building data-driven models for singing analysis, given the degree of domain transfer between spoken language and vocal music. Meanwhile, accompanying instrumentation complicates traditional assumptions regarding noise in speech processing, in that typically all signals in recorded music are both harmonically and temporally correlated. With that in mind, we now turn our attention to methods for language identification, the alignment of audio and lyrics, and lyrics transcription.

Language identification

Singing language identification (SLID) can be seen as a simplification of comprehensive lyrics transcription. In music services for global populations, the predominant language of performance

is a valuable attribute: It provides deeper insight into music catalogs in linguistically diverse settings, such as India or the Philippines; and, through greater comprehension of the content, enables a deeper understanding of a listener's language preferences. The latter is a complex issue facing recommender systems because of asymmetrical preferences toward music consumed in different origins (e.g., users in country *X* might listen to music from country *Y*, but not the inverse).

SLID systems conventionally approach the task by modeling the statistics of phonemes over long time scales, building different templates on a per-language basis. One modern effort of note is that of [32], which focuses on 25 languages drawn from 25,000 music videos. The authors explore a variety of feature representations, leveraging both acoustic and visual descriptors aggregated over the temporal context of the signal, fed into a number of binary SVM classifiers (one per language). Experimental results show that a mix of acoustic features—spectrograms, MFCCs, and stabilized auditory images—led to a performance on a test set of 44.7%; by adding visual features, the system achieved 47.8% accuracy. Interestingly, this system considers general-purpose feature representations, placing the burden of modeling on a powerful classifier, and calls into question the need to distinguish between vocal and nonvocal segments.

Audio–lyrics alignment

Time-alignment of lyrics with the corresponding audio is necessary for such popular applications as karaoke and subtitling of music videos. The availability of alignments also makes possible a host of applications, such as automatic radio edits, playback starting/ending at specified lines, and analyses of how words in music correspond to beats, melodies, and other musical structures [33]. Manual alignments do not scale to large collections of audio, raising the need for accurate automated alignment algorithms.

The goal of automated alignment, shown in Figure 5, is to take the audio and lyrics and produce a time alignment of the two inputs. Alignments are typically at the word level, but may also be at the level of lines or phonemes, depending on the downstream application. Line-level alignments may be sufficient for such products as subtitling or some karaoke interfaces. LyricAlly is one system of note that detects such structural elements as beats and rhythm, which are used to segment the audio into the introduction, verses, chorus, bridge, and coda [34]. The lines in the lyrics corresponding to these sections are then aligned to the segmented audio. Word-, syllable-, or phoneme-level alignments require greater precision. Some works rely on annotations, such as Musical Instrumental Digital Interface (MIDI) files or lead sheets; however, these cannot generalize to unannotated music.

The speech technology community uses a method called *forced alignment* to time-align audio and transcripts. Forced alignment involves finding the Viterbi path through HMMs that map phonemes to MFCCs or other features of the acoustics. These HMMs are trained from large corpora of transcribed speech. Several speech toolkits, such as CMU Sphinx (https://cmusphinx.github.io), the Hidden Markov Model Toolkit (http://htk.eng.cam.ac.uk), and Kaldi (http://kaldi-asr.org) implement forced alignment, including the ability to train the acoustic HMM models, with wrappers, such as the Montreal Forced Aligner (http://montreal-forced-aligner.readthedocs.io), providing interfaces to these programs. Forced alignment works best when line or phrase-level boundaries are specified, since alignment quality degrades with audio longer than a minute. Forced alignment forms the basis of most lyrics–audio alignment algorithms. However, some characteristics of singing make it challenging to apply alignment models developed for speech to music [35].

Introduced earlier, lyrics alignment is one area that makes use of vocal detection and separation as preprocessing steps before alignment to mitigate challenges posed by recorded music. In addition, it is possible to reduce the sound of

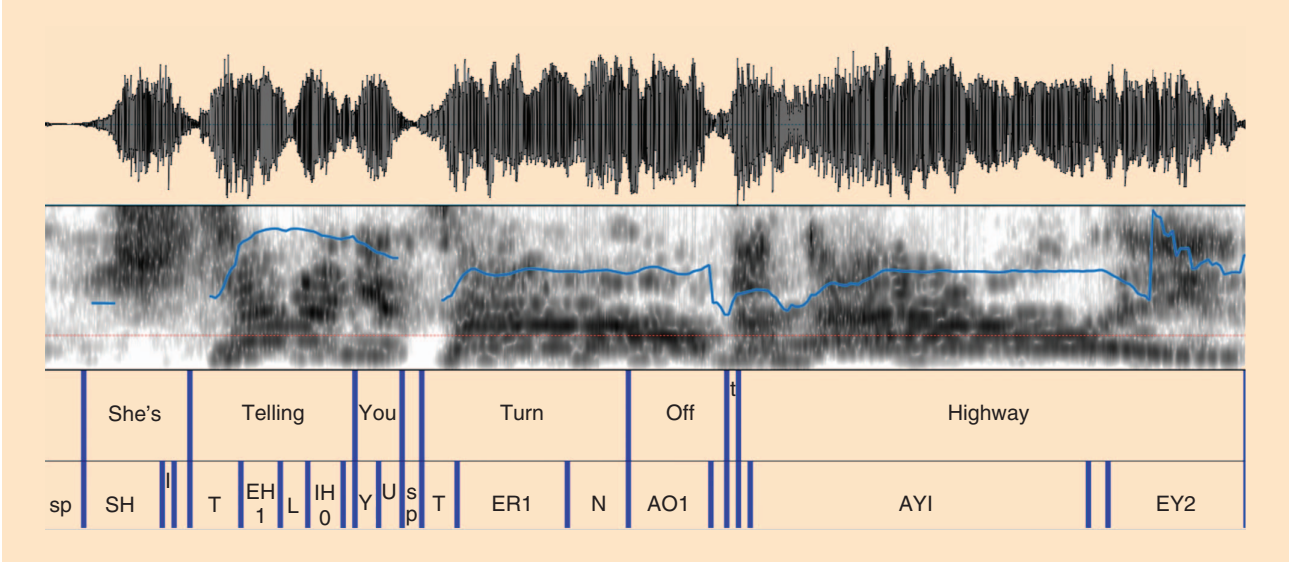


FIGURE 5. Visualization of automated word- and phoneme-level alignments from a segment of a song generated with the Praat software (http://www.fon.hum.uva.nl/praat).

accompanying instruments with f_0 estimation and resynthesis and to adapt the acoustic HMM models trained on speech to a small corpus of music [36]. Systems may also use placeholders in the HMM for such words as "yeah yeah" that may not be captured in the reference lyrics. Integrating musical information, such as chord sequences, is also helpful for improving lyrics-alignment performance [37].

Lyrics transcription

Lyrics transcription is generally performed in two steps: first, phoneme probabilities are recognized in the singing audio by using an acoustic model; then, the results are processed with a language model to obtain plausible word sequences. As in speech recognition, most early lyrics-transcription systems relied on HMMs for acoustic modeling. Due to the lack of lyrically transcribed singing data, many systems trained acoustic models on read speech, with language models built on actual texts of lyrics. For example, language modeling can be achieved with a finite-state automaton fitted to the lyrics of a collection of Japanese children's songs [38]. The system is tested on sung phrases consisting of five words, without accompaniment, achieving a word error rate of 36%. By training speaker-specific acoustic models, the word error rate is lowered to 27%.

Several improvements have been proposed that incorporate intuition about human perception to lyrics. It is noted that source separation can be used as a preprocessing technique to improve model accuracy. Repetition and structure in music, such as the chorus, may also be exploited to improve transcription accuracy [39]. Three different strategies are proposed for combining individual results: feature averaging, selection of the chorus instance with the highest likelihood, and combination using the Recognizer Output Voting Error Reduction (ROVER) algorithm. Twenty unaccompanied English-language songs from the Real World Computing (RWC) database were used for testing; chorus sections were selected manually. The best-instance selection and the ROVER strategies improve results significantly; with the ROVER approach and a general-purpose language model, the phoneme error rate is 74% (versus 76% in the baseline experiment), while the word error rate is improved from 97% to 90%. Interestingly, cases with a low baseline result benefit the most from exploiting repetition information.

To overcome the lack of realistic training data, forced-alignment algorithms may be used to fit a set of unaccompanied singing with unaligned lyrics [40]. For example, deep neural networks are trained on MFCCs of music signals to produce singing-specific acoustic models. These models produce better results compared to those trained on speech, with the phoneme error rate falling to 80%. Notably, both word and phoneme error rates are expected to be higher in lyrics transcription than in speech recognition. While the limits of human lyrics recognition are unknown, the phenomenon of "misheard" lyrics is common [41].

A simplified form of lyrics transcription is the ability to pinpoint specific words (e.g., expletives) in recordings. Many song lyrics contain expletives, and there are numerous scenarios in which it is necessary to know when these words occur (e.g., "family-friendly" listening sessions). In the case of airplay, expletives are commonly "bleeped" or acoustically removed. The task of finding such words is based on the alignment strategies described previously, taking advantage of the wide availability of textual lyrics. The system proceeds by automatically aligning text lyrics to audio, searching for predefined expletives in the result, and subsequently modifying the signal where any flagged instances occur (e.g., adding white noise as an obfuscation) [40]. The test data set consists of 80 popular songs, most of them hip-hop. Annotations indicated 711 instances with 48 expletives on these songs, and the matching textual, unaligned lyrics were manually retrieved from the Internet. Using the acoustic models described therein, 92% of the expletives were detected in their correct positions with a tolerance of 1 s.

Next steps

Getting started with singing analysis

As illustrated by the breadth of the previous section, singing-voice analysis is a diverse area of study with potential to enable a variety of large-scale applications. However, this rich array of possibilities may also make it difficult to decide where and how to first dive into this topic. To help direct new explorations in singing-voice analysis, there are three tasks we recommend as good entry points: vocal-activity detection, singer-ID, and SLID. Each can be framed as a straightforward classification problem with objective evaluation measures (i.e. precision, recall, f-score) and in each case the task of finding or collecting labeled data is relatively easy. To further facilitate this exploration, we also provide an open-source software tutorial for self-guided exploration (https://github.com/spotify/ieee-spm-vocals-tutorial).

Vocal-activity detection is a logical starting point for those new to music signal processing with an interest in singing analysis. Recognizing vocal activity as a low-level percept, computational systems can focus on short-time observations drawn from audio signals, simplifying both labeling and modeling as a binary classification task. Given the increasingly mature state of machine learning, the challenge of building a VAD system resides more in obtaining or curating data for training and evaluation. The two conventional data sets used in VAD research are the Jamendo collections, though newer collections like MedleyDB (http://medleydb.weebly.com/), OpenMIC-2018 (https://github.com/cosmir/openmic-2018), or AudioSet (http://research.google.com/audioset/) provide more data for training such models. A particular advantage of VAD as a task is that its simple framing allows one to study the effects of data-set composition on model performance. As mentioned previously, the inclusion of a cappella (solo voice) or instrumental music in a data set can help address false negatives or false positives, respectively, but it is also possible to synthesize more training data from multitrack recordings (e.g., MedleyDB).

Another attractive, near-field opportunity suitable for newcomers to the topic of singing-voice analysis is that of singer-ID. As discussed, methods for singer-ID are somewhat under-represented in the literature, leaving ample room to improve upon the state of the art. Additionally, there is often a 1:1 correspondence between recording artist (or group) and vocalist (i.e., a band features a single singer in all of its recordings), and it is

possible to collect large data sets for training machine-learning models without too much effort. This observation can be combined with modern source-separation algorithms to produce reasonable approximations of vocals in isolation, mitigating any confounding factors of instrumentation. This approach can be applied to the Free Music Archive (FMA) data set (https://github.com/mdeff/fma), which contains 100,000 recordings from more than 16,000 unique artists, with more than 1,000 artists having at least 20 recordings. Alternatively, Stanford's Digital Archive of Mobile Performances collection (https://ccrma.stanford.edu/damp/) features 35,000 solo voice recordings from roughly 350 amateur singers, which mostly bypasses the need for source-separation preprocessing. This data could be used to train a model as in the VAD scenario, with a classifier applied to short-time observations of audio signals. We emphasize that these artist–singer labels can be used to fit deep-learning models whose intermediary representations (e.g., the penultimate layer) can be used as an embedding model for similarity and retrieval.

A third accessible voice-analysis application is that for identifying the language of the song. While there is traditionally no mutually agreed upon data set for this problem, the FMA contains non-English-language tags for several hundred recordings, and global music services no doubt contain playlists or artists that consist of music performed in a given language. Similar to the formulation of singer-ID, language identification may benefit from the application of source separation as a preprocessing step, and there is considerable opportunity to advance the state of the art in the area of sung lyrics.

Challenges and opportunities

Singing-analysis research is rich with opportunities and challenges. We summarize a few. Subjective evaluation of singing-voice models, as in source separation and similarity, remains a challenge [42]. Objective metrics of source-separation quality are widely used (e.g., signal–noise ratio) but their ability to mirror perception is limited. Expert or crowdsourcing listening tests are often used, but researchers have yet to adopt a standard and well-controlled protocol. Singing-style models have mostly been evaluated using listening tests, and these have been small in scale due to the significant human effort involved. Larger models that cover diverse music require more quantitative methods. There is not yet a standard for benchmarking models of vocal style, for defining vocal similarity or style, or for quantifying listeners' perception of the singing voice. While there is some work that investigates the relationship of phonation modes with vocal styles, it is unclear how it relates to perception of the voice and remains an open area of research.

Machine-learning-based approaches are becoming ubiquitous to most aspects of computational analysis of vocals, but we have yet to see the kinds of dramatic improvements that have been achieved recently in related fields. On reflection, this is likely due to a lack of large, readily available collections for music signal processing research, like ImageNet for object recognition. Thus, while the newer data sets mentioned here, such as the FMA, may help address this shortcoming, more effort is needed to curate or mine large-scale data sets for other tasks in singing-voice research. For example, user-contributed lyrics are widely available on the Internet, and the ability to align these text documents with audio would transform the field.

Curating labeled music data sets for every task may prove cost prohibitive, given the skills required, as in the case of melody annotation. For these tasks, it may be more practical in the short term to artificially generate training data from symbolic signals, such as MIDI files and lead sheets, using realistic instrument synthesizers. This is not yet feasible for all tasks involving vocals, since modern voice synthesizers have yet to fully replicate natural singing. However, advances in melody estimation may provide realistic voice approximations, thereby producing more realistic data for training. Similarly, vocal source separation or an increase in the availability of multitrack recordings makes it possible to create mixes of arbitrary pairs of vocals and instrumentals. Importantly, unlabeled vocal music content is abundant. By mining large catalogs of music, we can build weakly labeled training sets or investigate multimodal approaches to data-set creation (e.g., music videos that feature lyrics).

Finally, most music informatics research is focused on analyzing commercially produced music content, which typically is created by professional musicians and follows basic tenets of music in accordance with the relevant genre or tradition. On the other hand, content produced by amateurs is not bound to follow these tenets and often poses a challenge to the existing singing information processing approaches. In recent years, the volume of such content and applications has risen significantly, often in the context of music education and gaming, (e.g., karaoke applications). The imprecision of amateur singing may be more pronounced than that for instrumental performances by amateurs since the frequencies produced by the voice are not naturally quantized, like they are, for example, for the flute, and have neither tangible nor visual feedback, as with a violin. Given that there are vastly more amateur than professional singers, the automatic analysis of the singing voice presents a considerable opportunity to enhance the human experience of music.

Authors

Eric J. Humphrey (ejhumphrey@spotify.com) received his B.S. degree in electrical engineering from Syracuse University, New York, his M.S. degree in music engineering technology from the University of Miami, Florida, and his Ph.D. degree in music technology from New York University, where he worked with Juan Pablo Bello in the Music and Audio Research Laboratory. He is a machine-learning engineering manager at Spotify in New York City, helping teams research and develop machine-learning algorithms to improve the experience of listeners around the world. Previously at Spotify, he was a senior researcher focusing on machine-learning approaches to understanding music audio signals. Beyond research, he is also a singer–songwriter and multi-instrumentalist.

Sravana Reddy (sravana@spotify.com) received her B.S. degree in computer science, mathematics, and creative writing from Brandeis University, Waltham, Massachusetts, and her Ph.D. degree in computer science from the University of

Chicago. She has spent time at the University of Southern California's Information Sciences Institute in Los Angeles, Dartmouth College Hanover, New Hampshire, and Wellesley College, Massachusetts. She is a machine-learning engineer at Spotify in Boston, where she works on projects related to natural language processing and machine learning. Her research spans natural language processing, speech, machine learning, and linguistics, with a particular emphasis on language variation, including both dealing with it in practical systems and analyzing it using large corpora. Her interests also include applications of computation to literature and writing.

Prem Seetharaman (prem@u.northwestern.edu) received his B.S. degree in computer science with a second major in music composition from Northwestern University Evanston, Illinois, where he is currently a Ph.D. candidate working with Bryan Pardo. He works on problems in creativity support tools, audio source separation, and machine learning. In addition to research, he is an active composer and musician in the Chicago, Illinois, area.

Aparna Kumar (aparna@spotify.com) received her B.S. degree in physics from Drexel University, Philadelphia, Pennsylvania, and her Ph.D. degree from the School of Computer Science at Carnegie Mellon University, Pittsburgh, Pennsylvania. She is a senior research scientist at Spotify in New York City, focusing on audio understanding, perceptual evaluation, user modeling, and data mining for business applications. Her research began in computational biology. Her prior work includes mining pathology images, experimental design, and data collection for oncology drug development.

Rachel M. Bittner (rachelbittner@spotify.com) received her B.S. degree in mathematics and her B.M. degree in music performance from the University of California, Irvine. She received her M.S. degree in mathematics from New York University's Courant Institute in 2013. She received her Ph.D. degree in music technology from New York University, working in the Music and Audio Research Laboratory with Juan Pablo Bello, with her dissertation focus on the application of machine learning to fundamental frequency estimation. Previously, she was a research assistant at NASA Ames Research Center working with Durand Begault in the Advanced Controls and Displays Laboratory. Her research interests are at the intersection of audio signal processing and machine learning, applied to musical audio.

Andrew Demetriou (andrew.m.demetriou@gmail.com) received his B.A. degree in political science and philosophy from Queens College, City University of New York, and his M.S. degree in social psychology from Vrije Universiteit, Amsterdam. He is currently a Ph.D. candidate in the Multimedia Computing Group at the Technical University at Delft, The Netherlands. His academic interests focus on the intersection of the psychological and biological sciences and the relevant data sciences. His academic interests also extend to furthering our understanding of love, relationships, and social bonding; optimal, ego-dissolutive, and meditative mental states; and people performing, rehearsing, and listening to music.

Sankalp Gulati (sankalp.gulati@gmail.com) received his B.Tech. degree in electrical and electronics engineering from the Indian Institute of Technology, Kanpur, India, and his M.S. degree in sound and music computing from the Universitat Pompeu Fabra, Barcelona, Spain. He received his Ph.D. degree from the University of Pompeu Fabra in Barcelona, Spain, where he worked the Music Technology Group with Xavier Serra on the CompMusic project. His research interests include signal processing, time series analysis, and machine learning applied to audio music signals. He has years of industrial experience working in the domain of audio and speech technologies, music content analysis, music education, and is currently working on machine learning and artificial intelligence in the area of financial technology.

Andreas Jansson (andreasj@spotify.com) received his B.S. degree in computer science from City University, London, where he is a Ph.D. degree student and he is also a research engineer at Spotify in New York City. He is currently exploring deep neural network architectures for source separation and mining large commercial music catalogs for training data. Before joining Spotify, he worked at music start-ups The Echo Nest and This Is My Jam. He enjoys playing the accordion, lingonberry picking, and Emacs Lisp.

Tristan Jehan (tjehan@spotify.com) received his B.S. degree in mathematics, electronics, and computer science, and his M.S. degree in electrical engineering, computer science, and signal processing from the Université de Rennes I, France. He received his Ph.D. degree in media arts and sciences from the Massachusetts Institute of Technology. He is a director of research at Spotify, where he cultivates new technologies that can grow into next-generation features and business opportunities. He was chief science officer and cofounder of the music intelligence company The Echo Nest, which was acquired by Spotify to establish a new global standard in music personalization. He has introduced to the industry machine-listening technologies, which involve applications related to music similarity, discovery, and algorithmic music remixing. His academic work combined machine-listening and machine-learning technologies in teaching computers how to listen and make music on their own.

Bernhard Lehner (Bernhard.Lehner@jku.at) received his B.S. and M.S. degrees in computer science in 2007 and 2010, respectively, from Johannes Kepler University, Linz, Austria, where he is currently pursuing a Ph.D. degree. From 1991 to 2004, he was with Virginia Polytechnic Institute and State University, Lenze, Siemens, and Infineon. His research interests include signal processing, audio event detection, audio scene classification, music information retrieval, image processing, neural networks, and interpretable machine learning.

Anna Kruspe (anna.kruspe@dlr.de) received her diploma and Ph.D. degrees in media technology from Technische Universität Ilemnau, Germany, in 2011 and 2017, respectively. She is a machine-learning researcher at the German Aerospace Center. Previously, she was a member of the Fraunhofer Institute for Digital Media Technology, Ilmenau, Germany, where her work focused on the application of speech recognition technologies to singing (e.g., for language identification, keyword spotting, or lyrics-based search), as well as the analysis of world music. She conducted research at Johns Hopkins

University in Baltimore, Maryland, and at the National Institute of Advanced Industrial Science and Technology in Tsukuba, Japan. Her current work deals with the development of machine-learning technologies for the analysis of social media data in the context of disaster management.

Luwei Yang (luwei.yang.qm@gmail.com) received his B. Sci. degree in engineering with law, first class, from the Bejing University of Post and Telecommunications, China and his Ph.D. degree in electronics engineering at the Centre for Digital Music at Queen Mary University of London, under the supervision of Elaine Chew and Khalid Z. Rajab. He is currently a senior algorithm engineer at Alibaba Group, where his work focuses on the application of machine-learning and deep-learning techniques to the areas of recommender systems, natural language processing, and intelligent agents.

References

[1] P. Juslin and P. Laukka, "Communication of emotions in vocal expression and musical performance: Different channels, same code?" *Psych. Bul.*, vol. 129, pp. 770–814, 2003.

[2] B. Gold, N. Morgan, and D. Ellis, *Speech and Audio Signal Processing: Processing and Perception of Speech and Music*. Hoboken, NJ: Wiley, 2011.

[3] M. Goto, T. Saitou, T. Nakano, and H. Fujihara, "Singing information processing based on singing voice modeling," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, 2010, pp. 5506–5509.

[4] J. Bonada and X. Serra, "Synthesis of the singing voice by performance sampling and spectral models," *IEEE Signal Process. Mag.*, vol. 24, no. 2, pp. 67–79, 2007.

[5] M. Goto, "Singing information processing," in *Proc. 12th Int. Conf. Signal Processing (ICSP)*, 2014, pp. 2431–2438.

[6] A. Demetriou, A. Jansson, A. Kumar, and R. Bittner, "Vocals in music matter: The relevance of vocals in the minds of listeners," in *Proc. 19th Conf. Int. Society for Music Information Retrieval (ISMIR)*, 2018, pp. 514–520.

[7] J. T. Foote, "Content-based retrieval of music and audio," in *Proc. Multimedia Storage and Archiving Systems II, Int. Society for Optics and Photonics*, 1997, vol. 3229, pp. 138–148.

[8] B. Lehner, J. Schlüter, and G. Widmer, "Online, loudness-invariant singing voice detection in mixed music signals," *IEEE Trans. Audio, Speech, Language Process.*, vol. 26, no. 8, pp. 1369–1380, Apr. 2018.

[9] Z. Rafii, A. Liutkus, F.-R. Stoter, S. I. Mimilakis, D. FitzGerald, and B. Pardo, "An overview of lead and accompaniment separation in music," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 26, no. 8, pp. 1307–1335, 2018.

[10] B. Pardo, Z. Rafii, and Z. Duan, "Audio source separation in a musical context," in *Springer Handbook of Systematic Musicology*, R. Bader, Ed. New York: Springer-Verlag, 2018, pp. 285–298.

[11] J. Driedger and M. Müller, "Extracting singing voice from music recordings by cascading audio decomposition techniques," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, 2015, pp. 126–130.

[12] Y. Luo, Z. Chen, J. R. Hershey, J. L. Roux, and N. Mesgarani. (2016). Deep clustering and conventional networks for music separation: Stronger together. arXiv. [Online]. Available: https://arxiv.org/abs/1611.06265

[13] D. Ward, R. D. Mason, R. C. Kim, F.-R. Stöter, A. Liutkus, and M. D. Plumbley, "SISEC 2018: State of the art in musical audio source separation-subjective selection of the best algorithm," in *Proc. 4th Workshop on Intelligent Music Production*, Huddersfield, United Kingdom, 2018, this workshop does not produce paginated proceedings.

[14] J. Schlüter and T. Grill, "Exploring data augmentation for improved singing voice detection with neural networks," in *Proc. 16th Conf. Int. Society for Music Information Retrieval (ISMIR)*, 2015, pp. 121–126.

[15] P. Proutskova, C. Rhodes, T. Crawford, and G. Wiggins, "Breathy, resonant, pressed–automatic detection of phonation mode from audio recordings of singing," *J. New Music Res.*, vol. 42, no. 2, pp. 171–186, 2013.

[16] J. Wilkins, P. Seetharaman, A. Wahl, and B. Pardo, "Vocalset: A singing voice dataset," in *Proc. 19th Conf. Int. Society for Music Information Retrieval (ISMIR)*, 2018, pp. 468–474.

[17] R. C. Repetto, R. Gong, N. Kroher, and X. Serra, "Comparison of the singing style of two jingju schools," in *Proc. 16th Conf. Int. Society for Music Information Retrieval (ISMIR)*, 2015, pp. 507–513.

[18] H. Fujihara, M. Goto, T. Kitahara, and H. G. Okuno, "A modeling of singing voice robust to accompaniment sounds and its application to singer identification and vocal-timbre-similarity-based music information retrieval," *IEEE Trans. Audio, Speech, Language Process.*, vol. 18, no. 3, pp. 638–648, 2010.

[19] G. K. Koduri, V. Ishwar, J. Serrà, and X. Serra, "Intonation analysis of rāgas in Carnatic music," *J. New Music Res.*, vol. 43, no. 1, pp. 72–93, 2014.

[20] J. Salamon, E. Gómez, D. P. W. Ellis, and G. Richard, "Melody extraction from polyphonic music signals: Approaches, applications and challenges," *IEEE Signal Process. Mag.*, vol. 31, no. 2, pp. 118–134, Mar. 2014.

[21] S. Balke, C. Dittmar, J. Abeßer, and M. Müller, "Data-driven solo voice enhancement for jazz music retrieval," in *Proc. Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, Mar. 2017, pp. 196–200.

[22] F. Rigaud and M. Radenen, "Singing voice melody transcription using deep neural networks," in *Proc. 17th Conf. Int. Society for Music Information Retrieval (ISMIR)*, 2016, pp. 737–743.

[23] R. M. Bittner, B. McFee, J. Salamon, P. Li, and J. P. Bello, "Deep salience representations for f0 estimation in polyphonic music," in *Proc. 18th Conf. Int. Society for Music Information Retrieval (ISMIR)*, Oct. 2017, pp. 63–69.

[24] K. K. Ganguli and P. Rao, "Discrimination of melodic patterns in indian classical music," in *Proc. IEEE 21st Nat. Conf. Communications (NCC)*, 2015, pp. 1–6.

[25] O. Nieto, "Unsupervised clustering of extreme vocal effects," in *Proc. 10th Int. Conf. Advances in Quantitative Laryngology*, 2013, p. 115.

[26] S. Gulati, "Computational approaches for melodic description in indian art music corpora," Ph.D. dissertation, Music Tech. Group, Universitat Pompeu Fabra, Barcelona, Spain, 2016.

[27] D. Gärtner, "Singing/rap classification of isolated vocal tracks," in *Proc. 11th Conf. Int. Society for Music Information Retrieval (ISMIR)*, 2010, pp. 519–524.

[28] M. Panteli, R. Bittner, J. P. Bello, and S. Dixon, "Towards the characterization of singing styles in world music," in *Proc. Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, 2017, pp. 636–640.

[29] A. Kumar, R. M. Bittner, N. Montecchio, A. Jansson, M. Panteli, E. J. Humphrey, and T. Jehan, "Learning a large-scale vocal similarity embedding for music," in *Proc. Machine Learning for Music Discovery Workshop, Int. Conf. Machine Learning*, 2017.

[30] W.-H. Tsai and H.-M. Wang, "Towards automatic identification of singing language in popular music recordings," in *Proc. 5th Conf. Int. Society for Music Information Retrieval (ISMIR)*, Barcelona, Spain, 2004, pp. 568–576.

[31] A. Loscos, P. Cano, and J. Bonada, "Low-delay singing voice alignment to text," in *Proc. Int. Computer Music Conf. (ICMC)*, 1999, pp. 437–440.

[32] V. Chandrasekhar, M. E. Sargin, and D. A. Ross, "Automatic language identification in music videos with low level audio and visual features," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, 2011, pp. 5724–5727.

[33] T. Nakano and M. Goto, "Vocarefiner: An interactive singing recording system with integration of multiple singing recordings," in *Proc. Conf. Sound and Music Computing (SMC)*, 2013, pp. 115–122.

[34] M. Kan, Y. Wang, D. Iskandar, T. L. Nwe, and A. Shenoy, "LyricAlly: Automatic synchronization of textual lyrics to acoustic music signals," *IEEE Trans. Audio, Speech, Language Process.*, vol. 16, no. 2, pp. 338–349, 2008.

[35] H. Fujihara and M. Goto, "Lyrics-to-audio alignment and its applications," in *Multimodal Music Processing*, M. Müller, M. Goto, and M. Schedl, Eds. Schloss Dagstuhl, Germany: Dagstuhl Publishing, 2012, pp. 23–36.

[36] H. Fujihara, M. Goto, J. Ogata, and H. G. Okuno, "Lyricsynchronizer: Automatic synchronization system between musical audio signals and lyrics," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 6, pp. 1252–1261, 2011.

[37] M. Mauch, H. Fujihara, and M. Goto, "Integrating additional chord information into HMM-based lyrics-to-audio alignment," *IEEE Trans. Audio, Speech, Language Process.*, vol. 20, no. 1, pp. 200–210, 2012.

[38] T. Hosoya, M. Suzuki, A. Ito, and S. Makino, "Lyrics recognition from a singing voice based on finite state automaton for music information retrieval," in *Proc. 6th Conf. Int. Society for Music Information Retrieval (ISMIR)*, 2005, pp. 532–535.

[39] M. McVicar, D. P. W. Ellis, and M. Goto, "Leveraging repetition for improved automatic lyric transcription in popular music," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, 2014, pp. 3117–3121.

[40] A. M. Kruspe, "Bootstrapping a system for phoneme recognition and keyword spotting in unaccompanied singing," in *Late-Breaking Workshop, 17th Conf. Int. Society for Music Information Retrieval (ISMIR)*, New York, NY, 2016.

[41] H. Hirjee and D. G. Brown, "Solving misheard lyric search queries using a probabilistic model of speech sounds," in *Proc. 11th Conf. Int. Society for Music Information Retrieval (ISMIR)*, 2010, pp. 147–152.

[42] D. Ward, H. Wierstorf, R. Mason, E. M. Grais, and M. Plumbley, "BSS Eval or PEASS? Predicting the perception of singing-voice separation," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, 2018, pp. 596–600.

[43] S. Dixon, M. Goto, and M. Mauch, "Why is voice interesting?" in *Proc. 16th Conf. Int. Society of Music Information Retrieval*, 2015.

SP

Karthika Vijayan, Haizhou Li, and Tomoki Toda

Speech-to-Singing Voice Conversion

The challenges and strategies for improving vocal conversion processes

©ISTOCKPHOTO.COM/TRAFFIC_ANALYZER

Digital Object Identifier 10.1109/MSP.2018.2875195
Date of publication: 24 December 2018

Speech-to-singing (STS) conversion is the task of converting the read lyrics of a song, spoken in natural manner, to proper singing. The most important aspect of the task is to change the prosody of the natural speech to match with that of proper singing, while retaining the linguistic content and the speaker's identity. STS conversion is a challenging task because speaking and singing are different in many ways.

Introduction

STS conversion is an enabling technology for many innovative services and applications. It can be employed to beautify the singing renditions of amateur singers with inadequate singing skills, if we consider STS as an extreme scenario of converting bad singing to good-quality singing. It can be used to automatically generate reference singing for vocal learners and to personalize singing synthesis. As the state-of-the-art singing synthesis systems, such as Vocaloid [1] and Realivox [2], mostly generate singing vocals in a fixed voice, the idea of personalized singing is certainly appealing to the public, hence creating a strong commercially motivated drive. Besides its applications in the entertainment industry, STS conversion also serves as a bridge between speech and singing analyses. It provides valuable insights into the production and perception of speech and singing voices that are useful in many music information processing applications.

Though STS conversion is eagerly sought after, its realization is far from easy. Speech and singing vocals are produced by the same human voice production system, and hence, they share several similar characteristics. However, because of the distinctive vocal production processes between speaking and singing, they manifest through different acoustic characteristics [3]–[5]. STS conversion has to address several research problems including the temporal alignment between two very different signals, mapping of dissimilar acoustic characteristics, preserving fine attributes of speakers during conversion, and so on. All these make STS conversion a challenging task.

In this article, we describe the major challenges that need to be overcome for effective STS conversion. We first present the fundamental differences between speech and singing voices. We

then look into the methodology of STS conversion by introducing two prominent technical frameworks, i.e., template- and model-based approaches. Later, we present the evaluation strategies to assess the quality of synthesized singing vocals using objective and subjective measures. Finally, we summarize the tools and resources currently available for STS conversion study, and we discuss some implementation issues and future directions.

Speech versus singing

The major challenges in STS conversion stem from the differences between speaking and singing. The human voice production system that generates speech and singing signals can be effectively described by the source-filter model [3]. A comparative study between speech and singing can be performed by analyzing the different characteristics of glottal excitation and the vocal tract system.

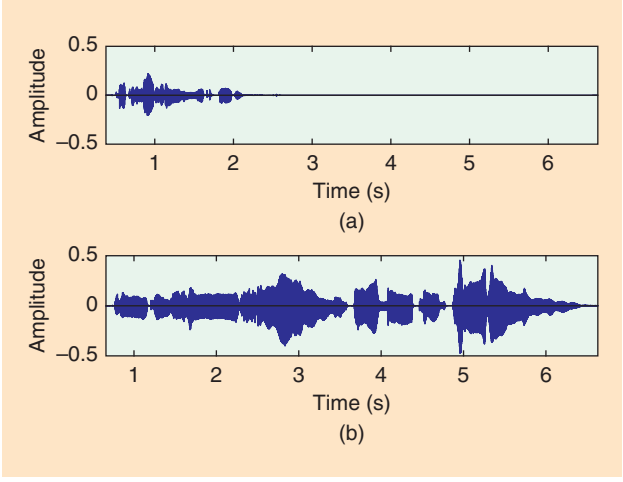


FIGURE 1. An illustration of the differences in vocal effort and dynamic ranges of amplitude between speech and singing vocals for "take a sad song and make it better," uttered by the same person: (a) the speech vocal and (b) the singing vocal.

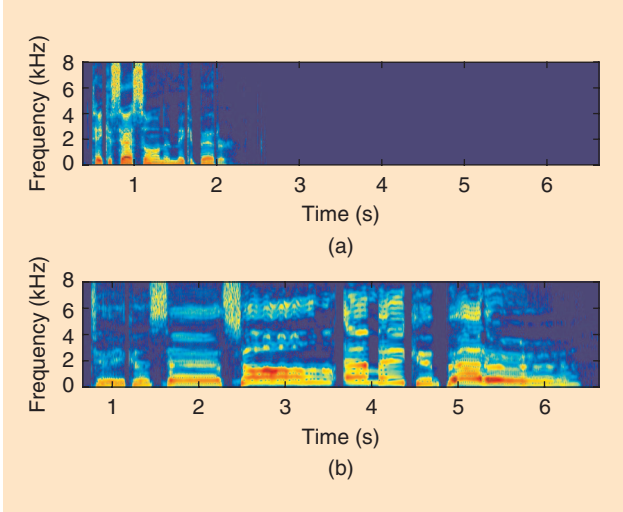


FIGURE 2. An illustration of the differences in spectral characteristics between speech and singing, corresponding to the utterances in Figure 1: (a) a spectrogram of the speech vocal and (b) a spectrogram of the singing vocal.

Singing requires a higher level of vocal effort, more active breathing during exhalation, and a larger range of variation in loudness than speaking. The dynamic range of short-time energy (instantaneous amplitude) of singing is thus larger than that of speech. This is achieved by trained singers with the careful management of subglottal pressure [6]. Figure 1 illustrates the difference between amplitudes of a speech and a singing sample.

For singing of high-level vocal effort, such as opera singing, a singer may not be able to rely only on the subglottal pressure variations because of physiological constraints. In such cases, a trained singer places his/her larynx in a particularly raised position by changing the form of articulators in the back end of the oral cavity, forcing the formants in the high-frequency section of the voice spectrum to cluster together. This peculiarly strong formant in the high-frequency spectrum of the singing voice is termed the *singing formant* [4]. The occurrence of the singing formant can be observed from Figure 2(b) (between 2 and 4 kHz from 2.5 to 3.5 s), which is absent in the corresponding speech spectrum in Figure 2(a). A trained singer, thus, manipulates the subglottal pressure with increased flexibility and introduces a singing formant efficiently to produce good-quality singing.

Smooth and soothing pitch variation is another integral factor of singing. A trained singer can carefully control the subglottal air pressure and volume to change the rate of vibrations of the vocal folds at the glottis [5]. A fine variation of pitch rendering adds expression to the singing, e.g., vibrato, preparation, and overshoot. Vibrato is manifested as quasi-periodic frequency modulations in pitch contour, overshoot is a deflection exceeding the target note observed after a note change and preparation is a deflection occurring just before a note change in the direction opposite to that of the note change [7]. Aided by the melody of singing, a singer produces a smooth fundamental frequency (F0) contour (pitch contour) with a larger dynamic range than that of speech signals, as can be observed in Figure 3(a) and (b). The fine characteristics of pitch of singing vocals are illustrated in Figure 3(b), as vibrato (between 2.5 and 3.5 s), overshoot (around 1.2 s), and preparation (around 1.5 and around 4 s). The particulars of pitch in singing affect the amplitudes and frequencies of formants as well, which results in the modulation of formants by the F0 contour [5]. Finally, singing follows a specific rhythm and melody by sustaining the vowels over the required duration that speaking does not require.

In terms of signal generation, singing has a close affinity to speech. They are both produced through the human voice production system, sharing many common acoustic properties. Therefore, the singing voice processing techniques have benefited from the recent advances in speech processing. The major differences between speaking and singing are manifested in characteristics of glottal excitation (subglottal pressure, pitch, and strength of excitation), the vocal tract system (singing formant), coupling between the vocal tract and excitation (modulation of formants by pitch), and the duration of voicing [3]. Such differences call for studies to adapt the parameterization and acoustic modeling methods from the speech voice to the

singing voice. In short, the singing voice presents a unique set of research problems that deserve systematic study.

STS conversion

An STS conversion system is designed to take the read speech from a user as input (user speech) and generate prosody characteristics (singing prosody) for synthesis of singing vocals as output (synthesized singing). The basic idea of STS conversion includes the manipulation of parameters of user speech with respect to the reference prosody of the song, according to some predesigned transformation schemes. The transformed parameters that resemble those of singing are then used to generate output singing vocals. Because singing adheres to specific rhythm and melody, sustained notes, and vibrato, it is described by a structured prosody pattern. An important aspect of STS conversion is to transform the prosody of user speech into that of singing. Another equally important aspect is the preservation of the user's speaker identity, such as spectral characteristics, during the transformation.

The STS conversion techniques can be broadly classified into two categories, i.e., the template-based conversion and the model-based conversion, depending on how the reference prosody of singing is generated. The template-based framework uses reference prosody as a template that is extracted from high-quality singing. The parameters of user speech are then converted to those of singing vocals, using learned mapping schemes [8], [9]. On the other hand, the model-based framework generates singing prosody via prosody control models that are built on reference prosody described by musical scores, such as Musical Instrument Digital Interface (MIDI), and prior knowledge, such as musical pitch transitions and vibrato. With the mapping schemes or models, the parameters of user speech are then converted to those of singing vocals [10], [11]. In Figure 4, we illustrate the two general frameworks.

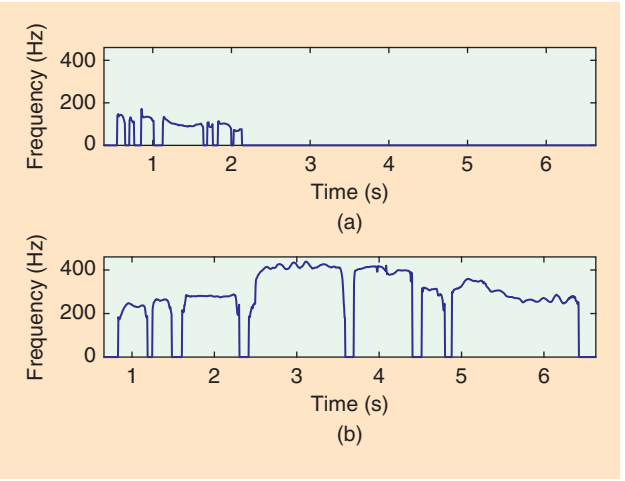


FIGURE 3. An illustration of the differences in the F0 contour between speech and singing, corresponding to the utterances in Figure 1: (a) the F0 contour of the speech vocal and (b) the F0 contour of the singing vocal.

The two conversion frameworks share a similar workflow. We convert the acoustic parameters from user speech to singing, then we synthesize the singing with a vocoder. The major difference between the two frameworks lies in the way they generate the singing prosody for synthesized singing from the reference. Once we have generated the intended singing prosody, the two frameworks face several common challenges. As illustrated in Figure 4, first we need to align the user speech to the musical rhythm and melody temporally [10]–[12]. We then map the phonetic content from user speech to synthesized singing without losing the speaker's identity. Finally, singing is a performing

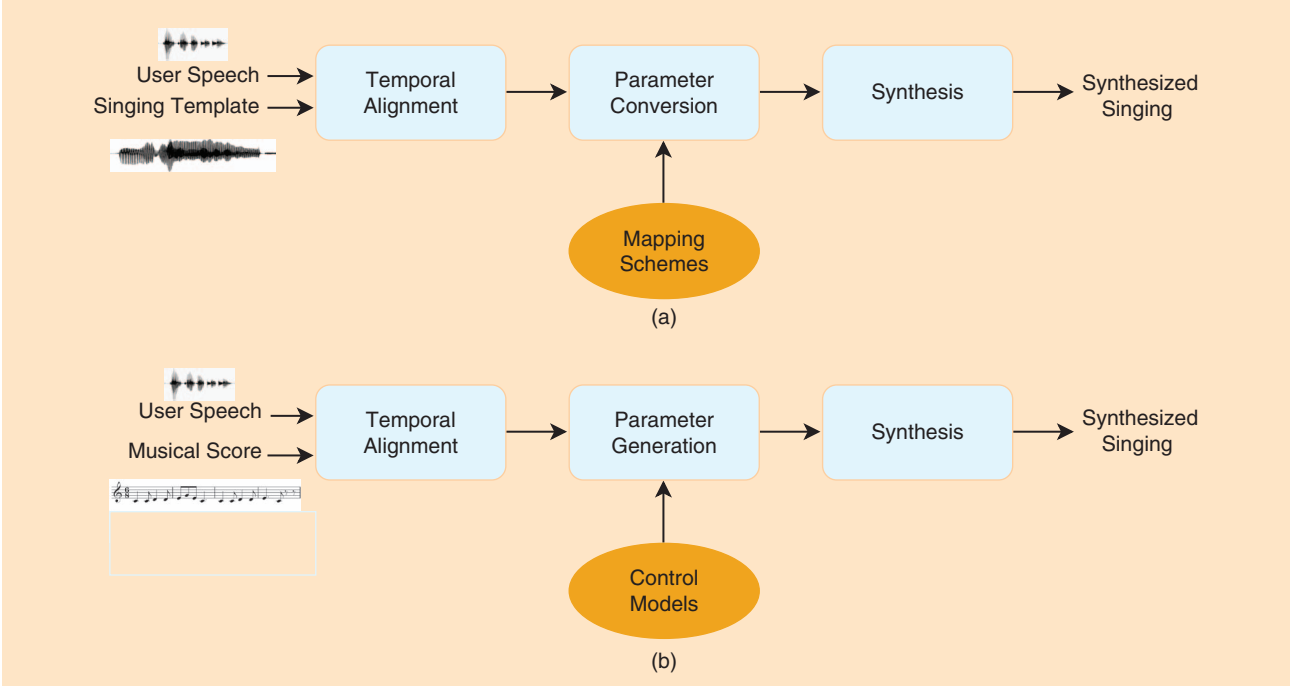


FIGURE 4. The (a) template- and (b) model-based frameworks for STS conversion.

art and a creative process that express an emotion, feeling, or taste that may not exist in the user's speech. The study of STS conversion is also about how to enable such expressions. Next, we discuss the fundamentals of the two conversion frameworks in detail.

Template-based STS conversion

The template-based approach assumes that a well-sung vocal is available. It uses the prosody derived from a natural singing vocal as the reference, thus minimizing the possible prosody errors in the synthesized singing. An important processing in the template-based approach is to synchronize the words between the user's speech and the singing template in time, which we call temporal alignment. The synchronization information obtained from the temporal alignment are needed for the subsequent frame-level parameter conversion.

Temporal alignment

Aligning linguistic content between the user's speech and the singing template is a crucial first step toward an accurate STS conversion because errors in temporal alignment are perceivable as distortions in synthesized singing. While manual alignment offers a high accuracy in general, it is not practical in real-time applications. Dynamic time warping is an effective algorithm for time-series alignment. The challenge is how to overcome the mismatch between speech and singing [13].

A dual alignment scheme was recently studied for effective speech to singing alignment [9], [13]. The dual alignment utilizes the read speech of lyrics of songs by the trained singer (singer's speech) to build a bridge between the user's speech and the singing template. Notice that we only need to manually align the singing template with the singer's speech once, which offers near perfect accuracy [13]. In dual alignment, the user's speech is first aligned with the singer's speech using dynamic time warping. Once such speech-to-speech alignment is obtained, the STS alignment can be established automatically. The problem of temporal alignment between the user's speech and the singing template is thus reduced to the alignment between two sets of speech signals via the user's speech and singer's speech.

To accurately align the user's speech to a singing template, we benefit from the understanding of speech and singing characteristics. The major differences between speech and singing are constituted by the properties of the glottal excitation source and the singing formant [4], [5]. It is advantageous to extract features that only represent the common properties between speech and singing, such as the voice activity contour and the lexical pause, and to remove their differences. Because the speech and singing share the same linguistic content, one can explore the use of multiple features, which we call *tandem features*, to represent their common characteristics. The analysis of signals for extraction of tandem features may include the following steps:

1) normalizing the short-time energy over the signals to nullify energy variations
2) performing source-filter decomposition to obtain smoothed spectral envelope, thus removing the glottal excitation source characteristics
3) restricting the smoothed spectrum to the low-frequency region to avoid the singing formant.

It has been observed that the temporal alignment performance of tandem features is superior to traditional features, such as the mel-frequency cepstral coefficients and the linear prediction cepstral coefficients [14].

Parameter extraction and conversion

Prominent signal analysis techniques, such as source-filter decomposition, can be employed to extract the parameters of the glottal excitation source and the vocal tract system from speech and singing signals. The parameters are then converted from the user's speech to the singing template and passed to the vocoder to generate time-domain signals. Because the analysis and conversion of the parameters of the signals are more effective in the spectral domain, we apply a short-time analysis to obtain the spectral parameters.

For the conversion of parameters from speech to singing, the characteristics of the excitation source and vocal tract system are considered separately. The properties of the glottal excitation source mostly contribute to the prosody and, hence, follow the melody of the song in singing voices. The properties of the vocal tract system, on the other hand, mostly characterize the timbre and provide valuable cues on speaker identity. The template-based framework extracts the F0 contour, representing excitation source parameters, from the singing template and retains them as the reference prosody for STS conversion. The smoothed spectral envelope, representing vocal tract system parameters, is extracted from the user's speech to preserve the speaker identity of the user in synthesized singing, except that these spectral characteristics have to be modified to resemble those of singing vocals. This is a particularly challenging task, involving speaker-dependent mapping of spectral characteristics from speaking to singing styles.

A simple solution to such speaker-dependent mappings can be provided by inducing the properties of the singing spectrum onto the speech spectrum. The most important spectral characteristics of singing voices are identified as the singing formant and the amplitude modulations of formant trajectories by the F0 contour [3]–[5]. The effect of the singing formant is introduced by emphasizing the speech spectrum around 3 kHz by a frequency-weighting function that resembles a bandpass filter. Also, the amplitude modulation in the temporal envelope of the speech signal at each vibrato in the F0 contour is estimated and added to the temporal envelope of the synthesized singing [10], [11]. The parameters for such manipulations can be obtained a priori empirically. Generally, a fixed set of control parameters do not provide a justified representation of a wide range of singing vocals in different genres. To estimate the parameters, other advanced techniques can be employed, such as partial least squares, Gaussian mixture models, exemplar-based representations, frequency warping, and deep learning [15]. While such techniques were studied for speaker identity conversion in general, they can be repurposed for voice conversion from speech to singing [16]. For training these voice conversion schemes, parallel databases of spoken and sung utterances that are temporally aligned at the frame level are generally required [17], [18].

To summarize, the converted parameters consist of the excitation source parameters from the singing template, and the spectral parameters are adjusted for singing from the user's speech. The source parameters represent the reference prosody of the song and are used as singing prosody for synthesized singing. The spectral parameters are converted from the user's speech, which carries forward the speaker's identity [8], [9], [13]. In Figure 5, we illustrate three spectrograms that are involved in the process. The smoothed spectral characteristics of the user's speech are preserved in synthesized singing, while the pitch harmonics resemble those in the singing template.

Model-based STS conversion

Instead of a singing template, the model-based framework uses musical scores, such as MIDI files, as the source of reference. The musical scores define accurate pitch transitions in the melody of singing vocals. Similar to the template-based conversion, the model-based conversion framework needs to align the user's speech to musical scores. The singing prosody and spectral parameters are generated from the musical score and the user's speech, respectively, by suitable control models [10], [11].

Temporal alignment

As a part of song writing, the lyrics writers align the lyrical words with the musical notes. Such alignment information is available for model-based STS conversion [10], [11]. Therefore, the actual task of temporal alignment is to align the user's speech to the lyrical words, either manually or with automatic speech recognition. Once the user's speech is aligned with the lyrical words, it is also aligned with the musical notes.

Parameter generation

The parameters from the user's speech can be extracted in the same way as that in the template-based framework. However, in the model-based framework, the excitation source parameters representing singing prosody (F0 contour and fine characteristics of pitch) are generated from the synthetic musical score.

The F0 contour for synthesized singing is generated from the reference prosody described by musical scores using an F0 control model. This control model transforms the unnaturally flat and discontinuous synthetic musical score into a human-realistic and continuous F0 contour, resembling that of natural singing vocals [19]. This can be achieved by inducing pitch variations found in natural singing onto the synthetic musical score [7]. The major characteristics of the F0 contour in singing voices are the gross representation of pitch in the melody and fine variations in the pitch that beautify the singing [5]. The fine attributes of the F0 contour include the vibrato, overshoot, preparation, and fine fluctuations [7]. With the F0 control model, we generate these fine attributes that are aligned with the musical scores to form a realistic F0 contour.

Vibrato is manifested as quasi-periodic frequency modulation in the F0 contour and can be generated using a second-order oscillatory system producing quasi-periodic sinusoidal variations. Overshoot and preparation are exhibited in the F0 contour as deflections associated with changes in musical notes. They are generated by second-order damping systems producing deflections of different polarities in the F0 contour [10], [11]. The parameters of these systems are estimated by least squares approximation of synthetic F0 contours with respect to natural F0 contours. The fine fluctuations in F0 are generated by high-pass filtering an amplitude-normalized white noise signal. The synthetically generated vibrato, overshoot, preparation, and fine fluctuations are added to the musical score to produce a human-realistic F0 contour that is later used as a singing prosody to generate synthesized singing outputs [10], [11].

Template-based versus model-based frameworks

The template-based and model-based frameworks mainly differ in the way they generate the singing prosody. The template-based framework employs the reference prosody extracted from the actual singing. On the other hand, the model-based framework generates the singing prosody from reference to musical scores using control models.

The template-based framework benefits from the near-perfect reference prosody that is directly obtained from the natural singing vocals. The reference prosody retains the pitch-rendering techniques in human singing such as vibrato, overshoot, and preparation. The model-based framework relies on the quality of the control models. In general, the control models generate singing prosody for output singing vocals that are less expressive than natural singing.

Yet, in practical implementations, recording high-quality reference singing templates for every song with variations, such

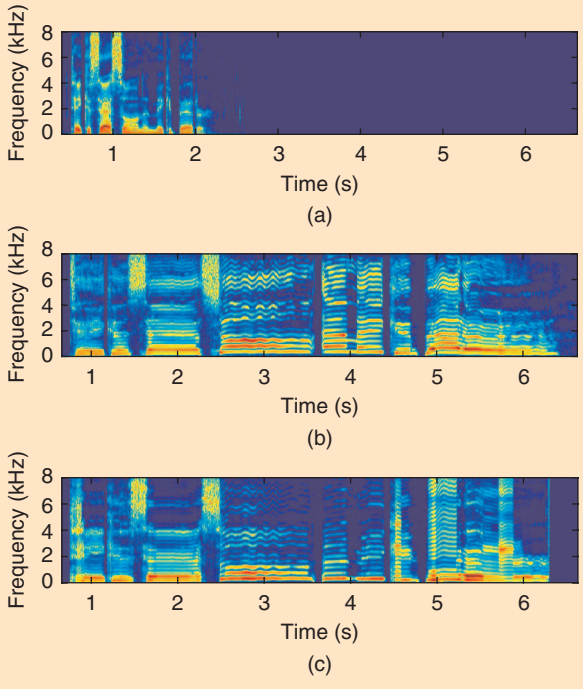


FIGURE 5. An illustration of the spectrograms in the template-based STS conversion: (a) the user's speech, (b) the singing template, and (c) the synthesized singing.

as classical singing versus Broadway-style singing or male versus female rendering, proves to be a tedious task. In contrast, preparing synthetic musical scores for a song will be relatively easier. Hence, the model-based framework is more scalable to a large song database in practical system deployment.

Evaluation of singing quality

The evaluation of perceptual quality of synthesized singing is necessary for the development of more sophisticated STS systems. The evaluation of singing quality is a research topic in itself. The perceptual quality of singing is generally evaluated using subjective tests in which listeners score the singing signals with respect to their intonation, rhythm, voice quality, and pronunciation. However, human listeners may not always be available. In such a scenario, objective evaluation strategies for the assessment of singing quality become extremely useful.

Objective evaluation

In the three processing modules of the processing pipeline in Figure 4, we evaluate the output of the modules against their respective ground truth in the objective evaluation. The evaluation of the synthesis module, also called the *vocoder*, is worth a separate study in speech synthesis [20], [21]. Here, we only discuss the evaluation of temporal alignment and parameter conversion.

The accuracy of temporal alignment between lyrical words of the user's speech and the target singing affects parameter conversion and, consequently, the perceptual quality of the synthesized singing. Let us take the template-based conversion as an example. We use the accuracy of temporal alignment between the user's speech and the singing template as one of the objective evaluation metrics.

The performance of temporal alignment can be reported over a database of parallel spoken-sung signals. The syllable-level and word-level manual transcriptions for each spoken and sung audio pair are required to set up the ground truth. The evaluation of temporal alignment between speech and singing signals can be reported using the average word boundary error, which is defined as the timing error between the ground truth and the aligned word boundaries [13]. A more detailed evaluation metric for temporal alignment can be defined as the timing error between the ground truth and the aligned syllable boundaries, which is termed the *average syllable boundary error*. To throw further insight into the effectiveness of a temporal alignment strategy, several other metrics can be defined, e.g., the percentage of gross alignment error or the percentage of syllable/word boundaries causing alignment errors of less than an acceptable error threshold.

The accuracy of parameter conversion is another contributing factor to the perceptual quality of synthesized singing. In parameter conversion, we convert the spectra of user speech to those of singing while preserving the speaker's identity. The resultant spectral characteristics represent the voice quality and pronunciation of the lyrics in synthesized singing. We may want to evaluate the parameter conversion in two aspects. One is to assess how well synthesized singing maintains the user's speaker identity. To do this, we can use techniques in speaker verification such as i-vector [22] and x-vector [23] to compare the speaker characteristics between the synthesized singing with the user's speech. Another factor is to assess the perceptual quality of the synthesized singing. A recent study on perceptual evaluation of singing quality (PESnQ) [24] suggests a systematic approach to the problem without the need of a reference singing. The PESnQ technique evaluates singing quality with a set of parameters covering pitch, rhythm, vibrato, voice quality, pronunciation, and volume, which provides objective evaluation close to human judgment.

Subjective evaluation

Assessing the perceptual quality of singing vocals can be done best by human evaluation. Upon availability of an expert panel of judges, samples of synthesized singing can be evaluated for their quality and various attributes. When an expert panel of judges is not available, the perceptual evaluation is conducted by collecting opinions of average listeners, who are music enthusiasts able to appreciate singing and notions of pitch, rhythm, and intonation. The common measures employed for perceptual evaluation of audio samples include the mean opinion score (MOS) and the best-worst score (BWS).

To provide an MOS, the listeners are asked to rate each synthesized sample according to a rating scale for assessing perceptual quality. They are typically asked to rate with respect to two factors:

1) how well the speaking user's identity is manifested in synthesized singing
2) how well the attributes of singing quality from the singing template are expressed in synthesized singing.

In the latter case, the listeners may be invited to evaluate different aspects of singing quality, e.g., voice quality, intonation, rhythm, intensity variations, and so on.

In many experiments, we have the need to compare among several synthesized singing samples. We use comparative statistics to rank the human preferences. For example, we use preference tests such as AB or ABX to identify detectable differences between the perceptual quality of samples. We know that humans are good at identifying the extremes, but their ability in ranking the preferences to fine details is limited. The BWS is a solution to provide comparative perceptual quality of samples. It was proposed for applications in economic surveys to evaluate the quality of products [25]. In STS conversion, the listeners are asked to choose the best and worst sounding samples from a set of audio signals after repeated listening to samples in all possible permutations [13]. If the sample i has appeared in many comparative groups, a BWS can be calculated for sample i by aggregating the statistics as, $\mathrm{BWS}_i = (B_i - W_i)/N_i$, where B_i and W_i denote the number of times the item i is chosen as best and worst, respectively, by listeners. N_i represents the total number of appearances of item i in the set of trials [13], [25]. A more positive BWS indicates that the corresponding sample is more appealing to the listeners. The combination of MOS and BWS can reveal information about absolute and relative perceptual quality of singing samples synthesized by an STS conversion system.

Implementation issues

We have discussed the common frameworks for STS conversion and how to evaluate the quality of their outputs. In real-world applications, we also face many other technical challenges. We now discuss the issues concerning the input and the output of the system.

As the inputs to the system, the user's speech and/or the singing template are typically assumed to be pure vocals recorded in noise-free circumstances. However, the user's speech is often corrupted by noise, while the singing template may come with background music. In such cases, other signal processing modules such as speech enhancement and vocals–music separation are needed as a front-end application to the processing pipeline to prepare the pure vocals.

It is noted that a user's speech at runtime can be spontaneous or impromptu in a way that does not exactly follow the reference lyrical words. In such cases, the forced temporal alignment discussed in the section "STS Conversion" will fail. This calls for a smart temporal alignment algorithm that is able to strategically distribute the spoken content to the target song with the designated singing prosody for the best effect. In practice, spoken words can be obtained via an automatic speech recognition system, while singing rhythm information, such as the timing of notes, beats, and onsets, can be acquired from musical scores or via music information processing techniques. While the systems generate dry vocals, typically for final applications, the vocals are mixed with music in the same way that a recorded vocal would be mixed, i.e., processed further by audio production techniques, such as room reverberation [26], noise shaping [27], and audio equalization to improve the listening quality.

Tools and resources

Singing databases are necessary for the study of temporal alignment and transformation schemes. We report two databases having parallel recordings of read lyrics and singing vocals by trained singers. The National University of Singapore (NUS) sung and spoken lyrics corpus consists of 48 English songs of pop genre, read and sung by 12 singers. The manually marked phoneme labels for all recordings are also available as part of this database [17]. Similarly, the NUS–human language technology spoken lyrics and singing corpus contains 100 English pop songs, read and sung by ten singers. For the recordings of speech and singing, manually marked sentence transcriptions are also provided [18]. These databases can be used to study algorithms for temporal alignment and parameter conversion between speaking and singing. In addition, databases of singing vocals are needed for the training of transformation schemes, control models, and singing vocoders. Examples of such databases include the Smule database, the Center for Research in Entertainment and Learning database from the University of California, San Diego [31], the Basque database [32], the Isophonics singing voice data set [33], and the RAVDESS database from Science of Music, Auditory Research, and Technology laboratory at Ryerson University [34].

We also rely on computer-assisted processing tools and utilities. Wavesurfer and Praat are useful tools for analysis, visualization, and editing of audio signals. Plug-ins can be developed in these tools for automatic pitch tracking, which can be used for extraction of the F0 contour from the singing templates.

Effective parameterization of source and spectral properties is critical to the STS conversion task. This is achieved by an analysis–synthesis framework. During analysis, we obtain short-time frequency-specific parameters of the voice production system from speech or singing, i.e., the F0 contour and the smoothed spectrum. During synthesis, we reconstruct time-domain signals from these parameters. The speech transformation and representation by adaptive interpolation of weighted spectrogram (STRAIGHT) analysis represents an effective solution to the required analysis-synthesis [20]. It computes the smoothed spectral envelope (SP) representing the vocal tract system, and the F0 and the aperiodicity component (AP) representing the glottal excitation characteristics. There have been several alternatives to STRAIGHT. WORLD [21] presents a vocoding alternative that is computationally more efficient than STRAIGHT. A WaveNet vocoder trained on STRAIGHT parameters offers a more natural voice quality [28]. Note that a speaker-independent WaveNet vocoder involves training on a considerable amount of training samples. However, a speaker-adaptive WaveNet vocoder can offer a high-quality target voice with a quick adaptation process [29].

An iPhone application, named *Sing 4 Singapore*, was announced in 2014 [30] as the first near real-time STS conversion implementation that offers three English and two Chinese songs. This application allows a user to read the lyrics of songs, one line at a time. The application converts the read lyrics into the singing vocals and plays back the song in the user's voice together with the background music, as soon as the user finishes the reading [30].

Conclusions and future directions

In this article, we summarized the challenges in STS conversion that are created by the differences between speaking and singing. We presented the two major frameworks for STS conversion, which differ from each other in the way they generate the singing prosody. With the advent of STS conversion technology, we advocate that everyone can sing like a professional.

While STS conversion has made major progress recently, many research problems remain to be resolved. To improve the perceptual quality of synthesized singing, better temporal alignment and parameter conversion are expected. Furthermore, many applications may require real-time implementation of the algorithms that have not been well studied in the past.

The state-of-the-art STS conversion is mostly implemented as a modular system that consists of multiple signal processing modules in a processing pipeline. The recent studies on deep learning over large database have opened up opportunities in many ways. Inspired by the success in other signal processing pipelines, we consider that an end-to-end architecture for STS conversion will allow us to optimize the process in a systematic manner by reducing the artifacts introduced by the individual modules. We foresee that a deep-learning approach to STS conversion will become an active topic in the near future.

Authors

Karthika Vijayan (vijayan.karthika@nus.edu.sg) received her Ph.D. degree from the Indian Institute of Technology (IIT) Hyderabad, in 2016. She is a research fellow at the Department of Electrical and Computer Engineering, National University of Singapore. Her research interests include speech and singing signal processing and characterization. She is a member of the International Speech Communication Association and the Asia Pacific Signal and Information Processing Association (APSIPA). She received several awards including Research Excellence from IIT Hyderabad (2014 and 2015) and Springer book prize at the 2017 APSIPA–Annual Summit and Conference. She is a Member of the IEEE.

Haizhou Li (haizhou.li@nus.edu.sg) received his Ph.D. degree from South China University of Technology, Guangzhou, in 1990. He is a professor in the Department of Electrical and Computer Engineering, National University of Singapore. His research interests include speech information processing and natural language processing. He is currently the editor-in-chief of *IEEE/ACM Transactions on Audio, Speech, and Language Processing* (2015–2018). He has served as the president of the International Speech Communication Association (2015–2017), and the president of Asia Pacific Signal and Information Processing Association (2015–2016). He is a Fellow of the IEEE.

Tomoki Toda (tomoki@icts.nagoya-u.ac.jp) received his D.E. degree from Nara Institute of Science and Technology, Japan, in 2003. He is a professor of the Information Technology Center at Nagoya University, Aichi, Japan. His research interests include speech, music, and sound processing. He has served as a member of the Speech and Language Technical Committee of the IEEE Signal Processing Society (SPS) (2007–2009, 2014–2016) and as an associate editor of *IEEE Signal Processing Letters* (2016–2018). He has received more than ten paper and achievement awards including the IEEE SPS 2009 Young Author Best Paper Award and the 2013 European Association for Signal Processing–International Speech Communication Association Best Paper Award (from *Speech Communication*). He is a Member of the IEEE.

References

[1] H. Kenmochi and H. Ohshita, "VOCALOID - commercial singing synthesizer based on sample concatenation," in *Proc. INTERSPEECH*, Antwerp, Belgium, Aug. 2007, pp. 4009–4010.

[2] Wikipedia. (2008). Realivox. [Online]. Available: https://en.wikipedia.org/wiki/Realivox

[3] B. Lindblom and J. Sundberg, "The human voice in speech and singing," in *Springer Handbook of Acoustics*, New York: Springer, Jan. 2014, pp. 703–746.

[4] J. Sundberg, "The level of the 'singing formant' and the source spectra of professional bass singers," *Q. Progress Status Report: STL-QPSR*, vol. 11, no. 4, pp. 21–39, Jan. 1970.

[5] H. Fujisaki, "Dynamic characteristics of voice fundamental frequency in speech and singing," in *The Production of Speech*, P. F. MacNeilage, Ed. New York: Springer, 1983, pp. 39–55.

[6] I. R. Titze and J. Sundberg, "Vocal intensity in speakers and singers," *J. Acoust. Soc. America*, vol. 91, no. 5, pp. 2936–2946, May 1992.

[7] M. Umbert, J. Bonada, M. Goto, T. Nakano, and J. Sundberg, "Expression control in singing voice synthesis: Features, approaches, evaluation, and challenges," *IEEE Signal Process. Mag.*, vol. 32, no. 6, pp. 55–73, Nov. 2015.

[8] L. Cen, M. Dong, and P. Chan, "Segmentation of speech signals in template-based speech to singing conversion," in *Proc. APSIPA Annu. Summit and Conf.*, Xi'an, China, Oct. 2011.

[9] L. Cen, M. Dong, and P. Chan, "Template-based personalized singing voice synthesis," in *Proc. 2012 IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, Kyoto, Japan, Mar. 2012, pp. 4509–4512.

[10] T. Saitou, M. Goto, M. Unoki, and M. Akagi, "Speech-to-singing synthesis: Converting speaking voices to singing voices by controlling acoustic features unique to singing voices," in *Proc. 2007 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, Oct. 2007, pp. 215–218.

[11] T. Saitou, M. Goto, M. Unoki, and M. Akagi, "Vocal conversion from speaking voice to singing voice using STRAIGHT," in *Proc. INTERSPEECH*, Antwerp, Belgium, Aug. 2007, pp. 4005–4006.

[12] T. L. Nwe, M. Dong, P. Chan, X. Wang, B. Ma, and H. Li, "Voice conversion: From spoken vowels to singing vowels," in *Proc. 2010 IEEE Int. Conf. Multimedia and Expo*, Singapore, July 2010, pp. 1421–1426.

[13] K. Vijayan, M. Dong, and H. Li, "A dual alignment scheme for improved speech-to-singing voice conversion," in *Proc. APSIPA Annu. Summit and Conf.*, Kuala Lumpur, Malaysia, Dec. 2017, pp. 1547–1555.

[14] K. Vijayan, X. Gao, and H. Li, "Analysis of speech and singing signals for temporal alignment," in *Proc. APSIPA Annu. Summit and Conf.*, Honolulu, Hawaii, Dec. 2018.

[15] S. H. Mohammadi and A. Kain, "An overview of voice conversion systems," *Speech Commun.*, vol. 88, pp. 65–82, Apr. 2017.

[16] S. W. Lee, Z. Wu, M. Dong, X. Tian, and H. Li, "A comparative study of spectral transformation techniques for singing voice synthesis," in *Proc. INTERSPEECH*, Singapore, Sept. 2014, pp. 2499–2503.

[17] Z. Duan, H. Fang, B. Li, K. C. Sim, and Y. Wang, "The NUS sung and spoken lyrics corpus: A quantitative comparison of singing and speech," in *Proc. APSIPA Annu. Summit and Conf.*, Kaohsiung, Taiwan, Oct. 2013, pp. 1–9.

[18] X. Gao, B. Sisman, R. K. Das, and K. Vijayan, "NUS-HLT spoken lyrics and singing (SLS) corpus," in *Proc. Int. Conf. Orange Technologies (ICOT)*, Bali, Indonesia, Oct. 2018.

[19] S. W. Lee, S. T. Ang, M. Dong, and H. Li, "Generalized F0 modelling with absolute and relative pitch features for singing voice synthesis," in *Proc. 2012 IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, Kyoto, Japan, Mar. 2012, pp. 429–432.

[20] H. Kawahara and M. Morise, "Technical foundations of TANDEM-STRAIGHT, a speech analysis, modification and synthesis framework," *Sadhana*, vol. 36, no. 5, pp. 713–727, Oct. 2011.

[21] M. Morise, F. Yokomori, and K. Ozawa, "WORLD: A vocoder-based high-quality speech synthesis system for real-time applications," *IEICE Trans.*, vol. E99-D, pp. 1877–1884, Jul. 2016.

[22] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 19, no. 4, pp. 788–798, May 2011.

[23] D. Snyder, P. Ghahremani, D. Povey, D. Garcia-Romero, Y. Carmiel, and S. Khudanpur, "Deep neural network-based speaker embeddings for end-to-end speaker verification," in *Proc. 2016 IEEE Spoken Language Technology Workshop (SLT)*, San Diego, California, Dec. 2016, pp. 165–170.

[24] C. Gupta, H. Li, and Y. Wang, "Perceptual evaluation of singing quality," in *Proc. APSIPA Annu. Summit and Conf.*, Kuala Lumpur, Malaysia, Dec. 2017, pp. 577–586.

[25] T. Flynn and A. Marley, *Best-Worst Scaling: Theory and Methods*. Cheltenham, UK: Edward Elgar Publishing, Inc., 2014.

[26] A. Tajadura-Jiménez, P. Larsson, A. Väljamäe, D. Västfjäll, and M. Kleiner, "When room size matters: Acoustic influences on emotional responses to sounds," *Emotion*, vol. 10, no. 3, pp. 416–422, 2010.

[27] T. Yoshimura, K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, "Mel-cepstrum-based quantization noise shaping applied to neural-network-based speech waveform synthesis," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 26, no. 7, pp. 1177–1184, July 2018.

[28] T. Hayashi, A. Tamamori, K. Kobayashi, K. Takeda, and T. Toda, "An investigation of multi-speaker training for wavenet vocoder," in *Proc. 2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Okinawa, Japan, Dec. 2017, pp. 712–718.

[29] B. Sisman, M. Zhang, and H. Li, "A voice conversion framework with tandem feature sparse representation and speaker-adapted wavenet vocoder," in *Proc. INTERSPEECH*, Hyderabad, India, Sept. 2018, pp. 1978–1982.

[30] M. Dong, S. W. Lee, H. Li, P. Chan, X. Peng, J. W. Ehnes, and D. Huang, "I2R speech2singing perfects everyone's singing," in *Proc. INTERSPEECH*, Singapore, Sept. 2014, pp. 2148–2149.

[31] Center for Research in Entertainment and Learning. (2008). Singing voice research database. [Online]. Available: http://crel.calit2.net/projects/databases/svdb

[32] X. Sarasola, E. Navas, D. Tavarez, D. Erro, I. Saratxaga, and I. Hernaez," A singing voice database in Basque for statistical singing synthesis of *bertsolaritza,"* in *Proc. Language Resources and Evaluation Conf. (LREC),* Portoroz, Slovenia, 2016, pp. 756–759.

[33] Isophonics. (2014). Singing voice audio dataset. [Online]. Available: http://isophonics.net/SingingVoiceDataset

[34] Science of Music, Auditory Research and Technology, Ryerson Univ. (2018). RAVDESS. [Online]. Available: https://smartlaboratory.org/ravdess/

SP

Balázs Bank and Juliette Chabassier

Model-Based Digital Pianos

From physics to sound synthesis

©ISTOCKPHOTO.COM/TRAFFIC_ANALYZER

Digital Object Identifier 10.1109/MSP.2018.2872349
Date of publication: 24 December2018

As a result of their complexity and versatility, pianos are arguably one of the most important instruments in Western music. The size, weight, and price of grand pianos as well as their relatively simple control surface (i.e., the keyboard), have led to the development of digital counterparts that mimic the sound of acoustic pianos as closely as possible. While most commercial digital pianos are based on sample playback, it is also possible to reproduce a piano's sound by modeling the physics of the instrument. The process of physical modeling starts with first understanding the physical principles, followed by creating accurate numerical models, and finally, finding numerically optimized signal processing models that allow sound synthesis in real time by excluding inaudible phenomena and adding some perceptually important features by using signal processing tricks. Accurate numerical models can be used by physicists and engineers to understand how the instrument functions or to help piano makers with instrument development. On the other hand, efficient real-time models are geared toward composers and musicians who perform at home or onstage. This article provides an overview of a physics-based piano synthesis beginning with computationally heavy, physically accurate approach followed by a discussion of the approaches that are designed to produce the best possible sound quality for real-time synthesis.

Motivation for piano modeling

In classical music, the piano can stand on its own (solo pieces), work as a lead instrument (piano concertos), or accompany other soloists. It also has a special role in jazz and other popular genres.

However, due to its size, weight, and price, it is not always practical for everyone to own such an instrument. Some of these factors are less critical for upright pianos, but the bulk of the instrument can be a problem when practicing at home as well as for performing musicians who transport them. Digital piano synthesizers have been developed to alleviate this problem. Known as *digital pianos*, they are different from early electromechanical instruments such as Fender Rhodes or Wurlitzers, which are commonly referred to as *electric pianos*.

Most digital pianos on the market use sampling technology, which means that the sounds of an acoustic piano are recorded and then played back when a key is pressed. In the early times of sampling technology, limited memory sizes meant compromised quality, but it is now possible to sample each note individually with various key velocities/loudness levels.

However, some phenomena, like the free vibration of the strings when the sustain pedal is pressed, the coupling between the strings of the sounding notes, or the restrike of an already sounding string cannot be easily and concisely reproduced by recorded samples. Another desired factor that cannot be easily achieved by sampling-based pianos is that of the player continuously altering the properties of the piano sound, e.g., by changing the hardness of the hammer, tuning the string, or positioning the (virtual) recording microphones. In sampling technology, these changes can only be faithfully reproduced by having different sample sets for all of the different scenarios.

With the ever-increasing computational capacity of computers and digital signal processors, a different approach called *physical modeling synthesis* is now possible for synthesizing piano sound. Physical modeling reproduces the functioning of the whole instrument rather than resynthesizing some recorded sound samples (which does not use any a priori knowledge of their physical origin). Therefore, in theory, it should allow a more faithful virtual reproduction of the piano with a better responsiveness to the actions of the player.

In addition to creating a more portable and affordable replacement for the acoustic piano, modeling the piano can be useful for various other reasons. First, modeling appeals to the physicists and engineers who want to capture the physics of the instrument, the phenomena happening at each step of the sound production, and the reasons behind its historical evolutions. It can therefore help the community to rationalize and understand the empirical statements heard for centuries, such as the reason for using multiple strings, increasing the string tension, or the appearance of the steel frame. From an acoustical and structural point of view, modeling is also important to piano manufacturers who would like to see the instrument evolve based on practical (e.g., processes, availability, cost and workability of materials, and technical evolutions) and musical (e.g., esthetic sound quality, playability, and dynamic response) motivations. Finally, modeling is essential for the composers and players who use virtual synthesized pianos that respond in real time to their playing on the keyboard and are attracted to the tunability and playability of the virtual instrument and the realism of its sound. Therefore, the acoustical requirements for digital pianos vary from one musical community to the next, specifically regarding the portion of the physics that "cannot be heard" but comes from the structural requirements of the piano's construction. Conversely, manufacturers sometimes have to choose processes (which have an impact on the sound that could be implemented differently) for practical reasons. Whether to model the physical phenomena faithfully or only reproduce their effect on the sound depends on the objective (i.e., the realism of the produced sound or the physical accuracy at each point of the instrument).

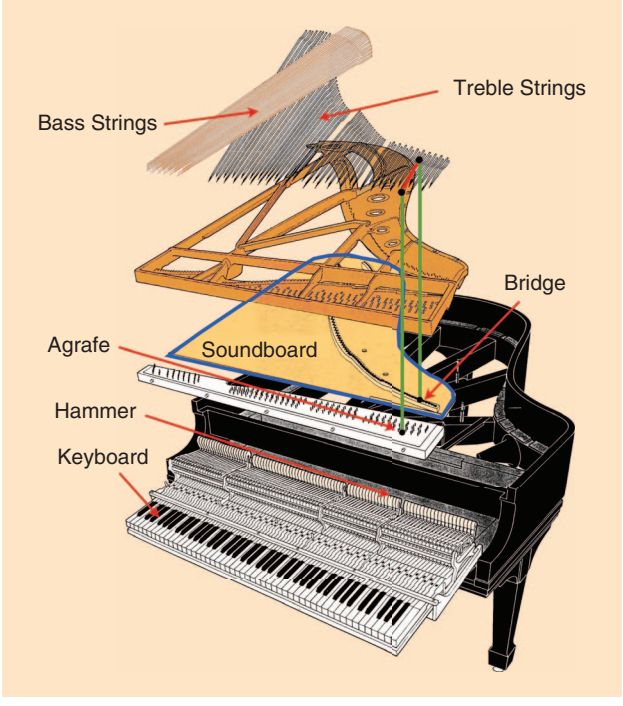


FIGURE 1. An enlarged view of a grand piano.

The goal of this article is to approach physics-based piano synthesis from the perspective of heavy, accurate, physical modeling techniques to real-time sound synthesis techniques aimed at the best possible player involvement and experience. Note that the methodology of piano measurements is outside the scope of this article (see [1] and [2] for more information).

Physics of the piano

When the key of a grand piano is struck (Figure 1), a complex mechanism transmits this motion to the hammer, a small piece of wood covered with felt that is attached to the end of a thin wooden shank. After a controlled phase, during which the hammer motion is directly related to the key motion, the hammer travels freely until the hammer strikes one, two, or three strings, depending on the played note. The steel strings (either solid or copper-wound for the bass range) extend from the tuning pin, are blocked at the agrafe, and pass over the bridge through two pins to join another blocking point linked to the steel frame. The vibrating part of the string, which is struck by the hammer, is between the agrafe and the bridge. However, the other parts of the string can vibrate by sympathy and originate the "duplex-stringing" effect.

Interesting properties of the piano's strings [2]–[6] include:

- inharmonicity
- beating (e.g., the partial f_5 in Figure 2) and two-stage decay for various reasons (double-string polarization, coupling with the soundboard, coupling at the bridge)
- nonlinear behavior that couples transversal waves (orthogonal to the string's elongation at its rest position) to longitudinal waves (parallel to the string's elongation at its rest position) and conducts to precursors in the sound and to

phantom partials in its spectral content (e.g., the nonlinear partials in Figure 2).

All of these effects contribute to the sound and physics of the piano and should be represented in a realistic model. When the string vibrations reach the bridge, which is composed of one or many long pieces of wood, they are transmitted to the soundboard via a complex coupling mechanism.

Experiments [5] demonstrate that both transversal and longitudinal string oscillations are transmitted to the soundboard, which is a prestrained shell of glued, laminated spruce wood carrying long wooden ribs on its underside and one or several bridges on its upper side. The strings pass over the bridge and exert a load that constrains the soundboard and makes it flat. Finally, the soundboard radiates in the surrounding air as a result of acoustomechanical coupling. The soundboard is attached to the wooden case called the rim, where the keyboard rests as well, while the strings are attached to the steel frame. The case can be closed or opened by using a lid. Experiments also show [7] that all of these parts vibrate as well and contribute (more or less) to the piano's sound depending on the range of the notes played. Finally, other features including dampers and una corda can be operated by the keys or pedals, which are important for playability and musical expression.

Comprehensive physical modeling of the piano

Although it would be theoretically possible to write the three-dimensional (3-D) equations that address each and every part of a grand piano to simulate the displacements, stresses and pressures at every point of the piano and the surrounding air, it would be impossible to solve the resulting model on the existing computational facilities in a reasonable amount of time. In addition to this, we also anticipate the need to design reduced models for real-time sound synthesis; therefore, each part of the piano should be modeled using the most adequate and concise description. Several comprehensive models can be found in literature: from the simplest ones having only a hammer and string [8]–[10], [44], to more elaborate models that include a soundboard and sound radiation [11], to the most extensive models, which take the nonlinearity of the string and all of the coupling phenomena into account [12]. These methods solve equations in the space-and-time domain so that the output of the computation represents the displacements and stresses at each point of the parts of the piano, at each time. Special care must be given to the design of stable and accurate numerical methods, which is not a simple task in the presence of nonlinear behavior and coupled systems.

Strings

The ideally flexible, lossless, and unterminated (infinite) string can be described by the d'Alembert equation $(1a-1)$, where $y = y(x,t)$ is the transversal displacement of the string at position x and time t, μ is the linear mass density (mass per unit length), and T_0 is the tension of the stretched string [13]. For treble strings made of steel, the linear mass density is the product of the steel density and the cross section area S. For bass strings, which are wounded with copper, the string mass must be measured from which an effective linear density is computed by dividing with the string length.

The d'Alembert equation $(1a-1)$ has the form of a "one-dimensional (1-D) wave equation" that describes various wave phenomena, including the longitudinal vibration of linear and homogeneous solids, air vibration in tubes, and ideal wave propagation in lossless electrical transmission lines (i.e., the telegrapher's equation).

In the case of the piano, the vibrating part of the string is terminated by the agrafe and the bridge. In this first approximation, we can assume that terminations $x = 0$ and $x = L$ are rigid, which leads to the null displacement boundary conditions in (1b).

Additionally, the string vibration decays due to internal and radiation losses of the string in a frequency-dependent way, which can be accounted for by adding a constant term and a frequency-dependent term $(1a-2)$ [10].

Piano strings are quite thick compared to strings in other instruments (e.g., guitar, violin, among others); therefore, they are not assumed to be perfectly flexible and will borrow some of the vibrational behavior of metal bars. This results in "stiffness," which makes high-frequency waves travel faster than low-frequency waves, a phenomenon referred to as *dispersion*. The subsequent physical behavior of the strings causes the overtones of the piano to deviate from the perfect harmonic series common to most musical instruments (this "inharmonicity" is a very important perceptual characteristic of the piano's tone [14]). Several models (all of which differ in their frequency-range validity) can explain stiffness. The most commonly used is the Euler–Bernoulli model, which adds $(1a-3)$ [9], [13], where E is the Young's modulus of the string. This model assumes that the sections of the string are rigid and remain orthogonal to the string's neutral axis. The equations must be completed by additional boundary conditions, which,

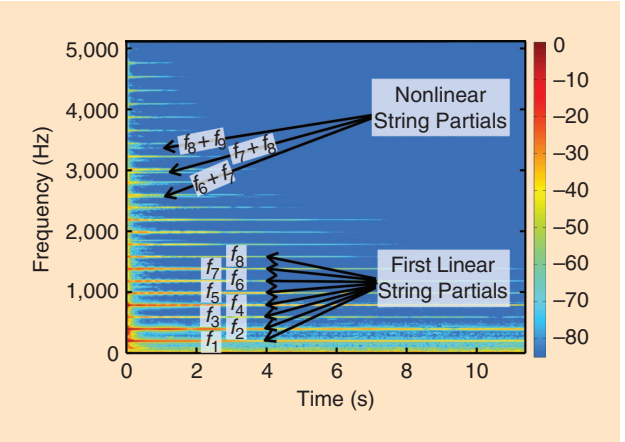


FIGURE 2. The spectrogram of a recorded G_3 *fortissimo* tone of a Steinway D grand piano. Linear partials are not exact harmonics because of dispersion in the string. Beating appears (e.g., the amplitude modulation of f_5) because three strings vibrate together for this note. Nonlinear "phantom" partials are visible between the quasi-harmonic series. Damping is frequency dependent, i.e., upper partials decay faster than lower ones. Finally, the soundboard contributes with a background shock sound.

in a first approximation, correspond to rigid terminations (1c). Timoshenko's model, used in [12], also considers rigid sections but allows them to rotate around their rest position and results in a system of two unknowns, rather than an augmented, scalar wave equation.

The final term, (1*a*–4), describes all of the external forces acting on the string, such as a hammer striking the string.

A possible comprehensive resulting string model accounting for all of the cited phenomena can be written as

$$\underbrace{\mu\frac{\partial^2 y}{\partial t^2}}_{(1a-1)} = \underbrace{T_0\frac{\partial^2 y}{\partial x^2}}_{(1a-1)} \underbrace{-2R\mu\frac{\partial y}{\partial t}}_{(1a-2)} + \underbrace{2\eta\mu\frac{\partial^3 y}{\partial t\partial x^2}}_{(1a-2)} \underbrace{-ES\kappa^2\frac{\partial^4 y}{\partial x^4}}_{(1a-3)} + \underbrace{d_y(x,t)}_{(1a-4)}, \tag{1a}$$

$$y(0,t) = y(L,t) = 0, \tag{1b}$$

and

$$\frac{\partial^2 y}{\partial x^2}(0,t) = \frac{\partial^2 y}{\partial x^2}(L,t) = 0. \tag{1c}$$

This system can then be discretized using finite difference (FD) in space and time, as in [9], or finite elements (FEM)–FD, as in [12]. Let us illustrate the resulting algorithm when we solve equations $[(1a\text{–}1)\text{–}(1b)]$ with a FEM discretization for the space coordinate and an FD discretization for the time coordinate. The basic idea of the FEM–FD method is to find the unknown $y(x,t)$ at regular times $t_n = n\Delta t$, where Δt is called the *time step of the method*, and n is an integer. The space discretization relies on a mesh that can be regular or irregular, depending on the physical problem. The mean space step is called h. On each element of the mesh, the solution is sought as a linear combination of high-order polynomial functions (called *basis functions*), which are triangle-shaped "hat-functions" for the first-order case, and third-order polynomials for the third-order case, the latter of which are shown in Figure 3. Accordingly, the unknown becomes a vector $u_k[n]$ describing the K-nodal amplitudes at each time instant n. This method also relies on the choice of quadrature formulae to compute integrals terms [15]. The adequate choice of the Gauss–Lobatto quadrature along with basis functions, which are chosen as Lagrange interpolation polynomials on these points, leads to an explicitly updated algorithm

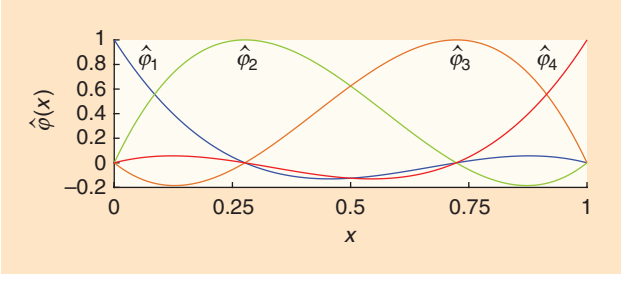


FIGURE 3. The third-order FEM basis functions on the unit interval, i.e., the Lagrange interpolation polynomials based on the Gauss–Lobatto points.

$$u_k[n+1] = 2u_k[n] - u_k[n-1] + \frac{\Delta t^2}{m_k}\sum_{j=1}^{K} A_{k,j}u_j[n], \tag{2}$$

where $A_{k,j}$ is the so-called stiffness matrix, a sparse matrix whose band size is related to the order of the FEM, and m_k is the kth mass coefficient. Their value depends on the physical coefficients T_0 and μ for (1a). Low-order FDs in space can actually be interpreted as first-order FEM. Increasing the order of the FEM decreases exponentially the numerical error induced by the spatial discretization on the solution.

In addition to the transversal wave, the presence of longitudinal waves in the string has an important effect on both the time- and frequency-domain behavior (referred to as the *nonlinear precursor* and *phantom partials*, respectively). A physical model that accounts for a geometrically exact tension is derived in [13] and leads to a nonlinear coupling between the transversal and longitudinal waves involving a square root. Although this model has very attractive mathematical properties, Taylor expansions have been performed [16], [17] to understand the effects at the first and second orders as well as to ease computational difficulties. It turns out that the longitudinal wave $v = v(x,t)$ is the solution of a d'Alembert equation forced by a nonlinear term that depends on y:

$$\mu\frac{\partial^2 v}{\partial t^2} = EA\frac{\partial^2 v}{\partial x^2} + \frac{EA - T_0}{2}\frac{\partial}{\partial x}\left(\frac{\partial y}{\partial x}\right)^2. \tag{3}$$

Reciprocally, the transversal wave equation is forced by a higher-order nonlinear expression of v and y.

Such expansions are not performed in [12], where a FEM space discretization is proposed along with an energy-consistent time discretization for a stiff and geometrically exact nonlinear string. Finally, other models eliminate the longitudinal wave by considering a nonlinear and nonlocal string equation [18].

The main displacement of the string is in the direction of the hammer strike; however, because of slight imperfections of the string and complex boundary conditions, the string also vibrates in the orthogonal polarization [19]. This double polarization is one possible explanation for the observed amplitude modulation, i.e., the two-stage decay and beating (e.g., the partial f_5 in Figure 2) of piano sounds. Another explanation is that when a single key is struck, two or three strings that are slightly detuned are sounded [3] (except for the lowest octaves).

Another component of piano strings is the presence of dampers, i.e., the long, felt strips that always contact the strings except when the sustain pedal is operated or the corresponding key is pressed. The dampers have a dissipative effect; however, a realistic model should account for the dynamic interaction between the dampers and strings and the fact that the dissipation is not perfect. The highest notes of the piano are not equipped with dampers, therefore, the corresponding strings are always vibrating.

Finally, the nonexcited parts of the strings are mainly damped with felt, although some piano makers choose not to damp them so as to create a resonance that contributes to the overall piano sound—this is called *duplex stringing*.

Action and hammer

The action that converts the key motion into the hammer motion is very complex [20], [21] and relies on many lever arms made of wood, joined by rollers covered with felt. At the end, the hammer head is made from a piece of wood covered with felt that is crushed when it interacts with the string.

In a first but efficient approximation, the 3-D deformation of the piano hammer head can be described as a small mass connected to a nonlinear zero-dimensional (0-D) spring that contacts the string around a point x_h. The equations describing the hammer–string interaction are as follows [22], [23]:

$$F_h(t) = F(\Delta y) = \begin{cases} K_h(\Delta y)^{P_h} & \text{if } \Delta y > 0 \\ 0 & \text{if } \Delta y \leq 0 \end{cases}, \tag{4a}$$

$$F_h(t) = -m_h \frac{d^2 y_h(t)}{dt^2}, \tag{4b}$$

where $F_h(t)$ is the interaction force, $\Delta y = y_h(t) - y_s(t)$ is the compression of the hammer felt, $y_h(t)$ is the position of the hammer, and $y_s(t)$ is the position of the string at the excitation point x_h, i.e., $y_s(t) = y(x_h, t)$. The hammer mass is denoted as m_h, K_h is the hammer stiffness coefficient, and P_h is the stiffness exponent. A hysteretic behavior can also be modeled by adjusting the force $F_h(t)$ with a dissipative term, thus accounting for the discrepancy between the hammer compression and relaxation phases.

The bending of the hammer shank, i.e., a small wooden 1-D beam that holds the hammer head, has been investigated to understand one possible mechanism through which the pianist's touch influences the piano sound [24].

Another component of piano strings is the presence of dampers, i.e., the long, felt strips that always contact the strings except when the sustain pedal is operated or the corresponding key is pressed.

Bridge and soundboard

Producing soundboards takes years, during which time the wood is dried and boards are glued together along the wood (spruce) fibers. The resulting plate is given a curvature called the *crown*, which is supposed to compensate for the string loads when they are put in tension. As a result, the soundboard looks flat but is prestrained by the strings. The crown has been studied [25] but is usually neglected in soundboard models, which describe the soundboard by employing typical two-dimensional (2-D) plate equations such as the Kirchhoff–Love or Reissner–Mindlin [26] systems. Both models exhibit good mathematical properties, although the latter is more suited to traditional FEM space discretization. These models can also take into account that the soundboard is thicker at its center than at its edges. Since the wood is orthotropic (i.e., the waves travel at different velocities in orthogonal directions because of the wood fibers), manufacturers arrange ribs under the soundboard to restore, at least at low frequencies, a certain isotropy.

An accurate model of the ribs considers each one as a beam coupled to the plate; however, at low frequencies it is sufficient to model their presence as a local change of thickness of the soundboard plate. The soundboard is attached to the rim in a nontrivial manner, making the boundary conditions difficult to express since it lies somewhere between simply supported and clamped. Finally, waves are damped by various phenomena inside the soundboard. Dissipation phenomena are much more complex than wave propagation; therefore, creating comprehensive dissipation models would first require a tedious parameter-fitting work, and second, a disproportionate computational effort with respect to the rest of the piano. This is why the dissipation is often described (and measured [27]) according to the vibration modes of the soundboard. An efficient computational process proposed in [12] consists of precomputing the modes using a FEM method and using these modes (Figure 4) as a representation basis for the soundboard displacement by adding a modal dissipation gathered from experimentation.

The bridge, which transmits the string vibrations to the soundboard and vice versa, is comprised of a laminated maple or beech beam that can be modeled with 1-D beam equations (e.g., Euler–Bernoulli or Timoshenko). However, a model with accurate string-bridge-soundboard coupling is still lacking, and most existing models consider the bridge to be an ideal coupling feature between the strings and the soundboard. Recent attempts to develop more complex models are described in [28]. It is possible that the vertical vibration of the string is transmitted via solid coupling, while the longitudinal vibration exerts a torque that induces a shear wave in the soundboard. Moreover, the pins through which the strings pass could also be responsible for the transmission of the orthogonal polarization. The bridge vibration will then also couple remote strings when they are sounding together and emphasize sympathetic vibrations.

Sound radiation in the air

Sound radiation in the air can be faithfully modeled by the 3-D linear acoustic wave equation

$$\partial_t^2 p - c^2 \Delta p = 0, \tag{5}$$

where $p = p(\mathbf{x}, t)$ is the sound pressure at a point $\mathbf{x}$ of the open space $\mathbb{R}^3$ and time t, c is the sound celerity, and Δ is the Laplace operator. The presence of the piano rim and lid can, during a first approximation, be considered as obstacles to sound propagation, although a refined model could help describe their respective vibrations and effects. The soundboard constitutes a singular surface in the propagation free space in which the mechanical normal velocity of the soundboard is set equal to the acoustical normal velocity. Reciprocally, the pressure jump between the upper and lower part of the soundboard exerts a load on the soundboard, which is modeled as a force at the right-hand side of the soundboard's equation.

These equations can easily be discretized in space using FDs [11], however, this method does not sufficiently capture the geometrical details of the rim and soundboard and leads to

severely spurious numerical dispersion. A more accurate option is that of a high-order spectral FEM [12]. The room must be artificially truncated to limit the computational domain, which can be done by using absorbing boundary conditions and perfectly matched layers. The acoustic pressure and velocity at a distant point can be recovered by analytical formulae based on closed surfaces (retarded potentials). Another option would be to precompute the impulse responses at several points around the soundboard, however this disregards the reciprocal coupling between the sound propagation and the soundboard.

About time discretization and computational efficiency

The resulting mechanical system of this modeling process is a nonlinear coupled system involving many dimensions [0-D, 1-D, 2-D, and 3-D] with reciprocal interactions. The time discretization must be performed with two main goals in mind: ensuring numerical stability, which is not straightforward in this complex context, and seeking the best possible computational efficiency. One possibility is to rely on energy-based techniques as in [29] for the string or in [11] and [12] for the whole piano. The final algorithm can be run in parallel on computational facilities, and it currently requires 24 h of computation on 300 processors to achieve accurate displacements, strains, and pressures in and around the piano during 1 s of physical time [12].

Overview and drawbacks

These comprehensive physical models provide access to all internal states of the instrument and can therefore be used to better understand the physics of the piano. In addition to estimating the effects of changes in the geometric or material properties of the virtual instrument [30], [31], it is also possible to model a piano that does not exist or is no longer in playing condition. Many features are still missing from existing comprehensive simulation tools, e.g., the key restrike, sympathetic

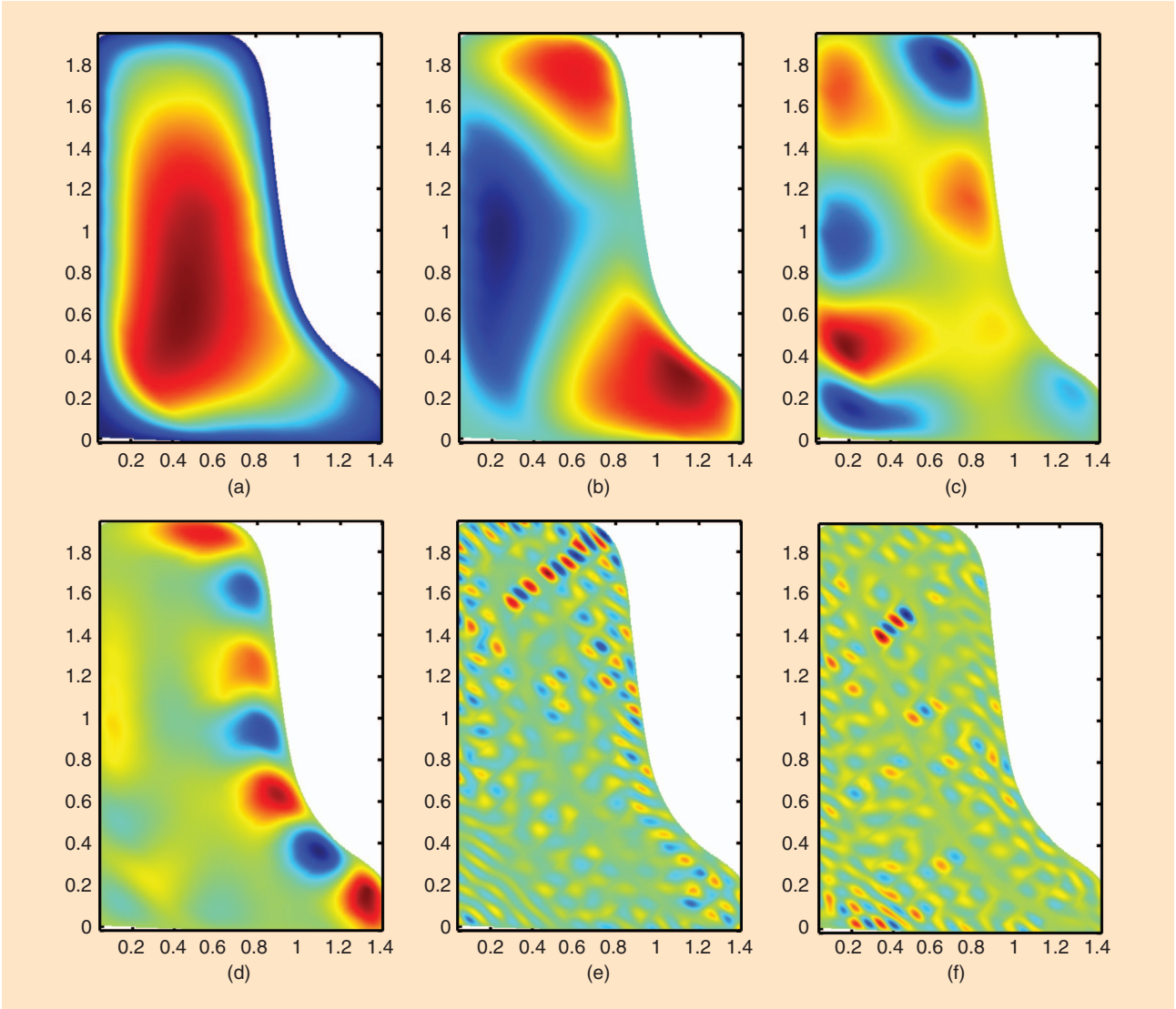


FIGURE 4. Computed soundboard modes from [12]. Low-frequency modes are not sensitive to the fine geometrical features, while high-frequency modes are trapped between the ribs. (a) Mode 1, with the frequency at 23 Hz; (b) mode 4, with the frequency at 67 Hz; (c) mode 9, with the frequency at 139 Hz; (d) mode 18, with the frequency at 252 Hz; (e) mode 392, with the frequency at 2,693 Hz; and (f) mode 436, with the frequency at 2,910 Hz.

strings, duplex stringing, aliquots, dampers, una corda pedaling, lid positioning, and so on. Some of these are relatively simple additions, while others would lead to a significant increase in computation time.

On the other hand, listening to the obtained sounds is disappointing not only because of the aforementioned missing features, but also because the ear is very sensitive to decay rates [32], which are linked to dissipation phenomena that we do not fully understand. In a sense, listening to the comprehensive physical model's sounds gives us an auditory measure of what we currently understand about the physics of the piano.

Modeling the dispersion is somewhat more complicated since it actually requires waves to travel at a frequency-dependent speed.

Reduced models for sound synthesis

Conversely, sound synthesis requires the best possible perceptual quality at a relatively low computational cost. The goal is to slightly depart from "physicality" in favor of sonic realism; this is accomplished by applying models that can be easily fine-tuned based on the analysis of recorded piano tones (as well as manually tuned by experts). While the fine structure of the model parameters are set during model creation, the user retains control over the general properties of the piano sound, such as the overall string decay, inharmonicity and detuning, hammer mass, and hardness in a physically meaningful way. Note that in comprehensive piano models geared toward understanding the physics of the instrument, parameterizing the model based on recorded piano sounds would be unacceptable since it would prevent understanding of how the physical parameters of the instrument (e.g., string mass and stiffness, soundboard geometry and material) influence the piano's behavior.

Efficient string modeling by digital waveguides

Because an acoustic piano has more than 200 strings, it is crucial to model them effectively. One of the most efficient methods of string modeling is digital waveguides [33]. The time-domain solution of the lossless wave equation [(1a–1)–(1b)] in an infinite medium was given by d'Alembert in 1747:

$$y(x,t) = f^{+}(ct - x) + f^{-}(ct + x), \tag{6}$$

which means that the vibrational behavior of the ideal, unterminated string can be described by two independent waveshapes traveling in opposite directions, known as the *traveling-wave solution*. The basic idea of digital waveguides is, rather than numerically solving the wave equation (1a) with the finite element method in (2), to implement its analytical solution (6) directly.

The efficiency of the method comes from the fact that the sampled versions of the two waveshapes can easily be stored in two arrays of computer memory, the content of which is shifted to the right or left at each time sample. In signal processing terms, the two traveling waves are represented by two delay lines. The string displacement $y(x_m, t_n)$ at discrete position x_m and discrete time t_n is the sum of the two delay lines. This is displayed in Figure 5(a).

The structure of Figure 5(a) illustrates the case of the infinite string; however, terminating the string with perfectly rigid boundaries creates wave reflections having opposite signs [13]. This can easily be modeled in digital waveguides by simply

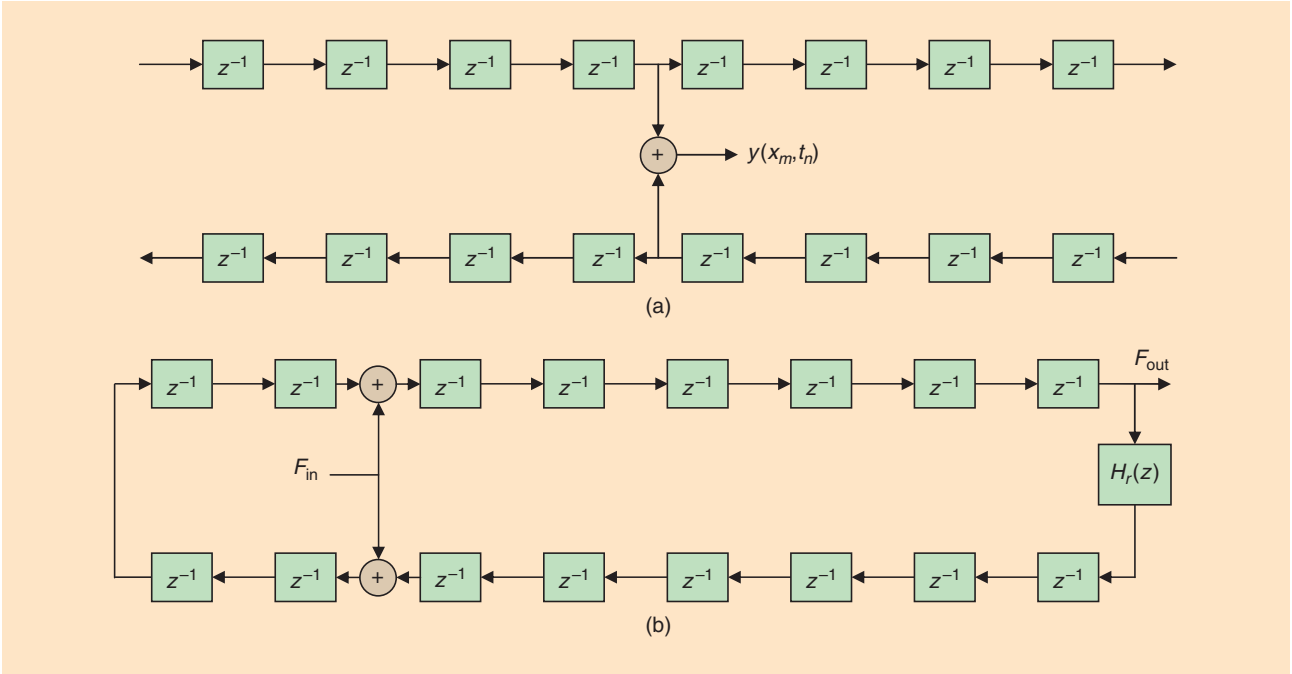


FIGURE 5. A digital waveguide modeling of (a) the model of an infinite ideal string and (b) the model of a terminated string with a reflection filter, force input, and output.

feeding back the output of one delay line to the other with opposite signs.

Real strings exhibit losses and dispersion, as discussed in the "Strings" section. Losses can easily be incorporated in the model by inserting attenuation filters between the delay elements in Figure 5(a).

Modeling the dispersion is somewhat more complicated since it actually requires waves to travel at a frequency-dependent speed. Because the points between the delay elements correspond to the sampled physical positions along the string, this can only be accomplished if the delay elements can shift the signal by a fractional sample, rather than by one, as with unit delays. This can be implemented by replacing the unit delays with all-pass filters, whose delay depends on frequency [34].

However, inserting individual loss and dispersion filters between the unit delays would complicate the structure, and all of the computational benefits achieved from the d'Alembert solution of the wave equation being discretized would be lost.

Before proceeding with this issue, the input and output of the string model should be added. The string is excited by the hammer strike acting at a single position of the string, as displayed by F_{in} in Figure 5(b). The string vibration then is transferred to the soundboard through the bridge, thus, the force must be computed at one of the endpoints of the string. This is displayed by F_{out} in Figure 5(b), with a digital waveguide that now transmits force waves.

In a sense, listening to the comprehensive physical model's sounds gives us an auditory measure of what we currently understand about the physics of the piano.

Accordingly, rather than computing the string shape at all positions, we are only interested in the behavior of the string between its input and output. We can therefore lump the effects of losses and dispersion that occur at one round trip of the waves in the string into a single filter. This is called the *reflection filter* and is displayed as $H_r(z)$ in Figure 5(b). Consolidating all of the losses and dispersion to a single point greatly increases the efficiency since delay lines can be implemented at almost no cost by using circular buffers and a single, relatively low-order filter is also efficiently realized in digital signal processors (DSPs). The transfer function of the digital waveguide is

$$H_{\text{wg}}(z) = \frac{F_{\text{out}}(z)}{F_{\text{in}}(z)} = H_c(z)\frac{1}{1 - z^{-N}H_r(z)}, \tag{7}$$

where $H_c(z)$ is a comb filter coming from the force input that acts at two points on the delay line with opposite signs, while N is the total length of the delay line.

The modal frequencies of the digital waveguide can be estimated by finding the local maxima of the transfer function $H_{\text{wg}}(z)$ where the feedback structure has a very high (nearly infinite) gain. These are the frequencies in which the denominator is approximately zero, i.e., $z^{-N}H_r(z) \approx 1$. The magnitude of the reflection filter $|H_r(z)|$ is close to unity, therefore, this condition is met when the phase of $z^{-N}H_r(z)$ is a multiple of 2π, e.g.,

$$\begin{aligned}\varphi\{z^{-N}H_r(z)\} &= \varphi\{e^{-j\vartheta_k N}H_r(e^{j\vartheta_k})\}\\ &= -N\vartheta_k + \varphi\{H_r(e^{j\vartheta_k})\} = -k2\pi,\end{aligned} \tag{8}$$

which gives a digital angular frequency ϑ_k for each k. The analog partial frequencies then become $f_k = [f_s/(2\pi)]\,\vartheta_k$ [35].

The decay time of mode k having the frequency f_k can be computed simply by knowing that mode k is attenuated by $|H_r(e^{j\vartheta_k})|$ each time it passes the reflection filter. As one period of mode k fits into the digital waveguide loop k times, it is attenuated at a periodicity of k/f_k. This gives the following expression for the decay times:

$$\tau_k = -k\big/(f_k \ln|H_r(e^{j\vartheta_k})|), \tag{9}$$

where $\vartheta_k = (2\pi f_k)/f_s$ [35].

Equations (8) and (9) show that the phase response of $H_r(z)$ determines the frequencies of the string partials, while the magnitude of $H_r(z)$ controls its decay time. This fact can be used to accurately tune the behavior of the partials by carefully designing a digital filter $H_r(z)$ with the magnitude and phase responses obtained from the inverses of (8) and (9). The reflection filter is typically implemented as a low-order loss filter $H_1(z)$ [a first-order infinite impulse response (IIR) low pass is a common approximation] and an all-pass filter $H_{\text{ap}}(z)$ (with filter orders between five and 20) in series.

The first step of this process is the analysis of real piano tones, from which the partial frequencies and decay times are obtained, e.g., by short-time Fourier transform or heterodyne filtering [36]. Then, the partial frequencies are used to determine the number of delay elements N and to design an all-pass filter $H_{\text{ap}}(z)$ whose total delay leads to synthesized partial frequencies [see (8)] close to the original. Then, the measured decay times are used to design a low-pass filter that attenuates the signal in every roundtrip in such a way that the synthesized decay times [given by (9)] are as desired. With this method, it is possible to closely match the sonic properties of real piano tones, while the model continues to preserve the physical behavior of the string.

Modeling the effect of multiple strings belonging to the same key and coupling of two transverse polarizations would require the use of six digital waveguides whose vibration is also coupled at the termination. However, for efficiency, simplified models are used, e.g., running two or three waveguide models in parallel for the same note [37], a few resonators in parallel [36], or using modulated bandpass filters tuned to specific partials [38].

When the sustain pedal is pressed, the sounding notes excite all of the unstruck strings as well; this is accomplished by feeding signals from all of the string models to all the others. Care must be taken to not create a positive feedback by such a connection. A usual approach to guarantee stability is to develop a model that corresponds to connecting all the strings to a common passive termination [39], [40]. When a physically passive system is discretized, the stability of the digital model is assured. Another less physical approach is to route the string models in such a way that there is no feedback path, which can

be done by sending signals from the primary string models to the secondary ones, and not vice versa [38].

Derived from the efficiency of digital waveguides, early on, this was the primary method for piano synthesis. The first waveguide-based piano model was developed in 1987 [39]; other piano models with digital waveguides include those described in [10], [36]–[38], and [40].

Modal synthesis of string vibrations

While digital waveguides are capable of the highly efficient modeling of linear string behavior, they are not very well suited to model nonlinear string vibration. On the other hand, the nonlinear longitudinal vibrations of the low and middle range of piano strings are very important for realistic piano sounds [41]. Because of this need, and with the help of increased computational resources, modal-based academic [17], [35], [42] and commercial [such as Pianoteq software created by Modartt (www.pianoteq.com)] piano models were developed between 2005–2006. Rather than the time-domain traveling-wave solution, modal synthesis is based on the standing-wave solution of the wave equation and describes the motion of the string with a set of vibrational modes. The modal shapes of the ideal string with perfect boundary conditions are sinusoidal functions. The string displacement at any time instant can be expressed as the linear combination of these modal shapes:

$$y(x,t) = \sum_{k=1}^{\infty} y_k(t) \sin\left(\frac{k\pi x}{L}\right) \quad x \in [0, L], \tag{10}$$

where $y_k(t)$ is the instantaneous amplitude of mode k.

If (10) is substituted into the wave equation (1a), then multiplied by the modal shape $\sin(k\pi x/L)$ and integrated over x from 0 to L (similar to calculating the Fourier transform), all of the derivatives with respect to space x vanish and only time derivatives remain. This results in an ordinary second-order differential equation that governs the behavior of mode k

$$\frac{d^2 y_k}{dt^2} + a_{1,k}\frac{dy_k}{dt} + a_{0,k} y_k = b_{0,k} F_{y,k}(t), \tag{11}$$

which is similar to the differential equation that describes the vibration of a mass-spring-damper system or an LRC circuit. The impulse response of such a system can be written analytically, and, when damping is moderate, it is an exponentially decaying sinusoidal function

$$y_{\delta,k}(t) = A_k e^{-\frac{t}{\tau_k}} \sin(2\pi f_k t). \tag{12}$$

The term $F_{y,k}(t)$ in (11) is the excitation force acting on mode k, and it is computed as the scalar product of the excitation force density and the modal shape. The exact values of the initial amplitude A_k, partial frequency f_k, and decay time τ_k can be computed by simple expressions from the physical parameters of the string (e.g., mass, stiffness, and losses) [42].

The importance of splitting the partial differential equation of the string into simple second-order differential equations (11) lies in the fact that now each vibrational mode of the string can be modeled by a second-order resonator, which, in discrete time, becomes a second-order IIR filter that can be implemented very efficiently. Additionally, the quality and computational complexity can be easily scaled by the number of resonators chosen, i.e., roughly 100 for the lowest tones and approximately five resonators for the highest tones.

We note that the modal decomposition can be seen as a special discretization method with sinusoidal basis functions. Compared to the FEM method, an important computational benefit is that the stiffness matrix $A_{k,j}$ in (2) becomes diagonal because of the orthogonality of the basis functions. Thus, the motion of individual modes can be computed independently from the other modes using simple equations such as (11) or (12). On the other hand, since the basis functions are not localized in space, the reconstruction of the motion of the whole string would be computationally very expensive, although this is not needed in the context of sound synthesis where only the force at the termination of the string is required.

By using the impulse-invariant transform, the discrete-time-impulse response of a vibrational mode is obtained by simply sampling the continuous-time-impulse response (12), which yields

$$y_{\delta,k}[n] = y_{\delta,k}(t_n) = A_k e^{-\frac{t_n}{\tau_k}} \sin(2\pi f_k t_n) \Big/ f_s, \tag{13}$$

where $t_n = nT_s, T_s = 1/f_s$ is the sampling interval. Equation (13) differs from (12) by a scaling factor of $1/f_s$. This scaling is required because the discrete-time unit pulse has an area of $1/f_s$, while the continuous-time Dirac impulse has unity area.

Taking the z transform of $y_{\delta,k}[n]$ after some algebraic manipulations gives

$$H_{\text{res},k}(z) = \frac{b_k z^{-1}}{1 + a_{1,k} z^{-1} + a_{2,k} z^{-2}}, \tag{14a}$$

$$p_k = e^{j2\pi\frac{f_k}{f_s}} e^{-\frac{1}{\tau_k f_s}}, \quad b_k = \frac{A_k}{f_s} \text{Im}\{p_k\}, \tag{14b}$$

and

$$a_{1,k} = -2\,\text{Re}\{p_k\}, \quad a_{2,k} = |p_k|^2. \tag{14c}$$

In other words, each mode is implemented by a two-pole filter and a delay in series, all of which are connected in parallel, as shown in Figure 6. The input coefficients $w_{\text{in},k}$ distribute the force input from the hammer F_{in} to the different vibrational modes, while $w_{\text{out},k}$ are the output weights for giving the force at the bridge F_{out}.

While the parameters of the vibrational modes, and thus, the coefficients of the second-order filters can be directly computed from the physical parameters of the string, it is also possible to set them directly, based on the analysis of recorded piano tones. As with digital waveguides, this consists of estimating the frequencies and decay times of the partials and then using these in (14).

Compared to digital waveguides, one of the main benefits of the modal string model is its complete control of the behavior

of partials, which allows for matching the sonic properties of a specific piano very accurately, a feature often desired by piano players. Another advantage is that the nonlinear longitudinal vibration responsible for the characteristic metallic sound of low piano strings can be very efficiently modeled by this technique, as opposed to the digital waveguide method.

Compared to digital waveguides, one of the main benefits of the modal string model is its complete control of the behavior of partials, which allows for matching the sonic properties of a specific piano very accurately, a feature often desired by piano players.

With second-order accurate approximation, the longitudinal vibration of the string can be described by (3) with additional loss terms leading to a similar equation as for the transversal wave (1a), thus, it can also be modeled as a parallel set of second-order resonators (IIR filters). The longitudinal modes gain energy from the transverse motion of the string by a nonlinear coupling, and as a result, the longitudinal mode with mode number k is excited by the product of two transversal modes whose mode numbers m and n satisfy $k = m + n$ or $k = |m - n|$, respectively [35], [42]. From a modeling point of view, this means that the nonlinear longitudinal vibrations can be generated by cross-multiplying the output of the resonators of the primary (transverse) string model and leading this second-order signal to the resonators of the longitudinal string model.

The effect of the coupling of different strings belonging to the same note is again implemented by running more transversal string models in parallel, in similarity with digital waveguides. However, since the computational complexity scales linearly here with the number of modes implemented, this secondary, less important string model may contain fewer resonators as compared to the main one [42].

Modeling the hammer

For reasons of efficiency, the 3-D nature of the hammers is neglected in real-time synthesizers. One of the approaches is to generate a signal that corresponds to the hammer's shape either as a simple function (e.g., the Hann window) or as one that is stored in a wavetable [38], [39]. This allows for the sonic properties of the resulting tone (i.e., the loudness of the partials) to be directly controlled.

Another approach is to run a simplified physical model of the hammer [9], [36], [40], [42], which results in more of a "physical" behavior of the hammer, i.e., for modeling the repeated strike of the same string.

The 0-D hammer equations described in the "Action and hammer" section can be easily discretized with respect to time. Equation (4a) is a static nonlinearity so it is implemented as is. Equation (4b) can be converted to a discrete-time system by first integrating (4b) with respect to time twice and then applying the impulse invariant transform. Thus, the discrete-time version of (4) is expressed as

$$F_h[n] = F(\Delta y) = F(y_h[n] - y_s[n]), \tag{15a}$$

and

$$y_h[n] = 2y_h[n-1] - y_h[n-2] - \frac{1}{m_h f_s^2} F_h[n], \tag{15b}$$

where f_s is the sampling rate.

One interesting feature of (15) is that there is a mutual dependence between $F_h[n]$ and $y_h[n]$. A simple remedy for this "delay-free loop" is inserting a unit delay between the equations, i.e., using the past values of the variables; however, this may lead to numerical instability. Accurate modeling requires a real-time solution of the two equations (15) for each time instant n during the time period when the hammer is in contact with the string [40].

Modeling the soundboard and sound radiation

From a modeling point of view, the most expensive part of the piano is its soundboard because it involves a 2-D vibrating structure and as well as computing the radiation in three dimensions. However, if we accept that we cannot change the physical parameters of the soundboard, a black box model can be used to speed up the computations instead of using a complete model based on the material and geometric properties of the instrument.

The effect of the piano soundboard is twofold: first, it provides a "termination" for the strings together with the bridge, and therefore influences the modal frequencies and decay times of the string partials, which creates a coupling between them. This termination effect is usually included in the string model because there it is easier to account for, e.g., by modifying the mode-decay parameters of the strings. The other effect is that the soundboard radiates the string vibrations, the outcome of which is amplification and frequency-dependent filtering. This latter radiation effect is the one used in physics-based soundboard models.

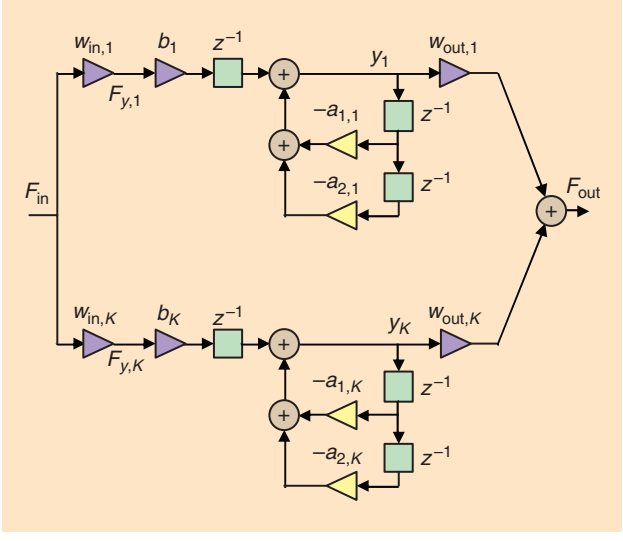


FIGURE 6. A modal-based string model using second-order IIR filters in parallel.

As a result, we uncouple the string-soundboard system and create a feedforward structure that is much more suitable for DSP implementation.

The computationally most efficient way of implementing the effect of the soundboard filtering is through a process known as *commuted synthesis* [37], in which the order of the model blocks (i.e., the hammer-string soundboard) is commuted. By assuming the linearity of the model blocks, their order can be changed; at this point, the impulse response of the soundboard excites the strings and the effect of the hammer is used as a filtering operation. This method assumes linearity and time invariance, therefore, some important effects, such as the restrike of the same string or the nonlinear vibration of strings, cannot be precisely modeled.

The impulse response of a piano soundboard is quite noise like, similar to the impulse response of a room, albeit with much shorter decay. With this similarity in mind, algorithms used to model room reverberation create a very efficient method of modeling the filtering effect of the soundboard (e.g., coupled digital waveguides [39] and feedback delay networks [36]). The advantage of this is very low computational complexity, but the difficulty with this approach is setting the parameters of the reverberation algorithm in such a way that it results in the sound of a specific piano.

A very accurate way of modeling the effect of piano's soundboard is to design a digital filter based on the measured vibration and radiation response of actual piano soundboards. This can be done most simply with a finite impulse response (FIR) filter, although more efficient approaches are available, such as multirate FIR filtering [35], a specialized IIR filter design [42], FFT-based convolution [42], and the combination of the two [43].

Conclusions

This article reviewed the main features of current piano models based on the physical descriptions of the instrument. While these comprehensive models allow for understanding how the instrument functions, the sounds produced are disappointing not only because many features are missing, but also because some phenomena (e.g., dissipation) are not yet accurately modeled.

Physics-based piano synthesis has a three-decades tradition of academic research, starting with a digital-waveguide-based piano model in 1987 [39]. Due to its computational efficiency, digital waveguide has remained the main modeling paradigm for two decades. With the availability of more computational power and the need for modeling nonlinear string vibrations, a modal-based piano model appeared in 2005 [17], [35]. In parallel, a modal-based software piano, Pianoteq, was introduced by Modartt in 2006 (Figure 7), and the first digital piano employing physical modeling, the V-piano, was presented by Roland in 2009 (www.roland.com/products/en/V-Piano). In 2012, the Viscount Corporation introduced the Physis piano (www.physispiano.com), which also uses modal synthesis (Figure 8). With the availability of increased computational power, it is expected that these existing models will continue to improve

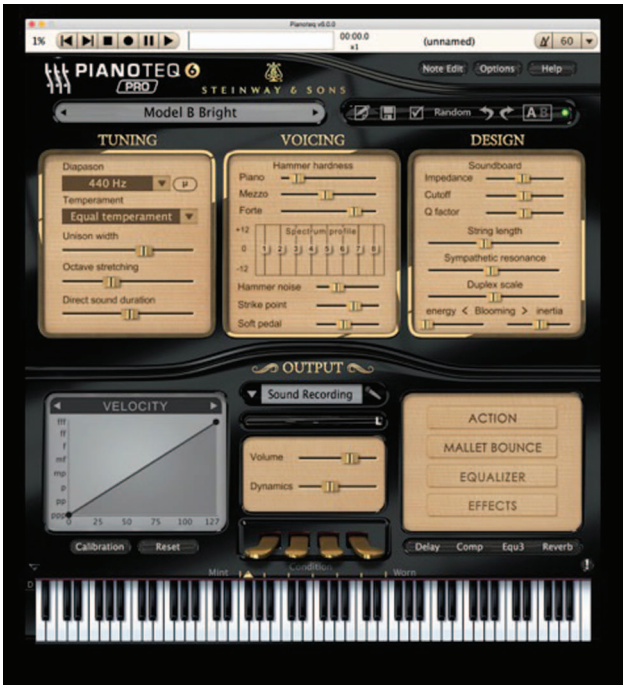


FIGURE 7. The Pianoteq PRO 6 interface. Pianoteq's software computes the piano's sound in real time using physical models. (Image courtesy of the Modartt Corporation.)

FIGURE 8. Viscount Corporation's Physis Piano H1, which applies real-time modal synthesis. (Image courtesy of the Viscount Corporation.)

and that other commercial products that use physical modeling for piano synthesis (as well as for other struck/plucked string instruments that have a similar physical functioning) will become available. Future research in piano modeling includes trying to better understand the string/soundboard coupling mechanism at the bridge, the effect of the crown on the soundboard vibrations, the way in which the pianist can influence the sound, and, finally, finding a way to better model the shock of the key on the structure. Regarding synthesis, future work will attempt to further the link between the model coefficients and the physical reality.

Authors

Balázs Bank (bank@mit.bme.hu) received his M.Sc. and Ph.D. degrees in electrical engineering from the Budapest

University of Technology and Economics, Hungary, in 2000 and 2006, respectively. Both his M.Sc. and Ph.D. theses dealt with the physics-based synthesis of piano sound, and he has also contributed to the development of the Physis Piano of the Viscount Corporation. He is currently an associate professor in the Department of Measurement and Information Systems, Budapest University of Technology and Economics. His research interests include physics-based sound synthesis and filter design for audio applications. He is an associate editor of *IEEE Signal Processing Magazine* and previously served as an associate editor of *IEEE Signal Processing Letters*. He is a Member of the IEEE.

Juliette Chabassier (juliette.chabassier@inria.fr) received her engineering degree from École des Ponts Paristech, Marne-la-Vallée, France, and her M.S. degree in numerical analysis from Paris VI University, France, in 2008. She then completed her Ph.D. thesis on the modeling of the piano as a candidate at École Polytechnique, Palaiseau, France, and was hosted by the Poems Inria team and the Mechanical Engineering Department of ENSTA Paristech. After receiving her Ph.D. degree, she joined the Magique-3D Inria team for the development of mathematical methods for imaging complex media. Since 2013, she has been collaborating with the Modartt team and contributes to the Pianoteq software, and she has developed a research axis aiming at designing optimal musical instruments.

References

[1] A. Askenfelt and E. V. Jansson, "From touch to vibrations. I: Timing in the grand piano action," *J. Acoust. Soc. Amer.*, vol. 88, no. 1, pp. 52–63, 1990.

[2] A. Askenfelt and E. V. Jansson, "From touch to vibrations. III: String motion and spectra," *J. Acoust. Soc. Amer.*, vol. 93, no. 4, pp. 2181–2196, 1993.

[3] G. Weinreich, "Coupled piano strings," *J. Acoust. Soc. Amer.*, vol. 62, no. 6, pp. 1474–1484, 1977.

[4] H. A. Conklin, "Generation of partials due to nonlinear mixing in a stringed instrument," *J. Acoust. Soc. Amer.*, vol. 105, no. 1, pp. 536–545, 1999.

[5] M. Podlesak and A. R. Lee, "Dispersion of waves in piano strings," *J. Acoust. Soc. Amer.*, vol. 83, no. 1, pp. 305–317, 1988.

[6] H. A. Conklin, "Design and tone in the mechanoacoustic piano. Part I. Piano hammers and tonal effects," *J. Acoust. Soc. Amer.*, vol. 99, no. 6, pp. 3286–3296, 1996.

[7] J. J. Tan, A. Chaigne, and A. Acri, "Contribution of the vibration of various piano components in the resulting piano sound," in *Proc. 22nd Int. Congr. Acoustics (ICA)*, 2016, pp. 1–10.

[8] L. Hiller and P. Ruiz, "Synthesizing musical sounds by solving the wave equation for vibrating objects, part 1," *J. Audio Eng. Soc.*, vol. 19, no. 6, pp. 462–472, June 1971.

[9] A. Chaigne and A. Askenfelt, "Numerical simulations of piano strings. I. A physical model for a struck string using finite difference methods," *J. Acoust. Soc. Amer.*, vol. 95, no. 2, pp. 1112–1118, 1994.

[10] J. Bensa, S. Bilbao, R. Kronland-Martinet, and J. O. Smith, "The simulation of piano string vibration: From physical models to finite difference schemes and digital waveguides," *J. Acoust. Soc. Amer.*, vol. 114, no. 2, pp. 1095–1107, 2003.

[11] N. Giordano and M. Jiang, "Physical modeling of the piano," *EURASIP J. Appl. Signal Process.*, vol. 2004, no. 7, pp. 926–933, 2004.

[12] J. Chabassier, A. Chaigne, and P. Joly, "Modeling and simulation of a grand piano," *J. Acoust. Soc. Amer.*, vol. 134, no. 1, pp. 648–665, 2013.

[13] P. M. Morse and K. U. Ingard, *Theoretical Acoustics*. Princeton, NJ: Princeton Univ. Press, 1968.

[14] D. Rocchesso and F. Scalcon, "Bandwidth of perceived inharmonicity for physical modeling of dispersive strings," *IEEE Speech Audio Process.*, vol. 7, no. 5, pp. 597–601, 1999.

[15] A. Quarteroni, R. Sacco, and F. Saleri, *Méthodes Numériques.* Milan, Italy: Springer Verlag, 2007.

[16] S. Bilbao, "Conservative numerical methods for nonlinear strings," *J. Acoust. Soc. Amer.*, vol. 118, no. 5, pp. 3316–3327, 2005.

[17] B. Bank and L. Sujbert, "Generation of longitudinal vibrations in piano strings: From physics to sound synthesis," *J. Acoust. Soc. Amer.*, vol. 117, no. 4, pp. 2268–2278, 2005.

[18] S. Bilbao, "Energy-conserving finite difference schemes for tension-modulated strings," in *Proc. IEEE Int. Conf. Acoustics Speech and Signal Processing (ICASSP)*, 2004, pp. 285–288.

[19] J.-J. Tan, C. Touzé, and B. Cotté, "Double polarisation in nonlinear vibrating piano strings," in *Proc. 3rd Vienna Talk Music Acoustics*, 2015, pp. 182–187.

[20] A. Izadbakhsh, J. McPhee, and S. Birkett, "Dynamic modeling and experimental testing of a piano action mechanism with a flexible hammer shank," *J. Comput. Nonlinear Dynamics*, vol. 3, no. 3, 2008. doi: 10.1115/1.2908180.

[21] A. Thorin, X. Boutillon, J. Lozada, and X. Merlhiot, "Non-smooth dynamics for an efficient simulation of the grand piano action," *Meccanica*, vol. 52, no. 11–12, pp. 2837–2854, 2017.

[22] X. Boutillon, "Model for piano hammers: Experimental determination and digital simulation," *J. Acoust. Soc. Amer.*, vol. 83, no. 2, pp. 746–754, 1988.

[23] A. Stulov, "Experimental and theoretical studies of piano hammer," in *Proc. Stockholm Music Acoustics Conf.*, 2003, pp. 1–4.

[24] J. Chabassier and M. Duruflé, "Energy based simulation of a Timoshenko beam in non-forced rotation. Influence of the piano hammer shank flexibility on the sound," *J. Sound Vib.*, vol. 333, no. 26, pp. 7198–7215, 2014.

[25] A. Mamou-Mani, J. Frelat, and C. Besnainou, "Numerical simulation of a piano soundboard under downbearing," *J Acoust. Soc. Amer.*, vol. 123, no. 4, p. 2401, 2008. doi: 10.1121/1.2836787.

[26] E. Reissner, "The effect of transverse shear deformation on the bending of elastic plates," *J. Appl. Mech.*, vol. 12, pp. 69–77, 1945.

[27] K. Ege, X. Boutillon, and M. Rébillat, "Vibroacoustics of the piano soundboard: (Non) linearity and modal properties in the low- and mid-frequency ranges," *J. Sound Vib.*, vol. 332, no. 5, pp. 1288–1305, 2013.

[28] J. J. Tan, "Piano acoustics: String's double polarisation and piano source identification," M.S. thesis, ENSTA, Univ. Paris-Saclay, France, 2017.

[29] S. Bilbao, "Sound synthesis for nonlinear plates," in *Proc. Conf. Digital Audio Effects*, 2005, pp. 243–248.

[30] J. Chabassier, M. Duruflé, and P. Joly, "Time domain simulation of a piano. Part 2: Numerical aspects," *ESAIM: M2AN*, vol. 50, no. 1, pp. 93–133, 2016.

[31] A. Chaigne, J. Chabassier, and M. Duruflé, "Energy analysis of structural changes in pianos," in *Proc. 3rd Vienna Talk Music Acoustics*, 2015, pp. 189–196.

[32] H. Järveläinen and T. Tolonen, "Perceptual tolerances for decay parameters in plucked string synthesis," *J. Audio Eng. Soc.*, vol. 49, no. 11, pp. 1049–1059, 2001.

[33] J. O. Smith, "Techniques for digital filter design and system identification with application to the violin," Ph.D. dissertation, Center for Computer Research in Music and Acoustics, Stanford Univ., CA, 1983.

[34] T. I. Laakso, V. Välimäki, M. Karjalainen, and U. K. Laine, "Splitting the unit delay—tools for fractional delay filter design," *IEEE Signal Process. Mag.*, vol. 13, no. 1, pp. 30–60, 1996.

[35] B. Bank, "Physics-based sound synthesis of string instruments including geometric nonlinearities," Ph.D. dissertation, Dept. Measurement and Inform. Syst., Budapest Univ. Technol. and Econ., Hungary, 2006.

[36] B. Bank, "Physics-based sound synthesis of the piano," M.S. thesis, Dept. Measurement and Inform. Syst., Budapest Univ. Technol. and Econ., Hungary, 2000.

[37] S. A. Van Duyne and J. O. Smith, "Developments for the commuted piano," in *Proc. Int. Computer Music Conf.*, 1995, pp. 319–326.

[38] J. Rauhala, H. M. Lehtonen, and V. Välimäki, "Toward next-generation digital keyboard instruments," *IEEE Signal Process. Mag.*, vol. 24, no. 2, pp. 12–20, 2007.

[39] G. E. Garnett, "Modeling piano sound using digital waveguide filtering techniques," in *Proc. Int. Computer Music Conf.*, 1987, pp. 89–95.

[40] G. Borin, D. Rocchesso, and F. Scalcon, "A physical piano model for music performance," in *Proc. Int. Computer Music Conf.*, 1997, pp. 350–353.

[41] B. Bank and H.-M. Lehtonen, "Perception of longitudinal components in piano string vibrations," *J. Acoust. Soc. Amer.*, vol. 128, no. 3, 2010. doi: 10.1121/1.3453420.

[42] B. Bank, S. Zambon, and F. Fontana, "A modal-based real-time piano synthesizer," *IEEE Trans. Audio, Speech, Language Process.*, vol. 18, no. 4, pp. 809–821, 2010.

[43] S. Zambon, "Distributed piano soundboard modeling with common-pole parallel filters," in *Proc. Stockholm Music Acoustics Conf.*, 2013, pp. 641–647.

[44] L. Hiller and P. Ruiz, "Synthesizing musical sounds by solving the wave equation for vibrating objects, part 2," *J. Audio Eng. Soc.*, vol. 19, no. 7, pp. 542–551, July 1971.

SP

Waldo Nogueira, Anil Nagathil, and Rainer Martin

Making Music More Accessible for Cochlear Implant Listeners

Recent developments

©ISTOCKPHOTO.COM/TRAFFIC_ANALYZER

Cochlear implants (CIs) have become remarkably successful in restoring the hearing abilities of profoundly hearing-impaired or deaf people. Although in most cases the understanding of continuously spoken speech reaches around 90% after a training and adaptation time of two years, key musical features like pitch and timbre are poorly transmitted by CIs, leading to a severely distorted perception of music. Because music is a ubiquitous means of sociocultural interaction, this handicap significantly degrades the quality of life of CI users. Therefore, in this article, we present recent developments that enable CI users to access music. After a brief review of the state of the art of CIs, we point out the problems of inaccurate pitch and timbre transmission as well as its implications for music perception with CIs. The main part of this article encompasses different emerging strategies for improving CI users' music enjoyment, such as customized music compositions, music preprocessing methods for the reduction of signal complexity, and improved sound coding strategies, and we describe subjective and objective instrumental evaluation procedures.

Introduction

Music plays an important role in people's lives and is part of many sociocultural and educational events. It is deeply rooted in our cultural heritage but is also considered to be a universal language, linking people across cultures. On a more analytical level, the most salient elements of music are dynamics, pitch, rhythm, and timbre [1]. Different music types or styles accentuate or discard some of these elements. People with severe hearing impairment face limitations when listening to the basic elements of sound and, therefore, also have difficulties in perceiving and appreciating music.

Currently, approximately 360 million people worldwide suffer from hearing loss [40]. Hearing loss significantly restricts the extent of interpersonal communication, leads to social isolation, and has developed into a significant socioeconomic factor. In mild-to-severe hearing loss cases, hearing aids (HAs) are able to restore communication. However, they become ineffective when the average bilateral hearing loss

Digital Object Identifier 10.1109/MSP.2018.2874059
Date of publication: 24 December 2018

goes beyond 90-dB hearing level at 2 and 4 kHz [41]. In such cases, the transmission of information between the auditory periphery and the auditory nerve is substantially interrupted. If the auditory nerve is still intact, a CI can partially restore the hearing ability. As illustrated in Figure 1, a CI system consists of an external sound processor and an implant bypassing the auditory periphery. The CI stimulates the spiral ganglion neurons of the auditory nerve via an array of 12–22 electrodes embedded in the conductive perilymph of the cochlea in the inner ear. The cochlea is a spiral-shaped bone cavity making 2.5 turns around its axis with a length that ranges, on average, between 32 and 43.5 mm [2], although shorter or longer cochleas exist. In a normal-hearing listener, high-frequency sounds are transmitted through displacements of the membranes and fluids near the base of the cochlea, while low-frequency sounds produce stronger displacements toward the apex. This so-called frequency-place transformation is mimicked by a CI such that each electrode stimulates a frequency-dependent place on the auditory nerve. The electrode arrays are manufactured with lengths ranging from 10 to 31 mm. Long electrode arrays can be more deeply inserted, achieving two full turns of insertion to reach low-frequency stimulation [3]. Short arrays are typically inserted in the basal turn of the cochlea to protect residual hearing at low frequencies in the most apical part of the cochlea, thus enabling combined electric-acoustic stimulation (EAS) in the implanted ear. The stimulation strategy (also known as *sound coding*) implemented in the sound processor uses a filterbank that decomposes the incoming acoustic signal into different analysis subband signals, which are encoded by a series of electric pulses to stimulate the auditory nerve via dedicated electrodes.

As of 2016, more than 600,000 registered devices had been implanted worldwide [42]. Studies have shown that most users of state-of-the-art CIs are able to understand around 90% of continuously spoken sentences within a training and adaptation phase of two years [4], which enhances their ability to engage in conversations and to participate in social activities. Nevertheless, the bottleneck created between the electrodes and the auditory nerve, different etiological characteristics of the users combined with a nonoptimal transmission of music information in electrical stimulation, results in many CI users who are not yet able to obtain satisfactory music perception. In particular, CI users experience greater difficulties in recognizing melodies or discriminating between different instruments in comparison to normal-hearing listeners [5], [6]. This can be attributed to the low number of electrodes, the spread of electrical fields in the cochlea causing broad excitation patterns and undesired channel interactions, and restrictions of CIs in transmitting the temporal fine structure (TFS) of acoustic signals. In this context, TFS refers to the fast-varying signal components in the subband signals, which carry substantial information about pitch and timbre and are modulated by the subband envelopes. Since commercially used CIs mainly encode the envelope, TFS information gets lost to a large extent. These deficiencies of CIs constitute a severe bottleneck for music transmission between the external acoustic world and the central auditory system and lead to a severely degraded perception of pitch and timbre. As a result, CI users often perceive music as too complex and are not able to derive meaning or enjoyment from it. This is especially true for complex polyphonic music pieces with multiple leading and accompanying voices. In fact, it has been shown that solo instrumental music is preferred by CI users over ensemble or orchestra music and that regularly structured pop and country music is favored over classical music [5], [7]. Given that music is a universal means of sociocultural interaction and triggers positive emotions, CI users with limited access to music face a considerable degradation in their quality of life.

Given that music is a universal means of sociocultural interaction and triggers positive emotions, CI users with limited access to music face a considerable degradation in their quality of life.

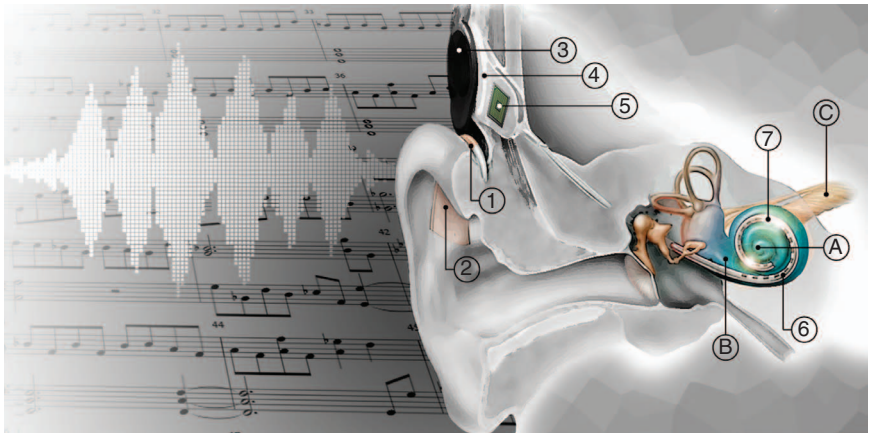


FIGURE 1. The CI with its external sound processor placed on the ear and the implanted electrode array in the inner ear. ① The microphone, ② the sound processor, ③ and ④ the external and internal coils, ⑤ the implanted electronics, ⑥ the electrode array, and ⑦ the electrode contact. Ⓐ The apex of the cochlea or helicotrema and Ⓑ the base of the cochlea, and Ⓒ the auditory nerve. This figure has been adapted with permission from a picture provided by Cochlear Limited.

Music perception with CIs has been described before in several articles, e.g., [5], [7], [8]. The purpose of this article is to briefly discuss the fundamental limitations of music perception and appreciation in CI users and focus on present strategies for their improvement. In recent years, several technological approaches have been proposed to improve music perception with CIs. Figure 2 summarizes some of these approaches including the creation of musical content specifically for CI users, music signal preprocessing algorithms, sound coding strategies, and stimulation modes for electrical stimulation. Moreover, current technology enables HAs

and CIs to stream music from a smartphone or any other device that has wireless capability.

Related earlier work on music perception in CI users

Many aspects of music perception have been studied up to now. In terms of loudness, psychophysical studies in CIs have shown an exponential loudness growth function with increasing electric current. However, CI users receive a much narrower dynamic range in comparison to the 100–120 dB of normal-hearing listeners. In terms of resolution, normal-hearing listeners are able to discern around 100 loudness steps, whereas CI users can only distinguish about 20 loudness steps [7]. Moreover, the high degree of neural synchrony and the steep rate-intensity functions experienced in electrical hearing lead to a compression of dynamic range in CI users [7].

CI users perceive pitch through the place and the temporal pitch mechanisms, e.g., [9]. The first mechanism, the place pitch, is related to the location of stimulation in the cochlea with electrodes located toward the apex producing the lower-pitch sensations than electrodes located toward the base of the cochlea. It has been argued that changes in place pitch caused by activation of different electrodes may be related to perceptual changes in timbre rather than changes in harmonic pitch, e.g., [10]. Place pitch might be limited by the low number of electrode contacts, the current spread produced in the cochlea by individual electrodes, and the mismatch between the acoustic frequency assigned to an electrode and the natural tonotopic frequency corresponding to a particular electrode location in the cochlea. The second mechanism is related to the temporal pattern of stimulation. A higher perceived pitch is obtained with an increased stimulation rate. Many CI users do not perceive differences in the rate of stimulation at individual electrodes if the rate exceeds 300 Hz [11]. Temporal pitch cues in CIs are important to discriminate the sound's fundamental frequency (F0) and to recognize melodies [5]. The low limit of temporal pitch perception also compromises the perception of the sounds' TFS.

Melody perception is very limited in CIs because of the poor pitch perception elicited by these devices, especially if rhythm and verbal cues are removed, e.g., [5], [7], and [8]. Some listeners can discriminate one semitone while others require frequency differences of as much as two octaves [12]. Melody perception is impaired by the limitations in transmitting the F0 of sound and the distortion in transmitting harmony by CIs. The potential mismatch between the frequencies associated with the electrodes and the natural tonotopic frequencies may cause an inconsistent transmission between place and temporal pitch, especially at low frequencies. Moreover, this mismatch depends on the individual size and shape of the cochlea, the electrode type and length, as well as its insertion depth [3]. From a research perspective, different approaches have been proposed to reduce the mismatch by changing the frequency ranges allocated to the electrodes based on postoperative computer tomography scans or by disabling electrodes to improve spectral discrimination. In contrast, in a clinical setting, this mismatch is typically not considered when fitting the CI, although it is possible that the mismatch is compensated to some extent by the adaptation of the human auditory system to the sound processor [13].

Timbre perception is based on the broad-band characteristics of sound, including the spectral energy distribution, the temporal envelope, and especially its attack and decay characteristics. CI users often have difficulties in identifying instruments because of the poor timbre perception they obtain, e.g., [5], [8]. Instrument identification tasks show that errors in normal-hearing subjects tend to be within the same instrument family [5], [7], whereas errors in CI users are spread across these families. In the context of instrument identification, it has been shown that CI users struggle even to detect the presence of polyphony [7].

Rhythm can be defined as regular temporal patterns of musical sounds that range from 50 ms to 5 s corresponding to frequencies between 20 and 0.2 Hz. Longer periodicities beyond 5 s relate to the overall dynamics of music, and faster periodicities below 50 ms carry pitch information, both of which have been already discussed here. On average, CI users perceive rhythm about as well as normal-hearing listeners, e.g., [5]–[7]. This can be explained by the fact that rhythmic information

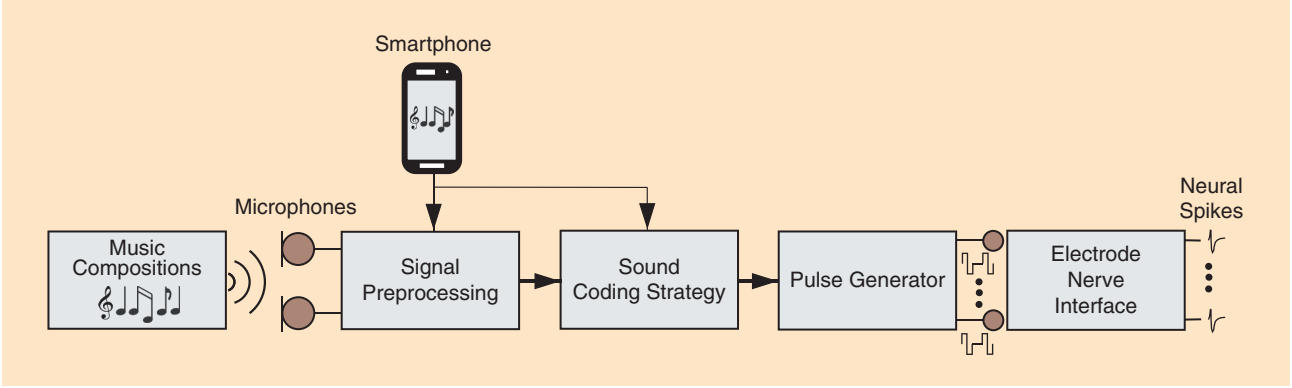


FIGURE 2. The strategies for improving music perception with CIs. The music signal is picked up by either the CI's microphones or is played back from a music source such as a smartphone. The smartphone can also implement preprocessing algorithms and substitute some parts of the sound processor. The electric currents computed by the sound coding strategy are then delivered through pulses to the auditory nerve where neural spikes are generated. These are further transmitted to the central auditory system where they are interpreted as sound.

is encoded in the slowly varying temporal envelope, which is transmitted accurately through CIs.

Music compositions for CI users

Researchers and musicians have developed innovative methods to make music more accessible to CI users beyond signal processing and sound coding developments. In this context, several projects have composed music while taking into account the limitations of electric hearing and have organized concerts such as "Noise Carriers" (Glasgow, United Kingdom, in 2007), "C4CI Grand Finale" (Southampton, United Kingdom, in 2011–2012), "Interior Design" (Melbourne, Australia, in 2010 [14]), and "musIC 1.0, 2.0, and 3.0" [43] (Barcelona, Spain, in 2011; Hannover, Germany, in 2013 and 2017). The goal of these concerts was to understand the differences in both appreciation and perception between normal-hearing and hearing-impaired listeners in live music concerts. Moreover, these concerts serve to increase awareness about hearing loss and the technologies available to rehabilitate hearing loss by means of HAs and CIs. Another goal was to motivate CI users to participate in music-related social activities together with friends and relatives and to have the opportunity to share a musical experience with other CI users and normal-hearing listeners.

The "Interior Design" and the "musIC" projects consisted of several seminars where musicians, CI users, and engineers met to compose, in its majority, electroacoustic music. Electroacoustic music is a genre developed around the middle of the 20th century that incorporates electric sound production techniques into compositional practice. It moves on the limits of what is considered music because it confronts common sense approaches to music based on melody and harmony, which are actually the dimensions most severely impaired when listening through a CI. The musIC concerts featured a combination of electroacoustic and more traditional pop and acoustic compositions.

The perception of the compositions was evaluated through postperformance questionnaires by normal-hearing listeners and CI users in live concerts. The questionnaires had sections to collect qualitative and quantitative data about technical, affective, and cognitive reactions, as well as demographic data from audience members. The use of both pop and experimental music allows the perceptual comparison for exploration of a wide variety of acoustic input. For example, some experimental music may be generated with noise as sounds, whereas pop music may be more melodic. If a CI user was not able to perceive and enjoy music at all, probably this user would not show any difference in the assessment for these two extreme pieces, whereas a normal-hearing subject would rate both pieces differently. The choice of different music styles during the concert was important to evaluate the responses of the different audience groups without being influenced by music genre preferences and to track the responses under very different acoustic scenarios. Some differences between the new compositions created for these concerts and existing music include the use of enhanced vocals with respect to the accompaniment because it has been shown that CI users prefer it [15]; the use of synthetic sounds with a clear fundamental frequency to facilitate the perception of melodies; or the use of simple passages at the beginning of the piece that are repeated with increased complexity trying to guide the listeners through the composition.

The results from the questionnaire of these concerts indicated that both normal-hearing listeners and CI users, in general, considered the events a success. The results revealed similar responses from both groups in terms of interest, enjoyment, and musicality, although melody and timbre perception were rated lower by CI users while their ratings of percussion pieces were typically higher [14]. This suggests that CI technology is still unable to deliver a complete musical experience to CI users.

Music preprocessing

Preprocessing techniques have been proposed to simplify music in order to make music more accessible for CIs. Research has indicated that CI users prefer solo instruments over ensemble or orchestra music and prefer simple and regularly structured music (e.g., pop or country) over complex musical arrangements (e.g., classical music) [7]. Hence, Buyens et al. [15] performed a listening experiment with CI users to whom remixed versions of multitrack pop music recordings were presented. This study revealed that CI users prefer an accentuation of vocals, bass lines, and drums in polyphonic music. However, a moderate vocal amplification of 6 dB was more appreciated than a strong amplification of 12 dB. A similar study was conducted by Kohlberg et al. [16]. Here, CI users listened to 20 modified versions of a country music piece with ten instruments, which was available as a multitrack recording. These modified versions contained either single instruments or different combinations of two to five different instruments. Results indicated a significant preference for versions with one to three instruments over the original piece.

Motivated by the outcomes of these pilot studies, researchers have recently proposed several algorithms for reducing the complexity of music signals. A general block diagram summarizing the different source separation (SS) and remixing approaches and dimensionality reduction techniques is illustrated in Figure 3. In general, an input signal, $x(n)$, is considered, which is transformed to a time–frequency representation, $X(\mu, \lambda)$, where μ and λ denote the discrete frequency and frame index, respectively. From the processed time–frequency representation, $\hat{X}(\mu, \lambda)$, an enhanced music signal, $\hat{x}(n)$, is reconstructed. A harmonic/percussive sound separation (HPSS) method was applied to monophonic pop music in [17]. The estimated harmonic and percussive contributions, $\hat{X}_{\mathrm{h}}(\mu, \lambda)$ and $\hat{X}_{\mathrm{p}}(\mu, \lambda)$, respectively, were remixed with a tunable attenuation G for the harmonic part, such that

$$\hat{X}(\mu, \lambda) = \hat{X}_{\mathrm{p}}(\mu, \lambda) + G\hat{X}_{\mathrm{h}}(\mu, \lambda) \tag{1}$$

denotes the remixed signal in the time–frequency domain. The results indicate that drums and even vocals can be well-preserved by this method for harmonic attenuations of up to 18 dB. This approach was tested with five normal-hearing listeners in

combination with a vocoder simulation, yielding an averaged preference score of 62%. In [18], the HPSS approach was extended to the case of stereophonic pop/rock recordings. The method exploits the fact that vocals, drums, and bass lines are often mixed to be in the center of the stereo image and, thus, are available with equal contributions in both stereo channels, whereas contributions of other instruments may differ. The results show that CI users preferred attenuations of the harmonic part by up to 24 dB. Moreover, the desired attenuation was found to be correlated with the complexity of the music pieces.

Gajecki and Nogueira [19] investigated a music remixing approach based on SS techniques. To this end, nonnegative matrix factorization (NMF), deep recurrent neural networks (DRNN), and deep convolutional autoencoders were used to decompose the pop music pieces used by [15] into estimated instrument sources $\hat{X}_{\mathrm{v}}(\mu, \lambda)$, $\hat{X}_{\mathrm{d}}(\mu, \lambda)$, $\hat{X}_{\mathrm{b}}(\mu, \lambda)$, and $\hat{X}_{\mathrm{a}}(\mu, \lambda)$, which denote the contributions of vocals, drums, bass, and the remaining accompaniment, respectively. The estimated sources were then remixed for each contribution yielding

$$\hat{X}(\mu, \lambda) = G_{\mathrm{v}}\hat{X}_{\mathrm{v}}(\mu, \lambda) + G_{\mathrm{d}}\hat{X}_{\mathrm{d}}(\mu, \lambda) + G_{\mathrm{b}}\hat{X}_{\mathrm{b}}(\mu, \lambda) + G_{\mathrm{a}}\hat{X}_{\mathrm{a}}(\mu, \lambda), \tag{2}$$

where the lowest attenuations were assigned to vocals, drums, and bass lines. Figure 4 presents the spectrograms generated

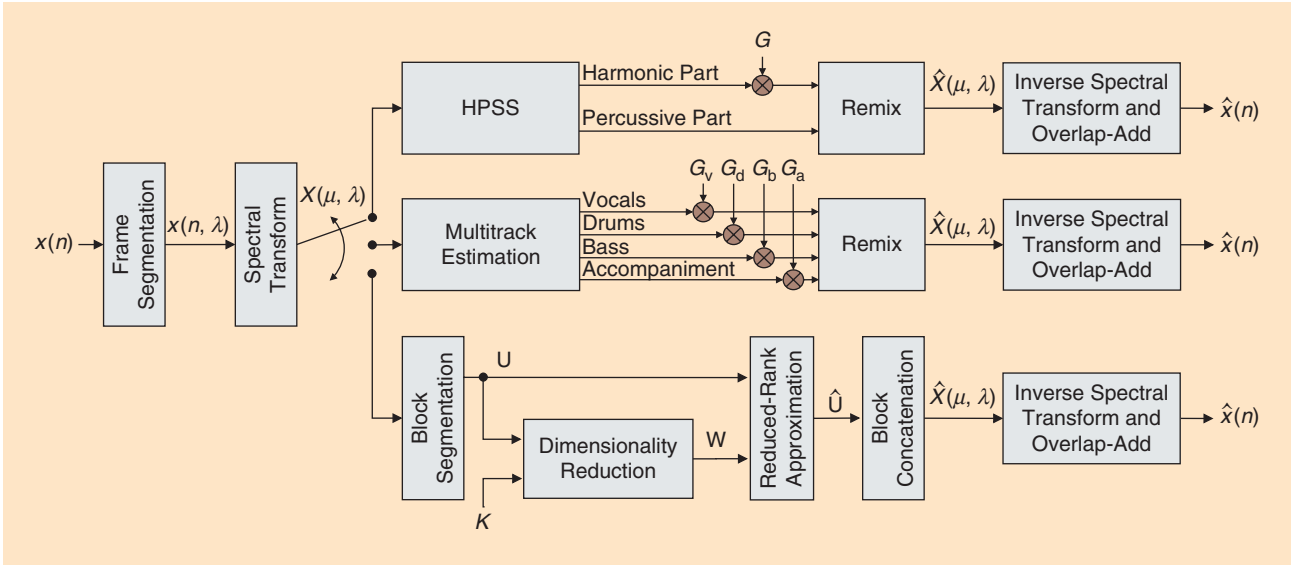


FIGURE 3. A block diagram illustrating the signal processing steps for music complexity reduction based on SS and remixing or dimensionality reduction techniques. First, a music signal, $x(n)$, is segmented into signal frames and transformed into a time–frequency representation, $X(\mu, \lambda)$. For the remixing methods, a sound separation into harmonic and percussive elements or into multitrack signals is performed, which are then remixed using desired attenuation factors. Alternatively, time–frequency blocks containing several frames are collected in the matrix, **U**, for which a dimensionality reduction is performed using a controllable number of components, K. On the basis of the retained components, **W**, a reduced-rank approximation, $\hat{\mathbf{U}}$, is computed. In all cases, the modified segments are concatenated and transformed back to the time domain.

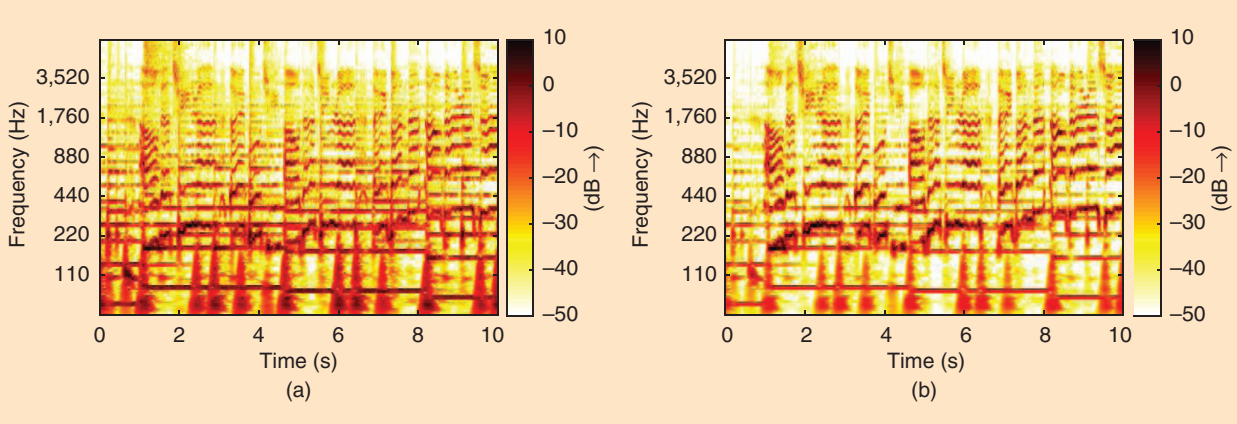


FIGURE 4. Spectrograms for a (a) sound and (b) remixed version for emphasizing the vocals. The sound corresponds to the excerpt 45378_chorus from the iKala data set (http://mac.citi.sinica.edu.tw/ikala/). The remixed version is created based on estimated multitracks using a DRNN to make the vocals 6 dB louder than the other music components.

for an exemplary pop song and a remixed version where the vocals are amplified by 6 dB over the accompaniment. The spectrograms were computed based on the constant-Q transform at a 16-kHz sampling frequency in a frequency range between 55 and 7,040 Hz (seven octaves) using 24 frequency bins per octave, a Hann analysis windows with a frequency-dependent length, and a frame shift of 2 ms. Despite artifacts that are generated during the nonideal SS, the results indicate a clear preference for the remixed music pieces and confirm the findings in [15]. Nagathil et al. [20] investigated spectral complexity reduction of music signals based on dimensionality reduction techniques. Specifically, a principal component analysis (PCA) and a partial-least squares (PLS) analysis were used to compute blockwise reduced-rank approximations of music signals, which preserve the most prominent and temporally correlated spectral contributions. While PCA operates in a fully blind fashion, PLS can be equipped with side information such as the score representation, which can enhance prominent attributes of a music piece like the predominant melody. To obtain a PCA-based reduced-rank approximation a number of L signal frames in the time–frequency domain are collected in a matrix $\mathbf{U} \in \mathbb{C}^{M \times L}$ with M denoting the number of frequency bins. Then, the first $K < M$ eigenvectors of the covariance matrix $\mathbf{C} \sim \mathbf{U}\mathbf{U}^{\mathrm{H}}$ are computed, which are stored as column vectors of the matrix $\mathbf{W} \in \mathbb{C}^{M \times K}$. The resulting principal component scores are computed as the mapping $\mathbf{S} = \mathbf{W}^{\mathrm{H}}\mathbf{U}$. Exploiting the orthogonality of the eigenvectors, a rank-K approximation of $\mathbf{U}$ is obtained by

$$\hat{\mathbf{U}} = \mathbf{W}\mathbf{S} = \mathbf{W}\mathbf{W}^{\mathrm{H}}\mathbf{U}. \tag{3}$$

This procedure has to be repeated for each time–frequency block, $\mathbf{U}$. The simplified time–frequency representation, $\hat{X}(\mu, \lambda)$, is obtained by concatenating or overlap-adding the reduced-rank approximations, $\hat{\mathbf{U}}$.

Although dimensionality reduction techniques like PCA or PLS do not separate sources, they attenuate low-energy harmonics of both leading voices and accompaniments and, thus, achieve an effective reduction of the spectral complexity. For classical chamber music, this approach was found to be significantly preferred by CI users over unprocessed music as well as over an SS and remix procedure [21]. Spectrograms of a chamber music excerpt before and after PCA-based spectral complexity reduction are shown in Figure 5.

In general, the presented methods accentuate salient information such as predominant pitch and rhythm while reducing broad-band interference of multiple instruments. However, the limitations of the electrode-nerve bottleneck persist.

Improved sound coding

Music perception with CIs can also be improved through sound coding strategies. These sound coding strategies are inspired by the speech production source-filter model used in vocoders in telephone communication. The algorithmic design of the CI coding strategy depends on the manufacturer. Advanced bionics implements the high-resolution processing with fidelity (F120), which obtains an improved spectral distribution of stimulation across channels using simultaneous electrode stimulation; Cochlear implements the advanced combinational encoder (ACE), which performs a selection of bands to obtain a better spectral representation; Med-El implements the fine structure processing (FSP), which aims to improve the temporal representation of the input signal; and Oticon Medical implements the postspectral analysis strategy (CrystalisXDP), which puts emphasis on dynamic compression. Moreover, this system encodes loudness, adjusting the pulse duration while keeping the amplitude fixed [22]. For a review on the different sound coding strategies, refer to [23].

Figure 6 presents a block diagram of a generic sound coding strategy that includes a set of techniques to specifically improve music perception with a CI. Here, the signal from the microphone is processed through an adaptive-gain control (AGC). In contrast to listening to speech in noise, a reduction of compression is desired for music listening, although the small dynamic range of electric hearing limits this possibility. Some approaches, such as the CrystalisXDP, implement the AGC at the back end of the sound coding strategy to

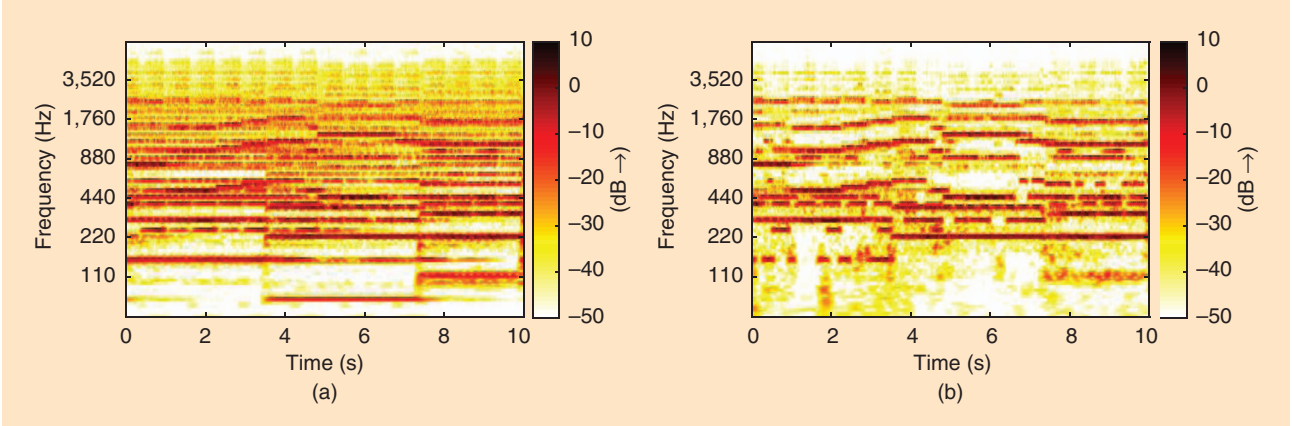


FIGURE 5. Spectrograms for a chamber music excerpt (Mozart, Clarinet Quintet in A major, K.581, II. Larghetto) using (a) the original sound and (b) a PCA-based reduced-rank approximation to reduce its complexity.

improve the dynamic in each analysis band. After the AGC, the signal is sent through a filter bank. The frequency bounds of the filter bank are approximately linearly spaced below 1,000 Hz and approach a logarithmic frequency scale above 1,000 Hz. An estimation of the envelope is calculated for each spectral band of the audio signal forming an analysis band. Each analysis band can be allocated to several electrodes and represents one channel. After envelope detection, sound coding strategies apply a modulation enhancement. Next, some bands may be selected for stimulation while discarding some others (the so-called N-of-M sound coding strategies because N bands are selected out of M possible ones). In the ACE strategy, the bands with the largest energies are selected, although psychoacoustic masking models have been also proposed for that purpose [24]. The selected bands are then mapped to the small dynamic range of electric hearing. Finally, a pulse is generated to stimulate the corresponding electrodes of that channel. A stimulation cycle is completed when all channels are stimulated, defining the stimulation rate in each electrode, which is also called the *channel stimulation rate*.

To improve music perception, detailed spectral and temporal sound features can be extracted and synthesized through novel stimulation modes, pulse shapes, or other stimulation parameters. The stimulation patterns delivered by the sound coding strategy, i.e., the current levels delivered by each electrode over time, are termed electrodograms. Electrodograms are useful to compare how acoustic features are transmitted

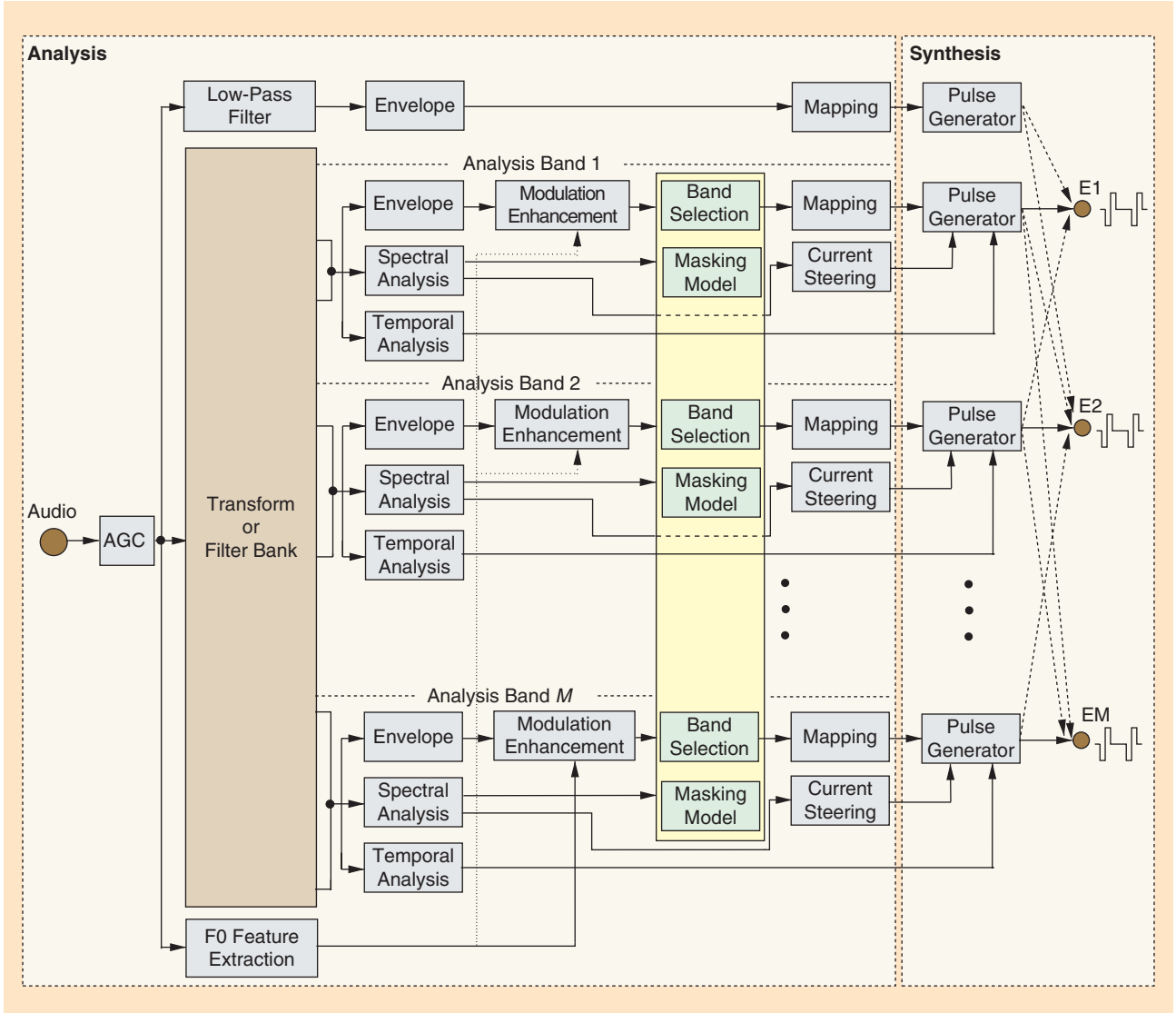


FIGURE 6. A basic block diagram of a generic sound coding strategy implementing different processing blocks to improve music perception with CIs. First, the audio signal is compressed using an AGC. Next, the sound is transformed into the frequency domain using a filter bank. The sound coding strategy includes a low-frequency analysis band that is transmitted through phantom electrode stimulation. In each analysis band, a modulation enhancement can be applied based on the F0 of the sound. Moreover, each analysis band may include a more detailed spectral and temporal analysis that is used to transmit the TFS or to be used in combination with current steering. Some analysis bands or spectral components are selected and others are discarded for stimulation using, e.g., a masking model. Finally, pulses are generated based on the compressed bands. Each analysis band can be associated to different electrode contacts to create specific stimulation modes as indicated by the dashed line arrows.

by different sound coding strategies. Figure 7 presents the electrodograms for a pop music excerpt. The electrodograms have been generated using an N-of-M sound coding strategy that uses 22 channels but only stimulates eight of them in each stimulation cycle. The stimulation rate on each channel was fixed to 500 pulses/s (Hz). For each electrode the amplitude of the stimulation current is represented by the height of vertical bars. Some strategies such as the F0mod [25] and the eTone [26] estimate the fundamental frequency F0 of the incoming sound and use it to enhance the modulation in the channel envelopes. Experiments in CI users show that explicit F0 modulation of the channel envelopes improves music perception [23]. The harmonic-single-sideband encoder [27] strategy obtains benefits in music perception by tracking the harmonics of a single musical source as represented by the spectral analysis block in Figure 6. This information is transformed into modulators conveying both amplitude and TFS. In a pilot study, some CI users obtained improvements in melody and timbre recognition with respect to the clinical baseline [27].

Some CI users have residual natural hearing at low frequencies in the same ear (via EAS) or the contralateral side (via so-called bimodal listening). Generally, this condition allows them to recognize melodies and instruments better, achieving a global music perception improvement compared to CI users who rely only on electrical stimulation, e.g., [8]. To this end, low-frequency signal contributions are transmitted acoustically, whereas for higher frequencies, the auditory nerve is stimulated electrically. The natural residual hearing available at the apex of the cochlea can then process TFS information, allowing for an enhanced perception of pitch and timbre cues. Still, further research is required to determine whether bimodal benefits can be enhanced by improving the compatibility of acoustic and electric stimulation [28].

CI users with no residual hearing can experience an insufficient stimulation in the apical part of the cochlea, since most CI electrode arrays are designed to be inserted only into the first (basal) 1 to 1.25 turns of the cochlea, e.g., [29], representing only frequencies of approximately 650 Hz and above along the spiral ganglion of a normal ear, e.g., [30]. Apical stimulation of the cochlea through deeply inserted electrodes has been shown to increase the range of place pitches, e.g., [29]. The transmission of TFS in the apical part of the cochlea also inspired the design of sound coding strategies to improve music perception. These strategies deliver pulses at a rate that corresponds with the estimation of the TFS, thus attempting to elicit neural responses in synchrony with the TFS [3]. In the FSP strategy [23], the fine structure of the sound is estimated in each analysis band using zero crossings together with deeply inserted electrodes as represented by the temporal analysis blocks in Figure 6. Using FSP, benefits for music perception and appreciation have been observed [31].

Improved stimulation modes

Most common sound coding strategies stimulate the electrodes in monopolar (MP) mode. In this mode, one active intracochlear electrode and one or two return extra-cochlear (EC) electrodes are used [Figure 8(a)]. The EC electrode is typically attached to the implant package. However, in some implants, it is possible to use an additional EC electrode placed under the temporalis muscle. In the MP mode, current flows between these widely spaced electrodes causing a broad current spread. For MP stimulation, it is commonly assumed that the centroid of the voltage distribution corresponds with the place pitch

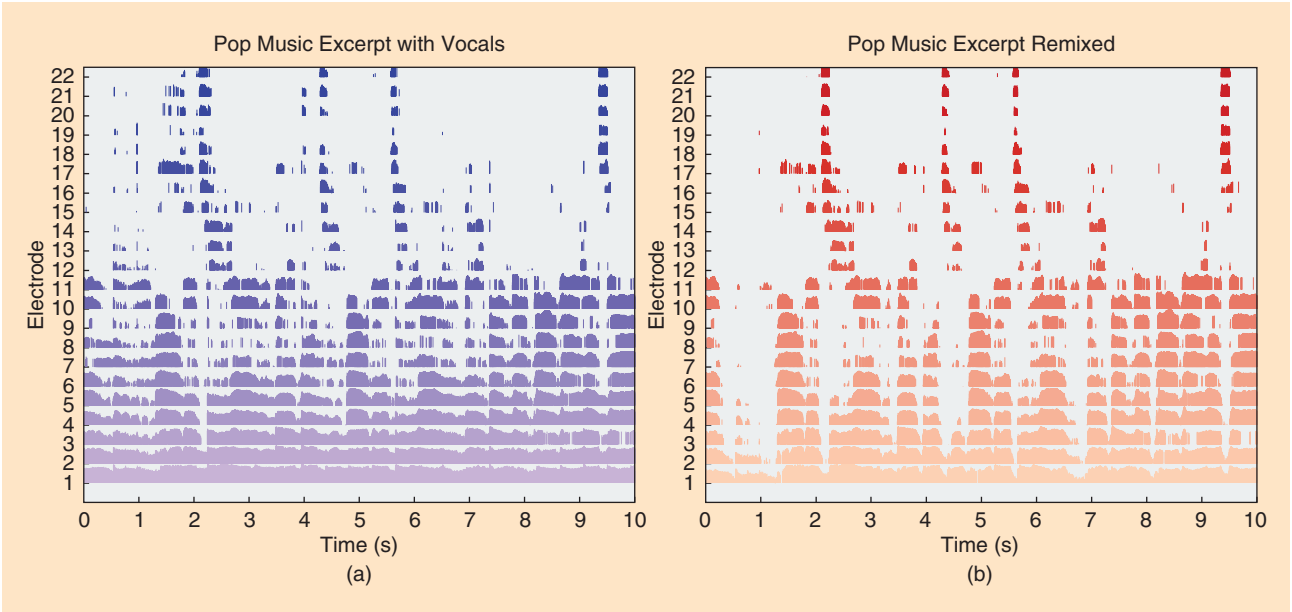


FIGURE 7. Electrodograms for a (a) sound and (b) remixed version for emphasizing the vocals. The sound corresponds to the excerpt 45378_chorus from the iKala data set (http://mac.citi.sinica.edu.tw/ikala/). The remixed version is created based on estimated multitracks using a DRNN to make the vocals 6 dB louder than the other music components. The lighter color indicates a lower frequency or more apical stimulation.

elicited by the stimulation and that the width of the voltage distribution limits the specificity of the stimulation, e.g., [9].

Several alternative stimulation modes have been proposed to improve sound perception with CIs. Phantom electrode (PE) stimulation is used to artificially extend the insertion depth in the cochlea by shaping the electrical field toward the apical regions [11]. PE electrode stimulation is achieved by simultaneously stimulating two adjacent intracochlear electrodes with out-of-phase pulses [Figure 8(e)]. If the basal electrode of the pair stimulates with a smaller amplitude than the apical electrode, the resulting electrical field is pushed away from the basal electrode, producing a lower pitch. PE stimulation can achieve pitch shifts equivalent to 0.5 to 2 MP electrodes without causing additional quality percepts, e.g., [32]. The questionnaire results showed a preference for PE when listening to music, which is likely driven by an improved balance between high and low frequencies [33]. Another method to extend the pitch sensation delivered by a CI is based on the assumption that short pulse phases are more effective (i.e., need less charge to evoke the same loudness) than longer phases and that anodic (positive) phases are more effective than cathodic (negative) ones [11]. Asymmetric pulses delivered simultaneously out-of-phase to two adjacent electrodes and at low pulse rates elicit a lower place pitch percept than symmetric pulses presented in the MP mode.

MP virtual channels or current steering is another form of electric field shaping that can be used to create pitch sensations corresponding to places between two physical electrodes [34] [Figure 8(b)]. The F120 coding strategy incorporates a high-resolution frequency estimator within each analysis band that is used to control the balance between simultaneously stimulated adjacent channels. In current steering, one electrode is stimulated with current $I\alpha$ and the adjacent electrode is simultaneously stimulated with current $I(1-\alpha)$. The parameter α can be varied between 0 and 1 to steer the locus of stimulation to sites between physical electrodes generating additional pitch percepts. Thus, both electrodes are stimulated with a total current I derived from the compressed envelope in each analysis band. The most dominant frequency in each analysis band is extracted based on a parabolic interpolation of spectral fast Fourier transform bins or by an analysis/synthesis matching pursuit algorithm combined with a psychoacoustic masking model (the SineEx sound coding strategy [24]). Some studies have shown improved music perception with a current steering strategy [24]. Still, even if more stimulation places can be created in the cochlea, the problem of current spread persists.

Improvements in spectral resolution performance can be also obtained using current focusing to reduce channel interaction. One current focusing implementation is tripolar (TP) stimulation [Figure 8(c)]. With TP stimulation, an active electrode is stimulated, and the two adjacent electrodes are stimulated in opposite polarity phase relative to the active electrode. Current focusing can be implemented in combination with current steering. Two central electrodes are then used for current steering, and the remaining two flanking electrodes are used

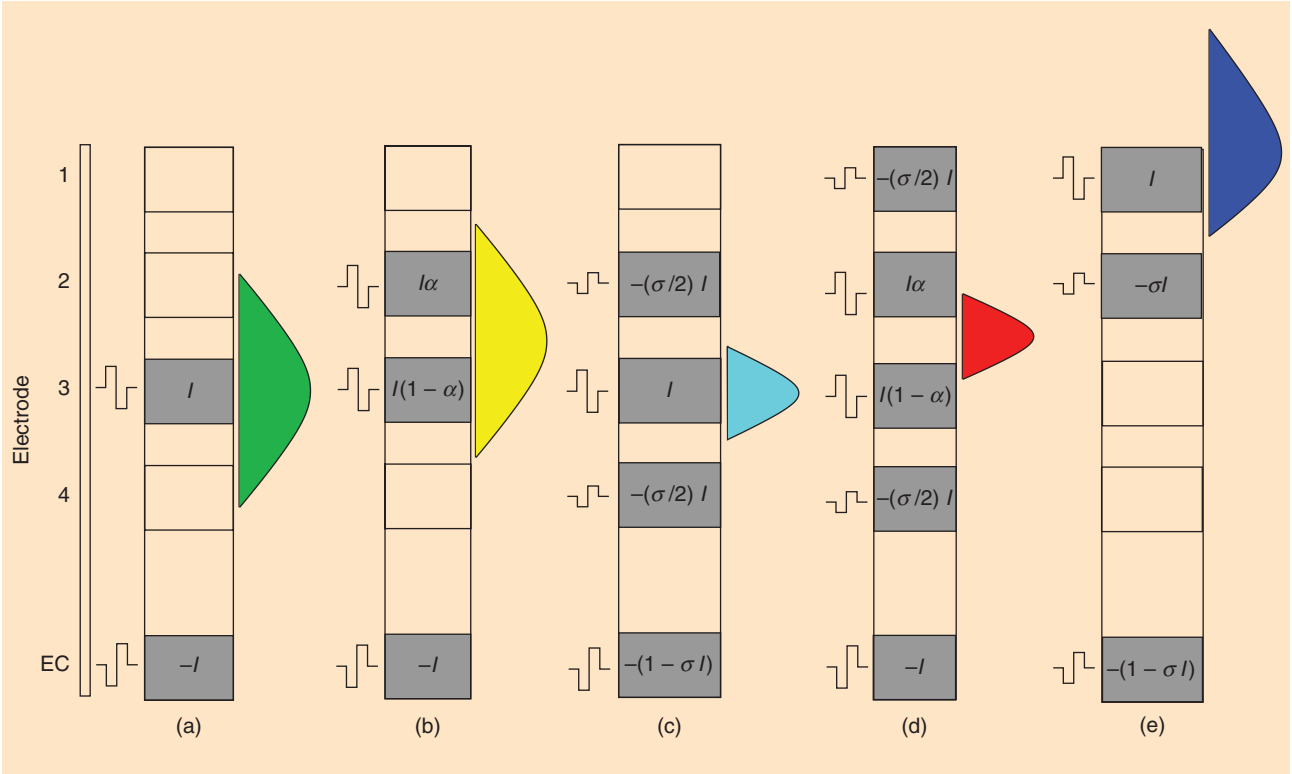


FIGURE 8. An illustration of different stimulation modes. The sign indicates the polarity, and the absolute value indicates the magnitude of the current, I, provided on the corresponding electrode. Electrode 1 corresponds to the most apical (lowest-frequency) contact. The vertical axis describes the electrode position [σ is the focusing coefficient ($0 < \sigma < 1$); α is the current steering coefficient ($0 < \sigma < 1$)]. For each stimulation mode the voltage distribution created around the electrode is presented. The (a) MP, (b) monopolar virtual channels, (c) TP, (d) quadrupolar virtual channels, and (e) PE modes.

as grounds to focus the stimulation as used in quadrupolar virtual channel stimulation [Figure 8(d)] [34].

The sound perception and performance that CI users obtain is variable depending on many factors, such as etiology of hearing loss, duration of deafness, and experience with the device. Moreover, CI users need time to adapt to the sound delivered by new sound coding strategies and stimulation modes. The large effects of these factors in the outcomes of clinical studies evaluating new sound coding strategies and stimulation modes often obscure their potential benefits.

Subjective measures for evaluating music performance with CIs

Techniques to make music more accessible for CI users, including compositions, signal preprocessing, sound coding strategies, and stimulation modes are typically evaluated through listening experiments. Music perception with CIs has been assessed through subjective evaluation procedures that quantify the accuracy of CI users to perceive the basic dimensions of music such as pitch, melody, timbre, and slower temporal features of music such as rhythm, tempo, and meter.

Pitch perception with CIs can be assessed through basic psychophysical experiments. For example, adaptive procedures have been employed to assess the number of current steering stimuli between two physical electrodes and midpoint comparison procedures have been used to quantify the upper limit of temporal pitch [11]. Multidimensional scaling (MDS) is another technique that has been shown to be very useful to understand which perceptual dimensions are affected by different stimulation parameters and modes. For example, [32] used MDS to demonstrate that a single perceptual dimension is responsible for the differences between PE and MP stimulation and to relate it to place pitch perception.

The sound perception and performance that CI users obtain is variable depending on many factors, such as etiology of hearing loss, duration of deafness, and experience with the device.

Melody perception in CI users can be assessed through forced choice tasks in which a melody is presented and the subject is asked to identify the melody from a given subset. A special melody test is the melodic contour identification (MCI) task [35]. MCI presents simple melodic contours composed of five notes and the subject is asked to identify them. Moreover, the timbre of each note or contour can be modified to assess the interaction between melody and timbre [35]. Timbre perception is typically evaluated through instrument identification tasks. Rhythm perception can be evaluated through forced choice procedures, in which stimuli are presented in pairs with different levels of complexity and the participants are asked whether the two stimuli are the same or different [36]. Tempo perception has been investigated, presenting pairs of rhythmic patterns with different tempos (60, 80, 100, or 120 beats/min). Participants were instructed to select the faster rhythm in the pair. Perception of meter has been investigated asking participants to categorize a piece as either march (double meter) or waltz (triple meter). Results from all these tests suggest that rhythm, tempo, and meter are well preserved in electric hearing [7].

Several test environments have been proposed integrating these different tests as music evaluation and training tools such as the University of Washington Clinical Assessment of Music Perception Test [37] or the MuSIC-test battery [36], among others. However, these tests require long testing periods to evaluate music perception in all its dimensions. For this reason, researchers have investigated faster methods to evaluate music perception as a whole with a single test. In that respect, the spectral ripple discrimination test has acquired a lot of popularity [29]. It evaluates the ability to resolve spectrotemporal modulations. Results have been shown to correlate significantly with melody and timbre recognition as well as pitch direction discrimination [12].

Music perception has been investigated, hoping that it relates to music appreciation as well. However, it has been shown that for CI users, music perception and appreciation are two different aspects that do not necessarily correlate with each other [7]. In this context, music appreciation tests have been developed to evaluate the sound quality elicited by different preprocessing and sound coding strategies. These tests are commonly based on subjective scaling questionnaires, e.g., [7], and are used to rate different descriptive adjectives of the sound (e.g., clean/noisy, bass/treble, poor/excellent, and pleasant/unpleasant). However, several studies have shown that different factors such as age, gender, level of musical training, previous familiarity with a musical style, and even personality can all influence ratings [7]. Consequently, these factors can hide the real acoustic or electric sound features affecting the quality of the sound.

Music signal preprocessing algorithms and sound coding strategies can be assessed through paired comparisons between unprocessed and processed music excerpts, e.g., [19]–[21], or between a baseline and a new sound coding strategy, e.g., [33]. However, the amount of paired comparisons increases dramatically when considering different conditions and algorithmic settings. To this end, the Multiple Stimuli with Hidden Reference and Anchor (MUSHRA [44]) test can be used to reduce the number of comparisons, because it was developed to be used in normal-hearing expert listeners. The CI-MUSHRA modifies the classic MUSHRA in that it offers researchers the opportunity to quantify specific features of music that contribute to sound quality impairments based on the acoustic or the electric parameters chosen to modify the stimuli [7]. Moreover, CI-MUSHRA scores can be compared across populations to understand how acoustic and electric sound features influence music appraisal. For example, the CI-MUSHRA has been used to quantify how much acoustic parameters contribute to musical sound quality deficits for CI users. For instance, [7] compared mean sound quality ratings between CI users and normal-hearing controls for stimuli with

various amounts of low-frequency content. Overall, CI users were less sensitive to sound quality differences as a function of low-frequency information, as compared to normal-hearing controls. In particular, CI users required a lower cutoff of at least 400 Hz before detecting an impairment in sound quality relative to the reference, suggesting that limitations in low-frequency representation contribute to musical sound quality deficits [7]. On the basis of these findings, the CI-MUSHRA test was used to investigate the effects on low-frequency perception produced by the PE and the FSP sound coding strategies.

The auditory-distortion ratio measure estimates changes in perceived auditory distortion in comparison with an unprocessed music piece.

Instrumental measures of music quality for CIs

As a supplement to time-consuming listening experiments with CI users, instrumental measures for predicting the benefits of music processing algorithms have recently been proposed [20], [38]. To this end, the auditory-distortion ratio (ADR) measure [20] estimates changes in perceived auditory distortion in comparison with an unprocessed music piece. This procedure is based on a spectral smearing technique [39], which simulates a reduction of frequency selectivity in hearing-impaired listeners by broadened auditory filters and has previously been used for speech intelligibility prediction. The ADR, primarily developed for harmonic signals, is defined as

$$\mathrm{ADR} = 10\log\left(\frac{\sum_n [x(n)-\tilde{x}(n)]^2}{\sum_n \left[\hat{x}(n)-\tilde{\hat{x}}(n)\right]^2}\right)\mathrm{dB}, \tag{4}$$

where $x(n)$, $\hat{x}(n)$, $\tilde{x}(n)$, and $\tilde{\hat{x}}(n)$ denote the original music signal, a processed version of the music signal, the spectrally smeared version of the original music signal, and the processed and subsequently smeared music signal, respectively. The numerator measures the error between the original signal and its spectrally smeared version, whereas the denominator quantifies the deviation between a processed signal and its spectrally smeared counterpart. If the latter error is smaller, the ADR becomes positive, and hence, an improvement is indicated pointing toward less auditory distortion. Note, that ADR values in dB are rather small since they measure the distortion of relatively weak higher-order harmonics. Figure 9(a) depicts the ADR evaluation results for a database of 110 classical chamber music pieces that were simplified by PCA-based or PLS-based reduced-rank approximations or using a SS and remix approach [20]. The ADR was computed using auditory filters with a broadening factor of three. The results are shown as a function of the source-to-interference ratio (SIR), which was used to adjust the SS and remix procedure. For the reduced-rank approximations also, the number of retained components K is indicated. Figure 9(b) plots listening test results obtained from 14 CI users against the results of the ADR measure. For the listening test, specific settings of each algorithm (i.e., PCA and PLS with eight or 13 retained components and the SS and remix approach with 3- or 5-dB attenuation of the accompaniment) were compared to the unprocessed original [21]. The test results indicate the averaged preference scores for each method and setting. The plot shows that a considerable proportion of variance in the preference scores can be explained by the ADR measure $(R^2 = 0.595)$. Moreover, a model for predicting different perceptual dimensions of music quality (e.g., naturalness, distortion, or complexity) has been proposed recently [38]. To account for effects of suboptimal music transmission by CIs, a listening test was performed with

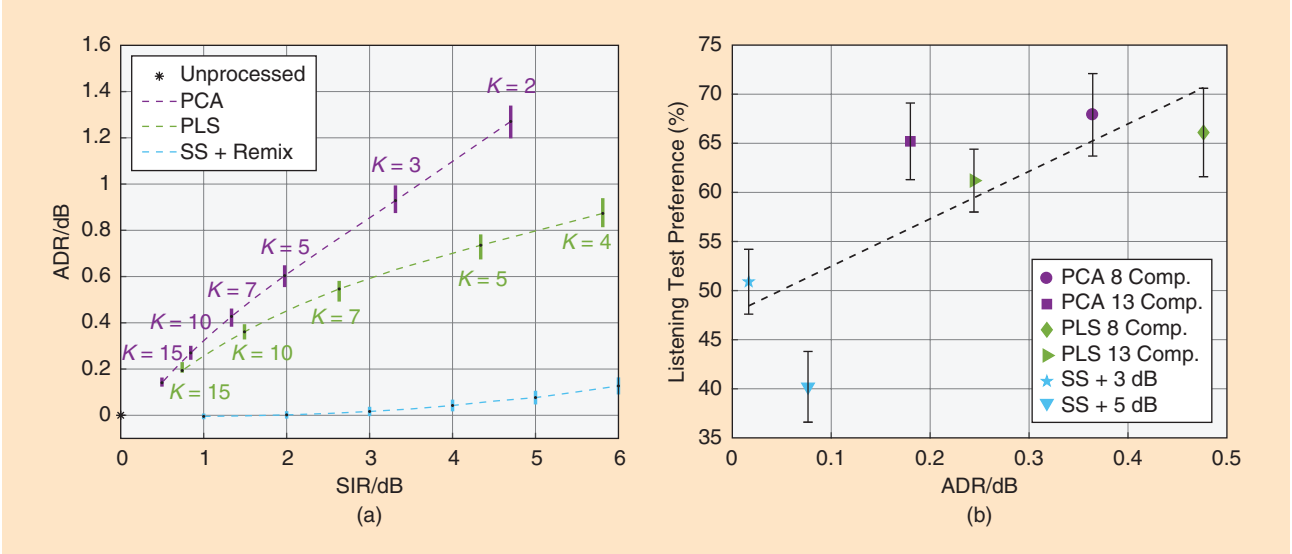


FIGURE 9. (a) An instrumental evaluation of different music preprocessing methods for a database of classical chamber music and (b) a comparison with actual listening test preference scores provided by CI users. The ADR measure is plotted as a function of the SIR and the number of retained components, K. Error bars indicate 95% confidence intervals and standard errors in (a) and (b), respectively.

ten CI users who rated excerpts of music on bipolar scales. A model for predicting the median ratings across CI users was developed using principal component regression and signal-based features to estimate the median ratings on the bipolar scales. The most successful features describe the amount of high-frequency energy in the signal, the frequency region with the highest concentration of spectral energy, the spectral bandwidth, and the degree of dissonance between pairs of spectral peaks.

In the music SS research community, the distortions introduced by the algorithms are typically objectively quantified by the source-to-distortion ratio, the source-to-artifacts ratio (SAR), and the SIR in dB units, e.g., [19]. These objective measures have also been applied in the field of CIs to assess the amounts of artifacts accepted by CI users to remix music. CI users were asked, through a pairwise comparison test, about their preference for different mixing presets using original and estimated multitracks from SS algorithms. The results from the pair-wise comparison in percentage were juxtaposed with the objective SS measures in dB to study which levels cause a change in preference from estimated to original multitracks. With this new methodology, the suitability for remixing music to improve CI users musical experience of novel SS algorithms can be directly considered.

Evaluation of music perception in live concerts by CI and normal-hearing listeners can give very valuable information to design new signal processing algorithms and sound coding strategies for music listening.

Conclusions

Music processing for CIs is an emerging and still under-researched topic in signal processing. It draws its importance from the increasing number of CI users worldwide and their desire to enjoy music in similar ways as normal-hearing people. To this end, this article has presented an overview of different approaches to make music more accessible for CI users as well as subjective and objective instrumental measures to evaluate their success. These advances have been presented based on the technical and perceptual limitations that CI users are exposed to when listening to music.

Several attempts have been made to create music with clear melodies and rhythms such that it can be better perceived by CI users. Evaluation of music perception in live concerts by CI and normal-hearing listeners can give very valuable information to design new signal processing algorithms and sound coding strategies for music listening.

Several sound coding strategies aim to improve pitch perception and melody recognition by enhancing the transmission of temporal aspects of the input sound in the apical part of the cochlea. Other approaches increase the number of stimulation locations beyond those provided by the physical electrodes attempting to improve frequency resolution. However, the perception of pitch, melody, and timbre still remains poor for most CI users.

Limitations in music perception by CI users inspired the design of signal processing algorithms that reduce the complexity of music. SS techniques based on NMF or DRNNs have been shown to be beneficial to remix music with clear vocals or enhanced melodies. Music complexity reduction based on PCA has been successfully implemented and evaluated in CI users showing that spectrally less complex music is preferred.

Future work should aim at a unification of preprocessing and sound coding to improve the efficiency of algorithms and to reduce their complexity and latency. Further research is necessary to develop improved objective instrumental measures of music perception and appreciation because subjective tests are time consuming and it is known that there is large variability in the results. Moreover, subjective tests in more realistic environments are required. Today, it is possible to reproduce music events in virtual reality laboratories that can be used to assess the perception of CI users in a more controlled manner than in live concerts. Therefore, real-time sound coding and signal processing techniques can be optimized and evaluated in virtual environments to facilitate future research efforts toward making music more accessible for CI users.

Acknowledgments

We would like to thank Tom Gajecki, Johannes Gauer, and Claus Weihs for fruitful discussions. We apologize for not being able to include all original references due to space constraints. This work was supported by the German Research Foundation (DFG) Cluster of Excellence EXC 1077/1 Hearing4all and the DFG-funded Collaborative Research Center 823, Subproject B3.

Authors

Waldo Nogueira (nogueiravazquez.waldo@mh-hannover.de) received his Dipl.-Ing. and Dr.-Ing. degrees from the Polytechnic University of Catalonia, Barcelona, Spain, and the Leibniz University of Hannover, Germany, in 2003 and 2008, respectively. From 2011 until 2013, he held a postdoctoral position at the Music Technology Group of the Pompeu Fabra University in Barcelona. He is currently with the Medical University Hannover, Cluster of Excellence Hearing4all, Germany. In 2008, he joined the research and development laboratories of advanced bionics in Belgium and Germany. His main research interest focuses on audio signal processing for auditory devices.

Anil Nagathil (anil.nagathil@rub.de) studied electrical engineering and information technology at Ruhr-Universität Bochum, Germany, and the University of Birmingham, United Kingdom. He received his Dipl.-Ing. and Dr.-Ing. degrees from Ruhr-Universität Bochum in 2009 and 2016, respectively. He is with the Ruhr-Universität Bochum Institute of Communication Acoustics, Germany, as a postdoctoral researcher. His research interests include statistical speech and audio signal processing for applications in hearing instruments.

Rainer Martin (rainer.martin@rub.de) received his M.S. degree in electrical engineering from the Georgia Institute of Technology, Atlanta, in 1989 and his Dipl.-Ing. and Dr.-Ing.

degrees from RWTH Aachen University, Germany, in 1988 and 1996, respectively. He is with the Ruhr-Universität Bochum, Institute of Communication Acoustics, Germany. Since 2003, he has been a chaired professor of information technology and communication acoustics at Ruhr-Universität Bochum. His research interests are signal processing for voice communication systems, hearing instruments, and human-machine interfaces. He coordinated two major European Union projects (AUDIS and IcanHear) in the area of signal processing for hearing instruments. He is a Fellow of the IEEE.

References

[1] M. Müller, *Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications*. New York: Springer, 2015.

[2] W. Wuerfel, H. Lanfermann, T. Lenarz, and O. Majdani, "Cochlear length determination using cone beam computed tomography in a clinical setting," *Hear. Res.*, vol. 316, pp. 65–72, Oct. 2014.

[3] I. Hochmair, E. Hochmair, P. Nopp, M. Waller, and C. Jolly, "Deep electrode insertion and sound coding in CIs," *Hear. Res.*, vol. 322, pp. 14–23, Apr. 2015.

[4] B. S. Wilson and M. F. Dorman, "Cochlear implants: A remarkable past and a brilliant future," *Hear. Res.*, vol. 242, pp. 3–21, Aug. 2008.

[5] H. J. McDermott, "Music perception with CIs: A review," *Trends Amplif.*, vol. 8, no. 2, pp. 49–82, 2004.

[6] V. Looi, K. Gfeller, and V. Driscoll, "Music appreciation and training for cochlear implant recipients: A review," *Semin. Hear.*, vol. 33, no. 4, pp. 307–334, 2012.

[7] C. J. Limb and A. T. Roy, "Technological, biological, and acoustical constraints to music perception in cochlear implant users," *Hear. Res.*, vol. 308, pp. 13–26, Feb. 2014.

[8] K. E. Gfeller, C. Olszewski, B. Gantz, C. Turner, and J. Oleoson, "Music perception with cochlear implants and residual hearing," *Audiol. Neurotol.*, vol. 11, pp. 12–15, Oct. 2006.

[9] J. Laneau, J. Wouters, and M. Moonen, "Improved music perception with explicit pitch coding in cochlear implants," *Audiol. Neurotol.*, vol. 11, pp. 38–52, Jan. 2006.

[10] M. Maarefvand, J. Marozeau, and P. J. Blamey, "Pitch matching in bimodal cochlear implant patients: Effects of frequency, spectral envelope and level," *J. Acoust. Soc. Amer.*, vol. 142, no. 5, pp. 2854–2865, 2018.

[11] O. Macherey, J. M. Deeks, and R. P. Carlyon, "Extending the limits of place and temporal pitch perception in cochlear implant users," *J. Assoc. Res. Otolaryngol.*, vol. 12, no. 2, pp. 233–251, 2011.

[12] W. R. Drennan and J. T. Rubinstein, "Music perception in cochlear implant users and its relationship with psychophysical capabilities," *J. Rehab. Res. Develop.*, vol. 45, no. 5, pp. 779–789, 2008.

[13] L. A. J. Reiss, R. A. Ito, J. L. Eggleston, J. J. Becker, S. Liao, C. E. Lakin, F. M. Warren, S. O. McMenomey, "Pitch adaptation patterns in bimodal cochlear implant users: Over time and after experience," *Ear Hear.*, vol. 36, no. 2, pp. e23–e34, 2015.

[14] A. Au, J. Marozeau, H. Innes-brown, E. Schubert, and C. J. Stevens, "Music for the CI: Audience response to six commissioned compositions," *Semin. Hear.*, vol. 1, no. 212, pp. 335–345, 2012.

[15] W. Buyens, B. van Dijk, M. Moonen, and J. Wouters, "Music mixing preferences of CI recipients: A pilot study," *Int. J. Audiol.*, vol. 53, no. 5, pp. 294–301, 2014.

[16] G. D. Kohlberg, D. M. Mancuso, D. A. Chari, and A. K. Lalwani, "Music engineering as a novel strategy for enhancing music enjoyment in the CI recipient," *Behav. Neurol.*, vol. 2015, 2015. doi: 10.1155/2015/829680.

[17] W. Buyens, B. van Dijk, J. Wouters, and M. Moonen, "A harmonic/percussive sound separation based music pre-processing scheme for cochlear implant users," in *Proc. European Signal Process. Conf. (EUSIPCO)*, 2013, pp. 1–5.

[18] W. Buyens, B. van Dijk, J. Wouters, and M. Moonen, "A stereo music preprocessing scheme for cochlear implant users," *IEEE Trans. Biomed. Eng.*, vol. 62, no. 10, pp. 2434–2442, 2015.

[19] T. Gajecki and W. Nogueira, "Deep learning models to remix music for cochlear implant users," *J. Acoust. Soc. Amer.*, vol. 143, no. 6, pp. 3602, June 2018.

[20] A. Nagathil, C. Weihs, and R. Martin, "Spectral complexity reduction of music signals for mitigating effects of cochlear hearing loss," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 24, no. 3, pp. 445–458, 2016.

[21] A. Nagathil, C. Weihs, K. Neumann, and R. Martin, "Spectral complexity reduction of music signals based on frequency-domain reduced-rank approximations: An evaluation with cochlear implant listeners," *J. Acoust. Soc. Amer.*, vol. 142, no. 3, pp. 1219–1228, Sept. 2017.

[22] B. Vaerenberg, P. J. Govaerts, T. Stainsby, P. Nopp, A. Gault, and D. Gnansia, "A uniform graphical representation of intensity coding in current generation cochlear implant systems," *Ear Hear.*, vol. 35, pp. 533–543, Sept. 2015.

[23] J. Wouters, H. J. McDermott, and T. Francart, "Sound coding in cochlear implants: From electric pulses to hearing," *IEEE Signal Process. Mag.*, vol. 32, no. 2, pp. 67–80, 2015.

[24] W. Nogueira, L. Litvak, B. Edler, J. Ostermann, and A. Büchner, "Signal processing strategies for cochlear implants using current steering," *EURASIP J. Adv. Signal Process.*, vol. 2009, no. 15, 2009.

[25] M. Milczynski, J. Wouters, and A. Wieringen, "Improved fundamental frequency coding in CI signal processing," *J. Acoust. Soc. Amer.*, vol. 123, no. 4, pp. 2260–2271, 2009.

[26] A. E. Vandali, R. J. M. Van Hoesel, A. E. Vandali, and R. J. M. Van Hoesel, "Enhancement of temporal cues to pitch in CIs: Effects on pitch ranking," *J. Acoust. Soc. Amer.*, vol. 132, no. 392, pp. 392–402, 2012.

[27] X. Li, K. Nie, N. S. Imennov, J. T. Rubinstein, and L. E. Atlas, "Improved perception of music with a harmonic based algorithm for cochlear implants," *IEEE Trans. Neural Syst. Rehab. Eng.*, vol. 21, no. 4, pp. 684–694, 2013.

[28] C. M. Sucher and H. J. McDermott, "Bimodal stimulation: Benefits for music perception and sound quality," *Cochlear Implants Int.*, vol. 10, suppl. 1, pp. 96–99, 2009.

[29] D. M. Landsberger, G. Mertens, A. K. Punte, and P. Van De Heyning, "Perceptual changes in place of stimulation with long cochlear implant electrode arrays," *J. Acoust. Soc. Amer.*, vol. 135, no. 2, pp. 75–81, Feb. 2014.

[30] O. Stakhovskaya, D. Sridhar, B. H. Bonham, and P. A. Leake, "Frequency map for the human cochlear spiral ganglion: Implications for cochlear implants," *J. Assoc. Res. Otolaryngol.*, vol. 8, no. 2, p. 220, 2007.

[31] C. Arnoldner, D. Riss, M. Brunner, M. Durisin, W.-D. Baumgartner, and J.-S. Hamzavi, "Speech and music perception with the new fine structure speech coding strategy: preliminary results," *Acta Otolaryngol.*, vol. 127, no. 12, pp. 1298–1303, Jan. 2007.

[32] S. Klawitter, D. M. Landsberger, A. Büchner, and W. Nogueira, "Perceptual changes with monopolar and phantom electrode stimulation," *Hear. Res.*, vol. 359, pp. 64–75, Mar. 2018.

[33] W. Nogueira, L. M. Litvak, A. A. Saoji, and A. Büchner, "Design and evaluation of a CI strategy based on a phantom channel," *PloS One*, vol. 10, no. 3, 2015.

[34] D. M. Landsberger and A. Srinivasan, "Virtual channel discrimination is improved by current focusing in cochlear implant recipients," *Hear. Res.*, vol. 254, no. 1-2, pp. 34–41, Aug. 2009.

[35] J. Galvin, Q.-J. Fu, and S. Oba, "Effect of a competing instrument on melodic contour identification by CI users," *J. Acoust. Soc. Amer.*, vol. 125, no. 3, pp. 98–103, 2009.

[36] S. J. Brockmeier, D. Fitzgerald, O. Searle, H. Fitzgerald, M. Grasmeder, S. Hilbig, K. Vermiere, M. Peterreins, S. Heydner, and W. Arnold, "The MuSIC perception test: A novel battery for testing music perception of cochlear implant users," *Cochlear Implants Int.*, vol. 12, no. 1, 2011.

[37] R. Kang, G. L. Nimmons, W. Drennan, J. Longnion, C. Ruffin, K. Nie, J. H. Won, T. Worman, B. Yueh, and J. Rubinstein, "Development and validation of the University of Washington Clinical Assessment of Music Perception Test," *Ear Hear.*, vol. 30, no. 4, pp. 411–418, 2009.

[38] A. Nagathil, J.-W. Schlattmann, K. Neumann, and R. Martin, "A featured-based linear regression model for predicting perceptual ratings of music by cochlear implant listeners," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Process. (ICASSP)*, 2017, pp. 346–350.

[39] B. C. J. Moore, B. R. Glasberg, and A. Simpson, "Evaluation of a method of simulating reduced frequency selectivity," *J. Acoust. Soc. Amer.*, vol. 91, no. 6, pp. 3402–3423, 1992.

[40] World Health Organization. (2018, Mar. 15). Deafness and hearing loss. [Online]. Available: http://www.who.int/mediacentre/factsheets/fs300/en/

[41] National Institute for Health and Care Excellence. (2018). [Online]. Available: https://www.nice.org.uk/

[42] The Ear Foundation. (2018). Cochlear implants. [Online]. Available: http://www.earfoundation.org.uk/hearing-technologies/cochlear-implants/

[43] Music for Cochlear Implants. (2017). [Online]. Available: www.music4ci.com

[44] *Recommendation ITU-R. Method for the Subjective Assessment of Intermediate Quality Level of Coding Systems*, ITU-R BS.1534-1, 2003.

SP

Brian McFee, Jong Wook Kim, Mark Cartwright,
Justin Salamon, Rachel Bittner, and Juan Pablo Bello

Open-Source Practices for Music Signal Processing Research

Recommendations for transparent, sustainable, and reproducible audio research

©ISTOCKPHOTO.COM/TRAFFIC_ANALYZER

Digital Object Identifier 10.1109/MSP.2018.2875349
Date of publication: 24 December 2018

In the early years of music information retrieval (MIR), research problems were often centered around conceptually simple tasks, and methods were evaluated on small, idealized data sets. A canonical example of this is genre recognition—i.e., Which one of *n* genres describes this song?—which was often evaluated on the GTZAN data set (1,000 musical excerpts balanced across ten genres) [1]. As task definitions were simple, so too were signal analysis pipelines, which often derived from methods for speech processing and recognition and typically consisted of simple methods for feature extraction, statistical modeling, and evaluation. When describing a research system, the expected level of detail was superficial: it was sufficient to state, e.g., the number of mel-frequency cepstral coefficients used, the statistical model (e.g., a Gaussian mixture model), the choice of data set, and the evaluation criteria, without stating the underlying software dependencies or implementation details. Because of an increased abundance of methods, the proliferation of software toolkits, the explosion of machine learning, and a focus shift toward more realistic problem settings, modern research systems are substantially more complex than their predecessors. Modern MIR researchers must pay careful attention to detail when processing metadata, implementing evaluation criteria, and disseminating results.

Reproducibility and Complexity in MIR

The common practice in MIR research has been to publish findings when a novel variation of some system component (such as the feature representation or statistical model) led to an increase in performance. This approach is sensible when all relevant factors of an experiment can be enumerated and controlled and when the researchers have confidence in the correctness and stability of the underlying implementation. However, over time, researchers have discovered that confounding factors were prevalent and undetected in many research systems, which undermines previous findings. Confounding factors can arise from quirks in data collection [2], subtle design choices in feature representations [3], or unstated assumptions in the evaluation criteria [4].

As it turns out, implementation details can have greater impacts on overall performance than many practitioners might

expect. For example, Raffel et al. [4] reported that differences in evaluation implementation can produce deviations of 9–11% in commonly used metrics across diverse tasks including beat tracking, structural segmentation, and melody extraction. This results in a manifestation of the reproducibility crisis [5] within MIR: if implementation details can have such a profound effect on the reported performance of a method, it becomes difficult to trust or verify empirical results. Reproducibility is usually facilitated by access to common data sets, which would allow independent reimplementations of a proposed method to be evaluated and compared with published findings. However, MIR studies often rely on private or copyrighted data sets that cannot be shared openly. This shifts the burden of reproducibility from common data to common software: although data sets often cannot be shared, implementations usually can.

In this article, we share experiences and advice gained from developing open-source software (OSS) for MIR research with the hope that practitioners in other related disciplines will benefit from our findings and become effective developers of open-source scientific software. Many of the issues we encounter in MIR applications are likely to recur in more general signal processing areas as data sets increase in complexity, evaluation becomes more integrated and realistic, and traditionally small research components become integrated with larger systems.

Open-source scientific software

We agree with numerous authors [6] that description of research systems is no longer sufficient, which follows from the position that scholarly publication serves primarily as advertisement for the scientific contributions embodied by the software and data [7]. Here, we specifically advocate for adopting modern OSS development practices when communicating scientific results.

The motivations for our position, although grounded in music analysis applications, apply broadly to any field in which systems reach a sufficiently high degree of complexity. Releasing software as open source requires more than posting code on a website. We highlight several key ingredients of good research software practices:

- *licensing*: to define the conditions under which the software can be used
- *documentation*: so that users know how to operate the software and what exactly it does
- *testing*: so that the software is reliable
- *packaging*: so that the software can be easily installed and managed in an environment
- *application interface design*: so that the software can be easily integrated with other tools.

We discuss best practices for OSS development in the context of MIR applications and propose future directions for incorporating open-source and open-science methodology in the creation of data sets.

System architecture and components

Figure 1 shows a generic but representative MIR system pipeline consisting of seven distinct stages. We describe each stage to provide a sense of scale involved in MIR research, document sources of software dependencies, and give pointers to common components.

The first stage is data storage, which is often implemented by organizing data on a disk according to a file naming convention and directory structure. Storage may also be provided by relational databases (e.g., SQLite [8]), key value/document stores (e.g., MongoDB at https://www.mongodb.com or Redis at https://redis.io), or structured numerical data formats (e.g., HDF5 [9]). As data sets become larger and more richly structured, storage plays a critical role in the overall system.

Input decoding, the second stage, loosely captures the transformation of raw data (compressed audio or text data) into formats more convenient for modeling (typically vector representations). For audio, this consists primarily of compression codecs, which are provided by a few standard libraries (e.g., ffmpeg [10] or libsndfile [11]). Although different (lossy) codec implementations are not guaranteed to produce numerically equivalent results, the differences are usually small enough to be ignored for most practical applications. For annotations and metadata, the situation is less clear. Many data sets are provided in nonstandard formats (e.g., comma-separated values) that require custom parsers that can be difficult to correctly implement and validate. Although several formats have been proposed for encoding annotations and metadata (MusicXML [12], MEI [13], MPEG-7 [14], and JAMS [15]), at this point none have emerged as a clear standard in the MIR community.

The third stage, synthesis and augmentation, is not universal, but it has seen rapid growth in recent years. This stage captures processes that automatically modify or expand data sets, usually with the aim of increasing the size or diversity of training sets for fitting statistical models. Data augmentation methods apply systematic perturbations to an annotated data set, such as pitch shifting or time stretching, to induce these properties as invariants in the model [16]. Relatedly, degradation methods apply similar techniques to evaluation data as a means of diagnosing failure modes in a model once its parameters have been estimated [17]. Synthesis methods, like augmentation, seek to generate realistic examples either for training or evaluation, and, although the results are synthetic, they are free of annotation errors [18]. Because these

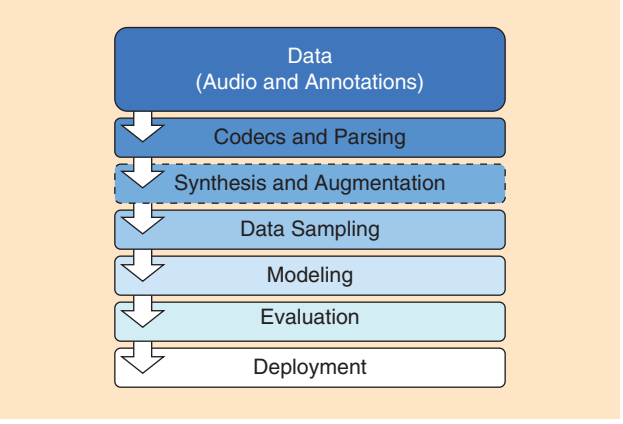


FIGURE 1. A system block diagram of a typical MIR pipeline.

processes can have a profound impact on the resulting model, it is important that augmentation and synthesis be fully documented and reproducible. Modern frameworks such as MUDA [16] and Scaper [18] achieve data provenance by embedding the generation/augmentation parameters within the generated objects, thereby facilitating reproducibility.

Data sampling, the fourth stage, refers to how a collection is partitioned and sampled when fitting statistical models. For statistical evaluation, data are usually partitioned into training and testing subsets, and this step is usually implemented within a machine-learning toolkit (e.g., SciKit-Learn [19]). We emphasize data partitioning because it can be notoriously difficult to implement correctly when dealing with related samples, such as multiple songs by a common artist [20]. Stochastic sampling is an increasingly important step, because it defines the sequences of examples used to estimate model parameters. Modern methods trained by stochastic gradient descent can be sensitive to initialization and sampling, so it is important that the entire process be codified and reproducible. Often, sampling is specified only implicitly and is provided by machine-learning frameworks without explicit reproducibility guarantees. Sampling also becomes an engineering challenge when the training data exceed the memory capacity of the system, which is common when dealing with large data sets. For problems involving large data sets, some framework-independent libraries have been developed to handle data sampling under resource constraints (e.g., Pescador [21] and Fuel [22]).

Modeling, as the fifth stage, includes both feature extraction and statistical modeling, although the boundary between the two has blurred in recent years with the adoption of deep-learning methods. Many open-source libraries exist for audio feature extraction, such as Essentia [23], librosa [24], aubio [25], Madmom [26], or Marsyas [27]. Different libraries may produce different numerical representations for the same feature (e.g., mel spectra), and even within a single library the robustness of different features to input encoding/decoding may vary [28]. Although robustness is distinct from reproducibility, it highlights the importance of sharing specific software implementations. The statistical modeling component is most often provided by a machine-learning framework, such as SciKit-Learn or Keras [29]. Although the specific choice of framework is largely up to the practitioner's discretion, we emphasize that consideration should be given to how this choice interacts with the remaining two stages.

Referring to measuring the performance of an entire developed system (not just the statistical model component) is the sixth stage, evaluation. For simple classification problems, this functionality is typically provided by a machine-learning framework (e.g., SciKit-Learn). However, for domain-specific MIR problems, software packages have been developed to standardize evaluations, such as mir_eval for music description and source separation [4], sed_eval for sound event detection [30], and rival for recommender systems [31].

Finally, the last stage is deployment, by which we broadly mean dissemination of results (publication), packaging for reuse, or practical application in a real setting. This stage is perhaps the most overlooked in research and is possibly the most difficult to approach systematically, because the requirements vary substantially across projects. If we limit attention to reproducibility, software packaging emerges as an integral step to both internal reuse and scholarly dissemination. We therefore encourage researchers to take an active role in packaging their software components, and in the "Best Practices for OSS Research" section we discuss specific tools for packaging and environment management.

Example: Onset detection

Although we focus on large, integrated systems, it is instructive to see how system complexity plays out on a smaller scale representative of earlier MIR work. As a conceptually simple example task, consider onset detection: the problem of estimating the timing of the beginning of musical notes in a recording. A method for solving this problem could be described next.

Audio was converted to 22,050 Hz (mono), and a 2,048-point short-time Fourier transform (STFT) was computed with a 64-sample hop. The STFT was reduced to 128 mel-frequency bands, and magnitudes were compressed by log scaling. An onset envelope was computed using thresholded spectral differencing, and peaks were selected using the method of Böck et al. [32]. This description is artificial, but the level of specificity given is representative of the literature.

Although precise enough to be approximately reimplemented by a knowledgeable practitioner, the description omits several details. To quantify the effect of these details, we conducted an experiment in which some unstated parameters were varied, and the resulting accuracy was measured on a standard data set [33]. We varied the window function for STFT (Hann or Hamming), the log scaling [bias-stabilized $\log(1 + X)$ or clipped 80 dB below peak magnitude], and the differencing operator (first-order difference or a Savitsky–Golay filter, as is commonly used in delta feature implementations [34]). These three choices produce eight configurations that are all consistent with the given description, any of which constitutes a reasonable attempt at reconstructing the described method. There are, of course, many other parameters unstated: the exact specification of the mel filter bank, how aggregation across frequency bands was computed, and so on. For the sake of brevity, we limit the scope of this experiment to the three aforementioned choices.

Figure 2 shows the distribution of F-measure (harmonic mean of precision and recall) for each configuration. Although the best-performing versions are approximately equivalent, the range of scores is quite large, spanning 0.43 to 0.76. Moreover, some decisions can have a significant effect in some conditions (e.g., the differencing filter when using Hamming windows) that vanishes in other conditions (e.g., using a Hann window). This demonstrates that an incomplete system description can lead to incorrect conclusions about a particular design choice. The interventions performed in this experiment are confined to a single stage of Figure 1 (modeling), but realistic systems are susceptible to variation at each stage of the pipeline.

Although this method is simple enough to be completely described in a short amount of text, a full description quickly becomes impractical as methods become more complex. In modern research systems, the only practical means of fully characterizing the implementation is to provide the source code and data.

Best practices for OSS research

As described in the previous sections, modern research pipelines consist of many components with complex interactions. The engineering cost for developing and maintaining these components often exceeds that of implementing the core research method for a particular study. Sculley et al. [35] discussed this cost as hidden technical debt, which is hard to notice and compounds silently. In this section, we provide recommendations for open-source research software development, which can help improve code quality and reproducibility and foster efficient long-term collaboration on large projects with distributed contributors. The suggestions we make here are broadly applicable outside MIR [or digital signal processing (DSP)], and we draw attention to these points specifically because domain experts are often not aware of their importance. Many of the recommendations given here are also implemented concretely in Shablona (https://github.com/uwescience/shablona), a template repository for starting scientific Python projects. Interested readers may wish to browse the Shablona repository while reading the following sections. Readers entirely new to software development and OSS may additionally benefit from the instructional materials provided by Software Carpentry (https://software-carpentry.org/), the Hitchhiker's Guide to Python (https://docs.python-guide.org/), and Wilson et al. [36].

Software licensing

The defining characteristic of OSS is the license. Licenses dictate the terms under which software can be used, modified, or distributed. If no license is explicitly stated, then no use, modification, or distribution is permitted [37], and, to put it mildly, this significantly impedes adoption, reuse, and open science. Therefore, it is important to include a license agreement with any software intended for reuse and distribution.

There are many open-source licenses to choose from, but four of the most popular licenses are the Massachusetts Institute of Technology (MIT), Berkeley Software Distribution (BSD), Apache, and General Public License (GPL). MIT and BSD are simple, permissive licenses with minimal requirements on how derivative works are distributed. Apache is also permissive, but it contains additional provisions, including a grant of patent rights from contributors to users. In contrast, GPL requires derivative works to be distributed under the same license terms.

Not all of these licenses will suit an individual's or organization's needs. Therefore, it is common for particular communities to tend toward a specific type of license: the scientific Python community generally uses the more permissive MIT- or BSD-style licenses, whereas the R programming language community mostly uses GPL-style licenses. A full discussion of the relative merits of different licensing options is far beyond the scope of this article, but we recommend https://choosealicense.com as a resource to help select and compare the various options.

Documentation

Documentation is the primary source of information for users of a piece of software, and it should be written and maintained with the most relevant and helpful content. A common practice for distributing documentation is to include it with the source code distribution so that it is tightly coupled to the specific software version in use. We recommend using a documentation build tool that can automatically generate a website using both explicit documentation files and the in-line comments in the source code for the application programming interface (API). Examples of such tools include Sphinx and MkDocs, and the generated website can be hosted on services such as Read the Docs (http://readthedocs.io). In addition to describing software functionality, documentation should also include relevant bibliographic references and instructions for attribution.

To prevent the common problem of documentation falling out of sync with the software, it is important to document concurrently with programming. Similarly, before each new version of a package is released, a thorough audit of documentation should be conducted with respect to the changes introduced since the previous release. All changes should be summarized in a CHANGELOG or release notes section of the documentation, ideally with time stamps, so that users can quickly discern changes introduced for each version. These simple steps, combined with semantic versioning and version control (described in the following sections), require little effort, but they substantially ease use and integration.

Finally, we emphasize the importance of providing example code in the documentation. Although examples cannot replace a textual description of functionality, including self-contained example usage for each function or class (along with the expected output of the example code) can often be a more

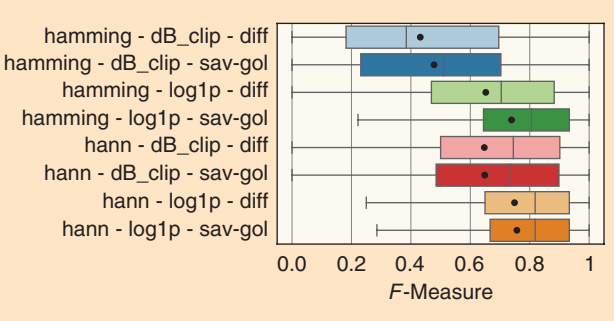


FIGURE 2. The results of the onset detection experiment: Each box corresponds to the interquartile range over test recordings, with the mean and median scores indicated by • and |, respectively. Each row corresponds to a system configuration that is consistent with the description given in the "Example: Onset Detection" section, but differs in unstated parameters.

effective way of communicating the behavior of a component to a novice user.

Version control software

Version control software (VCS) is an essential tool for modern software development that 1) keeps track of who changed what, when, and for what reason; 2) supports creating and recreating snapshots of everything in the project's history; and 3) enables a variety of tooling related to the software, such as test automation (see the "Automated Software Testing" section) and quality control (see the "Code Quality and Continuous Integration" section). Git, currently the most popular VCS in OSS development, is a distributed VCS where the full history of the project is stored in every developer's computer. GitHub (https://github.com) is a service that offers free hosting for open-source projects and leverages the decentralized nature of Git to provide a platform for collaboration of software developers. Bundled with the pull-request feature (see the "Project Management, Pull Requests, and Code Review" section) that allows users (internal and external to a project) to suggest changes; issues trackers; and provides wikis, service integration, and website hosting, GitHub serves as the home for the majority of open-source projects.

VCS is also important for managing releases, which are packaged versions of the software intended to be easily downloaded and used. Each release is marked with a version string (such as *1.5.3*), and semantic versioning (https://semver.org/) is a recommended practice of assigning software versions that can systematically inform the users about the incompatible changes to expect when updating versions. At a high level, semantic versioning states that API-compatible revisions to a package retain the same major version index, which allows users (including other libraries) to loosely specify version requirements.

Unfortunately, there are no guarantees that a commercial hosting service like GitHub will persist indefinitely. Therefore, for software accompanying publications, we recommend using a funded research data repository such as Zenodo (see the "Data Distribution" section) in conjunction with GitHub. Large research data repositories typically guarantee multiple decades of longevity.

In short, we recommend using Git for efficient collaboration and sustainable development of software, with the help of GitHub for software distribution and issue tracking. GitLab is an alternative to GitHub that also offers free hosting and issue tracking but can be locally installed and self-administered.

Automated software testing

It is beneficial for software projects to regularly perform automated tests to ensure the correctness of implementation. Automated software testing involves a set of specifications that precisely define the intended behaviors of the software, along with a testing framework that controls the execution of the tests and verifies that the software produces the expected outputs. The purpose of test automation is not only to verify that the current code works as intended but also to quickly detect any regressions caused by changes to any part of the software.

Unit testing refers to automated testing of the smallest testable parts of the software—units—which are usually individual functions or classes. Specifying the behaviors of the individual units not only helps programmers find errors in the earliest stage of development but also encourages a modular design composed of loosely coupled, testable components. Other forms of testing include integration testing, where tests are designed to ensure that small components produce desired results when combined, and regression testing, which compares current outputs to archived previous outputs so that unexpected changes (regressions) can be easily and automatically detected.

Large research data repositories typically guarantee multiple decades of longevity.

By defining the guarantees of each part of the software and writing tests that can detect deviations from the guarantees, automated testing helps improve the stability and reliability of the software and ultimately reduces the potential cost of undetected or late-detected errors. Test-driven development (TDD) is a software development process in which the specification is written before the actual development of features, and the implementations of features are then made to pass the tests [38]. Although TDD protocol is not always strictly followed, writing tests early in development can help programmers clarify the intended behavior of a function and discover components that need to be simplified into smaller units.

Code quality and continuous integration

Software developers should strive to maintain high-quality code, meaning that it is well formatted, well organized, and clear to read. Static analysis tools are utilities that quantify various dimensions of code quality without executing the software. Many programming languages include static analyzers to test that code adheres to a style (formatting and variable naming) guideline, such as Python's pycodestyle tool. Similarly, a linter is a static analysis tool that can suggest stylistic improvements to the structure of code and identify possible sources of errors. Linters can also perform a measurement of code complexity and produce warnings if, e.g., a function is too complex in its structure. A metric commonly used for measuring this is the cyclomatic complexity, which is the number of independent code paths in a unit of code.

Another important metric for code quality is test coverage, the proportion of code executed by the tests. Low code coverage implies that the software is not thoroughly tested and thus is unlikely to be reliable. Having a low cyclomatic complexity is helpful in achieving high code coverage, because it determines the number of test cases required to achieve the full code coverage.

Integration is the task of putting the development outputs to a product, i.e., ensuring quality by performing various automated tests and packaging the software for deployment. Continuous integration (CI) is a practice of performing integrations as frequently as possible by automating the process so that the status of every change to the code is automatically verified. In

addition to ensuring good software quality through automated tests, continuous integration provides a platform for automatic analysis of code quality. By using a version control system, continuous integration can be performed automatically at every registered change to the software, and services such as Travis CI (https://travis-ci.org), CircleCI (https://circleci.com), and AppVeyor (https://www.appveyor.com) provide free hosting for open-source projects.

Project management, pull requests, and code review

Although automated testing and static analysis are powerful tools, they must be used effectively to produce high-quality OSS. Ultimately, software is developed and maintained by humans, and there is no total substitute for proper project management. A widely adopted practice in OSS development is to require that all changes to a code base be submitted via pull requests. A pull request combines one or more proposed revisions to the software as a unit that can either be accepted (merged into the main repository) or rejected. The benefit of this practice is that a pull request provides a convenient point for human intervention without the need to manually track each individual change. Continuous integration systems typically execute all tests on a proposed pull request, which gives the project manager—who may be the same person as the pull-request author—a quick way to determine whether the proposed changes conform to style requirements, are sufficiently tested and documented, and do not introduce test regressions.

Typically, a pull request should not be merged if any of the following conditions are not satisfied: 1) all tests pass, 2) test coverage has not decreased, 3) the code adheres to style requirements, and 4) the proposed changes are properly documented. The first condition verifies that the proposed changes do not break existing behavior. The second condition requires that the proposed changes include a minimum amount of corresponding tests. The third condition checks that the proposed changes are stylistically consistent with the project's goals and existing code. The fourth condition ensures that the project's documentation does not fall out of sync with the source code. Of these, the first three conditions can be automated by continuous integration. However, none of the conditions ensures the correctness of the proposed change, which ultimately should be determined (as best as possible) by a thorough code review by one or more parties beyond the author of the proposed changes. Incidentally, code review is also the ideal time to check the fourth condition and request any modifications to the pull request. Adopting this workflow early in a project's life cycle can provide structure to software development and ease the burden of adhering to best practices (especially documentation and testing).

Interoperability and interface design

Publishing an OSS library means its functions and classes can be used by many users, who will benefit from a maintainable, extensible, and easy-to-understand API design. This includes programming practices such as descriptive function and variable naming, intuitive organization of functionality into submodules, and sensible default parameter values.

In addition to the importance of intuitive API design, we argue that function-oriented interfaces are often better than object-oriented designs. In research settings, use cases are often procedural executions of steps in a pipeline, and using class hierarchies may entail unnecessary cognitive load. Functions have well-defined entry and exit points, making their life spans explicit, but objects maintain state indefinitely, making it difficult to infer their scope. Moreover, classes do not easily traverse library boundaries, impeding interoperability between components. If an API expects or produces an instance of a certain class, it forces every package depending on the API to conform to the specification of the class, and this makes such packages sensitive to future changes in the class definition. For this reason, data containers can be better represented in the standardized, primitive collection types, such as dictionaries, lists, or NumPy `ndarray` type.

Despite these arguments for function-oriented design, object-oriented interfaces can be useful when the primary goal explicitly requires persistent state. This is the case, for instance, when packaging statistical models, where the state encapsulates the model parameters.

Packaging and environments

Software is often organized into packages to facilitate maintainability and distribution, and it is a responsibility of a package management system to provide means to install specific versions of desired packages. Many programming languages provide package management systems that help organize installed libraries and applications, such as `pip` for Python and CRAN for R. These provide a way to specify dependency requirements and user interfaces to install and upgrade software. Because installing a software package becomes as simple as running a single-line command—`[package-manager] install [package-name]`—it is often a good idea to distribute the software as a package for easier and wider adoption, even if the project is not primarily a library. Packages are constructed by build tools, which vary across languages, such as Python's `setuptools` or Java's `Gradle`. Working in conjunction with package management software, build tools allow a project to be packaged with its dependencies and with their exact versions specified, along with the metadata to help index the project in a repository.

Within Python, there are two dominant package systems: the Python Package Index (PyPI or pip package manager) and Conda. The key distinction between these two systems is that pip can package only Python modules (and extensions written in C), but Conda packages can be written in any language. Conda packages thus allow dependency tracking across languages, so that a package written in Python, e.g., can have dependencies written in C. This property is useful when developing large systems with heterogeneous components, as is common in MIR and likely to become common in DSP more broadly in the future.

With all dependencies and their versions specified for a project, one can ensure the interoperability between components

and thus have an environment that provides reproducible results. However, libraries are known to change over time and introduce incompatibilities across version upgrades. This can present a problem when reproducing an old experiment in a modern environment or when working on multiple projects with conflicting dependencies. Environment managers (such as Conda or virtualenv in Python) resolve this by providing isolated environments in which packages can be installed. Virtual machines or containers like Docker can also provide isolated and reproducible environments that do not depend explicitly on the programming language in question. Container tools like ReproZip [39] can significantly ease reproducibility by automatically generating virtual machine images to reproduce a specific experiment.

Project structure

Figure 3 provides our recommended repository structure for MIR projects using Python, although the template could be easily adapted to other domains and languages. The top-level directory should at least include the license and a readme.txt file that describes the project at a high level and provides contact information for the authors. The file env.yaml (or requirements.txt) describes the software dependencies (and versions) necessary to reproduce the project's working environment; these should be automatically generated by a package or environment manager, e.g., by executing `conda env export` or `pip freeze`.

The `data` subdirectory should contain any static data used in the experiment, such as a filename index of a data set or configuration files associated with various software components. Entire data sets need not be included in the repository here (to limit the size of the repository), but a script or instructions to procure the data should be provided.

The `scripts` subdirectory contains all of the scripts needed to generate the results of the project. Here, we have taken inspiration from the UNIX System V init system, which organizes (system startup) scripts alphanumerically to ensure a consistent order of execution. This simple convention eases reproducibility by eliminating any ambiguity in how the various components should be executed. The exact subdivision of steps is not critical, but the four listed here—synthesis/augmentation, preprocessing, model estimation, and evaluation—apply broadly to many situations. We have found this loose organization to be flexible and useful in our own projects.

```
project/
    LICENSE.txt
    README.txt
    env.yaml (or requirements.txt)
    data/
        index-all.json
        ...
    scripts/
        01-data-augmentation.py
        02-pre-process.py
        03-model.py
        04-evaluate.py
        ...
    generated/
        split01/
            index-train.json
            index-test.json
            model_parameters.h5
            results.json
        split02/
            ...
        notebooks/
            01-analysis.ipynb
            ...
```

FIGURE 3. An example file structure for an MIR research project.

The preprocessing step can entail a variety of processes that generate intermediate data, such as precomputed feature transformations or train-test splits of a data set. For diagnostic purposes, we specifically advocate generating train-test index partitions independent of model estimation and saving all index sets to disk as index files (e.g., `splitNN/index_train.json`). This small amount of bookkeeping can significantly ease debugging and reproducibility and can facilitate fair, paired comparisons between different methods over the same data partitions. All data produced automatically should be kept separate from the static `data` directory, e.g., in a dedicated `generated` directory; if there are multiple train-test splits, then all split-dependent data should be kept in their own subdirectory (or otherwise separated by filename) to prevent statistical contamination across partitions.

We recommend that any (interactive) post hoc analysis of the results including figure generation for publications and be stored separately under `notebooks`. Here, we suggest Jupyter notebooks (https://www.jupyter.org/), which are portable and support interactive execution in a variety of languages. If multiple steps are necessary, we again recommend ordering the files alphanumerically to disambiguate execution order.

As a final note, we suggest that all (pseudo-)randomized computations throughout the process use a fixed seed, which can be easily set by a user. This ensures that the entire system is deterministic and can significantly aid in debugging and reproducibility.

Proposal: Tools for data collection and distribution

Just as complex systems often require multiple software components, they increasingly also require multiple data sets. Similar to software, data sets can also change over time, either from extension or correction [40]. In addition, even small changes in the data collection and processing pipeline can affect results. For example, previous studies have shown that even the visualization used in audio annotation can affect annotation quality [41]. Researchers also often process or clean annotations by removing outliers or aggregating annotations. These processes must be documented to appropriately use and extend annotations. Although many open-source principles can also be applied to data, there is much work to be done regarding tooling and infrastructure to support OSS practices for data collection. This section is both a position statement and a proposal to the community in which we outline what has been done and propose what needs to be done to move forward regarding the tooling of data collection and distribution.

Data annotation

First, we propose that the research community should develop and adopt standard, open-source tools for audio annotation. This ensures not only that we are not replicating existing work with several ad hoc annotation solutions but also that we are following best practices and can extend existing data sets developed by other research groups.

In addition to following the OSS principles outlined earlier in this article, these tools should also be configurable, extensible, and web-based so that they can be easily deployed without requiring users (annotators) to install software. Web-based solutions enable easy distribution of audio and crowdsourced annotation, now a standard method for obtaining large numbers of annotations. Although many of our data needs can be met using strong or weak labeling tasks, some of our data needs require more specialized, unforeseen tasks. Therefore, these tools should be extensible, i.e., with the capability to support new tasks and workflows. Finally, the configuration of these tools—instructions, workflow definitions, task configurations, and so on—should also be stored in a single location in a human-readable format.

A number of open-source desktop applications have been already developed for annotation, such as Raven [42], Audacity [43], or Tony [44], but only recently have we seen the emergence of web-based tools for crowdsourcing. Audio Annotator is a simple web-based front end for strong labeling of audio with standard audio visualizations [41]. Although a good starting point, its functionality is limited, and it is not easily extensible. Freesound Datasets is a new web-based platform for crowdsourcing weak labels of audio, hosted on https://freesound.org [45]. However, it is currently limited to Freesound data and also is not extensible. Zooniverse is the most popular citizen science platform, with over a million registered users [46]. Zooniverse supports audio content and audio visualizations, but the available task types are limited to weak labeling and survey questions, and its extensibility is limited.

First, we propose that the research community should develop and adopt standard, open-source tools for audio annotation.

Data set file formats

As described in the "System Architecture and Component" section, standardized tools for reading and writing data file formats minimize the risk of parsing errors and ease distribution and use of data. There are several formats for encoding music annotations (MusicXML [12], MEI [13], MPEG-7 [14], and JAMS [15]), but these formats are primarily for managing annotations for a single recording rather than collections of annotated audio. To increase transparency and usability of data sets, we propose to develop a package to support collection management. Only the raw annotations and audio would be stored as data, and views could be defined to filter and process the data for a specific task. For example, if a data set needs to be cleaned to remove erroneous annotations or outliers, then users could write a clean view of the data without discarding information. Additionally, preregistered splits of the data could be implemented as a view on top of an existing view. Data set files would also contain standardized metadata and documentation of the data set creation process.

Data documentation

To understand the content of data sets, use them appropriately, and extend them when necessary, data sets must be thoroughly documented. This motivates researchers to develop standard reporting mechanisms and tools to facilitate the documentation of the data collection process. Although standards should be developed and ratified by the community, the following are possible items to include for each annotation:

- annotation software and version
- annotation software configuration
- description of all tasks, including participant screening, training, annotation tasks, and surveys
- description of annotator recruiting
- monetary (or other extrinsic) compensation mechanisms
- anonymized annotator identifiers
- time stamps
- data cleaning or processing procedures
- data synthesis procedures description and code (if applicable).

All such documentation should provide reasonable explanations and justifications for the choices made. This again helps the community understand the data and what it can be used for. As a community, we should also determine screening and demographic survey procedures and annotator quality metrics. Once these have been established, documentation tools, in combination with standardized annotation file formats, should be able to quickly aggregate and display this information about the population as a whole. Best practices for data documentation have been proposed before in MIR, although adoption by the community has been slow [47]. Recently, Gebru et al. [48] proposed a standardized data sheet format for general machine-learning data sets, inspired by the standardized data sheets that accompany electronic components.

Data distribution

Finally, we need tools to distribute, maintain, and index public data sets. Although many of the requirements for data are similar to those for software, data typically require more storage than software, rendering many existing services unsuitable. Data hosting should support versioning to support changes to data sets, provide digital object identifiers (DOIs), and guarantee longevity for several decades to prevent broken URLs and ephemeral data. These data requirements files would specify the data sets and versions required by software, and they should be distributed along with the software requirements files. There are currently several hosting solutions that support large data sets, versioning, and DOIs and guarantee decades of longevity (e.g., Zenodo at https://zenodo.org, Figshare at https://figshare.org, Dryad at https://datadryad.org, and Dataverse at https://dataverse.org). Unfortunately, these solutions have yet to develop a data management tool like we have described. However, it

may be possible for a third party to build such a tool around the existing infrastructure.

In addition to hosting and distribution, we also need a platform for developing and maintaining data. At the minimum, this would include an issue tracker for reporting errors and proposing/discussing improvements to existing data sets. However, this could also double as a platform for proposing and discussing the creation of new data sets. Although such functionality would ideally be integrated into hosting services, this could also be developed around existing infrastructure or supported with existing platforms such as GitHub.

Conclusions

Although MIR has long been data driven and necessarily complex because of the long chain of steps involved in bridging audio signals and semantically meaningful representations, we expect the core issues of system complexity to eventually pervade all data-driven areas of signal processing. The general architecture outlined in the "System Architecture and Components" section is generic enough to capture most MIR use cases, and, although different domains might exhibit slightly different workflows, we expect that the overall system complexity issue will arise across domains. The recommendations put forward in the "Best Practices for OSS Research" section should serve as a solid basis for improving the quality and reproducibility of scholarly research. While we do not expect signal processing researchers to become experts in software engineering, we focus here on software precisely because it is often overlooked as a crucial component of research systems. Although most of our recommendations concern software, we see data management as the next frontier in improving data-driven research in general and signal processing research specifically. Our proposal is intended to resolve certain shortcomings in our current practices for data set construction, but it may be readily adapted to different application domains. We encourage future researchers to think carefully about data construction, preservation, and management issues moving forward.

Authors

Brian McFee (brian.mcfee@nyu.edu) received his B.S. degree in computer science from the University of California, Santa Cruz, in 2003 and his M.S. and Ph.D. degrees in computer science and engineering from the University of California, San Diego, in 2008 and 2012, respectively. He is an assistant professor of music technology and data science at New York University. His work lies at the intersection of machine learning and audio analysis. He is an active open-source software developer and the principal maintainer of the librosa package for audio analysis.

Jong Wook Kim (jongwook@nyu.edu) received his B.S. degree in electrical engineering from the Korea Advanced Institute of Science and Technology, Daejeon, and his M.S. degree in computer science and engineering from University of Michigan in 2009 and 2011, respectively. From 2012 to 2015, he was a back-end software engineer at NCSOFT Corporation and Kakao Corporation, South Korea, and he was a research scientist intern at Pandora in 2017 and Spotify in 2018, focusing on music recommender systems and neural music synthesis. He is a Ph.D. candidate at New York University's Music and Audio Research Laboratory. His research interests include automatic music transcription and music language models.

Mark Cartwright (mark.cartwright@nyu.edu) received a B.M. degree in music technology from Northwestern University, Evanston, Illinois, in 2004. He received an M.A. degree in music science and technology in 2007 from Stanford University (CCRMA), California, and a Ph.D. degree in computer science in 2016 from Northwestern University, where his research focused on developing new interaction paradigms for audio production tools. Currently, he is a postdoctoral researcher at New York University's Music and Audio Research Laboratory. He was previously a visiting researcher at the Center for Digital Music at Queen Mary University of London and an intern at Adobe's Creative Technology Lab. His research lies at the intersection of human–computer interaction, audio signal processing, and machine learning.

Justin Salamon (justin.salamon@nyu.edu) received his B.A. degree in 2007 in computer science from the University of Cambridge, United Kingdom, and his M.Sc. and Ph.D. degrees in computer science from the Universitat Pompeu Fabra, Barcelona, Spain, in 2008 and 2013, respectively. In 2011, he was a visiting researcher at the Institut de Recherche et de Coordination Acoustique/Musique, Paris, France. In 2013, he joined New York University as a postdoctoral researcher, where he has been a senior research scientist since 2016. He is a senior research scientist at New York University's Music and Audio Research Laboratory and Center for Urban Science and Progress. His research focuses on the application of machine learning and signal processing to audio signals, with applications in machine listening, music information retrieval, bioacoustics, environmental sound analysis, and open-source software and data.

Rachel Bittner (rachelbittner@spotify.com) received her B.S. degrees in music performance and math at the University of California, Irvine, her B.M. degree in math at New York University's Courant Institute, and her Ph.D. degree in music technology in 2018 at the Music and Audio Research Lab at New York University under Dr. Juan Pablo Bello. She was a research assistant at NASA Ames Research Center, working with Durand Begault in the Advanced Controls and Displays Laboratory. She is a research scientist at Spotify in New York City. Her research interests are at the intersection of audio signal processing and machine learning, applied to musical audio. Her dissertation work applied machine learning to fundamental frequency estimation.

Juan Pablo Bello (jpbello@nyu.edu) received his B.Eng. degree in electronics in 1998 from the Universidad Simón Bolívar in Caracas, Venezuela, and in 2003 he received his Ph.D. degree in electronic engineering from Queen Mary University of London. He is a professor of music technology and computer science and engineering at New York University. His expertise is in digital signal processing, machine listening, and music information retrieval, topics that he teaches and on

which he has published more than 100 papers and articles in books, journals, and conference proceedings. He is the director of the Music and Audio Research Lab, where he leads research on music informatics. His work has been supported by public and private institutions in Venezuela, the United Kingdom, and the United States, including Frontier and CAREER Awards from the National Science Foundation and a Fulbright scholar grant for multidisciplinary studies in France. He is a Senior Member of the IEEE.

References

[1] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Trans. Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002. doi: 10.1109/TSA.2002.800560.

[2] B. L. Sturm, "Revisiting priorities: Improving MIR evaluation practices," in *Proc. 17th Int. Society for Music Information Retrieval Conf., (ISMIR),* New York, 7–11 Aug. 2016, pp. 488–494.

[3] T. Cho, R. J. Weiss, and J. P. Bello, "Exploring common variations in state of the art chord recognition systems," presented at the Sound and Music Computing Conf., 2010.

[4] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. P. W. Ellis, "mir_eval: A transparent implementation of common MIR metrics," in *Proc. 15th Int. Society for Music Information Retrieval Conf., (ISMIR),* Taipei, Taiwan, 27–31 Oct. 2014, pp. 367–372.

[5] H. Pashler and E. Wagenmakers, "Editors' introduction to the special section on replicability in psychological science: A crisis of confidence?" *Perspectives Psychological Sci.*, vol. 7, no. 6, pp. 528–530, 2012.

[6] P. Vandewalle, J. Kovacevic, and M. Vetterli, "Reproducible research in signal processing," *IEEE Signal Processing Mag.*, vol. 26, no. 3, pp. 37–47, 2009.

[7] J. B. Buckheit and D. L. Donoho, "Wavelab and reproducible research," in *Wavelets and Statistics*, A. Antoniadis, G. Oppenheim, and B. McFee, Eds. New York: Springer, 1995, pp. 55–81.

[8] M. Owens and G. Allen, *SQLite*. New York: Springer-Verlag, 2010.

[9] M. Folk, A. Cheng, and K. Yates, "HDF5: A file format and I/O library for high performance computing applications," in *Proc. Supercomputing*, vol. 99, 1999, pp. 5–33.

[10] F. Bellard, M. Niedermayer, et al. (2012). Ffmpeg. [Online]. Available: http://ffmpeg.org.

[11] E. de Castro Lopo. (2011). Libsndfile. [Online]. Available: http://www.mega-nerd.com/libsndfile/

[12] M. Good, "MusicXML for notation and analysis," *Virtual Score: Representation, Retrieval, Restoration*, vol. 12, pp. 113–124, 2001.

[13] P. Roland, "The music encoding initiative (MEI)," in *Proc. First Int. Conf. Musical Applications Using*, 2002, pp. 55–59.

[14] S.-F. Chang, T. Sikora, and A. Purl, "Overview of the MPEG-7 standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 6, pp. 688–695, 2001.

[15] E. J. Humphrey, J. Salamon, O. Nieto, J. Forsyth, R. M. Bittner, and J. P. Bello, "JAMS: A JSON annotated music specification for reproducible MIR research," in *Proc. 15th Int. Society for Music Information Retrieval Conf., (ISMIR),* Taipei, Taiwan, 27–31 Oct. 2014, pp. 591–596.

[16] B. McFee, E. J. Humphrey, and J. P. Bello, "A software framework for musical data augmentation," in *Proc. 16th Int. Society for Music Information Retrieval Conf., (ISMIR),* Málaga, Spain, 26–30 Oct. 2015, pp. 248–254.

[17] M. Mauch and S. Ewert, "The audio degradation toolbox and its application to robustness evaluation," in *Proc. 14th Int. Society for Music Information Retrieval Conf., (ISMIR),* Curitiba, Brazil, 4–8 Nov. 2013, pp. 83–88.

[18] J. Salamon, D. MacConnell, M. Cartwright, P. Li, and J. P. Bello, "Scaper: A library for soundscape synthesis and augmentation," presented at the Workshop on Applications of Signal Processing to Audio and Acoustics, New Paltz, NY, Oct. 2017.

[19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, et al. "Scikit-learn: Machine learning in Python," *J. Mach. Learning Res.*, vol. 12, pp. 2825–2830, Oct. 2011.

[20] B. Whitman, G. Flake, and S. Lawrence, "Artist detection in music with minnowmatch," in *Proc. 2001 IEEE Signal Processing Society Workshop*, 2001, pp. 559–568.

[21] B. McFee, C. Jacoby, E. J. Humphrey, and W. Pimenta. (2018). Pescadores/pescador: 2.0.0. [Online]. Available: https://doi.org/10.5281/zenodo.1165998

[22] B. Van Merriënboer, D. Bahdanau, V. Dumoulin, D. Serdyuk, D. Warde-Farley, J. Chorowski, and Y. Bengio. (2015). Blocks and fuel: Frameworks for deep learning. arXiv. [Online]. Available: https://arxiv.org/abs/1506.00619

[23] D. Bogdanov, N. Wack, E. Gómez, S. Gulati, P. Herrera, O. Mayor, G. Roma, J. Salamon, et al., "Essentia: An audio analysis library for music information retrieval," in *Proc. 14th Int. Society for Music Information Retrieval Conf., (ISMIR),* Curitiba, Brazil, 4–8 Nov. 2013, pp. 493–498.

[24] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in Python," in *Proc. 14th Python in Science Conf.*, 2015, pp. 18–25.

[25] P. Brossier. (2009). Aubio, a library for audio labelling. [Online]. Available: https://aubio.org/

[26] S. Böck, F. Korzeniowski, J. Schlüter, F. Krebs, and G. Widmer, "Madmom: A new Python audio and music signal processing library," in *Proc. 2016 ACM Multimedia Conf.*, 2016, pp. 1174–1178.

[27] G. Tzanetakis and P. Cook, "Marsyas: A framework for audio analysis," *Organised Sound*, vol. 4, no. 3, pp. 169–175, 2000. doi: 10.1017/S1355771800003071.

[28] J. Urbano, D. Bogdanov, P. Herrera, E. Gómez, and X. Serra, "What is the effect of audio quality on the robustness of MFCCs and chroma features?" in *Proc. 15th Int. Society for Music Information Retrieval Conf., (ISMIR),* Taipei, Taiwan, 27–31 Oct. 2014, pp. 573–578.

[29] F. Chollet, et al. (2015). Keras. [Online]. Available: https://keras.io

[30] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for polyphonic sound event detection," *Appl. Sci.*, vol. 6, no. 6, p. 162, 2016. doi: 10.3390/app6060162.

[31] A. Said and A. Bellogín, "Rival: A toolkit to foster reproducibility in recommender system evaluation," in *Proc. 8th ACM Conf. Recommender Systems*, 2014, pp. 371–372.

[32] S. Böck, F. Krebs, and M. Schedl, "Evaluating the online capabilities of onset detection methods," in *Proc. 13th Int. Society for Music Information Retrieval Conf.,* Mosteiro S. Bento Da Vitória, Porto, Portugal, 8–12 Oct. 2012, pp. 49–54.

[33] S. Böck, "oneset_db." Accessed on: Jan., 2018. [Online]. Available: https://github.com/CPJKU/onset_db

[34] D. P. Ellis. (2006). PLP and RASTA (and MFCC, and inversion) in MATLAB using melfcc.m and invmelfcc.m. [Online]. Available: http://www.ee.columbia.edu/\~dpwe/resources/matlab/rastamat

[35] D. Sculley, G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Chaudhary, M. Young, et al., "Hidden technical debt in machine learning systems," in *Proc. Advances in Neural Information Processing Systems*, 2015, pp. 2503–2511.

[36] G. Wilson, J. Bryan, K. Cranston, J. Kitzes, L. Nederbragt, and T. K. Teal, "Good enough practices in scientific computing," *PLoS Computational Biology*, vol. 13, no. 6, 2017. doi: 10.1371/journal.pcbi.1005510.

[37] GitHub, Inc. No license. [Online]. Available: https://choosealicense.com/no-permission/

[38] K. Beck, *Test-Driven Development: By Example*. Reading, MA: Addison-Wesley, 2003.

[39] F. S. Chirigati, D. E. Shasha, and J. Freire, "ReproZip: Using provenance to support computational reproducibility," presented at the 5th USENIX Conf. Theory and Practice of Provenance (TAPP'13), 2013.

[40] B. L. Sturm, "An analysis of the GTZAN music genre dataset," in *Proc. 2nd Int. ACM Workshop on Music Information Retrieval With User-Centered Multimodal Strategies*, 2012, pp. 7–12.

[41] M. Cartwright, A. Seals, J. Salamon, A. Williams, S. Mikloska, D. MacCbonnell, E. Law, J. Bello, and O. Nov, "Seeing sound: Investigating the effects of visualizations and complexity on crowdsourced audio annotations," *Proc. ACM on Human-Computer Interaction*, vol. 1, no. 1, 2017. doi: 10.1145/3134664.

[42] Bioacoustics Research Program. (2014). Raven pro: Interactive sound analysis software (version 1.5). [Online]. Available: http://www.birds.cornell.edu/raven

[43] D. Mazzoni and R. Dannenberg. (2000). Audacity. Avaliable: https://www.audacityteam.org

[44] M. Mauch, C. Cannam, R. Bittner, G. Fazekas, J. Salamon, J. Dai, J. Bello, and S. Dixon, "Computer-aided melody note transcription using the Tony software: Accuracy and efficiency," presented at the 1st Int. Conf. Technologies for Music Notation and Representation, 2015.

[45] E. Fonseca, J. Pons Puig, X. Favory, F. Font Corbera, D. Bogdanov, A. Ferraro, S. Oramas, A. Porter, and X. Serra, "Freesound datasets: A platform for the creation of open audio datasets," in *Proc. 18th Int. Society for Music Information Retrieval Conf. (ISMIR)*, Suzhou, China, Oct. 2017, pp. 486–493.

[46] K. Borne and Z. Team, "The Zooniverse: A framework for knowledge discovery from citizen science data," in *Proc. AGU Fall Meeting Abstracts*, 2011.

[47] G. Peeters and K. Fort, "Towards a (better) definition of the description of annotated MIR corpora," in *Proc. 13th Int. Society for Music Information Retrieval Conf., (ISMIR),* Porto, Portugal, 8–12 Oct. 2012, pp. 25–30.

[48] T. Gebru, J. Morgenstern, B. Vecchione, J. W. Vaughan, H. Wallach, H. Daumeé III, and K. Crawford. (2018). Datasheets for datasets. arXiv. [Online]. Available: https://arxiv.org/abs/1803.09010

SP

Ervin Sejdić, Georgia A. Malandraki, and James L. Coyle

Computational Deglutition

Using signal- and image-processing methods to understand swallowing and associated disorders

Swallowing is a sensorimotor activity by which food, liquids, and saliva pass from the oral cavity to the stomach. It is considered one of the most complex sensorimotor functions because of the high level of coordination needed to accomplish the swallowing task over a very short period of 1–2 s and the multiple subsystems it involves. *Dysphagia* (i.e., swallowing difficulties) refers to any swallowing disorder and is commonly caused by a variety of neurological conditions (e.g., stroke, cerebral palsy, Parkinson disease), head and neck cancer and its treatment, genetic syndromes, and iatrogenic conditions or trauma. The signs and symptoms of dysphagia range from anterior loss of food while eating, difficulty chewing, and subjective difficulty swallowing food or liquids to choking or coughing before, during, or after eating because of impaired clearance of swallowed material from the throat into the digestive system. When not effectively treated, dysphagia can cause malnutrition, dehydration, immune system failure, psychosocial degradation, and generally decreased quality of life.

The major medical consequence of dysphagia is aspiration of food and liquids into the airway, which often leads to airway obstruction, pneumonia, and increased risk of mortality. Dysphagia affects approximately 9 million adults per year in the United States [1] and is especially prevalent among the elderly. Characteristically, 50–75% of stroke patients and 60–70% of patients who undergo radiation therapy for head and neck cancer have dysphagia. In addition, each year, more than 60,000 people die of complications associated with swallowing dysfunction. Complications of dysphagia drastically increase health-care costs. Overall, together with the costs incurred by hospitals, costs of dysphagia in the health-care system exceed US$1 billion per year.

In the past 30–40 years, we have gained increased understanding of this potentially devastating condition and have made remarkable improvements in the management of dysphagia. Given recent advances in signal- and image-processing algorithms, we believe that the signal- and image-processing community is poised to make further fundamental contributions to the understanding of swallowing and swallowing difficulties and improve patient outcomes. There is a widespread need for signal- and image-processing algorithms that can help clinicians manage dysphagia. Therefore, we propose the establishment of a new signal- and image-processing subfield called *computational deglutition*. This newly established translational subfield will be a collaboration between clinicians and the signal- and image-processing community aimed at developing clinically relevant algorithms that will help clinicians assess and treat swallowing disorders.

When not effectively treated, dysphagia can cause malnutrition, dehydration, immune system failure, psychosocial degradation, and generally decreased quality of life.

Swallowing function and the swallowing mechanism

Oropharyngeal swallowing is not simply the act of propelling food and liquids toward the digestive system. It is an intricately timed, short-duration, centrally programmed patterned response designed to deliver nutrients, fluids, and medications to the digestive system and, at the same time, prevent aspiration of swallowed material into the airway. Within a few seconds, up to two dozen kinematic and valvular events, performed by more than 30 pairs of muscles, occur to simultaneously enable the upper aerodigestive tract to alternate between its respiratory and digestive functions.

Whether swallowing reflexively during sleep or consciously when enjoying a meal, swallowing delivers saliva and ingested nutrients through the pharynx, which is a single tube shared by both the respiratory (airway) and digestive systems, while valving gas (breathing) or food (swallowing) flow between them (see Figure 1 for relevant anatomical

Digital Object Identifier 10.1109/MSP.2018.2875863
Date of publication: 24 December 2018

landmarks). During oral preparation, sensory receptors in the oral mucosa receive and carry sensory information through afferent pathways to the brain stem and the brain. During this stage, liquids are contained by oral valves (lips, tongue, and soft palate), solid foods are mechanically reduced by mastication into a relatively cohesive bolus, and saliva is mixed with the bolus. When the bolus is considered adequately prepared for transfer to the pharynx, it is propelled posteriorly, while a cascading sequence of events that direct flow away from the airway and toward the esophagus begins. These events include pharyngeal and laryngeal kinematic events that mediate the opening and closing of respiratory and digestive valves and reconfigure the oropharyngeal cavity, closing the airway and opening the inlet to the esophagus. Hyolaryngeal excursion, a kinematic pattern analogous to a series of pulleys between the mandible and skull base on one end and the hyolaryngeal complex on the other, leads to this alternating valving by displacing the larynx anteriorly and superiorly out of the path of the oncoming bolus and closing its inlet valve, while simultaneously contributing to distension of the upper esophageal sphincter (UES).

Laryngeal displacement and airway closure are accompanied by inversion of the epiglottis, the cartilaginous valve at the laryngeal inlet, and closure of the internal larynx, further ensuring protection of the airway. Because the posterior wall of the larynx is shared as the anterior wall of the UES, this upward and forward displacement delivers concurrent traction forces to the UES. This traction is a necessary factor contributing to opening of the digestive valve while progressive pharyngeal pressures continue to propel the bolus into the esophagus. Figure 1 shows a description of the swallowing process.

A brief introduction to dysphagia and its common causes

More than 700,000 new cases of dysphagia are reported every year in the United States, with neural damage or impairment (e.g., stroke) as the most common cause [2], [3]. Usually, swallowing disorders after a stroke are related to disruption of the sensorimotor functions mediated by the cranial nerves, which directly control the structures of the mouth and throat [4]. Naturally, if some of the 30 pairs of oral, pharyngeal, and laryngeal muscles are not receiving proper neural inputs, the patient will not have a fully functional swallow and may be unable to adequately transfer a bolus past the pharynx and into the esophagus.

Neurodegenerative conditions, such as Huntington or Parkinson disease, also often result in swallowing disorders. In these cases, dysphagia typically manifests as oral and pharyngeal discoordination, rigidity, and/or reduced sensation in the oropharynx, all of which can result in mistiming of airway closure and upper esophageal sphincter opening. In addition to neurogenic causes, there are several anatomically related causes of dysphagia as well. Conditions that result in an inflamed and swollen esophagus, such as eosinophilic esophagitis or gastroesophageal reflux, can make it difficult for the patient to transfer a bolus through the esophagus. This can often lead to the feeling of food becoming stuck in the throat. Various abnormal benign or malignant growths, such as tumors, swollen lymph nodes, or esophageal webs, can obstruct the path of a bolus as well, leading to similar

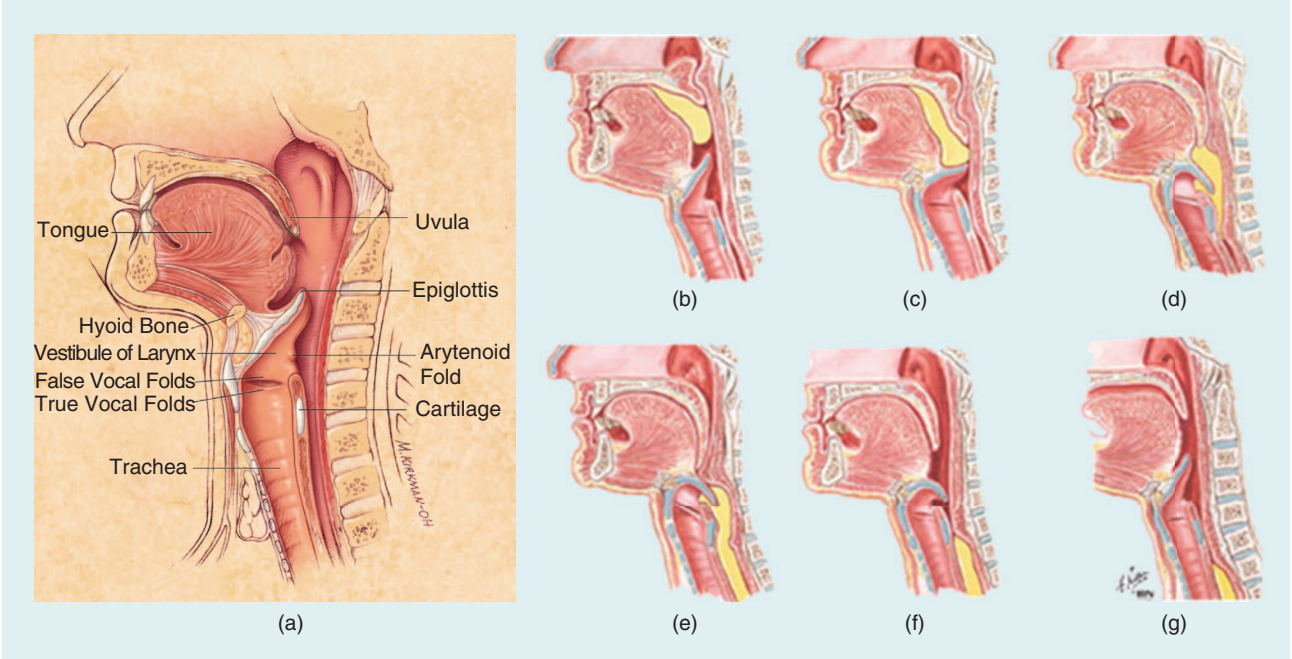


FIGURE 1. (a) Anatomical landmarks in the sagittal view [20] ©2000 KO Studios. The swallowing process: (b) the swallow initiation; (c) the bolus is propelled by tongue, and the upper esophageal sphincter opens, anticipating bolus arrival; (d) the bolus enters the pharynx associated with epiglottal downward tilt, hyolaryngeal excursion, and upper esophageal sphincter opening; (e) the bolus passes through the pharynx; (f) the bolus passes the upper esophageal sphincter, and the oropharyngeal swallow is completed; and (g) the entire bolus is in the esophagus [21].

feelings of food obstruction and risks of aspiration.

Finally, direct damage to the muscles and structures of the throat can also lead to swallowing difficulties. Surgical procedures or radiation therapy typically used to manage head and neck cancer can cause disrupted propulsion and airway protection, and other sources of physical trauma can lead to dysphagia by altering anatomy and sensorimotor integrity.

Swallowing and dysphagia assessment

Swallowing assessment can be sorted into two categories: screening and diagnostic testing. Screening tests are relatively simple pass/fail procedures performed by anyone trained in their administration; they identify patients with a high likelihood of dysphagia who need further testing. Like screenings for breast cancer or heart disease, dysphagia screening provides no diagnostic information regarding the physiological nature of the disorder nor does it provide information to guide treatment. Screenings typically include simple water-swallowing challenges in which patients fail if they cough or produce other overt signs of aspiration of the swallowed water [5]. If these signs are not present, it is assumed that dysphagia is absent, and no intervention or further testing is performed.

Conversely, diagnostic testing identifies the physiological nature of the disorder and informs the examiner, typically a qualified speech–language pathologist (SLP), about treatment options to mitigate dysphagia and its adverse effects [6]. After a failed screening result, a clinical/bedside evaluation is performed without the use of instrumentation. It involves a detailed examination of oropharyngeal and laryngeal sensorimotor function, assessment of cognitive status, and observations of the patient swallowing a variety of textures and volumes of foods and liquids. The examiner synthesizes the results and determines whether the cause of dysphagia can be determined and remediated and, in some cases, clinical evaluations are adequate to achieve these goals. However, because pharyngeal disorders and impaired airway protection are not observable without imaging technology, the clinical evaluation cannot detect asymptomatic impairments, such as silent aspiration, or any pharyngeal events that occur beyond the intraoral view of the examiner. In such cases, instrumental testing is performed.

Instrumental diagnostic tests characterize the physiological nature of the dysphagia and identify potential interventions to mitigate its adverse effects by elucidating the exact mechanisms of dysphagia along with its underlying causes. However, these tests are more complex, costly, invasive, and time-consuming than a clinical examination, and they require expert clinicians and imaging instrumentation, such as fluoroscopic or fiber optic equipment. Overall, the need for more highly trained personnel increases as the diagnostic process flows from screening to clinical to instrumental testing, respectively.

Nonimage-based information or signals about some components of the swallowing function can be collected with noninvasive methods, such as surface electromyography (sEMG) and cervical auscultation (CA). sEMG involves placing electrodes on the patient's anterior neck and recording the electrical activity of the underlying muscles during a swallow [7]. The theory is that, if the nerves or muscles involved in swallowing are affected, the signal will change in a clinically significant way compared with a recording from a healthy patient or a healthy/normal swallow. sEMG can only indirectly describe a swallow because it is limited to monitoring regional muscle activation and does not allow for isolated muscles or other regions to be assessed. As a result, this technique remains mostly experimental and complementary to other diagnostic methods; it can also be used as a treatment biofeedback tool.

CA is another popular screening method in which a clinician listens to the throat with a stethoscope while the patient swallows. The clinician then makes inferences regarding swallow integrity. The theory behind this, much like the sEMG procedure, is that the sounds recorded from a patient with dysphagia will be significantly different than those recorded from a healthy individual. However, stethoscopes and the human auditory system are incapable of transmitting or perceiving the entire spectrum of signals produced during swallowing; therefore, interpretation of these signals is imprecise and incomplete. Although attractive in its simplicity, CA is unable to identify specific physiological events or abnormalities. Currently, high-resolution sensors (i.e., piezoelectric sensors, microphones, and accelerometers) are being investigated as ways to advance CA by recording the entire spectrum of displacement, acoustic, and vibratory signals emanating from the throat during swallowing.

The most widely accepted imaging method of assessing dysphagia is the videofluoroscopic (VFS) diagnostic examination [Figure 2(a)–(c)] [8], [9]. During this test, the patient is asked to swallow small amounts of food or liquid mixed with a contrast agent, typically barium sulfate. The X-ray equipment is aligned to produce a sagittal view of the oropharynx, pharynx, and upper esophagus containing all of the major swallowing structures, allowing an imaging clinician (i.e., radiologist) and a swallowing specialist (i.e., SLP) to observe and analyze the physiological events that produce bolus movement in real time, determine which aspects of the swallow are not functioning properly, assess the timing and severity of impaired airway protection, and then deploy trial interventions. All of these factors are necessary to form a comprehensive assessment of a patient's swallow and have led to the widespread adoption of VFS as a diagnostic test.

Fiber optic endoscopic evaluation of swallowing (FEES) is also used to assess swallowing disorders [Figure 2(d)–(f)]. Rather than using an X-ray imaging machine, a small fiber optic camera attached to a flexible endoscope is directed into the oropharynx and beyond through the naris while the clinician observes events occurring before and after the swallow. The advantages of this method are that the examiner can directly observe the patient's anatomy without X-ray imaging and examine much finer details and the color of surrounding tissues and symmetry of laryngeal function, both of which can provide important

diagnostic information while the patient swallows regular foods (not barium). Because FEES tests are performed without radiation risks, they can last longer than VFS to assess issues like fatigue. However, this method has two key drawbacks relative to VFS. The first is that only a small range of oropharyngeal anatomy is visible at one time because of a limited field of view. Second, FEES techniques cannot view the entire swallowing mechanism before, during, and after the swallow, leading to imaging blindness during the pharyngeal swallow as the pharynx collapses over the camera lens. As a result, FEES cannot meaningfully assess the actions of the pharynx or larynx during a swallow and is blind to oral and esophageal structure and function, further limiting the information provided.

Advanced neuroimaging techniques, such as functional magnetic resonance imaging (fMRI), positron emission tomography, magnetoencephalography, and electroencephalography (EEG), are also instrumental methods that provide significant insights into brain activity during swallowing. These methods are mostly used experimentally at this time, but they are critical to understanding and improving computational deglutition. The following sections discuss signal- and image-processing approaches and challenges relating to the aforementioned instrumental swallowing assessments.

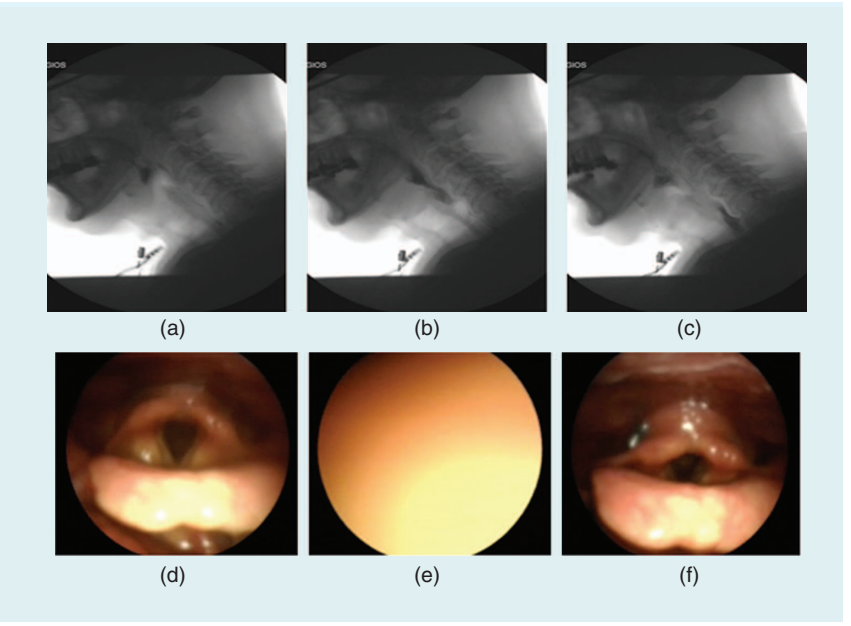


FIGURE 2. (a)–(c) A patient passing a bolus through the oropharyngeal area as seen on videofluoroscopy. (d)–(f) In this endoscopic video sequence, a patient is passing a bolus through the oropharyngeal area. Although structures can be easily viewed, the material swallowed is not as visible.

Signal processing approaches and challenges

There are many signal processing challenges in computational deglutition. Here, we focus on two prevalent cases that rely on physiological signals occurring during swallowing. In the first part, we rely on signals acquired from the neck (e.g., electromyographic, acoustic), and in the second part, we focus on EEG signals (and concurrently acquired deglutition signals from the neck) during swallowing.

A typical data acquisition and processing setup of deglutition signals acquired from the neck are shown in Figure 3. The first step is a choice of a sensor: accelerometers and microphones are typically used in most contributions [10]. Most recent contributions have shown that a combination of multiple sensors may be the most beneficial to obtain a comprehensive, noninvasive assessment of the events occurring during swallowing. However, a major issue is that there is no consensus on sensors to be used for data acquisition, and sensors of varying frequencies and bandwidths have been used in different contributions. In general, it is recommended that sensors with a flat frequency response from 0 Hz to 3 kHz be used and that these signals be sampled at 4 kHz. This is a sufficiently high sampling frequency given that most of the frequency content of these signals is below 500–600 Hz. Finally, when multiple sensors are used to acquire deglutition signals, it is strongly recommended that these signals be time synchronized in hardware via a data-acquisition board/card, i.e., the same data-acquisition system is used to synchronously acquire multiple signals. Otherwise, important swallowing events that are very short (<100 ms) may be misaligned and difficult to compare across different modalities.

The choice of a sensor affects most of the subsequent signal processing steps. Raw EMG and acoustic signals during swallowing provide potentially valuable information but in a practically useless form because raw signals cannot be quantitatively compared among patients or across sessions. Therefore, pre- and postprocessing steps are essential. The next typical task is preprocessing of deglutition signals. Here, we typically use various filtering techniques and/or denoising to remove background/electrical noise but also to annul the effects of a data acquisition apparatus (i.e., whitening). Filters are developed based on sensors and amplifiers used for data acquisition. Denoising is typically achieved via wavelet denoising. Upon completion of filtering and denoising operations or other required steps, such as normalization, swallowing recordings are segmented into multiple region of interests, typically individual swallows. Several different algorithms have been proposed in the literature, most relying on some form of machine learning. Nevertheless, the exact steps of the preprocessing tasks differ significantly in published contributions, and this often poses a challenge when trying to recreate previously obtained results. Hence, the entire field would benefit from a systematic approach to preprocessing of physiological recordings obtained from the neck during swallowing.

The third step involves feature extraction from deglutition recordings. Most contributions relied on extracting

mathematical features in time, frequency, or time-frequency domains. Although this approach was warranted in initial contributions, because there was a lack of knowledge about basic properties of deglutition signals, we strongly believe that the field should move toward the extraction of physiologically relevant features, i.e., signal features that can be related to physiological events occurring during swallowing. Hence, we need to acquire simultaneous deglutition recordings during VFS or endoscopy imaging to relate these signals and features to actual swallowing physiological events. Newer methods based on deep learning may be useful for feature extraction because these new methods can automatically extract features that maximize class differentiation.

The last step is typically a decision-making process during which we infer the integrity of the swallowing function or swallowing tasks that were carried out during an experimental procedure. In many instances, this decision process relies on a statistical analysis of features extracted in the previous step. In recent years, various machine-learning algorithms have been developed to aid the decision process. These machine-learning approaches mostly rely on differentiating swallowing safety/efficiency states. Nevertheless, there is a wide-open field for the development of machine-learning algorithms that can not only infer the state of the swallowing function but even make inferences about swallowed food or drinks. Our recent review article [10] showed that most machine-learning methods have already been used, from traditional Bayesian methods to neural networks.

Similar processing steps are taken when inferring the brain activity via EEG during swallowing [11]. After acquiring EEG signals, one would typically start with preprocessing steps that include low-pass filtering with a cutoff frequency up to 128 Hz, a notch filter at 50/60 Hz, and an artifact removal step involving the independent component analysis or other blind source separation algorithms. The preprocessing can also include a segmentation step, where one identifies regions of interest (i.e., EEG activity during swallowing) for further analysis. The segmentation process can be aided by auxiliary signals, such as CA recordings. However, to use CA recordings during the EEG segmentation process, EEG recordings need to be synchronized with these CA recordings, and this is typically achieved via a hardware system, such as a data-acquisition card.

Hence, the entire field would benefit from a systematic approach to preprocessing of physiological recordings obtained from the neck during swallowing.

The next step diverges depending on the analysis used. Researchers can use a feature-based analysis, in which they attempt to extract mathematical features they think are relevant for the decision process. These features are extracted from EEG recordings and often have no physiological meaning. A different approach relies on network-based analysis, in which researchers rely on graph theory to establish brain networks. Here, these networks during swallowing can be established in two different ways: 1) during a swallowing process that includes multiple single swallows or 2) on a swallow-by-swallow basis. The first approach is suitable when one desires to understand a global swallowing network, whereas the second approach is more suitable for understanding time-dependent changes in swallowing networks, which may be particularly of interest when clinicians are attempting to

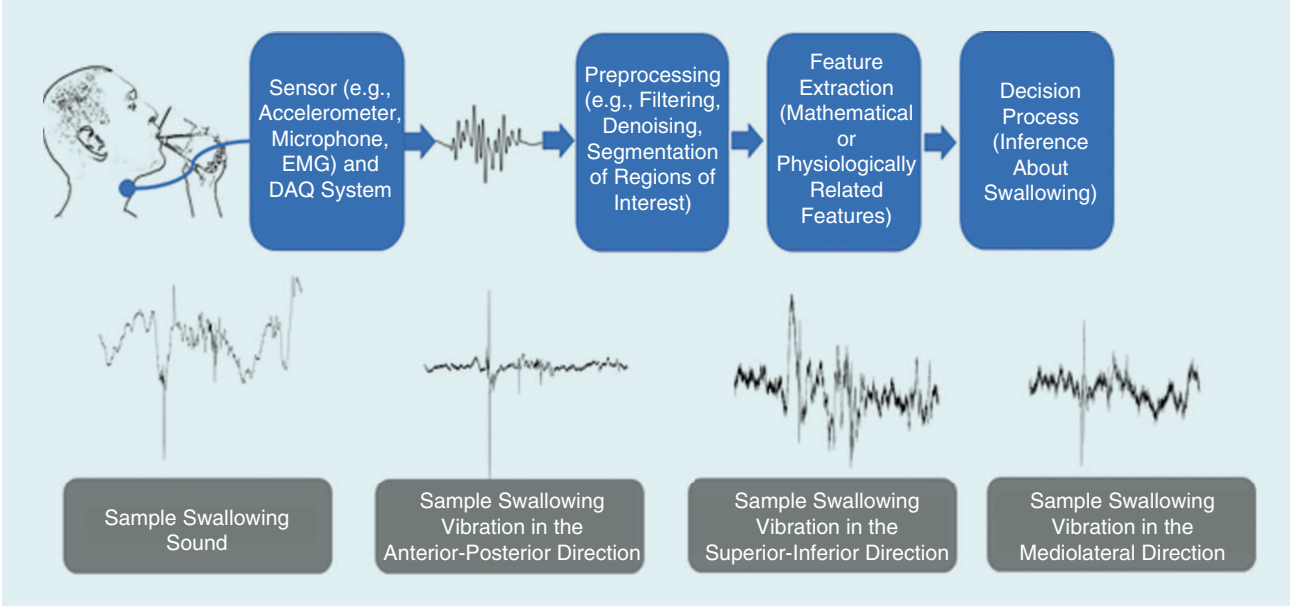


FIGURE 3. An overview of a typical setup to acquire and process deglutition signals from the neck. Sample swallowing sounds and vibration signals in the three anatomical directions are shown as well. DAQ: data acquisition; EMG: electromyography.

understand the effects of various treatments on swallowing safety and efficiency. The network-based approach is also very interesting to the signal processing community because it opens many interesting problems for the field of graph signal processing. In our own research, we use the vertex-frequency analysis (see a recent lecture note in [12]) to understand swallow-by-swallow changes in brain networks [13]. However, we anticipate that other graph signal processing approaches will be suitable as well.

The last step involves various machine-learning techniques, from traditional Bayes classifiers and support vector machines to the newest algorithms, such as deep belief networks [14]. Although various accuracies have been reported, we believe that most of those results are not generalizable for clinical use. In many cases, these contributions are proposed by signal processing practitioners with little or no understanding of clinical needs. Hence, such contributions are technologically elegant but have little clinical value. Therefore, our signal processing community needs to work more closely with clinicians to propose clinically relevant technological solutions.

Image processing approaches and challenges

Computational deglutition introduces several image processing challenges that are associated either with VFS/endoscopy or dynamic MRI and neuroimaging (e.g., fMRI) during swallowing. In this section, we briefly review some of these open challenges.

Image processing approaches have been historically constrained to human judgment. There is no dispute regarding the accuracy of human judgments of swallow kinematics, airway protection, residue patterns, and swallow efficiency. However, too few clinicians receive advanced training in the performance of these judgments, and even then, reliability ratings can vary. Efforts to standardize clinical decision making have succeeded in increasing access of validated decision-making algorithms to clinicians. A penetration–aspiration scale was developed in 1996 to describe the extent of airway compromise during disordered swallowing on a swallow-by-swallow basis and has high reliability among trained judges [15]. Residue rating scales have also been developed that use relatively convenient anatomical landmarks with which to make judgments. The Modified Barium Swallow Impairment Profile (MBSImP) was recently developed to characterize 17 components of oropharyngeal swallowing on a swallow-by-swallow basis and has acceptable interrater reliability after training [16]. Other judgments that characterize motor integrity, such as the displacement of the hyoid bone during swallowing, are commonly made in clinical imaging studies, as are inferences regarding the summative motor functions producing airway closure and UES opening. However, these judgments are largely subjective and variable, unfortunately, because the evidence indicating the range of typical displacement requires computerized analysis to characterize normal from abnormal.

Efforts to standardize clinical decision making have succeeded in increasing access of validated decision-making algorithms to clinicians.

Efforts to automate certain judgments from VFS data are expanding. Currently, residue ratings are possible using a combined human–computer interface that exploits geometric relationships to quantify post swallow pharyngeal residue (e.g., [17]). Likewise, hyoid bone-tracking methods, in which machine learning is combined with expert human judgment to measure and quantify the completeness of hyoid displacement, are under investigation in many laboratories, including our own.

There is a widespread need for algorithms that can help clinicians analyze VFS/endoscopy images. Currently, such images [Figure 2(a)–(c)] are analyzed manually on a frame-by-frame basis. Such an approach is time-consuming and prone to errors due to fatigue or lack of clinician expertise. The field currently needs algorithms to 1) segment individual physiological landmarks (e.g., cervical vertebrae) or any transient objects (e.g., a bolus) from other objects present in images, 2) identify the beginning and end of swallows, and 3) identify swallowing safety and efficiency.

In recent years, several researchers, including our team, have also used MRI techniques to investigate swallowing function and neural activity during swallowing. These techniques offer substantial benefits in image quality, but they come with their own challenges. Dynamic MRI of swallowing allows for better visualization of soft tissues than VFS and can even provide insights on muscle integrity. Another advantage of dynamic MRI is that it does not require use of ionizing radiation, so regular food can be evaluated during swallows. Imaging speed used to be superior with VFS (30 frames/s), but recently dynamic MRI has achieved serial imaging rates of up to 26 frames/s or more, providing increased temporal resolution. This is particularly critical for swallowing events, because most are completed in under a second. Data acquisition challenges that remain include the need to swallow in a supine position (most facilities do not have an upright magnet) and the magnetic susceptibility differences that occur at interfaces between air and tissue, which are plentiful in the oropharynx. These artifacts can be successfully addressed by using either multiple-shot acquisition sequences or susceptibility corrective reconstruction algorithms during post processing [18].

Unlike scales and tools designed for VFS analysis (e.g., penetration-aspiration scale, MBSImP), no similar standardized or validated tools exist to enable clinicians to complete respective measurements of swallowing events using MRI. To initiate MRI analysis, accurate registration and segmentation of swallow events and/or anatomical structures is a critical first step, because the number of volumes and slices acquired is large, and the amount of anatomical displacements during swallows is abundant. Recently, algorithms have been developed that allow semiautomatic segmentations of MRI

volumes of the tongue and hyolaryngeal structures and enable faster calculations of displacement/deformation events. In these approaches, experimenters typically identify anatomical landmarks and calculate their movement and shape changes during swallowing using advanced statistical methods. Despite their promise, such techniques continue to be validated, and they require substantial training and time to be completed. Therefore, at this time their clinical use is significantly limited.

Another popular MRI method used in swallowing research includes task-related fMRI (task fMRI), which allows us to noninvasively examine brain activations during swallows and has provided important insights into the neurophysiology of human swallowing. Image processing of fMRI data involves sophisticated pre- and postprocessing steps. After fMRI images have been acquired, preprocessing steps typically include brain extraction, removal of the first volumes that correspond to the stabilization period of the magnetic signal, despiking, slice-timing correction, motion correction, spatial smoothing, and bandpass filtering (see Figure 4). Multistage registration and normalization are also performed to register the data on standard anatomical atlases. Currently these steps are automated or semiautomated and performed via a pipeline of commands or graphical user interface systems provided in well-developed fMRI analysis programs, such as the Analysis of Functional NeuroImages or FMRIB Software Library.

To compute the task-onset timings used in the postprocessing analysis, tasks performed during the fMRI scans are often cued by visual or audio stimuli. The subject must perform the task in strict compliance with the stimulus. Secondary-monitoring devices, including surface electrodes or pneumographic belts placed around the neck, are needed to ensure the subjects' swallows comply with the stimuli [18]. During

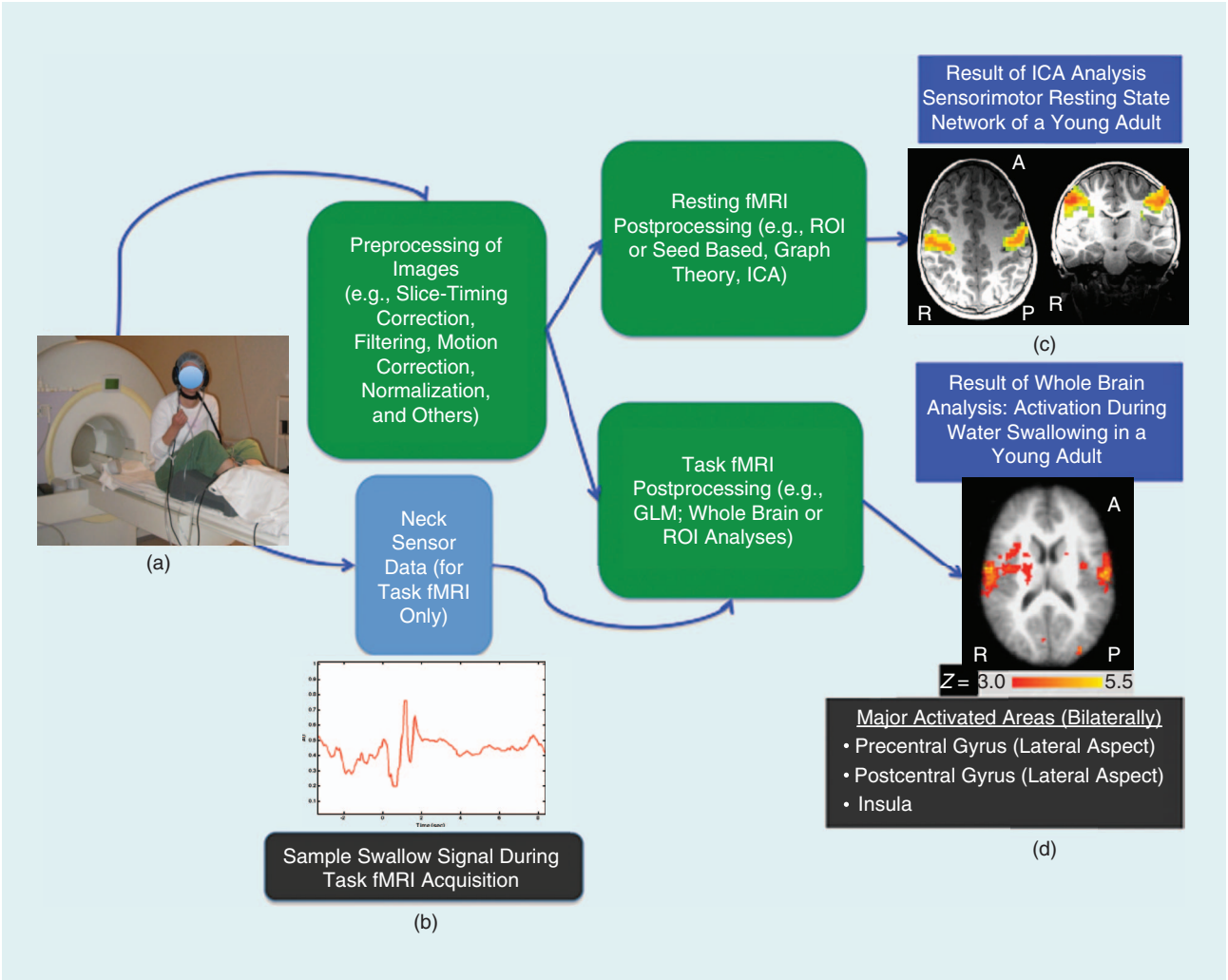


FIGURE 4. The setup to acquire and process fMRI images during the resting state and during swallowing. (a) The subject is shown wearing respiratory bellows around the neck over the thyroid cartilage (to capture swallow signal during swallows) before experiment initiation. (b) The time course of the output of the bellows for a water-swallowing trial for a single subject. (c) Results of ICA analysis of a resting-state fMRI scan of a young adult male showing the symmetrically activated sensorimotor network at rest. (d) Results of whole-brain task-related fMRI analysis of a young adult male showing areas of significant activation during water swallowing. The neurological images are shown in the radiological convention (i.e., the right hemisphere is shown on the left). A: anterior; ICA: independent component analysis; P: posterior; R: right hemisphere.

postprocessing, the task onsets are convolved with the canonical hemodynamic response function for use in general linear models (GLMs) to analyze each subject's activation during the scan. Contrasts between GLM parameters of interest are then used to compare activations among different tasks (when more than one task is examined). Multiple-comparisons corrections are further necessary because the large number of brain voxels significantly increases the false positives for any given statistical threshold. Whole-brain and region-of-interest or seed-based analyses are both widely used. For an example of a single-subject whole-brain analysis of swallowing brain activations, see Figure 4(d).

To improve signal interpretation accuracy, task fMRI results should be interpreted relative to another comparison condition (e.g., rest). Furthermore, during swallowing-specific experiments, motion-related artifacts are very common, because swallowing includes movements of the neck/throat during scanning, and these need to be carefully examined, eliminated, or postprocessed [19].

An alternative to task fMRI paradigms is resting-state functional connectivity MRI (resting-state fcMRI). Resting-state fcMRI allows us to investigate the functional connections of brain areas at rest and correlate that information with behavioral measures obtained outside the scanner. It is based on the fact that areas of the brain that are functionally related (even if they are far apart) show low-frequency fluctuations of the blood oxygenation-level-dependent signal that have the same temporal patterns. As such, resting-state fcMRI has helped us identify several resting-state networks in the brain that are altered or even absent in individuals with diseases or in older-aged compared with healthy young adults. The advantage of resting-state paradigms for studying populations with dysphagia is that patients are not required to swallow in the magnet (i.e., in the supine position), a task that is frequently challenging for patients with dysphagia. Preprocessing steps are almost identical to the task-based preprocessing analysis with the addition of nuisance factors regression (cerebrospinal fluid and white matter regressors) to further improve data quality. For postprocessing of resting-state scans, popular methodologies include advanced mathematical models, such as graph theory, independent component analyses techniques, and clustering algorithms. The contribution of resting-state fcMRI to our understanding of swallowing neural control can only be indirect and remains experimental at this time, but it holds promise.

A new imaging technology to comprehensively image swallowing physiology and neurophysiology and alleviate some of the challenges of task fMRI was examined by one of the authors and her collaborators [18]. This technology, known as *SimulScan*, allows the simultaneous dynamic imaging of the oropharyngeal area and functional imaging of the brain during swallowing and provides the ability, for the first time, to directly and simultaneously evaluate both central (brain) and peripheral (oropharyngeal) physiological signals during swallowing. This technique has been used successfully to image natural, uncued, spontaneous swallows and the brain activation associated with those swallows in healthy young adults [18], but it requires further validation.

Over the years, we have often witnessed the signal- and image-processing community developing computationally or mathematically elegant solutions with limited practical usability.

Although sophisticated algorithms are now available for the analysis of fMRI images, extensive training and expertise with this methodology are necessary, and costs remain prohibitive for clinical use. Therefore, its direct clinical application at this time is questionable, although its contribution to our understanding of the neural control of swallowing and the neuroplastic adaptations needed for functional swallowing is substantial and will continue to increase.

For dynamic MRI, which in time may be able to replace VFS, significantly more synchronous work from the image processing and clinical communities is needed. For this method as well, the field needs algorithms and models to improve segmentation and automated analysis of events, help predict swallowing pathologies, and ultimately treatment outcomes.

Future directions in computational deglutition

To foster the development of computational deglutition as a field, we encourage researchers to share data sets with other researchers. Specifically, we invite and encourage the community to produce clinical protocols and consent forms that include a clause about publicly sharing deidentified data sets to foster the growth of computational deglutition as a field. Furthermore, we anticipate that such publicly available data sets will also result in faster standardization of instrumentation and development of algorithms that can improve health care delivery and patient outcomes.

Over the years, we have often witnessed the signal- and image-processing community developing computationally or mathematically elegant solutions with limited practical usability. Computational researchers interested in this new field should work closely with clinicians to ensure that new developments are addressing clinically relevant problems. We, as the community, should also strive to ensure that our new algorithms are applicable across different patients and patient groups, rather than in a limited number of patients. Similarly, the clinical community should work closely with computational researchers to understand how to acquire data in a systematic view to ensure that the collected data are useful for further algorithmic developments.

Acknowledgments

Research reported in this publication was supported by the Eunice Kennedy Shriver National Institute of Child Health and Human Development of the National Institutes of Health under award R01HD092239. The content is

solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

Authors

Ervin Sejdić (esejdic@ieee.org) is an associate professor at the University of Pittsburgh, Pennsylvania. He received his B.E.Sc. and Ph.D. degrees in electrical engineering from the University of Western Ontario, London, Canada, in 2002 and 2008, respectively. His research interests include biomedical signal processing, swallowing, and gait. He received the Presidential Early Career Award for Scientists and Engineers in 2016 and the National Science Foundation CAREER Award in 2017. He is a Senior Member of the IEEE.

Georgia A. Malandraki (malandraki@purdue.edu) is an associate professor at Purdue University, West Lafayette, Indiana, and a board-certified specialist in swallowing disorders. She received the Early Career Contributions in Research Award by the American Speech–Language-Hearing Association in 2011. She received her B.S. degree in speech and language therapy in 2001 from the Technological Educational Institute of Patras, Greece, her M.A. degree in communication sciences and disorders in 2004 from Ohio University, Athens, and her Ph.D. degree in speech, language, and hearing science in 2008 from the University of Illinois at Urbana–Champaign. She held a postdoctoral position in dysphagia and neuroscience from 2008 to 2010 at the Department of Medicine, University of Wisconsin, Madison. Her research focuses on neuroimaging, neurorehabilitation of swallowing, and telehealth.

James L. Coyle (jcoyle@pitt.edu) is a professor at the University of Pittsburgh, Pennsylvania, and a board-certified specialist in swallowing disorders. He received his B.A. degree in health and safety studies in 1978 from the California State University, Los Angeles, his M.A. degree in communicative disorders in 1988 from California State University, Northridge, and his Ph.D. degree in rehabilitation science in 2008 from the University of Pittsburgh, Pennsylvania. He is a Fellow of the American Speech–Language-Hearing Association and received the University of Pittsburgh Chancellor's Distinguished Teaching Award in 2016.

References

[1] N. Bhattacharyya, "The prevalence of dysphagia among adults in the United States," *Otolaryngology Head Neck Surgery*, vol. 151, no. 5, pp. 765–769, 2014. doi: 10.1177/0194599814549156.

[2] G. D. Eslick and N. J. Talley, "Dysphagia: Epidemiology, risk factors and impact on quality of life–a population-based study." *Aliment. Pharmacol. Ther.*, vol. 27, no. 10, 971–979, 2008.

[3] D. L. Doggett, K. A. Tappe, M. D. Mitchell, R. Chapell, V. Coates, and C. M. Turkelson, "Prevention of pneumonia in elderly stroke patients by systematic diagnosis and treatment of dysphagia: An evidence-based comprehensive analysis of the literature," *Dysphagia*, vol. 16, no. 4, 279–295, 2001.

[4] J. Logemann, "The evaluation and treatment of swallowing disorders," *Otolaryngology Head Neck Surgery*, vol. 6, no. 1, pp. 395–400, 1998. doi: 10.1097/00020840199912000-00008.

[5] R. Martino, F. Silver, R. Teasell, M. Bayley, G. Nicholson, D. Streiner, and N. Diamant, "The Toronto bedside swallowing screening test (TOR-BSST): Development and validation of a dysphagia screening tool for patients with stroke," *Stroke*, vol. 40, no. 2, pp. 555–561, Feb. 2009. doi: 10.1161/STROKEAHA.107.510370.

[6] J. L. Coyle, "The clinical evaluation: A necessary tool for the dysphagia sleuth," *Perspectives Swallowing Disorders (Dysphagia)*, vol. 24, no. 1, pp. 18–25, Feb. 2015.

[7] R. Ding, C. Larson, J. Logemann, and A. Rademaker, "Surface electromyographic and electroglottographic studies in normal subjects under two swallow conditions: Normal and during the Mendelsohn maneuver," *Dysphagia*, vol. 17, no. 1, pp. 1–12, Jan. 2002. doi: 10.1007/s00455-001-0095-3.

[8] J. L. Coyle and J. Robbins, "Assessment and behavioral management of oropharyngeal dysphagia," *Otolaryngology Head Neck Surgery*, vol. 5, no. 1, pp. 147–152, 1997. doi: 10.1097/00020840-199706000-00001.

[9] J. L. Coyle, "Biomechanical analysis," *Videofluoroscopy: A Multidisciplinary Team Approach*. San Diego: Plural Publishing, 2012, pp. 107–122.

[10] J. M. Dudik, J. L. Coyle, and E. Sejdic, "Dysphagia screening: Contributions of cervical auscultation signals and modern signal-processing techniques," *IEEE Trans. Human–Mach. Syst.*, vol. 45, no. 4, pp. 465–477, Aug. 2015. doi: 10.1109/THMS.2015.2408615.

[11] I. Jestrovic, J. L. Coyle, and E. Sejdic, "Decoding human swallowing via electroencephalography: A state-of-the-art review," *J. Neural Eng.*, vol. 12, no. 5, Oct. 2015. doi: 10.1088/1741-2560/12/5/051001.

[12] L. Stankovic, M. Dakovic, and E. Sejdic, "Vertex-frequency analysis: A way to localize graph spectral components," *IEEE Signal Process. Mag.*, vol. 34, no. 4, pp. 176–182, 2017.

[13] I. Jestrovic, J. L. Coyle, and E. Sejdic, "Differences in brain networks during consecutive swallows detected using an optimized vertex–frequency algorithm," *Neuroscience*, vol. 344, pp. 113–123, Mar. 2017. doi: 10.1016/j.neuroscience.2016.11.047.

[14] F. Movahedi, J. L. Coyle, and E. Sejdic, "Deep belief networks for electroencephalography: A review of recent contributions and future outlooks," *IEEE J. Biomed. Health Inform.*, vol. 22, no. 3, pp. 642–652, May 2018. doi: 10.1109/JBHI.2017.2727218.

[15] J. C. Rosenbek, J. A. Robbins, E. B. Roecker, J. L. Coyle, and J. L. Wood, "A penetration-aspiration scale," *Dysphagia*, vol. 11, no. 2, pp. 93–98, 1996.

[16] B. Martin-Harris, M. B. Brodsky, Y. Michel, D. O. Castell, M. Schleicher, J. Sandidge, R. Maxwell, and J. Blair, "MBS measurement tool for swallow impairment—MBSImp: Establishing a standard," *Dysphagia*, vol. 23, no. 4, pp. 392–405, 2008. doi: 10.1007/s00455008-9185-9.

[17] W. G. Pearson, S. M. Molfenter, Z. M. Smith, and C. M. Steele, "Image-based measurement of post-swallow residue: The normalized residue ratio scale," *Dysphagia*, vol. 28, no. 2, pp. 167–177, 2013. doi: 10.1007/s00455-012-9426-9.

[18] T. L. Paine, C. A. Conway, G. A. Malandraki, and B. P. Sutton, "Simultaneous dynamic and functional MRI scanning (SimulScan) of natural swallows," *Magnetic Resonance Medicine*, vol. 65, no. 5, pp. 1247–1252, 2011. doi: 10.1002/mrm.22824.

[19] G. A. Malandraki, S. Johnson, and J. Robbins, "Functional MRI of swallowing: From neurophysiology to neuroplasticity," *Head Neck*, vol. 33, no. S1, pp. S14–S20, Oct. 2011. doi: 10.1002/hed.21903.

[20] J. B. Palmer, J. C. Drennan, and M. Baba, "Evaluation and treatment of swallowing impairments," *Amer. Family Physician*, vol. 61, no. 8, pp. 2453–2462, Apr. 2000.

[21] J. A. Robbins, A. D. Bridges, and A. Taylor. (2006). Oral, pharyngeal and esophageal motor function in aging. GI Motility Online. [Online]. Available: https://www.nature.com/gimo/contents/pt1/full/gimo39.html

SP

Ioannis Mademlis, Nikos Nikolaidis, Anastasios Tefas,
Ioannis Pitas, Tilman Wagner, and Alberto Messina

Autonomous Unmanned Aerial Vehicles Filming in Dynamic Unstructured Outdoor Environments

The recent mass commercialization of affordable unmanned aerial vehicles (UAVs), known as *drones*, has significantly altered the media production landscape, allowing for the easy acquisition of impressive aerial footage. Relevant applications include the production of movies, TV shows, or commercials as well as the filming of outdoor events or news stories. In the near future, increased drone autonomy is expected to reduce shooting costs and shift focus to the creative process, rather than the minutiae of UAV operation. This article introduces and surveys the emerging field of autonomous UAV filming and familiarizes the reader with the inherent signal processing aspects and challenges.

Obstacles to UAV filming

The rapid popularization of commercial, battery-powered, camera-equipped, vertical take-off and landing (VTOL) UAVs in the past five years has significantly impacted media production and coverage. UAVs, which partially replace dollies and helicopters in the field, are an affordable, flexible option that swiftly acquire impressive aerial footage in diverse scenarios, such as movie/TV shoots, outdoor event coverage for live or delayed broadcasts, and advertising or newsgathering. They offer fast and adaptive shot setups, can hover above a point of interest, provide access to narrow spaces, and make possible novel aerial shot types not easily achievable otherwise, all at a minimal cost. For amateur and professional filmmakers alike, the popularity of UAVs is expected to continue to rise [1].

However, a number of challenges accompany these new opportunities. Severe battery-autonomy limitations (usually fewer than 25 min of flight time), finite bandwidth in the wireless communication channels [e.g., Wi-Fi, fourth-generation (4G)/long-term evolution (LTE) cellular, or radio link] and safety-motivated legal restrictions complicate UAV usage and highlight issues that are not present when filming conventionally. Legal restrictions include a requirement that a UAV pilot maintain a direct line-of-sight with the vehicle at all times (fully autonomous civilian drones are illegal) and a maximum permissible flight altitude and minimum distance from human crowds is obeyed. Energy consumption restrictions are also important, given the amount of UAV continuous flight time possible with current battery technology as well as the related limitations on processing power and payload weight; the latter are factors that further reduce battery life.

Single-UAV shooting with a manually controlled drone is the norm in media production today, with a director/cinematographer, a pilot, and a camera operator normally required for professional filming. Initially, the director specifies the targets to be filmed, i.e., the subjects or areas of interest within the scene. Then, during preproduction, he/she designs a cinematography plan that is comprised of a temporally ordered sequence of target assignments, UAV/camera motion types (orbit, fly-by, and so on) relative to the current target and framing shot types (close-up, medium shot, and so on), which the pilot and camera operator, working in conjunction, attempt to implement during shooting. In such a setting, each target may only be captured from a specific viewpoint/angle and with a specific framing shot type at any given time instance, thus limiting the cinematographer's artistic palette. Moreover, there can only be a single target at each time, which restricts the scene coverage and results in a more static, less-immersive visual result. Finally, the "dead" time intervals required for the UAV to travel from one point to another, to shoot from a different angle, aim at a different target, or return to the recharging platform, impede smooth and unobstructed filming.

Swarms/fleets of multiple UAVs, comprised of many collaborative drones, are a viable option for overcoming the previously mentioned limitations. This is accomplished by eliminating dead time intervals and maximizing scene coverage since the participating drones can simultaneously view overlapping portions of space from different positions. Because of the possibly large number of UAVs in a fleet, a degree of decisional and functional autonomy would significantly ease

Digital Object Identifier 10.1109/MSP.2018.2875190
Date of publication: 24 December 2018

their control and lighten the burden on human operators.

However, in civilian settings, certified human pilots are required to legally operate a UAV due to safety considerations and a lack of reliable vehicle autonomy. In media production, the filming costs may become prohibitive by employing both a pilot and camera operator for each drone. Additionally, the use of multiple UAVs inherently gives rise to various coordination challenges that may limit the transparency of the shooting process, e.g., the swarm members avoiding collisions and staying out of each other's field of view (FoV).

Overcoming these issues without prohibitive resource expenditure or human intervention, and with taking into account UAV-specific concerns (e.g., battery-autonomy limitations, FoV/collision avoidance, restricted flight zones, and so on), requires intelligent algorithms for automating UAVs' flight and shooting in concert. Thus, autonomous UAV filming incorporates aerial cinematography, aerial robotics, computer vision/machine learning, and intelligent shooting. Within this field, applied signal processing has a significant role to play, especially in the form of real-time image/video analysis and in overcoming communication challenges.

Skywand is a virtual reality system that allows the user to aerially explore a three-dimensional graphics model of the scene he/she wants to cover and identify/place desired key frames within a virtual environment.

An introduction to this emerging field is presented in the next section, along with an assessment of its current state and possible directions of future progress. Conforming to recent research [2], our main focus is on outdoor live event coverage, whose challenges include filming over long distances and using potentially vast maps for navigation in (at least partially) unscripted and uncontrolled settings. The various production scenarios involve either only subsets of the challenges previously discussed or significantly more controlled shooting settings (e.g., movie sets). Indoor filming comes with its own set of issues, due to the more problematic concerns of UAV positioning/self-localization (with localization errors giving rise to heightened safety concerns in cluttered environments). However, accurate, modern indoor positioning systems exist, which mitigate such issues.

Intelligent UAV shooting

Intelligent UAV shooting is an emerging research area with significant industry potential. In general, its goal is to automate as much of the media production process as possible, while ensuring the adherence to artistic and cinematographic constraints. Some low-hanging fruit has been grabbed, but the problem remains unresolved.

For UAVs, the feasibility of manually designed drone trajectories with regard to vehicle physical limitations is an important concern. The methods in [3] retime such a trajectory and output an optimized variation that is guaranteed to be feasible without disturbing the intended visual content in the captured footage. More importantly, end-to-end systems that can execute single-UAV shooting missions have been developed [4], [5]. These systems are capable of guiding an outdoor UAV to autonomously capture high-quality footage based on cinematographic rules. Static shots and the transitions between them are computed automatically, based on well-established, visual composition principles and a list of canonical shots. Typically, the user implicitly specifies the UAV's path and the shot types to be filmed before executing a drone mission by prescribing desired "key frames," i.e., actual, temporally ordered example video frames of the intended shot within a virtual scene representation, used for preplanning autonomous footage capture during flight. The flight process is automated based on this cinematography plan.

A few commercial applications of a similar nature were released recently. Notably, Skywand is a virtual reality (VR) system that allows the user to aerially explore a three-dimensional (3-D) graphics model of the scene he/she wants to cover and identify/place desired key frames within a virtual environment. The system then computes the real UAV trajectory as well as the corresponding sequence of camera rotations necessary for a smooth shot containing these key frames to actually be filmed. Freeskies CoPilot is a mobile software suite that offers similar functionality but with a simple 3-D map instead of a VR interface. In both cases, the resulting drone autonomy and environment perception is minimal; the cinematography plan simply consists of example desired key frames that are cumbersome to define, the computed flight paths are not on-the-fly adjustable, and legal restrictions are not considered.

Although there are examples of algorithms that simply calculate the appropriate number of drones necessary to provide maximum coverage of targets from appropriate viewpoints, little to no effort has been expended on investigating the automated shooting of dynamic scenes in unstructured environments using multiple cooperating UAVs under battery autonomy limitations, with FoV/collision avoidance, and with flight-zone restrictions. In [6], an online, real-time planning algorithm that jointly optimizes feasible trajectories and control inputs for multiple UAVs is proposed. This algorithm films a cluttered dynamic indoor scene with FoV/collision avoidance by processing user-specified aesthetic objectives and high-level cinematography plans.

Autonomous UAV filming

Automated UAV flight and filming requires a number of underlying enabling technologies to be in place to ensure satisfactory operation. In the following section, relevant, state-of-the-art techniques are clustered into three groups: a two-dimensional (2-D) group, a 3-D group, and a video capture/communication group.

The 2-D group

The first required technology group, hereafter called the *2-D group*, heavily involves image/video processing and

semantic analysis operating on the image plane. It consists of a combination of 2-D visual target detection, 2-D visual target tracking, and image-based visual servoing. In principle, computer-vision and machine-learning algorithms facilitate the real-time execution of these tasks by employing only the monocular camera, which is also used for shooting.

Two-dimensional visual target detection is necessary for localizing the target's image, i.e., the region of interest (ROI) on a video frame, so that the system knows exactly how to rotate the camera to achieve central composition framing. Additionally, visual target detectors can also be exploited to identify a possible obstacle or an on-ground UAV landing site. The extracted ROI is a rectangle (described in pixel coordinates) that encloses the target's image. In drones currently available, similar methods are already employed to better adjust a manually prespecified ROI, based on the video content. In the future, greater automated UAVs are expected to rely solely on automatic visual target detection. Relevant, state-of-the-art algorithms, based on deep neural networks, are accurate and optimized for parallel execution on general-purpose graphical processing units (GPGPUs). Such high-performance hardware has recently been commercialized in a small, power-efficient form factor for embedded systems, which are ideal for on-board inclusion in UAVs (e.g., the NVIDIA Jetson series). However, current processing power and energy consumption restrictions limit what is possible with a UAV, when compared to the functionality of desktop computers.

Two-dimensional visual target tracking is used to track a prespecified ROI on the consecutive frames of a video sequence by taking advantage of spatiotemporal locality constraints and updating the ROI pixel coordinates at each video frame. Although tracking can be performed by simply redetecting the target at each video frame, a better approach is to periodically reinitialize the ROI using a 2-D visual target detector and apply a separate visual tracker for the intermediate intervals. Correlation filter-based trackers are suitable for real-time operation [7]. Currently, it is very difficult to achieve the highest accuracy in real time with state-of-the-art 2-D visual detectors and trackers given the processing power limitations of UAV hardware; however, future progress intended to reduce their computational requirements, e.g., by novel research in lightweight neural networks, is expected to alleviate this issue.

Image-based visual servoing can be used to properly rotate the camera and send suitable motion commands to the UAV motors to achieve a specific cinematography (e.g., maintaining central composition framing) or to control (e.g., landing) its purpose in an autonomous manner. In essence, it is a visual feedback-control loop that only requires a target ROI—possibly automatically derived from 2-D visual detection/tracking—as input. More-advanced visual servoing can also be used for controlling UAV motion to autonomously capture a number of desired shot types, based solely on visual input.

Two-dimensional visual target tracking is used to track a prespecified ROI on the consecutive frames of a video sequence by taking advantage of spatiotemporal locality constraints and updating the ROI pixel coordinates at each video frame.

An alternative to image-based visual servoing is reinforcement learning (RL) that uses raw video input and motor command output. Thus, the need for accurate vehicular or environmental models is eliminated. The resulting controller is therefore more adaptive to dynamic situations at the risk of losing precise, analytical solutions. It also requires advanced robotics simulator software and/or large, properly annotated image data sets for training. Deep neural networks have recently been applied in similar settings for UAV collision avoidance, indoor flight control in search and recovery operations, or high-level flight navigation [8]. An imitation-learning (IL) variation has also been explored for drone racing [9], in which a neural network learns to map video input to proper motor control commands in a supervised setting using data sets obtained by employing human pilots in a photorealistic simulator. However, these approaches have not yet been investigated for cinematography applications.

Generally, the methods contained in the 2-D group will suffice for autonomously achieving physical target following and rudimentary cinematic coverage by the drone, as well as for effective landing.

The 3-D group

The second required technology group, hereafter called the *3-D group*, operates on top of the first one and consists of a set of methods and devices that allows for functioning in global 3-D Cartesian space. These technologies are essential for achieving fully autonomous, nontrivial UAV filming with safe and effective obstacle/collision avoidance. This is mainly accomplished by applying visual simultaneous localization and mapping (SLAM) as well as by the presence of GPS receivers onboard the UAV and, ideally, on the targets being filmed.

Visual SLAM [10] can be used to detect and avoid obstacles during flight time by mapping the immediate environment and localizing the drone with respect to its specific 3-D map. Localization includes an estimation for both the position and orientation of the UAV-mounted camera at each time instance. Visual SLAM performs an incremental 3-D scene reconstruction based on the camera feed, using real-time, online variations of structure-from-motion algorithms, which are augmented by visual place recognition, graph-based map modeling, and loop closure modules. The computed map is typically a 3-D point cloud that is sparse, semidense, or dense, with the first estimated location of the UAV used as the arbitrary origin of the map coordinate system. However, since a point cloud cannot distinguish between unobserved and observed-to-be-empty space, different approaches are typically utilized for safe map representation in autonomous vehicles

(Octomap [11], an octree-based, 3-D occupancy grid is a popular choice).

Despite the fact that visual SLAM-based obstacle detection can, in principle, be performed using a single camera, additional sensors may greatly enhance the algorithm effectiveness. These sensors include an altimeter and an ultrasound module for assisting with obstacle avoidance as well as a secondary stereoscopic camera and an inertial measurement unit (IMU) that allow for a more robust operation. In fact, altimeters, IMUs, and ultrasonic sensors constitute standard equipment for all professional drones. On the other hand, lidars, which are more rarely employed visual sensors, may be used in place of stereoscopic 3-D cameras to achieve increased accuracy and performance—as well as robustness to variable environmental lighting conditions—for tasks such as SLAM. Their main benefit is the dense, 3-D scene reconstructions of unmatched quality they can provide. Although top-of-the-line lidars currently have lower refresh rates, lower resolution, lack of color perception, greater weight, and significantly higher costs than good cameras, it is very likely that future technology improvements will increase their appeal.

The 3-D maps built by visual SLAM (preferably by jointly exploiting stereoscopic 3-D camera and IMU inputs) can be aligned with the common GPS coordinate frame, using a similarity transformation. These maps can assist in global target, obstacle and UAV localization, leading to more robust operation which exploits multiple information sources.

The dynamic 3-D map built and constantly maintained by the drone can then serve as the input to a 3-D path-planning algorithm. Such algorithms for UAVs are currently able to deal with complex dynamic and kinematic constraints in real time, resulting in nearly optimal collision-free paths being computed online. Everything seen by the camera can then be mapped onto a common, 3-D-world coordinate system, and elaborate UAV motion trajectories can be planned to autonomously capture any cinematic shot type desired. Due to the dynamic nature of the environment, path planning may take place at one of two levels: a high-level, long-term plan that must be devised periodically or when important events are detected, and a low-level plan, which can locally adjust that path according to the current situation (e.g., in case a moving target suddenly changes motion direction), or according to cinematography requirements. The need for such a partitioning, however, can be reduced (to a degree) if the vehicle paths are always being planned in a variable, target-centered coordinate system, thus outputting a set of temporally ordered waypoints relative to the target. Subsequently, at each time instance during the actual execution of the path plan, the next relative waypoint can be located on the fly in the global 3-D map by exploiting the known, current target's 3-D position in the GPS coordinate frame.

Low-level motion control is an issue directly related to path planning because it involves the actual execution of the current path plan. For VTOL UAVs such as quadrotors, motion control that relies on GPS-IMU fusion is already a mature technology. In general, proportional-integral-derivative (PID) or linear-quadratic regulator controllers are used for related tasks. The PixHawk/PX4 Autopilot, a popular low-level flight trajectory control system, offers a commercial off-the-shelf PID cascade-control solution for UAVs that allows vehicle steering at various levels, ranging from designating path waypoints to directly feeding raw motion commands to the motors.

The fusion of IMU, GPS, and visual SLAM information, in principle, allows for accurate, real-time, global UAV localization in both position and orientation. Targets, on the other hand, can only be localized with regard to their position. However, target orientation must be known to accurately steer the UAV and guide the shooting process so as to autonomously capture a number of nontrivial shot types (e.g., the cinematographic requirement of filming a subject from a very specific view angle). Fortunately, operating in global, 3-D Cartesian coordinates makes it meaningful to integrate a 3-D visual-target pose estimation algorithm into the vision-processing pipeline, thus bringing image/video analysis to the forefront once more. There are two main approaches for achieving this: the computer vision approach, in which predefined landmark points are detected/tracked on the target's image and used to solve the perspective-n-point problem, or the machine-learning approach, in which the target's pose is directly regressed by a trained model that only uses the visual input. The former approach requires a 3-D model of the target to be known, while the latter approach requires a regressor properly trained on a representative, fully annotated image data set. In the event a deep neural regressor is employed, the machine-learning approach allows for integration with the 2-D visual target detector and execution on a GPGPU in real time as a unified neural network. However, to date, no commercial UAV offers these capabilities.

The existence of the global, dynamic 3-D map also makes it necessary to detect human crowds in the 2-D visual input. This process can also be integrated into the 2-D group using a deep neural network running on GPGPU in real time [12]. Subsequently, the detected crowd ROI (in pixel coordinates) may be mapped to the relevant terrain areas of the 3-D map by perspective back-projection to achieve a semantic annotation of the map. This is important because of the regulations that restrict UAV flight above human crowds. A similar process can be followed for recognizing and localizing potential emergency landing sites and flying toward them if needed.

The GPS signal, which has a usual position error of up to 5 m, is typically not available indoors and may be temporarily lost outdoors. These problems can be overcome by using differential GPS units (accurate in the range of approximately 20 cm), IMU-/GPS-/visual SLAM-fused localization, and replacing the GPS with an active radio-frequency identification or wireless positioning system solution in GPS-denied environments. These approaches, however, come with associated monetary and computational costs, which explains the fact that state-of-the-art commercial UAVs lack several capabilities derived from the 3-D group, despite being universally equipped with simple GPS receivers.

Video capture and communication group

Infrastructure for communications and related issues is critical for the successful deployment of UAV swarms in practical scenarios, especially in live event media coverage applications. Even for single-UAV missions, it is challenging to stream high-resolution video [especially 4K ultra-high definition (UHD), i.e., the norm in media production] down to a ground station with quality-of-service (QoS) guarantees, while simultaneously executing all of the previously described algorithms in real time. Video acquisition, compression, synchronization, and transmission are procedures that are easily implemented using professional cameras and open-source software. However, they consume significant processing power and energy on a computing platform already strained in resources. This issue cannot simply be solved by dedicated hardware, since the latter would come with additional energy consumption, monetary, and weight overhead considerations. That is, additional on-board hardware would increase the UAV's weight and, thus, limit flight time due to increased strain on the battery. Finally, the lack of media-production quality models with Camera Serial Interface connectivity (which allows rapid and stable capture for reliable online processing) is an additional practical concern. Therefore, at the current stage of technology, a tradeoff has to be made between the broadcast video resolution, the hardware cost, and the level of vehicle cognitive autonomy.

In simpler, nonlive coverage, i.e., when filming for a deferred broadcast or shooting a scripted sequence, on-the-fly video transmission is not required (video may simply be stored on board and retrieved later). In fact, if all processing is performed on board in a completely autonomous manner, the need for networking does not exist. However, communications are required in all other cases, including the nonlive, single-UAV filming in which a subset of the less critical algorithms previously described, e.g., crowd-/landing-site detection and high-level path planning, are executed on a computationally powerful ground station at the expense of significant latency (at best, roughly 100 ms). In general, a private QoS-guaranteeing 4G/LTE infrastructure suffices for the task, given the high mobility of the UAVs and the possibly long distances that need to be covered in outdoor event filming. Traditional Wi-Fi is a less costly, suboptimal alternative with higher latency and significantly smaller range, while public LTE networks are not reliable because of their inability to prioritize UAV communications over telephony. The main challenge lies in live broadcasting; even private LTE will not allow consistent 4K UHD video streaming, inevitably leading to reliance upon full-HD resolution.

If a swarm of multiple cooperating UAVs is employed, additional issues will likely arise. Most importantly, in live coverage, the available bandwidth may not be enough to support live, full-HD video streaming from all of the drones concurrently, resulting in a hard upper limit on the number of drones (a simple linear relation exists between the required total bandwidth and the number of employed UAVs). Furthermore, if direct coordination between the drones themselves is required (so as to autonomously capture a multiple-UAV shot to execute distributed variations of algorithms such as SLAM, or simply for redundancy/fault tolerance), then an intraswarm flying ad hoc network (FANET) should be used to support ad hoc routing and accounting for high node mobility, long distances, and rapidly varying network topology. Despite recent advances, FANETs are not yet a mature technology; for actual deployment, either custom, optimized Wi-Fi extensions must be developed, or, if inevitable, rely on LTE infrastructure at the expense of increased latency.

Autonomous features in current commercial UAVs

The algorithms used in current commercial drones do not cover all of the research methods presented in the "Autonomous UAV Filming" section. For instance, instead of pure, image-based visual servoing, more traditional optimal control methods are typically applied. These control signals are computed by explicitly constructing trajectories through configuration space, subject to costs formulated in image space. Other tasks, such as 3-D target-pose estimation or human-crowd detection, are not being performed at all, while learned control policies (e.g., via reinforcement or IL) are not commonly utilized outside laboratory settings. Advances in processing hardware (e.g., using the NVIDIA Jetson TX2 board or a future model) and algorithm efficiency/performance are expected to reduce the gap between research and commercial implementations/capabilities of autonomous UAV features.

The technologies presented are visualized in Figure 1, where the ones

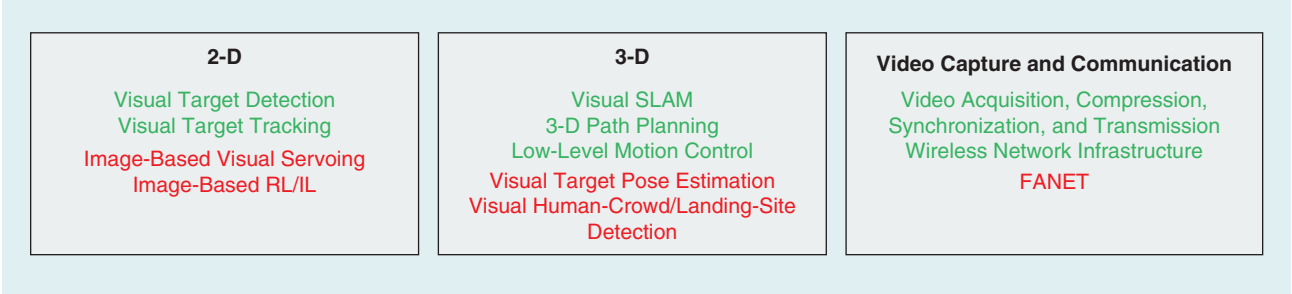


FIGURE 1. A visualization of the presented technologies clustered in three groups. Within each group, the methods currently only appearing in research settings are shown in red type, while the methods currently used in commercial UAVs are shown in green type.

currently appearing only in research settings are clearly separated from methods already employed in commercial UAVs. The methods in the 2-D and 3-D groups are further examined in Figure 2, where the input/output exchanges between them and the most important sensors are visible.

The two most popular commercial state-of-the-art UAVs for videography purposes are DJI Phantom IV Pro (using the Intel Movidius Myriad 2 vision processing unit) and the more recent Skydio R1 (built around the more powerful NVIDIA Tegra X1 system-on-a-chip). They offer similar autonomous capabilities, such as obstacle detection and avoidance, automated landing, physical target following/target orbiting enabled by visual target tracking (for low-speed, manually preselected targets) as well as automatic central composition framing, i.e., continuously rotating the camera to always keep the preselected target properly framed in the center.

However, Skydio R1 is a more advanced platform because of its more capable computing hardware and the multiple pairs of stereoscopic cameras, which function to build a 3-D occupancy volume as an environmental map. Integrating improved visual SLAM, path planning, and deep-learning-based object-detection functionalities, its main selling point is the impressive obstacle avoidance behavior, even in highly cluttered spaces. However, the resulting footage is typically lacking in cinematic quality since the encoded knowledge concerning cinematography is rudimentary, and there is no integration with intelligent-shooting algorithms.

Future prospects

During the 21st century, UAVs have evolved from remotely controlled curiosities with purely military applications into a technological revolution, taking multiple industries by storm and paving the way for widespread, embodied autonomous agents. Aerial cinematography has already been transformed by readily available advanced VTOL drones, but there is considerable room for improvement in multiple aspects. The currently limited UAV autonomy, the lack of commercial off-the-shelf cooperative UAV swarm platforms, the multitude of complications arising from legal or technological restrictions, the absence of multiple-UAV cinematography expertise are all issues prescribing directions for advancement.

We can easily imagine an ideal scenario in which a director gives high-level, concise cinematography instructions in near-natural language before filming. Subsequently, a fully autonomous UAV swarm would acquire the desired footage, while constantly and optimally adapting to the ever-changing situations arising within the shooting area under the minimal oversight of a single flight supervisor. In a less-ambitious variation, arguably more realistic at the current level of technology, the director would devise a detailed cinematography plan and, if deemed necessary, would be able to manually intervene during production.

For both scenarios, further advancements are required to realize them. Aside from upgrades in sensor technology and computational hardware, progress in UAV cognitive and functional autonomy, enabled by improvements in real-time image/video analysis and mobile networking, respectively, must be achieved in the near future.

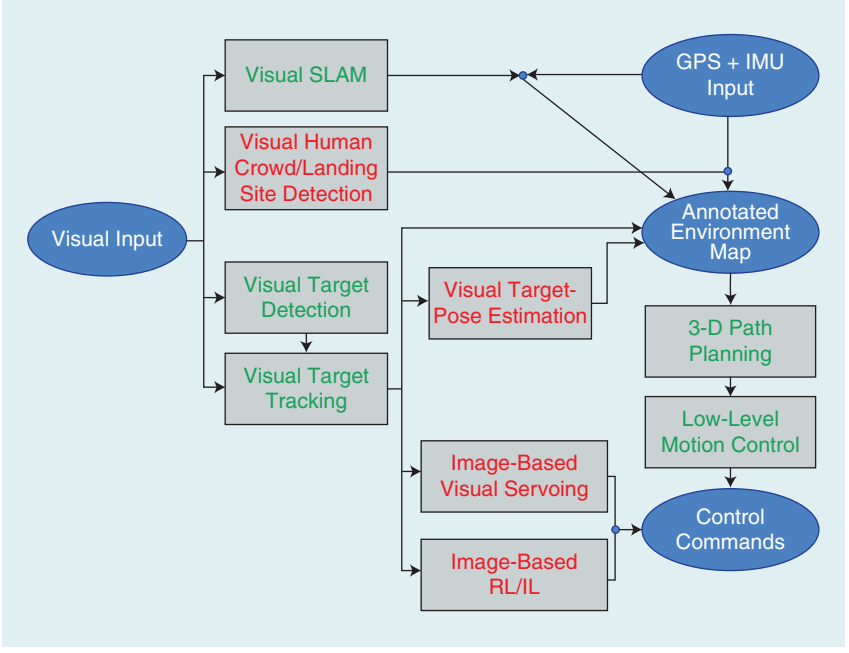


FIGURE 2. A visualization of the input/output exchanges between the presented technologies from the 2-D/3-D groups and the most important sensors.

Acknowledgments

The research leading to these results has received funding from the EU's Horizon 2020 Research and Innovation programme under grant 731667 (MULTIDRONE).

Authors

Ioannis Mademlis (imademlis@aiia.csd.auth.gr) received his B.Sc. and M.Sc. degrees in computer science from the University of Ioannina, Greece, in 2007 and 2010, respectively. He received his M.Sc. degree in cognitive science from the School of Electrical and Computer Engineering in 2014 and his Ph.D. degree in machine learning and computer vision from the Department of Informatics in 2018, each at the Aristotle University of Thessaloniki, Greece, where he is currently employed as a research assistant at the Artificial Intelligence and Information Analysis Laboratory. A computer scientist specializing in machine learning and computer vision, he has coauthored five journal articles and 14 papers presented at international conferences that detail his participation in three EU-funded research and development projects. His research interests include computer vision, machine learning,

autonomous robotics, and intelligent cinematography/editing.

Nikos Nikolaidis (nikolaid@aiia.csd.auth.gr) received his B.Sc. and Ph.D. degrees in electrical engineering from the Aristotle University of Thessaloniki, Greece, in 1991 and 1997, respectively. From 1998 to 2002, he was a postdoctoral researcher and teaching assistant in the Department of Informatics, Aristotle University of Thessaloniki, where he is currently an associate professor in the same department. He has participated in 24 European and national research and development projects and serves as a reviewer for a number of international scientific journals and conferences related to his field of expertise. He is the coauthor of *3-D Image Processing Algorithms*. He has coauthored 15 book chapters, 53 journal papers, and 164 conference papers. His works have been cited more than 4,100 times and he has an h-index of 31. His research interests include computer graphics, image and video processing and analysis, computer vision, and three-dimensional image processing.

Anastasios Tefas (tefas@aiia.csd.auth.gr) received his B.Sc. and Ph.D. degrees in informatics from the Aristotle University of Thessaloniki, Greece, in 1997 and 2002, respectively. Since 2017, he has been an associate professor in the Department of Informatics, Aristotle University of Thessaloniki, where he was also a lecturer and assistant professor from 2008 to 2017. He has participated in 12 European- and nationally funded research projects. He has coauthored 91 journal papers, 203 papers presented at international conferences, and contributed eight chapters to books in his area of expertise. His works have been cited more than 3,730 times and his h-index is 33. His research interests include computational intelligence, deep learning, pattern recognition, statistical machine learning, digital signal and image analysis, and retrieval and computer vision.

Ioannis Pitas (pitas@aiia.csd.auth.gr) received his B.Sc. and Ph.D. degrees in electrical engineering from the Aristotle University of Thessaloniki, Greece, in 1980 and 1985, respectively, where he has been a professor in the Department of Informatics since 1994. He has served as an associate editor/coeditor of eight international journals and general/technical chair of four international conferences. He has also been a member of program committees for numerous scientific conferences and workshops. He has published more than 861 papers, contributed to 44 books in his areas of expertise, and edited or coauthored 11 books. His works have been cited more than 30,000 times, and his h-index is 81. His research interests include the areas of image/video processing, intelligent digital media, machine learning, human-centered interfaces, affective computing, computer vision, three-dimensional imaging and biomedical imaging. He is a EURASIP fellow, an IEEE Distinguished Lecturer, and a Fellow of the IEEE.

Tilman Wagner (tilman.wagner@dw.com) received his M.Sc. degree in media technology from the University of Ilmenau, Germany, and his M.A. degree in international media studies from the Deutsche Welle Academy, Bonn, Germany, where he is currently an innovation manager in the Research and Cooperation Projects Department, Germany's public foreign broadcaster. His work is focused on international research projects in the areas of social media and verification, data-driven journalism, and technology for journalistic storytelling.

Alberto Messina (alberto.messina@rai.it) received his M.Sc. degree in electronic engineering in 1996 from Politecnico di Torino, Italy. He received his Ph.D. degree in computer science in 2010 from the University of Turin, Italy. He began working as a research engineer with RAI in 1996, after completing his M.Sc. thesis about objective quality evaluation of MPEG-2 coding. Since then, he has been involved in several corporate and international research projects in the areas of digital archiving, automated documentation, and automated production. He currently works as a research and development coordinator in multimedia information engineering and has authored more than 80 technical and scientific publications on the subject. He is an active member of several European Broadcasting Union technical projects and led the Strategic Programme on Media Information Management between 2010 and 2018. He is an active participant in several international standardization bodies, mainly in the areas of EBU and MPEG, where, most notably, he contributed to MPEG-7, MPEG-21, and MPEG-A extensions.

References

[1] C. Smith, *Photographer's Guide Drones*. San Rafael, CA, USA: Rocky Nook, 2016.

[2] I. Mademlis, V. Mygdalis, N. Nikolaidis, and I. Pitas, "Challenges in autonomous UAV cinematography: An overview," in *Proc. IEEE Int. Conf. Multimedia and Expo (ICME)*, 2018, pp. 1–6.

[3] M. Roberts and P. Hanrahan, "Generating dynamically feasible trajectories for quadrotor cameras," *ACM Trans. Graph.*, vol. 35, no. 4, 2016. doi: 10.1145/2897824.2925980.

[4] N. Joubert, M. Roberts, A. Truong, F. Berthouzoz, and P. Hanrahan, "An interactive tool for designing quadrotor camera shots," *ACM Trans. Graph.*, vol. 34, no. 6, 2015. doi: 10.1145/2816795.2818106.

[5] N. Joubert, D. B. Goldman, F. Berthouzoz, M. Roberts, J. A. Landay, and P. Hanrahan. (2016, Oct. 5). Towards a drone cinematographer: Guiding quadrotor cameras using visual composition principles. arXiv. [Online]. Available: https://arXiv:1610.01691

[6] T. Nägeli, L. Meier, A. Domahidi, J. Alonso-Mora, and O. Hilliges, "Real-time planning for automated multi-view drone cinematography," *ACM Trans. Graph.*, vol. 36, no. 4, 2017. doi: 10.1145/3072959.3073712.

[7] O. Zachariadis, V. Mygdalis, I. Mademlis, N. Nikolaidis, and I. Pitas, "2D visual tracking for sports UAV cinematography applications," in *Proc. IEEE Global Conf. Signal and Information Processing (GlobalSIP)*, 2017, pp. 36–40.

[8] A. Carrio, C. Sampedro, A. Rodriguez-Ramos, and P. Campoy, "A review of deep learning methods and applications for unmanned aerial vehicles," *J. Sensors*, vol. 2017, Aug. 2017. doi: 10.1155/2017/3296874.

[9] G. Li, M. Mueller, V. Casser, N. Smith, D. L. Michels, and B. Ghanem. (2018, Mar. 3). Teaching UAVs to race with observational imitation learning. arXiv. [Online]. Available: https://arXiv:1803.01129.

[10] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, 2015.

[11] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Auton. Robots*, vol. 34, no. 3, pp. 189–206, 2013.

[12] M. Tzelepi and A. Tefas, "Human crowd detection for drone flight safety using convolutional neural networks," in *Proc. of EURASIP European Signal Processing Conf. (EUSIPCO)*, 2017, pp. 743–747.

SP

Rodrigo Capobianco Guido

Paraconsistent Feature Engineering

Today's modern world is filled with uncertainties and contradictions. As artificial intelligence (AI) advances, machines are frequently expected to mimic the human brain and, consequently, face the conflicts associated with this task. To overcome them, feature engineering has emerged as the field of science responsible for turning raw data into relevant input information, setting up classifiers in the fused digital signal processing (DSP) and pattern recognition (PR) domain. Despite the ongoing efforts to improve feature learning, handcrafted extraction still plays a very important role. In this context, a careful choice of features is extremely relevant for creating an accurate classification. This article sheds light on the problem of feature quality by using a nonclassical logical system capable of handling conflictive situations. It is known as *paraconsistent logic* (*PL*).

Relevance

More often than not, AI algorithms specifically dedicated to PR have dominated DSP systems, stimulating further studies on feature engineering [1]. Whenever classic logic fails to address an issue in this field, PL [2] may be a solution. Therefore, this study, which is complemented by a numerical example, is of paramount importance.

Prerequisites

Very basic notions of classic logic and PR are desirable but not imperative. Readers who are unfamiliar with the topics discussed in this article may want to consult the literature referenced in [3] and [4] before proceeding any further. Basic comments about PL, which comprises the focus of this article, are provided in future sections and can also be drawn from [2], [5], and [6].

Problem statement and solution

Problem statement

Consider an N-class classification problem for which the classes $\{C_1, C_2, \ldots, C_N\}$ are represented by a certain number of T-sample long feature vectors, all of which are obtained based on a handcrafted extraction. The system engineer, who intuitively selects the features, needs a quantitative evaluation on their suitability for either supervised or unsupervised learning [4]. In other words, the question is: "Are those features convenient to classify my data?"

Essentially, the answer not only depends on the features themselves but also on the technique adopted to analyze them. Typically, some features yield poor categorization when associated with a certain classifier, while being excellent whenever used in conjunction with another. A modest strategy, such as an ordinary distance metric, requires exceptionally prepared features to treat a real-world problem successfully. On the contrary, a deep neural network is possibly capable of solving difficult tasks if it receives only a set of modest features as input.

Let us consider the case in which a weak classifier is able to generate accurate results for a specific task using a given set of features. A better classifier, therefore, is guaranteed to produce good results using the same set of features. Thus, selecting the best features based on a modest method, which unavoidably works as a simple classifier, allows for generalization, i.e., the features will efficiently address the problem in conjunction with basic or sophisticated classifiers. For this reason, elementary techniques are adopted here.

As the PR community knows, favorable feature vectors exhibit considerable similarity when extracted from inputs of a particular class and a notable distinction when coming from different classes. Furthermore, whenever the features substantially contribute to solve a problem, they avoid the simple forwarding of a nonpolished issue to the classifier, which is the next stage of the classification system. As a result, the problem is that of defining a quantitative strategy for investigating the extracted features, observing that independent intraclass and interclass analyses may cause conflicts.

Digital Object Identifier 10.1109/MSP.2018.2874549
Date of publication: 24 December 2018

 1053-5888/19©2019IEEE

Solution

As most of the classifiers, the technique adopted to solve the stated problem also requires all of the feature vectors to first be normalized within the $0 \sim 1$ range, which allows for a proper scale of inspection. There are different ways to do this; for example, to analyze the behavior of a particular physical entity at the time intervals it is observed, we frequently register its percentual distribution, not its values, as in methods A_1, A_2, B_1, and B_2 from [7] and [8]. Contrarily, proper records of magnitudes related to a maximum amount contain the unity as the largest possible value, as in methods A_3 and B_3, also defined in [7] and [8]. Hypothetical examples of the former and latter cases are the feature vectors {0.28, 0.25, 0.24, 0.23}, for which the components add up to 1, and {0.82, 0.86, 0.89, 1}, where 1 is the baseline, respectively. Occasionally, measuring the amplitudes related to a predefined independent value also makes sense, as in the feature vector {0.80, 0.30, 0.98, 0.04}, for which 1 neither appears nor consists of the additive sum of its elements. Therefore, normalization as well as the choice of features are important preprocessing steps that help to conveniently represent the physical entity of interest, depending on the specific problem assessed. Works in [7]–[9] contain various illustrative examples and allow for practical hands-on experience in such a task, which is at the discretion of the system engineer.

Once the normalization is completed, we are ready to study the feature vectors adequately. As I mentioned, this problem is solved by using two independent criteria: one to quantify the intraclass similarities and another to reflect the interclass dissimilarities. Each is represented by the quantities α, which expresses the level of faith in the features, and β, which specifies their level of discredit, respectively, where $(0 \le \alpha, \beta \le 1)$. Independence indicates that α and β are not complementary, i.e., $\alpha + \beta$ might be different from the unity, implying that ordinary logic [3] is not the proper tool for treating this problem.

As a basis for performing the intraclass analysis, we note that the amplitude, or range, of a set of K real numbers, i.e., $s[\cdot] = \{s_0, s_1, \ldots, s_{K-1}\}$, is the simplest way to measure its deviation [10]. It is defined as $A = L(s[\cdot]) - S(s[\cdot])$, i.e., the difference between the largest and the smallest values in $s[\cdot]$. Notably, the lesser A is, the closer the scalars in the set are, and vice versa. Since $(0 \le A \le 1)$ due to its previous normalizations, $Y = (1 - A)$ can be used as a standardized measure of similarity among the scalars in such a way that $Y \approx 0$ and $Y \approx 1$ indicate low and high similarities, respectively.

In this article, we are interested in finding the similarity among feature vectors of size T, not scalars. Consequently, we can perform, separately for each class, an element-wise similarity-vector computation where the ith element represents the similarity among the corresponding components of the feature vectors, as shown in Figure 1. Hereafter, the similarity vectors are intuitively named as $\mathrm{svC}_1[\cdot], \mathrm{svC}_2[\cdot], \ldots, \mathrm{svC}_N[\cdot]$. Their corresponding arithmetic means, i.e.,

$$\bar{Y}(C_1) = \frac{1}{T}\sum_{i=0}^{T-1} \mathrm{svC}_1[i],$$

$$\bar{Y}(C_2) = \frac{1}{T}\sum_{i=0}^{T-1} \mathrm{svC}_2[i],$$

$$\cdots$$

$$\bar{Y}(C_N) = \frac{1}{T}\sum_{i=0}^{T-1} \mathrm{svC}_N[i],$$

which are used to balance their respective individual values, correspond to the intraclass similarities. Ideally, all of them would be close to 1. Thus, to assess the worst case, we define α as the smallest among the intraclass similarities, i.e., $\alpha = \min\{\bar{Y}(C_1), \bar{Y}(C_2), \ldots, \bar{Y}(C_N)\}$. Once the calculations to find α are completed, the next step is to define β as follows.

To perform the interclass analysis, as shown in Figure 1, we initially compute two range vectors of size T for each class: one with element-wise minimum values of the whole set and the other with element-wise maximum values of the whole set. Thus, range vectors store the exact interval containing all the feature vector values from their respective classes. Just to clarify, if a certain class is composed of the feature vectors {0.5, 0.4}, {0.6, 0.1} and {0.3, 0.2}, then, the corresponding range vectors for the smallest and largest values are {0.3, 0.1} and {0.6, 0.4}, respectively.

We then perform an element-wise comparison between the feature vectors from each class and the range vectors from all of the other classes to define R, i.e., the number of feature vector elements, if any, that overlap the respective range. Each overlapped element means that a feature from one class invaded the range used by the features from another class, which is disadvantageous for the classification, i.e., a straight line is not capable of separating between those interclass features, and a nonlinear technique is likely necessary. Finally, we define

$$\beta = \frac{R}{F},$$

where F is the maximum possible number of overlaps. Assuming that each one of the N classes contains X feature vectors of size T, then $F = N \cdot (N-1) \cdot X \cdot T$. Notably, we consider one comparison as a look at a particular position of both range vectors of a class.

The quantities α and β create two distinct, and possibly conflictive, measures. At this point, it is important to note that $\alpha = 1$ strongly suggests that the intraclass feature vectors are similar and represent their respective classes precisely. Complementarily, $\beta = 0$ suggests that the interclass feature vectors do not overlap, thus reassuring our faith in them. Both of these conditions are expected, at least approximately, for an accurate and easy-to-perform classification.

Despite this best case, there are three other extreme possibilities involving those measures, i.e., $(\alpha, \beta) = (0, 1)$, $(\alpha, \beta) = (0, 0)$, and $(\alpha, \beta) = (1, 1)$, in addition to an infinite number of intermediary values where $(0 < \alpha, \beta < 1)$. To shed some light on this scenario, we adopt a simple yet flexible tool used to treat conflictive information as potentially informative: PL. Described in [2], [5], and [6], it uses α and β to calculate the degree of certainty, i.e., $G_1 = \alpha - \beta$, and the degree of contradiction, i.e., $G_2 = \alpha + \beta - 1$, of a statement,

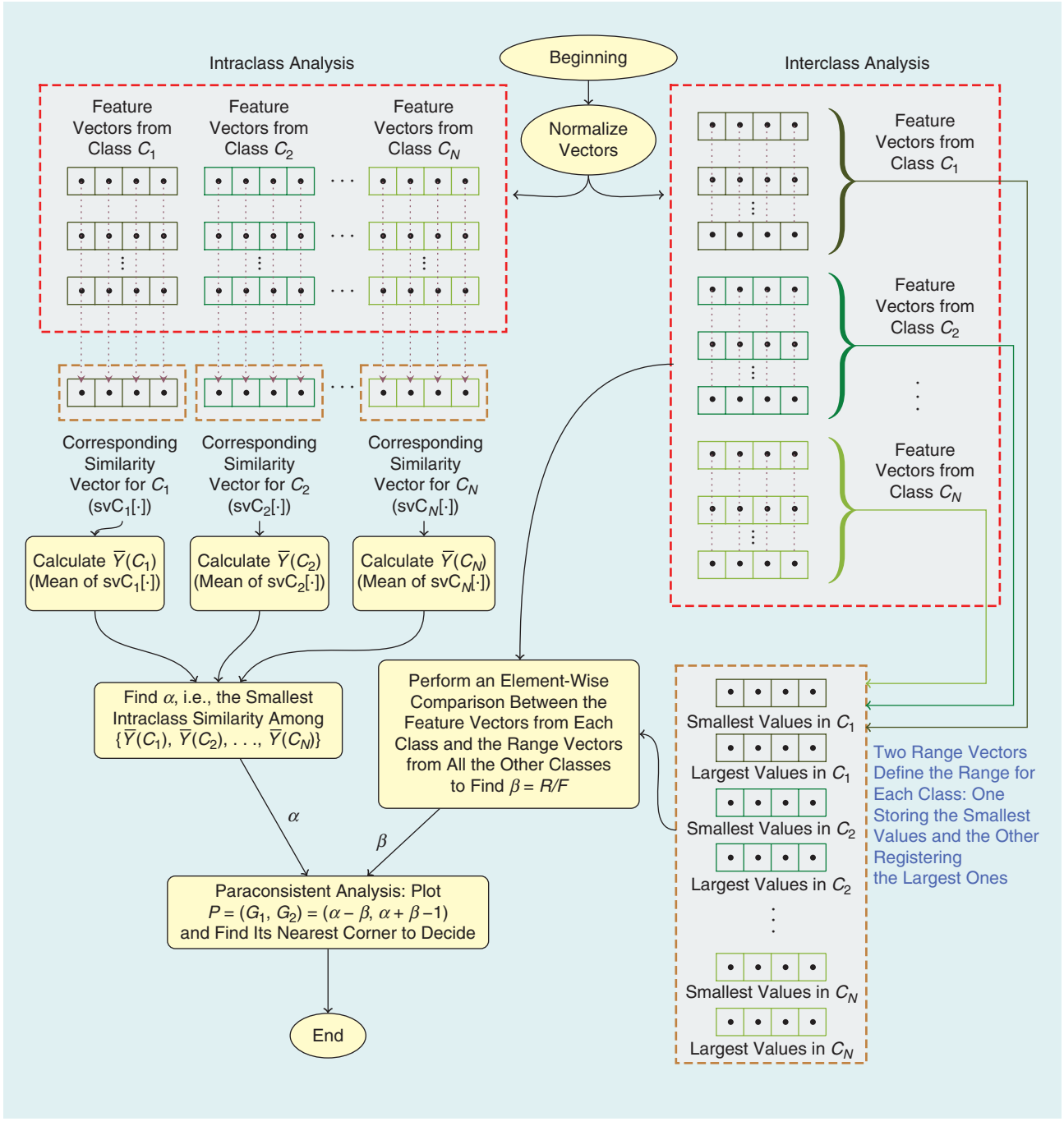


FIGURE 1. The proposed approach based on PL: (upper left) intraclass analysis, (right) interclass analysis, and (lower left) paraconsistent analysis.

where $(-1 \le G_1, G_2 \le 1)$, as shown in Figure 1.

On one hand, certainty varies from falsehood to truth, i.e., $G_1 = -1$ and $G_1 = 1$, respectively. On the other hand, contradiction varies from indefinition to ambiguity, i.e., $G_2 = -1$ and $G_2 = 1$, respectively. Figure 2, which contains additional explanations, shows the paraconsistent plane where the point $P = (G_1, G_2)$ is plotted to allow for the intended analysis. Particularly, the distances from P to the corners $(-1, 0)$, $(1, 0)$, $(0, -1)$ and $(0, 1)$, i.e., $\sqrt{(G_1+1)^2+(G_2)^2}$, $\sqrt{(G_1-1)^2+(G_2)^2}$, $\sqrt{(G_1)^2+(G_2+1)^2}$, and $\sqrt{(G_1)^2+(G_2-1)^2}$, respectively, reveal the following conclusion: favorable feature vectors place P closer to the corner $(1, 0)$ than to the other corners.

Numerical example

Problem statement

To illustrate the proposed approach numerically, we assume a hypothetical three-class classification problem in which the four bidimensional normalized feature vectors in each class are such that:

- class C_1: $\{0.90, 0.12\}$, $\{0.88, 0.14\}$, $\{0.88, 0.13\}$, and $\{0.89, 0.11\}$

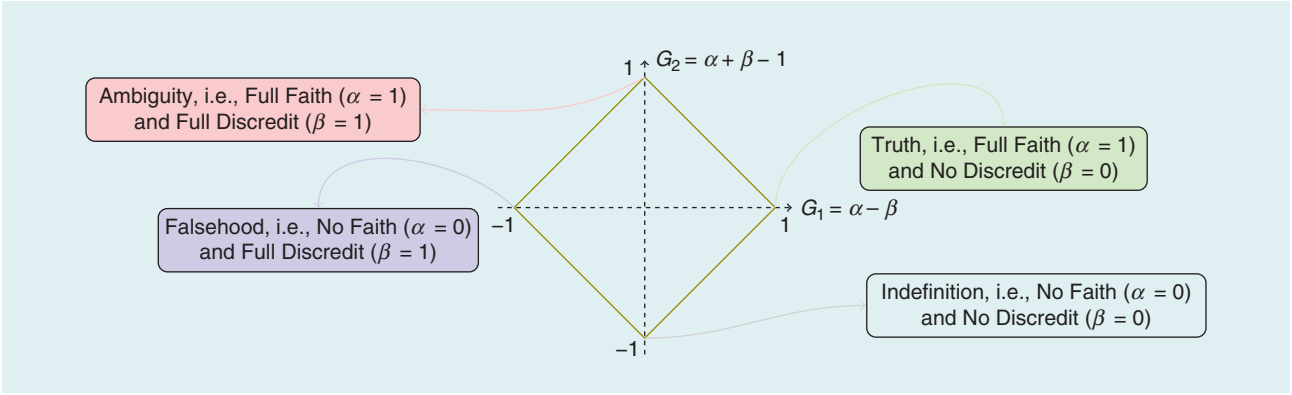


FIGURE 2. The paraconsistent plane. Axes G_1 and G_2 represent the degrees of certainty and contradiction, respectively. As shown, $(G_1, G_2) = (-1, 0)$, $(G_1, G_2) = (1, 0)$, $(G_1, G_2) = (0, -1)$, and $(G_1, G_2) = (0, 1)$ represent falsehood, truth, indefinition, and ambiguity, respectively. In the context of the proposed technique, as $P = (G_1, G_2)$ approaches these corners, we see that "a strong classifier is likely to be required because intraclass feature vectors are notably scattered and interclass feature vectors significantly overlap"; "a weak classifier is likely to solve the problem because intraclass feature vectors are consistently grouped together and interclass feature vectors minimally overlap"; "the features are likely to cause an indefinition, i.e., both intraclass and interclass feature vectors are much different," and "the features are likely to cause an ambiguity, i.e., both intraclass and interclass feature vectors are considerably similar," respectively. Overall, the shorter the path from $P = (G_1, G_2)$ to the corner (1, 0) is, the better the feature vectors are, and the weaker the classifier used in conjunction with them can be. Additionally, if P approaches the corner (1, 0) by the fourth quadrant, i.e., the lower-right triangle that forms the paraconsistent plane, a linear classifier solves the problem because, in that region, $\beta = 0$, i.e., no interclass overlap exists.

- class C_2: {0.55, 0.53}, {0.53, 0.55}, {0.54, 0.54}, and {0.56, 0.54}
- class C_3: {0.10, 0.88}, {0.11, 0.86}, {0.12, 0.87}, and {0.11, 0.88}.

The problem is to determine how adequate these feature vectors are to classify the corresponding raw input data.

Solution

To perform the intraclass analysis, we compute the amplitude vectors as follows:

- class C_1: $\{0.90 - 0.88, 0.14 - 0.11\} = \{0.02, 0.03\}$
- class C_2: $\{0.56 - 0.53, 0.55 - 0.53\} = \{0.03, 0.02\}$
- class C_3: $\{0.12 - 0.10, 0.88 - 0.86\} = \{0.02, 0.02\}$.

Thus, the corresponding similarity vectors are

- class C_1: $\{1 - 0.02, 1 - 0.03\} = \{0.98, 0.97\}$
- class C_2: $\{1 - 0.03, 1 - 0.02\} = \{0.97, 0.98\}$
- class C_3: $\{1 - 0.02, 1 - 0.02\} = \{0.98, 0.98\}$.

and the respective means are

$$\bar{Y}(C_1): \frac{0.98 + 0.97}{2} = 0.975$$

$$\bar{Y}(C_2): \frac{0.97 + 0.98}{2} = 0.975$$

$$\bar{Y}(C_3): \frac{0.98 + 0.98}{2} = 0.980.$$

Consequently, $\alpha = \min\{0.975, 0.975, 0.980\} = 0.975$.

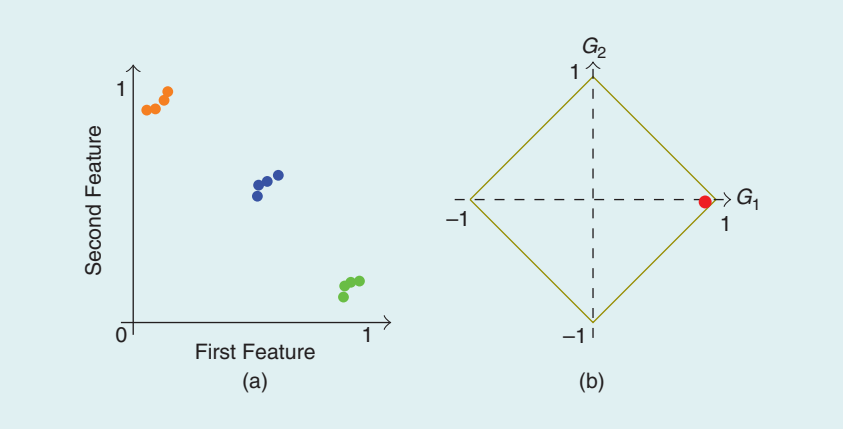


FIGURE 3. (a) The feature vectors from classes C_1, C_2, and C_3 in green, blue, and orange, respectively, localized in the space (clearly, they are linearly separable). (b) The point P is represented as a red dot in the paraconsistent plane for the numerical example.

To perform the interclass analysis, we initially compute the range vectors as follows:

- class C_1: {0.88, 0.11}, containing the smallest components, and {0.90, 0.14} with the largest elements
- class C_2: {0.53, 0.53}, containing the smallest components, and {0.56, 0.55} with the largest elements
- class C_3: {0.10, 0.86}, containing the smallest components, and {0.12, 0.88} with the largest elements.

Then, comparing each feature vector component from one class to the range vector components from all of the other classes, we detect no overlap, i.e., $R = 0$, among $F = 3 \cdot (3-1) \cdot 4 \cdot 2 = 48$ possible overlaps, as shown in Figure 3(a). Thus, $\beta = R/F = 0/48 = 0$.

Proceeding with the paraconsistent analysis, we compute $G_1 = \alpha - \beta = 0.975 - 0 = 0.975$ and $G_2 = \alpha + \beta - 1 = 0.975 + 0 - 1 = -0.025$. Plotting $P = (G_1, G_2)$, as shown in Figure 3(b), and calculating its distance (d) to the four corners of the paraconsistent plane, we have

$$\begin{aligned} &\mathrm{d}((G1, G2), (-1, 0)) \\ &\quad = \sqrt{(0.975 + 1)^2 + (-0.025)^2} \\ &\quad = 1.975, \end{aligned}$$

$$d((G1, G2), (1, 0)) = \sqrt{(0.975 - 1)^2 + (-0.025)^2} = 0.035,$$

$$d((G1, G2), (0, -1)) = \sqrt{(0.975)^2 + (-0.025 + 1)^2} = 1.415$$

and

$$d((G1, G2), (0, 1)) = \sqrt{(0.975)^2 + (-0.025 - 1)^2} = 1.416.$$

Therefore, since the smallest among the distances places P closer to (1, 0) than to the other corners and P is in the fourth quadrant, the features are linearly separable, thus providing an accurate classification based on a modest strategy. Although an advanced classifier may also be used to correctly interpret the input data, it is not required in this case.

What we have learned

Based on the information in this article, the reader may effectively use PL to overcome conflictive information and investigate how adequate a set of features is to classify data in an N-class problem. The proposed solution provides highly generalizable results and disregards the specific classifier that will be used in conjunction with these features.

Author

Rodrigo Capobianco Guido (guido@ieee.org) received his B.Sc. degree in 1998, his M.Sc. degree in 2000, his Ph.D. degree in 2003, and his L.D. degree in 2008. He is an associate professor at São Paulo State University (UNESP) in José do Rio Preto, Brazil, and is a Senior Member of the IEEE. Contact him for more information on how to obtain the C++ source code necessary to implement the technique described in this article.

References

[1] A. Zheng and A. Casari, *Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists*. Birmingham, UK: O'Reilly Media, 2018.

[2] W. Carnielli and M. E. Coniglio, *Paraconsistent Logic: Consistency, Contradiction and Negation*. New York: Springer-Verlag, 2016.

[3] R. M. Smullyan, *A Beginner's Guide to Mathematical Logic*. New York: Dover, 2014.

[4] C. M. Bishop, *Pattern Recognition and Machine Learning*, New York: Springer-Verlag, 2018.

[5] F. R. Carvalho and J. M. Abe, *A Paraconsistent Decision-Making Method*. New York: Springer-Verlag, 2018.

[6] J. M. Abe, *Paraconsistent Intelligent-Based Systems: New Trends in the Applications of Paraconsistency*. New York: Springer-Verlag, 2015.

[7] R. C. Guido, "A tutorial on signal energy and its applications," *Neurocomput.*, vol. 179, pp. 264–282, Feb. 2016.

[8] R. C. Guido, "ZCR-aided neurocomputing: A study with applications," *Knowledge-Based Syst.*, vol. 105, pp. 248–269, Aug. 2016.

[9] R. C. Guido, "A tutorial-review on entropy-based handcrafted feature extraction for information fusion," *Inform. Fusion*, vol. 41, pp. 161–175, May 2018.

[10] P. Bruce and A. Bruce, *Practical Statistics for Data Scientists: 50 Essential Concepts*. Sebastopol, CA: O'Reilly Media, 2017.

SP

IEEE Signal Processing Society

JOIN the
IEEE Signal Processing Cup 2019:
Search & Rescue with Drone-Embedded Sound Source Localization

The IEEE Signal Processing Society is proud to announce the sixth edition of the Signal Processing Cup: an exciting audio-based drone-embedded search and rescue challenge.

- **Goal:** Building a system capable of localizing a sound source based on audio recordings made with an 8-channel microphone array embedded in an unmanned aerial vehicle (UAV)
- **Eligibility:** Any team composed of one faculty member, at most one graduate student and 3-10 undergraduate students is welcomed to join the open competition
- **Dataset:** A novel dataset of UAV-embedded microphone-array recordings is provided for the challenge
- **Website:** The detailed guidelines, dataset and inscription portal are available on the official website: https://signalprocessingsociety.org/get-involved/signal-processing-cup
- **Prize:** The three teams with highest performance in the open competition will be selected as finalists and will be invited to participate in the final competition at ICASSP 2019. The champion team will receive a grand prize of $5,000. The first and the second runner-up will receive a prize of $2,500 and $1,500, respectively, in addition to travel grants and complimentary conference registrations.

A joint initiative of

The IEEE Technical Committee for Audio and Acoustic Signal Processing

The IEEE Autonomous System Initiative

Important dates:

Data release: November 14, 2018
Submission deadline: February 28, 2019
Finalists announcement: March 20, 2019
Final @ICASSP: May 12-17, 2019

Sponsored by MathWorks®

Digital Object Identifier 10.1109/MSP.2018.2876521

Leonid Moroz and Volodymyr Samotyy

Efficient Floating-Point Division for Digital Signal Processing Application

Floating-point division is an expensive operation for processors in digital signal processing (DSP). The basis of the division operation is finding the reciprocal of the divisor. We present a reciprocal algorithm with four multiplications for single accuracy and six for double accuracy. The algorithm specifically includes bithack (known as *magic constant*) operations and floating-point addition, multiplication, and the fused multiply-add (fma) operation. The proposed approach improves two characteristics of the division process: accuracy and number of steps. This second characteristic has a direct impact on the division operation performance time and on the amount of equipment required for hardware implementation.

Challenges of floating-point division

Division is one of the more difficult of the four arithmetic operations (addition, subtraction, multiplication, and division) in processors, microcontrollers, and field-programmable gate arrays (FPGAs), which are used for DSP. Division or square root extraction can be used for fixed-point [1], [2] and floating-point [3]–[5] numbers. All modern DSP devices execute division in floating-point format. For example, TMS320F2807x Piccolo Microcontrollers have a built-in IEEE-754 single-precision floating-point unit (FPU) [4]. Microcontrollers, such as the STM32 Cortex-M4 and STM32 Cortex-M7, are equipped with single-precision FPUs for division [3]. However, many modern floating-point microcontrollers (such as those of the Cortex A53 and Cortex A57) have fast accurate reciprocal instructions like the SSE instructions in a central processing unit (CPU) for the initial approximations, which are then improved with the help of Newton–Raphson iterations. (SSE is a set of instructions for general purpose processors. These instructions give us the ability to perform, for example, reciprocal over several processor cycles.) Some modern FPGAs also have a set of single-precision floating-point blocks for add, multiply, and fma operations [5] that can be used to the build division units. Therefore, it is important to develop appropriate algorithms to perform division operations for platforms that support floating-point computing and that do not have a hardware division unit or the proper instructions for the division.

In this article, the reciprocal of the divisor's simple algorithms (which are the basis for division) are proposed. These algorithms belong to the group of iterative algorithms. They initially receive an approximation using the so-called magic constant and are then used to implement an iterative process with Newton–Raphson formulas. The reciprocal algorithms are suitable for software and hardware implementations, e.g., in microcontrollers that lack an FPU and in FPGAs.

Known algorithms

An algorithm with magic constant for the implementation of a reciprocal function $1/x$ for float-type number x was described for the first time in [6]. It consists of the following code:

```
1.float rcp_1(float x)
2.{
3.  int i=*(int*)&x;
4.  i=0x7f000000-i;
5.  float y=*(float*)&i;
6.  y=y*(2.0f-x*y);
7.  y=y*(2.0f-x*y);
8.  return y;
9.}
```

This code, written in C/C++, will be referred to as the *reciprocal*. In line 3, we convert bits of the variable x (type float) to a variable i (type int). In line 4, we determine an initial approximation (then subject to the iteration process) of the reciprocal, where R = 0x7f000000 is a magic constant for IEEE-754 implementations. In line 5, we convert bits of the variable i (type int) to a variable y (type float). Lines 6 and 7 contain two classic subsequent Newton–Raphson iterations.

If the maximum relative error of calculations is designated by $|\delta_{\max}|$, then the accuracy of this algorithm is only

Digital Object Identifier 10.1109/MSP.2018.2875977
Date of publication: 24 December 2018

$$|\delta_{\max}| = 2.44 \cdot 10^{-4},$$
$$\text{or } -\log_2(|\delta_{\max}|) = 12$$

correct bits. Better accuracy with magic constants is achieved in [7] and [8]: 16.5 correct bits. However, the improved algorithm with modified Newton–Raphson iterations from [9] is more precise:

```
float rcp_2(float x)
{
  int i=*(int*)&x;
  i=0x7ef311c3-i;
  float y=*(float*)&i;
  y=y*(2.00130856f-x*y);
  y=y*(2.00000084f-x*y);
  return y;
}
```

The accuracy of the algorithm is approximately $|\delta_{\max}| = 1.014 \cdot 10^{-6}$, or 19.9 correct bits. Graphics of the initial approximations and the relative errors of these algorithms are shown; e.g., the initial approximations, $(x \in [1,2))$, are shown in Figure 1. Relative errors of the initial approximation for both algorithms are shown in Figure 2. The described algorithms include only four multiplication operations.

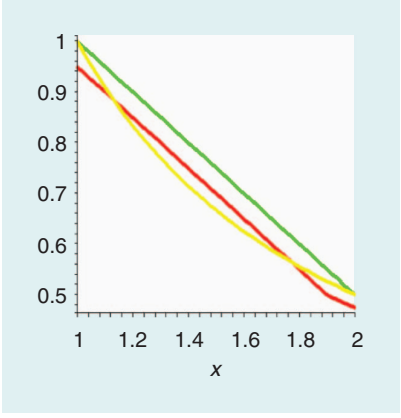


FIGURE 1. Initial approximations of the rcp_1 (green) and rcp_2 (red) algorithms. The yellow line indicates 1/*x*.

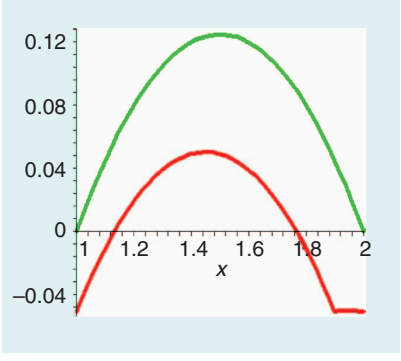


FIGURE 2. Relative errors of the initial approximations for the rcp_1 (green) and rcp_2 (red) algorithms.

The magic constant for double-type numbers is described in [7] (0x7fde9f73aabb2400) and [8] (0x7fde623822fc16e6). Fewer errors occur for the second constant at the $|\delta_{\max}| = 4.237 \cdot 10^{-11}$ level, which is 34.4 correct bits after three classic Newton–Raphson iterations (six multiplications), and at $|\delta_{\max}| = 2.22 \cdot 10^{-16}$, which is 52 correct bits after four classic Newton–Raphson iterations (eight multiplications).

This article discusses the construction of reciprocal algorithms with a diminished number of multiplications without loss of accuracy.

General theory of the algorithm

We now consider how the reciprocal algorithm works. Suppose we have a floating-point number

$$x = (-1)^{S_x} M_x \cdot 2^{E_x}, \tag{1}$$

where S_x is the sign ($S_x = \{0,1\}$, 0, 0 for positive numbers and 1 for negative numbers] and E_x is the order (exponent), which in general cases is determined by

$$E_x = \lfloor \log_2 x \rfloor = \text{floor}(\log_2 x). \tag{2}$$

The mantissa M_x is calculated according to

$$M_x = \frac{x}{2^{E_x}}. \tag{3}$$

It lies in the range $M_x = [1,2)$ and has the form $M_x = 1 + m_x$, where m_x is the fractional part of the mantissa. From this point on, to simplify the calculations, it will be assumed that $x > 0$ so that $S_x = 0$. The single-precision format of the IEEE-754 standard [11] uses a 32-bit register to store the binary representation of the number x:

- 1 bit (sign field) for S_x
- 8 bits (exponent field) for e_x, where $e_x = E_x + \text{bias}$ is a shifted order (shifted bias = 127 for single precision)
- 23 bits (mantissa field) for m_x.

The mantissa has a phantom (hidden) most significant bit that is not shown, so the appropriate 23-bit part of the 32-bit register stores only m_x. Because of this, the integer decimal number I_x corresponding to the binary representation of the number x in the IEEE-754 standard is depicted as

$$I_x = e_x \cdot N_m + m_x \cdot N_m = (e_x + m_x)N_m$$
$$= (\text{bias} + E_x + x \cdot 2^{-E_x} - 1) \cdot N_m. \tag{4}$$

$N_m = 2^{23}$ for single precision. The values of I_x could be considered as an approximation of the binary logarithm x.

If we take the binary logarithm of the number x, change its sign, and then convert it into a number y through the exponential function (base 2), we obtain the inverse of the value. This exact idea underlies the basis of the algorithm. However, float numbers in integer type are only coarse approximations of the binary logarithm x, so the inverse value obtained in this way can be considered only as an initial approximate reciprocal, which is clarified with the help of Newton–Raphson iterative equations.

The magic constant R is located in the 32-bit register as the positive integer number

$$R = Q \cdot N_m + T, \tag{5}$$

where Q is an integer number in the exponent field, and T is an integer number in the mantissa field; thus, $Q = 2 * \text{bias} - 1$. Next, the integer difference d is sought (the change of sign of the approximate binary logarithm of x occurs here):

$$d = R - I_x. \tag{6}$$

Then, the integer d is converted to a real number in the floating-point format, which will be the initial approximation y_0. The sequence of operations follows:

- a biased exponent should be found

$$e_y = \text{floor}\left(\frac{d}{N_m}\right) \tag{7}$$

- a nonbiased exponent should be identified

$$E_y = e_y - \text{bias} \tag{8}$$

- a mantissa fractional part should be found

$$m_y = \frac{d - E_y \cdot N_m}{N_m} = \frac{d}{N_m} - E_y \quad (9)$$

- an initial approximation y_0 should be found in the form

$$y_0 = (1 + m_y) \cdot 2^{E_y}. \quad (10)$$

The relative error of the initial approximation could be estimated from y_0 as

$$\delta_0 = y_0 x - 1. \quad (11)$$

After finding y_0, iteration is performed with the classic Newton–Raphson formula for the reciprocal,

$$y_n = y_{n-1}(2 - x y_{n-1}). \quad (12)$$

The strict mathematical model for the initial approximation of y_0 can be formed. First, it is necessary to write an expression of y_0. If we sequentially describe the converting of x to I_x, and the inverse converting d to y_0 using (1)–(10), then we conclude that the initial approximation of y_0 in the general case is described by

$$y_0 = (1 + t - 2^{E_x} x - E_x - E_y) \cdot 2^{E_y}, \quad (13)$$

where

$$t = \frac{T}{N_m}. \quad (14)$$

The expression (13) is a mathematical model for the formation of an initial approximation y_0 for number x, which is performed in the IEEE-754 standard.

Analysis of (13) shows that the approximation could be presented in the form

$$y_0 = (\alpha x + \beta \cdot 2^{E_y}), \quad (15)$$

where

$$\alpha = -2^{E_x}, \quad (16)$$
$$\beta = 1 + t - E_x - E_y, \quad (17)$$

and y_0 is the piecewise linear approximation of reciprocal.

Now a detailed analysis of (7) and (8) can be performed. It is possible to write

$$E_y = -E_x - 1 + \text{floor}(t - m_x). \quad (18)$$

Therefore, taking into account $0 \le t < 1$ and $0 \le m_x < 1$, the order E_y could have only two values:

$$E_y = -E_x - 1, \text{ when } m_x \le t \quad (19)$$

and

$$E_y = -E_x - 2, \text{ when } m_x > t. \quad (20)$$

From (15), (19), and (20), it follows that the interval $x \in [1, 2)$ is divided into two parts: $x \in [1, x_t)$ and $x \in [x_t, 2)$, where $x_t = 1 + t$. In the first part of the interval $x \in [1, x_t)$, E_y is described by (19), so the linear initial approximation is

$$y_{01} = 1 - \frac{x}{2} + \frac{t}{2}. \quad (21)$$

In the second part, $x \in [x_t, 2)$, E_y is determined by (20), so that

$$y_{02} = \frac{x}{4} + \frac{3}{4} + \frac{t}{4}. \quad (22)$$

For other values of x that lie in ranges that are multiples of (1, 2), conditions (19) and (20) are saved, and the initial approximation will be

$$y_{01} = \frac{1}{2}(1 - m_x + t)2^{-E_x}, \quad (23)$$

and

$$y_{02} = \frac{1}{4}(2 - m_x + t)2^{-E_x}. \quad (24)$$

Relative errors in this case will be the same as well for $x \in [1, 2)$, and all performed calculations will be valid for negative values if the sign is taken into account.

Proposed algorithm

To provide full accuracy in single-precision format, the first iteration should be executed using a modification of (12):

$$y_1 = k_1 y_0 (2 + k_2 - x y_0). \quad (25)$$

The second iteration uses the classic formula

$$y_2 = y_1 (2 - x y_1). \quad (26)$$

The classic Newton–Raphson formula gives an exact result if an exact enough initial approximation y_0 is used in it. (The formula has quadratic convergence—the number of correct bits is doubled.) If the initial approximation is inexact (i.e., as in our case), then it is possible to modify a classic formula similar to (25) (in our case, $k_1 = 2$) to get the same exact result, as well as in the case described previously.

It is necessary to find optimal values of the magic constant R; its unknown parameter T, which is determined from $t - (14)$; and the coefficient k_2. Although the coefficient k_1 could also be optimized to minimize the error of the approximation similarly as for variables t and k_2, we have chosen to set $k_1 = 2$ because the multiplication by two can be efficiently implemented by addition or by increasing the exponent, without the need for floating-point multiplication. The following method should be used. After the first Newton–Raphson iterations with linear initial approximation y_0, the variable y_1 is described by a polynomial of the third order. To provide the smallest error, y_1 should be the best uniform approximation polynomial (in the sense of relative error).

The best uniform approximation for the reciprocal is the polynomial of the third order, P_3, in the form

$$P_3 = \alpha x^3 + \beta x^2 + \gamma x + \Psi, \quad (27)$$

where $x \in [a, b]$

$$\alpha = -\frac{128}{q}, \quad \beta = \frac{256(a + b)}{q},$$

$$\gamma = -\frac{32(5a^2 + 14ab + 5b^2)}{q},$$

$$\Psi = \frac{32(7ab^2 + 7a^2 b + a^3 + b^3)}{q},$$

and

$$q = a^4 + b^4 + ab(28(a^2 + b^2) + 70ab).$$

We obtained this polynomial by performing a minimization of the error, which will be detailed in a separate article. Because y_0 has two constituents, y_{01} and y_{02}, it is necessary to select values R and k_2 so that the total maximum relative error has a minimum value throughout the interval (1, 2). It is necessary to write (25) and (27) using the conditions

$$y_{11} = 2y_{01}(2 + k_{21} - x y_{01}),$$

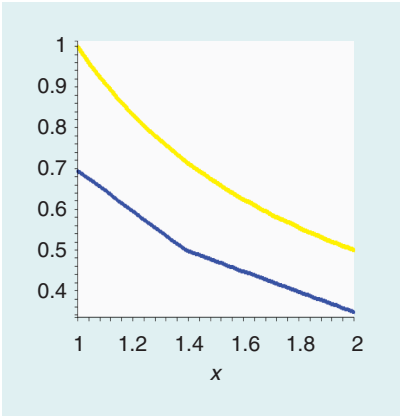


FIGURE 3. Initial approximations of rcp_3 algorithm (blue). The yellow line indicates 1/*x*.

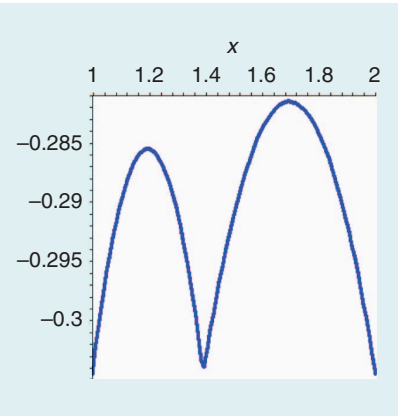


FIGURE 4. Relative errors of initial approximations for rcp_3 algorithm.

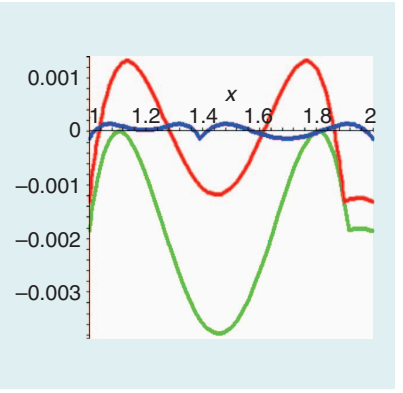


FIGURE 5. Relative errors of the first iteration for the rcp_1 (green), rcp_2 (red), and rcp_3 (blue) algorithms.

$$P_{31},\ x \in [a = 1, b = 1 + t],$$

and

$$y_{12} = 2y_{02}(2 + k_{22} - xy_{02}),$$
$$P_{32},\ x \in [a = 1 + t, b = 2].$$

Solving the two equations, $y_{11} = P_{31}$ and $y_{12} = P_{32}$, will yield the two pairs of meaning for t and k_2. The details are omitted here because of space limitations.

It can be shown that $t = 0.3911389\,72948632956$ ensures the lowest possible value of relative errors after the first iteration carried out by (25). With this the value t, the magic constant R for float numbers, will take the form 0x7eb210d8, and $k_2 = -0.585691542659989199$. Then, after rounding, (25) becomes

$$y_1 = 2y_0(1.41430846 - xy_0).$$

C code of the proposed algorithm

```
float rcp_3(float x)
{
  int i=*(int*)&x;
  i=0x7eb210d8-i;
  float y=*(float*)&i;
  y=y*(1.41430846f-x*y);
  y=y+y;
  float r=1.0f-x*y;
  y=y+r*y;
  return y;
}
```

The multiplication operation in line 2 is replaced by an addition operation. Figures 3 and 4 show graphics of the initial approximations y_{01}, y_{02}, and $1/x$ and the relative errors of the suggested algorithm accordingly. The large bias visible in Figure 3 is eliminated by the modified Newton–Raphson iteration (25) without additional mathematical operations. The maximum relative errors are $\delta^{+}_{\max} = 1.169 \cdot 10^{-7}$, $\delta^{-}_{\max} = -1.344 \cdot 10^{-7}$, or 22.89 correct bits. In this variant of the algorithm (rcp_3), the relative error for the first iteration has a minimum value ($\delta^{+}_{\max} = 1.341 \cdot 10^{-4}$ and $\delta^{-}_{\max} = -1.345 \cdot 10^{-4}$, or 12.86 correct bits). Relative errors of the first iteration for the three described algorithms are shown in Figure 5.

However, the most accurate results are given by the following implementation of the algorithm:

```
float rcp_4(float x)
{
  int i=*(int*)&x;
  i=0x7eb210da-i;
  float y=*(float)&i;
  y=y*(1.4143113f-x*y);
  y=y+y;
  float r=fmaf(y,-x,1.0f);
  y=fmaf(y,r,y);
  return y;
}
```

The second iteration is performed using the operation $fmaf(y, -x, 1.0f)$, that is equivalent to $1.0f - xy$. (The *fmaf* operation reduces errors due to use of the formula realization, e.g., in $z = y + r \cdot y$, the rounding is executed only once to receive the final result.) The *fmaf* use gives an opportunity to reduce the maximum relative errors of both signs to values $\delta^{+}_{\max} = 5.895 \cdot 10^{-8}$ and $\delta^{-}_{\max} = -7.608 \cdot 10^{-8}$, or 23.64 correct bits. In hardware implementation, multiplying by two makes it possible to use the second integer constant 0x7f3210da, which differs from the magic constant on 0x00800000 (i.e., the exponent increased by one).

The C code for hardware implementation of this algorithm in FPGA is

```
float rcp_5(float x)
{
  int i=*(int*)&x;
  int ii=0x7f3210da-i;
  i=0x7eb210da-i;
  float y=*(float*)&i;
  float yy=*(float*)&ii;
  y=yy*(1.4143113f-x*y);
  float r=fmaf(y,-x,1.0f);
  y=fmaf(y,r,y);
  return y;
}
```

The calculation of integer variables *i* and *ii* and their conversion in float can be conducted simultaneously in this case, and that algorithm includes only four multiplication operations without lookup table for initial approximation. The C code for double precision is

```
double rcp_6(double x)
{
  uint64_t i=*(uint64_
  t*)&x;
  uint64_t ii=0x7fe6421af
  0901626-i;
```

```
    i=0x7fd6421af0901626-i;
    double y=*(double*)&i;
    double yy=*(double*)&ii;
    y=yy*(1.4143084573400108
    -x*y);
    y=y*(2.0000000090062634
    -x*y);
    y=y+y*fma(y,-x,1.0);
    return y;
}
```

The algorithm has maximal relative errors in the whole range of normalized double-type numbers, $\delta^{+}_{\max} = 0$: $\delta^{-}_{\max} = -2.22 \cdot 10^{-16}$, or 52 correct bits.

Technical specifications of the computer running the experiments are as follows: Windows 7 64-bit operating system, Intel Pentium G850 with 2.90-GHz CPU, 4-GB memory using MS Visual C++ Compiler 12.0.

Experimental results for the microcontroller

Our reciprocal algorithm was tested in C++ on the Espressif (ESP) WROOM-32 microcontroller [10], which supports floating-point computing (addition, multiplication, and fma single-precision instructions) and does not have hardware reciprocal instructions. Relative errors and latency are shown in Table 1. The reciprocal algorithm for the single precision is 21% faster than usual float division 1.0f/x. If we conduct the first iteration as

$$y_1 = 2y_0(1.4143113 - xy_0) \qquad (28)$$

(see rcp_4_28 algorithm), the advantage will grow to 32%.

For the ESP microcontroller WROOM-32, multiplying by two (or number aliquot of two) occurs faster than any other real number that is not aliquot to two.

```
float rcp_4_28(float x)
{
  int i=*(int*)&x;
  i=0x7eb210da-i;
  float y=*(float)&i;
  y=2*y*(1.4143113f-x*y);
  float r=fmaf(y,-x,1.0f);
  y=fmaf(y,r,y);
  return y;
}
```

Table 1. The performance of reciprocal algorithms on ESP WROOM-32 microcontroller.

Single Precision	1.0f/x	Our rcp_4	Our rcp_4, (28)
$\delta^{-}_{\max}$	$-5.9558602 \times 10^{-8}$	$-7.6075395 \times 10^{-8}$	$-7.6075395 \times 10^{-8}$
$\delta^{+}_{\max}$	$+5.9604638 \times 10^{-8}$	$+5.8947094 \times 10^{-8}$	$+5.8947094 \times 10^{-8}$
Latency (ns)	327.31399	268.48363	255.90167

Summary

We have described simple floating-point division, which is based on the reciprocal of the divisor, and presented reciprocal algorithms with four multiplications for single precision and six multiplications for double precision. The proposed algorithms belong to the iterative algorithms group. They initially receive an approximation using the so-called magic constant and are then used to implement an iterative process with the Newton–Raphson formula. Improved accuracy of the proposed algorithms compared with known algorithms was achieved by reducing the magnitude scope of the relative errors of the initial approximations (the difference between the smallest and largest values) and, as a consequence, the modification of the first iterations. For all normalized floating-point numbers, both algorithms have maximal relative errors of $\delta_{\max} = 7.61 \cdot 10^{-8}$ and $\delta_{\max} = -2.22 \cdot 10^{-16}$ accordingly.

Authors

Leonid Moroz (moroz_lv@lp.edu.ua) received his Sc.D. degree in technical sciences from Lviv Polytechnic Institute, Ukraine, in 1978. He is a professor in the Department of Information Security, Institute of Computer Technologies, Automation and Metrology, Lviv Polytechnic National University, Ukraine. His research interests include computer arithmetic, numerical methods, and digital signal processing.

Volodymyr Samotyy (vsamotyy@pk.edu.pl) received his Sc.D. degree in technical sciences from Lviv Polytechnic Institute, Ukraine, in 1984. He is a professor in the Department of Automatic Control and Information Technology, Cracow University of Technology, Poland, and the Department of Information Security Management, Lviv State University of Life Safety, Ukraine. His research interests include evolutionary models, parametric optimization, information security, computer arithmetic, numerical methods, and digital signal processing.

References

[1] M. Allie and R. Lyons, "A root of less evil," *IEEE Signal Process. Mag.*, vol. 22, no. 2, pp. 93–96, 2005. doi: 10.1109/MSP.2005.1406500.

[2] F. Auger, Z. Lou, B. Feuvrie, and F. Li, "Multiplier-free divide, square root, and log algorithms," *IEEE Signal Process. Mag.*, vol. 28, no. 4, pp. 122–126, 2011.

[3] STMicroelectronics. (2016, May). Floating point unit demonstration on STM32 microcontrollers. Application note AN4044. [Online]. Available: https://www.st.com/content/ccc/resource/technical/document/application_note/10/6b/dc/ea/5b/6e/47/46/DM00047230.pdf/files/DM00047230.pdf/jcr:content/translations/en.DM00047230.pdf

[4] Texas Instruments. (2015, Oct.) TMS320F2807x Piccolo™ microcontrollers. SPRS902B. [Online]. Available: http://www.ti.com/lit/ds/symlink/tms320f28075.pdf

[5] Intel. (2017, 8 May). Intel Cyclone 10 GX device Overview. C10GX51001. [Online]. Available: https://www.intel.com/content/www/us/en/programmable/documentation/grc1488182989852.html

[6] J. Blinn, "Floating-point tricks," *IEEE Comput. Graph. Appl. Mag.*, vol. 17, no. 4, pp. 80–84, 1997. doi: 10.1109/38.595279.

[7] K. Huang and Y. Chen, "Improving performance of floating point division on GPU and MIC," in *Proc. 15th Int. Conf. Algorithms and Architectures for Parallel Processing Part II*, 2015, vol. 9529, pp. 691–703.

[8] P. Khuong. (2011, Mar. 16). A magic constant for double float reciprocal. [Online]. Available: https://www.pvk.ca/Blog/LowLevel/software-reciprocal.html

[9] L. Moroz and A. Hrynchyshyn, "A fast calculation of function y=1/x with the use of magic constant," *Bull. Lviv Polytech. Nat. Univ.* (Automation, Measurement and Control Series 821, in Ukrainian), pp. 23–29, 2015.

[10] Espressif Systems. (2018). ESP32-WROOM-32 (ESP-WROOM-32) datasheet. Version 2.4. [Online]. Available: https://www.mouser.com/ds/2/891/esp-wroom-32_datasheet_en-1223836.pdf

[11] IEEE Computer Society. (1985). IEEE Standard for Binary Floating-Point Arithmetic. [Online]. Available: https://standards.ieee.org/standard/754-2008

SP

Arash Mohammadi, Parnian Afshar, Amir Asif, Keyvan Farahani,
Justin Kirby, Anastasia Oikonomou, and Konstantinos N. Plataniotis

Lung Cancer Radiomics

Highlights from the IEEE Video and Image Processing Cup 2018 Student Competition

The volume, variety, and velocity of medical imaging data are exploding, making it impractical for clinicians to properly utilize such available information resources in an efficient fashion. At the same time, the interpretation of such a large amount of medical imaging data by humans is significantly error prone, reducing the possibility of extracting informative data. The ability to process such large amounts of data promises to decipher encrypted information within medical images, develop predictive and prognosis models to design personalized diagnosis, allow comprehensive study of tumor phenotype, and allow the assessment of tissue heterogeneity for diagnosis of different types of cancers.

Recently, there has been a great surge of interest in *radiomics* [1], which refers to the process of extracting and analyzing several semiquantitative and quantitative features from medical images, with the ultimate goal of obtaining predictive or prognostic models. Radiomic features can be extracted from different imaging modalities, including magnetic resonance imaging, positron emission tomography, and computed tomography (CT)—which is the most commonly used modality for lung cancer radiomics due to its imaging sensitivity, high resolution, and isotropic acquisition in locating lung lesions.

The 2018 Video and Image Processing (VIP) Cup is a student competition sponsored by the IEEE Signal Processing Society (SPS). Each participating team is required to be composed of 1) one faculty member (the supervisor), 2) at most, one graduate student (the mentor), and 3) at least three, but no more than ten, undergraduate students. Participation in the VIP Cup is open to all teams from around the world that satisfy the aforementioned eligibility criteria. The top three finalist teams were selected to present and compete at the final stage of the competition, which was held at the 2018 IEEE International Conference on Image Processing (ICIP) in Athens, Greece, on 7 October 2018; see "Winners of the 2018 VIP Cup" for details. In the remainder of this article, we will share an overview of the 2018 VIP Cup experience, including competition setup, technical approaches, statistics, and competition experience from the point of view of the organizers and members of the finalist teams.

The 2018 VIP Cup challenge

A radiomic workflow [1] typically consists of four main processing tasks: 1) image acquisition, 2) image segmentation, 3) feature extraction and qualification, and 4) statistical analysis and model construction. Segmentation is considered a critical step among processing tasks within the radiomic pipeline, and it was the focus of the 2018 VIP Cup competition. More specifically, the main task of the 2018 VIP Cup was "Segmentation of Tumor Region in Lung CT Images." Segmentation (i.e., assigning a class label to each pixel) and localization (i.e., providing the tumor bounding box) are critical steps within the radiomic workflow, as radiomic features are extracted from the segmented sections. Although manual delineation of the gross tumor is the conventional (standard) clinical approach, it is time-consuming and extremely sensitive to interobserver variability. Development of accurate and robust automatic segmentation methods, to minimize manual error and increase consistency of delineating regions, was of paramount importance and the focus of the competition.

Competition data set

The 2018 VIP Cup data set consisted of images from nonsmall cell lung cancer (NSCLC) subjects provided by Dr. Andre Dekker and Dr. Leonard Wee from the Maastricht Radiation Oncology (MAASTRO) Clinic. The introduced data set was an updated and modified version of the NSCLC-Radiomics data set [2], [3], available at the Cancer Imaging Archive [4]. In particular, the new data set consisted of improved and extended annotations and also contained missing annotations for the 108 subjects in the original release, which were used for evaluation purposes as the unseen data set.

The initial training data released on 26 May 2018 consisted of pretreatment CT scans from the NSCLC-Radiomics data set [2], [3] divided into two categories: 1) 100 subjects with all slices, including

Digital Object Identifier 10.1109/MSP.2018.2877123
Date of publication: 24 December 2018

 1053-5888/19©2019IEEE

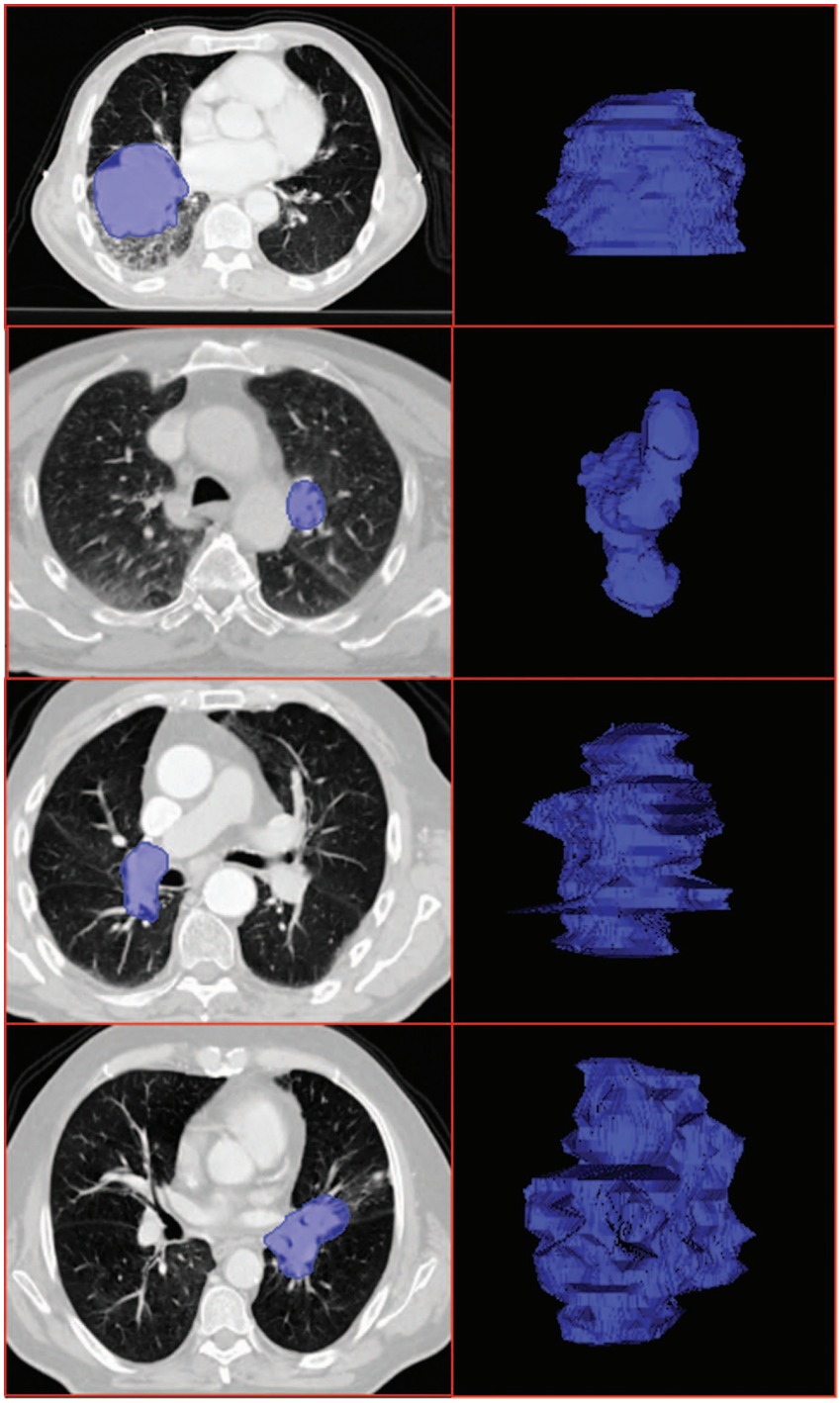


FIGURE 1. An example of CT images of lung cancer patients and CT images with tumor contours created from the competition's data set using 3DSlicer software.

nontumor ones, and 2) a selected set of 60 subjects with tumor-only slices. Each data set contained manual delineations of the gross tumor volume provided by a radiologist. Figure 1 illustrates an example of CT images of lung cancer patients with tumor contours. The final data sets that were provided for training, validation, and test purposes were from the newly introduced data, briefly outlined here.

- *Training data set*: The 260 subjects for training purposes. The labels (RTStruct files) for these patients were provided.
- *Validation data set*: The 40 subjects for validation purposes. The labels (RTStruct files) for these patients were provided; therefore, each team could evaluate its algorithms and report the results. Each team provided evaluation results based on these 40 subjects.
- *Test data set*: The 40 subjects used for test purposes. The labels (RTStruct files) for these patients were not provided. For the 40 test subjects, as the true labels were not publicly available, each team provided the segmentation results in a binary image format.

Submission guidelines

For the final competition stage, each participating team was required to submit a competition package consisting of four main items, as specified in Figure 2 and briefly outlined here:

- *Code executable*: The executable file was prepared based on the following instructions: 1) take as an input the path to the location of all patient folders to be analyzed; 2) if the patient folders included associated labels (RTStruct files), then the output of the executable should be a file containing all results (as specified below); 3) if the provided patient folders did not have any labels (RTStruct files), then the executable should return a folder containing all patient folders, where, in each folder, segmentation results were presented as binary images.
- *Reporting results*: Results submitted in a comma-separated-values (CSVs) file provided the final evaluation results based on the three identified criteria [i.e., (1), (3), and (5)] over the validation data set (i.e., the 40 subjects for which the labels (RTStruct files) were provided). The CSV file included in the submission package was prepared based on the following specifications: 1) the CSV file needed to contain the results for all 40 subjects based on the three identified criteria, and 2) for each subject, the CSV file needed to report the average value of each of the criteria for all of the patients' slices.
- *Segmentation files*: This item provided the segmentation results for the test data set [i.e., the 40 subjects for which the labels (RTStruct files)

First Place: Team Markovian

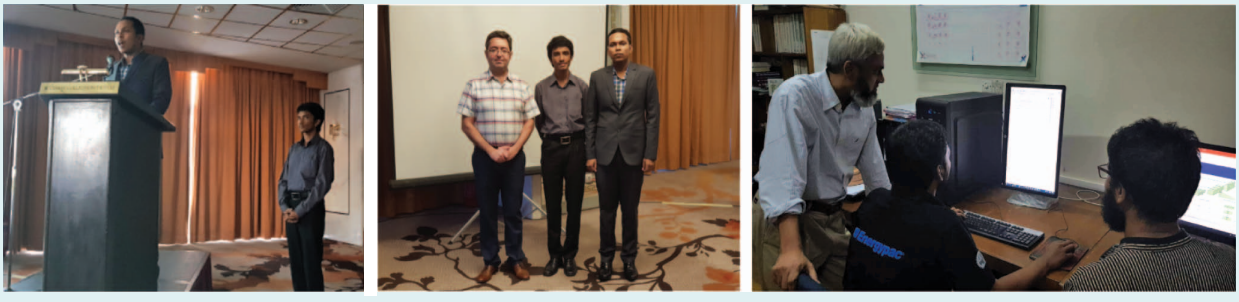
- Bangladesh University of Engineering and Technology
- Undergraduate students: Shahruk Hossain, Zaowad Rahabin Abdullah, A.K.M. Naziul Haque, Fahim Hafiz, Farhan Shadiq, Monayem Hassan, Mushfiqur Rahman, Tariqul Islam, Muhammad Suhail Najeeb, and Asif Shahriyar
- Supervisor: Mohammad Ariful Haque
- Technical approach: Team Markovian (Figure S1) developed a pipeline involving a binary classifier front end that determines the cancerous slices. The slices identified as containing tumors are then passed to a segmentation model based on a two-dimensional, fully convolutional neural network using dilated convolutions, instead of pooling, to generate segmentation masks for the tumor. Finally, the predicted masks are passed through a postprocessing block that cleans up the masks through morphological operations.

First Runner-Up: Team Spectrum

- Bangladesh University of Engineering and Technology
- Undergraduate students: Uday Kamal, Abdul Muntakim Raf, and Rakibul Hoque
- Supervisor: Kamrul Hasan

(a) (b)

FIGURE S1. First-Place Team Markovian (a) presenting at the 2018 IEEE International Conference on Image Processing (ICIP) final competition on 7 October 2018. (b) Dr. Arash Mohammadi (left), IEEE Signal Processing Society director of membership services and organizer of the 2018 VIP Cup, stands with Asif Shahriyar a member of the team.



(a) (b) (c)

FIGURE S2. First Runner-Up Team Spectrum (a) presenting at the ICIP 2018 final competition on 7 October 2018. (b) (from left) Dr. Arash Mohammadi (left)stands with members of Team Spectrum, Uday Kamal and Abdul Muntakim Raf. (c) A behind-the-scenes look: (from left) Dr. Kamrul Hasan, Abdul Muntakim, and Rakibul Hoque.

- Technical approach: Team Spectrum (Figure S2) developed a framework that is based on the recurrent 3D-DenseUNet, i.e., a new fusion of convolutional and recurrent neural networks. The developed approach is to train the network using image volumes with tumor-only slices. A data-driven adaptive weighting method is also used to differentiate between tumorous and non-tumorous image slices, which shows more promise than crude intensity thresholding. In other words, adaptive thresholding is used to generate binary images, and then morphological dilation is applied in postprocessing to overcome the random missing pixel and to enlarge the boundary.

Second Runner-Up: Team NTU-MiRA

- National Cheng Kung University and National Taiwan University
- Undergraduate students: Chun-Ting Wu, Chao-Te Chou, Min-Sheng Wu, Jhih-Yuan Lin, Yun-Chun Chen, and Yu-Cheng Chang
- Supervisor: Winston Hsu
- Technical approach: Team NTU-MiRA (Figure S3) developed an end-to-end trainable data-driven solution that learns to predict volumetric segmentation outputs. To handle the imbalance distribution between the foreground and background regions, the dice-loss function and the focal loss are used for optimization.

FIGURE S3. (a) The team members of Second Runner-Up Team NTU-MiTRA, (from left) Chih-Yuan Lin and Yu-Cheng Chang, presenting at the ICIP 2018 final competition on 7 October 2018. (b) (from left) Dr. Arash Mohammadi with members of the team: Yun-Chun Chen, Chao-Te Chou, Yu-Cheng Chang, Min-Sheng Wu, and Chih-Yuan Lin. (c) The team members at the event. (d) Various team members with the jury members: (from left) Abdul Muntakim (Team Spectrum), Uday Kamal (Team Spectrum), Dr. Lucio Marcenaro, Asif Shahriyar (Team Markovian), Parnian Afshar, Dr. Farnoosh Naderkhani, Dr. Arash Mohammadi, Chih-Yuan Lin (Team NTU-MiRA), Min-Sheng Wu (Team NTU-MiRA), Yu-Cheng Chang (Team NTU-MiRA), Chao-Te Chou (Team NTU-MiRA), and Yun-Chun Chen (Team NTU-MiRA).

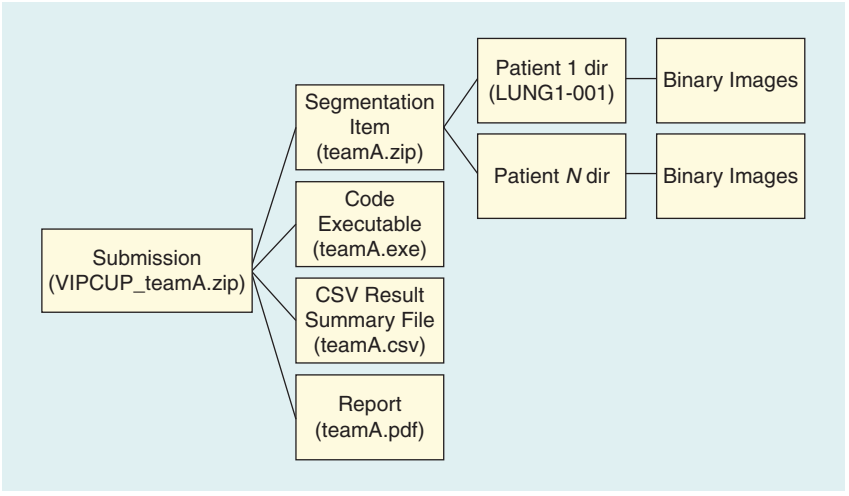


FIGURE 2. Different items to be included in the final submission package.

were not provided]. This segmentation file was prepared based on the following specifications: 1) the file must contain the binary images in .png format and must be of the same resolution (size) as that of the input images, and 2) each subfolder within the segmentation folder needed to have binary images for all of the slices provided for that specific patient.

- *Report*: The report should be prepared in the IEEE format providing: 1) required background; 2) incorporated signal processing and preprocessing algorithms; 3) information regarding the algorithms used to perform the segmentation tasks; 4) explicit details on other data sources, trained models, and/or similar items (if any) that was/were used for training purposes; 5) potential novelties of the processing algorithms; and 6) presentation and discussion of the evaluation results.

The 2018 VIP Cup started with a global engagement of more than 600 users from 42 countries to access competition data from all around the world.

Evaluation schemes

The segmented contours computed by the participating teams were compared against the manual contours for all validation and test images based on the following three evaluation criteria.

Dice coefficient

The dice coefficient is a measure of relative overlap (1 represents perfect agreement and 0 represents no overlap), computed as

$$\text{Evaluation Criteria 1: } D = \frac{2|X \cap Y|}{|X| + |Y|}, \tag{1}$$

where $\cap$ denotes the intersection operator, and X and Y are the ground truth and test regions, respectively. Dice coefficient (D) has a restricted range of [0, 1]. The following two conventions are recommended for computation of the dice coefficient: 1) for true-negative (i.e., there is no tumor and the processing algorithm correctly detected the absence of the tumor), the dice coefficient would be 1; and 2) for false-positive (i.e., there is no tumor but the processing algorithm mistakenly segmented the tumor), the dice coefficient would be 0.

Mean-surface distance

The directed average Hausdorff measure is the average distance of a point in X to its closest point in Y:

$$\vec{d}_{H,\text{avg}}(X,Y) = \frac{1}{|X|}\sum_{x\in X}\min_{y\in Y} d(x,y). \tag{2}$$

The (undirected) average Hausdorff measure is the average of the two directed average Hausdorff measures given by

$$\text{Evaluation Criteria 2: } d_{H,\text{avg}}(X,Y) = \frac{\vec{d}_{H,\text{avg}}(X,Y) + \vec{d}_{H,\text{avg}}(Y,X)}{2}. \tag{3}$$

Hausdorff distance (95% Hausdorff distance)

The directed percent Hausdorff measure for a percentile r is the rth percentile distance over all distances from points in X to their closest points in Y, e.g., the directed 95% Hausdorff distance is the point in X whose distance to its closest point in Y is greater or equal to exactly 95% of the other points in X. In mathematical terms, denoting the rth percentile as K_r,Kr, this is given by

$$\vec{d}_{H,r}(X,Y) = K_r\left(\min_{y\in Y} d(x,y)\right)\forall x \in X. \tag{4}$$

The (undirected) percent Hausdorff measure is defined again with the mean:

$$\text{Evaluation Criteria 3: } \vec{d}_{H,r}(X,Y) = \frac{\vec{d}_{H,r}(X,Y) + \vec{d}_{H,r}(Y,X)}{2}. \tag{5}$$

The 2018 VIP Cup statistics

The 2018 VIP Cup started with a global engagement of more than 600 users from 42 countries (Figure 3) to access competition data from all around the world (Figure 4). At the start, the highest engagement was from Bangladesh, Canada, India, and the United States. At the registration stage, there were 129 members clustered into 28 teams from ten countries, including Australia, Bangladesh (11 teams), Canada, China, Hong Kong, India (three teams), Iran (two teams), Greece, Taiwan, and the United States. Out of these 28 teams, nine teams, with a total of 56 members, from Australia, Bangladesh (three teams), India (two teams), Iran, Hong Kong, and Taiwan, made it to the final stage, from which six teams, that submitted all of the

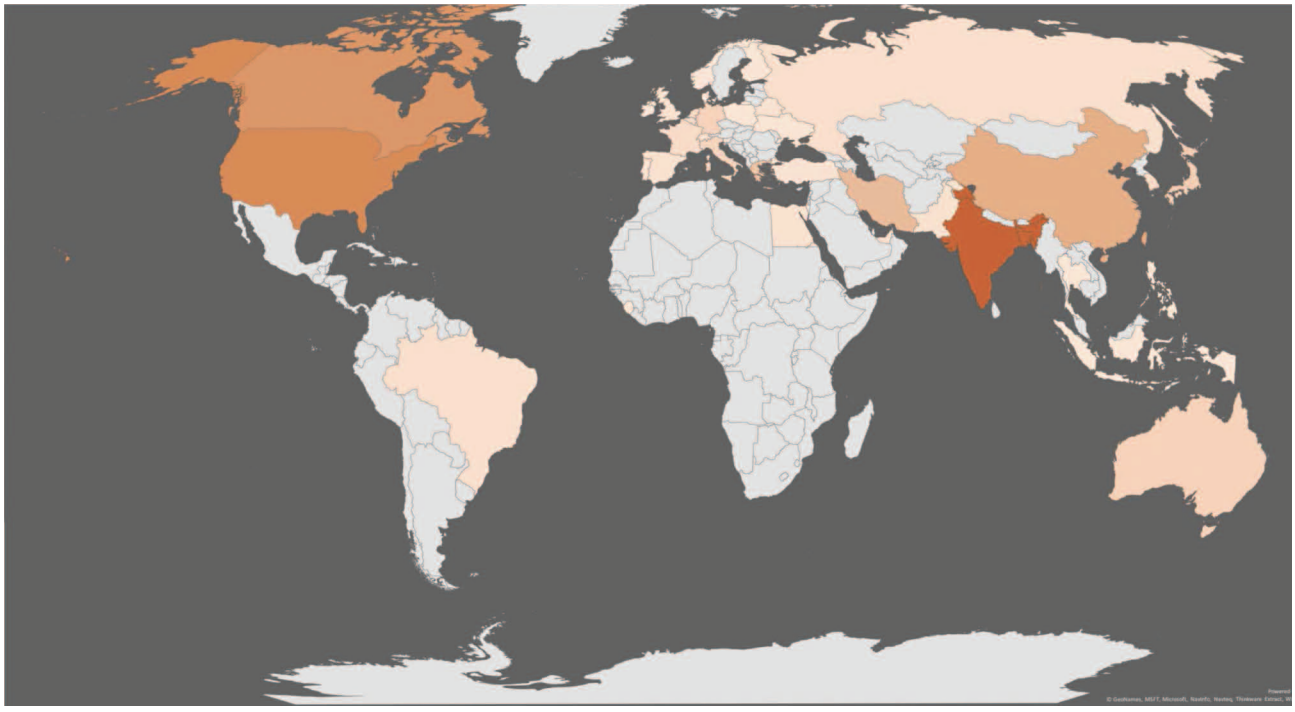


FIGURE 3. A global engagement map of the 2018 VIP Cup.

required components, were evaluated to select the three finalist teams.

Final stage of the competition

The 2018 VIP Cup organizers evaluated the submissions and announced the three finalist teams on 10 September 2018: Markovian, NTU-MiRA, and Spectrum. Evaluation was based on the results computed via the aforementioned criteria [(1)–(5)] over the validation and test data sets. Finalist teams were invited to the final competition stage at ICIP 2018, which was held on 7 October 2018. Prior to the final competition and as the final challenge, the organizers provided a list identifying the tumor-containing slices in the test data for the three finalists to apply their algorithms and provide the binary files for the tumor-only slices. The results were used as part of the ranking procedure of the top three teams.

During the final competition on 7 October, the finalist teams presented their work. Each team had 15 min for its presentation and 5 min for questions and answers. After the team presentations, the jury had an internal discussion to finalize the ranking. The jury included Dr. Farnoosh Naderkhani, Parnian Afshar, Dr. Lucio Marcenaro, and Dr. Arash Mohammadi. At the ICIP 2018 Student Career Luncheon on Monday, 8 October, Dr. Marcenaro, SPS's director of student services, highlighted the second edition of the VIP Cup and Dr. Mohammadi, organizer of the 2018 VIP Cup and SPS's director of membership services, publicly announced the winners of the competition.

Highlights of technical approaches

All teams were composed of mainly undergraduate students supervised by a

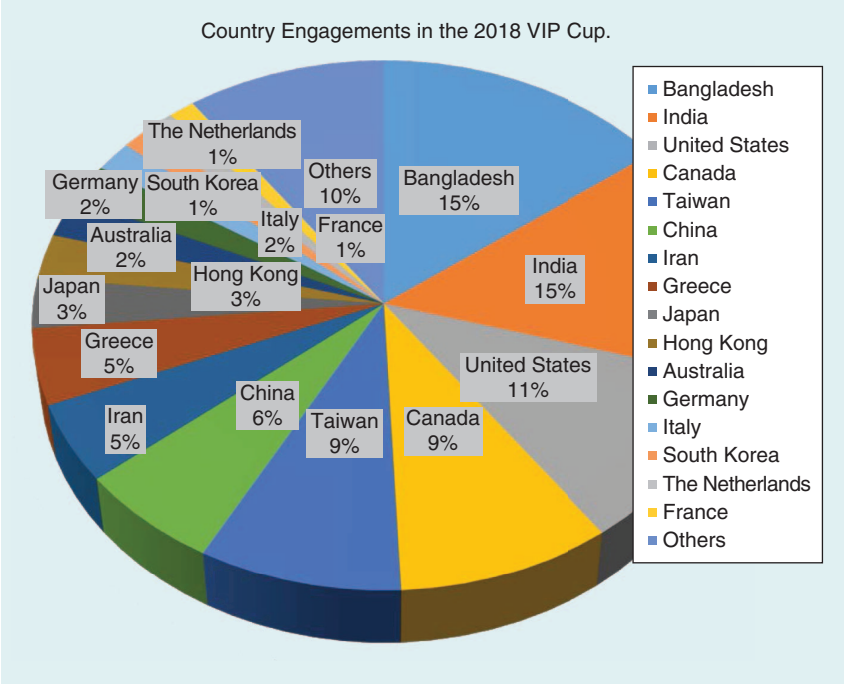


FIGURE 4. Country engagements in the 2018 VIP Cup.

single faculty advisor and, at most, one graduate student mentor. Although different classical segmentation algorithms, such as thresholding methods, histogram-based methodologies, morphological techniques, and clustering approaches, were investigated by the participants, deep neural networks constitute the main building block of the processing pipeline developed by the finalists. Several different deep architectures were studied and implemented from encoder–decoder style models, such as the U-Net [8], ResNet-like architectures [10], DenseNet architectures, [9] and, finally, LungNet architecture [7], which was proposed in March 2018. A very interesting aspect of the processing techniques implemented by the finalists is that all three teams went one step ahead of what is currently introduced in the literature and developed and designed hybrid and innovative architectures.

The 2018 VIP Cup organizers evaluated the submissions and announced the three finalist teams on 10 September 2018: Markovian, NTU-MiRA, and Spectrum.

In particular, Team Markovian developed a pipeline involving a binary classifier front end that detects the tumorous slices, as the first step. The identified slices containing the tumor are then passed to a segmentation model based on a two-dimensional, fully convolutional neural network, using dilated convolutions instead of pooling, to generate segmentation masks for the tumor. Finally, the predicted masks are passed through a postprocessing block, which cleans them up through morphological operations. The average dice coefficient of 0.627 over the validation data set is reported. The average dice coefficient of 0.594 over the test set is computed. Team Spectrum's implemented framework is based on the recurrent 3D-DenseUNet and a novel fusion of convolutional and recurrent neural networks that are introduced and incorporated for performing the competition tasks.

The developed approach is to train the network using image volumes with tumor-only slices of size ($256 \times 256 \times 8$ pixels). A data-driven adaptive weighting method is then used to differentiate between tumorous and nontumorous image slices, which shows more promise than crude intensity thresholding. In other words, adaptive thresholding is used to generate binary images, and then morphological dilation is applied as postprocessing to overcome the random missing pixel and to enlarge the boundary.

The average dice coefficient of 0.74 over the validation data set is reported. The average dice coefficient of 0.521 over the test set is computed. Team NTU-MiRA developed an end-to-end trainable network that performs lung cancer tumor segmentation in a supervised fashion. Inspired by the dense up-sampling layer proposed in [11], the dense down-sampling convolutional layer is incorporated to reduce the computational burden. Dense lateral connections are introduced into the proposed model, which facilitates the network training and results in faster convergence. Finally, to handle the imbalance distribution between the foreground and background region, the dice-loss function and the focal loss are used for optimization.

Organizers' opinions

It was a demanding but intriguing journey to organize the 2018 VIP Cup. It was during the 2018 IEEE International Conference on Acoustics, Speech, and Signal Processing, held 15–20 April 2018, that we had to coordinate this year's VIP Cup and, given the limited time we had, the early stage was very challenging. Specifically, the initial issue was to find unseen data for test purposes due to the common difficulties found in acquiring and distributing labeled medical images. During the course of the competition, and especially during the initial phase, organizing members had weekly conference calls to address the aforementioned issue and discuss different aspects of the competition. It was also very inspiring to meet the finalists during ICIP 2018. Moreover, we were greatly impressed and encouraged during the course of the competition and, at the final stage, by the high technical level of implemented processing solutions developed, and even proposed, by the competitors and extensive dedication of the participating teams.

Participants' opinions

Throughout the 2018 VIP Cup competition, there was a great deal of interaction, not only through questions for the instructors posted to Piazza but also among the students who often engaged in discussion over the provided responses. We, as organizers, were both encouraged and delighted to observe such a collaborative spirit among the participating students. Next, we will provide an overview of some feedback and perspectives received from the winning teams.

Team Markovian

- "Thanks to the IEEE SPS for organizing this wonderful competition regularly for the undergraduate students. We are learning a lot from these events."

  —Dr. Mohammad Ariful Haque, supervisor

- "It was an amazing opportunity to participate in the IEEE VIP Cup for the second time; the first time was in 2017, where our team made it to second runner-up. We learned from our previous experience and with the guidance of our supervisor, Dr. Mohammad Ariful Haque, we were able to capture the champion title this time. Knowing that the task at hand had real-world applications and could potentially help people around the world was a great motivating factor for me. Through the VIP Cup, my understanding of neural networks, segmentation problems, and biomedical image processing in general have grown quite a lot. I also learned a lot about teamwork, and what it takes to get things done under deadlines. Some of us will be graduating this year,

but I am sure that Team Markovian will be rejuvenated with new members and come back stronger in the next VIP Cup!"

—Shahruk Hossain, undergraduate

- "Being able to take part in the VIP Cup for the second time for Team Markovian has been a great experience. This challenge allowed us to dig deeper into the emerging world of biomedical image processing. The new challenges, the short time frame, and tight deadlines demanded immense patience and relentless work. It's great to see all the efforts pay off."

—Suhail Najeeb, undergraduate

- "This was my first VIP Cup, and it really helped me a lot in understanding neural networks. The problem was challenging, and I learned a great deal about using ML libraries in Python. The teamwork I experienced was really something else. In the future, I wish to work on biomedical image processing. The skills I improved by participating in the 2018 VIP Cup certainly boosted my confidence in many ways."

—Zaowad Rahabin Abdullah, undergraduate

- "My VIP Cup experience was a very exciting one as I arrived at Greece only three hours before the presentation. I was not sure whether I was going to make it or not. Special thanks to my fellow teammate, Suhail, who prepared some comprehensive slides so that I could prepare for the presentation within one hour. Also thanks to Shahruk for joining me during the questions and answers session. While presenting, I just kept in mind that I had to explain to the judges the pros and cons of our model and illustrate its effectiveness. But I was very uncertain about my own presentation due to not getting enough time for preparing. However, when the results were published, I couldn't believe it. All our efforts were worth it, I felt. It seems that the simplest solution often performs the best."

—Asif Shahriyar Sushmit, undergraduate

Team Spectrum

- "This was my second time participation in the IEEE VIP Cup. The problem itself was very challenging. To train a robust model from scratch using a very limited amount of data was the toughest part of the competition. I gained some valuable experience and learned a lot by working on the challenge. Above all, I would like to thank everyone, especially the organizers, our respected supervisor, and my parents for their constant help and support."

It was a demanding but intriguing journey to organize the 2018 VIP Cup.

—Uday Kamal, undergraduate

- "What I learned from this year's IEEE VIP Cup is that, eventually, it comes down to the robustness of the whole pipeline. We were provided separate training, validation, and test data in the competition. We trained our model with the training data only. But while selecting a threshold for the output probabilities of the model, we used the validation data to choose the best threshold. This somehow overfit our results, to some degree, on the validation data, and we couldn't attain our expected dice scores in the test data. This was a great experience for me. I would like to thank the organizers for this thrilling problem and exceptional data set."

—Abdul Muntakim Rafi, undergraduate

- "It was an awesome experience. I am glad that parts of my work have been presented before the world through this competition. This challenging competition has helped me to develop a research-oriented mindset under our respected supervisor, Prof. Kamrul Hasan Sir. Also, the organizers this year were very friendly and helpful as well. Last of all, I would like to thank my friends and family who constantly supported me, and my team members for their dedication and hard work."

—Rakibul Hoque, undergraduate

Team NTU-MiRA

- "This was my first time participating in an international competition. I would like to thank the organizers for providing such a challenging stage for us. I have learned many techniques about deep learning and medical images. I am also grateful to our supervisor and my teammates for overcoming many difficulties together. I believe that this experience can make us grow. I wish for the 2019 VIP Cup to be as delightful and successful."

—Jhih-Yuan Lin, undergraduate

- "Accomplishing this challenging tumor segmentation task was a milestone for me. I learned lots of deep-learning technologies about 3-D medical image processing from the 2018 VIP Cup and tried all my best to overcome each problem, such as the position and scale variants, the data imbalance, and the limitation of the hardware resources, building an accurate, robust, and efficient model. On top of that, the experience of sharing our efforts on an international competition was the most precious thing. I'm very glad that we won third place. Last, I want to thank our supervisor, Prof. Winston H. Hsu, my teammates, and all the friends who have helped us."

—Min-Sheng Wu, undergraduate

- "It was an unforgettable experience for me to present at the international conference. I learned a lot about medical image segmentation skills during

the competition. I'm thankful that the organizers provided such a challenging and exciting competition."

—Yu-Cheng Chang, undergraduate

- "As an undergrad student, participating in an application-oriented challenge that involves competitors from all over the world was an exciting, intriguing, and challenging experience. While many of the existing methods have been proposed to address various medical tasks from different aspects, many of the methods are built upon pretrained models or transfer-learning mechanisms, which are prohibited in this competition. Due to the limited duration, we conducted extensive pilot studies and found many methods applicable to the given task. Driven by these methods, we developed an accurate and efficient network to address the task at hand in a 3-D learning fashion. In conclusion, we sincerely thank the organizers for holding this competition. In the future, we hope there will be more competitions so that people from different backgrounds can all be involved."

  —Yun-Chun Chen, undergraduate

- "I am an undergraduate student in my last year of study. My research mainly focuses on deep generative models, but I always hoped to have a chance to work on an image segmentation problem, especially in the 3-D domain. This contest provided me with a perfect opportunity. This contest was very challenging. We needed to take a lot of time to visualize each data, read a lot of papers, and confer with each other to come up with the solution. I really enjoy the contest since it made me grow and gave me a chance to work with some good teammates. I am also very happy that we won third place in the contest."

  —Chao-Te Chou, undergraduate

- "As a graduate student focusing on medical imaging, it was my first time leading a team in order to compete with people from all over the world. From this challenge, I learned a lot about the domain of medical image segmentation techniques and found that there is still a long way to go in this domain. Noise, imbalance, and scarcity of data are the main issues that make it challenging for us to build an accurate and robust segmentation model. I'm so excited to win third place. There was more work than I expected, but it was a precious and rewarding experience. Many thanks to my hardworking team members and to the organizers for holding this contest."

  —Chun-Ting Wu, graduate

"This challenge allowed us to dig deeper into the emerging world of biomedical image processing."

Acknowledgments

The organizers of the 2018 VIP Cup would like to express their utmost gratitude to all who made this adventure a reality, including, but not limited to, the participating teams, the judging panel, the local organizers, and IEEE SPS Membership Board. This competition would have not been possible without the timely and prompt help and follow-up from members of the Cancer Imaging Archive and National Cancer Institute. In addition, great appreciation goes to Dr. Andre Dekker and Dr. Leonard Wee from the Maastricht Radiation Oncology Clinic for providing the competition data set. Special thanks to Dr. Lucio Marcenaro, director of student services of IEEE SPS, for his support and great and exceptional follow-up and dedication, which made it possible for members from all three finalist teams to attend ICIP 2018 and present at the final stage. Finally, great appreciation goes to the student organizers who were involved throughout the course of the 2018 VIP Cup, especially Parnian Afshar, Suzette Slim, and Wu Xin for their dedication and hard work in preparing different competition data sets and performing the evaluations. This project has been funded in whole or in part with Federal funds from the U.S. National Cancer Institute, National Institutes of Health, under contract HHSN261200800001E. The content of this publication does not necessarily reflect the views or policies of the U.S. Department of Health and Human Services, nor does its mention of trade names, commercial products, or organizations imply endorsement by the U.S. Government.

Authors

Arash Mohammadi (arash.mohammadi@concordia.ca) received his B.Sc. degree in electrical and computer engineering at Tehran University, Iran, his M.Sc. degree in biomedical engineering from the Amirkabir University of Technology, Tehran, Iran, and his Ph.D. degree in electrical and computer science from York University, Toronto, Canada. He is an assistant professor with the Concordia Institute for Information Systems Engineering at Concordia University, Montréal, Canada. He is the director of membership services of the IEEE Signal Processing Society. He was the lead guest editor for a special issue in *IEEE Transactions on Signal and Information Processing Over Networks* titled "Distributed Signal Processing for Security and Privacy in Networked Cyber-Physical Systems." He was the organizing committee chair of the IEEE Signal Processing Society Winter School on Distributed Signal Processing for Secure Cyber-Physical Systems and cochair of the Symposium on Advanced Bio-Signal Processing for Rehabilitation and Assistive Systems at the 2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP). He is currently the cochair of the Symposium on Advanced Bio-Signal Processing and Machine Learning for Medical Cyber-Physical Systems at IEEE GlobalSIP 2018.

Parnian Afshar (p_afs@encs.concordia.ca) received her B.Sc. and M.Sc. degrees in industrial engineering from the Amirkabir University of Technology, Tehran, Iran. She is a Ph.D. candidate at the Concordia Institute for Information System Engineering, Montréal, Canada. Her research interests include signal processing, biometrics, image and video processing, pattern recognition, and machine learning. She has an extensive research/publication record in medical image processing-related areas.

Amir Asif (amir.asif@concordia.ca) received his M.Sc. and Ph.D. degrees in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, in 1993 and 1996, respectively. He was a professor with the Electrical and Computer Engineering Department at Concordia University, Montréal, Canada, since 2014, where he is now serving as the dean of the faculty of engineering and computer science. Previously, he was a professor with the Department of Electrical Engineering and Computer Science at York University, Canada. He has served on the editorial boards of numerous journals and international conferences, including as an associate editor for *IEEE Transactions on Signal Processing* (2014–2018) and *IEEE Signal Processing Letters* (2002–2006 and 2009–2013). He has organized four IEEE conferences on signal processing theory and applications.

Keyvan Farahani (farahank@mail.nih.gov) received his M.S. and Ph.D. degrees in biomedical physics from the University of California, Los Angeles, in 1989 and 1993, respectively. He is a program director in the Image-Guided Interventions (IGI) Branch at the Cancer Imaging Program of the National Cancer Institute (NCI), Rockville, Maryland. In this capacity, he is responsible for the development of NCI initiatives that address the diagnosis and treatment of cancer through the integration of advanced imaging and minimally invasive therapies. Since 2002, he has led the NCI initiatives in Oncologic IGI with programs focused on industrial developments, early phase clinical trials, and image-guided drug delivery using nanotechnology. Additionally, he has led a series of NCI workshops that promote an open science model to develop, optimize, and validate platforms for IGI. Prior to joining NCI in 2001, he was a faculty member in the Department of Radiological Sciences at the University of California, Los Angeles.

Justin Kirby (kirbyju@mail.nih.gov) received his undergraduate degree in information technology at Duquesne University, Pittsburgh. He is currently with the Frederick National Laboratory for Cancer Research, Maryland, which provides imaging informatics support to the National Cancer Institute's Cancer Imaging Program. His current work focuses on creating open science resources to improve reproducibility and transparency in cancer imaging research. Most notably, his team manages the Cancer Imaging Archive, which is a service aimed at helping researchers share deidentified radiology and pathology images.

Anastasia Oikonomou (anastasia.oikonomou@sunnybrook.ca) received her undergraduate degree from the Aristotle University of Thessaloniki, Greece, and her Ph.D. degree from the Democritus University of Thrace, Greece. She is the head of the Cardiothoracic Imaging Division at Sunnybrook Health Science Centre, Toronto, Canada, site director of the Cardiothoracic Imaging Fellowship program at the University of Toronto, Canada, and an assistant professor in the Department of Medical Imaging at the University of Toronto. Her research interests include imaging of pulmonary malignancies and interstitial lung diseases, radiomics, and machine-learning methods in pulmonary disease imaging.

Konstantinos N. Plataniotis (kostas@ece.utoronto.ca) received his B.Eng. degree in computer engineering from the University of Patras, Greece, and his M.S. and Ph.D. degrees in electrical engineering from the Florida Institute of Technology, Melbourne. He is the Bell Canada chair in multimedia and a professor with the Electrical and Computer Engineering Department at the University of Toronto, Canada. He is a registered professional engineer in Ontario, Canada, Fellow of the IEEE, and fellow of the Engineering Institute of Canada. He was the IEEE Signal Processing Society inaugural vice president for membership (2014–2016) and the general cochair for the 2017 IEEE Global Conference on Signal and Information Processing. He served as cochair for the 2018 IEEE International Conference on Image Processing in Athens, Greece, and will serve at the 2021 IEEE International Conference in Acoustics, Speech, and Signal Processing in Toronto, Canada.

References

[1] P. Afshar, A. Mohammadi, K. N. Plataniotis, A. Oikonomou, and H. Benali, "From handcrafted to deep-learning-based cancer radiomics: Challenges and opportunities," *IEEE Signal Process. Mag.*, to be published.

[2] H. J. W. L. Aerts, E. Rios Velazquez, R. T. H. Leijenaar, C. Parmar, P. Grossmann, S. Carvalho, and P. Lambin. (2015). Data from NSCLC-Radiomics. The Cancer Imaging Archive. [Online]. Available: https://wiki.cancerimagingarchive.net/display/Public/NSCLC-Radiomics

[3] H. J. W. L. Aerts, E. R. Velazquez, R. T. H. Leijenaar, C. Parmar, P. Grossmann, S. Carvalho, J. Bussink, R. Monshouwer, B. Haibe-Kains, D. Rietveld, F. Hoebers, M. M. Rietbergen, C. R. Leemans, A. Dekker, J. Quackenbush, R. J. Gillies, and P. Lambin, "Decoding tumour phenotype by noninvasive imaging using a quantitative radiomics approach," *Nature Communications*, vol. 5, no. 4006, 2014.

[4] K. Clark, B. Vendt, K. Smith, J. Freymann, J. Kirby, P. Koppel, S. Moore, S. Phillips, D. Maffitt, M. Pringle, L. Tarbox, and F. Prior, "The Cancer Imaging Archive (TCIA): Maintaining and operating a public information repository," *J. Digital Imaging*, vol. 26, no. 6, pp. 1045–1057, 2013.

[5] A. Oikonomou, F. Khalvati, P. N. Tyrrell, M. A. Haider, U. Tarique, L. Jimenez-Juan, M. C. Tjong, I. Poon, A. Eilaghi, L. Ehrlich, and P. Cheung, "Radiomics analysis at PET/CT contributes to prognosis of recurrence and survival in lung cancer treated with stereotactic body radiotherapy," *Sci. Reports*, vol. 8, no. 4003, 2018.

[6] Y. Zhang, A. Oikonomou, A. Wong, M. A. Haider, and F. Khalvati, "Radiomics-based prognosis analysis for non-small cell lung cancer," *Sci. Reports*, vol. 7, no. 46349, 2017.

[7] M. Anthimopoulos, S. Christodoulidis, L. Ebner, T. Geiser, A. Christe, and S. Mougiakakou. (2018). Semantic segmentation of pathological lung tissue with dilated fully convolutional network. arXiv. [Online]. Available: https://arxiv.org/abs/1803.06167

[8] O. Ronneberger, P. Fischer, and T. Brox. (2015). U-net: Convolutional networks for biomedical image segmentation. arXiv. [Online]. Available: https://arxiv.org/abs/1505.04597

[9] G. Huang, Z. Liu, L. V. Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2017, p. 3.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[11] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell. (2017). Understanding convolution for semantic segmentation. arXiv. [Online]. Available: https://arxiv.org/abs/1702.08502

SP

Please send calendar submissions to:
Dates Ahead, Att: Samantha Walter, E-mail: walter.samantha@ieee.org

2019

MARCH

The Data Compression Conference (DCC)
26–29 March, Snowbird, Utah, United States.
General Chairs: Michael W. Marcellin and James A. Storer
URL: http://www.cs.brandeis.edu/~dcc/index.html

APRIL

IEEE International Symposium on Biomedical Imaging (ISBI)
8–11 April, Venice, Italy.
General Chairs: Marius George Linguraru and Enrico Grisan
URL: https://biomedicalimaging.org/2019/

MAY

44th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)
12–17 May, Brighton, United Kingdom.
General Chairs: Saeid Sanei and Lajos Hanzo
URL: http://icassp2019.com

JUNE

IEEE Data Science Workshop (DSW)
2–5 June, Minneapolis, Minnesota, United States.
General Chairs: Georgios B. Giannakis, Geert Leus, and Antonio G. Marques
URL: 2019.ieeedatascience.org

Digital Object Identifier 10.1109/MSP.2018.2877272
Date of publication: 24 December 2018

©ISTOCKPHOTO.COM/GORANQ

The 26th IEEE International Conference on Image Processing will be held at the Taipei International Convention Center, Taipei, Taiwan, 22–25 September.

JULY

IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)
2–5 July, Cannes, France.
General Chair: David Gesbert
URL: http://www.spawc2019.org/

IEEE International Conference on Multimedia and Expo (ICME)
8–12 July, Shanghai, China.
General Chairs: Feng Wu, Lina J. Karam, and Tao Mei
URL: http://www.icme2019.org

SEPTEMBER

27th European Signal Processing Conference (EUSIPCO)
2–6 September, A Coruña, Spain.
General Cochairs: Mónica F. Bugallo and Luis Castedo
URL: http://eusipco2019.org

IEEE International Conference on Image Processing (ICIP)
22–25 September, Taipei, Taiwan.
General Chairs: C.-C. Jay Kuo, Homer H. Chen, and Hsueh-Ming Hang
URL: http://2019.ieeeicip.org

IEEE 21st International Workshop on Multimedia Signal Processing (MMSP)
27–29 September, Kuala Lumpur, Malaysia.
General Chairs: Jenq-Neng Hwang, Chee Seng Chan, and Wen-Huang Cheng
URL: http://mmsp2019.org

DECEMBER

IEEE International Workshop on Computational Advances in Multisensor Adaptive Processing (CAMSAP)
14–18 December, Guadeloupe, West Indies.
General Chairs: David Brie and Jean-Yves Tourneret
URL: https://camsap19.ig.fpms.ac.be

SP

© GRAPHIC STOCK

The Advertisers Index contained in this issue is compiled as a service to our readers and advertisers: the publisher is not liable for errors or omissions although every effort is made to ensure its accuracy. Be sure to let our advertisers know you found them through *IEEE Signal Processing Magazine*.

IEEE SIGNAL PROCESSING MAGAZINE REPRESENTATIVE

Mark David, Director, Business Development — Media & Advertising, Phone: +1 732 465 6473, Fax: +1 732 981 1855, m.david@ieee.org

COMPANY	PAGE NUMBER	WEBSITE	PHONE
MathWorks	CVR 4	mathworks.com/deeplearning	

Digital Object Identifier 10.1109/MSP.2018.2884886

Are You Moving?

Don't miss an issue of this magazine—update your contact information now!

Update your information by:

E-MAIL: address-change@ieee.org
PHONE: +1 800 678 4333 in the United States or +1 732 981 0060 outside the United States

If you require additional assistance regarding your IEEE mailings, visit the IEEE Support Center at supportcenter.ieee.org.

IEEE publication labels are printed six to eight weeks in advance of the shipment date, so please allow sufficient time for your publications to arrive at your new address.

IEEE


© ISTOCKPHOTO.COM/BRIANAJACKSON

Machine-Learning Billboard Collection

by Robert W. Heath, Jr. and Nuria González-Prelcic

Warnings: Claims made here are based on absolutely no facts. Applying machine learning to your research may not lead to more publications, more funding, or a higher citation index. Using machine learning might cause periodic frustration due to the challenges of getting the software to actually work, the search for large enough data sets, or the constant acquisition of more processing power.

(c) Robert W. Heath, Jr. and Nuria González-Prelcic 2018

Digital Object Identifier 10.1109/MSP.2018.2881507
Date of publication: 24 December 2018

New benefit from the IEEE Signal Processing Society

SPS Resource Center

The SPS Resource Center is the new home for the IEEE Signal Processing Society's online library of tutorials, lectures, presentations, and more. Unrestricted access to our fast-growing archive is now included with your SPS membership.

http://rc.signalprocessingsociety.org

We accept submissions, too!
Interested in submitting your educational materials?

sps-resourcecenter@ieee.org

Digital Object Identifier 10.1109/MSP.2018.2884784

MathWorks®

MATLAB SPEAKS DEEP LEARNING

With just a few lines of MATLAB® code, you can use CNNs and training datasets to create models, visualize layers, train on GPUs, and deploy to production systems.

mathworks.com/deeplearning

©2019 The MathWorks, Inc.