Real-Time Piano Music Transcription Based on Computer Vision

Mohammad Akbari and Howard Cheng

***Abstract*—One important problem in musical information retrieval is automatic music transcription, which is an automated conversion process from played music to a symbolic notation such as MIDI file. Since the accuracy of previous audio-based transcription systems is not satisfactory, we propose an innovative computer vision-based automatic music transcription system named claVision to perform piano music transcription. Instead of processing the music audio, the system performs the transcription only from the video performance captured by a camera mounted over the piano keyboard. In this paper, we describe the architecture and the algorithms used in claVision. The claVision system has a high accuracy (F_1 score over 0.95) and a very low latency (about 7.0 ms) in real-time music transcription, even under different illumination conditions. This technology can also be used for other musical keyboard instruments.**

***Index Terms*—Automatic music transcription, claVision, computer vision, multipitch estimation, piano.**

I. Introduction

THE GROWTH of digital music has considerably increased the availability of content in the last few years. Digital music is a representation of analog sound as discrete numerical values in a computer. In order to access, organize, and analyze this vast amount of data, Musical Information Retrieval (MIR) has become an important field of research [1].

One important problem in MIR is Automatic Music Transcription (AMT), which is the process of automatically converting music to a symbolic notation such as a music score or a Musical Instrument Digital Interface (MIDI) file using a computer. AMT is typically considered as the process of extracting the musical sounds from the audio of a piece of music and transcribing them to musical notations. The main input in this procedure is audio or sound because the music is generated from sound. That is why all previous developed AMT methods are based on audio processing techniques [2]–[6]. However, their accuracy is not satisfactory because of numerous difficulties resulting from the use of audio signals (Section III-A) [7]–[11]. Thus, new technologies are required to deal with this problem. One proposed approach is to use computer vision techniques for visually analyzing and transcribing music [12]–[17].

Fig. 1. Ideal location (30–45 degrees from vertical) of the camera over the piano and electronic keyboards.

In this paper, a new computer vision-based system named *claVision* is introduced to perform automatic transcription of piano music. In this system, a camera is located at top of the piano keyboard to capture a video of the performance (Fig. 1). claVision visually analyzes the music played on the piano based on the pressed keys and the pianist's hands. Finally,the transcription of the played music is automatically produced without analyzing the audio of the music. The name claVision is a combination of two words, *clavier*, which means keyboard instrument, and *vision*. Unlike other similar products that perform automatic music transcription by analyzing the audio signal, the audio of the played music is ignored in claVision. As a result, the drawbacks of existing transcription techniques from audio (described in Section III-A) are no longer present [18]. For instance, multiple notes played simultaneously as well as note durations can be accurately detected by claVision because their corresponding pressed keys and the attack and release times of them can be captured by the camera.

The contribution of this paper includes both the design of the claVision system as well as the resulting working implementation. All musical keyboards such as piano, harpsichord, electronic organs, etc. can be used with this system. Even if only a portion of the keyboard is captured by the camera, claVision still transcribes music correctly, as long as all the keys played are visible. It has a high accuracy in transcribing piano music from video performances, over a wide range of speed and complexity of the played music. claVision handles all steps in music transcription automatically with no need for human music experts to assist the transcription. Both live (real-time) and recorded video processing can be handled by claVision. The real-time transcription is performed with a very low latency. claVision can also

Manuscript received March 18, 2015; revised July 23, 2015; accepted August 15, 2015. Date of publication August 26, 2015; date of current version November 13, 2015. This work was supported by the Natural Sciences and Engineering Research Council Discovery Grant Program and by the Alberta Innovates Technology Futures geekStarter Program. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Jiebo Luo.

M. Akbari was with the Department of Mathematics and Computer Science, University of Lethbridge, Lethbridge, AB T1K3M4, Canada. He is now with the School of Engineering Science, Simon Fraser University, Burnaby, BC V5A1S6, Canada (e-mail: akbari@sfu.ca).

H. Cheng is with the Department of Mathematics and Computer Science, University of Lethbridge, Lethbridge, AB T1K3M4, Canada (e-mail: howard.cheng@uleth.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2015.2473702

1520-9210 © 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

Authorized licensed use limited to: COLORADO SCHOOL OF MINES. Downloaded on November 17,2022 at 01:28:16 UTC from IEEE Xplore. Restrictions apply.

deal with many different lighting conditions using an illumination normalization algorithm. Accurate multipitch, onset, and offset detection in real-time is the main goal of claVision. The algorithms chosen for developing this system are kept as simple as possible to decrease the latency in real-time processing while maintaining good accuracy under various lighting conditions.

claVision was the winner of the 2014 Microsoft Imagine Cup Competition in the category of innovation in both Canadian national finals and world semifinals. As one of the top 11 teams in the world, claVision advanced to World Finals in Seattle to be demonstrated at a number of different venues, including the University of Washington, Microsoft headquarters, and the Museum of History & Industry.

The paper is organized as follows. A number of different applications of claVision is mentioned in Section II. In Section III, we describe some existing work related to claVision. The system architecture of claVision is described in Section IV, and the algorithms used are described in Section V. Experimental results and an evaluation of claVision are given in Section VI.

II. Applications

In addition to automatic transcription for musicians, claVision has a number of other useful applications:

- claVision can be used in piano education. For example:
  - claVision highlights pressed keys during a piano performance. It can be used to teach piano visually to hearing-impaired people who wish to learn piano;
  - if claVision has the musical information of a specific piece already, it can watch a student's performance and identify his/her mistakes in playing the piano;
  - remote music lessons: a teacher can watch the live video of the student playing the piano and see if the keys pressed are correct. The played keys are highlighted so they are easier to see remotely even if the quality of the video is low. Symbolic representations such as music scores are more resilient to noise.
- using claVision, concert pianists can visually show their performances (pressed keys on the piano and the music score) in a live concert;
- claVision only requires a camera looking over a "piano-like" keyboard. It can be used on pianos, electronic keyboards, or even toy pianos. Thus, anyone can build their own keyboard and have it sound like a real piano using music synthesis software.

III. Related Work

Different approaches to automatic music transcription developed by other researchers are discussed. First, a brief background of audio processing techniques is given. Then, we describe some previous works based on image and video processing for visually analyzing and transcribing music.

A. Audio Signal Processing

In 1977, Piszczalski and Galler proposed a spectrum-based system for analyzing the recordings of single instruments with strong fundamental frequency [19]. In the same year, the first method for polyphonic pitch-tracking was proposed by Moorer [20], and it can detect two simultaneous musical sounds. In 1990, another approach was developed by Maher [21], which could process real musical recordings if the voices did not cross each other. Most of the recent polyphonic music transcription research have focused on multi-pitch detection and note tracking algorithms [2]–[6].

Although there has been some recent progress in multi-pitch estimation techniques, it remains a research challenge [7]. In spite of the robustness of their algorithms in processing and manipulating audio signals and waves, they are not very accurate and efficient in the transcription tasks because of a number of difficulties as described below.

- There may be multiple lines of music (melodies and chords) being played, for example, in polyphonic and homophonic music. The combination of various audio signals makes it very difficult to distinguish all the notes that are played at the same time [8].
- Octave ambiguity occurs when the same two or more notes from different octaves are played at the same time [9]. As a result, they cannot be differentiated. For example, the spectrum of a single note $C3$ is almost identical to spectrum of the mixture of $C3$ and $C4$.
- The process is susceptible to audio noise.
- It is difficult to detect the exact duration of a note from audio signals. In other words, onset (the beginning of the musical note) and offset (the end of the musical note) of the played notes cannot be accurately detected [10]. This process is much more difficult for some instruments such as the violin because the pitches of the played notes gradually change over a long time period [11].
- If the instrument is not perfectly tuned, it is impossible to extract the played notes correctly by identifying the correct pitches of the notes.

B. Digital Video Processing

Detecting the keys pressed on the piano from a video performance is a challenging problem in computer vision due to some difficulties such as drastic lighting changes, inappropriate camera view angle, hands coverage of the keyboard, vibrations of the camera or the piano, etc. [18]. An automated approach was proposed by Suteparuk for visually detecting and tracking the piano keys played by a pianist using image differences between the background image and current video frame [12]. The background image containing only the piano keyboard was required to be manually taken at first. As described in [12], a number of YouTube videos was used for testing the algorithm instead of using a video camera to capture the performance video. Some of the keys were not detected correctly because of noise and shadows of the hands produced due to the use of skin colour model to detect and remove the hands. The algorithm had a 63.3% precision rate, a 74% recall rate, and an F_1 score of 0.68 for identifying the pressed keys [12]. However, for faster and more complicated pieces of music, the proposed algorithm was less accurate and slower. The algorithm was not tested under different illumination, scales, or image quality. Since the played music was not transcribed to the actual musical notes in [12], the described method cannot be regarded as an automatic music transcription system.

Authorized licensed use limited to: COLORADO SCHOOL OF MINES. Downloaded on November 17,2022 at 01:28:16 UTC from IEEE Xplore. Restrictions apply.

The first visual method for automatic transcription of violin music was proposed in 2007 [13]. The goal of this work was to use visual information of fingering and bowing on the violin in order to improve the accuracy of audio-only music transcription. As reported in [13], the multiple finger tracking algorithm had an accuracy of 92.4%. The string detection algorithm was successful in 94.2% of the cases for the correct detection of the starting and ending points of the strings. However, automatic note inference including the onset, offset, and pitch detection of an inferred note had a very low accuracy of 14.9% compared to human annotated results.

Many musical instruments are played using hands and fingers. In order to make the hand and finger movements comfortable for players, especially beginners, most music scores include appropriate fingering information indicating which hand and finger should be used for playing which note. Thus, fingering information is another useful piece of information that can be extracted using AMT. There exists some research related to the visual extraction of fingering information of music played on piano in recent years. However, they are generally not very accurate or fast, and often assume that the transcription is already available from MIDI keyboards [14], [15]. For example, only half of the finger annotations obtained by the algorithm in [14] were correct. The accuracy of some of the other previous methods has not been reported. Retrieval of fingering information is also a very important task for other musical instruments, especially the guitar [16], [17].

Recent developments have focused on multi-modal transcription systems utilizing both audio and video processing techniques [16], [22]–[24]. Frisson [22] implemented a multi-modal system for extracting information from guitar music. However, this system requires artificial markers on the performer for visual tracking. In 2008, Paleari *et al.* [16] presented a complete multi-modal approach for guitar music transcription. Their method first used visual techniques to detect the position of the guitar, the fret-board, and the hands. Then, the extracted visual information were combined with the audio data to produce an accurate output. According to the results described in [16], this technique had an 89% accuracy in detecting the notes.

IV. System Architecture

The requirements for performing real-time video capturing and music transcription using claVision are a digital camera and a tripod, stand, or any kind of stable mounts holding the camera at the top of the piano. Any type of low-cost camera with an appropriate resolution and video frame rate can be used. To achieve a good accuracy and performance, a camera with spatial resolution and frame rate of at least 320×240 and 24 frames per second (FPS) is recommended. The higher the video frame rate, the more accurate fast pieces of music can be transcribed. However, it may also increase the processing time as more frames need to be processed.

The camera should be mounted over the piano keyboard so that the appropriate portion is visible. The camera does not need to cover the entire piano keyboard. Even if only a portion of the keyboard is captured by the camera, the software can still function properly, as long as all the keys played are visible. Although the software can adjust for variations in camera positions leading to rotated and angled views of the keyboard, it is recommended that the camera be located not far from the ideal location as demonstrated in Fig. 1. Unlike some of the previous works [12], [14], [15], [25], the camera views the piano keyboard at an angle. The ideal angle is 30–45 degrees from vertical, so pressed keys are seen more clearly.

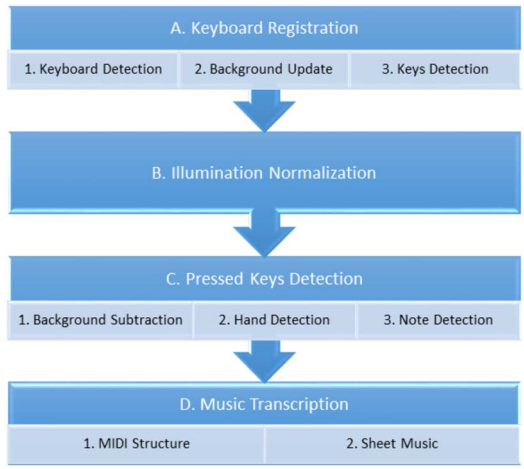


Fig. 2. Four-stage music transcription process and the required sub-tasks performed in each stage.

Although the main output of claVision is the notes with their corresponding durations (onset/offset), it can produce three different output representations:

1) highlight of the pressed keys in the video window showing the piano performance in the user interface, which can also be recorded and saved as a video file;
2) MIDI sound synthesized from the extracted musical information, which the user can save as a MIDI file and play it back after;
3) music score of the played music that can be exported as a Portable Document Format (PDF) file. It is produced by using a simple MIDI to sheet music converter.[1]

V. Description of Algorithms

In claVision, there are four main stages to perform music transcription: keyboard registration, illumination normalization, pressed keys detection, and note transcription (Fig. 2). In the following subsections, each of the four stages and the required sub-tasks will be described. Most of the algorithms used in claVision are relatively simple. The design philosophy is to use the simplest methods possible in order to reduce the latency in real-time processing, while maintaining a high transcription accuracy.

A. Keyboard Registration

This stage is divided into three steps.

1) Keyboard Detection: The keyboard is a quadrilateral shaped by four edges or lines. To compute the location of the

[1]"Midi sheet music," [Online]. Available: http://midisheetmusic.sourceforge.net

Authorized licensed use limited to: COLORADO SCHOOL OF MINES. Downloaded on November 17,2022 at 01:28:16 UTC from IEEE Xplore. Restrictions apply.

Fig. 3. Sample input image with the lines detected using Hough line transform (top) and the transformed rectangle corresponding to the keyboard in the image (bottom).

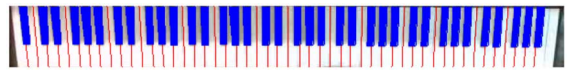


Fig. 4. Estimated lines of the white keys based on the adjacent black keys.

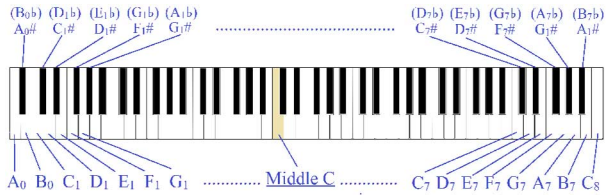


Fig. 5. Standard 88-key piano keyboard, including the associated notes.

keyboard, the positional features of these four lines are extracted using Hough line transform [26]. All 4-combinations of the set of extracted lines are evaluated based on their intersection points. Only those combinations that result in at least four intersections in the plane are considered for further processing. The four intersections in each combination are considered to be the four corners of each quadrilateral. Since the rectangles in the video image may have minor distortions (e.g., rotation, perspective), they are transformed to rectangular images using homogeneous transformations (Fig. 3) [27].

The next step is to find the transformed rectangle that contains the keyboard. From the structure of piano keyboards (Fig. 5), a rectangle can be considered as the keyboard if it has: 1) the maximum brightness in the lower one-third and 2) the maximum number of black keys located in the upper two-thirds part. For the first condition, the means of all intensities in the lower one-third of all candidate rectangles are compared with each other. For counting the number of the black keys in the second condition, a connected components labelling algorithm [28] is used to extract and count individual objects in the image. Since the connected component detection algorithm is performed on binary images, the keyboard image needs to be binarized with an appropriate threshold. The Otsu thresholding method [29] is used to automatically calculate this threshold point for differentiating the objects (black keys) from the background (white keys). The image of the detected keyboard is considered as the initial background image, which will be required in the next stages.

2) Background Update: After detecting the keyboard location, a procedure is used to continuously improve the initial background image by analyzing the next video frames based on the two conditions described in the previous step. In other words, if a keyboard image is found whose brightness in the lower one-third and its number of black keys in the upper two-thirds are more than the ones in the previous background image, it is replaced as the new background image.

There are two main reasons for updating the background image continuously. In some situations, when the camera starts capturing the video of the performance on the piano, the pianist's hands have already covered the keyboard. In this case, the initial background image determined in the first frames includes the hands as well. To handle this issue, the background image is replaced in the next frames in the hope that there is a frame the hands are not present. The other difficulties are noise and variations in lighting conditions during the performance. In order to accurately detect pressed keys, the background image should be consistent with the current lighting conditions. Therefore, the background image needs to be updated as the lighting conditions change.

3) Keys Detection: Keys detection is applied to the obtained background image because it does not include any hand covering the keys. Considering black keys as objects in white background, they are extracted using the connected component detection algorithm. Unlike the black keys, it is difficult to differentiate the white keys from each other, particularly the ones with no black key in between. This is because the lines separating the white keys on the keyboard are often not obvious enough in the captured image (e.g., the bottom image shown in Fig. 3). However, according to the standard dimensions of the piano keys [30], [31], it is possible to estimate these dividing lines based on the positions of the black keys (Fig. 4).

The next step is to assign musical information such as the octaves and the notes to all located keys. In some cases, the captured video does not include the entire piano keyboard with all octaves because of the position of the camera. As a result, identifying the exact octave numbers is impossible even for humans. However, the octave numbers can be estimated by considering the middle visible octave as the octave including the Middle C key (it is usually known as octave 4). Then, the other octaves located on the left and right can be numbered accordingly (Fig. 5).

The black keys on the piano are divided into groups of two black keys ($C\sharp$ and $D\sharp$) and three black keys ($F\sharp$, $G\sharp$, and $A\sharp$). This pattern is repeated in the whole keyboard (Fig. 5). Since there is no black key between these two groups, the space separating them is doubled in comparison to that between the black keys inside each group. Thus, these two groups can be distinguished to determine their associated musical notes. The natural notes corresponding to the white keys are then determined based on the notes of the black keys and the estimated separating lines. For example, the two quadrilaterals which are separated by the line with the start point on the black key $G\sharp$ ($A\flat$) are assigned the notes G and A.

B. Illumination Normalization

Dealing with different illumination conditions can be a difficult issue in video processing, especially when the lighting conditions (brightness or darkness) may change in each video

Authorized licensed use limited to: COLORADO SCHOOL OF MINES. Downloaded on November 17,2022 at 01:28:16 UTC from IEEE Xplore. Restrictions apply.

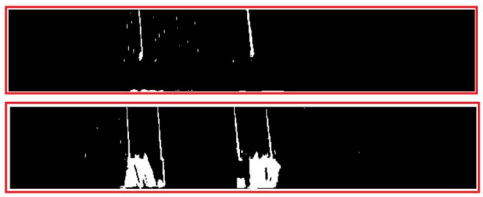


Fig. 6. Positive difference image showing that two black keys have been pressed (top) and negative difference image showing that four white keys have been pressed (bottom).

frame. Different types of noise and shadows are other factors that also cause problems in video processing. In order to deal with these issues, various correction techniques are used by researchers. The proposed method in claVision is to consider the background image as the overlay image and then, based on the intensities in this overlay image, a low level manipulation of pixels is performed in other video frames. As a result, the minor differences (e.g., in illumination, noise, and/or shadows) between the overlay and other images are reduced. In the AForge.NET Framework,[2] this method is called the MoveTowards filter, which applies the following formula to each pixel in all video frames:

$$res := frm + \min(|bgr - frm|, step) \times \mathrm{Sign}(bgr - frm) \quad (1)$$

where frm and bgr are the pixel values in the source image (video frames) and the overlay image (background image). The resulting pixel values are assigned to res. The parameter $step$ (0–255) defines the maximum amount of change per pixel in the source image. The resulting image will be the same as the overlay image if the step size is 255 and it will be equal to the source image if the step size is 0. Having an appropriate step size results in reducing the minor differences caused by different illumination, noise, and shadows between the background image and other video frames.

C. Pressed Keys Detection

Given the background image and the normalized image in each video frame, pressed keys detection is done in three steps.

1) Background Subtraction: Pressing the keys on the piano keyboard causes some changes at the pixel values at the locations of the pressed keys. In order to detect these intensity variations, the background subtraction technique is utilized. Every pixel in the background image is subtracted from the corresponding one in each normalized video frame. The positive and negative values resulted from this subtraction are used separately for analyzing the pressed black and white keys (Fig. 6). Although we consider positive and negative difference images separately in this discussion, the two separate images can actually be represented by a single difference image.

- *Positive difference image*: When a black key is pressed, the adjacent white keys are more prominent. In other words, some of the dark pixels of the pressed black key are replaced by the brighter pixels of the adjacent white keys in the image. The background image includes the unpressed black key and the working video frame includes the pressed one. As a result, a difference image with positive values is produced. These positive values represent the changes caused by pressing the black keys.
- *Negative difference image*: When a white key is pressed, the adjacent black keys are more prominent. In other words, some of the bright pixels of the pressed white key are replaced by the darker pixels of the adjacent black keys in the image. The working video frame includes the pressed white key and the background image includes the unpressed one. As a result, a difference image with negative values is produced. These negative values represent the changes caused by pressing the white keys.

The detection of the pressed white and black keys is done separately to ensure that the white keys are not incorrectly detected as their adjacent black keys and vice versa. The resulting difference images are both converted to the corresponding binary images. The foreground pixels can be either a few connected components or some isolated pixels. The binary positive and negative difference images will be referred as Bkeys and Wkeys images in the next sections.

2) Hand Detection: The keys are usually pressed in the areas that the pianist's hands and fingers are present. Thus, the search domain for finding the pressed keys can be limited by detecting the location of the hands on the keyboard. As a result, the computation required for pressed keys detection can be reduced. In addition, noisy detection results from other parts in which no hands exist can be ignored.

Since the pixel intensities associated to skin colour are lower than those associated to the white keys (even in the grayscale image), the hands and fingers on the piano keyboard can be determined using the Wkeys image. The connected component detection algorithm is then applied to the Wkeys image in order to extract and locate the bounding boxes containing the hands and fingers.

3) Note Detection: Having the list of all piano keys located and registered in the keyboard registration stage, the two difference images (Wkeys and Bkeys), and the location of the hands, the notes played by the pianist in each video frame can be detected as follows. A key is considered as pressed if:

- it is located in at least one of the bounding boxes containing the hands;
- its associated quadrilateral in the difference image includes at least one connected component; and
- the height of the considered connected component is at least half of the key's height.

The white and black keys identified by the first condition are searched in the difference images Wkeys and Bkeys. The third condition is used to reduce the chance that the considered connected component is noise.

For the located white keys with no black key in between (e.g., E and F), the situation is challenging. By pressing one of these keys, the connected component(s) appear(s) in both keys. In this case, the one with more connected components is chosen. If the number of connected components are the same, the key with more isolated pixels in its associated quadrilateral is chosen.

[2]"AForge.NET Framework," [Online]. Available: http://www.aforgenet.com/framework

Authorized licensed use limited to: COLORADO SCHOOL OF MINES. Downloaded on November 17,2022 at 01:28:16 UTC from IEEE Xplore. Restrictions apply.

Fig. 7. Pressed keys detection: highlight of the pressed white and black keys with the colors red and blue.

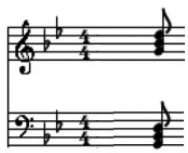


Fig. 8. Produced sheet music corresponding to the pressed keys in Fig 7.

In addition to the connected components resulting from pressing white keys in the Wkeys image, there are other connected components caused by pianist's hands covering the keyboard. To avoid the incorrect detection of these connected components as pressed keys, the lower quadrilateral of the white keys is not considered because it is mostly covered by the hands.

After detecting the pressed keys, their related features such as the octave number and the note name are added to a list named PlayedList indicating the played notes in each video frame. All quadrilaterals related to the pressed keys are then highlighted by drawing colored polygons (Fig. 7).

D. Music Transcription

For transcribing a played note to symbolic notation (e.g., MIDI structure), information such as note name, octave number, and note duration is required. The list of the played notes including the note names and their corresponding octave numbers were determined from the pressed keys detection stage. By maintaining a list of notes that are currently played and comparing it to the list of detected pressed keys in the next frame, we can update the list and also obtain the onset and offset times (attack and release times) of the played notes.

Given the note features such as name, octave number, and onset/offset, a new MIDI event including the required Note On and Note Off messages related to each played note is created. The other corresponding MIDI messages such as the instrument name, tempo, and time signature can be manually determined by the user. If not, they are assumed to be acoustic grand piano, 120 beats per minute, and 4/4 by default. The MIDI stream produced is complete MIDI structure representing the transcription of the played music.

Finally, in order to visualize the results more clearly, the constructed MIDI file is converted to the corresponding sheet music using a simple converter presented in the MidiSheetMusic library (Fig. 8). The sheet music consists of a grand staff including a treble clef for higher notes and a bass clef for lower notes. The notes played in the first octaves (from 1 to 3) are written in the staff with the bass clef and the ones in the last octaves (from 4 to 7) are written in the treble one.

TABLE I
LIST OF THE SAMPLE VIDEOS INCLUDING DIFFERENT SLOW AND FAST PIECES OF MUSIC (THE COLUMNS "FRAMES" AND "FPS" SHOW THE NUMBER OF FRAMES AND THE FRAME RATE OF THE VIDEO)

ID	Frames	Resolution	FPS	Keyboard	Tempo
V1	2596	320×240	24	Piano	40 BPM
V2	1048	640×360	30	Piano	60 BPM
V3	2090	640×360	30	Electronic	80 BPM
V4	1491	640×360	30	Electronic	80 BPM
V5	857	640×360	30	Electronic	120 BPM
V6	4502	640×360	30	Piano	140 BPM
V7	2281	640×360	30	Electronic	200 BPM
V8	3607	640×360	30	Piano	240 BPM

TABLE II
TEST RESULTS OF PRESSED KEYS DETECTION. ALL FRAMES IN THE SAMPLE VIDEOS WERE CONSIDERED FOR CALCULATING THESE RESULTS

Video	Recall %	Precision %	F_1 Score
V1	96.5	96.2	0.963
V2	100.0	96.8	0.983
V3	98.5	99.4	0.989
V4	96.6	99.1	0.978
V5	97.9	93.2	0.955
V6	96.9	96.8	0.968
V7	97.4	99.3	0.983
V8	96.4	98.3	0.973
Average:	**97.5**	**97.4**	**0.974**

TABLE III
PROCESSING TIMES OF DIFFERENT STEPS IN CLAVISION

Step	Processing Time (ms)	
	Maximum	Average
Keyboard Detection	1100.0	691.9
Background Update	53.0	6.1
Keys Detection	20.0	16.6
Image Correction	6.0	1.8
Pressed Keys Detection	45.0	6.6

VI. EXPERIMENTAL RESULTS

claVision was installed and tested on a laptop with an Intel Core i7-4510U CPU (2.00 GHz) and 16 GB DDR3 RAM. Although many videos of different piano and electronic keyboard performances have been used to evaluate the effectiveness of claVision, a few of the representative ones as samples are taken to demonstrate the results.[3] They were captured using two different digital cameras, an SD 240p webcam (with resolution and frame rate of 320×240 pixels and 24 FPS) and an HD 720p webcam (with resolution and frame rate of 640×360 pixels and 30 FPS). The camera was set up close to the ideal configuration, at the angle of 45 degrees from vertical. Table I presents the list of the test videos with their features used in this evaluation. These sample videos include music performances with a wide range of speed and complexity.

The effectiveness of the system in different steps was evaluated based on F_1 score (Table II) and processing time (Table III). The locations of the keyboard in all sample images were successfully detected with a maximum and an average processing time of 1100.0 ms and 691.9 ms. The keys detection process had a very high accuracy of 95.2% and a very low maximum processing time of 20.0 ms. The background update process was done in a maximum and an average time of 53.0 ms and 6.1 ms.

[3]The test videos can be downloaded from http://clavision.ca/samplevideos.

Authorized licensed use limited to: COLORADO SCHOOL OF MINES. Downloaded on November 17,2022 at 01:28:16 UTC from IEEE Xplore. Restrictions apply.

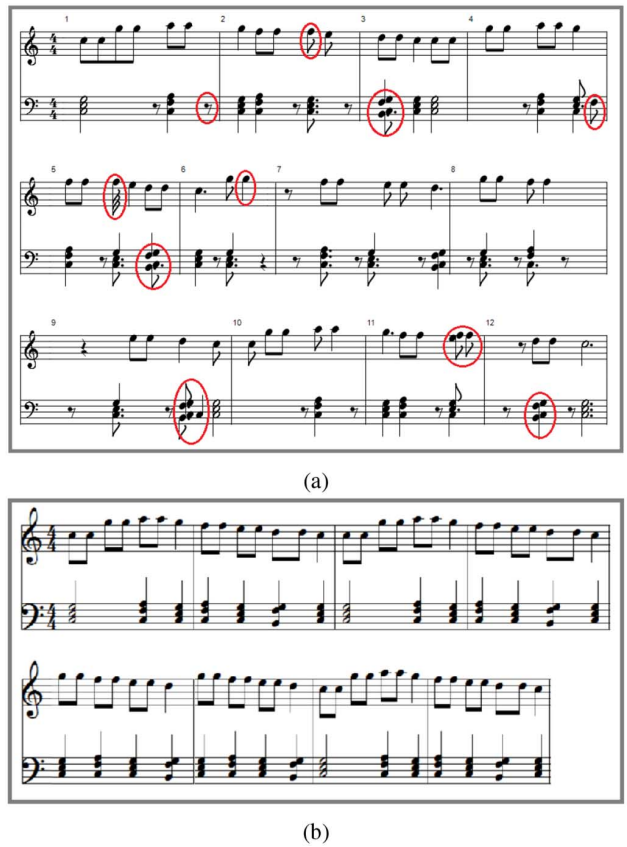
The processing times for this stage did not affect the real-time processing because it was done in a separate thread.

According to the experimental results, issues such as varying illumination conditions, noise, and shadows were satisfactorily dealt with using the illumination normalization algorithm. For example, in the Microsoft Imagine Cup Competition, claVision was demonstrated under a variety of illumination conditions in different locations such as a tent (with natural and artificial lighting), a museum (with shadows from people walking around), conference rooms (with camera flashes of photographers), etc. and it functioned very well. Experiments show that the most appropriate default value for the step size in this algorithm (1) is 50. The illumination normalization procedure has a maximum and an average processing time of 6.0 ms and 1.8 ms, and did not cause any significant latency for real-time processing in claVision. It was demonstrated that the system has a very high accuracy in pressed keys detection with recall, precision, and F_1 score of 97.5%, 97.4%, and 0.974. The latency for pressed keys detection is 6.6 ms.

For each sample, the synthesized MIDI file was accurately produced based on the given musical information and it sounded the same as the played music. The processing time of creating the MIDI structure in each video frame is very small (less than 0.1 ms) for all sample videos.

In order to visually evaluate the correctness of the transcription results, the sheet music (generated from the MIDI file produced by claVision) of the song performed in the sample video $V3$ is demonstrated in Fig. 9(a). This song is called "Twinkle Twinkle Little Star" used for piano learners. The time signature of this music is 4/4. There are 12 measures numbered in the sheet music. No key signature is written in the sheet music because the song is in C major. The ground truth of this song is also provided in Fig. 9(b). 10 keys are incorrectly detected as pressed (false positive) in this piece of music. These incorrect notes are circled in the sheet music [Fig. 9(a)] in measures 3 ($C3$ note), 4 ($F3$ note), 5 ($C3$ note), 9 (two $C3$ notes), and 12 ($C3$ note) of the bass staff as well as measures 2 ($F5$ note), 5 ($F5$ note), and 11 (two $F5$ notes) in the treble staff. In addition, there are 3 pressed keys that are not detected (false negative) in measures 2 ($E5$), 5 ($E5$), and 11 ($E5$) of the treble staff. All errors are related to the adjacent white keys with no black key in between (e.g., E and F). This is because they are more difficult to be distinguished from each other in the difference images.

In video $V3$, the pianist performed a variety of tempo rubato (flexibility in time) in the performance, which caused some notation errors in the duration and location of the notes and rests in the measures. It should be mentioned that these errors were a result from the MIDI to sheet music converter used for producing the music score not from the computer vision algorithms in claVision as it accurately recognizes the times at which a note is played. For example, the number of beats in the first measure of the treble staff is only 3 beats, while it is 4 beats in the same measure in the bass staff. The extra beat in the bass staff is related to the eighth rest highlighted in the sheet music. Another example is the quarter note $G5$ in measure 6 of the treble staff, which should be an eighth note. Most of these "errors" can be corrected manually using different tools developed for manipulating MIDI files and converting them to music scores. Since the main contribution of our method is to automatically extract the notes as well as their timing information, we avoided correcting them.



(a)

(b)

Fig. 9. (a) Produced sheet music of sample video $V3$ (the song is "Twinkle Twinkle Little Star"). Some of the notation errors are highlighted with red circles in the sheet music. (b) The ground truth sheet music.

TABLE IV
LIST OF THE SAMPLE VIDEOS RECORDED IN DIFFERENT CAMERA VIEW ANGLES. THE COLUMN "ANGLE" IS THE CAMERA VIEW ANGLE IN DEGREES. THE COLUMN "KEYS DETECTION%" SHOWS THE DETECTION RATE OF THE KEYS DETECTION PROCEDURE. THE RECALL AND PRECISION RATES AND F_1 SCORE ARE THE RESULTS OF PRESSED KEY DETECTION

Video	Angle	Keys Detection %	Recall %	Precision %	F_1 Score
V9	0	100.0	74.6	85.4	0.796
V10	30	94.9	66.7	93.3	0.778
V11	45	80.3	63.0	100.0	0.773
V12	60	63.9	40.7	64.1	0.498
V13	75	38.6	15.7	28.6	0.165

In order to evaluate the effectiveness of claVision when different camera view angles are used, a few videos of a piano performance were recorded and evaluated at various angles. In these video performances, all keys on the piano including white and black keys were pressed only once to see how different angles affect the accuracy of pressed keys detection over the entire keyboard. Table IV shows the test results of keys detection and pressed keys detection. All videos are in resolution and frame rate of 640×360 pixels and 30 FPS. The speed of pressing the keys in all videos is almost the same.

Since all white and black keys in the video $V9$ are distinguishable by the camera, they were properly located using the keys detection process. The angle of the camera in this video is

Authorized licensed use limited to: COLORADO SCHOOL OF MINES. Downloaded on November 17,2022 at 01:28:16 UTC from IEEE Xplore. Restrictions apply.

0, which means the camera looks straight down over the piano keyboard. In this case, the pressed keys located right under the camera cannot be detected because there are few changes between video frames. As seen in the table, the videos $V10$ and $V11$ have the best compromise in key detection and accuracy. Although fewer keys were detected in $V10$ and $V11$, the undetected keys were located at the two extreme ends of the keyboard. Typically most music pieces are played in the center portion of the keyboard, and this explains the good performance we achieved with actual performances (Table II). The experimental results validate our choice of an angled view instead of straight down view compared to many previous works. Moreover, it is obvious from the table, as the view angle of camera increases, fewer piano keys are successfully located on the piano keyboard because of a drastic perspective view of the camera. The accuracy of pressed keys detection decreases because the keys at the two ends of the keyboard are not detected correctly.

claVision has a number of limitations as described below. Some of these limitations are intrinsic to the approach and cannot be removed even if a high-quality camera is used. However, there are some that can be removed using more sophisticated and often more time-consuming algorithms. We have chosen not to use these approaches in claVision in order to allow real-time transcription.

- In situations where the brightness or darkness of the images is very intense or the lighting conditions change sharply during the performance, claVision cannot function properly.
- There are some situations in which the keyboard and the keys cannot be located using keyboard registration. For example, the claVision system cannot deal with rotations of more than 45 degrees, or drastic perspective views of the camera. Another issue is the unsuccessful detection of the pressed keys that are directly below the camera.
- Covering the piano keyboard by the pianists hands does not cause any problem in the keyboard detection as long as this coverage is not more than 60% of the piano keyboard. One way to ignore the hands covering the keyboard and the keys is to use skin colour algorithm in order to identify the pixels related to the hands. However, it may not work with different skin colours.
- If the camera vibrates or moves slightly during the performance, the pressed key detection algorithm does not work because the background image is not aligned with the current image.
- There are some aspects in music that are related to the human or emotional or expressive side of music (e.g., tempo rubato, dynamics, and articulations), which cannot be dealt with using the claVision system.
- Piano pedals are used to modify the sound of the played notes. Since the audio is ignored, these sound modifications cannot be detected. However, the effect of piano pedals can also be visually analyzed using a second camera located at the base of the piano.

VII. Conclusion

In this paper, a new way for automated music transcription named claVision was proposed to transcribe music played on a piano keyboard or any similar musical keyboard, using only a digital camera (e.g., web-cam) looking over the keyboard. In claVision, the audio is ignored. As a result, many of the drawbacks of existing transcription techniques from audio are no longer present. A four-stage process including keyboard registration, illumination normalization, pressed keys detection, and note transcription is performed to visually transcribe piano music. The claVision system has a very high F_1 score (over 0.95) and a very low latency (less than 7.0 ms) in real-time transcription of piano music.

Some of the possible directions to extend claVision currently being considered are fingering information extraction, music recognizer, mobile- and service-based versions of claVision, and visual music transcription on other musical instruments. Developing a multi-modal music transcription approach utilizing both visual and audio information is another future direction in order to deal with the limitations our approach such as illumination changes, drastic camera views, covered keys, and expressive aspects of music (e.g., dynamics).

Acknowledgment

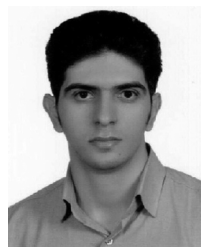
The authors would like to thank the Microsoft Imagine Cup competition for the opportunity to showcase their software, as well as B. Buxton for his encouragement of their work.

References

[1] J. S. Downie, "Music information retrieval," *Annu. Rev. Inf. Sci. Technol.*, vol. 37, pp. 295–340, 2003.
[2] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri, "Automatic music transcription: Challenges and future directions," *J. Intell. Inf. Syst.*, vol. 41, pp. 407–434, 2013.
[3] C. Yeh, "Multiple fundamental frequency estimation of polyphonic recordings," Ph.D. dissertation, Universite Paris VI, Paris, France, 2008.
[4] K. Dressler, "Multiple fundamental frequency extraction for MIREX 2012," in *Proc. 8th Music Inf. Retrieval Eval. eXchange*, 2012, pp. 1–4.
[5] P. Peeling and S. Godsill, "Multiple pitch estimation using non-homogeneous Poisson processes," *IEEE J. Sel. Topics Signal Process.*, vol. 6, no. 5, pp. 1133–1143, Oct. 2011.
[6] P. Smaragdis and J. C. Brown, "Non-negative matrix factorization for polyphonic music transcription," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust.*, Oct. 2003, pp. 177–180.
[7] T. Fernandes Tavares, J. Garcia Arnal Barbedo, R. Attux, and A. Lopes, "Survey on automatic transcription of music," *J. Brazilian Comput. Soc.*, vol. 19, no. 4, pp. 589–604, 2013.
[8] "Ableton Live 9 Audio to MIDI vs. Melodyne," Mar. 2013 [Online]. Available: https://www.youtube.com/watch?v=25MqclpXI7k, Accessed on: Sep. 27, 2014.
[9] C. Lee, Y. Yang, and H. Chen, "Multipitch estimation of piano music by exemplar-based sparse representation," *IEEE Trans. Multimedia*, vol. 14, no. 3, pp. 608–618, Jun. 2012.
[10] J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. B. Sandler, "A tutorial on onset detection in music signals," *IEEE Trans. Speech Audio Process.*, vol. 13, no. 5, pp. 1035–1047, Sep. 2005.
[11] N. Collins, "A comparison of sound onset detection algorithms with emphasis on psycho-acoustically motivated detection functions," in *Proc. Audio Eng. Soc. Conv. 118*, May 2005, Art. ID 6363.
[12] P. Suteparuk, "Detection of piano keys pressed in video," Dept. of Comput. Sci., Stanford Univ., Stanford, CA, USA, Tech. Rep., Apr. 2014 [Online]. Available: http://web.stanford.edu/class/ee368/Project_Spring_1314/
[13] B. Zhang, J. Zhu, Y. Wang, and W. K. Leow, "Visual analysis of fingering for pedagogical violin transcription," in *Proc. 15th Int. Conf. Multimedia*, 2007, pp. 521–524.
[14] D. O. Gorodnichy and A. Yogeswaran, "Detection and tracking of pianist hands and fingers," in *Proc. 3rd Can. Conf. Comput. Robot Vis.*, 2006, p. 63.

Authorized licensed use limited to: COLORADO SCHOOL OF MINES. Downloaded on November 17,2022 at 01:28:16 UTC from IEEE Xplore. Restrictions apply.

[15] A. Oka and M. Hashimoto, "Marker-less piano fingering recognition using sequential depth images," in *Proc. 19th Korea–Japan Joint Workshop Frontiers Comput. Vis.*, 2013, pp. 1–4.
[16] M. Paleari, B. Huet, A. Schutz, and D. Slock, "A multimodal approach to music transcription," in *Proc. 15th IEEE Int. Conf. Image Process.*, Oct. 2008, pp. 93–96.
[17] J. Scarr and R. Green, "Retrieval of guitarist fingering information using computer vision," in *Proc. 25th Int. Conf. Image Vis. Comput. New Zealand*, 2010, pp. 1–7.
[18] M. Akbari, "ClaVision: Visual automatic piano music transcription," Ph.D. dissertation, Dept. of Comput. Sci., Univ. of Lethbridge, Lethbridge, AB, Canada, 2014.
[19] M. Piszczalski and B. A. Galler, "Automatic music transcription," *Comput. Music J.*, vol. 4, no. 1, pp. 24–31, 1977.
[20] J. A. Moorer, "On the transcription of musical sound by computer," *Comput. Music J.*, vol. 4, no. 1, pp. 32–38, 1977.
[21] R. C. Maher, "Evaluation of a method for separating digitized duet signals," *JAES*, vol. 12, no. 38, pp. 956–979, 1990.
[22] C. Frisson, L. Reboursire, W. Chu, O. Lhdeoja, J. Mills Iii, C. Picard, A. Shen, and T. Todoroff, "Multimodal guitar: Performance toolbox and study workbench," *QPSR Numediart Res. Program*, vol. 2, no. 3, pp. 67–84, 2009.
[23] G. Quested, R. Boyle, and K. Ng, "Polyphonic note tracking using multimodal retrieval of musical events," in *Proc. Int. Comput. Music Conf.*, 2008, vol. 2008.
[24] L. Reboursière, C. Frisson, O. Lähdeoja, J. Mills Iii, C. Picard, and T. Todoroff, "MultimodalGuitar: A toolbox for augmented guitar performances," in *Proc. New Interfaces Musical Expression++*, 2010, pp. 415–418.
[25] M. Sotirios and P. Georgios, "Computer vision method for pianist's fingers information retrieval," in *Proc. 10th Int. Conf. Inf. Integration Web-Based Appl. Services*, 2008, pp. 604–608.
[26] P. V. C. Hough, "Method and means for recognizing complex patterns," U.S. Patent 3 069 654, Dec. 18, 1962.
[27] P. Heckbert, "Projective mappings for image warping," Master's thesis, Dept. of Comput. Sci., Univ. of California, Berkeley, CA, USA, 1989.
[28] H. Samet and M. Tamminen, "Efficient component labeling of images of arbitrary dimension represented by linear bintrees," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 10, no. 4, pp. 579–586, Jul. 1988.
[29] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-9, no. 1, pp. 62–66, Jan./Feb. 1979.
[30] R. H. Dorf, *Electronic Musical Instruments*. New York, NY, USA: Radiofile, 1968.
[31] J. G. Savard, "The size of the piano keyboard," 2011 [Online]. Available: http://www.quadibloc.com/other/cnv05.htm, Accessed on: Oct. 10, 2014.



Mohammad Akbari received the B.S. degree in software engineering from Shahid Bahonar University, Shiraz, Iran, the M.Sc. degree in computer science from the University of Lethbridge, Lethbridge, AB, Canada, and is currently working toward the Ph.D. degree in engineering science at Simon Fraser University, Burnaby, BC, Canada.

His research interests include image, video, and audio processing.

Howard Cheng is an Associate Professor with the Department of Mathematics and Computer Science, University of Lethbridge, Lethbridge, AB, Canada. His research interests include image and video processing, computer vision, computer algebra, and symbolic computation.

Authorized licensed use limited to: COLORADO SCHOOL OF MINES. Downloaded on November 17,2022 at 01:28:16 UTC from IEEE Xplore. Restrictions apply.