

OMMR4all - ein semiautomatischer Online-Editor für mittelalterliche Musiknotationen

Wick, Christoph

christoph.wick@uni-wuerzburg.de
Universität Würzburg, Deutschland

Hartelt, Alexander

alexander.hartelt@uni-wuerzburg.de
Universität Würzburg, Deutschland

Puppe, Frank

frank.puppe@uni-wuerzburg.de
Universität Würzburg, Deutschland

Einleitung

Insbesondere für Musikwissenschaftler im Bereich von historischen Manuskripten besteht der Wunsch nach digitalen Bibliotheken, die die gewaltigen Mengen an Material in maschinenlesbare Form (z. B. MEI) speichern. Die Kodierung der alten Werke ist jedoch oft sehr mühselig, da großer menschlicher Einsatz erforderlich ist. Das Aufkommen von künstlicher Intelligenz offenbart hier neue Ansätze, um die Arbeitsprozesse größtmöglich zu automatisieren, indem Algorithmen aus dem Bereich der optischen Musikererkennung (OMR) eingesetzt werden.

Die im Folgenden vorgestellte Software OMMR4all (Optical Medieval Music Recognition For All) realisiert diesen Ansatz für mittelalterliche Manuskripte, die in verschiedenen Neumennotationen, z. B. Quadratnotation, geschrieben sind. Der semi-automatische Workflow erwartet eine einzelne eingescannte Seite als Eingabe und erzeugt als Ausgabe die kodierte Musik z. B. als MEI oder in einer graphischen Anzeige. Hierbei werden verschiedene existierende OMR-Werkzeuge zur Notenlinien-, Notensystem- und Symbolerkennung eingesetzt, die mit einem Overlay-Editor zur Korrektur kombiniert werden. Neumen werden gemäß dem aktuellen MEI-Standard (4.0.1) repräsentiert. Eine manuell in einem modernen Stil gerenderte Beispieltranskription zeigt Abbildung 1.

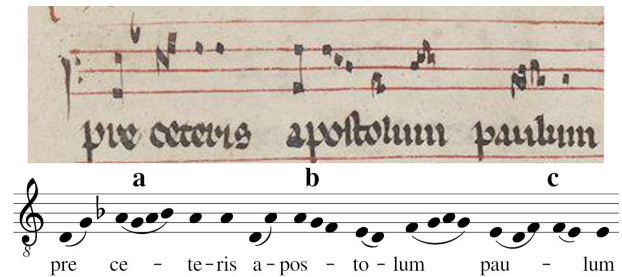


Abbildung 1: Beispieltranskription. Eine oder mehrere Neumen werden zu Silben zugeordnet. Eine Neume besteht wiederum aus einzelnen Notenkomponenten, die entweder graphisch (Bindebogen in a) oder logischen (kleiner Abstand in b) zur vorherigen Komponente verbunden sind. Mehrere Neumen, die zu einer Silbe gehören, werden mit einem größeren Abstand dargestellt (c).

Vigliensoni et al. (2019) stellten bereits einen vergleichbaren OMR-Workflow, der im SIMSSA-Projekt (Fujinaga 2014) eingebettet ist, für Musik des Mittelalters und der Renaissance vor. Hierbei arbeitet die OMR mittels eines Neuronalen Netzes (Calvo-Zaragoza 2018), das jeden Pixel des Originalmanuskripts in verschiedene Klassen wie z. B. Note, Notenzeile, Hintergrund oder Text einteilt. Die automatische Ausgabe kann durch das Webtool Pixel.js korrigiert werden (Zeyad 2017). Für die Weiterverarbeitung werden die klassifizierten Bilden in Ebenen gleichen Typs separiert, sodass ein nachfolgender Algorithmus nicht mehr das Originalbild, sondern ein Binärbild, das z. B. nur noch Notenlinienpixel oder Musiksymbolpixel umfasst. Ein weiterer Algorithmus kann so die Musiksymbole separat erkennen, welche anschließend im Overlay-Editor Neon.js (Regimbal 2019) korrigiert werden können. Der Workflow teilt mehrere Gemeinsamkeiten mit OMMR4all, dessen Umsetzung zeigt jedoch auch Grenzen auf. So entfällt in OMMR4all die erforderliche, teils mühsame pixelgenaue Korrektur zum Erzeugen der separaten Typebenen, da die in OMMR4all verwendeten OMR-Algorithmen direkt auf dem Originalmanuskript arbeiten. Auch arbeitet der von uns vorgestellte neue Overlay-Editor näher am Original, da Notensysteme und Musiksymbole akkurat an die Positionen im Manuskript gezeichnet werden, was einen sehr schnellen Abgleich von Vorlage und Vorhersage ermöglicht.

OMMR4all

Workflow

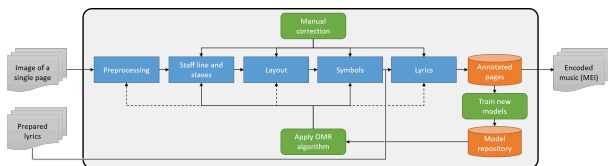


Abbildung 2: Der Workflow von OMMR4all.

Der Workflow von OMMR4all ist in Abbildung 2 gezeigt. Der hochauflösende Scan einer Seite dient neben dem vorbereiteten Liedtext (z. B. in einer Textdatei) als Eingabe. Das Bild wird zunächst durch eine automatische Vorverarbeitung geradegestellt und binarisiert. Anschließend werden Notenlinien und Notensysteme mittels eines Fully-Convolutional Neuronalen Netzes (FCN) erkannt. Darauf aufbauend werden Musik- oder Textregionen separiert, wobei im Standardfall keine exakten Regionen erforderlich sind und so die Layoutanalyse vollständig automatisch abläuft. Basierend auf den Notensystemen werden nun durch ein weiteres FCN Neumen, Notenschlüssel und Vorzeichen erkannt. Abschließend werden die Silben des vorgefertigten Liedtextes an die passenden Neumen gesetzt. Die Algorithmen zu Notenlinien, Notensystem- und Symbolerkennung wurden hierbei direkt von Wick et al. (2019) übernommen.

Iterativer Trainingsansatz

OMMR4all ermöglicht das Training von individuellen Modellen für die Notenlinien- und Symbolerkennung, um die automatische Erkennung auf einen spezifischen, aber noch unbekannten Stil eines Buches anzupassen. Das Training erfordert wenige Seiten an manuell ausgezeichneten Material. Auch dieser Schritt kann durch Verwendung existenter ähnlicher Modelle semiautomatisch erfolgen.

Im Allgemeinen ist zu erwarten, dass die Notenzeilenerkennungsmodelle sehr gut auf verschiedene Notationsstile generalisieren, da Linien in allen Notationen sehr ähnlich sind. Die Symbolnotationen weisen hingegen eine größere Varianz auf, wie beispielsweise an Gotischer- oder Quadrat-Notation zu sehen ist.

Softwarearchitektur

OMMR4all¹ ist eine quelloffene Software², die ein auf einer REST API basierendes Client-Server-Modell implementiert und eine Benutzerverwaltung umfasst. Dies ermöglicht eine niedrige Einstiegshürde für den Einsatz der Software in der Forschung von Musikwissenschaftlern, da keine Installation nötig und die Web-Applikation

plattformunabhängig ist. Zusätzlich wird die Last des Rechnersystems zum Trainieren neuer Modell auf den Server ausgelagert. Dieser kann hierbei mit high-end GPUs ausgestattet werden, um die Rechenzeiten weiter zu reduzieren. Auch werden die Daten zentralisiert gespeichert, was einen weltweiten Zugang von jedem internetfähigen Arbeitsplatz ermöglicht. Demnach ist jeder einfache Laptop oder Desktop PC als Zugriffspunkt zu OMMR4all vollkommen ausreichend und sofort einsetzbar.

Overlay-Editor

Da nicht zu erwarten ist, dass die automatischen Tools von OMMR4all perfekt arbeiten, müssen die Ergebnisse in eleganter und benutzerfreundlicher Art korrigiert werden. Dies ermöglicht der integrierte Overlay-Editor, der eine Überlagerung der Annotationen und der Originalseite anzeigt. Unterschiede von Musiksymbolen können so leicht und mit einem Blick festgestellt und anschließend korrigiert werden. Außerdem können Kommentare hinzugefügt werden, um kritische oder unklare Stelle zu markieren, die dann auch in den kritischen Apparat aufgenommen werden können oder die zur Kommunikation zwischen dem Editor und einem Reviewer, der die Nachkorrektur durchführt, dienen können.

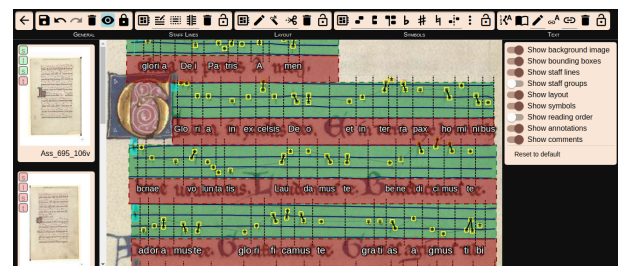


Abbildung 3: Benutzeroberfläche des Editors. Grüne Regionen definieren Notensysteme, rote Regionen Liedtext. Individuelle Notenkomponenten werden als gelbe Boxen dargestellt, grafische Verbindungen von Neumen werden als durchgezogene schwarze Linien, die zwei Noten verbinden gezeichnet, wohingegen die gestrichelten vertikalen Linien den Start einer neuen Neume angeben. Notenschlüssel sind türkis markiert. Die Silben des Liedtextes sind unter der zugehörigen Neume in der jeweiligen Textregion ausgerichtet. Die verschiedenen Knöpfe der Werkzeugleiste dienen zur Korrektur der Annotationen oder um die automatischen Algorithmen zu starten. Nicht gezeigt sind Lesereihenfolge oder Kommentare des Korrektors.

Abbildung 3 zeigt die Benutzeroberfläche des Editors. Der Editor ist konzipiert, damit er ohne steile Lernkurve leicht bedient werden kann: Die Interaktionen zur Selektion, zur Bewegung, zum Ziehen oder zum Einfügen von Elementen erfolgen mit der Maus. Erfahrene Nutzer

können jedoch auf Tastaturkürzel zurückgreifen, um den Bearbeitungsprozess zu beschleunigen.

Evaluation

Zur Evaluation der Transkriptionszeit verglichen wir OMMR4all mit Monodi+ (Eipert 2019), einem Tool, das unter anderem einen hochentwickelten Editor für eine tastaturbasierte Eingabe von Cantus Planus (monophone Musik der Westlichen Kirche) anbietet. Die Bedienung der beiden Softwaresysteme erfolgte stets durch Experten des jeweiligen Programms. Als Material wählten wir jeweils fünf Seiten in gotischer und Quadratnotation (eine Beispielseite wird in Abbildung 3 bearbeitet). Wir maßen und verglichen die mittleren Zeiten, die bei Vorlage des Liedtextes nötig waren, um eine korrekte Transkription der Manuskripte zu erzielen. Somit wurden nur die Zeiten zur Korrektur oder Eingabe von Notenlinien und Musiksymbole gemessen. Die Modelle für die Quadratnotation waren auf 49 Seiten trainiert, die aus einem weiteren Buch stammen. Die gotischen Modelle basieren auf vier weiteren Seiten aus dem identischen Buch. Tabelle 1 fasst die Ergebnisse zusammen.

Tabelle 1: Evaluation der Transkriptionszeiten in Minuten.
Wir listen die Anzahl der Symbole, die erforderlichen Zeiten für die Korrektur der Notenlinien, der Symbole, der Silbenzuordnung, und die Gesamtzeit. Alle Werte sind gemittelt und relativ zu einer Seite angegeben.

Notation	#Symbole	OMMR4all	Monodi+ +	Speed-up			
		Notenlinien	Symbole	Silben	Gesamt		
Gotisch	158	0,3	2,3	1,8	4,5	5,6	1,3
Quadrat	267	0,6	3,3	2,9	6,9	8,5	1,2

OMMR4all zeigt einen Speed-Up von 1,3 und 1,2. Hierbei ist zu beachten, dass die Bearbeitungszeit mittels Monodi+ bereits am Limit ist, da jedes Symbol manuell eingegeben werden muss, worauf das Interface perfekt zugeschnitten ist. OMMR4all hingegen kann sich stets weiterentwickeln und selbständig genauere Modelle lernen. Insbesondere wenn die verwendeten Modelle ausgehend von der aktuellen Fehlerrate von 10% genauer arbeiten, ist mit einer deutlichen Reduktion der Bearbeitungszeit für die Symboleingabe zu rechnen. Weiterhin verspricht die

Entwicklung einer automatischen Silbenerkennung und -zuweisung eine weitere Beschleunigung. Natürlich können, falls die Modelle menschliche Genauigkeit erreichen, alle Seiten vollständig automatisch verarbeitet werden. Im Allgemeinen liefert OMMR4all im Vergleich zu einer manuellen Eingabe des Notentextes eine inhärente Erklärungskomponente für den Ursprung eines jeden Symbols, was insbesondere für einen kritischen Apparat relevant ist.

Geplante Erweiterungen

Trotz der vielen Funktionen von OMMR4all, existieren etliche weitere mögliche Verbesserungen oder Erweiterungen. Einige anstehenden Aufgaben werden im Folgenden vorgestellt.

Zunächst planen wir verschiedene Werkzeuge und Algorithmen, um den Liedtext automatisch zu erfassen. Das Hauptproblem hierbei ist, dass derzeit keine OCR-Engine verlässlich mit handgeschriebenen Text ohne spezielles Training umgehen kann. Deswegen soll in einem ersten Schritt zunächst die automatische Silbenzuordnung gelöst werden, wobei immer noch der vorgefertigte Liedtext vorliegen muss. Hierzu werden die fehlerhaften Ergebnisse der OCR-Engine Calamari (Wick 2018) als Vorschläge für die Position verwendet indem die bestmögliche Übereinstimmung von erkanntem und korrektem Text gefunden wird. Vorläufige Ergebnisse sind vielversprechend, selbst wenn ein großer Prozentsatz der erkannten OCR-Zeichen falsch ist. Die eigentliche automatische Erfassung des handgeschriebenen Textes stellt eine große Herausforderung dar, da eine Seite meist nur wenige Zeilen umfasst, jedoch viele manuell transkribierte Zeilen für ein Training erforderlich sind. Alle modernen Ansätze verwenden hierfür ein Sprachmodell mit n-Grammen oder zumindest einem Wörterbuch sowie tiefe Neuronale Netze, meist bidirektionale LSTMs, die mit CNNs gekoppelt werden. Chammas et al. (2018) verwendeten mehrere tausend Seiten mit reinem Text von beliebiger Handschrift und erhielten eine Wortgenauigkeit von etwa 80%. Auf sauberer geschriebenen mittelalterlichen Manuskripten erzielten Fischer et al. (2014) eine Wortgenauigkeit von etwa 93% mit etwa 11.000 Wörtern im Trainingsdatensatz und einem eingeschränkten Vokabular von etwa 5.000 Wörtern. In einem Szenario, in dem nur die Wörter des Trainingsdatensatzes bekannt waren, wurde eine Wortgenauigkeit von unter 78% erreicht, da etwa 15% der Wörter unbekannt waren. Eine Umsetzung der Techniken für Liedtexte steht noch aus, denn hier besteht ein zusätzliches Problem, dass viele Worte in Silben aufgeteilt sind, was den Einsatz von Wörterbüchern erschwert.

Andere Pläne umfassen weitere monophone Notationsstile zu unterstützen. Hierunter fallen sowohl ältere Neumennotationen, sowohl solche ohne Notenlinien, als auch spätere Mensurennotationen. Hierzu bedarf es nur kleinere kosmetischer Änderungen am Editor, jedoch ist

größerer Aufwand beim Entwickeln neuer Algorithmen nötig. Polyphone Notationen, auch solche bei denen die Stimmen jeweils in separaten Notensystemen vorliegen, stellen semantische bzw. hierarchische Änderungen der Musiknotation dar, da z. B. zwei oder mehrere gleichzeitig klingende Notenzeilen zu einer Akkolade zusammengefasst werden müssen, was Änderungen am Datenformat und somit auch am Editor erfordert.

In der Praxis wird OMMR4all im Corpus Monodicum Projekt³ der Universität Würzburg eingesetzt, um den Prozess der Transkription der Bestände von einstimmiger Musik des lateinischen Mittelalters zu beschleunigen. Der Overlay-Editor, der mit einem Blick erlaubt Fehler zu erkennen, hebt den Transkriptionsprozess bereits auf ein hohes Qualitätsniveau. Trotzdem ist zur Qualitätssicherung ein zweistufiger Prozess notwendig, indem ein musikwissenschaftlicher Reviewer das Ergebnis des Transkriptionsprozesses überprüft, das in Zusammenarbeit von OMMR4all und einem menschlichen Editor erzeugt wurde. Technisch wird dies durch die Möglichkeit unterstützt, pro Seite eine Freigabe zu dokumentieren oder, wie oben erwähnt, Kommentare zur Nachbearbeitung anzugeben.

Fußnoten

1. Eine Demo-Anwendung, die das testweise Bearbeiten zweier unterschiedlicher Bücher erlaubt ist unter <https://ommr4all.informatik.uni-wuerzburg.de/> verfügbar.
2. Der Quellcode kann auf <https://github.com/OMMR4all/> eingesehen werden.
3. <http://www.musikwissenschaft.uni-wuerzburg.de/forschung/corpus-monodicum/>

Bibliographie

Calvo-Zaragoza, Jorge / Castellanos, Francisco / Vigliensoni, Gabriel / Fujinaga, Ichiro (2018): "Deep neural networks for document processing of music score images", in: *Applied Sciences* 8: 654.

Chammas, Edgard / Mokbel, Chafic / Likforman-Sulem, Laurence (2018): "Handwriting Recognition of Historical Documents with Few Labeled Data", in: *13th IAPR International Workshop on Document Analysis Systems (DAS)*, Vienna.: 43-48.

Eipert, Tim / Herrmann, Felix / Wick, Christoph / Puppe, Frank / Haug, Andreas (2019): "Editor Support for Digital Editions of Medieval Monophonic Music", in: *Proceedings of the 2nd International Workshop on Reading Music Systems (submitted to)*.

Fischer, Andreas / Baechler Michael / Garz, Angelika / Liwicki, Marcus / Ingold, Rolf (2014): "A Combined System for Text Line Extraction and Handwriting Recognition in Historical Documents", in: *11th IAPR International Workshop on Document Analysis Systems*, Tours, 2014: 71-75.

Fujinaga, Ichiro / Hankinson, Andrew / Cumming, Julie E. (2014): "Introduction to SIMSSA (single interface for music score searching and analysis)", in: *Proceedings of the 1st International Workshop on Digital Libraries for Musicology*: 1-3.

Saleh, Zeyad / Zhang, Ké / Calvo-Zaragoza, Jorge / Vigliensoni, Gabriel / Fujinaga, Ichiro (2017): "Pixel.js: Web-based pixel classification correction platform for ground truth creation.", in: *14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*: 39-40.

Regimbal, Juliette / McLennan, Zoé / Vigliensoni, Gabriel / Tran, Andrew / Fujinaga, Ichiro (2019): "Neon2: A verovio-based square-notation editor", in: *Music Encoding Conference 2019*.

Saleh, Zeyad / Zhang, Ké / Calvo-Zaragoza, Jorge / Vigliensoni, Gabriel / Fujinaga, Ichiro (2017): "Pixel.js: Web-based pixel classification correction platform for ground truth creation.", in: *14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*: 39-40.

Vigliensoni, Gabriel / Daigle, Alex / Lui, Eric / Calvo-Zaragoza, Jorge / Regimbal, Juliette / Nguyen, Minh Anh / Baxter, Noah / McLennan, Zoe / Fujinaga, Ichiro (2019): "Overcoming the challenges of optical music recognition of early music with machine learning", in: *DH2019*.

Wick, Christoph / Hartelt, Alexander / Puppe, Frank (2019): "Staff, Symbol and Melody Detection of Medieval Manuscripts written in Square Notation using Deep Fully Convolutional Networks", in: *Applied Sciences* 9.

Wick, Christoph / Reul, Christian / Puppe, Frank (2018): "Comparison of OCR Accuracy on Early Printed Books using the Open Source Engines Calamari and OCRopus", in: *JLCL: Special Issue on Automatic Text and Layout Recognition*: 79-96.