Mobile System for Optical Music Recognition and Music Sound Generation

Julia Adamska, Mateusz Piecuch, Mateusz Podgórski, Piotr Walkiewicz, Ewa Lukasik

Mobile System for Optical Music Recognition and Music Sound Generation

Julia Adamska, Mateusz Piecuch, Mateusz Podgórski,
Piotr Walkiewicz, Ewa Lukasik

Institute of Computing Science
Poznan University of Technology
Piotr.Walkiewicz@yahoo.com
Ewa.Lukasik@cs.put.poznan.pl

Abstract. The paper presents a mobile system for generating a melody based on a photo of a musical score. The client-server architecture was applied. The client role is designated to a mobile application responsible for taking a photo of a score, sending it to the server for further processing and playing mp3 file received from the server. The server role is to recognize notes from the image, generate mp3 file and send it to the client application. The key element of the system is the program realizing the algorithm of notes recognition. It is based on the decision trees and characteristics of the individual symbols extracted from the image. The system is implemented in the Windows Phone 8 framework and uses a cloud operating system Microsoft Azure. It enables easy archivization of photos, recognized notes in the Music XML format and generated mp3 files. An easy transition to other mobile operating systems is possible as well as processing multiple music collections scans.

Keywords: Optical Music Recognition, OMR, mobile applications, Windows Phone

1 Introduction

Optical Music Recognition (OMR) is an automatic recognition and classification of symbolic music notation. It is usually performed on scanned musical sheets and uses specialized methods of image processing and classification. The research in the domain of OMR has a long history that began in sixties of last century [21,22], was popularized by Bainbridge [4,5,6] and Fujinaga [8,9] and continued through the years [10, 11] up to nowadays [12,19,23] leaving still open issues for further research. The advanced algorithms are used not only for printed scores, but also for handwritten music notation [1,17,18,23]. Various methods of musical symbols recognition have been applied, e.g. statistical classification methods using support vector machines (SVMs), neural networks (NN), k-nearest neighbours (kNN) and hidden Markov models (HMM) compared by Rebelo [30]. Decision trees have been used by Baumann [29] and for mobile technology by Keon-Hee Park et al [31], unfortunately the paper is available only in Korean. It is difficult to compare results of OMR, as

a ground truth database does not exist. The mobile part of the system performance evaluation has been proposed by Szwoch [27], who introduced a method for comparing and evaluating results of recognition systems stored in MusicXML format [15].

MusicXML is a digital sheet music interchange and distribution open format invented by M. Good at 2000 and since then developed collaboratively by a community of musicians and software developers [15]. It has already been used by numerous sites offering sheet music in this format, or formats that may be easily converted to it.

Despite the fact that many research problems have not been fully solved yet, as e.g. recognition of hand written scores, there exist computer applications that perform this task for users. Well known are e.g. commercial programs, like SmartScore [26], PhotoScore Ultimate [20], SharpEye [25] and open source –Audiveris [3]. Each of them process an image of a music sheet and transcribe it to MusicXML format. However, as it might be deduced from various internet forums, users are hardly satisfied with existing systems and this should push researchers to pursue towards finding some other reliable solutions in this domain.

Wider popularization of OMR systems may be expected from mobile technology. Smartphones, with their ability to take pictures, are replacing personal scanners. The rise of cloud computing enhances possibilities of storing and sharing digital sheet of music. First mobile OMR applications are already on the market. The most popular is iSeeNotes [13] available for Android and iOS, NoteReader [14] - available for iOS and PhotoScore Ultimate that has its version for Android and iOS. As the best of authors' knowledge in the moment of writing this paper none of mobile OMR applications are available for Windows Phone. The presented solution tries to fill this gap.

The paper presents a mobile system for generating a melody from a photo of a musical score. The system has an educational purpose and is addressed to a large group of people that cannot read music notation. The presented solution is complete, i.e. applies all stages of musical symbols recognition using the original method based on decision trees, applies MusicXML format for storing the semantic information and enables playing the recognized melody with a timbre of various popular musical instruments. An easy archivization of photos, and MusicXML documents is also possible. Thanks to its client-server architecture the proposed solution enables its easy transition to other mobile operating systems and to process multiple scans of music collection. While being complete, the system still enables introducing the improvements to the algorithm of musical symbols recognition.

The paper is structured as follows. Section two describes the OMR and music sound generating system architecture. Section three is devoted to the presentation of music notation recognition algorithm. Section four presents the experiments results and Section five concludes the paper.

2 Mobile OMR and sound generating system architecture

The discussion of problems presented in the Introduction shows, that many issues of OMR is not solved yet and the proposed solutions has their limitations. The goal posed by the authors of this paper was to built a mobile OMR application employing

up-to-date elements of software technology able to serve users with no musical background to give them possibility to listen to the music encoded in musical notation. The client-server architecture presented in Figure 1. has been employed. The image processing is at the server side giving the possibility to modify musical notes recognition procedure without any interventions on the mobile client side.

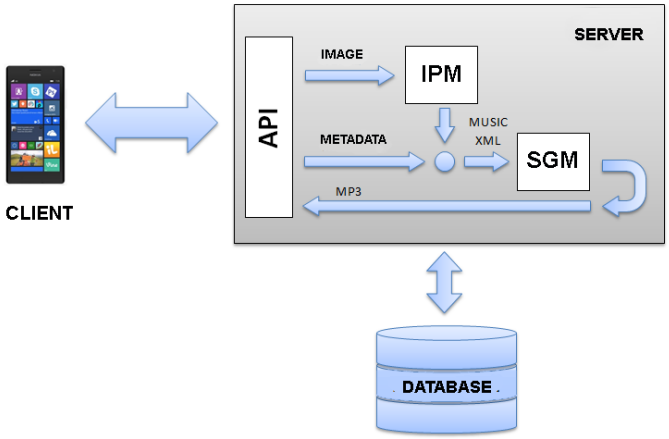


Fig. 1. Architecture of the OMR and sound generation system

The client. The client role is designated to a mobile application (built on Windows Phone 8 platform) responsible for taking a photo of a score, sending it to the server for further processing and playing mp3 file received from the server. Figure 2 presents the screenshot of a score before and after starting the playback. The note being played is highlighted on the photo of the score by a round marker (Figure 2b). The pictures of the scores may be stored in the database on the server, the same procedure of playing music on the client side is possible for items from the repository.

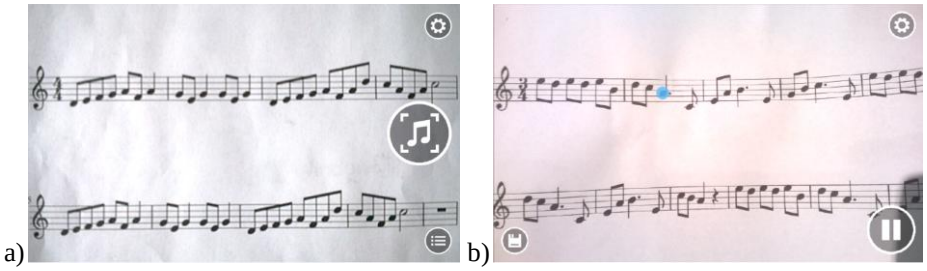


Fig. 2. The photo of a score a) before and b) at the time of playing sound (the note being played is highlighted)

The server. The server communicates with the client through the API created in the REST architectural style (Representational State Transfer) [24]. First an image is

sent to the Image Processing Module (IPM) that recognizes musical symbols and outputs them in MusicXML format. Combined with the additional metadata obtained from the client concerning e.g. tempo and musical instrument they are further processed by the Sound Generation Module,SGM (see Section 3). The resulting mp3 file is sent back to the client and the information about the file is stored in the database. A relational database – an SQL Server – stores users' login information and all other resources including analyzed musical pieces in Music XML format accessible through the website.

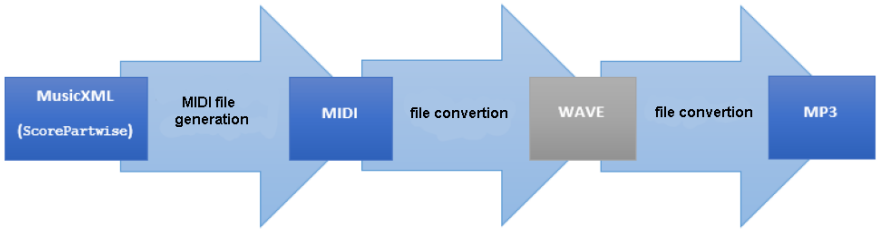
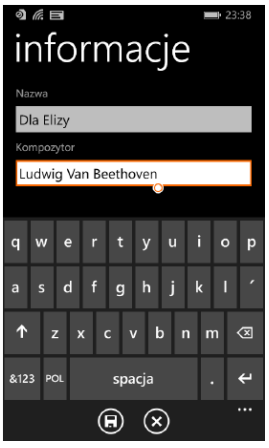
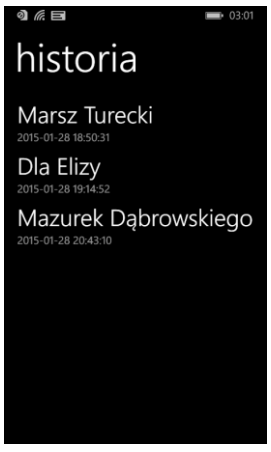


Fig. 3. Generating an .mp3 music file from MusicXML notation.

Sound Generation Module. In the time of building the application, MIDI APIs for Windows Phone 8.1 were still in preview and there was a chance that they might have changed when the final version was released. Therefore it was decided to apply a safer and stable solution in the Sound Generation Module (SGM) – to convert MIDI file to .mp3 format. First a MIDI file is created from MusicXML notation. Then it is converted to .wav format, compressed and sent as mp3 file. Since MIDI is an intermediate stage of sound generation, it is possible to change its several attributes, like playing instrument or tempo in bpm (beats per minute). The flowchart of the sound generation procedure is presented in Figure 3.

a) 

b) 

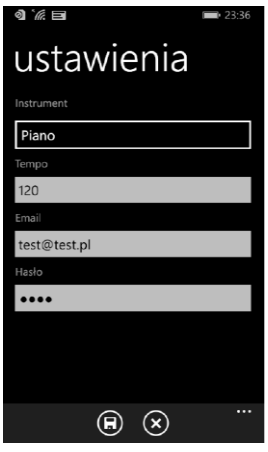
c) 

Fig. 4. Elements of user interface: a) file description, b) processing history review, c) settings (the prototype application has been prepared in Polish)

User interface. A user can navigate between several views and has access to previously opened pages. At the start of the application the user or registers to the system, or logs in, or, if she is already logged in, may start scanning the score (2a). If the user receives the phone call while using the application, it gets into the *dormant* state and may be accessed again later. The generated melody may be played, paused, restarted and stopped (Figure 2b). User may also store the processed musical piece together with textual descriptions (Figure 4a), access history of the processed files (Figure 4b) and navigate to the settings of the system (Figure 4c).

3 Algorithm for Optical Music Recognition

The key element of the system is the program realizing the algorithm of musical symbols recognition. It is based on the rules based decision trees. The procedure follows the typical OMR scenario consisting of the following steps:

- image preprocessing,
- staff lines detection and removal,
- merging broken elements,
- recognition of musical notation elements,
- semantic reconstruction.

In the subsequent subsections each of the above steps will be described from the applied algorithm perspective.

3.1 Image preprocessing

Since the system is devoted to recognize the photographic images, their quality may substantially differ, e.g. the luminance may be too low, or staves may be skewed or rotated. Therefore the preprocessing of the image is, like in many other image processing applications, indispensable. The operations chain is following:

- conversion of the image to the grayscale,
- filtering for sharpening the lines of staff,
- noise reduction using median filter,
- elimination of skewness and rotation of an image based on Hough transform,
- image binarization using adaptive filtering.

All those steps are performed using the elements of the AForge.Net [2] software library with or default, or experimentally adjusted parameters. The conversion of the color image to the grayscale is performed with the values: R - 0,2125, G - 0,7154 and B - 0,0721. Filtering for sharpening the lines of staff is performed by convolving the image with a 3x3 points kernel: { -2, -1, 0 }, { -1, 1, 1 }, { 0, 1, 2 }

For noise reduction a median filter with the same size 3x3 pixels window is used. Image binarization is made using adaptive filtering. We quote after [2]:" The brief idea of the algorithm is that every image's pixel is set to black if its brightness is t

percent lower than the average brightness of surrounding pixels in the window of the specified size". The default values are t = 0.15 and window size – 41 pixels.

Hough Transform determines the angle of the document by detecting the most distinct straight lines, in this case – staff lines. It uses polar coordinate system to represent lines: distance of its closest point from the origin and an angle. With the angle detected, the whole image may be rotated to get horizontal lines (Figure 5).

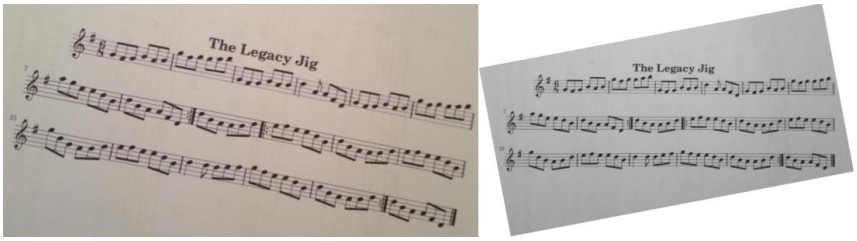


Fig. 5. Example of staves rotation based on straight lines detection using Hough Transform

3.2 Staff lines detection and removal

Staff lines, fundamental elements of western musical notation, enabling symbolic notation of musical pitch, distort the shape of musical symbols and in most OMR systems are objects to be removed [7]. On the other hand staff line thickness and staff space height are reference values to find out the size of other music symbols. Therefore the detection of staves and their thorough analysis is the crucial part of the algorithm.

The most common algorithm is based on the horizontal projections [8]. A binary image is horizontally mapped into a histogram by accumulating the number of black pixels in each row. To indicate the line, a local maximum is searched. Since lines mostly are not entirely straight, are undulated, may be broken and have some other distortions, they are analyzed in segments and then a procedure ensuring continuity of line segments is applied. Such approach was proposed i.a. by Szwoch [28].

3.3 Merging broken elements

The operations performed during the preprocessing step, as well as the staff lines removal cause the continuity of musical symbols shapes to be distorted or broken (Figure 6). In order to make the algorithm recognize them, the broken parts have to be connected again. This task is difficult, as there are symbols that are very close one to another and should not be connected. Finding a rule that is not computationally demanding and able to distinguish a broken element from small parts inherent to musical notation is difficult.

Typical points at which continuity is interrupted are upper and lower parts of the whole notes, half notes head, stems, flags or beams. Thus those primitive symbols are first extracted. The principle is simple: a set of connected black pixels is treated as one coherent object. The rules for merging objects are following:

- a rectangle boxes bounding objects are close enough each other (at a distance of three times the thickness of the line).
- the height of the bounding box after the merge of several elements does not exceed the height of the staff or exceeds it, but one of the components shape suits linked notes, and the size of the second one is the same as a single note (this rule concerns a group of eighths/sixteenths that have to form a single object).
- none of the rectangles bounding connected objects is equal in shape to the to the rectangle bounding a box.

All fragments are sorted according to their distance. For a given fragment a match is sought in the order of this list. If any of them match up, then the two are combined and next match is sought for, without returning to fragments previously browsed. In practice this limitation does not influence correctness of the merge and covers most cases while speeding up processing time.

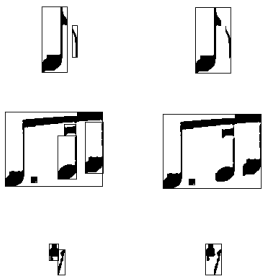


Fig. 6. Merging broken elements of music symbols after staff lines removal and filtering

3.4 Recognition of music notation elements

The recognition of musical notation elements is performed using a decision tree based on rules related to musical symbols construction. Each musical object may be described by a set of characteristic features. For example, the whole notes may be described by the height of its bounding box that is approximately equal to the distance between the staff lines, the center of gravity located near the center of the bounding box and a relatively small number of black pixels in comparison with the white pixels in that box.

Descriptive features. After analyzing the entire set of musical symbols that were to be recognized according to the system requirements the following set of features was distinguished:

1. *object bounding box height (or width) in relation to a staff line space* - allows for an initial segregation of objects;
2. *center of gravity* of an object - allows to specify whether the notes are directed upwards or downwards.
3. *black-to-white pixels number ratio* – e.g. this ratio for whole note- and half note-pauses or dots are above 70%;

4. *object bounding box aspect ratio;*
5. *number of intersections of vertical or horizontal lines with black elements of the objects;* this parameter is useful for distinguishing between eighth and sixteenth notes at their flags.

The number of features may be increased in the future.

Decision tree construction. The root of the tree performs the operation according to the feature 1. - *object bounding box height (or width) in relation to the staff line space*. Exemplary fragment of the decision tree from which the decision rules may be acquired is presented in Figure 7. A branch on the left provides to the recognition of the dot symbol. The algorithm in its current form recognizes: bar lines, whole notes, half notes, quarter notes, eighths, sixteenths single or linked, corresponding pauses and a dot. The tree may be further expanded in along with new characteristic features for new music symbols.

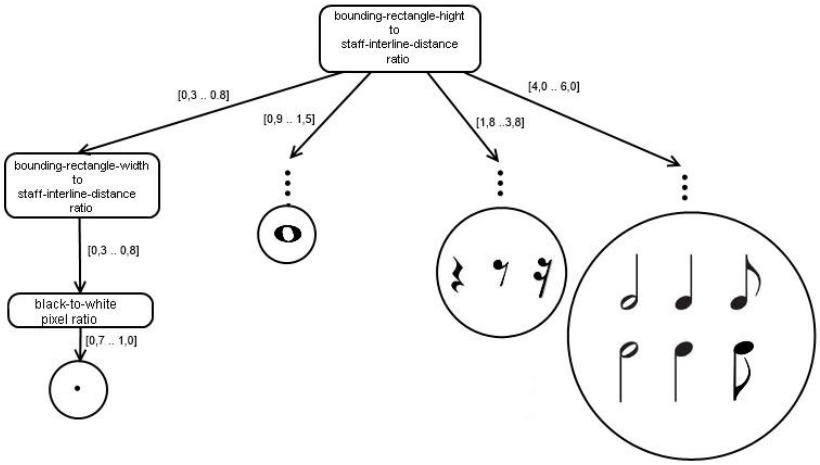


Fig. 7. Fragment of rule based decision tree recognizing musical objects

Recognition of linked notes. Linked notes are recognized outside a decision tree. The features taken into account are the size of a bounding box and a number of stems. First the stems are removed to separate the notes heads in linked notes. Then using the set of vertical lines for each head and counting the number of intersections with beams their number may be found: one for eighths and two for sixteenths (Figure 8).

For a given fragment to review all others (list sorted by distance). If any of them match up, then we combine and go on (without checking for this combined fragment previously browsed). You can imagine that such checking all the pieces should be done after each merge. This limitation is firstly due to the processing time and observation, that this approach covers most typical cases.

Fig. 8. Linked notes without stems and sets of lines to recognize eighths and sixteenths

3.5 Semantic reconstruction and generation of output data

Another set of heuristic rules is used to perform the semantic reconstruction of musical information and to code it to MusicXML format. First all elements other than musical symbols have to be removed (text, some notes etc.). Then the musical object position relating to a staff is calculated. A so called characteristic point of the element is calculated. For notes it is the center point of a note head and for the pause it is the center of its bounding box. Musical object maximum distance from the staff is also defined. For a pause – only the duration time is calculated. Additionally the coordinates of the notes have to be found in order to highlight them when playing the melody received from the server.

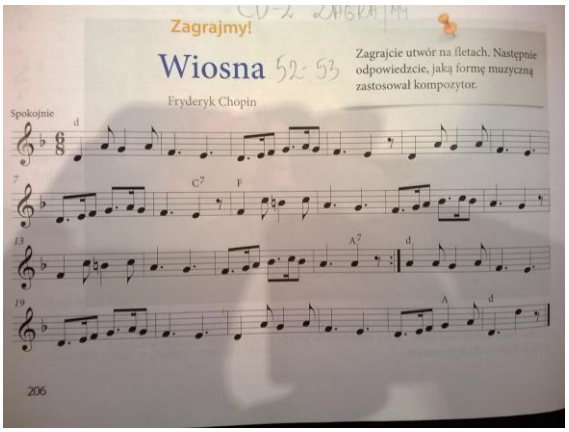
4 Experimental results

A test data set were collected comprising images presenting musical notation of varying difficulty. Images were taken from both music textbooks and specially prepared sheet music using MuseScore [16] program. Six musical pieces were tested: three prepared in the MuseScore and three taken from music manual. Processed examples had a variety of notes. Three photos were taken for each musical piece, containing two, three and four staves in a horizontal position. They were taken in daylight, during a dark whether, using a Nokia Lumia 730 Phone. This camera is equipped with a 6.7 megapixel sensor and Carl Zeiss optics. The pictures were taken at a resolution of 316 dpi. The size of photo files in JPEG format was about 1.3 MB.

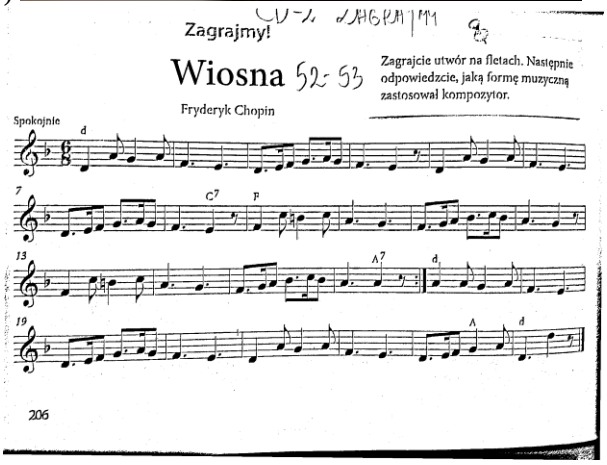
The quality of music recognition depends on the symbols to be recognized, number of linked notes, on the number of staves in the image and . The average recognition rate for all test examples was 78,85%. The errors concerned both: musical symbols and their pitch. It is worth noting, that the error rate for misclassified symbols is only 5,29%, i.e. the symbols are well defined by the rules.

Figure 9 a) presents a musical score of Fryderyk Chopin piece, that was the most difficult example to recognize. This is a photo of a book page – the staff lines are rounded, the background is not homogeneous – with the shadow of a photographer. Also the set of notes to be recognized is demanding: there are many linked eighths and sixteenths, notes with dots is a challenge. The intermediate steps of musical symbols recognition are presented in Figure 9b) and 9c).

a)



b)



c)

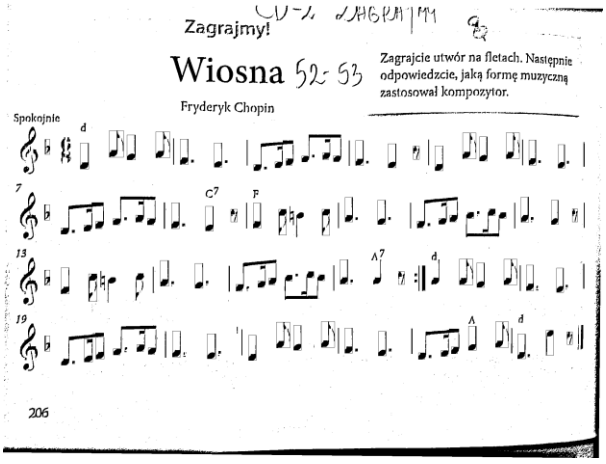


Fig. 9. The stages of the procedure of optical music recognition using smartphone illustrating a) the original photo, b) after preprocessing, c) at the stage of musical objects recognition.

However the overall recognition rate was quite high: 86,11% for two staves, 88,46% for three staves and 85,21% for four staves.

Basically, the more detailed testing has not been performed, as the proposed application is rather a proof of concept to be further developed in many aspects – first of all within the Image Processing Module, where the algorithms will be significantly changing. Also steps towards reducing processing time will be undertaken. Now the average data transfer time (from the device to the server and back) is approximaely 10 seconds, whereas the approximate processing time on the server (image and music) is 5 seconds. This is acceptable by users.

5 Conclusions

The goal of the project presented in this paper was twofold. First – to built a reliable multi-user mobile system to perform Optical Music Recognition and playing the recognized melody on the smartphone with any, chosen by the user, tempo and with a sound of a favorite musical instrument. The second goal was to test the new algorithm of the optical music recognition based on a decision tree and heuristically created rules describing symbols' shapes - no learning algorithms were needed to recognize musical symbols.

The proposed system with a modular architecture based on the client-server paradigm proved to be a correct solution. The implementation of the server performed in line with modern technologies providing its services via REST API ensures transparent communication with the mobile application. Server running on Microsoft Azure provides high degree of scalability. The Sound Generation Module is designed to support files MusicXML compatible, therefore data, from which it can generate sound, is not limited only to that from the system. It may process multiple music tracks and allows selection of musical instrument to be played and serve to other purposes, like e.g. style recognition.

The system is still being developed. The main directions are: enlarging the set of musical symbols that can be recognized, providing music documents e.g. in pdf format from recognized musical symbols, creating a music database in MusicXML format and adaptation to handwriting. The system architecture enables also rather easy implementation in other operational systems (Android and iOS).

Acknowledgements. Authors thank the reviewers for their insightful comments.

References

1. Alirezazadeh, F., Ahmadzadeh, M.R.: Effective staff line detection, restoration and removal approach for different quality of scanned handwritten music sheets. JACST 3(2) (2014)
2. AForge.NET. http://www.aforgenet.com
3. Audiveris. https://audiveris.kenai.com
4. Bainbridge, D.: Extensible Optical Music Recognition. Ph.D. thesis. Department of Computer Science, University of Canterbury, Christchurch, NZ (1997)

5. Bainbridge D.: An extensible optical music recognition system. In: Proceedings of the nineteenth Australasian Computer Science Conference, 308–317 (1997)
6. Bainbridge D, Bell T.: The challenge of optical music recognition. Comp.H. 35(2), (2001)
7. Dalitz C., Droettboom M., Pranzas B., Fujinaga I.: A comparative study of staff removal algorithms, IEEE Trans.PAMI, 30 (5), 753-766 (2008)
8. Fujinaga, I.: Optical Music Recognition Using Projections. M.Sc. thesis, McGill University, Montreal, Canada (1988)
9. Fujinaga, I. Adaptive Optical Music Recognition. Ph.D. thesis, Department of Theory, Faculty of Music, McGill University, Montreal, Canada (1997)
10. Homenda W.: Optical music recognition: the case study of pattern recognition. In: Kurzynski M. et al.. (eds.) Computer Recognition Systems. Advances in Soft Computing, vol 30. Springer, Heidelberg, 835–842 (2005)
11. Homenda W., Lesinski W.: Optical Music Recognition: Case of Pattern Recognition with undesirable and Garbage Symbols. In: Choras et al. (eds.) Image Processing and Communications Challenges. Exit.Warsaw. 120-127 (2009)
12. Lesinski W., Jastrzebska A.: Optical Music Recognition as the Case of Imbalanced Pattern Recognition: a Study of Single Classifiers. In: Skulimowski A.M.J. (ed.) Proc. of KICSS 2013. Progress&Business Publishers. Kraków. 267-278 (2013)
13. iSeeNotes. http://www.iseenotes.com/
14. NoteReader. http://www.musitek.com/mobile/
15. MusicXML, http://www.musicxml.com/.
16. MuseScore. https://musescore.org/pl
17. Ng, K.C., Cooper, D., Stefani, E., Boyle, R.D., Bailey, N.: Embracing the Composer: Optical Recognition of Handwritten Manuscripts. In: Proc. of the ICMC. 500–503 (1999)
18. Ng, K.: Optical Music Analysis for Printed Music Score and Handwritten Music Manuscript. In: George, S.E. (ed.) Visual Perception of Music Notation: On-Line and Off Line Recognition. IGI Global. 108–127. (2004)
19. Novotny J., Pokorny J.: Introduction to Optical Music Recognition: Overview and Practical Challenges. In: Necasky M.et al. (eds.): Datseo 2015, 121–130 (2015)
20. PhotoScore. http://www.neuratron.com/
21. Prerau, D.S.: Computer Pattern Recognition of Standard Engraved Music Notation. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA (1970)
22. Pruslin, D.: Automatic Recognition of Sheet Music. Sc.D. dissertation, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA (1966)
23. Rebelo A., Fujinaga I., et al.: Optical music recognition: state-of-the-art and open issues, Int. J. Multimed. Info. Retrieval. 1(3) 173-190 (2012)
24. RESTful Tutorial, http://www.restapitutorial.com/
25. Sharp Eye. http://www.visiv.co.uk/
26. SmartScore. http://www.musitek.com/.
27. Szwoch, M.: Guido: A Musical Score Recognition System, Document Analysis and Recognition. ICDAR 2007, Vol. 2, 809- 813 (2007)
28. Szwoch M.: A Robust Detector for Distorted Music Staves. In: Gagalowicz A., Philips W.(eds), CAIP, LNCS 3691, Springer, 701-708 (2005)
29. Baumann, S. and Dengel A. Transforming Printed Piano Music into MIDI, ASSPR World Scientific. (1992)
30. Rebelo A., Capela G.,·Cardoso J.S., Optical recognition of music symbols. A comparative study, IJDAR, 13(1):19–31, (2010)
31. Keon-Hee Park et al, Decision-Tree Algorithm for Recognition of Music Score Images Obtained by Mobile Phone Camera, The J. of the Korea Contents Ass. 8(6), 16-25,(2008)