J Inf Process Syst,
http://dx.doi.org/10.3745/JIPS

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS

A Lightweight and Effective Music Score Recognition on Mobile Phone

Tam Nguyen* and Gueesang Lee**

Abstract

Recognition systems of scanned or printed music scores which have been implemented on personal computer received consideration of many scientists and achieved significant results in many years. A modern trend with music scores captured and played directly on mobile devices become more interesting to researchers. The limit of resource and the effects of illumination, distortion, inclination on input images are still challenges to these recognition systems. In this paper, we introduce a novel approach to recognize music scores captured by mobile camera. To reduce the complexity as well as computational time of the system, we groups all symbols extracted from music score into ten main classes. Each major class is applied to SVM to classify music symbols separately. The experimental results showed that our proposed method could be applied to real time applications and the performance is competitive to others.

Keywords

Music Score, SVM, Mobile Camera, Symbol Classification

1. Introduction

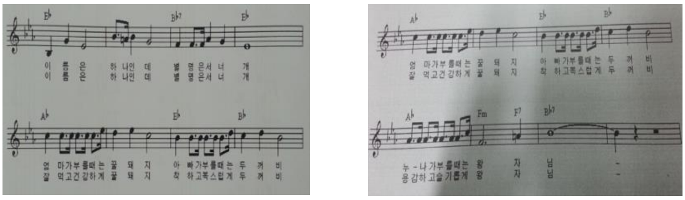
Nowadays, the explosion of mobility is setting a new standard for information technology industry. Mobile devices are not only limited in calling, or texting, but also cover a variety of entertainment such as multimedia applications. One of the entertainment applications, which are potential and interesting, is to play music scores captured directly from a mobile camera. In the early stages, systems for recognizing and playing OMR-based music scores on standalone PC have been introduced [3-4, 8-12]. However, such systems are relatively heavy because of adopting machine learning algorithms [1-2, 5-7] which could be difficult to be deployed on mobile devices with limited computational resources. Moreover, they only work well with music scores which are captured from scanners to guarantee the quality of the input data [1-2]. Meanwhile, music scores captured directly from the mobile camera could be affected by illumination, distortion, different viewpoints (in the Figure 1). Recently, music score recognition systems have been proposed on mobile phone and achieved potential results [16-17]. These systems could overcome some problems related to illumination. But as in PC-based approaches, they

still use machine learning to recognize all symbols which is known as un-lightweight approach to

Manuscript received: Month 06, 2014; accepted Month 06, 2015.
Corresponding Author: Gueesang Lee (gslee@jnu.ac.kr)
* Saigon Technology University, Ho Chi Minh City, Vietnam (nguyentamdtmc@gmail.com)
** Chonnam National University, Gwangju 500-070, Korea (gslee@jnu.ac.kr)

recognize music symbols. This may lead to increase the resource consumption and processing time on mobile devices.

In this paper, we introduce a novel and effective approach to recognize music scores on mobile devices. First, our method could deal with music scores captured directly from a mobile camera under the influences of environment acquisition, including distortion, illumination and different viewpoints. Unlike previous approaches, our system use some preprocess techniques such as global and local based binarization and line fitting method [15] to eliminate the effects of environment on input images. Moreover, to reduce the complexity, all symbols are grouped into ten main classes which have the similar features before applying SVM to recognize each symbol in these main classes.



(a) (b)

Fig.1. Music score which is scanned from printed sheet (a), a part of music score which is captured from mobile camera (b) with illumination and distort effects.

In summary, our contributions are:

- Our approach is applied global and local threshold based binarization and line fitting method to remove illumination effect, distortion and noise from input music scores to improve the performance of system.
- Heuristic characteristics are used to classify all symbols into ten subclasses with the similar features reduced the number of classes and the difficulty level for SVMs.

The rest of paper is organized as follows. In section 2, we present all related works about recognizing the music score recognition systems. Section 3 describes our proposed methods in details. The experimental results are presented in Section 4. Finally, Section 5 draws out the conclusion and future researches.

2. Related Work

Most approaches of music symbol recognition is that all symbols are split by staffs and the elementary graphic symbols such as note head, rest, dot, stem, tag [4-6, 11, 13] .Then using machine learning to recognize. In [9], the k-nearest neighbor rule is used while in [3, 6, 14] neural networks is the selected classifier. Choudhury et al. [4] proposed the extraction of symbol features, such as width, height, are, number of holes, and low-order central moments, whereas Taubman [12] preferred to extract standard moments, centralized moments, normalized moments, and Hu moments with the k-nearest neighbor method.

In [7, 8], the authors introduced various approaches that avoid the prior segmentation phase. They are superior methods in which, both segmentation and recognition steps are implemented simultaneously using Hidden Markov Models (HMMs). Features are extracted directly from images but this process is not only difficult to be carried out but also sensitive to errors.

Homenda and Luckner [10] used five classes of music symbols with two different classification approaches: classification with and without rejection. In case of leaving staff line and segmentation, [15] adds horizontal lines to extend to the top and bottom of the staves.

[16-17] are approaches on mobile devices. [16] applied on android platform and [17] on window phone 7. Both of two methods used learning machine to recognize music symbols from printed or camera captured music sheets

3. Proposed Method

With the above analysis, our proposed method provides a novel way to recognize music symbol which achieved by music scores captured from camera. Unlike to previous methods, we use two steps for recognition. The first, all symbols are classified into ten main classes with the similarity features (number of note heads, number of lines). Each of ten main classes includes some symbol subclasses. After that, SVM method is applied to classify separate symbols in each class.

3.1 The Overall Method

The figure 2 presents the overall model of our proposed method. With each symbol, it would be classified into one of ten major classes at first based on the heuristic criterions such as location symbols located, number of note heads, and number of vertical lines. These features are achieved with high accuracy by using symbol's proper information, projection, template matching and shift. Each main class contains one symbol or some kind of symbols. With later cases, nine SVMs are used to classify symbols into subclasses.

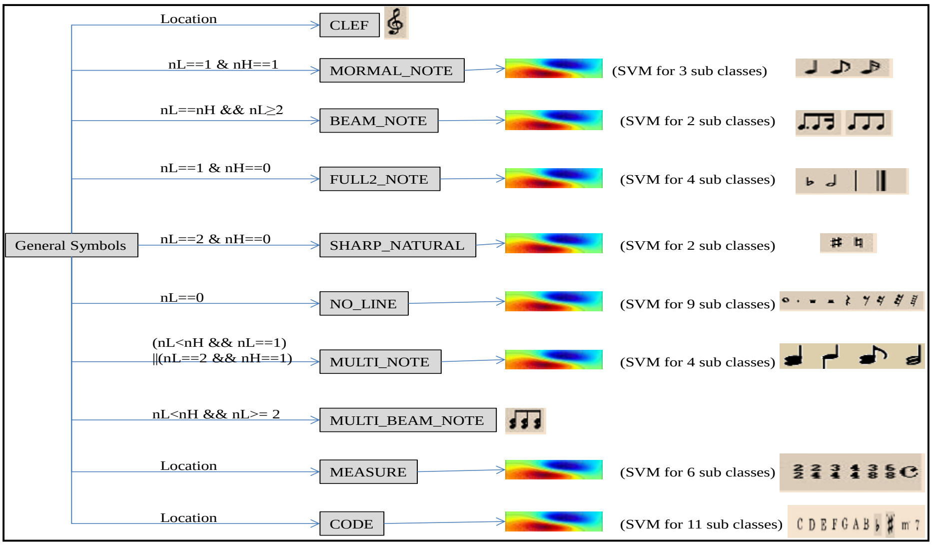


Fig. 2. The overall model of our proposed method

3.2 Preprocessing

This step implements two tasks, including image binarization and staff line information extraction.

Under the effect of illumination, in some cases, the background color of music image becomes swarthy. It is difficult to separate background from foreground if using only local or global threshold method [15]. Therefore, the combination of local and global threshold is used to binary image.

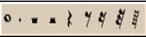
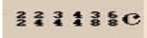
As be mentioned, the music sheets captured from mobile camera are often affected by distortion, illumination, different viewpoints. Therefore, the staff lines in the achieved binary image could be distorted and tilted. To resolve this issue, we use line fitting method [15] to detect and correct these staff line.

3.3 Ten Main Classes

To achieve ten classes from all symbols after segmentation, we base on the feature of symbols. Ten classes includes: CLEF, _NORMAL_NOTE, _BEAM_NOTE, _FULL2_NOTE, _SHARP_NATURAL , _NO_LINE, _MULTI_NOTE, _MULTI_BEAM_NOTE, _MEASURE, _CODE as in the table 1.

With general ideal, there are three special kinds of symbols: _CLEF, _MEASURE and _CODE. These independent on the number of note heads and number of lines and they have the fix locations on the music score. So after segmentation, with the information about the locations of these symbols, we can be easy to classify them into 3 main classes: the first class is _CLEF, the size of this symbol is large, the top and bottom boundary of cleft are outside the top and bottom of staff line and this symbol is located at the beginning of stave. The second class is _MEASURE, these symbols are located immediately after the clef. And the last is the _CODE class, the location of this class is above the staves, the bottom of the boundary always approximates the first line location in the staff.

Table 1. Ten classes and their features (nL: number of lines, nH: number of head notes)

Number	Name	Criterion	Symbol image
1	(CLEF)	Location	
2	(_NORMAL_NOTE)	nL==1 & nH==1	
3	(_BEAM_NOTE)	nL==nH && nL≥2	
4	(_FULL2_NOTE)	nL==1 & nH==0	
5	(_SHARP_NATURAL)	nL==2 & nH==0	
6	(_NO_LINE)	nL==0	
7	(_MULTI_NOTE)	(nL<nH && nL==1) ǁ (nL==2 && nH==1)	
8	(_MULTI_BEAM_NOTE)	nL<nH && nL>= 2	
9	(_MEASURE)	Location	
10	(_CODE)	Location	

The rest seven classes, they have the same feature is that they relate to the head-notes and lines. With the "_NO_LINE" class, the number of line equals to zero and include 6 subclasses, the "_NORMAL_NOTE" class includes three subclasses and the both number of lines and number of note heads equal to 1. The "_BEAM_NOTE" class has only two sub classes with the number of line equals to the number of head-notes and the number of lines has to more than 1. If the number of lines equals to 1 and the number of note heads is zero, four subclasses are classified into "_FULL2_NOTE" class. Sharp and natural are in the same class with the number of lines is 2 and the number of note heads is zero. "_MULTI_NOTE" has the number of lines being smaller than the number of note heads and the number of lines being 1 or the number of lines being 2 and the number of note heads being 1. "_MULTI_BEAM_NOTE" class has only one subclass. This class includes symbols which the number of lines is smaller than the number of note heads and the number of lines is larger than one.

The location of each symbol could achieve from symbol's information which is gathered in segmentation step. The issue remains is that how to determine the number of lines and the number of note heads for each symbol. Projection, template matching and shift are applied to resolve this problem.

3.3.1 Vertical Projection for Vertical Line Detection

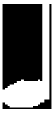
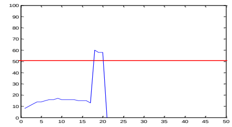
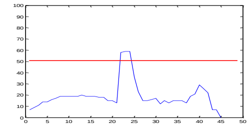
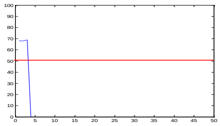
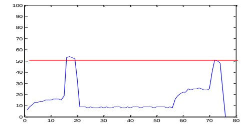
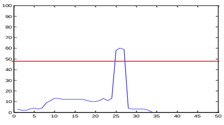
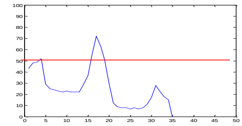
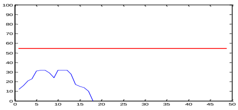
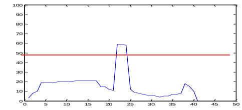
With each symbol image achieved after segmentation, we use vertical projection to get the histogram of this symbol, base on the feature of line with the number of pixels, we set some thresholds to find out the local peaks. By experimental, Threshold τ for vertical lines based on the height of staff line and the distance between lines is estimated by:

$$\tau = \alpha * rH + \beta * rS \tag{1}$$

where rH, rSr are the distance between lines and line height respectively. α, β are user-defined values.

According to the acknowledgement about music score' rules, vertical lines' length always smaller than bar-line (includes $4 * rH$ and $5 * rS$) and bigger than a half of bar-line. So in this work, α, β are selected as 2 and 3 respectively in practice to ensure that all vertical lines are collected precisely. Therefore, the interesting positions where black note or white note could appear are the vertical lines' location. As can be seen in the Table 2, the red horizontal line which is calculated from the Eq. (1) is the threshold for detecting the vertical line of symbols after vertical projection. With proper threshold, quarter note, bar line, sixteenth note, half note, eight note get one vertical line, multi note and beam note get two line and sharp with the number of line equaling to zero.

Table 2. The original symbol after segmentation (the 1st row) and the vertical projection of this symbol (the 2nd row)

Symbol	Projection	Symbol	Projection
 Quarter Note		 Sixteenth Note	
Bar Line		 Beam Note	
 Half Note		 Multi Note	
 Sharp		 Eight Note	

3.3.2 Head Note Detection

Template matching for black note head is applied for each position of vertical line. The template is shifted in both two sides of vertical line with step being equal to distance between staff lines (Figure 3. (a, b)).

(a) (b)

Fig. 3. The template of head note (a) and the positions of window for shifting (b)

After this step, the number of lines and the number of head notes for each symbol are achieved. All symbols are categorized into ten main classes for next process.

3.4 SVM for Music Symbol Recognition

After the first step, all symbols are divided into ten main classes, some classes such as multi beam note, cleft include only have one subclass. We can ignore these main classes. Therefore, we only use seven SVM for the rest seven main classes. Because SVM runs for each main class, the number of subclass for each main class is small and we can limit the symbols having the similar shapes after segmentation. So the performance would be improved.

Eight SVMs are used for the same number of main classes. Two main classes of "_BEAM_NOTE" and "SHARP_NATURAL" apply binary classification for recognize two sub classes with standard support vector machine algorithm. For the rest cases, because each main class contains more than two sub classes, the multi – class SVMs are implemented for recognition. With mobile devices, power is a serious issue. Therefore, to optimize our proposal method, we use one – against – one method for classifying [15]. It means that for k sub classes which need to classify, $\frac{k(k-1)}{2}$ classifiers where each one is trained on data from two classes are constructed. Given l training data $(x_1, y_1), \ldots, (x_l, y_k)$, where $x_i \in R^n, i = 1, \ldots, l$ and $y_i \in \{1, \ldots, k\}$ is the class of x_i. For training data from the i^{th} and j^{th} sub classes, the following binary classification problem is solved:

$$\min_{w^{ij}, b^{ij}, \varepsilon^{ij}} \frac{1}{2}(w^{ij})^T w^{ij} + C\sum_t \varepsilon_t^{ij}$$
$$(w^{ij})^T \emptyset(x_t) + b^{ij} \geq 1 - \varepsilon_t^{ij}, if\ y_t = i,$$
$$(w^{ij})^T \emptyset(x_t) + b^{ij} \leq -1 + \varepsilon_t^{ij}, if\ y_t = j, \tag{2}$$
$$\varepsilon_t^{ij} \geq 0$$

Where the training data x_i are mapped to a higher dimensional space by the function $\emptyset$ and C is the penalty parameter, ε_t^{ij} is the classification error and w^{ij}, b^{ij} are parameters in linear equation, respectively. Minimizing $\frac{1}{2}(w^{ij})^T$ is to maximize $2/\|w^{ij}\|$, the margin between two groups of data. $C\sum_t \varepsilon_t^{ij}$ is the term to reduce the number of training errors.

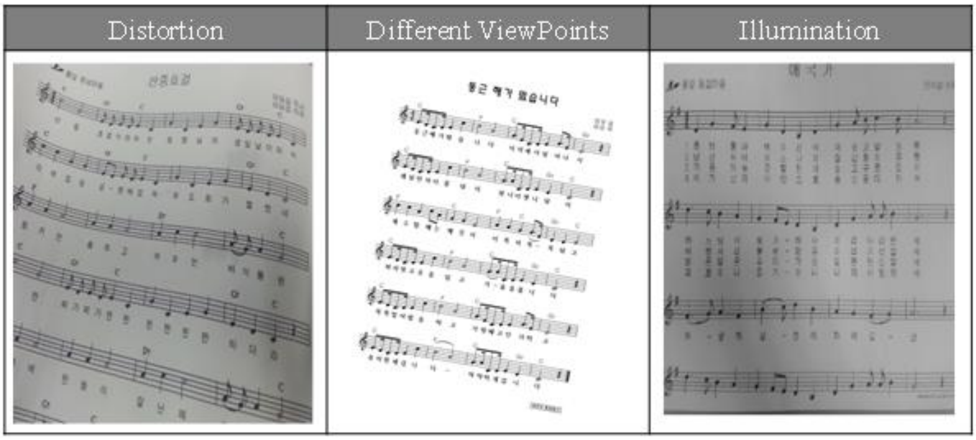


Fig.4. Music scores capture from camera

4. Experimental Results

To achieve the performance for music symbol recognition, we use 22 music scores which captured from the camera with various morphologies: difference view point, distortion, illumination in the Figure 4. Then, two scenarios are used for implementation.

After segmentation step, all symbols are gathered and categorized into ten main classes based on the number of head notes and the number of vertical lines. These symbol images are stored in ten folders corresponding to ten main classes. Based on the specific scenario, the number of symbols for training and testing is decided. Then SVMs are applied on the testing patterns to evaluate our proposed method's performance. Multi SVMs are used to multi main classes. Each class contains a small number of sub classes. Therefore, the time for training for each main class is reduced significantly. The probability of confusion among the kinds of symbols is cut down appreciably.

The scenario one uses 22 music scores to collect all of 2273 symbols. The number of patterns for training take two-third of the total symbols (1538 symbols) and the rest of them spend to test (735 symbols). The table 3 shows the detail number of symbols in each class as well as the total of testing and training number.

The table 4 shows the accuracy of scenario one. The first step of recognition is to classify all symbols into ten main classes using heuristic characteristics. This method is simple and fast to implement. Then SVMs are applied for each main class with the small number of subclasses. So the accuracy for each main class is very high as well as the average accuracy of our method (99.2%) in table 5.

We want to investigate the accuracy of our method in the difference scenes to evaluate the performance. So in the scenario 2, we still use 22 music scores captured from camera but 12 scores are used for training and the rest numbers are used for testing. The detail number of patterns for each main class as well as subclass and the accuracy (for each main class and average) is showed in the table 6 and table 7.

Table 3. Symbols and the total number of them for classification

Main Classes	Sub Classes	Total Number
NORMAL_NOTE	Quarter Note	320
	Eight Note	267
	Sixteenth Note	116
BEAM_NOTE	Double	67
	Single	52
FULL2_NOTE	Flat	41
	Half Note	59
	Bar Line	196
	Double Bar Line	22
SHARP_NATURAL	Sharp	48
	Natural	48
NO_LINE	Whole Note	18
	Dot	196
	Whole Rest	20
	Eight Rest	21
	Sixteenth Rest	36
MEASURE	Time Signature: 2	8
	Time Signature: 3	6
	Time Signature: 4	12
	Time Signature: 6	6
	Time Signature:8	8
	Time Signature: C	2
CODE	Chord A	45
	Chord B	35
	Chord C	91
	Chord D	62
	Chord E	63
	Chord F	59
	Chord G	95
	Chord m	66
	Chord 7	74
	Flat	54
	Sharp	60

Table 4. The scenario 1: The accuracy for each main class

Main Classes	Sub Classes	Accuracy
NORMAL_NOTE	Quarter Note	99.12%
	Eight Note	
	Sixteenth Note	
BEAM_NOTE	Double	100%
	Single	
FULL2_NOTE	Flat	98.13%
	Half Note	
	Bar Line	
	Double Bar Line	
SHARP_NATURAL	Sharp	100%
	Natural	
NO_LINE	Whole Note	97.75%
	Dot	
	Whole Rest	
	Eight Rest	
	Sixteenth Rest	
MEASURE	Time Signature: 2	100%
	Time Signature: 3	
	Time Signature: 4	
	Time Signature: 6	
	Time Signature:8	
	Time Signature: C	
CODE	Chord A	99.43%
	Chord B	
	Chord C	
	Chord D	
	Chord E	
	Chord F	
	Chord G	
	Chord m	
	Chord 7	
	Flat	
	Sharp	

Table 5. The average accuracy of our method in scenario one

Training Number	Testing Number	Total	The Accuracy
1538	735	2273	99.20%

Table 6. The scenario 2: the accuracy for each main class

Main Classes	Sub Classes	Accuracy
NORMAL_NOTE	Quarter Note	97.89%
	Eight Note	
	Sixteenth Note	
BEAM_NOTE	Double	94.59%
	Single	
FULL2_NOTE	Flat	92.42%
	Half Note	
	Bar Line	
	Double Bar Line	
SHARP_NATURAL	Sharp	100%
	Natural	
NO_LINE	Whole Note	92.42%
	Dot	
	Whole Rest	
	Eight Rest	
	Sixteenth Rest	
MEASURE	Time Signature: 2	96.30%
	Time Signature: 3	
	Time Signature: 4	
	Time Signature: 6	
	Time Signature:8	
	Time Signature: C	
CODE	Chord A	93.85%
	Chord B	
	Chord C	
	Chord D	
	Chord E	
	Chord F	
	Chord G	
	Chord m	
	Chord 7	
	Flat	
	Sharp	

Table 7. The average accuracy of our method in scenario two

Training Number	Testing Number	Total	The Accuracy
1538	735	2273	95.36%

With the scenario two, the training and testing symbols are collected independently from 22 music scores (12 scores for training and 10 scores for testing). So the symbols for testing can be different greatly to the training. Therefore, the accuracy is smaller than scenario one. With more training, the accuracy would be higher.

To compare to the previous methods (using learning machine methods to recognize all symbols directly), we run SVM, NN with our database (according to [1], SVM and NN are the best method for scanned and printed music score recognition). The accuracy in the Figure 5 shows that our proposal was the highest performance for music scores captured from the camera.

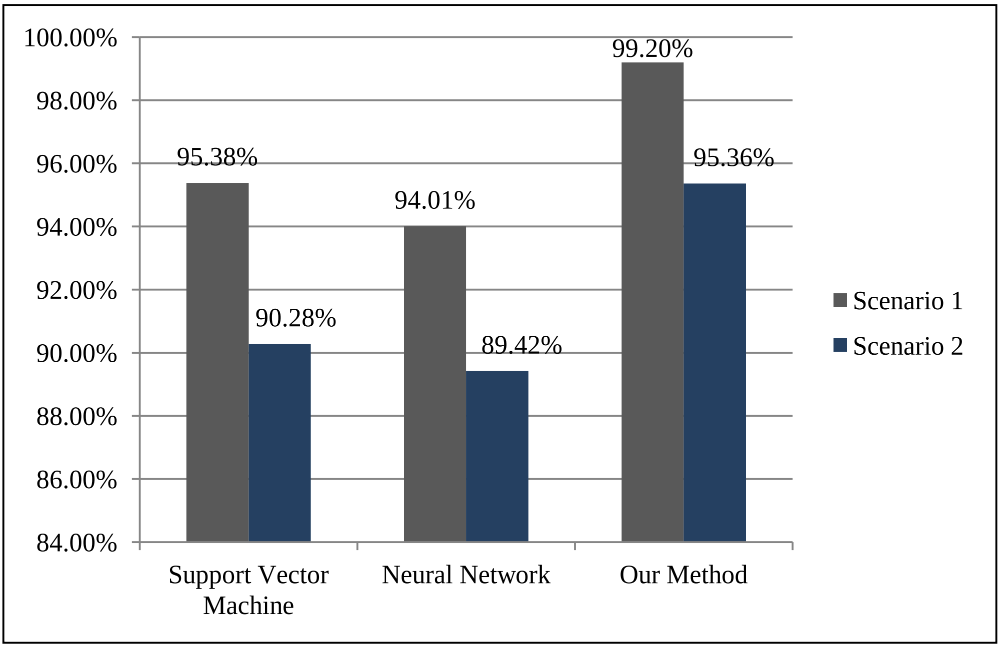


Fig.5. The comparison between three methods in accuracy

5. Conclusion

In this paper, we introduced a new approach to recognize music symbols extracted from the music scores captured from camera with different view point, distort, illumination. The recognition process include heuristic acknowledges for splitting all symbols into subclasses before using SVM. The experimental results showed the good performance of our method. Based on the way to choose the set of training and testing, the accuracy could change. In the next work, we continue to research about this field to improve the performance much higher for all scenes.

Acknowledgement

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MEST)(2014-024950). Also this work (Grants No. SM122965) was supported by Business for Cooperative R&D between Industry, Academy, and Research Institute funded Korea Small and Medium Business Administration in 2014.

References

[1] Rebelo et al., "Optical recognition of music symbol: A comparative study," *In Int. J. Doc . Anal. Recognit*, pp. 19-31, 2010.

[2] P. Bellini, I. Bruno, and P. Nesi, "Opticalmusic recognition: architecture and algorithms," *Interactive multimedia music technologies*, pp 80–110, 2008.

[3] G. Choudhury et al., “Optical music recognition system within a large scale digitization project,” *Proceedings of the International Society for Music information retrieval*, 2000.

[4] M. Droettboom, I. Fujinaga, and K. MacMillan, “Optical music interpretation,” *IAPR*, pp 378–386, 2002.

[5] H. Miyao and Y. Nakano, “Note symbol extraction for printed piano scores using neural networks,” *IEICE Trans Inform Syst*, 1996.

[6] L. Pugin, “ Optical music recognition of early typographic prints using Hidden Markov models,” *Proceedings of the International Society for Music*, pp 53–56, 2006.

[7] L. Pugin, J.A. Burgoyne, and I. Fujinaga, “MAP adaptation to improve optical music recognition of early music documents using Hidden Markov models,” *Proceedings of the 8th International Society for Music*, pp 513–516, 2007.

[8] S. E. George, and D. Sammon, *Visual perception of music notation: online and offline recognition*, Idea Group Inc., Hershey, pp 1–39, 2004.

[9] W. Homenda, and M. Luckner, “Automatic knowledge acquisition: recognizing music notation with methods of centroids and classifications trees,” *Proceedings of the international joint conference on neural networks*, pp 3382–3388, 2006.

[10] J. TardónL et al., “Optical music recognition for scores written in white mensural notation,” *EURASIP*, 2009.

[11] G. Taubman, “Musichand: a handwritten music recognition system,” *Technical report*, 2005.

[12] K.T. Reed, and J.R. Parker., “Automatic computer recognition of printed music,” *ICPR*, vol 3, pp 803–807, 1996.

[13] P. Bellini, I. Bruno, and P. Nesi, “Optical music sheet segmentation,” *Proceedings of the first international conference on web delivering of music*, pp 183–190, 2001.

[14] G. Abraham, and B. I. P. Rubinstein, *Proceedings of the Second Australian undergraduate students’ computing conference*, University of Mebourne, pp 1–7, 2004.

Tam Nguyen
She received B.S degree in School of Electronics and Telecommunications from Hanoi University of Sciences and Technology, Vietnam in 2011. Since 2012, she has been taking the M.S. course in Electronics & Computer Engineering at Chonnam National University, Korea. Her research interests are mainly in the field of Image Processing, Computer Vision.

Gueesang Lee
He received the B.S degree in Electrical Engineering from Seoul National University in 1980.In 1982 He received the M.S degree in Computer Engineering from Seoul National University. In 1991, He received Ph.D. degree in Computer Science from Pennsylvania State University. He is currently a professor of the Department of Electronics and Computer Engineering in Chonnam National University, Korea. Research Interests: Image processing, computer vision and video coding.