http://dx.doi.org/10.5392/IJoC.2014.10.4.011

Camera-based Music Score Recognition Using Inverse Filter

Tam Nguyen
Faculty of Information Technology
Saigon Technology University, Ho Chi Minh City, Vietnam

SooHyung Kim, HyungJeong Yang, GueeSang Lee*
Dept. of Electronics and Computer Engineering
Chonnam National University, Gwangju, South Korea

ABSTRACT

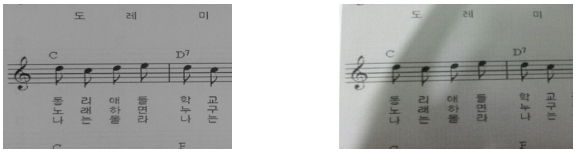
The influence of acquisition environment on music score images captured by a camera has not yet been seriously examined. All existing Optical Music Recognition (OMR) systems attempt to recognize music score images captured by a scanner under ideal conditions. Therefore, when such systems process images under the influence of distortion, different viewpoints or suboptimal illumination effects, the performance, in terms of recognition accuracy and processing time, is unacceptable for deployment in practice. In this paper, a novel, lightweight but effective approach for dealing with the issues caused by camera based music scores is proposed. Based on the staff line information, musical rules, run length code, and projection, all regions of interest are determined. Templates created from inverse filter are then used to recognize the music symbols. Therefore, all fragmentation and deformation problems, as well as missed recognition, can be overcome using the developed method. The system was evaluated on a dataset consisting of real images captured by a smartphone. The achieved recognition rate and processing time were relatively competitive with state of the art works. In addition, the system was designed to be lightweight compared with the other approaches, which mostly adopted machine learning algorithms, to allow further deployment on portable devices with limited computing resources.

Key words*: Music Scores, Staff line Detection, Note, Stem, Note Head, Projection, Inverse Filter.*

1. INTRODUCTION

Music Score Recognition has been an interest field recently. There are many systems recognizing and playing music scores achieved from the scanner or handwritten. There are a set of steps from reading input music scores to playing them in MIDI format. The music symbol recognition plays an important role which greatly impacts on the performance of the complete system.

Far away, there are many methods recognizing music symbols in a music score. However, the recognition process based on machine learnings using Support Vector Machine (SVM) [1], Hidden Markov Model (HMM) [7], [8], Neural Network (NN) [3], [6], [14], K Nearest Neighbor (KNN) [9], etc. are implemented after all music symbols are segmented into separated parts. Moreover, input data of above methods are scanned from a printed music score, providing clear images for the recognition. They are not affected by environment conditions such as distortions, or illuminations. Hence, symbol images are flagrant to be recognized. The recognition process is applied after all staff lines are removed or ignored. In more details, staff line removal leads to problems of fragmentation and deformation of music symbols. In case of staff line ignorance, the recognition capability is lower [15]. All methods mentioned above do not provide a sufficient performance to be commercialy used. In [15], the authors introduced another approach called as music score defacement. Additional horizontal lines are firstly placed exactly halfway between the existing staff lines and then, be extended to the top and bottom of the core at a half of staff line's width for the height of the score. Although the accuracy of some specific notes such as head note and whole note is increased because of creating same appearances for same notes, this method has some undesirable attributes in certain typesets, where it could partially obscure musical objects such as beams, slurs, hold dots.



(a) (b)

Fig 1. a part of 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.

* *Corresponding author, Email: gslee@chonnam.ac.kr*
Manuscript received Jun. 25, 2014; revised Oct. 23, 2014; accepted Oct. 30, 2014

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, where users could use resources on their portable devices to create and replay favorite melodies. Music scores captured from mobile camera are getting more and more popular. However, they are mostly affected by environmental conditions such as distort, illumination and different viewpoints, etc (in the Fig. 1). All above methods, the input images are scanned or printed music scores with clear content that are created under laboratory conditions. Therefore if these methods are applied to recognize such scores captured from mobile camera, it is easy to release that the performance is not unacceptable. In this paper, we propose a novel method to 1) deal with images captured from mobile camera and 2) adapt with limited computational resources on mobiles. As above analysis, because of noise, distort, illumination after segmentation, results include some unfavorable information remained in the symbol images, these can get wrong classification. To recognize major music symbols including black note, white note, stem, bar line, whole note, tags, beam, dot, pitch from mobile-captured music score, we implemented a lightweight method in which, staff lines are remained and symbols are recognized by heuristic acknowledgements instead of using machine learning algorithms as previous studies [1], [3], [6]-[9], [14] to reduce the computational complexity. After binarizing images, the vertical lines including stems and bar lines are detected by using horizontal projection. Template matching based on inverse filtering is used to determine position of black and white note heads by sliding a window according to the vertical lines. The rest of music symbols including the tags, beam are straightforward to be detected using run length code and matching. Whole note is located using distance between two bar lines with template matching and hold filling. Moreover, the pitch of each note is easily released based on the position of note heads. Finally, to increase the performance of template matching methods, we use inverse filter to create the artificial templates that are supposed to be influence by environment conditions. Such templates will be used to match with the images captured directly by mobile camera.

In summary, the main contributions of our study are:

1) Our approach is applied on input images captured from camera with distort, illumination, different viewpoints. Until now, all music symbol recognition systems are deployed on standalone powerful computer, so a lightweight music score architecture running on mobile devices with limited computational capability is expected. The problems of input images captured from mobile camera could be solved using our proposed method.
2) The performance is improved through detecting and recognizing music symbols with template matching by inverse filtering. Staff line is remained so problems of fragmentation and deformation are eliminated. Inverse filter is used to create templates for matching which are closest to the real data.
3) The processing speed is significantly increased since in this study, instead of using machine learning which requires huge computations, we use template matching created by using Inverse filter, projection, run length code to reduce the time processing. This is ready necessary when this system is directly run on mobile devices.

The rest of paper is organized as follows. In section 2, we present all related works about 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 WORKS

In most approaches for music score recognition, main steps are split into two parts: staff line detection and music symbol recognition for noteheads, rest symbols, dot, stem, and tags [4]-[6], [11], [13]. Then classification phase is followed with various methods using features extracted from the projection profiles. In [9], the k-nearest neighbor is used while in [3], [6], [14] neural networks are used. 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 segementaion phase. In such methods, 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. So, music scores are required to be very simple to be suitably applied.

Homenda and Luckner [10] used five classes of music symbols with two different classification approaches: classification with and without rejection. Rebelo et al. [1] compared four classification methods including Support Vector Machines (SVMs), Neural Networks (NNs), Nearest Neighbor (kNN) and Hidden Markov Models. The result shows that SVMs gave the best performance but the performance is not improved in the case of the elastic deformation. This could be result in some issues: the diverse dataset of symbols, improper features extracted, and inappropriate distortions.

In the case of staff line segmentation, [15] adds horizontal lines to extend to the top and bottom of the stave. This method improves the recognition accuracy of some symbols (e.g. head note, whole note) but it also causes difficulties to recognize the rest of components (e.g. beams, slurs, hold dots).

3. PROPOSED METHOD

3.1 Inverse Filter Transformed Template

Template is a pattern used to find small parts of the image which match with it. The more standard the template is, the higher the template matching performance is. Specially, in music scores captured by mobile camera, there are many effects from environmental conditions such as light, illumination, distort, view point. To get the good matching result, the template should be closest to the real sample.

To resolve this issue, we use inverse filter to achieve the template with features being nearest to the real data. Suggest $G(u,v)$ is the template we create normally, $F(u,v)$is the template we want to restore (all parameters are suggested in frequency domain). The degradation model $G(u,v)$ is determined by

$$G(u,v) = F(u,v)H(u,v) + N(u,v) \quad (1)$$

To achieve H(u, v), we take a sample from the music score as F(u, v) and obtain H(u, v) by transforming Eq. (1) to

$$H(u,v) = G(u,v)/F(u,v) \quad (2)$$

We consider noise as zeros to get H(u, v) - called degradation function. After H(u, v) is determined, an estimate $\check{F}(u,v)$ of the transformation of the original image is calculated by simply dividing the transform of the degraded image G(u, v)by the degradation function

$$\check{F}(u,v) = \frac{G(u,v)}{H(u,v)} \quad (3)$$

The divisions are between individual elements of the functions, as explained in connection with Eq. (1). Substituting the right side of Eq. (1) by G(u, v)in Eq. (2) yielding

$$\check{F}(u,v) = F(u,v) + \frac{N(u,v)}{H(u,v)} \quad (4)$$

Because N(u, v) is a random function whose Fourier transform is not known. If the degradation has zeros or very small values, then the ratio N(u, v)/H(u, v) could easily dominate the estimation of $\check{F}(u,v)$. To avoid the side effect of enhancing noise, we can apply Eq. (3) to frequency component (u, v) within a certain radius from the center of H(u, v).

3.2 Major Music Components Detection and Recognition

Major music components including stems, black note head, white note head, whole note and dot are detected and recognized by heuristic acknowledges. After staff line is detected, based on the information of staff line, vertical lines in each staff are determined using vertical projection. The threshold τ for vertical lines based on the height of staff line and the distance between lines is estimated.

$$\tau = \alpha\, rH + \beta\, rS \quad (5)$$

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

In this work, α, β are selected as 2 and 3 respectively in practice to ensure that all vertical lines are collected precisely Fig. 3.

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 Fig. 2. (b, c). For white note head, after finding out all black note head, the rest of vertical lines are checked for holes around them and matched to black note head after filling holes Fig. 2.d.

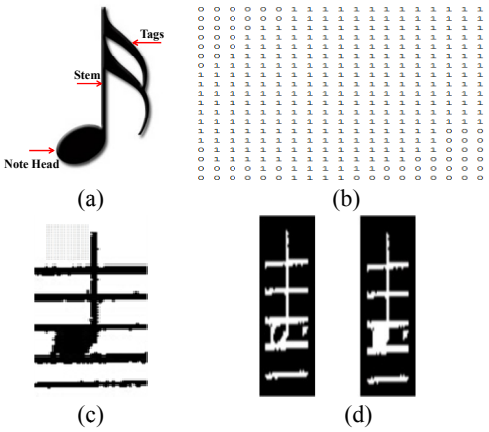


Fig 2. The major music components (a), the template of head note and the positions of window for shifting (b, c), filled white note (d).

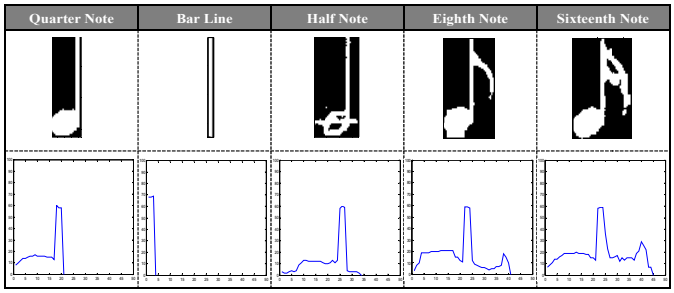


Fig 3. The vertical projection of some music symbols.

To detect whole note, we bases on the characteristics of whole note in the music score. If whole note exists, it picks up entire measure Fig. 4.(a). Therefore, after determining black and white note, the rest of vertical lines are bar lines. If there is not any black or white note between two bar lines, the whole note could appear. To detect the position of whole note, we extract the measures which do not have black or white note. The position of whole note is detected by determining holes, then we use template matching to recognize whole note at the position of filled holes.

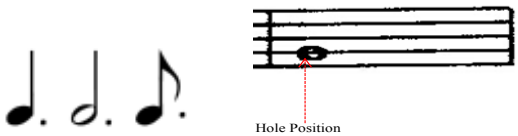


Fig 4. The measure of music score with a whole note and the hole position of this note (a) and the dot's position (b).

To recognize dot symbol in the music score, we consider its position. According to the music score's rule, the dot symbols always are located immediately after and middle of the note head (both black and white note) Fig. 4.(b) Therefore, based on the information about the position of note head determined above, the interesting area around behind the head

note is extracted. Run length code is used to find out the black regions with the size of height and width approximating rS.

3.3 Pitch and Other Components

3.3.1 Pitch Detection: The main target of music symbol recognition is that we need to determine which type of symbol belongs to and which position a symbol is located. Therefore, pitch detection after recognition is an important step in a music symbol recognition system.

To determine the pitch, we base on the position of head note recognized in the above step. After getting the center position of note head, a run length code is applied to find out the above and bottom boundary of note head. Then center position of note head is calculated again by taking average of above and bottom boundary according to vertical line Fig. 5.(a). A reference map that covers all possible positions of note head is created. The lowest pitch is -2 corresponding to lowest position and the highest pitch is 14 corresponding to the highest position on the staff line. Each pitch level has a step which equals to a half of the distance between staff lines. Each note has a pitch corresponding to the position of staff line being nearest to the center of note head Fig. 5.(b).

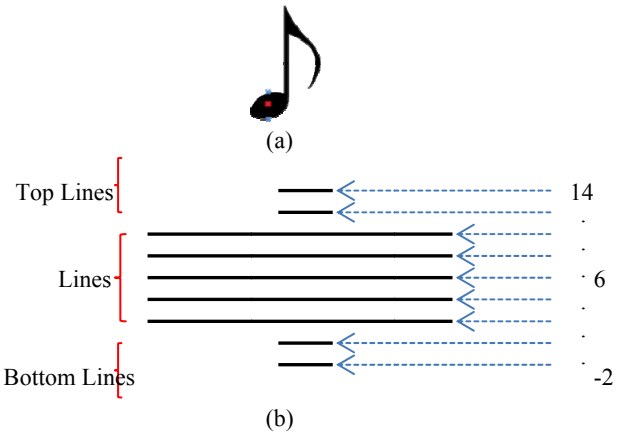


Fig 5. The above and bottom boundary of note head and the its center (a) and the pitch map for pitch detection (b).

3.3.2 Other Components: Other components of major symbols in a music score include tags, beam. In this paper scope, we only focus on tag detection. To detect tags and recognize how many tags are assigned to one note symbol, we firstly extract the part can have tags based on the center position of note head Fig. 6. We use run length code for right side of stem to find out the tags with the thickness of tags being larger than the height of lines. To verify the tags again, tag template matching is applied with all notes satisfying above condition.

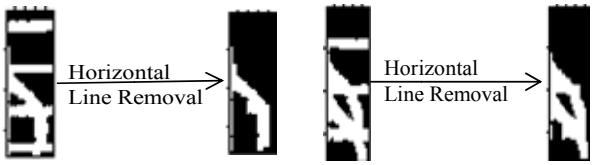


Fig 6. The tags of note symbol in music score when remains the horizontal lines and without horizontal lines.

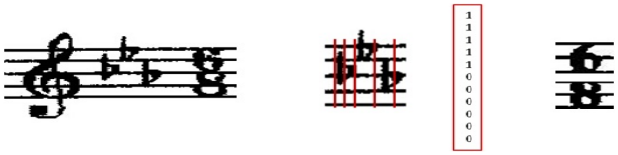


Fig 7. The beginning of each frame includes clef, time signature and key signature (a), the time signature and the run length code (b, c), the detected key signature (d).

Besides, clef and time signature are straightforward to be detected. Clef is always located at the beginning of each staff line, and its height is always higher than the height of staff line. The time signature is located either after clef or after key signature Fig. 7.(a). The vertical projection of key signature takes a part of staff line whereas the time signature takes all the height of staff line Fig. 7.(b-d).

4. EXPERIMENTAL RESULT

4.1 Dataset

We captured 37 images of music scores using Galaxy Note 2 mobile phone with various morphologies in Fig. 8. They have the different size of shape, illumination, distort and view point, etc. Totally, they include 1920 black notes, 118 white notes, 1515 tags (both single tags and double tags), 2038 stems, 15 whole notes, 430 dot and 2038 notes needed to determine the pitch. The total of symbols which needs to be recognized is up to 8074 patterns Table 1.

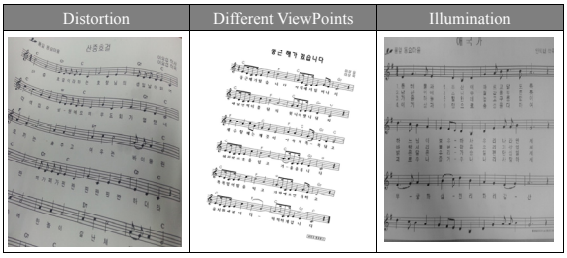


Fig 8. Music scores capture from camera Galaxy Note 2

Table 1. The components of dataset for evaluation

	Black Note	White Note	Stem	Tags	Whole Note	Dot	Pitch
Symbols							
The number of patterns	1920	118	2038	1515	15	430	2038

4.2 Performance

The result in Table 2 shows that black notes, white notes, whole note and stem have an accuracy of 100% with total of note being up to 2038. The tag detection gets the result of 99.93% with only one fail in total of 1515 patterns. And in pitch recognition, the result is achieved with a percentage of 99.55%. There are 9 fails in total of 2038 patterns. The dot detection is sufficient accuracy of 96.05%. From the experiments, we

release that these failures are caused by noise with high level in the input images.

Table 2. The accuracy of for basic symbols in music score image captured from camera.

Pitch		Black Note		White Note		Whole Note		Stem		Tag		Dot	
Correct	Total	Correct	Total	Correct	Total	Correct	Total	Correct	Total	Correct	Total	Correct	Total
2029	2038	1920	1920	118	118	15	15	2038	2038	1514	1515	430	413
99.55%		100%		100%		100%		100%		99.93%		96.05%	

With the same dataset of 37 music scores captured by mobile camera, we redid previous studies using machine learning as [1], [3] and then, make a comparison between such studies with our proposed method. To ensure a fair comparison, we only recognize black notes, white note, whole note, stem and tags. With a total of 6036 patterns (except the 2038 patterns of pitch recognition), we take two third patterns for training and one third patterns for testing. The detail is showed in Table 3.

The Fig. 9 shows the comparison result among methods. Because input music scores are images captured from mobile phone camera, each image has different view point, level of illumination, distort. Therefore, the same music symbol images achieved after segmentation step have inconsistent shape, size, and features. The number of trainings with 4025 symbols is not enough to cover all cases of input image. Moreover, the segmentation step causes the fragmentation and deformation of symbols. All above reasons lead to a lower accuracy of SVM and NN for recognizing components of music symbol in a music score. In detail, the whole note recognition result of NN is lowest with the accuracy of 80% and that of SVM is 90%. The highest accuracy which SVM gets is 99.7% for stem recognition with the number of trainings being very big (1359 patterns). That of NN is also 98.83 % for tag recognition with the number of trainings being up to 1010 patterns. Whereas, due to implement a new approach which is overcome weaknesses of previous studies, our method remains an accuracy of 100% for black note, white note, whole note, stem recognition and 99.93% for tag recognition. In the case of tag recognition, there is one failure. Because the level of noise in input image is high. This leads to the staff line being too thick to distinguish from tag. In the dot detection, both SVM and NN get a good performance (approximate 95%) but false acceptance rate is high.

Objectively, the average accuracy of three methods are calculated and illustrated in the Fig. 10.

The whole system has been tested on about 100 music scores taken from mostly elementary school textbook of music classes. It took minimum 376ms, maximum 1758ms and average 702ms.

4.3 Computation Complexity

Besides, our method execute in the short time for each image (average with 1.5135 seconds) on the mobile phone (Galaxy S3, Note II). While the time complexity of standard SVM training T_{SMV} is calculated by

$$T_{SMV} = O(dn^2) \quad (6)$$

where d, n are the number of dimensional and the number of training patterns respectively. Therefore, with the number of training is up to 4025 patterns and the number of features is up to 20x20, the time for training takes a huge amount of time. NN with neural network structure takes more time than SVM.

Table 3. The statistic of testing and training number

	Black Note	White Note	Whole Note	Stem	Tag	Dot	Total
Testing	640	39	5	679	505	143	2011
Training	1280	79	10	1359	1010	287	4025

5. CONCLUSION

In this paper, we introduced a novel approach to recognize music symbols extracted from the music scores captured by mobile camera with different view point, distort, illumination and noise. To get a desired performance, we do not follow the previous methods in which staff lines are removed or more horizontal lines are added into stave. We remain all staff lines after detecting and restoring its information. With the prior knowledge about music symbol as well as the effective way to create template for matching by inverse filtering, our method shows higher performance compared with previous works. This work in this paper is a well-known incorporation's project and gets good judgments .In the future, we would continue to research in this field to recognize entire symbols appearing in the music score and establish a complete framework for music symbol recognition system.

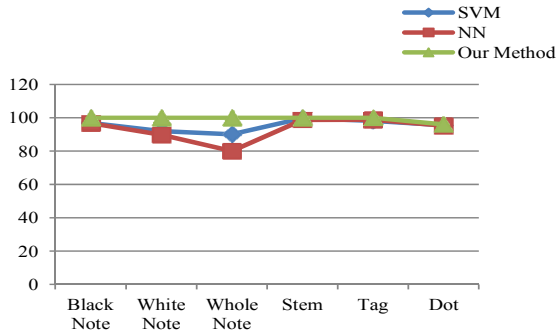


Fig 8. The comparison between our method with other methods.

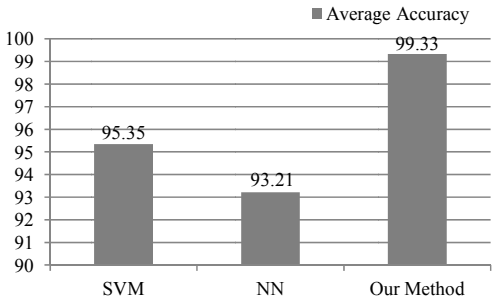


Fig 9. The average accuracy of our method and other methods.

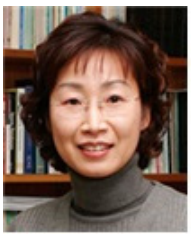
ACKNOWLEDGEMENT

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (2014-024950) and by the technological innovation R&D Program of SMBA(S2173771).

REFERENCES

[1] Rebelo et al., "Optical recognition of music symbol: A comparative study," In Int. J. Doc. Anal. Recognit, 2010, pp. 19-31.
[2] P. Bellini, I. Bruno, and P. Nesi, "Opticalmusic recognition: architecture andalgorithms," In Interactive multimedia music technologies, 2008, pp. 80-110.
[3] G. Choudhury et al., "Optical music recognition system within a large scale digitization project," In Proceedings of the International Society for Music information retrieval, 2000.
[4] M. Droettboom, I. Fujinaga, and K. MacMillan, "Optical music interpretation," In *IAPR*, 2002, pp. 378-386.
[5] H. Miyao and Y. Nakano, "Note symbol extraction for printed piano scores using neural networks," In IEICE Trans Inform Syst, 1996.
[6] L. Pugin, " Optical music recognition of early typographic prints using Hidden Markov models," In Proceedings of the International Society for Music, 2006, pp. 53-56.
[7] L. Pugin, J. A. Burgoyne, and I. Fujinaga, "MAP adaptation to improveopticalmusic recognition of early music documents using Hidden Markov models," In Proceedings of the 8th International Society for Music, 2007, pp. 513-516.
[8] I. Fujinaga, "Staff detection and removal," In George S (ed) Visual perception of music notation: online and offline recognition, Idea Group Inc., Hershey, 2004, pp. 1-39.
[9] W. Homenda and M. Luckner, "Automatic knowledge acquisition: recognizing music notation with methods of centroids and classifications trees," In Proceedings of the international joint conference on neural networks, 2006, pp. 3382-3388.
[10] J. TardónL et al., "Optical music recognition for scores written in white mensural notation," In EURASIP, 2009.
[11] G. Taubman, *Musichand: a handwritten music recognition system*, In technical report, 2005.
[12] K. T. Reed and J. R. Parker, "Automatic computer recognition of printed music," In ICPR, vol. 3, 1996, pp. 803-807.
[13] P. Bellini, I. Bruno, and P. Nesi, "Optical music sheet segmentation," In Proceedings of the first international conference on web delivering of music, 2001, pp. 183-190.
[14] S. Sheridan and S. George, " Defacing music score for improved recognition," In Abraham G, Rubinstein BIP (eds) Proceedings of the Second Australian undergraduate students' computing conference, 2004, pp. 1-7.
[15] Vo Quang Nhat and GueeSang Lee, "Adaptive line fitting for staff detection in handwritten music score images," ICUIMC 2014.
[16] Nawapon Luangnapa, et al., "Optical Music Recognition on Android Platform," Advances in Information Technology, Springer Berlin Heidelberg, 2012, pp. 106-115.
[17] Thanachai Soontornwutikul, et al., "Optical Music Recognition on Windows Phone 7," In The 9th International Conference on Computing and Information Technology (IC2IT2013), Springer Berlin Heidelberg, 2013.

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. She is currently a lecturer in Saigon Technology University, HoChiMin City, Vietnam. Her research interests are mainly in the field of Image Processing and Computer Vision.



Hyung-Jeong Yang
She received her B.S., M.S. and Ph.D from Chonbuk National University, Korea. She was a Post-doc researcher at Carnegie Mellon University, USA. She is currently an associate professor at Dept. of Electronics and Computer Engineering, Chonnam National University, Gwangju, Korea. Her main research interests include multimedia data mining, pattern recognition, artificial intelligence, e-Learning, and e-Design.

Soo-Hyung Kim
He received his B.S. degree in Computer Engineering from Seoul National University in 1986, and his M.S. and Ph.D degrees in Computer Science from Korea Advanced Institute of Science and Technology in 1988 and 1993, respectively. From 1990 to 1996, he was a senior member of research staff in Multimedia Research Center of Samsung Electronics Co., Korea. Since 1997, he has been a professor in the Department of Computer Science,

Chonnam National University, Korea. Research interests: Pattern recognition, image processing, image processing and ubiquitous computing.

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.