Optical Music Recognition for Traditional Thai Sheet Music

Worapan Kusakunniran¹, Attapol Prempanichnukul², Arthid Maneesutham³, Kullachut Chocksawad⁴, Suparas Tongsamuï⁵, and Kittikhun Thongkanchorn⁶
Faculty of Information and Communication Technology, Mahidol University, Thailand
Email: worapan.kus@mahidol.ac.th, attapol.pre@student.mahidol.ac.th, arthid.man@student.mahidol.ac.th³, kullachut.cho@student.mahidol.ac.th⁴, suparas.ton@student.mahidol.ac.th⁵, kittikhun.tho@mahidol.ac.th⁶

Abstract—Optical music recognition (OMR) is a system to transform a sheet music into a format readable by a machine. Over the last few years, several methods of OMR have been proposed for a standard sheet music. However, these methods cannot be applied directly for a traditional Thai sheet music (TSM) which uses Thai characters to represent music notes in the traditional way. This paper proposes a novel method to interpret an image of TSM into an editable or playable form. The proposed method contains three main processes including: 1) edge detection; 2) music note segmentation; and 3) music note recognition. First, canny edge detection is applied on an image of TSM. Second, music notes are individually segmented by using statistical analyses. Third, support vector machine (SVM) is used to recognize the segmented music notes. The experimental results based on many images of TSMs demonstrate that the proposed method can achieve very promising performance of above 80% accuracy, with a perfect performance (i.e. 100% accuracy) for some TSMs.

Keywords—Optical music recognition, Thai sheet music, canny edge detection, support vector machine

I. INTRODUCTION

OMR is a traditional term of a system used for recognizing sheet musics or music scores into a machine-readable format. It is firmly related to optical character recognition (OCR) [1]. OMR has several benefits including: 1) making music available and understandable to people even without prerequisite knowledge on music [2]; and 2) preserving music in a handwritten or a hard-copy format from being lost through normal ravages of time [3]. This paper focuses on OMR for TSM. It provides one additional benefit of converting Thai music notes into standard music notes which can be understandable by non-Thai people who do not know Thai characters. This can help to promote traditional Thai music and culture.

OMR is tightly linked to processes of music symbol segmentation and recognition [1]. Bellini et al. [1] presented an object oriented optical music recognition (O³MR). This method was based on an extensive use of projection profiles for locating basic music notations without removing staff lines. Fornes et al. [4] used median filters with a vertical structuring element to detect vertical lines of vertical shapes in a music sheet. Then, a morphological opening with an elliptical structuring element was applied to detect filled note heads. Chen and Xia [5] proposed a similar method, but they used a morphological hit and miss transform in place of a morphological opening. Raphael and Wang [6] identified candidate composite symbols by using grammatically-formulated top-down model-based methods. Then, template matching was employed to find isolated rigid symbols.

The above methods of OMR were developed for international/western sheet musics. There are only a few works developed for traditional sheet musics. For example, Chen and Sheu [7] introduced an optical music recognition system for traditional Chinese Kunqu Opera scores written in Gongche Notation. It obtained musical information with Bayesian, genetic algorithm, and k-nearest neighbor classifiers. However, based on our literature review, there is no existing works developed for traditional Thai sheet musics.

In this paper, a novel method is proposed for OMR of traditional Thai sheet musics (TSM) which use Thai alphabets to represent music notes, as can be seen in Fig. 1. TSM records music notes in a table format which is divided into multiple rows. Each row consists of eight cells. Each cell is called ‘hong (i.e. room)’. Then, each row consists of four music notes.

<table>
<thead>
<tr>
<th>Thai music notes</th>
<th>International/Western music notes</th>
<th>Vocal scales</th>
</tr>
</thead>
<tbody>
<tr>
<td>ฮ</td>
<td>C</td>
<td>Do</td>
</tr>
<tr>
<td>ต</td>
<td>D</td>
<td>Re</td>
</tr>
<tr>
<td>ง</td>
<td>E</td>
<td>Mi</td>
</tr>
<tr>
<td>จ</td>
<td>F</td>
<td>Fa</td>
</tr>
<tr>
<td>ฉ</td>
<td>G</td>
<td>Sol</td>
</tr>
<tr>
<td>ฐ</td>
<td>A</td>
<td>La</td>
</tr>
<tr>
<td>ฑ</td>
<td>B</td>
<td>Ti</td>
</tr>
</tbody>
</table>

Fig. 1: An example of traditional Thai sheet music (TSM).
There are seven basic music notes represented by seven Thai alphabets, as shown in Table I. Table I also shows linkages between each Thai music note and its corresponding international/western music note. In TSM, ‘-’ is used to replace a music note in order to indicate a range of music sound. For example, ‘-’ represents 1/4 rhythm, ‘- -’ represents 2/4 rhythm, ‘- - -’ represents 3/4 rhythm, and ‘- - - -’ represents 4/4 rhythm.

Our proposed framework for OMR of TSM is shown in Fig. 2. The ultimate objective of this work is to transform an image of TSM into a machine-readable form. An image of TSM is first converted into a binary image of edge information by using canny edge detection [8]. This process is performed to reduce image noise and unnecessary image information (e.g., colors) for this work. Then, some basic statistic techniques are employed to segment Thai music notes in the edge image of TSM. At the end, each segmented Thai music note is recognized as ‘-’ or one of the seven music notes shown in Table I, by using SVM [9].

The rest of this paper is organized as follows. Edge detection is explained in section II. Music note segmentation and music note recognition are proposed in sections III and IV respectively. Experimental results are shown in section V and conclusions are drawn in section VI.

II. EDGE DETECTION

Canny edge detection [8] with five processes is used in this study. Given an image of TSM (I), the first process is to convert I into a corresponding greyscale image (I₁), which can be done by using equation (1) [10].

\[ I₁(x, y) = 0.2989I_r(x, y) + 0.5870I_g(x, y) + 0.1140I_b(x, y) \]  

where \( I₁(x, y) \) denotes the pixel’s value at index (x, y) of a greyscale image \( I₁ \), and \( I_r(x, y) \), \( I_g(x, y) \) and \( I_b(x, y) \) denote the pixels’ values of red, green and blue respectively at index (x, y) of an image I. As can be seen in equation (1), red, green and blue are not weighted equally. This is because pure blue is darker than pure red which is darker than pure green, regarding their sensitivities to human eyes.

The second process is to use a Gaussian filter (\( G \)) to smooth the image \( I₁ \) as in equation (2). It will trade-off between image’s noise reduction and image’s edge localization.

\[ I₂(x, y) = G \ast (x, y) I₁ \]  

where \( I₂(x, y) \) denotes the pixel’s value at index (x, y) of \( I₂ \), and \( G \ast (x, y) \) denotes the convolution by using the index (x, y) as the center point.

The third process is to compute the gradient magnitude and angle of \( I₂ \) in both x and y directions by using approximations of partial derivatives. The derivatives \( D_x(x, y) \), \( D_y(x, y) \) of \( I₂ \) in the x and y directions can be computed as follows.

\[ D_x(x, y) = G_x \ast (x, y) I₂ \]  
\[ D_y(x, y) = G_y \ast (x, y) I₂ \]  

where \( G_x \) and \( G_y \) are the following 3 \( \times \) 3 kernels.

\[ G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \]

Then, the magnitude (\( M(x, y) \)) and orientation (\( \theta(x, y) \)) of the gradient are calculated as follows.

\[ M(x, y) = \sqrt{D_x^2(x, y) + D_y^2(x, y)} \]  
\[ \theta(x, y) = \arctan\left(\frac{D_x(x, y)}{D_y(x, y)}\right) \]

The orientation \( \theta \) is rounded to be one of four directions 0°, 45°, 90°, or 135°.

The fourth process is to thin edges by applying a non-maxima suppression to the gradient magnitude. The output image is \( I₃ \). This process will keep only those pixels on an edge with the highest gradient magnitude. For each pixel at index (x, y), three pixels in a 3 \( \times \) 3 surrounding area will be examined. The three pixels are selected based on the gradient orientations. Case 1: if \( \theta(x, y) = 0^\circ \), then the three pixels are at index \((x + 1, y), (x, y), \) and \((x - 1, y)\). Case 2: if \( \theta(x, y) = 45^\circ \), then the three pixels are at index \((x + 1, y + 1), (x, y), \) and \((x - 1, y - 1)\). Case 3: if \( \theta(x, y) = 90^\circ \), then the three pixels are at index \((x, y + 1), (x, y), \) and \((x, y - 1)\). Case 4: if \( \theta(x, y) = 135^\circ \), then the three pixels are at index \((x + 1, y - 1), (x, y), \) and \((x - 1, y + 1)\). Then, among the three pixels, if the pixel at index (x, y) has the highest gradient magnitude, it is kept as an edge. Otherwise, it is not classified as an edge pixel. This process can be summarized as:

- when \( \theta(x, y) = 0^\circ \), if \( M(x, y) > M(x + 1, y) \) and \( M(x, y) > M(x - 1, y) \) then \( I₃(x, y) = M(x, y) \), otherwise \( I₃(x, y) = 0 \)
- when \( \theta(x, y) = 45^\circ \), if \( M(x, y) > M(x + 1, y + 1) \) and \( M(x, y) > M(x - 1, y - 1) \) then \( I₃(x, y) = M(x, y) \), otherwise \( I₃(x, y) = 0 \)
- when \( \theta(x, y) = 90^\circ \), if \( M(x, y) > M(x + 1, y - 1) \) and \( M(x, y) > M(x, y - 1) \) then \( I₃(x, y) = M(x, y) \), otherwise \( I₃(x, y) = 0 \)
The image $I_4$ will be resulted as the edge image of the inputted image $I$ by applying the five processes above. Fig. 3 shows an example of the edge image.

### III. MUSIC NOTE SEGMENTATION

Given the edge image ($I_4$) of TSM, the next process is to segment individual music notes from the sheet. This process contains three main steps of: 1) line removal; 2) music line detection; and 3) music note detection.

As can be seen in Fig. 3 (b), there are many vertical and horizontal lines of the music tables. In this study, hough line technique [12] is applied to detect such lines. The detected lines are then removed to remain with only music notes. The next step is to detect music lines as shown in Fig. 4. In this paper, a music line means a set of consecutive rows of $I_4$, which contain music notes. In Fig. 4, each music line is a set of consecutive rows, which starts with a blue horizontal line and ends with a green horizontal line.

To detect a music line, white pixels (i.e. edge pixels) are counted for each row of $I_4$, as below.

$$R(r) = \sum_{c=1}^{N_c} I_4(r,c), \quad 1 \leq r \leq N_r$$  \hspace{1cm} (6)

where $R(r)$ is the number of white pixels in row $r$, $N_c$ is the number of columns in $I_4$, and $N_r$ is the number of rows in $I_4$. Then, the average number ($\bar{R}$) of white pixels in one row is calculated as below.

$$\bar{R} = \frac{\sum_{r=1}^{N_r} R(r)}{N_r}$$  \hspace{1cm} (7)

If $R(r) > \bar{R}$ then row $r$ is a candidate row to be in a music line. A music line is formed up by grouping a set of consecutive candidate rows, as shown in Fig. 4.
In this work, the input space is a detected music note with shown in Table I.

A classification model for '-' and seven types of music notes as negative samples. Thus, SVM will be used to construct eight classification models for a music note 'Do', and of music note recognition.

To detect music notes in each music line, white pixels (i.e. edge pixels) are counted for each column of the music line, as below.

\[ C(c) = \sum_{r=r_1}^{r_2} I_4(r, c), \quad 1 \leq c \leq N_c \]  

(8)

where \( C(c) \) is the number of white pixels in column \( c \) within the music line, \( r_1 \) is the starting row (i.e. blue horizontal lines in Fig. 4) of the music line, and \( r_2 \) is the ending row (i.e. green horizontal lines in Fig. 4) of the music line. Then, the average number (\( \bar{C} \)) of white pixels in one column within the music line is calculated as below.

\[ \bar{C} = \frac{\sum_{c=1}^{N_c} C(c)}{N_c} \]  

(9)

If \( C(c) > \bar{C} \) then column \( c \) is a candidate column to be a part of a music note. A music note is formed by grouping a set of consecutive candidate columns, as shown in Fig. 4. Red vertical lines are starting columns and white vertical lines are ending columns, of music notes. The detected music notes are then rescaled to have a fixed size of 30 x 30 pixels, as shown in Fig. 5.

IV. MUSIC NOTE RECOGNITION

In this paper, support vector machine (SVM) [13][14][15] is used as a classification method. This section is divided into three sub-sections of: 1) SVM concept; 2) SVM in a training phase of music note recognition; and 3) SVM in a testing phase of music note recognition.

A. SVM concept

Let \( S = \{(x_i, y_i)\}_{i=1}^{N} \) be a training dataset, where \( x_i \) is a data point (i.e. a detected music note), \( y_i \) is a class label of \( x_i \) (i.e. a class label of a music note), and \( N \) is the total number of training data points. In this work, \( y_i \in \{-1, 1\} \), considering a binary classification. For example, when SVM is used to train a classification model for a music note 'Do', \( y_i \) of a music note 'Do' will be labeled '1' (i.e. called positive samples), while \( y_i \) of other music notes will be labeled '-1' (i.e. called negative samples). Thus, SVM will be used to construct eight classification models for '-' and seven types of music notes as shown in Table I.

Let \( \phi \) be a mapping from the input space to a feature space. In this work, the input space is a detected music note with 30 x 30 pixels and the feature space is the concatenation of all rows in a detected music note.

SVM algorithm will find a hyperplane \((w, b)\) to maximize the classification margin \( \gamma \).

\[ \gamma = \min_{i} y_i \langle w, \phi(x_i) \rangle - b \]  

(10)

where \( \langle \cdot, \cdot \rangle \) denotes an inner product, \( w \) is a weight vector, \( b \) is a real number, and \( \gamma \) is called the margin. The term \( \langle w, \phi(x_i) \rangle - b \) corresponds to the distance between the point \( x_i \) and the SVM decision boundary.

Then, a class label \( f(x) \) is assigned to a new data point \( x \) as below.

\[ f(x) = sgn(\langle w, \phi(x) \rangle - b) \]  

(11)

To maximize the margin hyperplane, the weight vector is calculated as below [14].

\[ w = \sum_{i=1}^{N} \alpha_i y_i \phi(x_i) \]  

(12)

where \( \alpha_i \) are positive real numbers such that the following objective function is maximized.

\[ \sum_{i=1}^{N} \alpha_i - \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \langle \phi(x_i), \phi(x_j) \rangle \]

subject to \( \sum_{i=1}^{N} \alpha_i y_i = 0 \), \( \alpha_i > 0 \)

Thus, the classification function can be re-written as in equation (14).

\[ f(x) = sgn(\sum_{i=1}^{N} \alpha_i y_i < \phi(x_i), \phi(x) > -b) \]  

(14)

B. SVM in a training phase of music note recognition

As mentioned above, SVM is applied to construct eight classification models \( f_i(x), \quad i \in \{-, \text{Do}, \text{Re}, \text{Mi}, \text{Fa}, \text{Sol}, \text{La}, \text{Ti}\} \) for '-' and seven types of music notes. To construct each model \( f_i \), SVM is applied on a training dataset which consists of both positive and negative samples. The positive samples are the detected music symbols of i, while the negative samples are the other detected music symbols. For example, to construct \( f_{\text{Do}} \), 80 images of a music note 'Do' are used as positive samples, and 80 images of other music notes including '-' are used as negative samples. In this study, 80 positive samples and 80 negative samples are used as a training dataset for creating each SVM-based classification model.

C. SVM in a testing phase of music note recognition

Given a newly detected music note \((x)\) with unknown label, it will be tested with all eight models for identifying its music note label \( l(x) \) as below.

\[ l(x) = \arg \max_i f_i(x) \]  

(15)
TABLE II: The performance (%) of the proposed music note recognition method in a format of the confusion matrix.

<table>
<thead>
<tr>
<th>Thai music notes</th>
<th>๑</th>
<th>๒</th>
<th>๓</th>
<th>๔</th>
<th>๕</th>
<th>๖</th>
<th>๗</th>
<th>–</th>
</tr>
</thead>
<tbody>
<tr>
<td>๑</td>
<td>85</td>
<td>2.5</td>
<td>0</td>
<td>2.5</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>๒</td>
<td>0</td>
<td>92.5</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>2.5</td>
<td>0</td>
</tr>
<tr>
<td>๓</td>
<td>30</td>
<td>0</td>
<td>60</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>๔</td>
<td>12.5</td>
<td>0</td>
<td>0</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>7.5</td>
<td>0</td>
</tr>
<tr>
<td>๕</td>
<td>15</td>
<td>7.5</td>
<td>2.5</td>
<td>0</td>
<td>72.5</td>
<td>0</td>
<td>2.5</td>
<td>0</td>
</tr>
<tr>
<td>๖</td>
<td>12.5</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>75</td>
<td>2.5</td>
<td>0</td>
</tr>
<tr>
<td>๗</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>17.5</td>
<td>0</td>
<td>5</td>
<td>72.5</td>
<td>0</td>
</tr>
<tr>
<td>–</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>95</td>
</tr>
</tbody>
</table>

(a) A TSM of the song 'Lao-Joy'
(b) A TSM of the song 'Noo-Ma-Lee'

Fig. 6: The two TSMs used in the experiment. (a) The song 'Lao-Joy'. (b) The song 'Noo-Ma-Lee'.

V. EXPERIMENT

In our experiment, 960 images of music symbols (i.e. 120 images of each music note and 120 images of ‘–’ are used to evaluate the performance of our proposed method. The 640 images of music symbols (i.e. 80 images of each music note and 80 images of ‘–’ are used to train SVM models in the training phrase. Then, the rest of 320 images of music symbols (i.e. 40 images of each music note and 40 images of ‘–’) are used in the testing phrase.

These 960 images are automatically detected from 5 images of 5 different TSMs by using our proposed music note segmentation method in section III. The 5 images of these 5 TSMs were taken by a smartphone’s camera with 8 megapixels and f/2.4 aperture. When taking a photo of each TSM, music notes in each music line should be aligned within a same horizontal line as much as possible. This will help to improve the quality of the segmented music notes in section III.

The proposed method has been implemented by using the library functions of OpenCV in Microsoft Visual C++ environment on the computer with 2.60-GHz CPU and 4.00-GB RAM. In this paper, the linear kernel is used as a kernel function for SVM. The performance of the proposed music note recognition method is shown in Table II in a format of the confusion matrix.

In Table II, the accuracies for recognizing the Thai music notes ‘Do’, ‘Re’, ‘Mi’, ‘Fa’, ‘Sol’, ‘La’, ‘Ti’ and ‘–’ are 85%, 92.5%, 60%, 80%, 72.5%, 75%, 72.5% and 95% respectively.

TABLE III: The performance (%) of the proposed method on a TSM of the song 'Lao-Joy'.

<table>
<thead>
<tr>
<th>Thai music notes</th>
<th>Numbers of Thai music notes in the TSM</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>๑</td>
<td>14</td>
<td>86</td>
</tr>
<tr>
<td>๒</td>
<td>18</td>
<td>100</td>
</tr>
<tr>
<td>๓</td>
<td>15</td>
<td>100</td>
</tr>
<tr>
<td>๔</td>
<td>0</td>
<td>none</td>
</tr>
<tr>
<td>๕</td>
<td>19</td>
<td>79</td>
</tr>
<tr>
<td>๖</td>
<td>10</td>
<td>100</td>
</tr>
<tr>
<td>๗</td>
<td>0</td>
<td>none</td>
</tr>
<tr>
<td>–</td>
<td>52</td>
<td>65</td>
</tr>
</tbody>
</table>

Average accuracy: 81%

TABLE IV: The performance (%) of the proposed method on a TSM of the song 'Noo-Ma-Lee'.

<table>
<thead>
<tr>
<th>Thai music notes</th>
<th>Numbers of Thai music notes in the TSM</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>๑</td>
<td>6</td>
<td>100</td>
</tr>
<tr>
<td>๒</td>
<td>20</td>
<td>100</td>
</tr>
<tr>
<td>๓</td>
<td>20</td>
<td>100</td>
</tr>
<tr>
<td>๔</td>
<td>0</td>
<td>none</td>
</tr>
<tr>
<td>๕</td>
<td>4</td>
<td>100</td>
</tr>
<tr>
<td>๖</td>
<td>0</td>
<td>none</td>
</tr>
<tr>
<td>๗</td>
<td>0</td>
<td>none</td>
</tr>
<tr>
<td>–</td>
<td>14</td>
<td>100</td>
</tr>
</tbody>
</table>

Average accuracy: 100%

The average accuracy for all music notes is about 79%.

For a comprehensive experiment, the proposed method is further verified by using two more TSMs. As shown in Fig. 6, the two TSMs are the songs named ‘Lao-Joy’ and ‘Noo-Ma-Lee’. The performances of the proposed method on these two TSMs are shown in Tables III and IV. The proposed method achieves approximately 81% accuracy for the song ‘Lao-Joy’ and obtains perfectly 100% accuracy for the song 'Noo-Ma-Lee'.
VI. Conclusion

This paper has proposed a method of optical music recognition (OMR) for a traditional Thai sheet music (TSM). The proposed method consists of three main processes including the edge detection, the music note segmentation, and the music note recognition. The canny edge detection has been used to extract an edge image of a given image of a TSM. Then, the thresholding technique based on the basic statistics has been applied to segment individual music notes from the edge image. After that, support vector machine (SVM) has been employed to recognize the detected music notes. The proposed method can achieve the very promising performance of above 80% accuracy.

In the future work, the proposed method will be improved by developing a more robust feature for describing the detected music notes. The proposed method will also carefully consider the case of low and high pitch of the music.

References