Musical Document Retrieval through Image Spotting

|  |  |  |
| --- | --- | --- |
| Rakesh Malik  School of Engineering & Technology, Department of Computer Science,  Pondicherry University  Puducherry, India  rakeshmalik91@gmail.com | Partha Pratim Roy  Laboratoire ďInformatique, Université François Rabelai,  Tours, France  2partharoy@gmail.com | Umapada Pal  Computer Vision and  Pattern Recognition Unit,  Indian Statistical Institute,  Kolkata-108, India  umapadap@gmail.com |

ABSTRACT

In this paper, we present a novel approach for retrieval of handwritten musical documents using a query sequence/word of musical symbols. In our algorithm, the musical score-words are described as sequences of symbols generated from a codebook vocabulary of musical scores. Staff lines are removed first from musical documents using structural analysis of staff lines and their connection with symbols. Symbol codebook vocabulary is created in offline by a clustering method. Three different feature extraction methods namely: Zernike Moments, 400 dimensional gradient feature and Dynamic Time Warping are tested and two unsupervised classifiers SOM and K-Mean are evaluated in our algorithms. Using this symbol codebook, the music symbol information in the document images is encoded. Given a query sequence of musical symbols in a musical score-line, the symbols in the query are searched. Finally, an approximate string matching based sub-string matching algorithm is applied to find query words. The matching similarity is used to rank the retrieved words. The proposed method is tested on a collection of handwritten musical documents from a public dataset. The performance results with the codebook creation and is compared.

**Keywords**

Musical Document Retrieval, Staff Removal, Symbol Classification, Approximate String Matching.

# 1. Introduction

Indexing/retrieval from musical document image collections is a challenging task due to complexity of handwriting, symbol touching with staff lines, etc. Segmentation and recognition of individual music-scores is difficult.

Recently, there has been a growing interest in the analysis of handwritten music scores. People are interested mainly the recognition of handwritten music scores (Optical Music Recognition), and the identiﬁcation (or veriﬁcation) of the writer of an anonymous music score. People have been working for staff removal algorithms, since a good detection and removal of the staff lines will allow the correct isolation and segmentation of the musical symbols, and consequently, will ease the correct recognition and classiﬁcation of the music symbols.

Some recent works on musical documents are mentioned here. Alicia Fornés et. al. [3] have present the CVCMUSCIMA database and ground truth of handwritten music score images. The dataset consists of 1,000 music sheets written by 50 different musicians. It has been especially designed for writer identification and staff removal tasks. In addition to the description of the dataset, ground truth, partitioning, and evaluation metrics, we also provide some baseline results for easing the comparison between different approaches. Christoph Dalitz et al [4] have presented a quantitative comparison of different algorithms for the removal of staff-lines from music images. It contains a survey of previously proposed algorithms and suggests a new skeletonization based approach.

Nowadays, musicologists must work very hard to identify the writer of an unknown manuscript. In fact, they do not only perform a musicological analysis of the composition (melody, harmony, rhythm, etc), but also analyze the handwriting style of the manuscript. In this sense, writer identiﬁcation can be performed by analyzing the shape of the hand-drawn music symbols (e.g. music notes, clefs, accidentals, rests, etc), because it has been shown that the author’s handwriting style that characterizes a piece of text is also present in a graphic document.

With the growing popularity of internet and web technology, Digital Libraries and archives are becoming interested in mass-digitization and transcription of the collection of historical documents. The purpose is to make the historical documents available to the wider users for accessing the valuable information. The searching/retrieval of content information depend on the transcription of these documents. Many document retrieval methods have already been stated so far, but these are done usually for text documents containing handwritten or printed text character of different language.

In these documents, there are many works for removal of staff lines, analysis and classification of music scores, writer identiﬁcation, etc.

In this paper we present such a method for retrieval of a musical score, which can help in searching or indexing in handwritten or printed historical documents.

The orientations of symbols in musical documents are different than orientations in text documents. The space between symbols in a musical document is much more than that in text documents. A major problem of musical document is that it usually contains several long horizontal lines called staff-lines, as shown in Fig.1.



**Figure 1: Examples of a musical score is shown.**

These lines need to be removed before using any retrieval process. The method proposed by Umapada Pal et al. [1] is used to remove the staff-lines. Several staff-line removed documents are taken and symbols are extracted from the documents. Multi-dimensional features are extracted from the symbols. The system is tested with several features such as Zernike Feature, 400-Dimensional Feature. The symbols are clustered some classes using the extracted features. Thus the system is trained. Now we have to search a sequence of musical symbols in a line of a musical score. First we extract the symbols of the query and text sequence and the same feature as the system is trained with is extracted for each symbols. The best matching classes are selected from system to create a cost matrix to perform the matching. Then the cost matrix is used to find the best match of query sequence in the text sequence.

**Figure 2: Examples of staff-line removal in a musical document.**

# 2. Preprocessing and Symbol Extraction

Staff Line Removal: First, the images are thinned and, by analyzing the thinned portions, the input images are automatically categorized in two groups: (a) images containing straight staff line and (b) other non-straight or curved staff-lines. Images containing straight staff lines are further divided into horizontal staff lines and non-horizontal straight lines. Next, staff lines are detected based on the characteristics of each group. Some smoothing techniques are also utilized to get better accuracy. The staff line detections methods developed here can be considered as passing a ring on a wire (here wire can be considered as staff-line). If there is any obstacle like music score the obstacle portions is retained or deleted based on some measures. For staffline detection the authors computed staff line height, staff space height, vertical positional variance of the pixels of thinned lines, etc. These parameters guided their system to detect the staff line part efﬁciently.

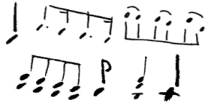
(a) A document with staff-lines

(b) Result of staff-line removal

**Figure 3: Examples of staff-line removal in a musical document.**

**Figure 4: Component joining algorithm.**

Symbol Extraction: The space between symbols in a musical document is usually much more than that in text document. But the most major problem of symbol extraction is broken symbols. Some examples of broken symbols are shown in Fig.5.

****

**Figure 5: Examples of some broken symbols are shown.**

Dilation is used to join the broken parts that are very near. The very small parts are filtered out to remove possible noises. Then the components, overlapped vertically are joined. Any small components over or below any long parts are joined. The components with overlapping bounding box are joined. Then after creating the symbols any straight long lines are filtered out to remove the vertical bar lines (as one shown in Fig.1).

# 3 Musical Score Codebook Vocabulary

To make the system scaling and rotation invariant, 2 scaling and rotation invariant features are chosen: Zernike feature and 400-dimensional feature. These features are used to train the classifier and while searching these features of the components of query and the component of text are extracted and used.

The Zernike, 400-dimensional and DTW feature are described below.

## 3.1 Feature Extraction

To make the system scaling and rotation invariant, 2 scaling and rotation invariant features are chosen: Zernike feature and 400-dimensional feature. These features are used to train the classifier and while searching these features of the components of query and the component of text are extracted and used. The Zernike, 400-dimensional and DTW feature are described below.

## 3.1.1 Zernike Moment Feature

Zernike moments are based on a set of complex polynomials that form a complete orthogonal set over the interior of the unit circle [10]. They are defined to be the projection of the image function on these orthogonal basis functions. The basis functions *Vn*,*m*(*x*,*y*) are given by:

 (1)

where n is a non-negative integer, m is non-zero integer subject to the constraints n-|m| is even and, ρ is the length of the vector from origin to, θ is the angle between vector ρ and the x-axis in a counter clockwise direction and *Rn*,*m*(ρ) is the Zernike radial polynomial. The Zernike radial polynomials, *Rn*,*m*(ρ),are defined as:



Note that, *Rn*,*m*(ρ) = *Rn*, −*m*(ρ). The basis functions in equation 1 are orthogonal thus satisfy,



where 

The Zernike moment of order n with repetition m for a digital image function *f*(*x*,*y*) is given by  where  is the complex conjugate of  .

To compute the Zernike moments of a given image, the image center of mass is taken to be the origin. In our approach, the blobs are normalized into 41\*41 before applying Zernike feature computation. The size is considered from the performance of experimental data.

## 3.1.2 400-dimensional feature

400 dimensional gradient based feature is used in our system for classification. The text image is normalized into 126x126 size and converted to gray-scale image by applying a set of mean-filtering. Next the resultant gray image is segmented into 9X9 blocks. Roberts filter is applied next to obtain gradient image. The direction of gradient is quantized into 16 directions and the gradient strengths are accumulated in each quantized direction. Histograms of 16 quantized directions are computed in each of 9x9 blocks. Finally, 9x9 blocks are down sampled into 5x5 by a Gaussian filter. Thus, we get 5x5x16 = 400 dimensional feature.

## 3.1.3 DTW feature

Dynamic time warping (DTW) is an algorithm for measuring similarity between two sequences which may vary in time or speed. DTW is a method that allows a computer to find an optimal match between two given sequences (e.g. time series) with certain restrictions. The sequences are "warped" non-linearly in the time dimension to determine a measure of their similarity independent of certain non-linear variations in the time dimension.

Here DTW is used on two signals of 4 multidimensional features (): projection profile, upper and lower word profile and foreground transition. These features are extracted from the image after normalising them. The DTW distance between two signals I1 and I2 is calculated using a matrix D.

where

Finally, the matching cost is normalized by the length of the warping path . If the length of a features is , then the DTW distance between 2 symbols are .

For a text of length and a query of length a matrix of order is created as following:

Then approximated string match (ASM) is used with help of the matrix to find the best matches.

# 3.2 Classifiers

The system has been tested with 3 classifiers: Self Organising Map, K-Means and Gaussian Mixture Model. The classifiers are described in brief below. These classifiers are used while using Zernike feature or 400-dimensional feature. No classifiers were needed to be used while using DTW.

## 3.2.1 Self Organizing Map

A self-organizing map consists of components called nodes or neurons. Associated with each node is a weight vector of the same dimension as the input data vectors and a position in the map space. The usual arrangement of nodes is a regular spacing in a hexagonal or rectangular grid. The self-organizing map describes a mapping from a higher dimensional input space to a lower dimensional map space. The procedure for placing a vector from data space onto the map is to first find the node with the closest weight vector to the vector taken from data space. Once the closest node is located it is assigned the values from the vector taken from the data space.

The algorithm is as following:

1. Randomize the map's nodes' weight vectors

2. Grab an input vector

3. Traverse each node in the map

a. Use Euclidean distance formula to find similarity between the input vector and the map's node's weight vector

b. Track the node that produces the smallest distance (this node is the best matching unit, BMU)

4. Update the nodes in the neighborhood of BMU by pulling them closer to the input vector

Wv(t + 1) = Wv(t) + Θ(t)α(t)(D(t) - Wv(t))

5. Increase t and repeat from 2 while t < \lambda

## 3.2.2 K-Means

Given a set of observations (x1, x2, …, xn), where each observation is a d-dimensional real vector, k-means clustering aims to partition the n observations into k sets (k ≤ n) S = {S1, S2, …, Sk} so as to minimize the within-cluster sum of squares (WCSS):

\underset{\mathbf{S}} {\operatorname{arg\,min}} \sum_{i=1}^{k} \sum_{\mathbf x_j \in S_i} \left\| \mathbf x_j - \boldsymbol\mu_i \right\|^2 

where μi is the mean of points in Si.

# 4. ReTrieval of Query Sequence

For searching a query word Q from the collection of

document images, Q is first segmented into primitives as explained in Section 3.1. Next, we find out most similar codebook models of \_m for each primitive segments. Q is then encoded into a sequence of labels Lq1Lq2 : : :Lqt, where Lqi 2 Lm and t is the number of primitives in Q. To handle noise, ‘c’ nearest codebook models are also stored for each query primitives.

Now, the objective is to find words that have similar sequence of labels. Thus, the matching between query word Q and a target word W is formulated as matching of 2 sequences of primitives. As, primitives are labelled by codebook primitives Lm, the problem is to match primitive label strings of query and target word.

Approximate string matching algorithm [2] has been

used in our system for text searching. This method has

frequently been used in the literature to refer to a class of pattern matching techniques, by which k errors are allowed between a pattern string S and a text string T. The length of the strings S and T may be different. The algorithm finds all substrings of the text T that have at most k errors (character that are not same) with the pattern S. When k = 0 (no mismatches) it is simple string matching algorithm.

## 4.1 Self Organizing Map

In our system a SOM of is trained with features extracted from many symbols. The features of symbols in query and text are extracted.

A matrix C of order is created as follows:

where

ASM is used with help of the matrix to search the query string in the text string.



**Figure 6: SOM clustering for Zernike features of Musical symbols**

## 4.2 K-Means

When a query string is to be searched in a text string, the features of the components of the query and text of length and respectively are extracted.

The system is trained with features of many symbols and clustered into classes. The cluster centres are calculated by mean of the elements of the cluster. The distance of feature of a symbol from all the cluster-centres is calculated. A weighted mean of nearest 3 vectors is considered as nearest vector. A matrix of order is calculated as follows:

where

Then approximated string matching is used with help of matrix to find the best match of query string into the text string.

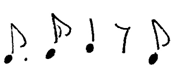
# 5. Result and discussions

## 5.1 Data collection

50 musical scores and 10 query samples are taken. Dataset is taken from a database, easily available on internet, “The CVC-MUSCIMA Database” [3].

## 5.2 Segmentation result

An example of result of segmentation is shown in following figure.





**Figure 7: Examples of segmentation result.**

­­

## 5.3 Retrieval result

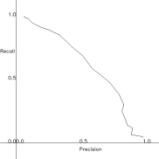
Some of retrieval results are shown in the following figures.



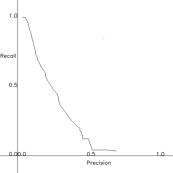
1. Query
2. Retrieval result for the shown query

**Figure 8: Example of retrieval results of a query in some texts.**

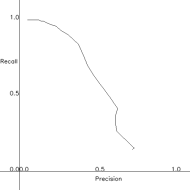
The following figures show the precision-recall graph for each feature and classifiers.



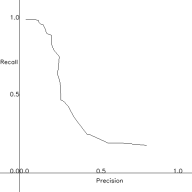
1. Zernike Feature and K-Means



1. Zernike Feature and SOM



1. 400 dimensional feature and K-Means



1. 400 dimensional feature and SOM



1. DTW

**Figure 9: Example of retrieval results of a query in some texts.**

The best result is from Zernike feature and SOM. The result for different number of clusters is shown below.

**Figure 10: Example of retrieval results Zernike feature and K-Means for different number of classes.**

# 6. Conclusion

In this system a new approach to musical document retrieval is shown. But there are scopes for improvements using this approach by extending the investigation to more accurate segmentation and classification and in future we plan to do it.

# 7. Acknowledgements

xxx

# 8. References

1. Jit Ray Chowdhury, Umapada Pal “The ICDAR 2011 Music Scores Competition: Staff Removal and Writer Identification”
2. S. Li, M. C. Lee, C. M. Pun, "Complex Zernike Moments Features for Shape-Based Image Retrieval", IEEE TSMEC, Part A: Systems and Humans, vol. 39, no. 1, pp. 227-237, 2008.
3. Alicia Fornés, Anjan Dutta, Albert Gordo, Josep Lladós “CVC-MUSCIMA: a ground truth of handwritten music score images for writer identification and staff removal” The CVC-MUSCIMA Database : <http://www.cvc.uab.es/cvcmuscima/index_database.html>
4. Christoph Dalitz, Michael Droettboom, Bastian Pranzas, and Ichiro Fujinaga “A Comparative Study of Staff Removal Algorithms”

# APPENDIX

## A.1 Introduction

## A.2 Preprocessing and Symbol Extraction

## A.3 Musical Score Codebook Vocabulary

**A.3.1 Feature Extraction**

A.3.1.1 Zernike Moment Feature

A.3.1.2 400-Dimensional Feature

**A.3.1 Classifiers**

## A.3.1.3 DTW Feature

*A.3.2.1 Self Organising Map*

*A.3.2.2 K-Means Clustering*

## A.4 Retrieval of Query Sequence

*A.4.1 Self Organising Map*

*A.4.2 K-Means Clustering*

## A.5 Result and Discussions

*A.5.1 Data Collection*

*A.5.2 Segmentation Result*

*A.5.3 Retrieval Result*

## A.6 Conclusion

## A.7 Acknowledgement

## A.8 References