

Master Thesis

**Cross-Modal Matching of Text, Image and
Symbolic Music Data**

submitted by

Sanu Pulimootil Achankunju

submitted

November 26, 2014

Supervisor / Advisor

Prof. Dr. Meinard Müller

Dipl.-Ing. Stefan Balke

Reviewers

Prof. Dr. Meinard Müller

Erklärung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus anderen Quellen oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet. Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form in einem Verfahren zur Erlangung eines akademischen Grades vorgelegt.

Erlangen, November 26, 2014

Sanu Pulimootil Achankunju

Acknowledgements

First and foremost, I would like to thank God almighty for his blessings.

I cannot find words to express my gratitude to my supervisor Prof. Dr. Meinard Müller, who has been a source of inspiration with his timely guidance throughout the process of my project work. Without his encouragement and guidance this project would not have materialized. I would like to extend my deepest gratitude to my co-supervisor Dipl.-Ing. Stefan Balke, who was abundantly helpful and offered invaluable assistance, support and guidance. I would also like to thank him for never getting tired of my questions.

I would like to thank Lukas Lamprecht for the valuable discussions and assistance. He provided me with the database subsets that were used in my thesis. My sincere gratitude to Dr. Vlora Arifi-Müller who helped me with the files I used for the segmentation tasks. Special thanks also to Thomas Prätzlich and Jithin Mohan for their precious time in proof reading my thesis.

I would also like to convey thanks to the researchers at Semantic Audio Processing, International Audio Laboratories Erlangen, for providing the technical know-how, computational resources and above all a pleasant working atmosphere. I also wish to extend my love and gratitude to my beloved family members for their endless love and support, through the duration of my thesis.

Abstract

There has been a rapid growth of digitally available music data including audio recordings, digitized images of scanned sheet music, album covers and video clips. The huge amount of readily available music requires retrieval strategies that allow users to explore large music collections in a convenient and enjoyable way. In this thesis, we deal with a challenging music retrieval scenario that involves text-based descriptions as well as visual sheet music representations. In our specific setting, we start with a printed book, which contains 9803 important musical themes (given as sheet music) from the Western classical music literature. The objective is to automatically match these themes to other digitally available sources. To this end, we introduce a processing pipeline that automatically extracts from the scanned pages of the printed book textual metadata using Optical Character Recognition (OCR) as well as symbolic note information using Optical Music Recognition (OMR). Due to the poor printing quality of the book, the OCR and OMR results are quite noisy containing numerous extraction errors. As one main contribution, we adjust information retrieval techniques for matching musical themes based on the OCR and OMR input. In particular, we show how the retrieval quality can be substantially improved by introducing suitable data fusion strategies. Finally, we report on extensive experiments within our specific matching scenario, which also indicate the potential of our techniques when considering other sources of musical themes such as digital music archives and the world wide web.

Contents

Erklärung	i
Acknowledgements	iii
Abstract	v
1 Introduction	3
1.1 Main Contribution	4
1.2 Thesis Organization	5
2 Automated Segmentation of Images	7
2.1 Description of the Images	7
2.2 Segmentation Methods	9
2.3 Summary	18
3 Text and Score Recognition	19
3.1 Optical Character Recognition	19
3.2 Optical Music Recognition	24
3.3 Summary	28
4 Matching Procedures	29
4.1 Retrieval Scenario	29
4.2 Evaluation Strategy	30
4.3 Text-based Matching	31
4.4 Score-based Matching	34
4.5 Summary	40
5 Retrieval Experiments	41
5.1 Text-based Matching Results	42
5.2 Score-based Matching Results	43
5.3 Oracle Fusion	46
5.4 Fusion	47
5.5 Summary	49
6 Applications and Conclusions	51
A Midi Number and Scientific Note	53

CONTENTS

B Subsets	55
C Inconsistencies	59
D List of Files were OMR Failed	61
E Source Code	67
Bibliography	71

Chapter 1

Introduction

Music played an inevitable role in the history of mankind. Cultures and ethnicities of every era in time are accompanied by their own music. In this 21st century, where digital revolution marked the beginning of the information age, music and its realm are changing tremendously. New methods of creation, storage and transmission of music data are examples for such changes. Large multimodal music collections in terms of textual, aural, and visual data came into existence¹². Efficient and convenient retrieval strategies are required to navigate and explore such music collections that are readily available [8, 17, 18, 19, 22]. The main challenge is to identify and establish semantic relationships across various music representations and formats. Key issues concern the development of methods for analyzing, comparing, correlating and annotating the available multimodal material. In particular for Western classical music, three prominent examples of digitally available types of music representations are sheet music (available as digital images), symbolic music data (e.g., score formats like MusicXML and LilyPond, piano roll representations, or the MIDI format), and audio recordings (e.g., given as WAV or MP3). These three classes of representations complement each other by describing different characteristics of music [15, 16]. The challenge now is to use all different modalities together to gain further knowledge about the underlying data.

In this thesis, we want to establish a semantic relationship between sheet music from the printed book “A Dictionary of Musical Themes” by Barlow and Morgenstern [4] and other digitally available databases. The book by Barlow and Morgenstern gives an overview of the 10,000 most important musical themes from Western classical music literature. Each theme consists of four bars of sheet music and additional information about the composer, the work and the theme title. An exemplary page of the book is given in Figure 1.1a.

The thesis consists of mainly two tasks: The first task is to split the scanned pages of the book by Barlow and Morgenstern [4] into single themes. This is done by segmentation techniques used in the field of image processing. The resulting images of the segmentation process ideally consist of the sheet music plus images including composer and work information as textual data (see Figure 1.1b). Symbolic data is extracted using Optical-Music Recognition (OMR) for the images containing sheet music, and Optical-Character Recognition (OCR) for the images containing textual data. The second task uses this text- and music-based symbolic data to match it against

¹http://en.wikipedia.org/wiki/List_of_Online_Digital_Musical_Document_Libraries

²http://en.wikipedia.org/wiki/List_of_online_music_databases

1. INTRODUCTION

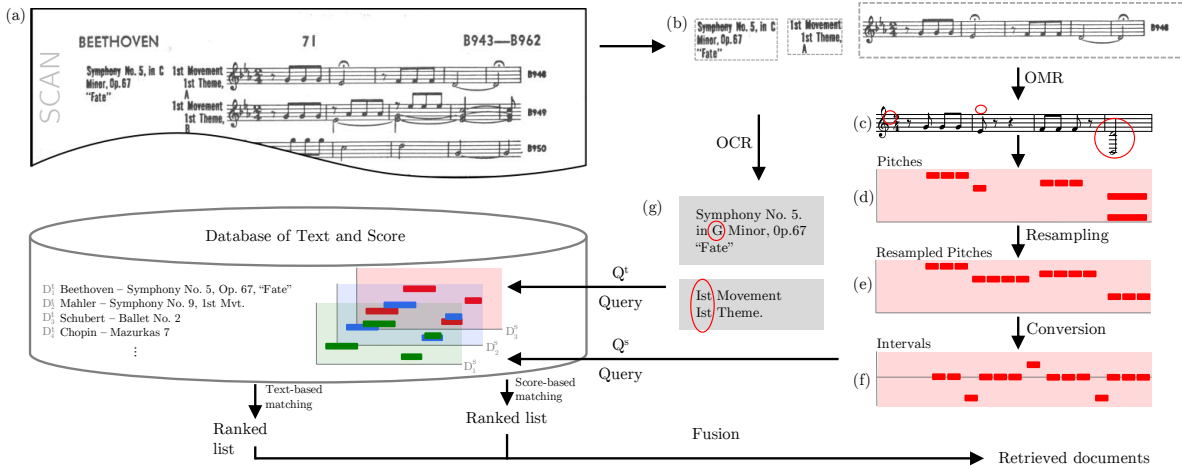


Figure 1.1: Overview of the processing pipeline closely related to [3]. Each page is segmented into text and sheet music parts. The cropped images are transformed into computer readable representations using OCR and OMR (typical extraction errors are highlighted by a red circle). The results are used to query against a database consisting of music documents. Using a fusion strategy based on text-based and score-based matching results, the retrieval system outputs a ranked list of documents.

other digitally available sources.

An overview of the designed retrieval pipeline is shown in Figure 1.1. The pipeline starts with the automated segmentation of the scanned pages (see Figure 1.1b), followed by applying OCR (see Figure 1.1g) and OMR (see Figure 1.1c) to the segmented images. This noisy symbolic data is used as a query to search for similar documents in a database. As a result, we receive possible candidates for the text-based and score-based symbolic data. Further experiments were made and evaluated to find the best match for a given query. For stabilizing the results, we combine the output of the text-based and score-based matching in a fusion step. We refer to the following chapters for a detailed explanation of each step of the pipeline.

1.1 Main Contribution

In this thesis, we deal with a challenging matching scenario by considering the book “A Dictionary of Musical Themes” by Barlow and Morgenstern [4]. This book yields an overview of the most important musical themes from the Western classical music literature, thus covering many of the pieces contained in International Music Score Library Project³ (IMSLP). The main contributions of this thesis are as follows.

First, we describe a fully automated processing pipeline that matches the music themes from the book by Barlow and Morgenstern to other digitally available sources. This pipeline involves automated segmentation, OCR, OMR, and alignment techniques (see Chapter 2, 4 and Figure 1.1). We also proposed modifications to be made on erroneous OMR data such that it performs better

³<http://imslp.org/>

with different alignment techniques.

Then, we report on extensive experiments that indicate the retrieval quality based on inconsistent and erroneous OCR and OMR input (see Chapter 5). In particular, we show how the quality can be improved by fusing the OCR-based and OMR-based matching results.

Finally, we discuss how our processing pipeline may be applied to automatically identify, retrieve, and annotate musical sources that are distributed in digital music archives and the world wide web.

1.2 Thesis Organization

In this section, an outline of this thesis is explained by briefly describing the contents of each chapter.

Chapter 2 deals with the automated segmentation of the images (scans), that serve as queries for the information retrieval task. This chapter also describes various segmentation methods for extracting individual components of the queries after clarifying various fundamentals used for the segmentation technique.

Conversion of images into symbolic notation is explained in *Chapter 3*, which basically deals with Optical Character Recognition and Optical Music Recognition. Different softwares tested and used for this recognition purpose are also explained in this chapter.

Symbolic notations extracted in Chapter 3 are matched to another database in *Chapter 4*. After the introduction of the database used for matching, various dynamic programming techniques used for the text and score based matching are also explained in this chapter.

Chapter 5 introduces various retrieval experiments along with the evaluation results of the same. Procedures and outcomes for combining text and score based information are also handled in this chapter.

Finally, *Chapter 6* informs us about various applications for which the cross modal information retrieval scenario can be applied. Conclusions and future scope of this thesis are also explained here.

Chapter 2

Automated Segmentation of Images

As starting point for our matching scenario, we use the book *A Dictionary of Musical Themes* [4] by Harold Barlow and Sam Morgenstern. It contains musical themes from the most important compositions of the Western classical music literature. It includes orchestral music, chamber music, and works for solo instruments. Each theme of this book is specified by a textual specification as well as a visual score representation of the notes. In particular, the respective composer, the underlying musical work, and the movement are listed. Within the book, the themes are systematically organized and suitably indexed. This chapter deals with the automated segmentation of these information, which will serve as the input for our retrieval scenario.

2.1 Description of the Images

The dictionary is divided into two parts. The first part contains 9803 musical themes arranged by the composers. Then comes the notation index or theme finder part that can help us to easily locate the theme which we are looking for in the first part. Apart from those parts, this book also contains a section named Index of titles which can help us to locate the theme if we know the work title. The themes contained in the book serve as a reference for the datasets, which are used as inputs for the retrieval scenario. In the following, we refer to these themes as ‘BM-themes’ and the information related to this book will be marked by a prefix **BM**. All the pages of the book were scanned in a grayscale image format with 600 dpi. Only the first part is used for this thesis. The scanned images of the first part were further processed to obtain all the elements mentioned in the Table 2.1.

2.1.1 Structure of BM-sheet Music

As show in Figure 2.1, each theme is associated with certain elements. They are given in Table 2.1. The original identifier given in [4], which we refer to as **BM-ThemeOrigID** (denoted by **g** in the figure), has numerous irregularities and exceptions (see Appendix C). Therefore we use **BM-ThemeID** as an identifier. It is a four digit code that simply enumerates the themes starting with 0001.

2. AUTOMATED SEGMENTATION OF IMAGES

The figure shows a page of musical notation with several themes. Red boxes highlight the following elements:

- (a) **BEETHOVEN**: Composer identifier for the first theme on the page.
- (b) **71**: Page number.
- (c) **B943—B962**: Range of theme IDs on the page.
- (d) **Symphony No. 5, in C Minor, Op. 67 "Fate"**: Musical work.
- (e) **1st Movement 1st Theme, A**: Description of the theme.
- (f) Musical score snippet for the 1st Movement 1st Theme, A.
- (g) **B948**: Theme ID for the 1st Movement 1st Theme, A.

Figure 2.1: Example of a scanned page from the book “A Dictionary of Musical Themes”. The various elements we consider here are highlighted with red boxes.

`BM-PageRangeComposer` helps us to understand the composer identifier of the first and last theme in the given page. In the figure shown, since the composer identifier is same for all the themes in that page, `BM-PageRangeComposer` is `BEETHOVEN`. This is denoted by `a` in the figure. However, if they are different, `BM-PageRangeComposer` will have last name of both the composers on those fields. `MOSZKOWSKI-MOZART` is an example for such a case. Page numbers are represented by the `BM-PageNumber`. This is denoted by `b` in the figure. The first and last `BM-ThemeOrigID` of a page are represented by `BM-PageRangeTheme`, which is denoted by `c` in the figure. `BM-Work`, represented by `d` in the figure, describes the musical work. `BM-ThemeDescription`, represented by `e`, gives the description of the theme. `BM-Composer`, the description of composer, also occurs at pages where the themes from a new composer begin.


Label	Element Name	Example
(a)	<code>BM-PageRangeComposer</code>	<code>BEETHOVEN</code>
(b)	<code>BM-PageNumber</code>	<code>71</code>
(c)	<code>BM-PageRangeTheme</code>	<code>B943-B962</code>
(d)	<code>BM-Work</code>	<code>Symphony No.5, in C Minor, Op.67 "Fate"</code>
-	<code>BM-Composer</code>	<code>Beethoven, Ludwig Van(1770-1827)</code>
(e)	<code>BM-ThemeDescription</code>	<code>1st Movement 1st Theme, A</code>
(f)	<code>BM-ThemeScore</code>	
(g)	<code>BM-ThemeOrigID</code>	<code>B948</code>

Table 2.1: Examples for the various elements of a scanned page from the book “A Dictionary of Musical Themes”. The labels refer to Figure 2.1.



Figure 2.2: Example of a binary image.

2.2 Segmentation Methods

Two methods used for segmentation are explained in the following sections. Method one (cf. Section 2.2.2) was used in the beginning when we were not having the complete information about the `BM-ThemeOrigID`. In this method, we had to manually control the segmentation and cropping of the musical themes. Since we were not having the `BM-ThemeOrigID`, we had to enumerate it while cropping was performed. This ID was required to name the cropped themes.

The knowledge about `BM-ThemeOrigID` was obtained by annotating it first. This was then used for naming (cf. Section 2.2.3). The program did not need user intervention as the exceptions and irregularities were handled in the implementation separately (see Appendix E).

2.2.1 Morphological Operations

The important concepts used in this thesis from the field of digital image processing are explained in this section. The term *morphology*, in our context, refers to a set of operations for extracting image components that are useful in the representation and description of region shape, such as boundaries, skeletons, and the convex hull [10]. In the following, we only deal with the morphological operations on binary images. There are mainly two kinds of morphological operations namely *dilation* and *erosion*.

2.2.1.1 Structuring Element

Binary morphological operations need a binary image B and a structuring element S as inputs. Let $\mathbb{B} := \{0, 1\}$ and the binary image $B \in \mathbb{B}^{M \times N}$, $\{M, N\} \in \mathbb{N}$ where M and N represent the size of the image. The structuring element S is basically a smaller binary image which represents a shape. It can also be of any size or can have an arbitrary structure. $S_{m,n}$ is used for denoting the translation of S such that the origin is located at (m, n) . Some of the basic structuring elements are as follows.

- **Line:** flat, linear structuring element that is symmetric with respect to the neighbourhood centre. An angle can also be applied to this element with respect to the horizontal axis.
- **Disk:** flat, disk-shaped structuring element with a particular radius.
- **Rectangle:** flat, rectangle shaped structuring element with a particular length and breadth.
- **Square:** flat, square shaped structuring element with a particular length.

It should be noted that our binary input image in Figure 2.2 is having ‘1’ as the background and ‘0’ as the foreground but the structuring element is having ‘0’ as the background and ‘1’ as the foreground.

2.2.1.2 Dilation

Dilation of a binary image B by the structuring element S is defined as a set of all points (m, n) such that $S_{m,n}$ hits B , i.e. they have a non empty intersection:

$$B \oplus S = \{(m, n) | S_{m,n} \cap B \neq \phi\}. \quad (2.1)$$

It is used for checking and expanding the shapes contained in the input image. Figure 2.3 shows the dilation of the image given in Figure 2.2 with a line structuring element of length 15 pixels.

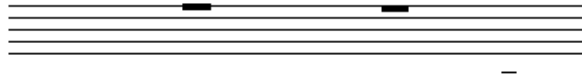


Figure 2.3: Dilation with line as structuring element.

2.2.1.3 Erosion

Erosion of a binary image B by the structuring element S is defined as a set of all points (m, n) such that $S_{m,n}$ is included in B , that is,

$$B \ominus S = \{(m, n) | S_{m,n} \subset B\}. \quad (2.2)$$

It is used to check and draw conclusions whether S fits or misses the input image. Figure 2.4 shows the erosion of the image given in Figure 2.2 with a line structuring element of length 15 pixels.



Figure 2.4: Erosion with line as structuring element.

2.2.1.4 Opening

Opening of a binary image B by the structuring element S is defined by erosion followed by dilation operation, that is,

$$B \circ S = (B \ominus S) \oplus S. \quad (2.3)$$

It removes small objects or islands from the foreground depending on S . Figure 2.5 shows the opening of the image given in Figure 2.2 with a line structuring element of length 15 pixels.



Figure 2.5: Opening with line as structuring element.

2.2.1.5 Closing

Closing of a binary image B by the structuring element S is defined by the dilation followed by erosion operation, that is,

$$B \bullet S = (B \oplus S) \ominus S. \quad (2.4)$$

It removes small holes or narrow channels from the foreground depending on S . Figure 2.6 shows the closing of the image given in Figure 2.2 with a line structuring element of length 15 pixels.

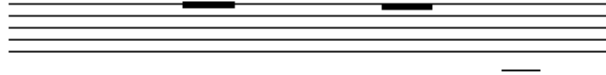


Figure 2.6: Closing with line as structuring element.

2.2.2 Method 1: Segmentation without Prior Knowledge

A segmentation method with the help of the above mentioned morphological operations is explained in this section. No prior knowledge about the themes is used for the segmentation purpose. This is an on the fly based cropping method. This program should be executed multiple times in order to crop the complete themes.

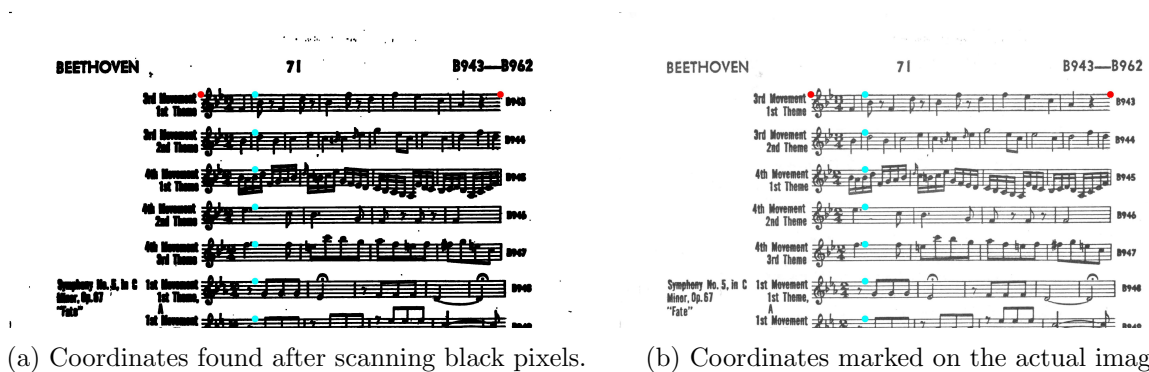


Figure 2.7: Various morphological operations are applied to the input image to find out the coordinates of musical themes. Red marks indicate the leftmost and rightmost pixels of the sheet music whereas the blue marks indicate the location of y-coordinates of each of the musical themes. Figure 2.7a is obtained by eroding the original image with square structuring element. Figure 2.7b shows the detected coordinates on the original image.

The important steps involved in the segmentation process of a scanned image are as follows.

1. Convert the scanned image into binary format.
2. Erode the binary image with a square structuring element of length 5 pixels.
3. Continuous black pixels in the horizontal direction are searched from the top of the image.



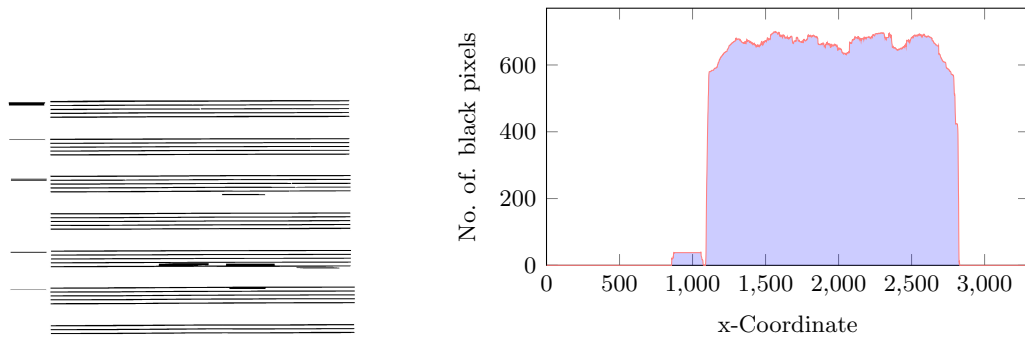
Figure 2.9: L and R positions are located by searching a white pixel in the directions marked from any one of the detected y-coordinate (blue).

Once all the coordinates of a particular page are found, we can start the cropping procedure. The cropping will be done in the original grayscale image which was kept in the computer memory. Let $n \in [1 : N]$ be the index of a staff on the current page where N represents the total number of staves at that page. So $loc(n)$ array contains the y-coordinate of the n -th staff on the current page. So the bounding box of this staff can be represented by $[L, loc(n), R-L, loc(n) + 200]$ which corresponds to $[x\text{-coordinate}, y\text{-coordinate}, \text{width}, \text{height}]$ of the bounding box. This region will be saved with the filename ' $\langle alphabet \rangle \langle index \rangle .jpg$ ' where **alphabet** and **index** are given as input data. The input data **alphabet** specifies the alphabet from which indexing should start. For example, if we want to crop **BM-ThemeOrigID** starting from 'B182' to 'B400', specify 'B' as **alphabet**. The input data **index** specifies the starting index which should be used for cropping. For example, if we want to crop **BM-ThemeOrigID** starting from 'B182' to 'B400', specify **index** as 182. 'B948.jpg' is an example for a file name given to a cropped theme. The index value will be incremented by 1 after a theme is cropped. Once all the themes of the particular page are cropped and saved, the next page is read and all of the above mentioned procedures are repeated. This procedure will continue till **index** reaches **indexmax**. The input data **indexmax** specifies the stopping index with which the program is terminated. For example, if we want to crop **BM-ThemeOrigID** starting from 'B182' to 'B400', specify **indexmax** as 400. Any irregularities or inconsistencies are handled by cropping the themes till that location by using **indexmax** and restarting the cropping program with a new **yloc**. The input data **yloc** specifies the location of the theme with which the cropping should be started. If we want to start the cropping from the top of the page (first location), the input should be '1'. By default, the value of **yloc** will be 1. But if we need to crop from an intermediate theme, we can do this by changing the value of **yloc**. For example, if we need to crop the theme 'B948' from page number 71 (see Figure 2.7b), then the value of **yloc** will be 6 because it is the sixth theme from the top of that page.

2.2.3 Method 2: Segmentation with Prior Knowledge

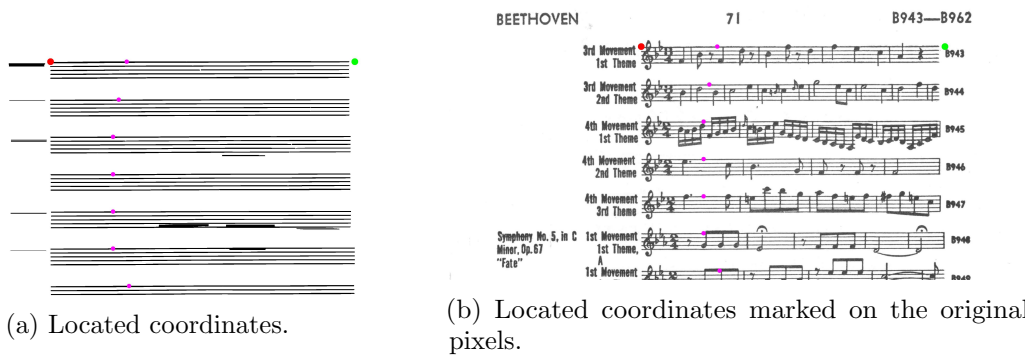
This section explains another segmentation method with the help of the morphological operations mentioned in Section 2.2.1. This method serves as a general model for all similar segmentation tasks if the theme identifier can be generated before the segmentation task. The theme identifier in our case is **BM-ThemeOrigID**. Since it is not consistent (see Appendix C), we introduce another four digit identifier **BM-ThemeID** to identify the musical themes.

2. AUTOMATED SEGMENTATION OF IMAGES



(a) Closing operation using 'line' structuring element (b) Histogram of the black pixels in the vertical direction

Figure 2.10: Procedure to find out the coordinates of all the themes on a scanned page. Figure 2.10a is used to detect the y-coordinates whereas Figure 2.10b is used to detect the leftmost and rightmost part of a theme.



(a) Located coordinates. (b) Located coordinates marked on the original pixels.

Figure 2.11: Intermediate steps while doing the segmentation is described using these figures.

The important steps involved in this segmentation method are as follows.

1. Convert the scanned image into binary format.
2. Close the binary image using 'line' structuring element of length 200 pixels (see Figure 2.10a).
3. Find the leftmost (L) and rightmost (R) part of the theme by taking histogram of black pixels in vertical direction (see Figure 2.10b).
4. Find the y coordinate of all the themes present in that page by checking the continuous horizontal lines in the closed image (see Figure 2.11a).
5. Crop all of the themes in the current page by using the coordinates obtained (see Figure 2.11b) and save it according to the inputs given (see Figure 2.14a).
6. Repeat from above steps for all the pages.

2.2.3.1 Description

In the input folder specified by `folder_name`, all the pages of the [4] should be present. The first and the last page of the book [4] which should be considered for cropping are passed into the MATLAB function as parameters. An excel file, which contains the identifier information, is also given as an input to this program. The content of the excel file should be in the format given in Table 2.2.

BM-ThemeID	BM-ThemeOrigID	BM-Composer
1061	B943	Beethoven
1062	B944	Beethoven
1063	B945	Beethoven
1064	B946	Beethoven
1065	B947	Beethoven
1066	B948	Beethoven
1067	B949	Beethoven

Table 2.2: An excerpt from the input excel file

After reading the input grayscale image, it is converted into the binary image file format, keeping the original image in the computer memory. All the computations are done on the binary image. This image is closed using a line structuring element of length 200 pixels. The result of this operation is shown in the Figure 2.10a. The closing operation extracts the five horizontal lines of all the staves. For finding the leftmost pixel (L) and the rightmost pixel (R), we can take the histogram of the black pixels in the vertical direction as shown in Figure 2.10b. The x axis in the figure indicates the pixel location in horizontal direction and the y axis indicates the total number of black pixels in the vertical direction. As we can see in the figure, there is a sudden jump and drop in the amount of black pixels. The location at which this sudden jump happens corresponds to L and the location at which the drop happens corresponds to R. Even if there are gaps in the horizontal lines of some of the staves due to printing or scanning errors, it is averaged out, resulting in the correct location of L and R. Y-coordinate in the closed image is found by counting the continuous black pixels in the horizontal direction as explained in the Section 2.2.2, which is equivalent to taking histogram of the black pixels in the horizontal direction.



Figure 2.12: Bounding box of a staff

For detecting the first horizontal line, we can have to search for continuous black pixels in the horizontal direction. According to the resolution of the image, a continuous 200 black pixels corresponds to one of the five horizontal lines of the staff. So if we search for continuous 200 black

2. AUTOMATED SEGMENTATION OF IMAGES

pixels from the top of the image, the first match will be at the first horizontal line of the first staff of the current image. Once detected, the corresponding y-coordinate is saved in the array *loc*. For visualization purpose, this coordinate is highlighted with a marker. For skipping the remaining 4 horizontal lines of the current staff, 190 pixels are skipped in the vertical direction. This searching process is repeated till the end of the current page resulting in the array *loc* which will be filled by the y coordinates of all the staves. Once all the coordinates of a particular page are found, we can start the cropping procedure. The cropping will be done in the original grayscale image which was kept in the computer memory.



Figure 2.13: Instances in which notes of a particular theme are very close to the same of the nearby themes.

After considering the resolution of the image (600dpi), the distance from first horizontal line to the fifth line of a staff is 100 pixels. Let $n \in [1 : N]$ be the index of a staff on the current page where N represents the total number of staves in that page. So $loc(n)$ contains the y coordinate of the n th staff on the current page. So the bounding box of this staff can be represented by $[L, loc(n-1)+100, R-L, loc(n+1)-(loc(n-1)+100)]$ which corresponds to $[x\text{-coordinate}, y\text{-coordinate}, \text{width}, \text{height}]$ of the rectangle. The bounding box of a particular staff is shown in the Figure 2.12. This region will be saved with the filename ' $\langle BM\text{-ThemeID} \rangle\text{-}\langle BM\text{-ThemeOrigID} \rangle\text{.jpg}$ ' where $BM\text{-ThemeID}$ and $BM\text{-ThemeOrigID}$ are given as input data. '1066_B948.jpg' is an example for a file name given to a cropped theme. $BM\text{-ThemeID}$ value will be incremented by 1 after a theme is cropped. In short, we extract all the available pixel information of the current theme as shown in Figure 2.14a and then post process the cropped theme to clear the notes from the nearby themes. The result of such a cleaned theme is also shown in the Figure 2.14b. The details of post processing is explained in Section 2.2.3.2. There are cases in which the notes of a particular theme are very close to the notes of the nearby themes. Such a case is highlighted in the Figure 2.13. These problems can also be handled by this method very easily. However, it is very difficult to automatically clean the data, since the notes of two themes are connected to each other.

2.2.3.2 Post Processing

Segmentation method explained in the Section 2.2.3 will try to include all the available information about a theme. However, this may result in the inclusion of notes or parts of notes from near by themes as shown in Figure 2.14a. So an efficient method to clean the themes from the near by themes should be performed. For this reason, a post processing program is executed to clean the themes. The details of the post processing MATLAB function is explained in Appendix E. Figure 2.14b shows the cleaned output of Figure 2.14a.

The post processing function cleans the theme by suppressing the structures, that are darker than



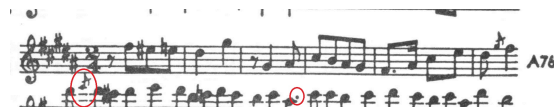
(a) Saved BM-ThemeScore.



(b) Cleaned BM-ThemeScore.

Figure 2.14: Input and output of the post processing method.

their surroundings, connected to the image border. It suppresses the borders in an 8 dimensional neighborhood. This function will also threshold the image such that white becomes whiter. These will clean majority of the cropped themes in a better way. However, if the note or parts of notes from another theme is not connected to the image border, then this cleaning algorithm will exclude such components. Such a case is shown in the Figure 2.15a where the region of interest is highlighted with red color and its output in Figure 2.15b.



(a) Notes highlighted in red can't be cleaned by post processing.



(b) Output of the post processing function when theme in Figure 2.15a is given as input.

Figure 2.15: Case in which post processing fails to clean all the notes of the nearby themes. This happens because these symbols are not connected to any of the themes.

Apparently, if the notes of the same theme are connected to the border, then the complete theme will be cleared. This is because such notes of a theme will be connected to all the other notes of the same theme by the staff. This case is specified with the example shown in Figure 2.16. In order to mitigate this problem, we crosscheck the output of a theme from the cleaning procedure. If the cleaned theme is not having any staves in the image, then the cleaning procedure is reverted for that particular theme and the output of such a theme will be the input itself.



Figure 2.16: Beam of a theme (highlighted in red) is connected to the same of a nearby theme.

2.3 Summary

This chapter explained the structure and contents of the book ‘A Dictionary of Musical Themes’[4]. Not only the two automated segmentation methods that are used to process this book but also the various operations required to elucidate these segmentation methods are explained in this chapter. Both the methods are having advantages and disadvantages. The advantage of the method without prior knowledge (see Section 2.2.2) is that we can crop the themes on the fly. All irregularities or exceptions can be handled easily. However, the disadvantage of this method is that the program should be executed multiple times according to the frequency of the irregularity. Human interventions are needed to handle these irregularities or exceptions.

The main advantage of the segmentation method with the prior knowledge is that it is fully automated. Human interventions are not needed in this method once the program is started. The segmentation and cropping are done automatically. The results will also be saved according to the file names specified by the excel file which was given as the input (see Section 2.2.3 for more details). However the downside of this method is that the irregularities and exceptions must be hard-coded into the MATLAB function in advance. But once the exceptions are hard coded into the program, then it can be used any number of times.

For the purpose of this thesis, we have used the results of the segmentation method with prior knowledge.

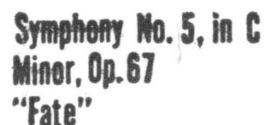
Chapter 3

Text and Score Recognition

This chapter deals with the conversion of the cropped images into symbolic notation. Optical character recognition is used for the conversion of text-based information and optical music recognition for the same of score-based information. `BM-Work` and `BM-ThemeDescription` constitute the text-based information and `BM-ThemeScore` serves as the score based information (cf. Table 2.1). We use different software for the above mentioned purpose and the details are explained in the subsequent sections.

3.1 Optical Character Recognition

Extracting various elements such as `BM-Work`, `BM-ThemeDescription` etc. helps us in linking `BM`-themes and the `EDM`-themes. The more accurate the results are, the better will be the cross modal matching. An automated segmentation procedure was done to `BM`-sheet music given in Figure 2.1 to separate the `BM-Work` and `BM-ThemeDescription` from the `BM`-sheet music. Figure 3.1a shows a sample `BM-Work` and Figure 3.1b shows a sample `BM-ThemeDescription`. By the segmentation procedure, we can easily map the cropped outputs directly to the `BM-ThemeID`. Adobe[®] Acrobat[®] XI Pro Version 11.0.0 and Tesseract OCR Version 3.02.02 OCR softwares/engines were tested for recognizing the characters of `BM-Work` and `BM-ThemeDescription`.



**Symphony No. 5, in C
Minor, Op. 67
"Fate"**

(a) `BM-Work`



**1st Movement
1st Theme,
A**

(b) `BM-ThemeDescription`

Figure 3.1: A sample element extracted from `BM`-sheetmusic

3. TEXT AND SCORE RECOGNITION

3.1.1 Adobe Acrobat XI Pro

Acrobat XI Pro has an inbuilt engine for text recognition. The segmented image files (`BM-Work` and `BM-ThemeDescription`) are given directly to the software by creating a macro with the action wizard. This macro could handle batch processing such that when the input image folder is specified, it reads the image files with the extension `'jpg'`, performs the character recognition and finally saves the output file in `'txt'` format with the same file name as the input image file. So after executing the macro, `'txt'` files corresponding to the input images will be present in the same folder.

Since the software is having an inbuilt OCR engine, prior knowledge about our image cannot be specified as input. Prior knowledge includes frequency of a word in an image, dictionary of the words that can come in the image, occurrence of special symbols or characters etc. The result of the text recognition is specified in the Section 3.1.4. If the input image can't be read by the software, then a blank output file will be created by the software.

3.1.2 Tesseract OCR

Tesseract OCR is one of the best available open source OCR engines released under Apache License 2.0¹. It uses Leptonica image processing library. It supports a wide variety of input image formats and over 60 languages. Moreover, it is fully trainable. It is very easy to make a custom made OCR engine with Tesseract. Even without training the engine, we can modify the frequency word list and add our own dictionary, which will help us to get better results. Following steps are needed to modify the inbuilt trained data.

1. Unpack the inbuilt trained data by using the command `combine_tessdata`.
2. Decode the word list and the frequency list by using the command `dawg2wordlist`.
3. Update the word list and the frequency list according to our prior information.
4. Code the word list and the frequency list by using the command `wordlist2dawg`.
5. Combine the updated files as trained data by using the command `combine_tessdata`.

By using the updated trained data and a MATLAB script, the OCR process can be automated. The input folder contains the `'jpg'` images of various elements are read by MATLAB which in turns invoke the Tesseract engine. The output `'txt'` files will be saved with the same filename as the given input image in a specified folder.

3.1.3 Error Sources

The possible error sources in the OCR process are explained in this section. Errors can occur also in the scanning procedure. For example, a slight tilt in the scanned page, which is not easily noticeable with the human eye, makes the OCR output worse. All the images are scanned in `'jpeg'` format at 600dpi resolution. Some of the major problems which deteriorated the OCR result are as follows.

¹<https://code.google.com/p/tesseract-ocr/>

3.1.3.1 Condensed Input Data

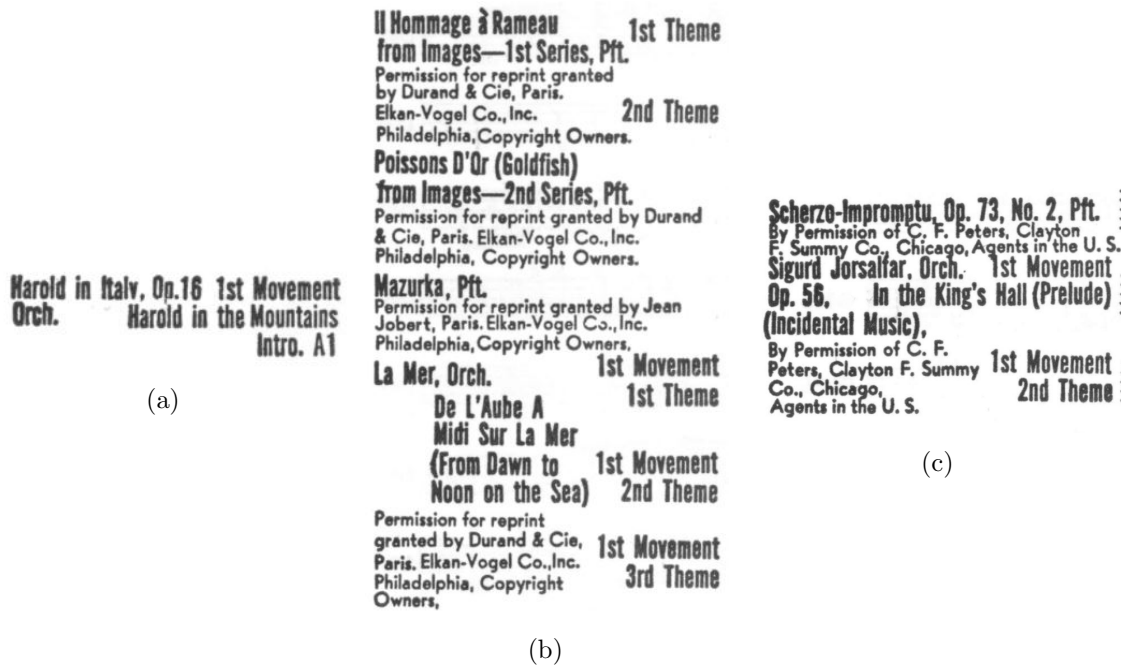


Figure 3.2: Error sources due to condensed input data.

When the `BM-Work` and `BM-ThemeDescription` texts given in the book [4] are condensed, errors can happen in the segmentation which in turn will result in bad OCR results. Typical examples of the condensed text data are given in the Figure 3.2. As we can see in this figure, `BM-Work` and `BM-ThemeDescription` along with the permission and copyright information makes it very hard for even humans to distinguish. For Figure 3.2a, the actual `BM-Work` is “Harold in Italy, Op.16 Orch”. and the corresponding `BM-ThemeDescription` is “1st Movement, Harold in the Mountains Intro A1”. In this case, assuming a perfect recognition of text information, the OCR will give the output as “Harold in Italy, Op.16 1st Movement Orch. Harold in the Mountains Intro. A1”. In Figures 3.2b and 3.2c, the permission and copyright information are so densely packed between `BM-Work` and `BM-ThemeDescription` such that it cannot be properly segmented and thus the entire image will be assigned as `BM-Work` of a theme and the corresponding `BM-ThemeDescription` will be empty. Some of the errors like the one explained in Figures 3.2b and 3.2c are compensated by an additional manual cropping. All the cropped images above a particular threshold dimension are re-cropped manually.

3.1.3.2 Segmentation Error

Segmentation errors will occur also due to the condensed input data, which is explained in the previous sub section. However, there are also some other reasons due to which these errors occur. If there is a slight tilt in the orientation of a page, which can probably happen due to scanning error, the segmented image will contain a very small portion of the nearby staff. This is shown in the Figure 3.3a. According to the tilt direction, error in segmentation can happen to the `BM-ThemeDescriptions` at the top or at the bottom of a page. Another reason for segmentation

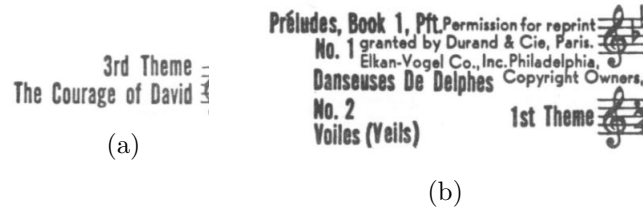


Figure 3.3: Error sources due to segmentation.

error is explained in the Figure 3.3b, where the text information is very close to the staves. The main cause of these kind of errors come from the input images, where permission and copyright information are written.

In both these scenarios, since a small portion of the musical staff is contained in `BM-ThemeDescription`, OCR engine will produce undesired symbols or characters.

3.1.3.3 Print Quality

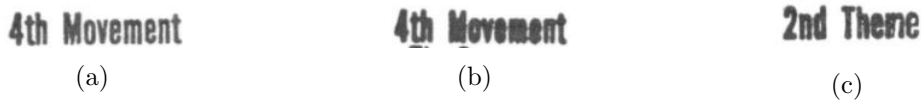


Figure 3.4: Error sources due to print quality.

Another cause of the poor OCR result is the print quality. Since the print quality of the book is not unique, various errors will occur. Some of the `BM-ThemeDescription` from different pages are given in the Figure 3.4. Figure 3.4a is an example where the OCR engine can produce great result since the ink density is optimal for an OCR engine. However in Figure 3.4b, this is not the case. Ink flow is so thick that the OCR engine misinterprets the data. Figure 3.4c explains a case where an error occurred in typesetting. All of these errors can create a huge impact on the OCR results. Nevertheless of all these errors, OCR engine can give better results if trained properly.

3.1.3.4 Language



Figure 3.5: Error sources due to the language of the input data.

Language of the text data also plays a key role in contributing to errors, especially if the OCR engine is not trained or not using a user given dictionary to process the data. The default language of OCR engine is set to 'English'. If the input data is not in this language and the OCR engine is not given an dictionary to process the data, it will generate an output which is closest to the English alphabet or numbers.

3.1.4 OCR Results

This section shows some samples of the results obtained from the OCR engines. Table 3.1 shows the result of the OCR engines when **BM-Work** images were given as input and Table 3.2 shows the result of the same when **BM-ThemeDescription** images were given as input. The first column of both the tables contain the input images which are extracted automatically using our segmentation method (see Section 2.2.3). The second and third column contain the output of Acrobat XI Pro and Tesseract OCR respectively.

Input Image	Acrobat XI Pro	Tesseract OCR
Symphony No. 5, in E Minor, Op.95 "From The New World" Published and Copyrighted 1928 by Oliver Ditson Co.	ony o. 5,in E inor, Op. 95 From The He World" Pub- lished end Copyrighted 1928 by Oliv Ditson Co.	Symphony lo. 5, in E Mi- nor, Op.!!5 "From The New World" Publishod and Copy- rihtod 1928 by Oiivor Dit- son Co.
Symphony No. 3, Op. 42 "Ilia Mourometz"	Symphony 3,0J.42 "Ilia ourometz"	Symphony No. 3. On. 42 "Ilia Mourometz"
Entry & Dance of the Tailors 1st Theme	&Dance of the T rs 1st Tile	Entry & Dance of the Tailors 1st Theme
Euryanthe, Overture	Euryantbe, Overture	Euryanthe. Overture

Table 3.1: Sample OCR results of **BM-Work**

There was a total of 9803 **BM-Themes** in the book. 9769 **BM-Works** were detected by the segmentation process. Out of these 9769 **BM-Work** images, Adobe Acrobat was not able to process 1782 files and Tesseract about 995 files. Out of 8883 **BM-ThemeDescription** images, Adobe Acrobat was not able to process 2861 files whereas for Tesseract it was only 272 files. A blank text file was produced for these images by both the softwares.

Input Image	Acrobat XI Pro	Tesseract OCR
1st Movement 1st Theme, A	1st 1st T A	Ist Movement Ist Ihene.
1st Movement Intro. A	1st ovement Intro. A	1:!! Movement Intro.A
5th Movement : Gigue :	5th ovement : Gigue :	5th Moveugent 3 Gigue =
2nd Movement : 1st Theme : Scherzo	2nd Movement _ 1st Theme Scherzo	2nd Movement. Ist Theme 5 Scherzo '
1st Movement 1st Theme, A	1st taoument 1st tbeme,	Ist Movement Ist Iheme.

Table 3.2: Sample OCR results of **BM-ThemeDescription**

As these examples suggest, Tesseract OCR outperforms Adobe Acrobat Pro with respect to the total number of detected input images, where text was recognized and the quality of the output. For this reason, Tesseract OCR are used for the cross modal matching described in Chapter 4.

3.2 Optical Music Recognition

Optical Music Recognition (OMR) is the process of identifying or recognizing musical symbols from sheet music such that it can be played or processed by a computer or digital instrument. It is also known as Music OCR because it is considered as an extension of Optical Character Recognition. OMR has been the focus of international research over three decades. The important stages in the recognition process are staff line identification, musical object location, musical feature classification, and musical semantics [2]. Many of the OMR engines available on the market are far away from being perfect. This results in subsequent error correction methods [6].

For the purpose of our thesis, we have used many different softwares. But for the sake of brevity, we are only mentioning two of them which were very useful for us. The output of the OMR process were stored in a MIDI file format because of the ease with which we can evaluate the results.

3.2.1 Neuratron PhotoScore Ultimate 7

Neuratron PhotoScore Ultimate is the full-featured version of the PhotoScore Lite scanning software for Sibelius. We can use this software for scanning, playback, music recognition and printing musical scores. For the purpose of this thesis, PhotoScore Ultimate version 7.0.2 was used, which is a part of Sibelius version 7.1.2 software. This proprietary software claims to have better OMR engine, even with the support for handwritten music². This was one of the main reason why we used Photoscore for our testing purpose.

We can specify whether the musical themes to be recognized are handwritten or printed in advance with this software. However, we cannot run the OMR engine in batch processing mode. So doing music recognition for around 10,000 files is a herculean task. Nevertheless, we have tested this software against BM-Mini data subset (cf. Appendix B) which contains around 26 files. The outputs were extracted in the internal file format of this software which was read by the Sibelius which in turn converted it into MIDI format.

3.2.2 Audiveris V4.3

Audiveris is an open-source OMR software which processes the image of a music sheet to automatically provide symbolic music information in MusicXML standard [5]. For the purpose of this thesis, Audiveris version 4.3.3406 was used. This software accepts only printed music as input file which is of the format PDF, JPG, PNG, TIFF, BMP or similar. It outputs the musical scores in MusicXML version 3 format. This software is also trainable, however for our purpose, we use the default OMR engine. One of the main features of this software is that it can

²<http://www.neuratron.com/v7.htm>

do batch processing. This software was also tested against BM-Mini data subset (cf. Appendix B) which contains around 26 files. Since the output of this software is in MusicXML format, we use MuseScore version 1.3 to convert this software to MIDI format. We use the *Batch Export* plugin of this software to do MIDI conversion in batch processing mode.

3.2.3 Error Sources

Some of the major error sources that deteriorated the OMR results are explained in this section. OMR failed to convert 1788 queries, the filenames of which are given in Appendix D. One or more of the below mentioned error sources were the most important reason for this failure.

3.2.3.1 Print Quality



Figure 3.6: Error sources due to print quality.

One of the main reason for the OMR errors is the input image quality itself. Two of the bad input images are show in the Figure 3.6. Printing error is depicted in Figure 3.6a, which resulted in broken staff lines, stems or other music symbols. In Figure 3.6b, the ink is so thick that it is hard to differentiate between naturals, sharps and flats. It can also happen that it is very difficult to differentiate between half and quarter notes for an OMR engine if the ink is too thick. However in both these examples, OMR engine failed to process the queries.

3.2.3.2 Segmentation Error



Figure 3.7: Themes like these causes segmentation errors which inturn gives an erroneous OMR

When the notes of two themes are too close to each other, then it will be very difficult to segment and clean the data. As a result, as shown in Figure 3.7, the notes will produce errors at the OMR engine.

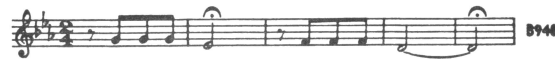
3. TEXT AND SCORE RECOGNITION

3.2.3.3 Font



Figure 3.8: Font marked in red is very difficult to read for many OMR engines.

The font with which the time signatures are written along with the staff lines are so difficult for majority of the OMR engines to correctly detect the time signatures. An example is given in the Figure 3.8. Another example for this type of errors is give below. As you can see in this



(a) Input to the OMR engine.



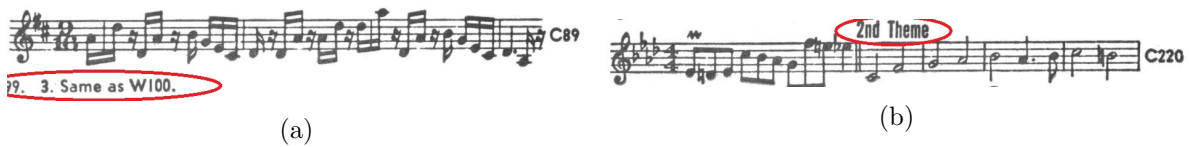
(b) PhotoScore output.

(c) Audiveris output.

Figure 3.9: Font information is misinterpreted by both the OMR engines. Figure 3.9a is the input given to the OMR engines and Figures 3.9b and 3.9c shows its output.

figure, both the OMR engines are not able to recognize the time signatures. The error shown here occurred not only due to the bad font, but also to the poor print quality.

3.2.3.4 Text Information



(a)

(b)

Figure 3.10: Text information near the themes, as marked in red, can result in unwanted OMR errors.

When text information are present near the themes, especially if they are not related to any musical symbols, they can contribute to errors in OMR engine. Some typical examples are shown in Figures 3.10a and 3.10b. The text is marked in red color in Figure 3.10. When this is fed to an OMR engine, it will generate notes or symbols close to the shape of the text or its alphabets. For example the text “2nd Theme” in the Figure 3.10b was interpreted as \flat and \sharp .

3.2.3.5 Post Processing Errors

Post processing is done to the segmented themes to clean the notes or related parts of the nearby themes. However, there are certain symbols, like the flat symbol marked in Figure 3.11,



Figure 3.11: Post processing was not able to clear the marked error.

which belongs to the nearby theme. Such information will not be cleaned by the post processing algorithm and it can aid OMR errors. Figure 3.12 shows such an example. As we can see in



(a) Input to the OMR engine. The red circle indicates the connection between the notes of the nearby themes due to which the post processing fails.



(b) PhotoScore output.



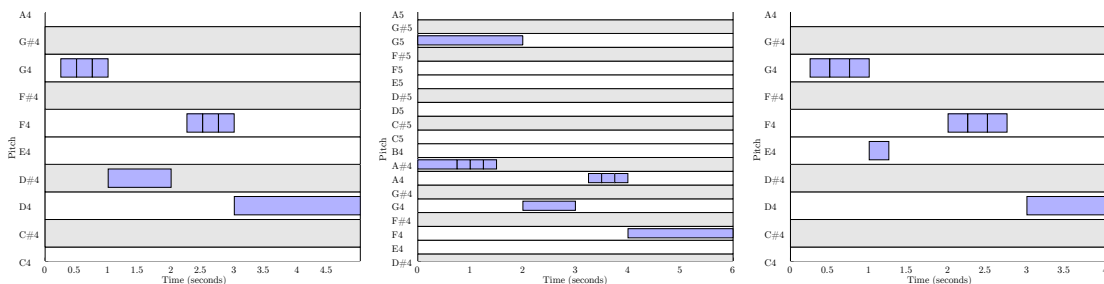
(c) Audiveris output.

Figure 3.12: Post processing was not successful in removing the nearby themes. Figure 3.12a is the input given to the OMR engines and Figures 3.12b and 3.12c shows its output.

the Figure 3.12a, the red circle indicates the connection between the notes of two neighboring themes. Due to such connections, post processing cannot clean the theme. This will eventually results in OMR errors as shown in Figure 3.12b and 3.12c.

It can be inferred from the above mentioned examples that the majority of key and time signatures of the input themes are not read by both the softwares. We have found, after observing the outcome of BM-Mini data subset, that majority of the articulation marks such as *Fermata* or *Staccato*, octave sign such as *Ottava*, ornaments such as *Turn* or *Trill*, dynamics and rest symbols are missing. However majority of the notes are preserved.

As mentioned in the previous section, the output of the OMR engine is saved as a MIDI file.



(a) Actual OMR output.

(b) PhotoScore output.

(c) Audiveris output.

Figure 3.13: Comparison between the outputs of OMR engines.

But for comparing different MIDI files, we use a mid level representation called *Piano roll representation*. This can be considered as a two dimensional graph or display with *time* on the x-axis and *pitch* on the y-axis, where the information on musical onset times, pitches and note durations are given explicitly. For comparison purpose, the piano roll representation of the query given in Figure 3.9 is given in Figure 3.13. Figures 3.13b and 3.13c show the output of OMR engines and Figure 3.13a shows the same theme without any OMR errors.

As we can infer from the Figure 3.13, using the MIDI pitches directly for the cross modal matching will end up in very poor results. Many of the notes are transposed upto ± 12 semitones. Duration of notes also differs from each other and silences are added at many places depending on the OMR errors. These effects can be seen clearly from the time axis of three piano roll representations. In addition to these artifacts, OMR changed the monophonic inputs to polyphonic sounds. So the MIDI pitches that we use for cross modal matching should be treated in a way such that it mitigates these effects. Such a method is explained in the Section 4.4.

After observing the outcome of both the programs, we have found out that it is better to continue our experiments with Audiveris program since it outperformed the Photoscore program. Moreover, batch processing mode is supported by it, which makes our conversion task easier.

3.3 Summary

It is important at this stage to recapitulate the main findings from this chapter where we used many softwares for the OCR and OMR process. From here on, we will only use the results of Tesseract OCR engine and Audiveris OMR engine. After comparing the results of both the OCR softwares (see Table 3.1), it is a good prognosis to say that Tesseract OCR works better. Adobe Acrobat failed to recognize any character for more than 18% of the total images. The trained Tesseract engine performed way better than the Acrobat engine with a better accuracy. Moreover, this is an open-source software that can be trained or modified to tailor our needs. Audiveris outperformed the Photoscore software because of various reasons. The main reason was of course the quality of the retrieved results. Ability to do the batch processing mode also helps Audiveris to surpass Photoscore. Moreover, Audiveris is an open-source source software where as Photoscore is a proprietary software. The retrieval experiments explained in Chapter 5 will corroborate these decision.

Chapter 4

Matching Procedures

An information retrieval (IR) process begins when a user specifies what he or she needs by means of a query. The retrieval system should then retrieve from a given data collection all documents or objects that are somehow related to the query. Suppose the user needs to find a particular word in this thesis document. One way to finish the task is to read through the complete thesis document. The easiest method for a computer to do the same IR task is to do a linear scan through the document. However, if the number of documents to be searched is large, this is not a practical method since it is a time consuming task. Another method is to do *indexing* which will improve the effectiveness of information retrieval systems. There are many new methods available for IR if the query is text information [1, 13, 24, 23]. However, if the user wants to use a query containing non-textual data, how should we compare and link the query to the data collection? On the other hand, what if the database documents contain non-textual information? This chapter explains one method to answer these questions where the query is an image and the database document contains text and symbolic music data.

4.1 Retrieval Scenario

As a result of the recognition process described in Section 3, we obtain a textual representation of the metadata (work and theme identifier) and a symbolic score representation for each of the 9803 themes of the book by Barlow and Morgenstern [4]. The goal is to use this information for identifying other digital sources that belong or relate to the musical themes. In our experiments, we consider a scenario that allows us to study various matching procedures and to systematically evaluate matching results. To this end we consider the “Electronic Dictionary of Musical Themes” (EDM), which is publicly available at [21]. The EDM collection contains 9825 Standard MIDI files for the musical themes, which are linked to textual metadata similar to the original book by Barlow and Morgenstern.

In the following, we formulate our setting as a retrieval task. We denote the set of BM themes by \mathcal{Q} , where each element $Q \in \mathcal{Q}$ is regarded as a *query*, with X being the total number of BM themes in [4]. Furthermore, let \mathcal{D} be the set of EDM themes, which we regard as a database collection consisting of *documents* $D \in \mathcal{D}$ and let Y correspond to the total number of EDM themes from the website [21]. Given a query $Q \in \mathcal{Q}$, the retrieval task is to identify the

semantically corresponding document $D \in \mathcal{D}$.

4.2 Evaluation Strategy

Evaluation is a very important step to assess the quality of IR. The method to evaluate the above mentioned retrieval task is explained in this section.

4.2.1 Ground Truth

While the EDM themes more or less agree with the BM themes, there are inconsistencies with regard to the number of themes, the metadata and the score representations. Using the printed BM book as a reference, we have manually linked the BM themes to corresponding EDM themes. For each BM theme, there is only one relevant document in the EDM database. These correspondences serve as ground truth in the subsequent experiments. For 174 out of 9803 music themes, we were not able to find the ground truth.

4.2.2 Evaluation Measures

Evaluation measures used for the retrieval task are explained here. These evaluation measures help us not only in analyzing the effects of various parameters used in this thesis, but also in identifying parameters that did not have any influence in the retrieval task. The two different measures are as follows.

4.2.2.1 Mean Rank

The mean rank (\bar{R}) is defined as the arithmetic mean of the ranks R_x , where R_x is the rank obtained for each query element Q_x . Suppose we have the set of ranks containing the values R_1, R_2, \dots, R_X for queries Q_1, Q_2, \dots, Q_X , then the mean rank is defined by the equation

$$\bar{R} = \frac{1}{X} \sum_{x=1}^X R_x \quad (4.1)$$

where $R_x \in \mathbb{N}$ such that $R_x \leq Y$, where Y corresponds to the total number of elements in the database. However, the disadvantage of this measure is that outliers have a strong influence and lead to fluctuations of this measure. Another disadvantage of this measure is that it alone will not give an overview of the efficiency of a particular algorithm. The mean rank should be interpreted relative to the size of the database. For example, if the size of the database is 10,000 and the mean rank is 20, it can be considered as a good algorithm where as for the same mean rank, the algorithm is considered as poor when the database size is 25.

4.2.2.2 Top N Match

A Top N list is a matching list to an information retrieval system that does not provide the complete matching list but only the top N or the first N results of the original matching list. The Top N match is defined as the percentage of queries which contain the correct match (ground truth) in its top N list. It is mathematically defined as

$$T_N = \frac{M}{X} \cdot 100\% \quad (4.2)$$

where $M \in \mathbb{Z}_{\geq 0}$ such that $M \leq X$, where X corresponds to the total number of queries and M to the number of queries for which the correct element (ground truth) appears in the top N list. N for the Top N list is defined as $N \in \mathbb{N} | N \leq X$.

4.2.3 Distance Matrix

Distance matrix, $\mathbb{D} \in \mathbb{R}^{Y \times X}$, is a two dimensional matrix containing the distances between the elements of the BM queries \mathcal{Q} and EDM database \mathcal{D} . It contains X columns and Y rows where X corresponds to the total number of queries in \mathcal{Q} and Y to the total number of elements in \mathcal{D} . This matrix helps us to understand the distance between each $Q \in \mathcal{Q}$ to every $D \in \mathcal{D}$. The higher the distance between Q and D , the lesser the similarity. The content of this matrix can be assigned as follows

$$\mathbb{D}(Q, D) := \begin{cases} c^t(Q, D) & \text{if text-based matching} \\ c^s(Q, D) & \text{if score-based matching} \end{cases} \quad (4.3)$$

where the functions c^t and c^s are explained in Sections 4.3 and 4.4 respectively.

4.3 Text-based Matching

Let us consider a fixed query $Q \in \mathcal{Q}$. In a first matching procedure, we only consider the textual representation, denoted by Q^t , which was obtained from the OCR step. Similarly, let D^t denote the text information for a document $D \in \mathcal{D}$. Both Q^t as well as D^t are represented as character strings. To compare these strings, we use standard string alignment techniques such as the edit distance or longest common subsequence [7]. In our scenario, the two strings to be compared contain a work descriptor as well as a movement and theme identifier. However, the strings may also differ substantially due to additional information, segmentation errors, and OCR errors. An example for common additional information and segmentation error is shown in the Figure 4.1 with red and blue color respectively. For the Q^t given in this figure, the corresponding D^t is “Iberia I, Pft., Evocacion”. The OCR output for this particular Q^t given in Figure 4.1 is “Iberia t, ,Evocacion By permission of Associated Music Publishers, Inc, Fite Dieu i -eville”. The additional information such as permissions or copyrights are not available in the database \mathcal{D} .

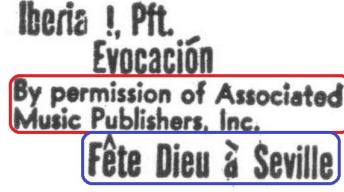


Figure 4.1: Additional information marked in red and segmentation error in blue.

4.3.1 Edit Distance

Edit distance (ED) is also known as Levenshtein distance¹. ED is a method of evaluating two different strings by counting the minimum number of operations required to transform one string into another. The operations (edits) used in ED are *deletion*, *insertion* and *substitution* [12]. The larger the ED value, the greater the cost in transforming one string to another. Let the two strings be defined as follows: $u = \{u_1, u_2, \dots, u_m, \dots, u_M\}$ and $v = \{v_1, v_2, \dots, v_n, \dots, v_N\}$. ED between two strings can be found using the ED matrix \mathbf{D} . The elements of the $\mathbf{D} \in \mathbb{N}^{(M+1) \times (N+1)}$, can be found using

$$\mathbf{D}(0, 0) = 0 \quad (4.4a)$$

$$\mathbf{D}(m, 0) = m, 1 \leq m \leq M \quad (4.4b)$$

$$\mathbf{D}(0, n) = n, 1 \leq n \leq N \quad (4.4c)$$

$$\mathbf{D}(m, n) = \begin{cases} \mathbf{D}(m-1, n-1) & \text{if } u_m = v_n \\ \min \begin{cases} \mathbf{D}(m-1, n) + 1 & \text{(deletion)} \\ \mathbf{D}(m, n-1) + 1 & \text{(insertion)} \\ \mathbf{D}(m-1, n-1) + 1 & \text{(substitution)} \end{cases} & \text{else} \end{cases} \quad 1 \leq m \leq M; 1 \leq n \leq N \quad (4.4d)$$

and the edit distance is the content of the cell $\mathbf{D}(M, N)$ which will give us the minimum number of operations required (note that the index starts with 0). For example, the ED matrix \mathbf{D} between the strings *music* and *mucik* is given in the Table 4.1. In this table, m and n indicate the indices of the string *mucik* and *music* respectively. As we can see from the table, the ED value between these two strings is 2 since two edit operations are required to transform one word to another.

For the purpose of comparing different string comparison methods, we use the normalized version of ED. In literature [14, 9], normalization is done using the length of the shortest edit path, the length of the larger string or some similar lengths.

As discussed in the beginning of this chapter, for a query Q^t and database document D^t , the edit distance operation can be defined as $ED(Q^t, D^t)$ and its cost is assigned as

$$c^t(Q, D) := \frac{ED(Q^t, D^t)}{|Q^t|}, \quad (4.5)$$

where $|Q^t|$ denotes the length of the string Q^t . In this thesis all normalizations are done by

¹http://en.wikipedia.org/wiki/Levenshtein_distance

		n	0	1	2	3	4	5
m		v						
	u		m	u	s	i	c	
0			0	1	2	3	4	5
1	m		1	0	1	2	3	4
2	u		2	1	0	1	2	3
3	c		3	2	1	1	2	2
4	i		4	3	2	2	1	2
5	k		5	4	3	3	2	2

Table 4.1: ED matrix \mathbf{D} between the strings music and mucik, in which ED value is highlighted in red

the length of the query unless it is specified. Usually the term normalization means that it is bounded by zero and one. However, for ED, the upper bound is not one since it is normalized by the length of the query. The value crosses 1 especially for shorter queries and longer database strings.

4.3.2 Longest Common Subsequence

Longest common subsequence (LCS), as the name indicates, is a method of finding the longest subsequences of elements common to two strings. Let the two strings be defined as follows: $u = \{u_1, u_2, \dots, u_m, \dots, u_M\}$ and $v = \{v_1, v_2, \dots, v_n, \dots, v_N\}$. The longest word w is the LCS of u and v if w is a subsequence of both u and v . For finding the LCS between two strings, we use the LCS matrix \mathbf{D} which is of size $(M + 1) \times (N + 1)$. For our purpose, we are only interested in the length of the word w , since it gives the cost of transforming one string to another. Unlike in the case of ED, the higher the LCS value, the lower the cost. The elements of the LCS matrix \mathbf{D} can be defined by

$$\mathbf{D}(m, n) = \begin{cases} 0 & \text{if } m = 0 \text{ or } n = 0 \\ \mathbf{D}(m - 1, n - 1) + 1 & \text{if } u_m = v_n \\ \max \begin{cases} \mathbf{D}(m - 1, n) \\ \mathbf{D}(m, n - 1) \end{cases} & \text{else} \end{cases} \quad 1 \leq m \leq M; 1 \leq n \leq N. \quad (4.6)$$

An example for LCS is given in the Table 4.2. The LCS matrix \mathbf{D} between the words music and mucik is given in the table. As we can infer from the table, the word w can either be mui or muc. However, as stated before, we are only concerned with the length of the word w which in this case is 3. This length can be obtained from the cell $\mathbf{D}(M, N)$ (note that the index starts with 0). As discussed in the beginning of this chapter, for a query Q^t and database document D^t , the LCS operation can be defined as $LCS(Q^t, D^t)$ and its cost is assigned to

$$c^t(Q, D) := 1 - \frac{LCS(Q^t, D^t)}{|Q^t|} \in [0, 1], \quad (4.7)$$

4. MATCHING PROCEDURES

		n	0	1	2	3	4	5
m	u	v		m	u	s	i	c
	0			0	0	0	0	0
1	m		0	1	1	1	1	1
2	u		0	1	2	2	2	2
3	c		0	1	2	2	2	3
4	i		0	1	2	2	3	3
5	k		0	1	2	2	3	3

Table 4.2: LCS matrix \mathbf{D} between the strings `music` and `mucik`, in which LCS value is highlighted in red

where $|Q^t|$ denotes the length of the string Q^t . For LCS operation, since the length of the longest word can never be greater than the query, normalization by the length of the query ensures that its value lies in the interval $[0,1]$. To make this value comparable with the edit distance's cost, we subtract this value from 1.

4.4 Score-based Matching

Here we deal with the matching procedures that only consider the score representation of the query $Q \in \mathcal{Q}$ denoted by Q^s . Similarly, let D^s denote the score information for a document $D \in \mathcal{D}$. Both Q^s as well as D^s are sequences of numbers obtained from the MIDI pitches of the query and the database respectively. The technique to convert the MIDI pitches to the sequence of numbers is explained in Section 4.4.1. In a first step, we convert the OMR results into a piano-roll like representation as shown in Figure 4.2. For the mathematical computations, we use the MIDI number instead of musical pitch notation. The correspondence between the MIDI numbers and pitches are given in Appendix A. Using the pitch information directly from the

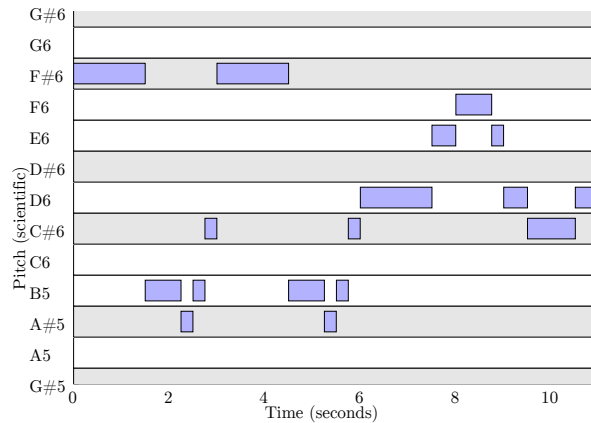


Figure 4.2: Piano-roll like representation

OMR results lowers the retrieval quality since the OMR results are very poor. As explained in

Section 3.2.3, in order to mitigate the artifacts caused from the OMR procedure, we have to come up with a modified pitch representation procedure as shown in Figures 1.1d, 1.1e and 1.1f. This is explained in the next section.

4.4.1 Differential Pitch

As explained in Chapter 1, we convert the OMR result into a piano-roll like representation as indicated by Figure 1.1d. Dealing with monophonic themes (a property that may be corrupted by the OMR step), we consider the upper pitch contour of the OMR result. Since OMR often fails at detecting the correct note durations but tends to correctly recognize the bar lines, we do not use the note durations but locally resample the pitch sequence to match the bar line constraints, see Figure 1.1e. This results in a sequence of pitch values. Furthermore, since OMR often misinterprets the global clef, we convert the pitch sequence into a sequence of intervals (differences of subsequent pitches), see Figure 1.1f. The interval sequence, denoted by Q^s , is used for the matching step. This interval sequence is nothing but the differential pitch of the resampled pitches. Similarly, we process a document $D \in \mathcal{D}$ for the sake of comparison. The resulting interval sequence is denoted by D^s . The whole process of conversion of MIDI pitches is explained with the example given in Table 4.3.

(a) Piano-roll like representation													
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	69	69	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	67	67	67	67	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	65	65	0	0	0	0
64	64	64	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	61	61	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
(b) Resampled Pitch Sequence													
64	64	64	67	67	67	67	69	69	61	61	61	61	61
(c) Interval Sequence													
-	0	0	3	0	0	0	2	0	-8	0	0	0	0

Table 4.3: Conversion of MIDI pitches to differential pitch. **(a)** A sample piano-roll representation. The non-zero values indicate midi pitch numbers at a particular time instance. **(b)** Resampled pitch information obtained from (a). **(c)** Interval sequence (differential pitch) obtained from (b).

The resampling process, which results in the resampled pitch sequence given in Table 4.3b, is as follows. When two or more notes start at the same time instance, then the resampling process converts it to monophonic tone by taking the highest pitch. This can be seen from the Table 4.3a, where only the note number 69 is considered when both the pitches 69 and 65 are present. However, if the tones starts at a different time instant, then the first occurring tone is taken until it is finished. This can be seen from the notes 64 and 67 from the table. Silence is filled with the previous pitch information. This is the reason why pitch 61 continues till the end. But taking the absolute pitch also weakens the comparison results. The idea of differential pitch (interval)

4. MATCHING PROCEDURES

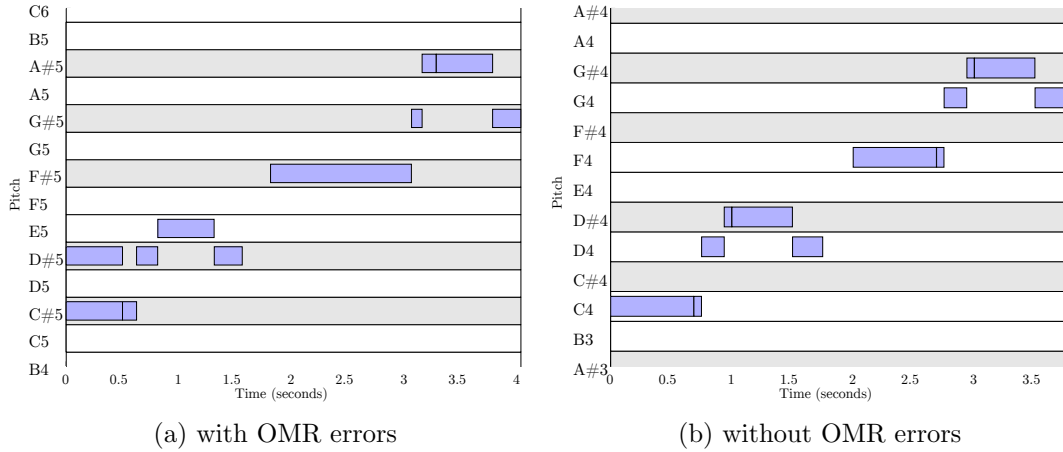


Figure 4.3: Piano-roll representations of the same theme with and without OMR errors

sequence was conceived to overcome transposition errors introduced by OMR engines.

For resampled pitch sequences $p_1, p_2, \dots, p_n, \dots, p_N$, the differential pitch (interval) sequence DP is obtained as follows.

$$DP(n) = p_{n+1} - p_n, 1 \leq n \leq N - 1 \quad (4.8)$$

Figure 4.3 motivates the usage of differential pitch. Figures 4.3a and 4.3b look similar, however when we examine the time and pitch information, they differ. As we can see, Figure 4.3b starts with the note $C4$ and it is monophonic in nature. Figure 4.3a is apparently polyphonic in nature since it starts with the notes $C\#5$ and $D\#5$. Not only the individual note durations, but also the total time durations of the two figures are different. These issues tell us about the need for the representation given in Figure 4.4.

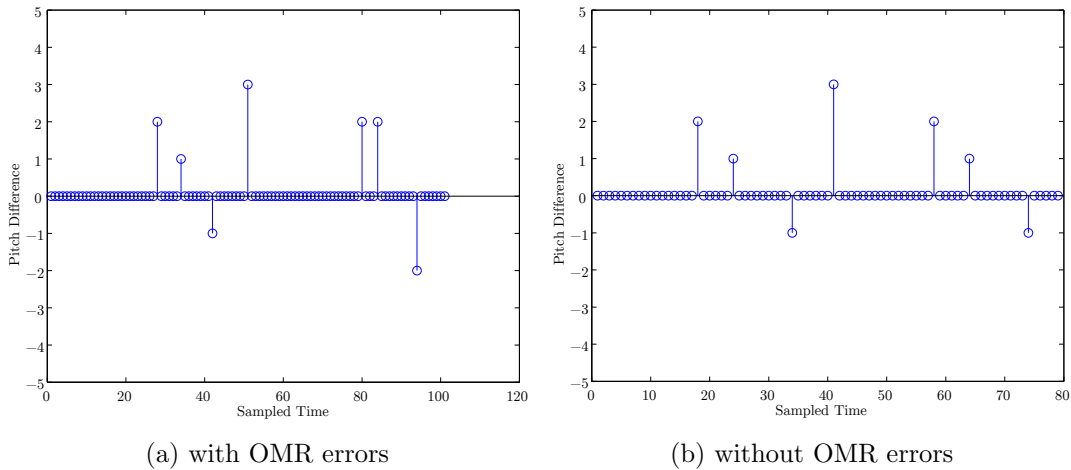


Figure 4.4: Differential pitch sequences with and without OMR errors

Figure 4.4 is the differential pitch sequence for the example given in Figure 4.3. Equation 4.8 is applied to the result of the resampling process and is shown in this figure. This sequence

is also mentioned as interval sequence in this thesis. Despite the fact that the pitch numbers are different, the relative pitches are the same. This phenomenon is exploited with the concept of differential pitch. The resampling process helps us to cope with the mismatches in the note durations. As a result, this sequence improves our information retrieval results. The experimental results are given in Section 5.2.

4.4.2 Dynamic Time Warping

Dynamic Time Warping (DTW) is a well known algorithm to retrieve the optimal alignment between two temporal sequences which may vary in time with certain restrictions [15]. The two sequences are warped in the time dimension in a non-linear fashion such that it is independent of certain non-linear variations. This dynamic programming algorithm is widely used in the field of speech recognition [20]. Let the two sequence be defined as follows: $u = \{u_1, u_2, \dots, u_m, \dots, u_M\}$ and $v = \{v_1, v_2, \dots, v_n, \dots, v_N\}$. From the musical perspective, we assumed that if the pitch deviates more than 12 semitones, then it is an OMR error. Under this assumption the cost matrix $C \in \mathbb{R}^{M \times N}$ is defined as

$$C(m, n) = \frac{\min\{|u_m - v_n|, 12\}}{12} \in [0, 1], 1 \leq m \leq M; 1 \leq n \leq N \quad (4.9)$$

We locally normalize by 12 to make sure that all the elements of the cost matrix are bounded by $[0,1]$. The *DTW distance* between u and v is defined as $DTW(u, v) = \mathbf{D}(M, N)$, where \mathbf{D} is the accumulated cost matrix. The accumulated cost matrix $\mathbf{D} \in \mathbb{R}^{M \times N}$, is defined as follows.

$$\mathbf{D}(m, 1) = \sum_{i=1}^m C(i, 1), 1 \leq m \leq M \quad (4.10a)$$

$$\mathbf{D}(1, n) = \sum_{i=1}^n C(1, i), 1 \leq n \leq N \quad (4.10b)$$

$$\mathbf{D}(m, n) = C(m, n) + \min \begin{cases} \mathbf{D}(m-1, n) \\ \mathbf{D}(m, n-1) \\ \mathbf{D}(m-1, n-1) \end{cases} \quad 1 < m \leq M; 1 < n \leq N. \quad (4.10c)$$

To understand the above mentioned concept better, let us consider the following example. Let $u = \{1, 12, 2, 3\}$ and $v = \{1, 1, 17, 2, 2, 3, 4\}$. Using Equation 4.9, the cost matrix C is shown below.

$$C = \frac{1}{12} \begin{bmatrix} 0 & 0 & 12 & 1 & 1 & 2 & 3 \\ 11 & 11 & 5 & 10 & 10 & 9 & 8 \\ 1 & 1 & 12 & 0 & 0 & 1 & 2 \\ 2 & 2 & 12 & 1 & 1 & 0 & 1 \end{bmatrix}$$

Using Equation 4.10, the accumulated cost matrix \mathbf{D} is as follows:

$$\mathbf{D} = \frac{1}{12} \begin{bmatrix} 0 & 0 & 12 & 13 & 14 & 16 & 19 \\ 11 & 12 & 5 & 15 & 23 & 23 & 24 \\ 12 & 12 & 17 & 5 & 5 & 6 & 8 \\ 14 & 14 & 24 & 6 & 6 & 5 & \mathbf{6} \end{bmatrix}.$$

The DTW distance is highlighted in red color in the \mathbf{D} matrix. We can use this matrix for finding the optimal warping path to align both sequences. However, since our interest is not in alignment but in a similarity measure, we do not compute the warping path. In order to compare the cost (similarity measure) of different queries with the database, we introduce a global normalization where we normalize the DTW distance by the length of the query. This cost is assigned as follows:

$$c^s(Q, D) := \frac{\text{DTW}(Q^s, D^s)}{|Q^s|}. \quad (4.11)$$

where $|Q^s|$ denotes the length of the query sequence Q^s . As in the case of the edit distance, we cannot guarantee that the upper bound of cost c^s is one. Very small queries (usually OMR errors) will result in costs larger than one.

4.4.3 Parameter Settings

Various parameter settings used in the computation of score-based matching are explained in this section. The experimental results using some of the important parameters are explained in the following chapter. The remaining experimental results using the parameters explained here are given in Appendix B. All the relevant parameters are tested on a small dataset, *BM-Small*, consisting of 159 selected queries (see Appendix B).

4.4.3.1 Resolution

Resolution of the MIDI pitch plays an important role in the MIR scenario. Depending on the resolution of the MIDI pitches, the step size in milliseconds changes. This parameter should be selected carefully, since it results in a trade off between computational time and retrieval quality. After doing extensive trial and error experiments on the *BM-Small* dataset (cf. Appendix B) and considering the trade offs, we found that the best step size for the files in the database is 30 ms. Step size in millisecond can be obtained using the formula

$$\text{step size} = \frac{1000}{\text{Resolution}}. \quad (4.12)$$

4.4.3.2 Time

Time signatures are often misinterpreted by the OMR engines. This may result in wrong note durations. This leads us to the use of two different settings for our experiments: `With time` and `Without time`.

`With time`: The pitch sequences are sampled with the specified **Resolution** parameter.

`Without time`: The time duration of the notes are not considered. The pitch sequence for this parameter is the sequence of MIDI events in the order of their appearance in the MIDI file.

Start Time(ms)	Duration(ms)	MIDI number
0	249	62
500	249	65
1000	249	67
1500	249	70
2000	749	70
2750	249	67
3000	499	65
3500	249	62
4000	249	65

Table 4.4: Sample note events

Consider the MIDI events given in Table 4.4. Let the resolution used for sampling be 250 ms. Then the corresponding `With time` pitch sequence is $\{62, 0, 65, 0, 67, 0, 70, 0, 70, 70, 70, 67, 65, 65, 62, 0, 65\}$. The 0 value in this pitch sequence represents silence. The corresponding `Without time` sequence will be $\{62, 65, 67, 70, 70, 67, 65, 62, 65\}$. This sequence is used to compare the MIDI events directly without considering their duration. Apparently, the silences are not considered in this scenario.

4.4.3.3 Normalization

As mentioned in the Equation 4.11, a global normalization is applied to the DTW distance of all the queries. DTW distances can be normalized by the length of the shorter time series, by the length of the optimal warping path, by the length of the longer time series etc. We evaluated two methods of normalization for our retrieval scenario.

N_{query} : When this parameter is selected, the normalization is done as in Equation 4.11 where the DTW distance is normalized by the length of the query, irrespective of the fact that it could be longer or shorter than the database element.

N_{WPP} : When this parameter is selected, the normalization in the Equation 4.11 is replaced by the length of the optimal warping path. For the sake of brevity, the computation of the optimal warping path is not shown in this thesis. We use the equations from [15, Chapter 4] to compute the warping path. The major disadvantage of this normalization method is its computational complexity. We need the complete accumulated cost matrix \mathbf{D} for finding the optimal warping path. Depending on the size of \mathbf{D} , the computational time may change.

4.4.3.4 Constant Cost

Since the input sequences to the DTW algorithm (differential pitch or interval sequence) contain a lot of zeros, a constant cost \mathcal{C} is added to the cost matrix C given in Equation 4.9. This parameter was designed to make sure that the warping path takes the best diagonal route since the majority of the elements in the cost matrix are 0. This can be represented as $C(m, n) = C(m, n) + \mathcal{C}$, where $\mathcal{C} \in \mathbb{Z}_{\geq 0}$.

4.4.3.5 DTW Step Size

Various step sizes for the computation of the DTW distance are $\Sigma_1 = \{(1, 0), (0, 1), (1, 1)\}$, $\Sigma_2 = \{(2, 1), (1, 2), (1, 1)\}$ and $\Sigma_3 = \{(1, 1)\}$. Experiments are done with these step sizes to study their influences on the retrieval quality.

4.4.3.6 Fixed Length MIDI Sequence

This parameter was used for studying the retrieval quality when both the query and the database files were of equal length. `Without time` parameter cannot be used with this parameter. The step size $\{(1,1)\}$ will work properly only when both the MIDI sequences are of equal length. For the purpose of this thesis, we have set the fixed length of MIDI files to 100 which means 100 equidistant points of the MIDI files are taken to obtain this representation. This parameter reduces the computational time as it deals with matrices of size 100×100 .

4.4.3.7 Threshold

This parameter is used to control the local cost used for the calculation of the DTW distance. By default, as shown in Equation 4.9, it makes sure that the local cost is bounded by $[0,1]$. If thresholding is disabled, then the equation will change to

$$C(m, n) = |u_m - v_n| \in [0, 1], 1 \leq m \leq M; 1 \leq n \leq N. \quad (4.13)$$

In other words, this parameter is used to enable or disable the local normalization used in DTW.

4.5 Summary

The database needed for the cross-modal comparison was introduced in this chapter. Various evaluation methods used in this thesis were also introduced here. Edit distance (ED) and longest common subsequence (LCS) methods are used for text-based matching and dynamic time warping (DTW) method for the score based matching. Various parameter settings used in conjunction with these methods are also explained in this chapter. LCS is more preferred for our retrieval scenario compared to ED because of the extra information which may be present in the input images. Extra information includes copyright and permission information, which elevates the cost of ED. Even at the presence of these extra information, LCS remains the same. Experimental results explained in Chapter 5 will prove the fact that this surmise is correct.

Chapter 5

Retrieval Experiments

We now evaluate the proposed matching procedures within a retrieval setting. At this point, we consider the set \mathcal{D} of EDM themes as a database collection of unknown musical themes. Using a BM theme $Q \in \mathcal{Q}$ as query, the task is to identify the database document that musically corresponds to the query. Note that in this retrieval scenario, there is exactly one relevant document for each query (see Section 4.2.1). For the purpose of evaluation, we use the methods explained in Section 4.2.2. In our evaluation, we compare the query Q with each of the documents $D \in \mathcal{D}$. Distance matrix \mathbb{D} , explained in Section 4.2.3, is used for the computation of individual ranks of the query and Top N match (T_N).

In the case that T_N contains the relevant document, we say that the retrieval process has been *successful*. We perform the retrieval process for all the 9803 queries $Q \in \mathcal{Q}$. In a Google-like retrieval scenario, a user typically first looks at the top match and then may also check the first five, ten or twenty matches at most. Therefore, in the following sections, we consider the values $N \in \{1, 5, 10, 20\}$. If the rank of a particular query is either 30 or 3000, it does not have a significant impact as the user is not interested in this result. Considering the above statement, we introduce another evaluation measure called *capped mean rank* \bar{R}_c . For the computation of \bar{R}_c , the individual rank of query above 20 is assigned to 21. Then the mean rank is found using the modified ranks of the individual queries.

The computation of T_N and \bar{R} are explained using the example given in Table 5.1. The table shows the costs between two queries and five database elements. Color coding shows the correspondence

	Q_1	Q_2
D_1	0.8	0.7
D_2	0.12	0.45
D_3	0.78	0
D_4	0.65	0.4
D_5	1	0.61

Table 5.1: Example to explain T_N and \bar{R} . D_4 and D_3 represent the database element of queries Q_1 and Q_2 respectively.

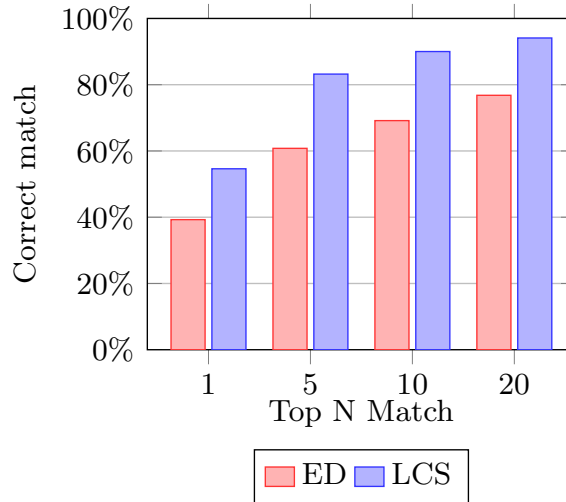


Figure 5.1: Top N match for the text-based retrieval. 9803 queries were used in this experiment. $\bar{R}_{ED} = 26.12$, $\bar{R}_{LCS} = 7.04$.

between the query and its ground truth (desired document). We need to sort the corresponding column in ascending order for a particular query to get its top match. For query Q_1 , the top match list contains D_2 , D_4 , D_3 , D_1 and D_5 . The query Q_2 contains D_3 , D_4 , D_2 , D_5 and D_1 . From this list, only the first N elements are only considered for finding the Top N match. The rank for query Q_1 is $2/5$ and that for Q_2 is $1/5$. So the mean rank \bar{R} can be computed from these individual ranks. All the computations on the following sections are done in a similar way. In the following sections, the experimental results using four different matching procedures are explained.

5.1 Text-based Matching Results

Let us start with a discussion of the text-based matching result. The details about text-based matching is detailed in section 4.3. Figure 5.1 explains the results of different text based matching techniques which are explained in the Section 4.3. Knowledge about **BM-Composer** was utilized in the text-based matching. Matching was performed after filtering out all the $D \in \mathcal{D}$ which correspond to a composer other than **BM-Composer**.

Mean rank for ED (\bar{R}_{ED}) and LCS (\bar{R}_{LCS}) are 26.12 and 7.04 respectively. Both methods worked well when we compared the \bar{R} with the size of the database which is 9803. It is evident from the figure that LCS outperformed ED method. \bar{R} of ED is 26.12 whereas the same for LCS is 7.04. As explained before, the permissions and copyright information hindered the performance of ED technique.

Similar tendency can be seen in the case of Top N Match. T_1 for ED was 39.3% whereas the same for LCS was 54.6%. This means that for more than half of the queries of BM-Themes, LCS methods retrieves the correct document. Considering the top five matches (T_5) of LCS, the percentage of successful cases increases to 83.2%. This improvement can be explained by the fact that the specifications of the musical themes from the same work often differ in only a few

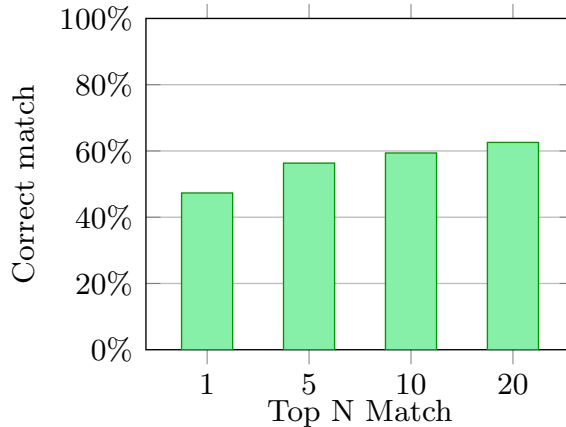


Figure 5.2: Top N match for the score-based retrieval. 9803 queries were used in this experiment. $\bar{R}_{DTW} = 1135.33$.

characters, e.g. “1st Movement, 1st Theme, A” versus “2nd Movement, 1st Theme, B”. Such small differences may lead to confusion among the top matches in the presence of OCR errors. Similar trend can be seen in the case of ED where the T_5 boosted to 60.8%. Considering T_{20} for LCS, one obtains 94.1%, which indicates that the text-based retrieval alone already yields a good overall quality. T_{20} for ED is 76.8%.

The clipped mean rank, \bar{R}_c , for ED is 7.68 and LCS is 3.64. This means that if we are taking the first 21 documents, on average we will find the desired document at the 3rd or 4th position in case of LCS and 7th or 8th position in case of ED.

5.2 Score-based Matching Results

Next, let us have a look at the results of score-based matching explained in Section 4.4. We were not able to process 1794 score-based queries (Q^s) as explained in Section 4.4.1. The details of these queries are specified in the Appendix D. The main reason for this failure is the poor quality of the input images. To compensate the effect of these unavailable files, the individual rank for these queries are assigned to the half of the size of database. It means that if the OMR engine failed to process a query, then its rank is assigned to 4901 ($\lfloor \frac{9803}{2} \rfloor$). The OMR corresponding to all the other queries are converted to interval sequences.

Figure 5.2 represents the retrieval results of the score-based representation. In contrast to the text-based retrieval results, we can see that there is a huge drop in percentage of top matches. One of the main reason for this behavior is that approximately 18% of the queries were not available, as explained before. High fluctuation in the mean rank is also because of the same reason.

In the case for $N = 1$ for T_N , the score-based retrieval has been successful for 47.3% of the total queries. This is an interesting results since for around 4700 files, the retrieval quality was perfect. When we increase the value of N from 1 to 20, we can see that there is a rise of information retrieval quality from 47.3% to 62.6%. Different parameters explained in Section

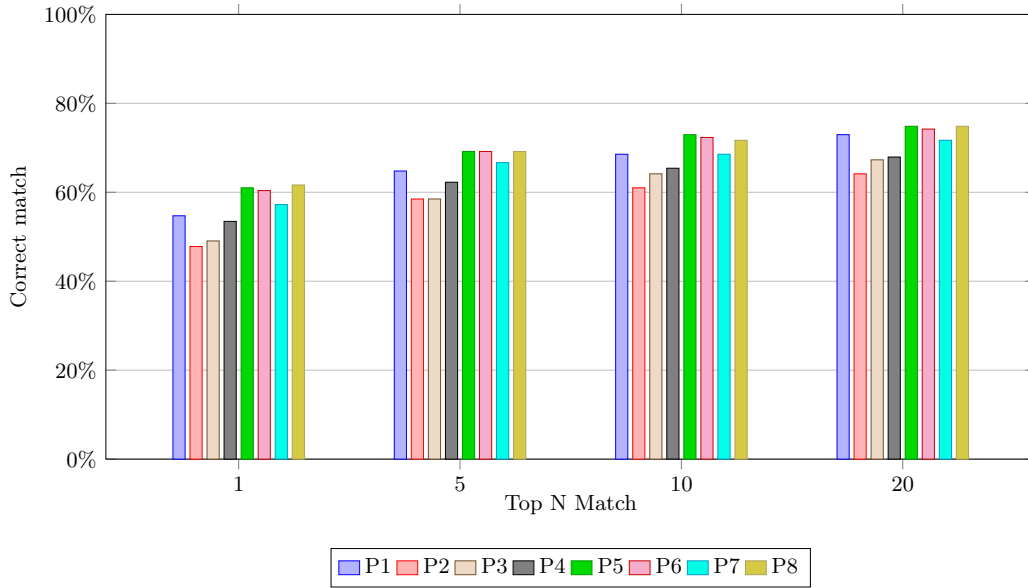


Figure 5.3: Effects of different parameter settings on BM-Small subset containing 159 elements. P1 to P8 are the combination of different parameter settings which is explained in Table 5.2.

4.4.3 were used for the score based retrieval. For the retrieval result explained in the Figure 5.2, following parameters were used. The resolution with which the MIDI was converted to piano-roll representation is 33 Hz. It means that every entry of the piano roll representation corresponds to 30.3 ms of the MIDI file. With `time` parameter was enabled for this IR experiment. DTW distance in this experiment is normalized by the length of the query (N_{query}) and not by the length of the warping path. Constant cost \mathcal{C} , was disabled in this experiment. The DTW step sizes used in this experiments were $\{(1,0),(0,1),(1,1)\}$. Fixed length MIDI sequences were not used in this experiment. Thresholding was enabled in this experiment.

Many of the parameters explained in the Section 4.4.3 did not elevate the retrieval quality considerably. The details of these parameters are given in Section 5.2.1. The reason for this behavior directs our attention to the ability of the OMR engine to handle low quality images. The inability of the OMR engines to process around 18% of the total queries also exposes its inefficiency to process these kind of images. However in the future, with the use of suitable OMR engines, these problems can be mitigated.

5.2.1 Effects of Various Parameters

Here we explain the reasons for using the particular parameter settings used for the retrieval experiment described before. Effects of various parameters are explained using BM-Small subset which contains 159 queries. For the sake of brevity, effects of only important combinations of the parameters are shown here. Less important combinations of the parameters are included in Appendix B.

As explained before, 159 elements were used to tune the parameters for our retrieval experiments. The combination of different parameters are explained in Table 5.2. P1 to P8 represent different

Parameters	P1	P2	P3	P4	P5	P6	P7	P8
Resolution (Hz)	-	-	-	-	20	33	50	33
Time (W/ WO)	W	W	W	WO	W	W	W	W
Normalization (Q/WP)	WP	WP	WP	WP	Q	Q	Q	Q
Constant Cost (Y/N)	Y	N	N	Y	N	N	N	N
Fixed Length (Y/N)	Y	Y	Y	N	N	N	N	N
Threshold (Y/N)	N	N	Y	Y	N	N	N	Y
Mean Rank (159 queries)	21.27	26.70	26.45	23.20	20.11	19.69	22.36	19.52
Computation Time (min)	0.73	0.85	0.87	0.27	0.27	0.51	1.24	0.61

Table 5.2: Different parameter settings used for finding the best retrieval quality. W: With time; WO: Without time; Q: Normalized by the length of the query; WP: Normalized by the length of the warping path; Y: Enabled; N : Disabled; DTW step sizes of $\{(1,0),(0,1),(1,1)\}$ were used for all these parameter settings.

parameter settings. Resolution parameter was disabled for P1 to P4 because either the fixed length parameter was enabled or it was a `Without time(WO)` sequence. Normalization by the length of the query (N_{query}) is denoted by Q in the table and the same by the length of the warping path (N_{WP}) is denoted by WP. If a parameter is active or enabled, it is denoted by Y else it is denoted by N. In our experiments, constant cost parameter (\mathcal{C}) is set to 1, if it is enabled. If fixed length sequence is enabled, the MIDI length is set to 100 samples which means that 100 equidistant points of the MIDI files are sampled to get this sequence. If threshold parameter is enabled, the local cost for the calculation of DTW distances are thresholded using Equation 4.9.

Mean rank (\bar{R}) of the retrieval experiment and the computation time required to process these 159 queries are also mentioned in the corresponding columns of the Table 5.2. Out of these 159 queries, OMR output files (MIDI format) are not available for 33 queries. So the ranks of these queries are assigned to 79 without any processing. As we can see from \bar{R} and the Figure B.1, P8 gives the best result among the different parameter settings. \bar{R} for this setting is 19.52, where as the T_1 and T_{20} for the same is 61.6% and 74.8%. P5 also works better since it gives \bar{R} closer to the same of P8. Besides, the computational time is the least for this setting since it is handling only a resolution of 20 Hz. As it is evident from Table 5.2, P5 and P6 only differs from the resolution parameter whereas P6 and P8 differs only from thresholding parameter. All these three settings have its trade offs between retrieval quality and computation time. N_{query} is preferred over N_{WP} since extra time is required to compute the warping path. As we can see from the results, it increases the computation time even though it was combined with fixed length parameter. Fixed length samples requires only very less computation since it deals with the accumulated matrix of size 100×100 only, where as the other settings are dealing with higher sizes depending on the resolution. Constant cost (\mathcal{C}) and `Without time` parameter does not play a huge role in the retrieval quality. This can only be observed along with the parameter settings given in Appendix B. All settings mentioned here used DTW step sizes $\Sigma_1 = \{(1, 0), (0, 1), (1, 1)\}$. Using the other step sizes like $\Sigma_2 = \{(2, 1), (1, 2), (1, 1)\}$ and $\Sigma_3 = \{(1, 1)\}$ rapidly decreased the retrieval quality. The reason for this behavior is because of the nature of interval sequences (see Section 4.4.1).

As explained before, our experiments with the complete dataset used P8 parameter setting since we focus on the retrieval quality. On the other hand, computation time is acceptable for this setting when compared to P5 and P6.

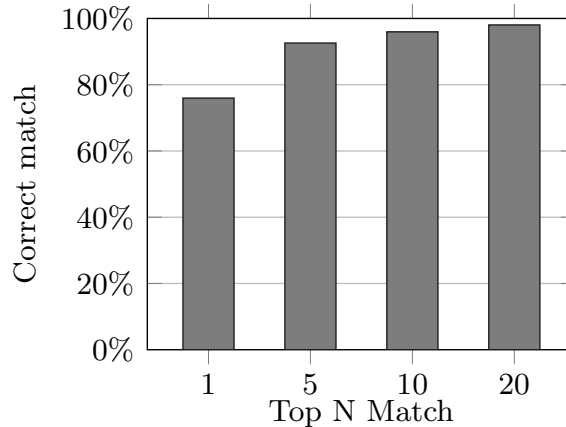


Figure 5.4: Top N match for the Oracle Fusion technique. 9803 queries were used in this experiment. $\bar{R}_{OracleFusion} = 3.13$.

5.3 Oracle Fusion

After analyzing individual results of the score-based and the text-based matching, we have found out that for many queries, OCR performed flawlessly where as the OMR results were not satisfactory. On the other hand, there were many cases for which the OMR worked better than the OCR. This is evident from the Figures 5.1 and 5.2. These findings lead us to the inception of the technique “Oracle Fusion”, where the help of an oracle is used for the retrieval process. Oracle uses ground truth information and inform us to select either text-based or score-based matching list. Oracle tells us to use the list where the required document (ground truth) for a particular query comes first. By doing so, we can fuse the perfect results of the text-based and score-based retrieval leaving out all the outliers and failures.

The results obtained from this oracle fusion procedure yield a kind of upper limit for the joint performance of the text-based and score-based matching procedures. These results are shown using the Figure 5.4. T_1 , T_5 , T_{10} and T_{20} for the oracle fusion are 76%, 92.6%, 96% and 98% respectively. Mean rank(\bar{R}) for this technique is 3.13 where as the capped mean rank (\bar{R}_C) is 2.16. The oracle fusion process is better understood with an example given in Table 5.3. Let the Q^s and Q^t be the score-based and the text-based retrieval results for a particular $Q \in \mathcal{Q}$. Let

Rank	Q^s	Q^t
1	D_3	D_6
2	D_2	D_5
3	D_5	D_4
4	D_6	D_3
5	D_1	D_2
6	D_4	D_1

Table 5.3: Example to explain the oracle fusion retrieval.

the database document highlighted in red color be the ground truth for the query Q . Then, the

oracle fusion uses its knowledge about ground truth to check its position in both the ranking lists. In this case, the ground truth document D_2 is having second rank in score-based matching and fifth rank in text-based matching. As a result, oracle informs the retrieval program to use the score-based matching results. If the rank in both the retrieval lists are the same, then the oracle will ask the retrieval program to use the text-based matching results since its mean rank is far better than that of the score-based result.

Thus, oracle fusion fuses the best of the text-based or score-based results for the retrieval scenario. The results given in Figure 5.4 inform us about the need of a method to combine the best of the individual retrieval results to transcend the IR quality without any oracle knowledge. One such method is explained in the following section.

5.4 Fusion

In this section, we present a method that fuses the individual text-based and score-based retrieval results without any oracle knowledge. Using this method, the text-based matching result is refined using the score-based information. Let $L^t = [L_1^t, L_2^t, \dots, L_i^t, \dots, L_Y^t]$ be the ranking list obtained from the text-based retrieval and Y corresponds to the total number of EDM themes. Here, the element L_1^t corresponds to the top best match, L_i^t to the best i th match and L_Y^t to the worst match of the text-based retrieval process. Similarly, let $L^s = [L_1^s, L_2^s, \dots, L_i^s, \dots, L_Y^s]$ be the ranking list obtained from the score-based retrieval where the elements L_1^s , L_i^s and L_Y^s correspond to the top best match, best i th match and the worst match respectively.

In this fusion method, we combine the lists L^t and L^s to obtain a new retrieval list L^{FUS} . Instead of using the ground truth information, we use certain heuristics to find out the best retrieval list for a particular query. Here, the text-based results are refined with the score-based results. This is mainly due to the fact that text-based information are filtered composer wise as explained in the Section 4.3. Besides, the OMR failed for more than 20% of the queries. So refining text-based results is apparently better than doing the same with score-based results. We also use the distance matrix \mathbb{D} for fusion since it contains the distances between the query and all the elements of the EDM database. As explained in the Equation 4.3, the elements of distance matrix originate from the cost of different alignment techniques used for text-based or score-based matching. Only the costs of score-based matching are exploited for the purpose of fusion since the costs of the top matches from text-based matching are very close to each other (they differs only by few characters). If the OMR engine can detect the musical themes without significant problems, then their costs will be lesser when aligned to the themes of the EDM database. But if the OMR outputs a lot of errors, then their respective costs will be higher. If the differences between the cost of first and second element of the score-based list is larger than a threshold τ , then we can rely on the top match of this list. Let δ denotes this difference and is described as $\delta = L_2^s - L_1^s$.

Trial and error methods showed that 0.05 was the best value for τ . Figure 5.5 also helps us to find the value of τ experimentally. δ value of all the queries are shown in this figure. The correct and the incorrect matches are also specified here when a particular δ is used as τ . As shown in the figure with dashed line, the τ obtained with trial and error method is a good trade off between the total number of correct and incorrect entries. So this value is used for fusion as τ .

5. RETRIEVAL EXPERIMENTS

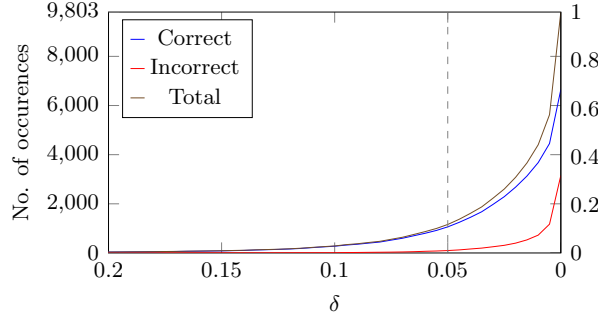


Figure 5.5: The plot shows the number of correct and incorrect matches for a $\delta \in [0, 0.2]$.

The fusion process to obtain the fusion list, L^{FUS} , can be described using the equation

$$L^{\text{FUS}} = \begin{cases} L^t & \text{if } L_1^t = L_1^s, \\ [L_1^s, \tilde{L}^t] & \text{elseif } \delta > \tau, \\ [L_1^s, \tilde{L}^t] & \text{elseif } L_1^s \in L_{1\dots N}^t, \\ L^t & \text{else,} \end{cases} \quad (5.1)$$

where $\tilde{L}^t = \{L^t \mid L_i^t \neq L_1^s\}$. Here $L_{1\dots N}^t$ represents first N entries of the text based matching list. In our retrieval experiments, the value of $N \in \mathbb{N}$ is assigned to 20. \tilde{L}^t represents the text based retrieval list without the top best match of the score-based matching contained in it.

The Equation 5.1 can be explained as follows. If T_1 match of the lists L^t and L^s are the same, then the L^t is used as the fusion list (L^{FUS}). In such a case, T_1 match is highly reliable since it is the same for two independent experiments done on different modality. If T_1 differs in both the lists, then we check whether cost δ is greater than the threshold τ . If it is greater, then we will concatenate L_1^s and \tilde{L}^t . If it is lesser, then we will check whether the L_1^s is contained in the T_N entries of the text based matching list. This is mainly because of the observation that the entries of T_N differ only by very few characters. If none of the above mentioned cases work, then L^t is used as the fusion list. This is an important condition since around 20% of the OMR results are not available.

Now, we discuss the results of the fusion experiment which uses the above mentioned condition. As we can see in the Figure 5.6, the mean rank of this technique is 6.24. The capped mean rank (\bar{R}_c) in this technique is 2.96 that substantiates the success of this method that is close to the upper bound. T_1 and T_{20} for this method is 69.6% and 94.3% and this is also close to the upper bound given by Oracle Fusion. Many of the queries were having bad OMR as well as OCR results. So excluding those queries, we can see that for almost all the queries, the correct document (ground truth) will be on an average in the second or third position, which is also the case for Oracle Fusion.

Now, let us discuss about influence of the four cases explained in the Equation 5.1. The first condition deals with the cases where both the text-based and score-based retrieval have the same T_1 match. 2229 of such cases occurred and out of them 2206 were matching with the ground truth. 23 queries failed because for many queries, the ground truth information was not available or wrongly assigned. Besides, some of these queries' text and score information were almost the

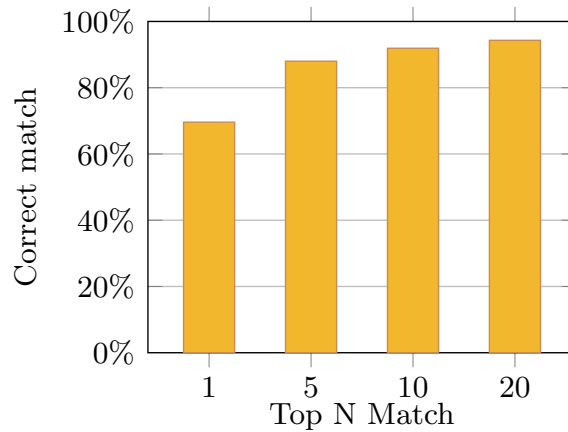


Figure 5.6: Top N match for the Fusion technique. 9803 queries were used in this experiment. $\bar{R}_{Fusion} = 6.24$.

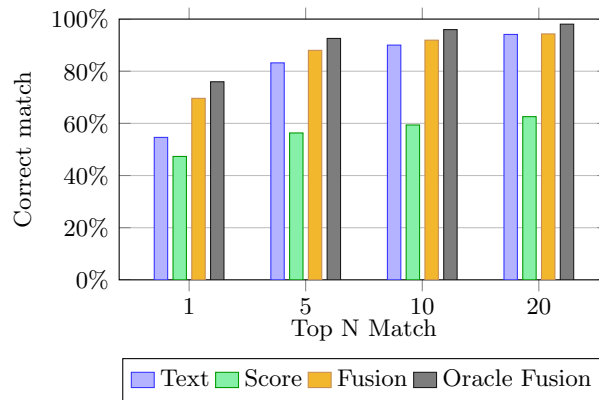


Figure 5.7: Comparison of the number of top K matches for the different procedures.

same. There were only around 71 cases that used the second condition and out of which 68 were correct. 1724 queries were satisfying the penultimate condition given in Equation 5.1 out of which 1644 were correct. 5779 queries did not satisfy the first three conditions and thus they used their respective text-based results.

As we can conclude from the results, fusion can be considered as a paragon technique which elevates the individual retrieval quality. This can be easily adapted to the other retrieval scenarios by tuning certain parameters.

5.5 Summary

Previous sections explained about various retrieval results that were conducted as a part of this thesis. To recapitulate, let us consider Figure 5.7, that compares the top matches of not only the text-based and score-based retrievals but also Fusion and Oracle Fusion retrievals.

As we can notice from the figure, the score-based retrieval list is used to improve the text-based

results. Oracle Fusion served like an upper bound where the best of the individual lists were analyzed and selected meticulously. Ground truth information served as the source of oracle in Oracle Fusion. However, Fusion technique modeled the absence of ground truth information as explained in the Section 5.4.

T_1 match of the text-based and the score-based results were only around 55% and 47% respectively. Using Fusion technique, the same was elevated to 70%. This shows that the quality of music information retrieval is improved substantially. When we analyze the results shown in the Figure 5.7, we will notice that T_{20} results were almost the same for the text-based retrieval and Fusion. This is because of the fact that the fusion list contains almost all the elements of the text-based list according to the Equation 5.1. Only the first element of the text-based list alters, if the score-based result is better (delete the particular entry from the OCR list and reinsert at the first position) and this is evident from the large increase of T_1 in the fusion results.

When we compare the results of Fusion and Oracle Fusion technique, we can see that the top matches are very close to each other. This means that Fusion technique was successful in emulating the Oracle Fusion technique. However, very small differences between the results pinpoint to the potential improvement to a mean rank of 1. Even though this can be achieved by tweaking the fusion program to adapt BM and EDM databases, our intent was only to point out the potential of fusing the matching results which can be easily adapted to any database.

Chapter 6

Applications and Conclusions

This thesis presented techniques for matching text-based and score-based musical information to improve the information retrieval quality. As a case study, sources from the musical dictionary by Barlow and Morgenstern were used as queries, whereas an online electronic dictionary, which represents online music libraries, were used as the database.

Going beyond the described (somehow controlled) scenario, we see the potential of music information retrieval techniques for a much wider range of application scenarios. As mentioned in the introduction, there are millions of digitized pages of sheet music publicly available on the world wide web. Furthermore, music related Wikipedia websites often contain information of various types including text, score, images, and audio as shown in Figure 6.1. Using similar techniques as described in this thesis, one can use such structured websites to automatically derive text-based and score-based queries (and queries of other types of information such as audio

The screenshot shows the Wikipedia article for "Symphony No. 5 (Beethoven)". The page layout includes a sidebar on the left with navigation links like "Main page", "Contents", and "Tools". The main content area features a title "Symphony No. 5 (Beethoven)", a search bar, and a paragraph of text. The text describes the symphony as one of the most frequently played symphonies, first performed in Vienna's Theater an der Wien in 1808. It mentions E. T. A. Hoffmann's description of the work as "one of the most important works of the time". Below the text is a musical notation snippet showing the opening motif: a four-note sequence (G4, A4, B4, C5) in C minor. To the right of the text is an image of the original coversheet of the symphony, which includes the title "SINFONIE" and the composer's name "LOUIS VAN BEETHOVEN". The coversheet also features a dedication to Prince J. F. M. Lobkowitz and Count Rasumovsky.

Figure 6.1

6. APPLICATIONS AND CONCLUSIONS

or video) to look for musically related documents on the world wide web. For example, using the work specification (Beethoven, Symphony No. 5) and the score excerpt from Figure 6.1, one may want to retrieve sheet music representations from IMSLP or resources from less structured websites.

One main contribution of this thesis was to show that matching procedures based on possibly corrupted score input (e. g., coming from OMR) may still be a valuable component, especially within a fusion scenario where the performance of an existing classifier is improved.

Fusion strategies that exploit multiple types of information sources play an important role to cope with the uncertainties and the inconsistencies in heterogeneous data collections, see [17]. In this context, audio-related information has been studied extensively, see, e. g., [19, 22, 11]. Future work deals with the integration of all available sources distributed on world wide web that describe a musical work in order to identify, retrieve, and annotate musical sources.

Appendix A

Midi Number and Scientific Note

Following table gives the relationship between MIDI number and scientific name of the pitches.

MIDI Number	Scientific Pitch	MIDI Number	Scientific Pitch	MIDI Number	Scientific Pitch	MIDI Number	Scientific Pitch
108	C8	86	D6	64	E4	42	F [#] 2/G ² 2
107	B7	85	C [#] 6/D ^b 6	63	D [#] 4/E ^b 4	41	F2
106	A [#] 7/B ^b 7	84	C6	62	D4	40	E2
105	A7	83	B5	61	C [#] 4/D ^b 4	39	D [#] 2/E ^b 2
104	G [#] 7/A ^b 7	82	A [#] 5/B ^b 5	60	C4	38	D2
103	G7	81	A5	59	B3	37	C [#] 2/D ^b 2
102	F [#] 7/G ^b 7	80	G [#] 5/A ^b 5	58	A [#] 3/B ^b 3	36	C2
101	F7	79	G5	57	A3	35	B1
100	E7	78	F [#] 5/G ^b 5	56	G [#] 3/A ^b 3	34	A [#] 1/B ^b 1
99	D [#] 7/E ^b 7	77	F5	55	G3	33	A1
98	D7	76	E5	54	F [#] 3/G ^b 3	32	G [#] 1/A ^b 1
97	C [#] 7/D ^b 7	75	D [#] 5/E ^b 5	53	F3	31	G1
96	C7	74	D5	52	E3	30	F [#] 1/G ^b 1
95	B6	73	C [#] 5/D ^b 5	51	D [#] 3/E ^b 3	29	F1
94	A [#] 6/B ^b 6	72	C5	50	D3	28	E1
93	A6	71	B4	49	C [#] 3/D ^b 3	27	D [#] 1/E ^b 1
92	G [#] 6/A ^b 6	70	A [#] 4/B ^b 4	48	C3	26	D1
91	G6	69	A4	47	B2	25	C [#] 1/D ^b 1
90	F [#] 6/G ^b 6	68	G [#] 4/A ^b 4	46	A [#] 2/B ^b 2	24	C1
89	F6	67	G4	45	A2	23	B0
88	E6	66	F [#] 4/G ^b 4	44	G [#] 2/A ^b 2	22	A [#] 0/B ^b 0
87	D [#] 6/E ^b 6	65	F4	43	G2	21	A0

Table A.1: Midi numbers and Scientific pitch

Appendix B

Subsets

The data that was used to for this thesis are mentioned here. All the subsets originated from either BM database or EDM database. BM subsets are chosen manually whereas the EDM subsets are chosen automatically after finding the match of each element from the BM-subsets. BM subsets contain filenames according to the following convention. ' $\langle BM-ThemeID \rangle_ \langle BM-ThemeOrigID \rangle$ '. 1066_B948 is an example for this type of naming convention. EDM subsets contain filenames in ' $Midi \langle Midi_number \rangle$ ' format. Midi1072 is a typical example for this naming convention. The details of the subsets are as follows.

Mini Subset

BM-Mini and EDM-Mini subsets constitutes the *Mini Subset*. BM-Mini, which contains 26 themes, is a subset of the BM database where as the EDM-Mini, which also contains the same number of themes, is a subset of EDM database. All the themes of this subset were chosen judiciously since it is a small representation of the entire BM database. Nuances of various parameters were visually examined using this subset. BM-Mini subset is given in table B.1 and EDM-Mini subset is given in table B.2.

0169_B83	0848_B730	1071_B953	2236_C249	7754_S535
0389_B301	1066_B948	1149_B1031	2276_C289	7940_S713
0807_B689	1067_B949	1511_B1375	2287_C300	
0808_B690	1068_B950	1512_B1376	2288_C301	
0846_B728	1069_B951	2219_C232	7752_S533	
0847_B729	1070_B952	2221_C234	7753_S534	

Table B.1: BM-Mini subset

Small Subset

BM-Small and EDM-Small subsets constitutes the *Small Subset*. BM-Small contains 159 elements whereas EDM-Small contains 153 midi files. This subset was used for tuning different parameters

B. SUBSETS

Midi1072	Midi1156	Midi6308	Midi853	Midi9291
Midi1073	Midi174	Midi813	Midi854	Midi9292
Midi1074	Midi394	Midi814	Midi9222	
Midi1075	Midi6120	Midi8508	Midi9224	
Midi1076	Midi6121	Midi8509	Midi9239	
Midi1077	Midi6122	Midi852	Midi9280	

Table B.2: EDM-Mini subset

such that it can be applied in the entire BM and EDM database. BM-Small subset is given in the table B.3 and EDM-Small subset is given in the table B.4.

0012_A12	1476_B1340	3440_G39	4889_K63	6115_M840	7606_S387
0013_A13	1551_B1415	3441_G40	4890_K64	6247_M972	8029_S802
0116_B30	1552_B1416	3442_G41	4891_K65	6248_M973	8098_S870a
0126_B40	1553_B1417	3443_G42	4892_K66	6419_P37	8099_S870b
0167_B81	1857_B1711j	3457_G56	4893_K67	6500_P114	8352_S1114
0179_B93	1858_B1711k	3458_G57	4894_K68	6820_R53	8353_S1115
0242_B156	1859_B1711l	3459_G58	4895_K69	6821_R54	8354_S1116
0257_B171	1860_B1711m	3460_G59	4896_K70	6892_R125	8355_S1117
0261_B175	2184_C198	3461_G60	5137_L189	6893_R126	8780_S1534a
0297_B209	2210_C223	3617_G206	5551_M289	7233_S14	8781_S1534b
0333_B245	2211_C224	3618_G207	5702_M429	7236_S17	8782_S1534c
0386_B298	2245_C258	3790_H9	5707_M434	7237_S18	8793_S1541
0387_B299	2598_D30	3794_H13	5718_M445	7266_S47	8794_S1542
0390_B302	2665_D97	3795_H14	5719_M446	7267_S48	8795_S1543
0391_B303	2666_D98	3928_H147	5796_M522a	7282_S63	8988_T58
0392_B304	2709_D141	3944_H163	5797_M523	7283_S64	8989_T59
0557_B461	2841_D262	3964_H183	5798_M524	7284_S65	8990_T60
0558_B462	2842_D263	4081_H300	5831_M557	7285_S66	9167_T198
0650_B554	2843_D264	4082_H301	5835_M561	7350_S131	9168_T199
0688_B592	2960_D372	4091_H310	5853_M579	7351_S132	9169_T200
0689_B593	2961_D373	4092_H311	5890_M615	7352_S133	9170_T201
0804_B687	3022_D434	4116_H336	5894_M619	7353_S134	9270_T301
0880_B762	3023_D435	4484_H703	5957_M682	7433_S214	9271_T302
0881_B763	3024_D436	4485_H704	5962_M687	7520_S301	9412_V123
1473_B1337	3031_D443	4856_K30	5963_M688	7538_S319	
1474_B1338	3032_D444	4857_K31	6011_M736	7589_S370	
1475_B1339	3033_D445	4858_K32	6097_M822	7590_S371	

Table B.3: BM-Small subset

Experimental Results

The results of different parameter settings that was excluded from the Section 5.2 is included here. As explained before, these combinations of parameters were used only to understand the variations it make in the retrieval quality. Since these combinations waned the retrieval quality and it was excluded from the main sections.

Midi121	Midi2132	Midi338	Midi4772	Midi5974	Midi7787
Midi1296	Midi2133	Midi3483	Midi4854	Midi6398	Midi810
Midi1297	Midi2266	Midi3899	Midi5181	Midi6467	Midi8470
Midi13	Midi2282	Midi391	Midi5182	Midi6468	Midi8471
Midi131	Midi2302	Midi392	Midi5254	Midi656	Midi8472
Midi1358	Midi2420	Midi395	Midi5255	Midi6723	Midi8473
Midi1359	Midi2421	Midi396	Midi5598	Midi6724	Midi8854
Midi1360	Midi2430	Midi397	Midi5601	Midi6725	Midi8855
Midi1367	Midi2431	Midi4051	Midi5602	Midi6726	Midi8856
Midi1368	Midi2455	Midi4056	Midi563	Midi694	Midi8857
Midi1369	Midi247	Midi4145	Midi5632	Midi695	Midi886
Midi14	Midi262	Midi4146	Midi5633	Midi7154	Midi887
Midi172	Midi266	Midi4147	Midi564	Midi7155	Midi9129
Midi1777	Midi2824	Midi4181	Midi5648	Midi7156	Midi9186
Midi1778	Midi2825	Midi4185	Midi5649	Midi7167	Midi9213
Midi1779	Midi302	Midi4203	Midi5650	Midi7168	Midi9214
Midi1780	Midi3201	Midi4241	Midi5651	Midi7169	Midi9607
Midi1794	Midi3202	Midi4244	Midi5717	Midi7362	Midi9674
Midi1795	Midi3234	Midi4307	Midi5718	Midi7363	Midi9675
Midi1796	Midi3235	Midi4312	Midi5719	Midi7364	Midi9718
Midi1797	Midi3236	Midi4313	Midi5720	Midi7541	Midi9850
Midi1798	Midi3237	Midi4362	Midi5800	Midi7542	Midi9851
Midi184	Midi3238	Midi4448	Midi5888	Midi7543	Midi9852
Midi1955	Midi3239	Midi4466	Midi5906	Midi7544	
Midi1956	Midi3240	Midi4598	Midi5957	Midi7645	
Midi2128	Midi3241	Midi4599	Midi5958	Midi7646	

Table B.4: EDM-Small subset

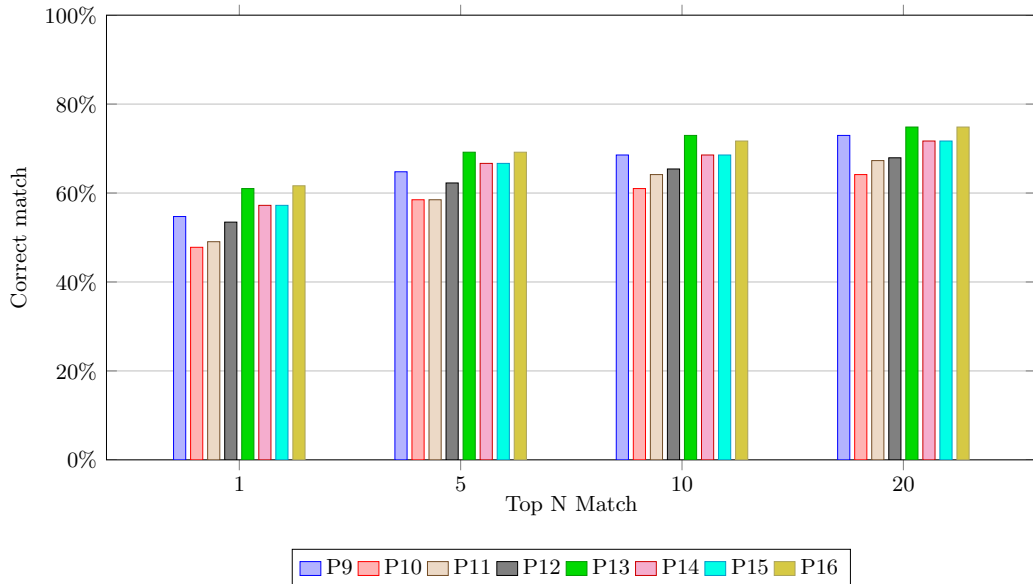


Figure B.1: Effects of different parameter settings on BM-Small subset containing 159 elements. P9 to P16 are the combination of different parameter settings which is explained in table B.5.

Parameters	P9	P10	P11	P12	P13	P14	P15	P16
Resolution (Hz)	-	-	-	50	50	-	-	-
Time (W/ WO)	W	W	WO	W	W	W	W	W
Normalization (Q/WP)	Q	WP	Q	WP	WP	WP	WP	WP
Constant Cost (Y/N)	Y	Y	Y	Y	Y	N	Y	N
Fixed Length (Y/N)	Y	Y	N	N	N	Y	Y	Y
Threshold (Y/N)	Y	Y	Y	Y	Y	N	N	N
DTW step size	Σ_1	Σ_1	Σ_1	Σ_1	Σ_1	Σ_3	Σ_2	Σ_2
Mean Rank (159 queries)	52.06	46.26	37.05	56.21	53.33	59.49	65.88	66.58
Computation Time (min)	0.25	0.70	0.16	1.39	3.59	0.67	0.55	0.60

Table B.5: Different parameter settings used for finding the best retrieval quality. W: With time; WO: Without time; Q: Normalized by the length of the query; WP: Normalized by the length of the warping path; Y: Enabled; N : Disabled; Σ_1 : $\{(1,0),(0,1),(1,1)\}$; Σ_2 : $\{(2,1),(1,2),(1,1)\}$; Σ_3 : $\{(1,1)\}$.

Appendix C

Inconsistencies

The inconsistencies occurred in the book [4] are reported in this section. They are as follows.

- BM-ThemeOrigID from C379 to C386 are missing (page 138).
- BM-ThemeOrigID G362 appears twice (page 212). First occurrence is corrected to G360.
- BM-ThemeOrigID H313 is missing (page 228).
- BM-ThemeOrigID from L71 to L74 are missing (page 277).

Due to the poor image quality, normal segmentation parameters failed for certain pages. For such pages, modified parameters were used. A list of these pages are given below. These numbers correspond to the page number in the book [4].

- 30
- 35
- 271
- 377
- 430
- 474
- 511

Appendix D

List of Files were OMR Failed

OMR engine failed to convert many musical themes into symbolic representations. The main reason, as explained in section 3.2, was the low quality input images. OMR failed to convert 1788 musical themes (queries). Here, we list the names of all the themes, for which OMR procedure failed.

0005_A5	2271_C284	3898_H117	5676_M406	7206_R436	8761_S1521a
0008_A8	2276_C289	3899_H118	5688_M418	7212_R442	8763_S1521c
0016_A16	2282_C295	3900_H119	5689_M419	7214_R444	8765_S1521e
0034_A34	2296_C309	3901_H120	5692_M422	7215_R445	8770_S1525
0037_A37	2332_C344	3905_H124	5699_M426c	7229_S10	8771_S1526
0054_A54	2333_C345	3920_H139	5716_M443	7230_S11	8773_S1528
0055_A55	2343_C355	3951_H170	5719_M446	7235_S16	8777_S1532
0059_A59	2352_C364	3962_H181	5721_M448	7239_S20	8778_S1533
0062_A62	2363_C375	3980_H199	5735_M462	7242_S23	8779_S1534
0068_A68	2370_C390	3981_H200	5740_M467	7243_S24	8783_S1534d
0071_A71	2371_C391	3982_H201	5746_M473	7244_S25	8797_S1545
0078_A78	2375_C395	3988_H207	5750_M477	7246_S27	8798_S1546
0093_B9	2391_C411	3996_H215	5754_M481	7255_S36	8799_S1547
0094_B10	2400_C420	3997_H216	5769_M496	7263_S44	8804_S1552
0105_B21	2401_C421	3999_H218	5775_M502	7266_S47	8812_S1560
0112_B28	2410_C430	4001_H220	5776_M503	7268_S49	8817_S1565
0114_B29a	2416_C436	4003_H222	5778_M505	7270_S51	8819_S1567
0125_B39	2422_C442	4005_H224	5779_M506	7271_S52	8824_S1572
0128_B42	2423_C443	4006_H225	5794_M521	7276_S57	8827_S1575
0140_B54	2427_C447	4022_H241	5798_M524	7277_S58	8828_S1576
0148_B62	2430_C450	4025_H244	5799_M525	7278_S59	8829_S1577
0155_B69	2471_C491	4056_H275	5804_M530	7281_S62	8836_S1584
0159_B73	2484_C504	4061_H280	5805_M531	7282_S63	8838_S1586
0163_B77	2487_C507	4078_H297	5806_M532	7285_S66	8840_S1588
0164_B78	2492_C512	4079_H298	5827_M553	7289_S70	8844_S1592
0171_B85	2496_C516	4081_H300	5871_M596	7293_S74	8845_S1593
0172_B86	2502_C522	4083_H302	5877_M602	7297_S78	8847_S1595
0185_B99	2506_C526	4084_H303	5895_M620	7305_S86	8857_S1605
0200_B114	2512_C532	4087_H306	5916_M641	7319_S100	8859_S1607
0249_B163	2516_C536	4122_H342	5956_M681	7322_S103	8860_S1608
0268_B180a	2532_C552	4124_H344	5964_M689	7324_S105	8861_S1609
0291_B203	2534_C554	4135_H355	5966_M691	7326_S107	8868_S1616
0292_B204	2537_C557	4160_H380	5969_M694	7329_S110	8869_S1617
0302_B214	2550_C570	4162_H382	5976_M701	7331_S112	8870_S1618
0304_B216	2551_C571	4171_H391	5980_M705	7335_S116	8871_S1619

D. LIST OF FILES WERE OMR FAILED

0307_B219	2556_C576	4178_H398	5997_M722	7337_S118	8874_S1622
0310_B222	2562_C582	4183_H403	6017_M742	7338_S119	8875_S1623
0315_B227	2568_C588	4198_H418	6018_M743	7339_S120	8877_S1625
0324_B236	2569_D1	4234_H453	6068_M793	7340_S121	8878_S1626
0326_B238	2578_D10	4235_H454	6133_M858	7341_S122	8882_S1629a
0331_B243	2579_D11	4243_H462	6153_M878	7342_S123	8883_S1629b
0333_B245	2581_D13	4261_H480	6155_M880	7343_S124	8884_S1629c
0334_B246	2586_D18	4264_H483	6159_M884	7345_S126	8888_S1631
0337_B249	2587_D19	4305_H524	6169_M894	7346_S127	8891_S1634
0350_B262	2590_D22	4311_H530	6183_M908	7348_S129	8892_S1635
0353_B265	2594_D26	4324_H543	6191_M916	7350_S131	8894_S1637
0354_B266	2602_D34	4325_H544	6194_M919	7354_S135	8898_S1641
0355_B267	2603_D35	4344_H563	6196_M921	7359_S140	8904_S1647
0356_B268	2605_D37	4350_H569	6215_M940	7363_S144	8907_S1650
0357_B269	2606_D38	4354_H573	6218_M943	7382_S163	8910_S1653
0359_B271	2609_D41	4355_H574	6224_M949	7385_S166	8911_S1654
0360_B272	2611_D43	4363_H582	6226_M951	7389_S170	8924_S1667
0361_B273	2615_D47	4364_H583	6231_M956	7390_S171	8929_S1672
0362_B274	2618_D50	4375_H594	6242_M967	7391_S172	8935_T5
0363_B275	2620_D52	4390_H609	6249_M974	7392_S173	8939_T9
0369_B281	2625_D57	4396_H615	6258_M983	7393_S174	8942_T12
0386_B298	2628_D60	4400_H619	6277_M1002	7399_S180	8943_T13
0387_B299	2630_D62	4408_H627	6285_M1010	7401_S182	8946_T16
0389_B301	2640_D72	4439_H658	6295_M1020	7410_S191	8952_T22
0390_B302	2643_D75	4442_H661	6298_M1023	7413_S194	8961_T31
0391_B303	2651_D83	4477_H696	6304_M1029	7414_S195	8966_T36
0397_B309	2652_D84	4480_H699	6310_M1035	7415_S196	8971_T41
0401_B313	2654_D86	4482_H701	6315_N2	7421_S202	8973_T43
0402_B314	2659_D91	4489_H708	6319_N6	7429_S210	8974_T44
0409_B321	2660_D92	4493_H712	6321_N8	7432_S213	8975_T45
0410_B322	2664_D96	4498_H717	6323_N10	7434_S215	8976_T46
0418_B330	2666_D98	4529_H748	6326_N13	7435_S216	8977_T47
0425_B337	2668_D100	4532_H751	6332_N19	7440_S221	8978_T48
0429_B341	2670_D102	4536_H755	6333_N20	7443_S224	8979_T49
0452_B364	2674_D106	4542_H761	6336_N23	7446_S227	8981_T51
0474_B386	2687_D119	4561_H780	6341_N28	7459_S240	8982_T52
0476_B388	2692_D124	4568_H787	6344_N31	7460_S241	8983_T53
0482_B394	2693_D125	4582_H801	6346_N33	7466_S247	8985_T55
0487_B399	2698_D130	4583_H802	6348_N35	7469_S250	8994_T64
0490_B402	2701_D133	4613_H832	6350_N37	7471_S252	8998_T68
0491_B403	2704_D136	4616_H835	6355_O4	7473_S254	8999_T69
0492_B404	2706_D138	4620_H839	6356_O5	7478_S259	9012_T82
0504_B416	2708_D140	4636_H855	6357_O6	7481_S262	9013_T83
0505_B417	2709_D141	4646_H865	6358_O7	7483_S264	9015_T85
0518_B430	2712_D144	4652_I4	6360_O9	7491_S272	9018_T88
0522_B434	2715_D147	4653_I5	6361_O10	7500_S281	9022_T92
0527_B439	2725_D157	4654_I6	6362_O11	7501_S282	9024_T94
0530_B442	2732_D164	4655_I7	6365_O14	7502_S283	9027_T97
0531_B443	2733_D165	4656_I8	6366_O15	7539_S320	9032_T102
0537_B449	2735_D167	4658_I10	6367_O16	7547_S328	9033_T103
0540_B452	2738_D170	4659_I11	6375_O24	7580_S361	9036_T106
0548_B454e	2746_D178	4660_I12	6378_O27	7602_S383	9038_T108
0550_B454g	2749_D181	4662_I14	6387_P5	7612_S393	9039_T109
0562_B466	2752_D184	4665_I17	6405_P23	7640_S421	9051_T111j
0565_B469	2753_D185	4680_I32	6409_P27	7659_S440	9078_T119
0587_B491	2757_D189	4682_I34	6412_P30	7677_S458	9079_T120
0605_B509	2762_D194	4685_I37	6415_P33	7689_S470	9080_T121

D. LIST OF FILES WERE OMR FAILED

0611_B515	2768_D200	4691_I43	6432_P50	7690_S471	9084_T122c
0618_B522	2772_D204	4692_I44	6433_P51	7700_S481	9088_T126
0619_B523	2792_D223a	4705_I57	6436_P54	7708_S489	9097_T135
0620_B524	2803_D229	4709_I61	6444_P61a	7709_S490	9115_T146
0621_B525	2822_D248	4710_I62	6445_P62	7712_S493	9117_T148
0627_B531	2823_D249	4716_I68	6446_P63	7721_S502	9151_T182
0652_B556	2824_D250	4725_I77	6448_P65	7724_S505	9156_T187
0659_B563	2829_D255	4728_I80	6451_P68	7730_S511	9161_T192
0661_B565	2831_D257	4730_I82	6461_P78	7743_S524	9185_T216
0668_B572	2834_D260	4734_I86	6463_P80	7771_S552	9186_T217
0691_B595	2843_D264	4738_I90	6476_P93	7774_S555	9187_T218
0699_B602a	2853_D274	4740_I92	6479_P96	7783_S564	9191_T222
0709_B612	2857_D278	4745_I97	6481_P98	7785_S566	9196_T227
0718_B621	2863_D284	4746_I98	6482_P99	7800_S581	9199_T230
0763_B663	2867_D288	4749_I101	6487_P104	7802_S583	9201_T232
0765_B663b	2871_D292	4751_I103	6491_P108	7804_S584a	9203_T234
0767_B665	2882_D303	4765_I117	6500_P114	7806_S584c	9205_T236
0771_B667b	2883_D304	4773_I125	6502_P116	7807_S584d	9212_T243
0787_B679b	2886_D307	4786_J7	6505_P119	7810_S584g	9221_T252
0790_B679e	2887_D308	4790_J11	6510_P124	7813_S586	9222_T253
0795_B680a	2896_D317	4795_J16	6516_P130	7832_S605	9225_T256
0830_B712	2902_D323	4797_J18	6517_P131	7837_S610	9226_T257
0831_B713	2903_D324	4799_J20	6518_P132	7840_S613	9227_T258
0838_B720	2916_D332e	4801_J22	6519_P133	7843_S616	9228_T259
0868_B750	2917_D332f	4802_J23	6524_P138	7847_S620	9231_T262
0880_B762	2918_D332g	4816_J37	6527_P141	7851_S624	9232_T263
0884_B766	2919_D332h	4817_K1	6533_P147	7856_S629	9243_T274
0888_B770	2921_D333	4824_K8	6540_P154	7858_S631	9250_T281
0892_B774	2926_D338	4826_K10	6541_P155	7865_S638	9264_T295
0899_B781	2927_D339	4832_K16	6542_P156	7882_S655	9265_T296
0900_B782	2928_D340	4834_K18	6543_P157	7890_S663	9268_T299
0910_B792	2938_D350	4835_K19	6545_P159	7894_S667	9269_T300
0918_B800	2942_D354	4845_K22c	6553_P167	7903_S676	9270_T301
0920_B802	2953_D365	4846_K22d	6554_P168	7919_S692	9271_T302
0923_B805	2955_D367	4847_K22e	6560_P174	7933_S706	9272_T303
0955_B837	2961_D373	4849_K23	6571_P185	7955_S728	9273_T304
0979_B861	2962_D374	4850_K24	6582_P196	7972_S745	9275_T306
0980_B862	2972_D384	4854_K28	6584_P198	7988_S761	9276_T307
0981_B863	2973_D385	4856_K30	6586_P200	7999_S772	9277_T308
0982_B864	2979_D391	4858_K32	6587_P201	8012_S785	9279_T310
0986_B868	2983_D395	4863_K37	6590_P204	8019_S792	9282_T313
1001_B883	2985_D397	4868_K42	6596_P210	8022_S795	9284_T315
1011_B893	2988_D400	4869_K43	6597_P211	8023_S796	9285_T316
1017_B899	2992_D404	4871_K45	6598_P212	8034_S807	9286_T317
1018_B900	3026_D438	4872_K46	6602_P216	8039_S812	9287_T318
1020_B902	3029_D441	4877_K51	6612_P226	8040_S813	9288_T319
1021_B903	3038_D450	4883_K57	6615_P229	8042_S815	9289_T320
1025_B907	3057_D469	4887_K61	6620_P234	8043_S816	9290_V1
1030_B912	3059_E2	4891_K65	6621_P235	8044_S817	9291_V2
1040_B922	3069_E12	4895_K69	6622_P236	8045_S818	9292_V3
1051_B933	3079_E22	4901_K75	6629_P243	8046_S819	9293_V4
1063_B945	3082_E25	4903_K77	6641_P255	8047_S820	9295_V6
1072_B954	3084_E27	4904_K78	6644_P258	8048_S821	9296_V7
1114_B996	3091_E34	4909_K83	6646_P260	8050_S823	9314_V25
1115_B997	3092_E35	4912_K86	6652_P266	8060_S833	9315_V26
1126_B1008	3093_E36	4913_K87	6655_P269	8063_S836	9322_V33
1130_B1012	3095_E38	4918_K92	6656_P270	8071_S844	9324_V35

D. LIST OF FILES WERE OMR FAILED

1154_B1036	3097_E40	4920_K94	6657_P271	8073_S846	9327_V38
1160_B1042	3104_E47	4923_K97	6663_P277	8076_S849	9331_V42
1167_B1049	3120_E63	4925_K99	6666_P280	8077_S850	9334_V45
1178_B1060	3131_E74	4926_K100	6669_P283	8078_S851	9337_V48
1188_B1070	3132_E75	4929_K103	6671_P285	8080_S853	9344_V55
1189_B1071	3136_E79	4931_K105	6672_P286	8088_S861	9347_V58
1194_B1076	3138_E81	4936_K110	6675_P289	8092_S865	9349_V60
1218_B1100	3143_E86	4939_K113	6679_P293	8096_S869	9352_V63
1224_B1106	3144_E87	4940_K114	6680_P294	8098_S870a	9353_V64
1233_B1115	3145_E88	4941_K115	6681_P295	8104_S870g	9354_V65
1242_B1117g	3148_E91	4953_L1	6683_P297	8106_S872	9358_V69
1262_B1126	3149_E92	4954_L2	6684_P298	8108_S874	9360_V71
1263_B1127	3157_E100	4955_L3	6685_P299	8112_S878	9361_V72
1264_B1128	3164_E107	4957_L5	6689_P303	8114_S880	9363_V74
1267_B1131	3165_F1	4959_L7	6690_P304	8115_S881	9364_V75
1268_B1132	3166_F2	4960_L8	6698_P312	8120_S886	9366_V77
1275_B1139	3167_F3	4961_L9	6711_P325	8122_S888	9367_V78
1285_B1149	3168_F4	4963_L11	6712_P326	8124_S890	9377_V88
1288_B1152	3169_F5	4964_L12	6714_P328	8128_S894	9383_V94
1294_B1158	3177_F13	4965_L13	6737_P351	8129_S895	9384_V95
1295_B1159	3180_F16	4967_L15	6738_P352	8130_S896	9389_V100
1309_B1173	3187_F23	4969_L17	6740_P354	8132_S898	9395_V106
1323_B1187	3188_F24	4972_L20	6742_P356	8134_S900	9398_V109
1333_B1197	3189_F25	4974_L22	6754_Q3	8140_S906	9400_V111
1338_B1202	3203_F39	4975_L23	6755_Q4	8142_S908	9402_V113
1340_B1204	3206_F42	4977_L25	6762_Q11	8145_S911	9408_V119
1347_B1211	3211_F47	4978_L26	6763_Q12	8149_S915	9419_V130
1348_B1212	3213_F49	4979_L27	6767_Q16	8151_S917	9420_V131
1353_B1217	3219_F55	4988_L36	6773_R6	8152_S918	9421_V132
1359_B1223	3220_F56	4998_L46	6780_R13	8153_S919	9427_V138
1368_B1232	3225_F61	5000_L48	6784_R17	8157_S923	9429_V140
1369_B1233	3233_F69	5001_L49	6791_R24	8168_S930	9430_V141
1371_B1235	3240_F76	5002_L50	6801_R34	8169_S931	9433_V144
1373_B1237	3241_F76a	5004_L52	6810_R43	8172_S934	9434_V145
1386_B1250	3245_F77	5006_L54	6811_R44	8181_S943	9437_V148
1394_B1258	3254_F86	5010_L58	6814_R47	8183_S945	9439_V150
1404_B1268	3256_F88	5012_L60	6822_R55	8184_S946	9440_V151
1411_B1275	3260_F92	5013_L61	6825_R58	8185_S947	9443_V154
1419_B1283	3273_F105	5014_L62	6829_R62	8186_S948	9445_V156
1424_B1288	3280_F112	5024_L76	6840_R73	8190_S952	9447_V158
1429_B1293	3283_F115	5029_L81	6848_R81	8198_S960	9448_V159
1464_B1328	3290_F122	5030_L82	6850_R83	8205_S967	9449_V160
1465_B1329	3291_F123	5032_L84	6851_R84	8209_S971	9454_V165
1467_B1331	3296_F128	5033_L85	6858_R91	8218_S980	9458_V168
1470_B1334	3300_F132	5035_L87	6860_R93	8219_S981	9461_V171
1490_B1354	3304_F136	5037_L89	6864_R97	8220_S982	9462_V172
1501_B1365	3307_F139	5040_L92	6865_R98	8221_S983	9470_W1
1507_B1371	3309_F141	5046_L98	6868_R101	8232_S994	9471_W2
1528_B1392	3311_F143	5055_L107	6871_R104	8238_S1000	9472_W3
1532_B1396	3315_F147	5062_L114	6874_R107	8281_S1043	9474_W5
1545_B1409	3317_F149	5063_L115	6875_R108	8304_S1066	9489_W20
1552_B1416	3318_F150	5064_L116	6876_R109	8341_S1103	9491_W22
1554_B1418	3321_F153	5067_L119	6878_R111	8342_S1104	9510_W41
1555_B1419	3323_F155	5068_L120	6880_R113	8347_S1109	9512_W43
1562_B1426	3332_F164	5073_L125	6883_R116	8357_S1119	9516_W47
1563_B1427	3343_F175	5076_L128	6887_R120	8358_S1120	9518_W49
1572_B1436	3346_F178	5083_L135	6889_R122	8360_S1122	9519_W50

D. LIST OF FILES WERE OMR FAILED

1574_B1438	3351_F183	5084_L136	6892_R125	8374_S1136	9524_W55
1577_B1441	3361_F193	5086_L138	6893_R126	8376_S1138	9532_W63
1583_B1447	3368_F200	5089_L141	6896_R129	8384_S1146	9540_W71
1597_B1461	3371_F203	5092_L144	6901_R134	8388_S1150	9549_W80
1601_B1465	3381_F213	5100_L152	6902_R135	8391_S1153	9552_W83
1603_B1467	3383_F215	5101_L153	6903_R136	8396_S1158	9555_W86
1609_B1473	3392_F224	5105_L157	6904_R137	8398_S1160	9557_W88
1642_B1506	3409_G12	5109_L161	6910_R143	8407_S1169	9563_W94
1643_B1507	3414_G17	5112_L164	6912_R145	8410_S1172	9565_W96
1644_B1508	3420_G20c	5130_L182	6913_R146	8412_S1174	9570_W101
1655_B1519	3421_G20d	5140_L192	6924_R157	8418_S1180	9571_W102
1659_B1523	3428_G27	5142_L194	6926_R159	8419_S1181	9572_W103
1661_B1525	3431_G30	5143_L195	6927_R160	8422_S1184	9575_W106
1700_B1564	3433_G32	5149_L201	6929_R162	8423_S1185	9576_W107
1701_B1565	3435_G34	5150_L202	6932_R165	8424_S1186	9578_W109
1707_B1571	3442_G41	5154_L206	6936_R169	8432_S1194	9579_W110
1710_B1574	3449_G48	5159_L211	6941_R174	8434_S1196	9580_W111
1741_B1605	3450_G49	5176_L228	6942_R175	8435_S1197	9582_W113
1742_B1606	3451_G50	5190_L242	6945_R178	8436_S1198	9584_W115
1744_B1608	3452_G51	5202_L254	6946_R179	8438_S1200	9589_W120
1748_B1612	3454_G53	5207_L259	6947_R180	8446_S1208	9593_W124
1760_B1624	3455_G54	5209_L261	6950_R183	8453_S1215	9594_W125
1763_B1627	3459_G58	5215_L267	6961_R194	8456_S1218	9601_W132
1801_B1665	3475_G74	5220_L272	6967_R200	8457_S1219	9602_W133
1811_B1675	3476_G75	5222_L274	6974_R207	8460_S1222	9604_W135
1830_B1694	3478_G77	5223_L275	6987_R220	8462_S1224	9605_W136
1848_B1711a	3480_G78a	5224_L276	6989_R222	8464_S1226	9615_W146
1849_B1711b	3490_G81	5226_L278	6991_R224	8467_S1229	9624_W155
1855_B1711h	3495_G86	5227_L279	6992_R225	8470_S1232	9626_W157
1857_B1711j	3498_G89	5230_L282	6993_R226	8474_S1236	9635_W166
1858_B1711k	3500_G91	5234_L286	6995_R228	8478_S1240	9641_W172
1880_B1730	3508_G98	5243_L295	7003_R236	8479_S1241	9643_W174
1884_B1734	3513_G103	5248_L300	7012_R245	8481_S1243	9648_W179
1885_B1735	3515_G105	5249_L301	7013_R246	8482_S1244	9649_W180
1889_B1739	3524_G114	5256_L308	7020_R253	8485_S1247	9650_W181
1902_B1752	3527_G117	5270_M10	7021_R254	8486_S1248	9653_W184
1915_B1765	3554_G144	5310_M50	7037_R270	8488_S1250	9654_W185
1936_B1786	3559_G149	5317_M57	7044_R277	8489_S1251	9656_W187
1944_B1794	3561_G151	5318_M58	7046_R279	8490_S1252	9658_W189
1957_B1807	3566_G156	5322_M62	7052_R285	8497_S1259	9659_W190
1958_B1808	3570_G160	5324_M64	7053_R286	8512_S1274	9660_W191
1963_B1813	3571_G161	5356_M96	7054_R287	8526_S1288	9664_W195
1966_B1816	3572_G162	5358_M98	7055_R288	8533_S1295	9667_W198
1968_B1818	3573_G163	5364_M104	7058_R291	8539_S1301	9669_W200
1969_B1819	3574_G164	5382_M122	7066_R299	8540_S1302	9670_W201
1970_B1820	3579_G169	5383_M123	7072_R303b	8542_S1304	9672_W203
1978_B1828	3581_G170	5385_M125	7076_R306	8543_S1305	9674_W205
1994_C8	3584_G173	5393_M131b	7078_R308	8547_S1309	9675_W206
1995_C9	3587_G176	5404_M142	7079_R309	8549_S1311	9677_W208
1996_C10	3589_G178	5405_M143	7081_R311	8551_S1313	9678_W209
2007_C21	3610_G199	5406_M144	7084_R314	8552_S1314	9679_W210
2010_C24	3611_G200	5409_M147	7086_R316	8566_S1328	9680_W211
2034_C48	3614_G203	5413_M151	7089_R319	8570_S1332	9681_W212
2035_C49	3619_G208	5418_M156	7093_R323	8572_S1334	9683_W214
2036_C50	3620_G209	5434_M172	7094_R324	8580_S1342	9687_W218
2046_C60	3632_G221	5437_M175	7095_R325	8581_S1343	9688_W219
2051_C65	3641_G230	5438_M176	7099_R329	8582_S1344	9690_W221

D. LIST OF FILES WERE OMR FAILED

2054_C68	3651_G240	5443_M181	7106_R336	8584_S1346	9696_W227
2057_C71	3656_G245	5447_M185	7108_R338	8586_S1348	9697_W228
2058_C72	3663_G252	5458_M196	7112_R342	8593_S1355	9698_W229
2075_C89	3669_G258	5471_M209	7113_R343	8599_S1361	9703_W234
2079_C93	3671_G260	5479_M217	7114_R344	8600_S1362	9708_W239
2091_C105	3698_G287	5482_M220	7115_R345	8602_S1364	9709_W240
2093_C107	3699_G288	5488_M226	7116_R346	8605_S1367	9714_W245
2102_C116	3701_G290	5489_M227	7119_R349	8613_S1375	9716_W247
2124_C138	3705_G294	5498_M236	7121_R351	8623_S1385	9725_W256
2130_C144	3706_G295	5502_M240	7123_R353	8627_S1389	9730_W261
2131_C145	3711_G299	5503_M241	7125_R355	8631_S1393	9737_W268
2133_C147	3714_G302	5505_M243	7131_R361	8633_S1395	9739_W270
2134_C148	3750_G338	5508_M246	7133_R363	8636_S1398	9740_W271
2135_C149	3757_G345	5515_M253	7134_R364	8640_S1402	9741_W272
2136_C150	3758_G346	5533_M271	7141_R371	8645_S1407	9742_W273
2145_C159	3761_G349	5535_M273	7142_R372	8647_S1409	9747_W278
2146_C160	3763_G351	5546_M284	7145_R375	8648_S1410	9759_W290
2149_C163	3766_G354	5575_M313	7146_R376	8652_S1414	9761_W292
2154_C168	3772_G360	5578_M316	7147_R377	8664_S1426	9764_W295
2155_C169	3779_G367	5581_M319	7151_R381	8673_S1435	9765_W296
2168_C182	3786_H5	5582_M320	7152_R382	8674_S1436	9770_W301
2172_C186	3798_H17	5586_M324	7154_R384	8676_S1438	9771_W302
2173_C187	3803_H22	5591_M329	7158_R388	8678_S1440	9772_W303
2175_C189	3804_H23	5609_M347	7159_R389	8679_S1441	9774_W305
2179_C193	3826_H45	5614_M352	7160_R390	8680_S1442	9777_W308
2183_C197	3832_H51	5622_M360	7161_R391	8686_S1448	9787_Z9
2186_C199	3839_H58	5626_M364	7163_R393	8702_S1464	9789_Z11
2188_C201	3840_H59	5628_M366	7164_R394	8715_S1477	9790_Z12
2191_C204	3847_H66	5641_M373	7168_R398	8719_S1481	9792_E73a
2195_C208	3848_H67	5643_M375	7172_R402	8732_S1494	9793_E73b
2196_C209	3849_H68	5644_M376	7185_R415	8733_S1495	9794_E73c
2197_C210	3853_H72	5649_M381	7192_R422	8743_S1505	9796_E73e
2199_C212	3855_H74	5651_M383	7200_R430	8748_S1509	9799_E73h
2266_C279	3873_H92	5657_M389	7201_R431	8755_S1516	9801_E73j
2270_C283	3895_H114	5661_M393	7203_R433	8756_S1517	9803_T153a

Table D.1: List of files were OMR failed

Appendix E

Source Code

In this chapter, the headers of selected MATLAB functions created during the writing of this thesis are reproduced. The headers contain information about the name of the described function and its input/output behaviour and the required files if any.

Segmentation without Prior Knowledge

`croppingv1` function is used extract the selected themes available in [4]. See Section 2.2.2 for more details.

Sample usage:

```
croppingv1('A Dictionary of Musical Themes in JPEG format', 'A', 1, 1, 84);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Name: croppingv1
% Version: 2
% Date of Revision: 2014-07
% Programmer: Sanu Pulimootil
%
% Description:
% Extract the selected themes from the jpeg pages of the book 'A Dictionary
% of Musical Themes'.
%
% Input:
%     folder_name
%     alphabet
%     yloc
%     index
%     index_max
%
%
% Output:
%     Themes are saved in the current directory
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Segmentation with Prior Knowledge

`dmt_processing` function is used extract all the themes available in [4]. See Section 2.2.3 for more details.

Sample usage:

```
dmt_processing('A Dictionary of Musical Themes in JPEG format', 'BM_MusicalThemes_ID_20140320.xls');
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Name: dmt_processing
% Date of Revision: 2014-07
% Programmer: Sanu Pulimootil
%
% Description:
% Extract all the themes from the jpeg pages of the book 'A Dictionary of Musical Themes'
%
% Input:
%     foldername
%     excel_filename
%     parameter.first_page = 17;
%     parameter.last_page = 523;
%     parameter.output_folder = 'BM-Theme';
%     parameter.ID = 1;
%     parameter.visualizaion = 0;
%
% Required files:
%     'segmentation.m'
%     'cropping.m'
%     'sort_nat.m'
%
% Output:
%     Themes are saved in the parameter.output_folder
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Extracting Side Information

`bm_complete_extraction` function is used extract all the side information contained in a `bm-sheetmusic` page of the book [4]. See Section 2.1.1 for more details.

Sample usage:

```
bm\_complete\_extraction('A Dictionary of Musical Themes in JPEG format','Theme','BM-Work',BM-ThemeDescription);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Name: bm_complete_extraction
% Version: 2
% Date of Revision: 2014-07
% Programmer: Sanu Pulimootil
%
% Description:
% Segment and crop all the extra information contained in a bm-sheetmusic page like
% bm-work, bm-themedescription, bounding boxes, page number and save it
% with the bm-theme file name
%
% Input:
%     Book_folder_name
%     theme_folder
%     output_foldername 1
%     output_foldername 2
```



```

%
% Output:
%     extracted information are saved in the output_foldername
%     directory and bounding box information in theme structure
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Post Processing

`cleaning_theme` function is used clean a theme from the notes of the nearby themes. See Section 2.2.3.2 for more details.

Sample usage:

```
cleaning_theme('BM-Theme', paramter)
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Name: cleaning_theme
% Date of Revision: 2014-06
% Programmer: Sanu Pulimootil
%
% Description:
%     This program is used to clean a theme by removing
%     the notes that belongs to other themes.
%
% Input:
%     folder_name
%     parameter.output_folder
% Output:
%     Stores the cleaned version in the folder parameter.output_folder
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

OCR Analysis

`ocr_analysis` program is used for extracting text based matching list. See Section 4.3 for more details. Sample usage:

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Name: ocr_analysis
% Date of Revision: 2014-08
% Programmer: Sanu Pulimootil
%
% Description:
%     Extracting the text-based matching list.
%
% Input:
%     User will be asked for inputs such as directories and paths.
%
% Output:
%     Ranked lists saved in the folder 'OCR-Results'
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

OMR Analysis

`dtw_comparison` program is used for extracting text based matching list. See Section 4.4 for more details.

E. SOURCE CODE

Sample usage:

dtw_comparison

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Name: dtw_comparison
% Date of Revision: 2014-08
% Programmer: Sanu Pulimootil
%
% Description:
%     Compare OMR of EDM and BM themes
%
% Input:
%     User will be asked for inputs
%
%
% Output:
%     Ranked lists saved in the folder 'OMR-Results'
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Oracle Fusion

oracle_fusion_processing program is used for fusing the text-based and score-based matching list using oracle information. See Section 5.3 for more details.

Sample usage:

oracle_fusion_processing

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Name: oracle_fusion_processing
% Date of Revision: 2014-10
% Programmer: Sanu Pulimootil
%
% Description:
%     Oracle based fusion
% Input:
%     user will be asked for input.
% Output:
%     Rank List of the all the Queries will be saved
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Fusion

fusion_processing program is used for coordinating the fusion between the text-based and score-based matching. See Section 5.4 for more details.

Sample usage:

fusion_processing

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Name: fusion_processing
% Date of Revision: 2014-10
% Programmer: Sanu Pulimootil
%
% Description:
%     Coordinate fusion program and evaluate the fusion results
% Input:
%     user will be asked for input.
% Output:
%     Rank List of the all the Queries will be saved
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Bibliography

- [1] C. C. AGGARWAL AND P. S. YU, *On effective conceptual indexing and similarity search in text data.*, in ICDM, N. Cercone, T. Y. Lin, and X. Wu, eds., IEEE Computer Society, 2001, pp. 3–10.
- [2] D. BAINBRIDGE AND T. BELL, *The challenge of optical music recognition*, Computers and the Humanities, 35 (2001), pp. 95–121.
- [3] S. BALKE, S. P. ACHANKUNJU, AND M. MÜLLER, *Matching musical themes based on noisy OCR and OMR input (unpublished)*.
- [4] H. BARLOW AND S. MORGENSTERN, *A Dictionary of Musical Themes*, Crown Publishers, Inc., revised edition third printing ed., 1975.
- [5] H. BITTEUR, *Audiveris - open music scanner*. Website <https://audiveris.kenai.com>, last accessed 09/29/2014, 2013.
- [6] D. BYRD AND M. SCHINDELE, *Prospects for improving OMR with multiple recognizers*, in ISMIR 2006, 7th International Conference on Music Information Retrieval, Victoria, Canada, 8-12 October 2006, Proceedings, 2006, pp. 41–46.
- [7] T. H. CORMEN, C. E. LEISERSON, R. L. RIVEST, C. STEIN, ET AL., *Introduction to algorithms*, vol. 2, 2001.
- [8] J. S. DOWNIE, *Music information retrieval*, Annual Review of Information Science and Technology (Chapter 7), 37 (2003), pp. 295–340.
- [9] S. GARCÍA-DÍEZ, F. FOUSS, M. SHIMBO, AND M. SAERENS, *Normalized sum-over-paths edit distances*, in 20th International Conference on Pattern Recognition, ICPR 2010, Istanbul, Turkey, 23-26 August 2010, IEEE Computer Society, 2010, pp. 1044–1047.
- [10] R. C. GONZALEZ AND R. E. WOODS, *Digital image processing*, Addison-Wesley, 1992.
- [11] F. KURTH AND M. MÜLLER, *Efficient Index-Based Audio Matching*, IEEE Transactions on Audio, Speech, and Language Processing, 16 (2008), pp. 382–395.
- [12] V. LEVENSHTAIN, *Binary Codes Capable of Correcting Deletions, Insertions and Reversals*, Soviet Physics Doklady, 10 (1966), p. 707.
- [13] G. LV, C. ZHENG, AND L. ZHANG, *Text information retrieval based on concept semantic similarity*, in Semantics, Knowledge and Grid, 2009. SKG 2009. Fifth International Conference on, Oct 2009, pp. 356–360.
- [14] A. MARZAL AND E. VIDAL, *Computation of normalized edit distance and applications*, IEEE Trans. Pattern Anal. Mach. Intell., 15 (1993), pp. 926–932.
- [15] M. MÜLLER, *Information Retrieval for Music and Motion*, Springer Verlag, 2007.
- [16] M. MÜLLER, M. CLAUSEN, V. KONZ, S. EWERT, AND C. FREMEREY, *A multimodal way of experiencing and exploring music*, Interdisciplinary Science Reviews (ISR), 35 (2010), pp. 138–153.

BIBLIOGRAPHY

- [17] M. MÜLLER, M. GOTO, AND M. SCHEDL, eds., *Multimodal Music Processing*, vol. 3 of Dagstuhl Follow-Ups, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Germany, 2012.
- [18] N. ORIO, *Music retrieval: A tutorial and review*, Foundation and Trends in Information Retrieval, 1 (2006), pp. 1–90.
- [19] J. PICKENS, J. P. BELLO, G. MONTI, T. CRAWFORD, M. DOVEY, M. SANDLER, AND D. BYRD, *Polyphonic score retrieval using polyphonic audio*, in Proceedings of the International Conference on Music Information Retrieval (ISMIR), Paris, France, 2002.
- [20] L. RABINER AND B.-H. JUANG, *Fundamentals of Speech Recognition*, Prentice Hall Signal Processing Series, 1993.
- [21] J. T. SCHWARTZ AND D. SCHWARTZ, *The electronic dictionary of musical themes*. Website <http://www.multimedialibrary.com/barlow/>, last accessed 08/07/2014, 2008.
- [22] J. SERRÀ, E. GÓMEZ, AND P. HERRERA, *Audio cover song identification and similarity: background, approaches, evaluation and beyond*, in Advances in Music Information Retrieval, Z. W. Ras and A. A. Wierzchowska, eds., vol. 274 of Studies in Computational Intelligence, Springer, Berlin, Germany, 2010, ch. 14, pp. 307–332.
- [23] R. SMITH AND J. HOOPER, *A computer architecture to support natural full text information retrieval*, in Southeastcon '88., IEEE Conference Proceedings, Apr 1988, pp. 197–199.
- [24] X.-H. XU, J. LAI HUANG, J. WAN, AND C.-F. JIANG, *An integrated method for text information retrieval*, in Semantics, Knowledge and Grid, 2008. SKG '08. Fourth International Conference on, Dec 2008, pp. 400–403.