Music Score Recognition – A Selective Attention Approach using Mathematical Morphology

Bharath Modayur
e-mail: bharath@george.ee.washington.edu
Intelligent Systems Laboratory
Electrical Engineering Department, FT-10
University of Washington
Seattle WA 98195
U.S.A.

March 2, 1996

Abstract

Music representation utilizes a rich repertoire of symbols. These symbols appear on a score sheet with relatively little shape distortion, differing from the prototype symbols mainly by a positional translation and scale change. The recognition system described in this article is aimed at recognizing printed music notation from digitized images of music scores. The system is composed of two parts: a low-level vision module that uses morphological algorithms for symbol detection and a high-level module that employs a selective attention strategy to direct the symbol detection algorithms to specific regions in the image in which to look for symbols. The high-level module also employs verification procedures to check the veracity of the symbol recognizer output. Mathematical morphology, a shape-based approach to image processing, provides us the theory and the tools to analyze shapes. This characteristic of mathematical morphology lends itself well to analyzing and subsequently recognizing music score which is rich in well-defined symbols. Since morphological operations can be efficiently implemented on general purpose computers with minimal graphical support, the recognition task can be performed in real-time. The system achieves accuracy in excess of 95% on the sample scores processed so far with a peak accuracy of 99.7% for quarter and eighth notes, demonstrating the efficacy of morphological techniques for shape extraction.

Keywords: Mathematical morphology – Image processing – Music notation – Connected components – Feature extraction

1 Introduction

The performance of efficient algorithms to recognize isolated two-dimensional shapes deteriorates rapidly as the shapes become increasingly interconnected. Music score representation is one such domain where the different symbols are arranged in a truly two-dimensional fashion. Utilization of prior knowledge about music representation becomes necessary to devise efficient methods to recognize the music symbols.

A musical-score layout is the visual manifestation of interrelated properties of musical sound – pitch, intensity, time, timbre and pace [9]. Symbols indicating the tones, their duration and the manner of performance form the music notation. The basic symbols that we are concerned with in this work include the following.

1. Staff – an arrangement of five parallel lines together with spaces between them.
2. Clefs – symbols that determine the pitch associated with a particular staff.
3. Note heads and stems – symbols that determine a note's time value.
4. Flags and beams – other symbols that determine the time value of a note.
5. Rests and pauses – symbols that indicate moments of silence in a musical piece.
6. Accidentals – symbols that are placed in front of a note to modify its pitch, i.e., to raise or lower the pitch.
7. Barlines – thin vertical lines drawn through the staff to set off the time length of each measure.

Mathematical morphology, a shape-based approach to processing images, provides us the theory and the tools to analyze shapes. Printed music score is rich in well-defined but interconnected shapes. The shape-based approach of mathematical morphology thus lends itself well to being used as an effective tool in understanding printed music scores.

The recognition system described in this paper proceeds through three distinct stages to recognize music scores. They are: **layout extraction**, **symbol recognition**, and **high-level reasoning**. In the first stage, the skew and the scale of the image are determined by locating the most dominant symbol in the score sheet – the staff lines. This is achieved by a combination of image processing and morphological techniques. In the second step the different music symbols are extracted using sequences of morphological operations. Prior knowledge of music representation is utilized in making the detection algorithms simple as well as efficient. The last stage employs a **selective attention strategy**, which uses prior knowledge of music representation [9] to direct the symbol recognizer to specific regions in the image in which to look for symbols. This high-level module also employs verification procedures to check the veracity of the output from the morphological symbol recognizer (see Fig. 2).

The remainder of the paper is organized as follows. Section 2 outlines the problem we are trying to solve and the final goal. Section 3 briefly discusses related work. Section 4 gives a basic introduction to the morphological operations and notations used. Section 5 outlines the prior assumptions. Sections 6 and 7 describe the layout extraction and the symbol recognition modules. Section 7 also describes the selective attention strategy and how it can lead to faster recognition algorithms. Section 8 describes the high-level reasoning module and explains some the postprocessing for a few important classes of symbols. Section 9 discusses the experiments and the results obtained so far.

2 The Problem and The Goal

We have a known representation of music and an image of a printed score sheet. We also have prior knowledge of the various symbols that can appear on the score, the set of possible positions different symbols can occupy and some definite sequences of symbols. The problem is then to recognize all the meaningful symbols in the image, their spatial positions and the spatial sequence of these symbols (see [6] for an earlier version of this work). If there are N symbols – $S_1, S_2, ...S_N$ which occur in the image in unknown locations, the image can be represented as $I = \bigcup_{i=1}^{N}(S_i)_{x_i,y_i}$, where the S_i do not intersect (this condition is relaxed for symbols such as staff line and note stems that intersect other symbols). The purpose of the morphological sequence of operations is to extract the individual symbols with their respective translations. Let us assume that the first symbol, S_1 has been extracted and removed from I. Now, the residue image I_{res} can be represented as $I_{res} = \bigcup_{i=2}^{N}(S_i)_{x_i,y_i}$. The symbol extraction proceeds until I has been decomposed completely.

The goal is to represent the recognition result as a sequence of text symbols as follows:

3C4 3D8 [4C4 4E4]

This representation indicates a C note in the third octave (this is the C below middle C) that is a quarter note, followed by a D note in the third octave that is an eighth note. Then there is a chord consisting of a quarter C in the fourth octave and a quarter E in the fourth octave.

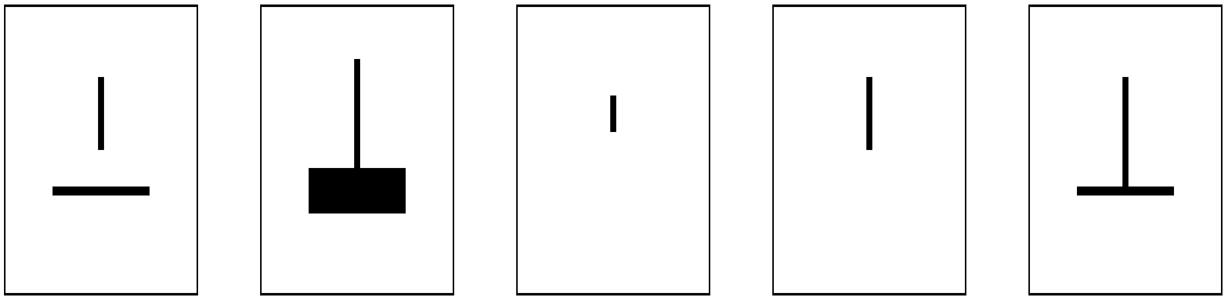
Since our recognition system is designed to work on printed forms of music representation, it enjoys the advantage of encountering symbols of regular shape. In this regard the problem is well defined. Once the image is normalized – which can be done if the spacing between the staff lines is known – the relative sizes of the various symbols can be determined. This is helpful especially in creating the appropriately sized structuring elements.

3 Related Work

The doctoral dissertations of Pruslin [8] and Prerau [7] used contour-tracing techniques for symbol segmentation and hence staff line separation was problematic. The gaps created as a result of staff line removal were filled and the aspect ratio of segmented symbols along with some music rules were used to recognize the symbols. Fujinaga's thesis work [3] used projection profile techniques for staff line detection as well as symbol segmentation and recognition. The projection techniques tolerate skew in the staff line orientation. This work assumed that the sizes of the different music symbols in a given score sheet are relative. The results were reported for four carefully chosen score samples. The recognition accuracy was 73% on the average.

The ENGRAVE system (for handwritten music score recognition) by Roach and Tatem reported in [10] exploited the fact that music representation is a highly structured domain and utilized knowledge of music to achieve significant improvements in low-level image processing to extract primitive features. The system processed six images of handwritten scores and the results were reported to be good. The system reported by Baumann and Dengel [1] used a top-down strategy in order to take advantage of the hierarchical structure of music representation. Using music rules and constraints, it applied a decision tree classifier to recognize music symbols with an accuracy of about 90%.

The MUSER system developed by Modayur et al. [5] utilizes music rules and constraints to some extent in the low-level symbol extraction stage and to a large extent in the high-level reasoning stage. The low-level stage in the recognition system uses morphological sequences in a *feature extraction-topology matching* strategy (to be explained later) to detect the music symbols. Staff line extraction, which is a lengthy and often troublesome process in other systems, is achieved by a single morphological operation. The biggest drawback of the implementation of MUSER is the fact that knowledge of music notation was



(a) Input image (b) Dilation (c) Erosion (d) Opening (e) Closing

Figure 1: The results of morphological operations using a vertical-line structuring element. The height of the structuring element equals the gap between the two strips in the original image.

not really integrated into the morphological symbol recognizer, leaving the high-level reasoning module to correct the mistakes made earlier in the symbol recognition process. This also led to the symbol recognizer being relatively inefficient and slow. In this work, I introduce a technique called **selective attention** that enhances MUSER by integrating the high-level reasoning module with the symbol recognizer in a tight, closed-loop. The use of the selective attention technique leads to: (i) a significant speedup of the recognition process, and (ii) lower probability of symbols being detected erroneously.

4 Morphological Operations and Notations

There are five basic morphological operations that are used in this work. A brief description of the operations and their mathematical notation is given in this section (see [4] for a more complete description).

4.1 Dilation

Dilation is the morphological transformation which combines two sets using vector addition of set elements. Let A and B be subsets of E^N. The dilation of A by B is denoted by $A \oplus B$ and is defined by

$$A \oplus B \quad = \quad \{c \in E^N \mid c = a + b, \quad a \in A, \quad b \in B\}.$$

In practice, the set A is considered as the image undergoing transformation and the set B is referred to as the structuring element. To dilate an image, the structuring element origin is run over all the binary one pixels of the image. The area swept by the structuring element is the dilated image. Fig. 1(a) shows a sample image and Fig. 1(b) shows the result of dilating the sample image with a vertical-line structuring element. The height of the structuring element is equal to the gap between the square and the narrow strip in the input image. As can be seen from the image, dilation has the property of expanding foreground regions.

4.2 Erosion

Erosion, the morphological dual of dilation, is a shrinking operation. It is the morphological transformation which combines two sets using vector subtraction of set elements.

Let A and B be subsets of E^N. The erosion of A by B is denoted by A $\ominus$ B and is defined by

$$A \ominus B \quad = \quad \{x \in E^N \mid x + b \in A \quad for \quad every \quad b \in B\}.$$

To erode an image, the structuring element origin is run over all the pixels in the image. Those pixels at which the structuring element origin can stand and where its entire area covers only binary one pixels are marked. The area of marked pixels is the eroded image. Fig. 1(c) shows the result of eroding the input image in Fig. 1(a) by the same line structuring element as before. Erosion has the effect of shrinking foreground regions.

4.3 Opening

The opening of an image A by a structuring element K is denoted by A $\circ$ K and is defined as $A \circ K = (A \ominus K) \oplus K$.

To open an image, the structuring element is run through the area of binary one pixels keeping the area of the structuring element contained in the area of binary one pixels in the image. The area swept by the structuring element is the opened image. Opening an image with a disk structuring element smooths the boundary, breaks narrow isthmuses and eliminates small specks (see [2] for a good look into various morphological sequences for pattern detection). Fig. 1(d) shows the result of opening the original image by the same line structuring element as in the previous cases. The narrow horizontal strip is absent in the opened image because it is not big enough to fit the vertical-line structuring element.

4.4 Closing

The closing of an image A by a structuring element K is denoted by A $\bullet$ K and is defined as $A \bullet K = (A \oplus K) \ominus K$.

Closing is the morphological dual of opening. What opening does to the foreground, closing does to the background. Closing an image with a disk structuring element smooths the boundary, fuses narrow breaks, fills small holes and fills gaps on the boundary. Fig. 1(e) shows the result of closing the input image in Fig. 1(a) by the same line structuring element as in the previous cases. Unlike the dilation operation, which resulted in the closing of the gap in Fig. 1(a) as well as the expansion of the foreground regions, the closing operation results only in the closing of the gap. Foreground regions not local to the gap remain unaffected.

4.5 Structuring Elements

The morphological symbol recognizer uses, among other shapes, three structuring elements of standard shape: (1) line-, (2) disk-, and (3) box-structuring elements. Their notations are as follows.

1. A line structuring element with origin at $(0, 0)$ and extending from (x_1, y_1) to (x_2, y_2) is denoted by $line(x_1, y_1, x_2, y_2)$.

2. A disk structuring element with its origin at the center and radius r is denoted by $disk\ \ r$.

3. A box structuring element with width w and height h is denoted by $box(w, h)$.

4.6 Image Assignment Notation

The equation

$$I_{original} = I_{original} \oplus disk5$$

would mean the following. The original image $I_{original}$ is to be dilated by a disk of radius 5 pixels and the resulting image is to replace the original image.

5 Prior Assumptions

The following assumptions are made about the structure of music scores that our system can handle.

1. The sizes of the different symbols are relative. This essentially means that once the scale of the image is computed, an approximate estimate of the sizes and shapes of the various symbols can be determined.

2. The staff lines are equally spaced. The layout extraction module does a median calculation on the spacing between the horizontal staff lines to determine the scale the image. Thus, we require the staff lines to be equally spaced.

3. The image does not have large skew. This is required to make the staff extraction algorithm remain simple. The staff is extracted by opening the image with a long horizontal line (35 pixels long). This would detect horizontal and near horizontal lines. If there is a large skew, the algorithm will fail to detect the staff lines.

The following assumptions are made about the representation of music scores.

1. The number of parallel lines in a staff is assumed to be five. This means that we do not allow for percussion staves, cue staves, etc.

2. We will deal with multiple voice music scores.

3. Current implementation assumes that the printed musical score follows some of the conventions specified in [11].

4. The distance between successive note heads in pixels is proportional to the number of pixels in the measure and the left note duration.

5. Accidentals are placed squarely on the line or the space directly in front of the note they alter. Our implementation just looks for notes in the bottom right quadrant of the accidental while associating accidentals with notes.

6. Stems, in general, go down when attached to the left of the note. They go up when attached to the right of the note. The stem length is normally the length of one octave.

There are a number of other conventions about rest symbols and augmentation dots that our system uses. For example, a quarter rest is assumed to be centered on the staff. A half rest touches the third staff line above, while a whole rest touches the fourth staff line below. Some of the assumptions mentioned above provide answers for the choices of thresholds used in the postprocessing stage that filters out spurious symbols that are detected in the feature extraction step.

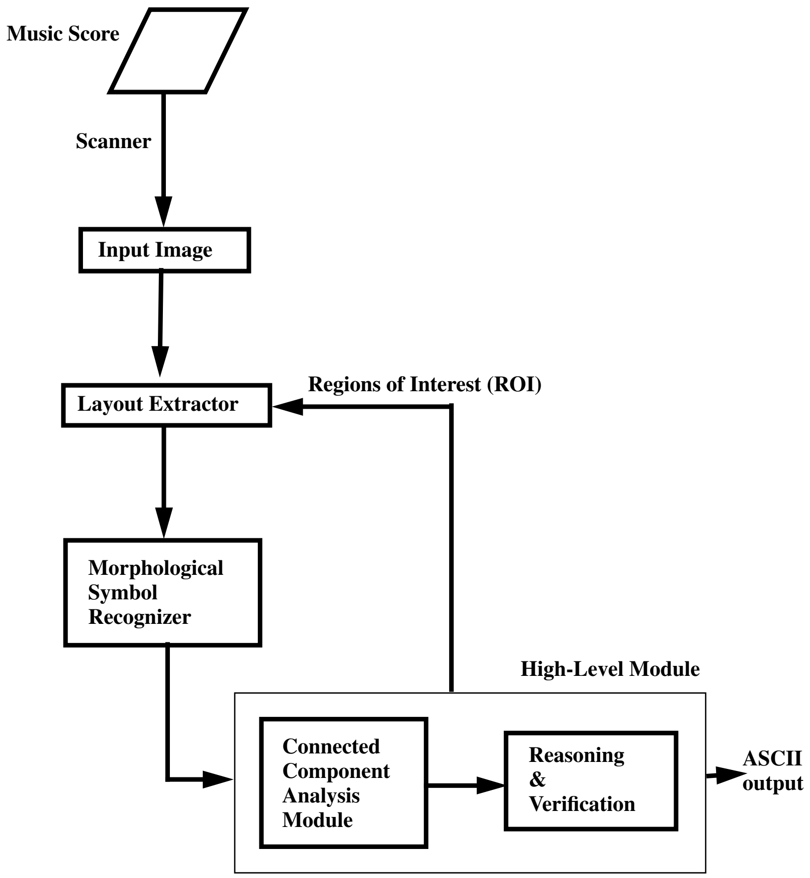


Figure 2: System Block Diagram. The symbol recognizer uses morphological techniques to extract the music symbols while the high-level module uses knowledge of music notation to check the veracity of the symbol recognizer output. The high-level module also directs the symbol recognizer to specific portions of the input image in which to look for symbols by employing a *selective attention* strategy.

6 Layout extractor

The first step in the recognition process is determining the scale and the skew of the image. The skew in the image causes the rectangular coordinate system, relative to which the different symbols are located, to be rotated. This rotation can be corrected once the skew has been determined. The scale of the image is used to fix the sizes of the different standard size structuring elements used by the symbol recognizer. The scaling of the standard structuring elements can be achieved in a semi-interactive way, provided polygonal approximations of the original structuring elements are available.

To extract the staff lines, the image is opened with a horizontal line 35 pixels long. Let this line structuring element be K.

$$I_{staff} = I_{input} \circ \mathbf{K}$$

A connected component analysis on I_{staff} would give information about the extremities of the horizontal lines in the image and their orientation. The median of the spacing between adjacent lines can now be computed. This would give an approximate estimate of the scale of the image. Let this estimated scale be denoted by S_I. The dimensions of all the structuring elements that are used for symbol extraction would then be scaled by this value. The orientation of the lines would indicate the skew of the image.

7 Morphological symbol recognizer

A preliminary examination of the symbols that occur in music representation reveals the following. There are some symbols whose features are exactly similar to (or in some cases, a subset of) others. The only factor that distinguishes the former from the latter may be the size. A typical example of this fact is the grace note. The only thing that distinguishes a grace note from a regular note is its size. This overlap of features among the different symbols necessitates the use of a prudent sequence of symbol extraction.

How do we decide which symbols to extract earlier and which ones later? The guideline that is employed in deciding the priority of a symbol in the extraction sequence is its foreground area and the lack of distinguishing concavities and curvatures. Symbols like the repeat measure bar, repeat symbol, clef, and full note heads would thus get a higher priority while the rest of the symbols, accidentals, and others that have distinguishing concavities, would come later in the symbol extraction sequence. A regular note would thus have to be extracted before the grace notes since we would be using similarly shaped structuring elements (but of different sizes) to extract both the symbols.

The algorithms used for symbol detection follow a two-step process. In the first step, the primitive features of a given symbol are extracted. The second step utilizes prior knowledge of the topological relationship among the different features to detect the symbol. Please note that the structuring elements employed throughout this symbol detection phase would "loosely" follow the shape of the medial axis (the skeleton) of the feature shape being sought. This is to incorporate a certain degree of tolerance in the detection process. Thus, imperfections such as a few missing foreground pixels, broken edges, and blurred corners will not affect the output of the symbol detection process. Algorithms that utilize the background features of symbols are less sensitive (less tuned) to structuring element sizes than the ones that use foreground features. Symbols like the notehead, however, do not possess background features and consequently are dependent on suitably sized structuring elements for their extraction. This is where the relative-size assumption comes in handy. We assume that with the scale of the image determined, the structuring element sizes can be fixed. This still does not imply that the structuring element sizes have no room for tolerance. The staff line extraction works as adequately with a horizontal line 20 pixels long as with the 35-pixel line that we used in our experiments. The notehead extraction still functions well with a disk-structuring element (7-pixel radius) as with the elliptical structuring element we used.

Once a symbol is detected, if it is local in nature, it is removed from the input image. This is to prevent the extracted symbol from giving positive results in subsequent processing. Exceptions to this are the staff lines and note stems which are not removed after detection.

In the following section, we will discuss a representative algorithm used in the feature extraction-topology matching strategy to detect different symbols. For a more complete discussion on the algorithms see [6].

7.1 A representative symbol detection algorithm

The bass clef is the last symbol in the symbol detection sequence. We discuss it here to illustrate the feature extraction-topology matching strategy. The important feature here is the head of the bass clef symbol with two dots, one below the other, to the right (see Fig. 3). The input image shown in Fig. 4 has two occurrences of the bass clef symbol. The head portion of the bass symbol is easily detected by opening the input image with a disk shaped structuring element K_{bass} (of $0.44\mathbf{S_I}$ pixel radius) with a downward tail (see Fig 3).

$$I_{bassHead} = I_{input} \circ K_{bass}$$

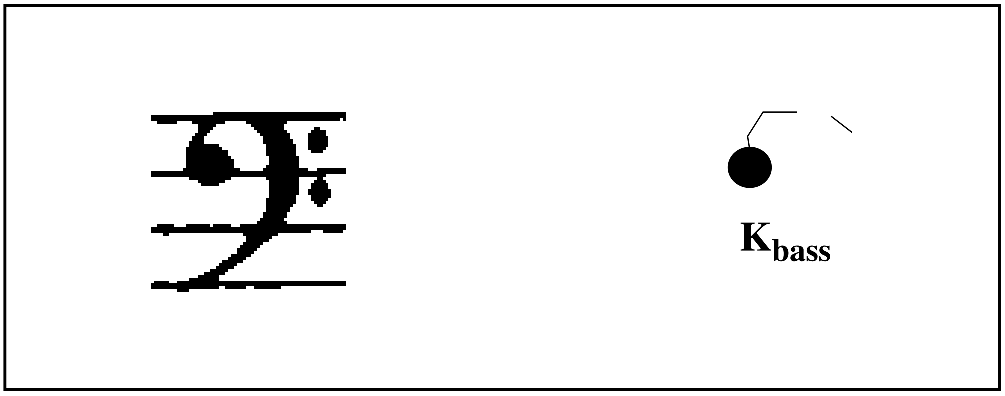


Figure 3: The bass symbol and the structuring element K_{bass} used in its detection.

Figure 4: A sample measure containing two occurrences of the bass clef symbol.

Since K_{bass} has gaps in it, $I_{bassHead}$ is closed with a box to fill the gaps. This produces a single connected component.

$$I_{bassHead} = I_{bassHead} \bullet box(1.67\mathbf{S_I}, 0.33\mathbf{S_I})$$

The result of this operation is illustrated in Fig. 5(a). Now, the two dots lying to the right of the head have to be extracted. To begin with, all the horizontal and vertical lines in the image are removed.

$$\begin{aligned} I_h &= I_{input} \circ line(0, 0, 1.11\mathbf{S_I}, 0) \\ I_v &= I_{input} \circ line(0, 0, 0, 1.11\mathbf{S_I}) \end{aligned}$$

Remove I_h and I_v from the image.

$$\begin{aligned} I_{temp} &= I_{input} \cap (I_h \cup I_v) \\ I_{noLines} &= I_{input} \;\; XOR \;\; I_{temp} \end{aligned}$$

Fig. 5(b) shows the image after the above operation. To locate the individual dots, the resultant image is opened with a disk. This should be followed by another opening to get rid of blobs bigger than the required size.

$$I_{dots} = I_{noLines} \circ disk0.11\mathbf{S_I}$$

Then the bigger blobs are removed.

$$\begin{aligned} I_{bigBlobs} &= I_{dots} \circ line(0, 0, 0.44\mathbf{S_I}, 0) \\ I_{temp} &= I_{bigBlobs} \cap I_{dots} \\ I_{dots} &= I_{dots} \;\; XOR \;\; I_{temp} \end{aligned}$$

Since we seek a pair of dots one vertically below the other separated by about $\mathbf{S_I}$ pixels, the image I_{dots} is closed with a vertical structuring element.

$$\begin{aligned} I_{dotPair} &= I_{dots} \bullet line(0, 0, 0, 1.11\mathbf{S_I}) \\ I_{dotPair} &= I_{dotPair} \;\; XOR \;\; I_{dots} \end{aligned}$$

The second operation produces results only at those places where pairs of dots occur, one vertically below the other and separated by less than $1.11\mathbf{S_I}$ pixels. This is illustrated in Fig. 5(c). The required features have now been extracted. What follows is another sequence of morphological operations to check the topology. Since the dot pair is expected to lie to the right of the bass head portion, the head is translated to the right and intersected with the dot pair.

$$\begin{aligned} I_{dilateRight} &= I_{bassHead} \oplus line(0.83\mathbf{S_I}, 0, 1.11\mathbf{S_I}, 0) \\ I_{bass} &= I_{dilateRight} \cap I_{dotPair} \end{aligned}$$

The result of the above operation is shown in Fig. 5(d). The operation identifies the locations of the bass clefs but also produces results in places where bass clefs are not present. In the actual symbol extraction sequence, bass clefs are extracted last. Thus, the other symbols would have been removed by then and the bass clef extraction results in foreground pixels only at those locations where the bass clef symbols are present. The following sequence is employed to get rid of the bass symbol from the input image.

$$
\begin{aligned}
I_{bassDil} &= I_{bass} \oplus disk0.22\mathbf{S_I} \\
I_{dilateLeft} &= I_{bassDil} \oplus line(0, 0, -1.67\mathbf{S_I}, 0) \\
I_{dilateDown} &= I_{dilateLeft} \oplus line(0, 0, 0, -1.67\mathbf{S_I}) \\
I_{temp} &= I_{dilateDown} \cap I_{input} \\
I_{input} &= I_{temp} \;\; XOR \;\; I_{input}
\end{aligned}
$$

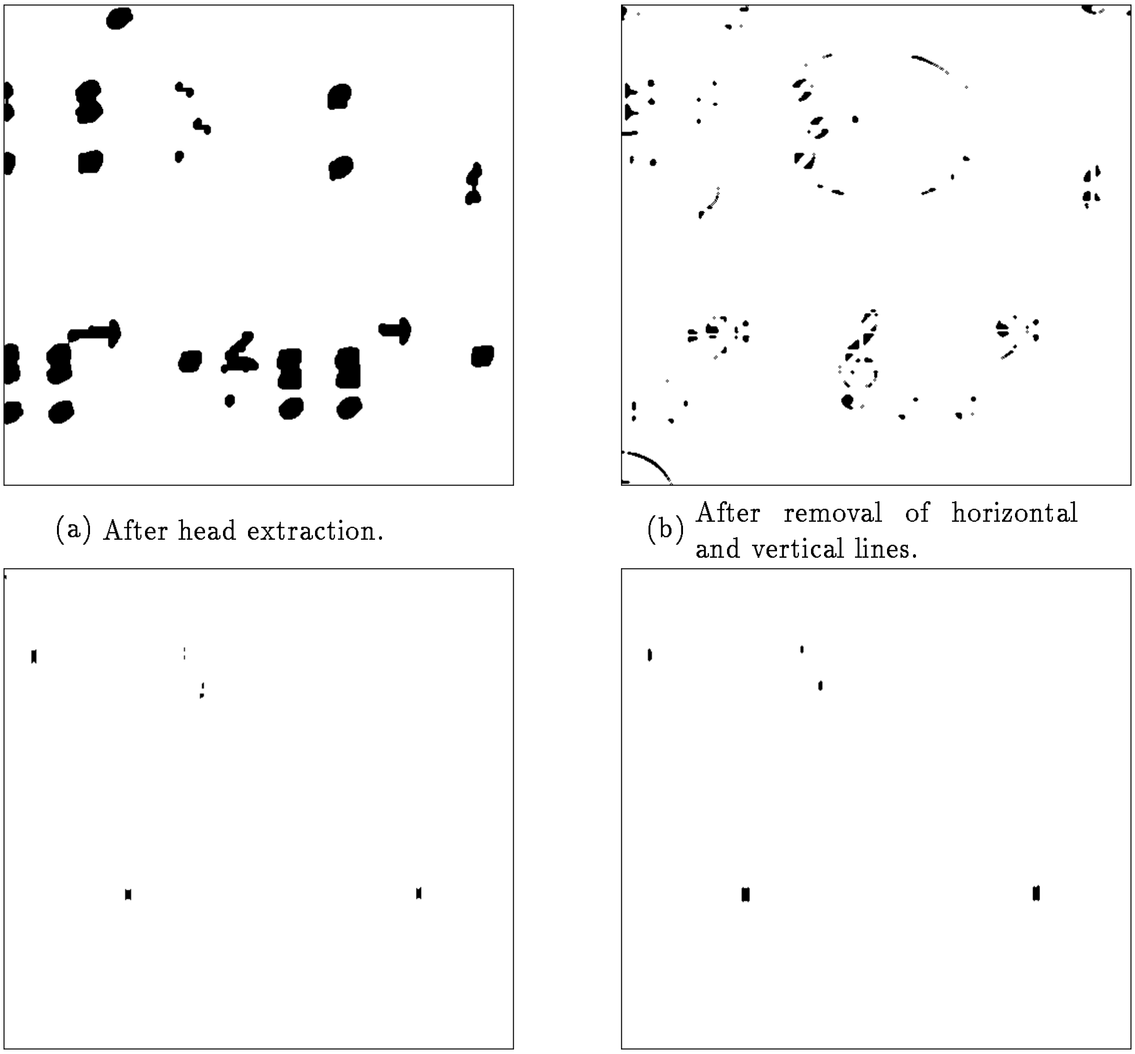
Since I_{bass} would produce output at the location of the dot pair to the right of the bass symbol, I_{bass} is dilated to the left and then down. The intention is to cover the entire bass symbol to facilitate later removal. The result of the three dilations is then removed from the input image in the usual way.

7.2 Selective Attention

An image processing operation such as a morphological dilation is computationally intensive. In its most generic implementation, the complexity of dilating an $N \times N$ image with a structuring element having S^2 elements is of the order $O(N^2S^2)$. For a 512 by 256 pixel image, dilation by a 7x7 box-shaped structuring element requires roughly six million operations! This is clearly prohibitive if a recognition system is to be implemented on a general purpose computer. The selective attention strategy seeks to reduce the computational complexity of morphological operations so that the recognition system can be easily implemented on computers with no special purpose hardware.

The symbols in a music score do not occur at random. The staff lines form the focal regions in a music score, about which the other symbols are arranged. Note stems are always associated with note heads and note beams are associated with note stems. Accidentals and note duration extenders always occur in conjunction with the notes they are modifying. Rest symbols and clefs occur at specific locations on a staff line. The idea behind selective attention is to direct the symbol recognizer to specific regions in the image in which to look for symbols. These regions are called **Regions Of Interest (ROI)**. The ROIs are produced by the high-level module by utilizing knowledge of music notation.

Fig. 6(a) shows the result of stem extraction. The original image shown in Fig. 4 is opened with a long, vertical structuring element producing the stem image. The location of the stems, along with the knowledge that note heads occur in conjunction with note stems, is used to generate the ROIs. Now, the



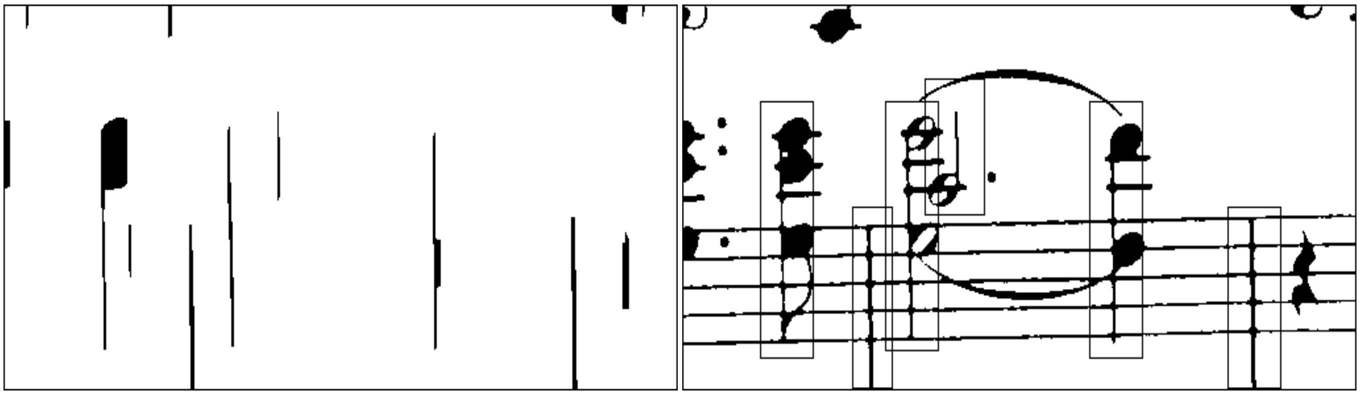
(a) After head extraction.

(b) After removal of horizontal and vertical lines.

(c) After dot pair extraction.

(d) The final result.

Figure 5: Extraction of the bass clef is illustrative of the *feature extraction-topology matching* strategy employed to detect the various symbols.



(a) Extracted Note stems. (b) Regions of Interest for notehead extraction.

Figure 6: Selective attention strategy. The location of the extracted note stems along with knowledge of music notation is used to define Regions Of Interest (ROIs) for notehead extraction.

note head extraction algorithm runs only in the regions defined by the ROIs shown in Fig. 6(b). This leads to a significant reduction in computations performed. For this particular example, the notehead extraction algorithm runs only on the ROIs shown, which together form a quarter of the original image area. The speedup is around 4.1 over an algorithm that has to run on the entire image. Since the symbol detection algorithms run only on specific regions of the image, another added benefit of the selective attention strategy is a reduction in the probability of symbols being erroneously detected.

8 High-level reasoning module

The output of the morphological symbol recognizer is in the form of binary images. Each image contains one or more symbols that fall into a particular shape category. The high-level module, which follows the symbol recognizer, is composed of two parts: a connected components analysis (CCA) module and a reasoning module. The CCA module performs connected component analysis on the images output by the symbol recognizer. For each connected component, descriptors such as centroid, area, and bounding rectangle are computed. This information and the knowledge of rules of music notation are used to perform final recognition of the symbols as well as generate regions of interest (ROIs) for other related symbols. In this section, we briefly describe the extraction of staff lines, note stems and measure bars, and association of accidentals with notes. Symbol data is represented internally in the form of linked lists. Several such lists are formed from the CCA module output, including the bar and stem list, note list, clef list, and accidental list. Subsequent processing is done by modifying/verifying the data in these linked lists.

Staff line processing. The output from the symbol recognition module is only a moderate partition of the symbols in the music score into several categories. This partition has to be refined by using prior knowledge about the elements in a music score. For example, the output image from the symbol recognizer containing the staff lines has extraneous information that needs to be filtered out. Since the detection algorithm assumes that the staff line is a long horizontal line, horizontal lines that are part of long arc ties between notes are also detected during the detection step. Moreover, a single staff line may be broken into multiple pieces because of tiny breaks in the input data.

The staff lines are identified by using the following steps:

1. Detect long horizontal lines in the input image (this is done by the morphological symbol recognizer).

2. Create a horizontal projection of this image.

3. Determine the connected components in the horizontal projection image. This is computationally efficient compared to doing connected component analysis on the original two-dimensional image. The complexity of connected component analysis of an $n \times n$ image is of order $O(n^2)$, whereas it is only $O(n)$ for the corresponding projection image.

4. The spacings between adjacent connected components are determined. This spacing gives the approximate sizes for the symbols expected in the music score.

5. A robust estimate of the spacing is obtained by taking the median of all the spacings between the staff lines.

Discriminating measure bars from note stems. Measure bars and note stems are both vertical lines. Measure bars are usually longer. For a music score that involves multiple voices, such as a piano sheet, we use the term measure bar to signify the vertical line that runs across both pairs of staffs (treble and bass). A significant difference between clefs and measure bars is due to their placement in the music score. Stems are always attached to note heads. The procedure used to remove stems from a list containing bars and stems does this by intersecting each item on the list with the bounding rectangles of elements in the note list. All items that do not have any intersection are potential measure bars. Only components that have significant area are retained as measure bars. The current implementation stops at this step, but further checks could be made, such as looking for nearby clef symbols.

Association of accidentals with notes. Accidentals such as sharp, flat, and natural symbols are associated with notes or staff lines by using the following steps. Accidentals can be associated with notes when they occur next to a note. They can also be part of a key signature, in which case they occur next to a clef symbol. Because of this we use the following steps to associate accidentals with notes:

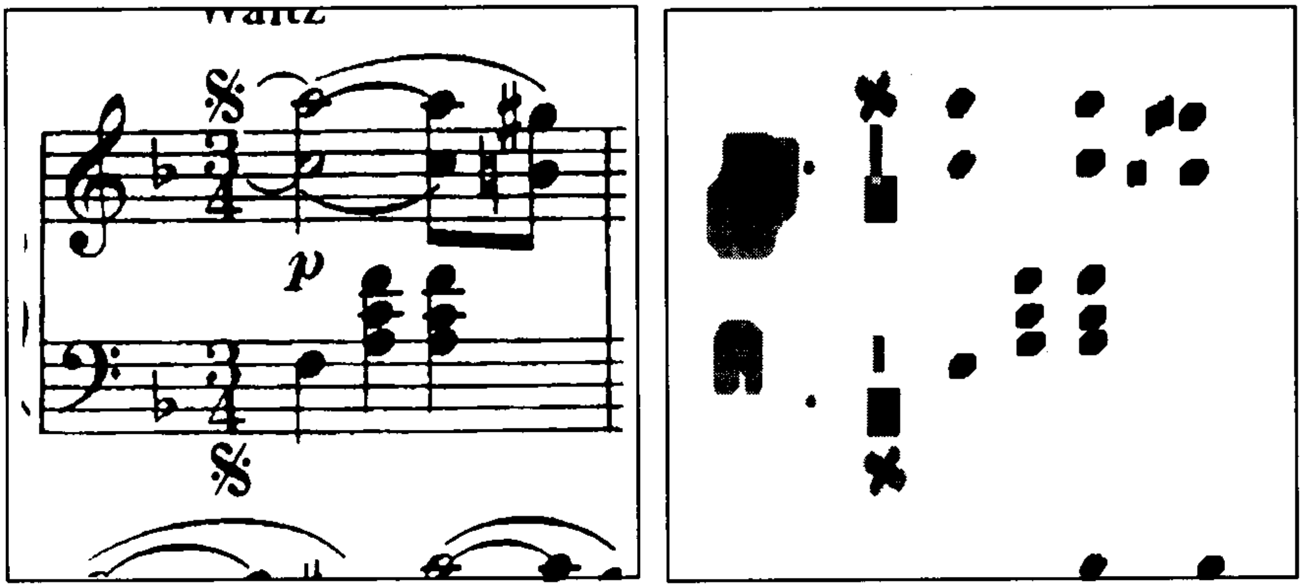
1. For each accidental in the accidental list, determine notes close to the accidental. Only notes that are in the bottom right quadrant, defined by coordinate axes with the origin at the upper-left corner of the accidental's bounding box, are potential candidates for association. This is due to the fact that the accidentals are always located to the left and above or on the same level as the notehead.

2. Compute the distance between the closest clef symbol and the accidental. If this distance is greater than the accidental-to-note-distance, then the accidental is associated with the note. If both distances are greater than a specified fraction of the measure length (number of pixels between successive measure bars), then the current implementation flags the accidental as a possible misclassification. The clef-to-accidental distance is compared with the note-to-accidental distance because the accidental could be part of the key signature or be a modifier for the note.

9 Experiments and results

The printed score sheets we worked with were digitized at 300 dpi. The recognition results reported here are based on the system implemented on a MVI – Genesis 2000 image processing workstation, which takes 2 minutes to process a 512x480 image. The results are for 74 such images containing 176 complete measures in all. There were 1311 occurrences of stock symbols (where a stock symbol is defined as a symbol that the system is designed to recognize) out of which 1252 symbols were recognized correctly. Thus the overall recognition accuracy is 95.5%. Fig. 7(a) shows the image of a sample measure and Fig. 7(b) shows the recognition result with the recognized symbols overlaid. Fig. 8 shows the misdetection and misclassification rates for the different stock symbols. The misdetection and misclassification rates are defined as follows.

$$\text{Misdetection\%} = \left(1 - \frac{\text{number of correctly identified occurrences}}{\text{Total number of occurrences}}\right) \times 100$$



(a) Original image of a sample measure. (b) Recognized symbols overlaid.

Figure 7: Recognition results for a sample measure.

$$\text{Misclassification of symbol A} = \frac{\text{number of times a non-A symbol was detected as A}}{\text{Total number of occurrences of non-A stock symbols}}$$

The misdetection was highest for the quarter rest and lowest for the quarter and eighth notes. This can be explained by the fact that the notes appeared with relatively less shape-size variation. In addition, the distinguishing features of notes are the note head, stem and the beams, all of which are stable features. For the quarter rest the bottom concavity was not a reliable feature. In a few cases, the concavity merged with the staff lines and was consequently obscured. The results can be improved if the foreground features of this symbol are included and the background features, such as the bottom concavity are excluded from the detection sequence.

There were in excess of 30 symbols in the scores we worked with, but the algorithms were designed to recognize only around 12 of those symbols (symbols such as grace notes, slurs, and note duration extension dots were not dealt with). This did not affect the recognition results at all as is evidenced by the very low misclassification rates.

Symbol type	*Occurrences*	*Correctly identified* (%)	*Misclassification* (%)
Halfnote	97	89 (91.75)	1 (0.0007)
Quarter and eighth notes	967	964 (99.7)	1 (0.0007)
Treble clef	24	23 (95.83)	0
Bass clef	23	21 (91.3)	0
Flat	63	61 (96.83)	4 (0.0031)
Sharp	13	13 (100)	0
Natural	23	22 (95.65)	0
Quarter rest	47	42 (89.36)	2 (0.0015)
Eighth rest	18	17 (94.44)	1 (0.0007)

Figure 8: Recognition results for a sample music score.

10 Conclusion

In this paper we described a system that recognizes printed music scores. The layout extraction and symbol detection modules utilize morphological operation sequences while the high-level module utilizes conventional image processing methods. For the data set used, the morphological symbol recognizer produced encouraging results, substantiating our belief that mathematical morphology could be used as a powerful tool to analyze and detect shapes. The selective attention strategy introduced in this work directs the symbol recognizer to specific regions of the image in which to look for particular symbols. This leads to: (a) a significant speedup of the recognition process, and (b) a reduction in erroneous detection of symbols. Incorporation of the selective attention strategy enables the realization of a real time recognition system on a general purpose computer. A fast, real time version of the system is already being implemented on a Pentium-based personal computer system.

References

[1] S. Baumann and A. Dengel. Transforming printed piano music into midi. *SSPR 90*, 9:532–550, July 1987.

[2] M. Costa. A practical guide to task-oriented sequences of morphological operations for use in image analysis. Technical Report EE-ISL-90-01, Department of Electrical Engineering, University of Washington, 1990.

[3] I. Fujinaga. Optical music recognition using projections. *M.S. Thesis, McGill University, Faculty of Music, Montreal, Canada*, 1988.

[4] R.M. Haralick, S.R. Sternberg, and X. Zhuang. Image analysis using mathematical morphology. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9:532–550, July 1987.

[5] B. Modayur, V. Ramesh, R.M. Haralick, and L.G. Shapiro. Muser: A prototype musical score recognition system using mathematical morphology. *Machine Vision and Applications*, 6:140–150, 1993.

[6] B.R. Modayur and R.M. Haralick. Music score recognition using mathematical morphology. In *Fifth International Conference on Symbolic and Logical Computing*, Madison, S.Dakota, April 1991.

[7] D.S. Prerau. Computer pattern recognition of standard engraved music notation. *Ph.D. dissertation, MIT, Cambridge, MA*, 1970.

[8] D.H. Pruslin. Automatic recognition of sheet music. *Sc.D. dissertation, MIT, Cambridge, MA*, 1966.

[9] G. Read. *Music Notation - A manual of modern practice.* Cresendo Publishers, second edition.

[10] J.W. Roach and J.E. Tatem. Using domain knowledge in low-level visual processing to interpret handwritten music: An experiment. *Pattern Recognition*, 21(1):33–44, 1988.

[11] C. Roemer. *The Art of Music Copying.* Roerick Music Company, Sherman Oaks, CA, Second edition.