2. REVIEW OF OMR RESEARCH

Until recently, research into OMR has been restricted to two MIT doctoral dissertations (Prusslin 1966, Prerau 1970). With the availability of inexpensive optical scanners, much research began in the 1980’s. More recent research projects have been reported in issues of *Computing in Musicology* (Hewlett and Selfridge-Field, 1987–94). An excellent historical review of OMR systems is given in Blostein and Baird (1992). Here, some of the Japanese-only papers and other research not covered in that review will be summarized. Commercial software is now available from Musitek (MIDISCAN), Grande Software (Note Scan), and Yamaha.

2.1 Aoyama and Tojo (1982)

This relatively early paper, published only in Japanese, contains many techniques that are used by more recent research in optical music recognition systems. The system is divided into three stages: input, segmentation, and recognition and syntax check. In the input stage, the image is binarized, staffline height and staffspace height are obtained, and stafflines are located. In the segmentation stage, the stafflines are removed and symbols are segmented using connected component analysis. Finally the segmented symbols are classified and verified.

The following observations about music score are made:

1) It is two-dimensional.
2) Spatial information is important.
3) Line drawing, image, and characters are mixed, and their position is not specified.
4) Because of fine lines, high resolution scanning is necessary.
5) Symbols having the same meaning may have different graphic representations, e.g.,
\[
\begin{align*}
\begin{array}{c}
\text{\includegraphics{music_note1}} \\
\text{\includegraphics{music_note2}} \\
\text{\includegraphics{music_note3}} \\
\end{array}
\end{align*}
\]
6) Symbols are placed according to spatial syntactic rules.
From the recognition viewpoint, scores contain symbols that are
1) suitable for template matching and
2) suitable for a structural analysis method.

The input score is assumed to be printed and free of broken symbols, but can be of any size (within limits) and staves may be bent or slightly broken. The system uses a 254-dpi (dots per inch) drum scanner with 8-bit gray level.

The image is scanned twice. In the first scan, groups of vertical scan lines are obtained (a figure shows nine groups across the page, each group containing a few lines separated by 1 mm). The stafflines are located as follows:
1) Binarization of the scan lines are achieved through the use of a histogram.
2) Y-projection of each group is taken, and if each group contains n lines, projection with n or n−1 pixels are considered to be staffline candidates.
3) By using the result of 2) and creating a histogram of black runs and white runs from the staffline candidates, staffspace height and staffline height are obtained.
4) The candidates for stafflines are finalized using the information obtained in 3).

In the second scan, because of the large amount of information involved, each staff is considered separately. In each staff window, the picture is vertically run-length coded (this is the direction in which the page is physically scanned in their drum scanner).

The system removes most of the stafflines, but to avoid excessive segmentation of symbols such as half-notes and flats when the stafflines are removed, the regions of the staffline left and right of the runs adjacent to the symbol are marked so as not to be deleted (see Figure 2.1). At the end, runs that straddle the staffline position and that have the staffline width are removed.

![Figure 2.1 Image after coarse segmentation (Aoyama and Tojo 1982).](image)

Next, black noteheads are searched with a template on stafflines or between stafflines, and temporarily removed if found. The black noteheads are only temporarily removed because the real goal of this section is to find holes (in flats, half noteheads, and whole noteheads). Once found these symbols can be marked so that when the rest of the...
stafflines are removed the symbols will not be fragmented. The holes are detected by the system looking for short horizontal white runs between stafflines. Once the holes are marked the black noteheads are restored, and stafflines are finally removed.

The resulting image is segmented through connected component analysis. The height and the width of the bounding box of each segment are used to coarsely separate the connected components into ten groups (see Figure 2.2). The height and width are normalized using the staffspace height.

![Figure 2.2 Coarse classification (Aoyama and Tojo 1982).](image)

In the group with flagged notes and beamed notes, flags and beams are separated from noteheads by removing thin regions (stems). Analysis of the note configuration is performed by way of features such as width, height, center of gravity, ratio of area / area of the bounding box, head count, flag count, and H-type (any of 11 head-stem configurations).

In another group of accidentals and rests, a tree classifier based on horizontal and vertical run-lengths is used to separate the members of this class. A table containing information about relative position of components is employed to recognize composite symbols (e.g. \( x \hat{8} \wedge 9 \)).
Finally, syntax rules concerning the position of symbols and the constant number of beats in a measure are used to double-check the recognition result. The spatial rules are:

1) key signatures appear after the clef symbol;
2) if there is a treble clef and key signature starts with a sharp, the sharp must be on the top staffline;
3) accidentals appear to the left of the notehead.

Although not implemented, the possibility of recognizing expressive markings (*pp, andante, a tempo*, etc.) by their character count is suggested.

### 2.2 Maenaka and Tadokuro (1983)

Maenaka and Tadokuro aimed at building a system that would be portable, compact, easy-to-use, and inexpensive. To meet these design goals, they used an 8-bit microprocessor (MC6809) and a TV camera as input device. They mention the possibility of using a facsimile machine as an alternate inexpensive input device. The overall system architecture is shown below (see Figure 2.3).

![Figure 2.3 Overall architecture (Maenaka and Tadokuro 1983).](image)

Since the maximum address space on an 8-bit processor is 64K bytes, which is not large enough to address the entire image information, a separate independent memory is used for the image. Although the memory had the capacity to store 1024(H) x 512(V) x 4-bit
of video information, the camera's hardware limitations resulted in only 416(H) x 480(v) x 2-bit subset of usable memory.

A simple memory access method is devised to access a pixel and its square neighbouring pixels so that filtering, projection calculations, and other basic pattern recognition algorithms can be performed efficiently. The TV camera is equipped with zoom lens and close-up lens is fixed on a camera stand. Three standard 100-watt lamps are used for lighting. Due to camera limitations, sheet music size of A4 format had to be divided into four sections. Adjusting the gain and the bias of the analog-to-digital converter and the lighting eliminated the need to use the histogram method or notchless binary transformation method for preprocessing. A simple fixed binary threshold method was sufficient for successful pattern recognition. Yet, because of the optical characteristics of the close-up lens, the four corners of the images were badly distorted. The paper also discusses the problem of the change in the aspect ratio during the acquisition.

The processing time of the system will be of an order of magnitude slower than if it uses a minicomputer system; hence, an effort was made to keep the processing algorithms simple and to avoid excess access to large image areas. It was decided not to implement expensive algorithms such as high-order pattern matching and spectral analysis.

The following symbols are considered a bare minimum set of music fonts and are used as recognizable objects: treble clef, bar line, double barline, repeat barline, final barline, whole note, half note, quarter note, eighth note, sixteenth note, beamed eighth and sixteenth notes, whole rest, half rest, quarter rest, flat, sharp, natural sign, and dot of prolongation.

In order to find a fragment of target object, pattern in the $i$th space of a staff, $S_i(x)$ is defined as:

$$S_i(x) = f(x, y_5(x)) + (2.5 - i \alpha),$$

where

$$f(x, y) = \begin{cases} 
0, & \text{if pixel is white;} \\
1, & \text{if pixel is black.}
\end{cases}$$

$y_5(x)$, is the position of the middle line in the y-direction (the vertical axis), and $\alpha$ is the space between stafflines.
$$P(x) \equiv \sum_{i=0}^{4} S_i(x)$$

counts the number of spaces, at x, contained in the object fragment. $P(x)$ can be used to locate a symbol but it can also be used for classification.

To track the position of the five stafflines the following algorithm is used. $B(i)$ shows the correlation against the position of the current five lines and is defined as

$$B(i) = \sum_{j=1}^{5} \sum_{k=0}^{N} f(x + j, y_5(x) + (2 - k)a + i)$$

$i = \{1, 0, -1\}.$

If $B(1) > B(0) > B(-1) \rightarrow y_5(x + 1) = y_5(x) - 1.$

If $B(-1) > B(0) > B(1) \rightarrow y_5(x + 1) = y_5(x) + 1.$

Thus the position of the middle staffline at the next position, $y_5(x + 1)$, is incremented or decremented by 1 relative to $y_5(x)$, the current position of the middle staffline.

Because a simple method usually means shorter processing time, the fixed-point sampling method and the Sonde method (counting of black-to-white transitions) are used for recognition of the objects.

The objects are first coarsely classified into three groups. At any point $x$ if $P(x) > 0$ and

$$\sum_{i=0}^{4} [S_i(x) * S_i(x + 1)] > 0$$

(to allow for noise), then the object is classified as follows:

Class A if $P(x) = 1$,

Class B if $P(x) = 2$, and

Class C if $P(x) \geq 3$.

To further classify the object, certain number of fixed regions are sampled to find any black pixels. For example, to find eighth rests, six regions are sampled. The six-bit long vector is compared with the standard pattern. If a series of tests fails, the object is considered to be a musical note and proceeds to the next stage. The size of the region for sampling is adjusted according to the size of a staffline height.
2.2.1 Classification of notes

If $P(x)$, which is a note candidate, has the value 1 or 2, it is either stem-less or has stem up (remember that $P(x)$ basically counts spaces that have black pixels in them), so that the smallest $i$ with $S_i = 1$ is chosen as the possible position. If $P(x) \geq 3$, it is considered to be a note with stem down, and thus largest $i$ with $S_i = 1$ is chosen as the possible position of the notehead.

Given $i$, there are still three possibilities for the position of the notehead: the notehead can be in the space, on the line above, or on the line below (see Figure 2.4). To precisely determine the position of the notehead, the area below and above the enclosing stafflines is traced.

![Figure 2.4 Possible position of the notehead (Maenaka and Tadokuro 1983).](image)

The existence of stems and flags can be determined by sampling fixed neighbouring regions. To distinguish between a black notehead and a white notehead, two different algorithms are used depending on whether the note is placed on the staffline or between the stafflines.

For the notehead between two stafflines, the lines equidistant from the two stafflines are scanned from left to right. If the black pixel changes to white before the notehead ends the note is considered white, otherwise it is considered black. For the notehead that is on a staffline, the area around the notehead is scanned vertically to look for black-to-white transition. This scan is performed several times at different positions along the horizontal axis. If only a very small number of vertical scans have the transition, then it is considered black; otherwise it is considered white (see Figure 2.5).
2.2.2 Classification of beams

When there is a beam, \( P(x) \geq 1 \), so that the existence of beams must be checked before proceeding with classifications for notes and rests. The vertical sums of black pixels are calculated for regions wider than the width of a notehead. If there is a sudden change in the sum, the position is noted, and \( P(x) \) is reduced by one and then passed onto one of the three classes (see Figure 2.6).
2.2.3 The output format

As real-time process was not possible and as there was no need to share the data, the output was coded in a way convenient to the sound generating device (MIDI was not yet available).

2.2.4 The experimental results and observations

The various algorithms are coded in Pascal and simulated on a computer system with the same microprocessor; thus it is estimated that it ran probably ten times slower than if everything had been coded in an assembler language and if a specialized memory access method had been used.

2.2.5 Recognition results

Because of the poor quality of the image and the noise, some of the algorithms are not as robust as expected. Also, owing to the large number of parameters involved, such as weights for the fixed sampling and beam windowing width, the correct choices were difficult to find. Further, the values had to be changed depending on the contrast level of the input image. The error rate is reported to be less than 1 error per image (1/4 of page); the accuracy can be increased by increasing the sampling points, but that also results in increase in process time. The process time for 3 measures of music containing 1 quarter note and 23 beamed eighth notes was 4 minutes and 11 seconds. In general, depending on the score, it took 4 to 10 minutes to process one line of monophonic music.

2.3 Kim, Chung, and Bien (1987)

This paper presents a complete OMR system using a TV camera as input and mechanical robot for playback. Unlike the WABOT-2 system (Matsushima 1985), this one is designed to recognize music scores with different font size under poor illumination and without special hardware. The five major processing steps are: preprocessing, coarse classification, fine classification, music syntax check, and interface to music performing device.

The music symbols recognized include: flagged and beamed notes and rests up to 16th note value, treble and bass clef, single and double bar lines, sharp, flat, natural, five
simple time signatures, and key signatures up to three accidentals. The system also makes the following assumptions:
   1) music symbols are darker than background;
   2) music symbols are randomly distributed on the staves; and
   3) the distance between two symbols is larger than a quarter of the staffspace.

In preprocessing, an input gray-image is enhanced by the 3x3 Laplacian convolution operator:

\[
H = \begin{bmatrix}
-1 & -1 & -1 \\
-1 & 12 & -1 \\
-1 & -1 & -1 \\
\end{bmatrix}
\]

to remove blurring between adjacent symbols.

The staff detection algorithm is as follows:
   1) Create histogram of average gray-level of horizontal lines.
   2) Assign threshold that maximizes the expected value of the between class variance.
   3) Label horizontal lines as staffline candidates depending on the threshold.

A gray-level input image is converted to a binary image by adaptive thresholding. At the same time each staff nucleus (staff and symbols belonging to that staff) is separated from the others.

To remove the stafflines, each point \( x \) on a staffline, is kept if the vertical neighbourhood satisfies one of three conditions: If only one pixel above is black, or if both of two pixels below are black, or if the four pixels above and four pixels below contain at least five black pixels. Otherwise, the point \( x \) is removed.

X-projection is used for symbol segmentation. Coarse classification is performed on each segmented symbol using the height and the width of the minimum bounding box after normalization by staffspace height. The symbols are classified into one of the nine groups. Four of the nine groups or regions in the height/width space (Prerau 1970) need no further processing since there is only one type of symbol within these classes. For the rest of the classes, fixed partial template matches and the Sonde method are used to finalize the classifications of the unknown symbols. Simple music syntax is invoked to check and correct rhythm and pitches of notes.
2.4 Martin and Bellissant (1991)

In the project by Martin and Bellissant (1991) a neural network is used both for staffline removal and connected component object classification.

2.4.1 Skew correction

For the skew correction of stafflines, the concept of chord is introduced. The chord of orientation $\theta$ in $P$ is the discrete line segment of slope $\theta$ inscribed in connected component $C$, where $P$ belongs to $C$ (see Figure 2.7).

![Figure 2.7 Chord of orientation $\theta$ in $P$ (Martin and Bellissant 1991).](image)

The chord length $L(P, \theta)$ is defined as the distance between the two boundary points of $C$ that intersect with the chord. In the continuous case, there would be an infinite number of chords of $\theta$ at $P$, but the number is finite in a discrete case, and if one limits $\theta$ to be $\pm$ few degrees, the number is greatly reduced.

Assuming that the whole page is skewed at some number of degrees (“less than one degree practically” [Martin & Bellissant 1991b, 418]), all points in the center column of the entire image are considered $P$ and a few values of $\theta$ are examined to find $P_0$ and $\theta_0$ so that $L(P_0, \theta_0)$ is maximized. Then rotation with $-\theta_0$ center at $P_0$ is applied to the entire image for deskewing. The chord length is calculated using an efficient line-tracing algorithm.
2.4.2 Finding and tracking the staves

Coarse approximation of the position of the staffline is derived by taking the y-projection of the entire unskewed image. This information is used to erase stafflines not overlaid by music symbols. Also, the upper and lower bounds of each staffline are computed, enabling greater accuracy in evaluating the position of the noteheads.

To erase the stafflines, each column is scanned, if a black run-length is found near the position of the y-projection histogram, has the similar width and does not belong to a symbol, then it is erased. The problem is how to determine if the black runs belong to a symbol or not. In other words, the black run has the width of the staffline but it may be part of a symbol, e.g., slurs, bass clef, etc. To solve this problem, a larger context is considered. Ideally, if the point does not belong to a symbol, there will only be one “long” chord at the horizontal, i.e., at \( \theta = 0 \). Yet in practice, due to noise and distortion, the longest chord may not actually occur at \( \theta = 0 \), so a multi-layered neural network with 228 inputs using gradient back propagation is used to recognize whether a point belongs to a symbol or not. The window used for chord calculation is 50x30 pixels centered at the center of the possible staffline (the black-run). This prevents most of the points belonging to a symbol, but also part of staffline, to be erased. The procedure also leaves some points not belonging to symbols intact. That artifact will be removed at a later stage.

Apparently, the notes are classified by some ad-hoc rule-based system using elliptical shaped template matching. The vertical and horizontal Sonde method is used to count the number of flags and beams attached to noteheads and stems. The other symbols are classified by thinning the symbols which are then processed by another neural net. After a classical thinning operation is performed, some points are marked as endpoints, junction points, and “bending” points. The minimum enclosing rectangle, which has been size-normalized, is arbitrarily partitioned into windows. A set of binary valued variables is used as input to the net. There are two classes of variables. One is of the type \((t,w)\), where \(t\) is one of end point, junction point, and bending point, and \(w\) is a window. The other type is of the form \((w_i,w_j), i \neq j\), for all \(i\) and \(j\), where \((w_i,w_j) = 1\), if at least one segment of the skeleton has one of its extremities in \(w_i\) and the other in \(w_j\), otherwise \((w_i,w_j) = 0\).
The neural net used here seems to include a decision-tree building algorithm to include specialized hidden cells that are connected only to certain input cells (features), as well as totally connected hidden cells, those that are connected to all input cells.

The authors conclude, despite the reported 96.5% recognition rate of the net, that “performance in the classification area is less impressive when compared to statistical methods; we noticed, as others before, that a nearest neighbour classifier is usually enough to reach the same recognition rate as best multi-layer perceptron.... But it should be noted that nearest neighbour can also be implemented as multi-layer automata networks” (Martin and Bellissant 1991b, 1109).

2.5 McGee and Merkeley (1991)

The subject of recognition is lined notation of chant with square neumes (see Figure 2.8). The elimination of four stafflines is performed by finding “sufficiently long” thin horizontal lines. At the same time they are straightened. Classification is performed using a set of bounding rectangles for each neume. The authors have also experimented with a “thin-line coding” method originally developed for fingerprint identification for neume classification. The input resolution is 300 dpi.

![Figure 2.8 Sample notation (McGee and Merkeley 1991).](image)

2.6 Sicard (1992)

Sicard uses a rather low-resolution 100 dpi input. The staffline detection uses a y-projection and fails if the skew is more than ±10°. The entire page seems to be rotated and stafflines are removed “using an algorithm similar to [Roach (1988)].” Different algorithms are specialized for different class of symbols: vertical run-lengths are calculated for finding thick lines (beams); vertical lines (stems and barlines) are located by using the x-projections; accidental identification involves a thinning algorithm; noteheads are localized using “edge detection, break-point extraction, and diameter
evaluation methods” (Sicard 1992, 575); and other symbols are identified using templates. Sicard reports an average 97% accuracy, where the 3% error is attributed to notehead location errors, with a process time of about three minutes per page on a Sun SPARC workstation.

2.7 RAMIT (1992)

Yadid-Pecht et al. use a neural network, named RAMIT, to recognize music symbols. The net used is a one-dimensional version of the two-dimensional Neocognitron (Fukushima and Miyake 1982). The Neocognitron is a multi-layered net that has variable connection between the cells in adjoining layers. It is shift-invariant, and selectivity to deformed pattern is adjustable. The net can learn supervised or non-supervised. RAMIT has two hidden layers in addition to the input layer, which presumably responds to each pixel. Layer 1 responds to “horizontal lines of 11x1 pixels and Layer 2 responds to three elements of Layer 1” (Yadid-Pecht et al. 1992, 128). During the preprocessing, the skew of staffline is determined, and coarse rotation of the whole page is performed. For finer adjustment, the stafflines are sheared.

2.8 Miyao et al. (1992)

The two interesting features of this system are that it incorporates a music notation grammar to aid in recognition, and that, unlike most systems, the stafflines are removed after the notes (including noteheads, stems, flag, and beams) are extracted. (The description of the research is available only in Japanese).

Three observations are made about music notation characteristics:

1) The position of the clef, key signature, and time signatures can be predicted from the position of the staff and bar lines.

2) Other symbols, including dots, ties, slurs, tenuto, accent, staccato, fermata, etc., are positioned relative to stems, barlines, and notes.

3) The size of symbols are relative to staffspace height.

The system finds the position of the staves, then notes are searched and removed. After the stafflines are removed, the remaining symbols are coarsely grouped according to their size and position, and symbols are classified by using structural features or template-matching.
A piece-wise linear Hough Transform is used to find the staffline based on the staffline and staffspace height calculated from vertical black and white run-lengths. Bar lines that span two staves are located using x-projections. The black noteheads are extracted using a rectangular mask (staffspace height x width of notehead, which is 2 x staffspace height). The position on the stafflines and another between the stafflines are scanned with the mask. White noteheads are distinguished from black noteheads by the number of white pixels in the mask area. The half note and whole note are distinguished using template matching.

Note candidates found outside of the staff are verified by searching for ledger lines. If no ledger line is found, the candidacy is revoked. Given a notehead, stems are searched by looking at the left and the right edges. If a stem is not found, the note candidacy is rescinded as well. Notational rules such as “no three stems to a notehead” are applied to make sure that recognized symbols are grammatically correct. The number of flags and beams are determined by counting the number of black runs near the stems. After removal of the stafflines, connected components are grouped, by the height, width, and relative position from the middle staffline. All measurements are normalized with staffspace height.

Coarsely grouped fixed-size symbols are further classified using 6x6 meshed templates. The symbol is divided into a 6x6 mesh and each mesh is represented by the ratio of the number of black pixels to white pixels. The thirty-six numbers are represented as a vector and compared with the vectors of prototypes using Euclidean distance measures. The unknown symbol is classified to be the same as the closest prototype above a certain threshold. Unclassified symbols are reconnected by inserting the stafflines that are removed, and then distance calculation is repeated. For size-varying symbols, such as slurs and dynamic hairpins, vertical and horizontal run-lengths are used for classification. Finally, spatial rules are used to finalize the classification decisions.

An accuracy of 93% to 98% with a processing time of 3 to 20 minutes per page using a Sony (NWS-821) workstation is reported. The input scanner has a resolution of 240 dpi.
2.9 Modyur et al. (1992)

The bi-level system described here uses morphological algorithms for symbol detection at a low-level and a high-level module that verifies the output of the low-level module and then incorporates notational syntax to aid in the spatial positioning of the symbols. The authors claim that the recognition task can be performed in near real time and achieves accuracy in excess of 95% on the sample they processed, with a peak accuracy of 99.7% for the quarter and eighth notes.

Some of the assumptions made include:

- The stafflines are equally spaced and there are five lines to a staff.
- The size of the different symbols is relative.
- The image does not have a large skew.
- The notes are proportionally spaced relative to note duration.
- Accidentals are placed directly in front for the note they alter.
- 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.
- A quarter rest is at the center of the staff.
- A half rest touches the third line above, while whole rest touches the fourth staffline below.

To locate stafflines, the image is opened with a 35-pixel wide horizontal line, but the stafflines are not removed. 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 done to incorporate a certain degree of tolerance in the detection process. Thus, a few missing foreground pixels, broken edges, blurred corners, etc., do not affect the output of the symbol detection process.

The system is able to recognize twelve symbols: treble and bass clefs, sharp, flat, whole notehead, half notehead, quarter notehead, eighth rest, quarter rest, stem, beam, and half-beam. The system runs on an MVI-Genesis 2000 image processing workstation and takes 2 minutes to process a 512x480 image.
2.10 Kobayakawa (1993)

A very efficient recognition system (10 seconds per page) is described. This is achieved by actively searching for common music symbols. The system consists of Sun SPARC 2 and Omron Luna workstations, the latter being connected to a 200 dpi scanner and a Yamaha DX7 MIDI synthesizer.

To locate the stafflines, thirty-two vertical lines spread across the page are scanned for black runs. Any runs whose length is less than the median of the black run lengths are considered as a candidate for a staffline. For each of these candidates, the image is scanned horizontally and if a horizontal line is found to cover 70% of the score width then that line is considered a staffline. These stafflines are removed if there is a white pixel a certain distance above and below the center of the stafflines.

To locate the black noteheads, the image is scanned horizontally for black runs at staffline positions and center point between the stafflines. Two maxima are found from the histogram of these run lengths. The maximum with few pixels (“about 2 pixels”) are considered to come from vertical line segments (stems and barlines) and the second peak (“about 15–18 dots”) is assumed to be contributed by black noteheads. In the rhombic (diamond-shape) region around the center of the longer runs, the number of black pixels is counted. If the count is greater than 95% of the region then it is considered to be a black notehead.

The sharp and the natural signs are distinguished from the noteheads by determining that the distance between two nearby vertical line segments are close together. The barlines are separated from other vertical line segments because of their height being the same as the height of the staff or longer. If these barlines are close together they are considered double barlines, in which case, two small dots indicating repeat signs are sought. The remaining vertical lines are considered stems if they are close to a notehead or if there are noteheads between the two endpoints of the line segment.

After the stems are removed, the side opposite to the noteheads is scanned in the vertical direction to look for flags or beams. If any black pixels are found, a connected component is assembled. If the width of the component is less than twice the width of the notehead and the slant (presumably the angle of the line connecting midpoints of the left and the right edges of the component) is steep, then it is considered a flag.
The remaining symbols are recognized using template-matching. These templates are prepared from various example scores, edited with a bit-map editor, then encoded in run-length format. The reported recognition rates are:

- *Scenes from Childhood*, op. 15/6 (Schumann): 99.6%
- *Fantasie-Impromptu*, op. 66 (Chopin): 98.3%
- *Turkish March* (Mozart): 94.8%

### 2.11 Roth (1994)

The system consists of the following seven steps.

#### 2.11.1 Rotation

To correct skews, the image is rotated by shearing horizontally and vertically. The actual amount of shearing is determined manually.

#### 2.11.2 Vertical run-length statistics

The median lengths of vertical runs of black and white pixels are used to estimate the staffline height (from black runs) and the staffspace height (from white runs). The size of all the staves on a page is assumed to be the same.

#### 2.11.3 Locate and delete stafflines

The stafflines are located by searching for groups of five peaks in the y-projection, then they are tracked from the middle outwards to get accurate y-position in each image column. This operation corrects slightly skewed or bent stafflines. Once located, the stafflines are deleted from the image. In order not to affect symbols too much, lines are deleted only when their width is close to the overall staffline height.

#### 2.11.4 Locate and delete vertical lines

By examining the x-projections of each staff, vertical lines are located. This task is refined later through application of the technique of mathematical morphology. Note that any vertical line segments (thin objects) are removed, which include stems, bar lines, and lines within sharps, flats, and naturals.
2.11.5 Connected component labeling

The remaining components are identified. A list of components and references from each pixel to the component it belongs is created. “A fixed space above and below the staff is included in the region of interest, the total height of the region is three times the staff height. This allows for recognition of up to four ledger lines. For this region connected components are derived” (Martin 1994, 18).

2.11.6 Symbol recognition

Before symbols are classified, “separated white notehead (due to staffline removal) are merged and connected black noteheads (due to chords) are separated using heuristics” (Roth 1994, 19). In addition, he employs a fairly complex decision tree to classify various music symbols using the following features: height, width, area, and center of gravity. The location with respect to other components, vertical lines, and stafflines is also taken into consideration.

2.11.7 Lipsia document generation

Finally the recognized element is reproduced using the Lipsia music notation editor. Preliminary but successful use of mathematical morphology operators is also reported.

2.12 Summary

Although many innovative OMR systems have been developed over the last decade, there are major limitations to be used as practical OMR. As mentioned, the number of different music symbols commonly used exceeds four hundred, yet, most of the available programs can recognize no more than a few dozen symbols. This is a serious limitation because these program are not designed to learn new symbols. The lack of learning capability also limits the recognition of handwritten music as well. The automatic recognition of well-formed handwritten music will be extremely useful for musicians. The AOMR described here overcomes these limitations by incorporating a flexible learning mechanism thus enabling it to recognize virtually unlimited number of music symbols, including handwritten manuscripts.