Control and Cybernetics

vol. **25** (1996) No. 2

Automatic recognition of printed music and its conversion into playable music data

by

Władysław Homenda

Centro de Investigacion Cientifica y
de Education Superior de Ensenada,
Ensenada, Mexico 22860,

On leave from: Institute of Mathematics,
Warsaw University of Technology, Warsaw, Poland
e-mails: homenda@cicese.mx, homenda@alpha.im.pw.edu.pl

Abstract: The paper describes MIDISCAN – a recognition system for printed music notation. Music notation recognition is a challenging problem in both fields: pattern recognition and knowledge representation. Music notation symbols, though well characterized by their features, are arranged in an elaborate way in real music notation, which makes recognition task very difficult and still open for new ideas, as for example, fuzzy set application in skew correction and stave location. On the other hand, the aim of the system, i.e. conversion of acquired printed music into playable MIDI format requires special representation of music data. The problems of pattern recognition and knowledge representation in context of music processing are discussed in this paper.

Keywords: music notation recognition, knowledge representation, music representation, MIDI format.

1. Introduction

Computer applications in music editors and recognizers seem to have common tendency with the development of text editors and OCR systems. Computer applications in both fields started from storing information, editing it and printing information on the paper in the form of text or music notation respectively, and, on the other hand, computer programs have been built to capture information from the paper and apply it to certain purposes. Nevertheless, there is no more analogy in these fields. Despite this fact, the analogy between both fields: text and printed music processing seems to be appropriate due to the contrast between them emphasizing difficulties in music notation processing

The main difference between these two fields is related to the way symbols are arranged in. It is considerably simpler for a text, and much more complex for a music notation. As to text processing, there are about 100 symbols of the same or similar shapes (letters, digits, special signs) and a text is nothing more than the sequence of symbols. Even if parts of a text differ in fonts and vary in fonts sizes, the text as a whole still can be easily represented in an electronic form, stored in memory, processed, etc. After many years of development, there are many text editors applied to different fields, beginning from simple programs used in everyday life, or more complex office-like editors, ending up with sophisticated desktop systems used in publishing.

On the other hand, music notation editors are not so widely used. The main reason is that communication between people is mostly done in the form of printed text rather than printed music, and so software developers are much more interested in building text editors rather than music editors.

Moreover, it is much more difficult to develop music than text editor: though the numbers of symbols used in music and text editors are similar, music symbols varying in size, shape and, what makes the biggest difference, are arranged in much more complex and confusing way. In fact, music notation is a two-dimensional language in which importance of geometrical relations between its symbols may be compared to the importance of the symbols alone.

This analogy between text and music notation may give an idea as to how complicated music notation editing is in comparison with text editing.

The situation is even more difficult when opposite flow of information is considered, i.e. automated recognition of printed text and printed music. There are several applicable computer systems for automated text recognition with considerably high rate of recognition (which can be calculated as ratio of recognized characters to all characters in the text). As to music notation recognition, commercial systems are still very rare despite the fact that several research systems of music notation recognition have already been developed, see e.g. Blostein, Baird (1992), Fujinaga (1988), Itagaki (1992), Kato, Inokuchi (1992). The main problem lies in difficulties the developers of such a system face. First of all, music notation does not have a universal definition. Although attempts to codify printing standard for music notation have been undertaken, see e.g. Ross (1970), Stone (1980), in practice composers and publishers feel free to adopt different rules and invent new ones. Though most of scores keep the standard, they still can vary in details. Moreover, music notation, as a subject of human creative activity, constantly develops and will probably be unrestricted by any codified set of rules. Thus, it may not be possible to built a universal recognition system accepting all dialects of printed music notation. Furthermore, the nature and structure of music, even that printed one, is much more complicated than the structure of a text, so representation of music is comparably much more difficult than representation of printed text.

The problem of measuring of recognition efficiency may be considered as an example of difficulties in music representation and processing. Unlike in

printed text, there is no obvious method for calculating the recognition rate. In the course of recognition, specific features of object of music notation may be mistaken while other ones are recognized properly. Some of these features may be important from specific point of view while others are not. For example: it is of lesser significance from performance point of view if the eighth note is flagged or beamed though this feature is important for music notation editors. Similarly, it is of lesser significance for editors if accidentals are associated with proper notes or are linked into group creating key signature, but is extremely important to playback program to have accidental associated with the proper note or have all accidentals creating key signature linked together.

2. MIDISCAN – overview of the system

The transformation of data given as printed music into playable MIDI format is the main idea of MIDISCAN software. This transformation is intended to be as far automated as possible. Unfortunately, at the present stage of development of both fields: methods of recognition of printed music notation and representation of music data, it is impossible to built fully automated system which could recognize music notation and create playable music data correctly performed with the electronic instrument. Errors appearing in recognition process, even a few of them, cause that correction of recognized music notation is necessary before it is performed with the instrument. Thus, a correction of acquired music data is necessary. The correction may be done at the output, playable data (e.g. MIDI format) or in the middle of the road: before acquired data are converted to playable format.

The first option, final MIDI format correction is regarded as to be less convenient than correction of recognized data before MIDI conversion. The reason is quite clear, for example, key signature correction needs only a few operations before MIDI conversion while pitches of many notes should be corrected if wrong key signature was assumed in MIDI conversion. But this assumption requires the music notation to be represented in a format allowing for editing, correction and, then, MIDI conversion.

Having all these problems in mind, the correction of recognized music notation before conversion to MIDI was assumed. This assumption implied the necessity for the acquired data to be represented in the special intermediate format called MNOD (Music Notation Object Description).

The idea of MIDISCAN is outlined in Figure 1.

3. The structure of recognition module

In this chapter the structure of basic part of MIDISCAN, i.e. the structure of recognition process, is presented. TIFF files (Tagged Image Format File) representing scanned sheets of music notation (the score) are accepted as input data.

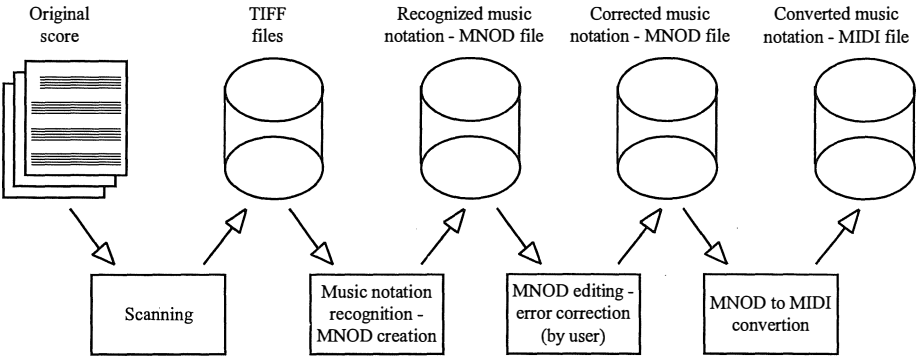

Figure 1. The outline for the MIDISCAN system

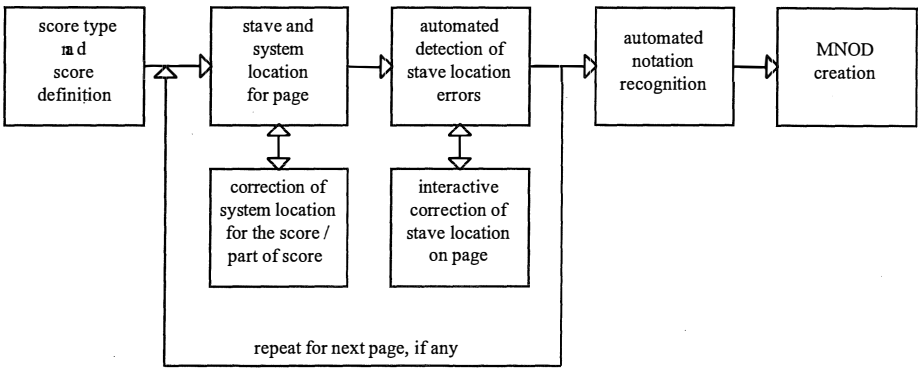

Figure 2. Structure of the recognition process

Of course, adaptation of the program to any graphical input data constitutes only a technical detail.

Once the score is scanned and stored on a hard disk as a set of TIFF files, MIDISCAN can start its task, i.e. recognition of music notation, writing acquired data as a MNOD file format, editing MNOD format and presenting it to the user for corrections (if necessary), conversion of MNOD file format to MIDI file format. The structure of the recognition process is presented in Figure 2.

3.1. Score definition

Two types of music notation may be processed by MIDISCAN: ensemble and part scores. MIDISCAN does not detect the type of a processed score, it must be defined by user. In case of ensemble score, i.e. score with all voices linked

together into systems, user only chooses the score type and defines the sequence of pages of the score (i.e. pages of the original notation scanned and stored in the form of TIFF files). When part score is processed, i.e. score with voices separated from each other, the number of voices and the number of pages for every voice must be described before the sequence of TIFF files is defined. The sequence of pages must keep the order of respective pages of the score. TIFF files have to be placed on hard disk, but it is not necessary to place them in any specific directory, they may be distributed anywhere on the disk.

3.2. System location

Automated stave location is performed for the whole score before recognition process is started. Simultaneously, the structure of the score is detected, i.e. the way systems are located. The number of staves in every system is detected for the whole score of ensemble type and for every part of the score of part type. Let us recall that the term "system" is used here in the meaning of:

- all staves performed simultaneously and
- joined together in the score or part of the score.

Some restrictions are assumed as to the score structure. For ensemble type of a score, the structure must be fully determined only on the basis of the first and second page. All systems must include constant number of staves or must start from smaller number of staves in the beginning systems, and then go to the regular number of staves for the rest of the ensemble type of a shore. For part type of a score, the structure of a part must be resolved on the basis of the first page of every part.

The score for voice and piano with three staves in the system is an example of ensemble type of score. If voice part is missing in the first and second system, the score has two irregular systems with two staves and the other systems are regular with three staves.

Part type score: only scores with constant number of staves in systems for every part are accepted. Up to 3 staves per system are permitted (e.g. organ part of the score consists of 3 staves, other voices consist of no more staves). A score for string quartet with separated voices is an example of part type score with four parts and one stave in part system for every part.

The above restrictions are justified as most of scores satify them. However, there is a possibility to process scores, which do not satify these restriction. Roughly speaking, this possibility is based on insertion of missing staves as empty ones or skipping of existing staves in order to prepare the score structure to be compatible with program restrictions.

3.3. Stave location

Stave location algorithms are based on horizontal projections. Theoretically, for non-distorted and non-skewed images, methods based on projections should be

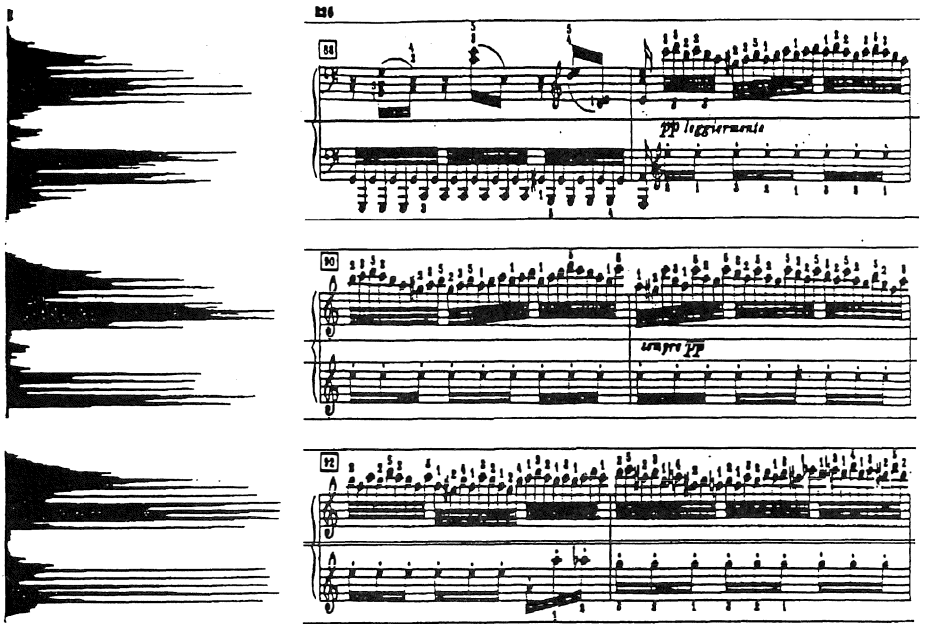

Figure 3. Stave location example for a low quality score

fully effective: high pass filtering gives clear image of a stave as five equidistant picks with the height equal to the length of the stave.

Unfortunately, in real images staff lines are distorted too much to give so clear projections, especially when projection is done for page width or even for wide region. Scanned image of a sheet of music is often skewed, staff line thickness differs for different lines and different parts of stave, staff lines are not equidistant and are often curved, especially in both endings of the stave, for ensemble type of score staves may have different sizes, etc. These problems cause that projections done in wide region are useless for stave location, see Figure 3 (taken from Fujinaga, 1988).

On the other hand, projections in narrow region are distorted by notation objects such as ledger lines, dynamic 'hairpin' markings, slurs, note heads, etc. Thus, simple filtering does not give information sufficient for stave location. In MIDISCAN program, horizontal projections in several narrow regions are analyzed for obtaining vertical placement of the staves. Once vertical placement of the stave is located, both endings of it are detected. Projections in relatively narrow regions are used in stave endings location task. Iterative process based on classical bisection algorithm is employed in this process – the process starts in the middle of the stave and then goes in the directions of both endings of the stave.

An advantage of applied methods is that distortions such as non-equidistant staff lines, varying thickness of staff lines, skewed images (skew angle up to 10-15 degrees) and stave curvature, do not influence the process of stave location as well as the process of notation recognition in an observable way.

It is worth mentioning that stave location process is not fully automatic. In some cases, especially for low quality images or for very dense notations, automatic stave location gets mistaken and must be corrected manually. Fortunately, the program is able to detect problems it has and only if automated correction of given problem is not possible, location process is suspended until the user fixes the problem.

Feasibility study resulted in skew correction algorithm for skew angle up to 35-40 degree. The skew angle detection was the basic task of that algorithm. The algorithm was based on the analysis of several narrow projections on the OY axis. The main problem in that algorithm was related to necessity of choosing projection region narrow enough to avoid distortions coming from a big skew. Information, acquired from high-pass filtering of the projections in narrow region, included image of the stave (five equidistant picks). Unfortunately, that image was heavily distorted by elements of notation, so it was necessary to apply special analysis to find the skew of a stave. This analysis was based on methods of the algebraic extension of fuzzy set theory primarily discussed in Homenda (1991) and Homenda, Pedrycz (1991). Once skew angle was calculated, the rotated coordinate system was used that gave non-skewed stave, so it might be assumed that further processing was done for non-skewed image.

The methods of skew correction, though interesting from theoretical and research points of view, were computationally more expensive and conceptionaly more complicated. Moreover, the methods applied for recognition were skew insensitive for practical images. Because of it the skew correction was not applied in the final product.

3.4. Recognition

Music notation is built around staves. The position and size of symbols are restricted and determined by the stave. So, location and identification of staves must be the first stage of recognition process. Having staves and systems located, the program starts fully automated recognition of music. Recognition is done for every stave, and then, after notation is recognized and analyzed for given stave, the acquired music data are filed into the MNOD format.

Recognition strategy can be seen as the three-step process:

- object location,
- feature extraction
- classification.

The first step – object location – is aimed at preparing a list of located symbols of music notation. Bounding boxes embodying symbols to be recognized are defined for located symbols. The process of object location is based on

projection analysis. First, the projection of whole region of given stave on the OX axis is processed. The location process is mainly based on the analysis of a derivative the projection, see Fujinaga (1988). The derivative analysis gives the first approximation of object location. Then, for every rough located object, a projection on OY axis is analyzed to obtain vertical location of the object and to improve its horizontal location. The most important difficulties are related to objects which cannot be separated by horizontal and vertical projections. Also wide objects as slurs, dynamic 'hairpin' signs, etc. are hardly located.

The next two steps of recognition process are based on the list of located objects. Both steps: feature extraction and classification overlap each other and it is not possible to separate them. Feature extraction starts from extracting the simplest and most obvious features as heighth and width of the bounding box containing given object. Examination of such simple features allows classification to be made only in a few cases, see Figure 4 (taken from Fujinaga, 1988). In most cases additional features must be extracted and context analysis must be done. The extraction of features is based on filtering of projections in the bounding box, analysis of chosen columns and rows of pixels, etc. Several classification methods are applied for final classification of object including context analysis, decision trees, and syntactical methods.

Only limited set of music notation objects can be processed in MIDISCAN. This set includes *notes, chords, rests, accidentals, clefs, bar lines, ties, key signatures, time signatures, change of key and time signature.* Rhythmic grouping can also be inserted into acquired music data, though they are not recognized. Other symbols are going to be recognized and represented in the future versions of the program.

Recognition confidence depends on many features of a printed score: font, printing quality, image quality, notation density, etc. The obvious, general rule may be formulated that the higher quality of printed music and scanned image, the higher the rate of recognition. In Figures 3, 5 and 6 scores of low, medium and high quality are presented respectively.

Recognition efficiency of MIDISCAN program may be estimated as 95% for good quality of image and 80-85% for low quality of image. Precise calculation of recognition rate is strictly related to the applied calculation method (see note in Section 1), but the scope of the paper does not allow to discuss extensively this problem.

4. Music representation

Making one more analogy between computer processing of a text and a printed music, it is worth underlining that, as to text processing, design and implementation of widely accepted data representation is considerably easy. Rich Text Format (RTF) format is an example of such a representation.

Music data representation is far more difficult and, up to now, there is no universal representation widely used and commonly accepted. Music data for-

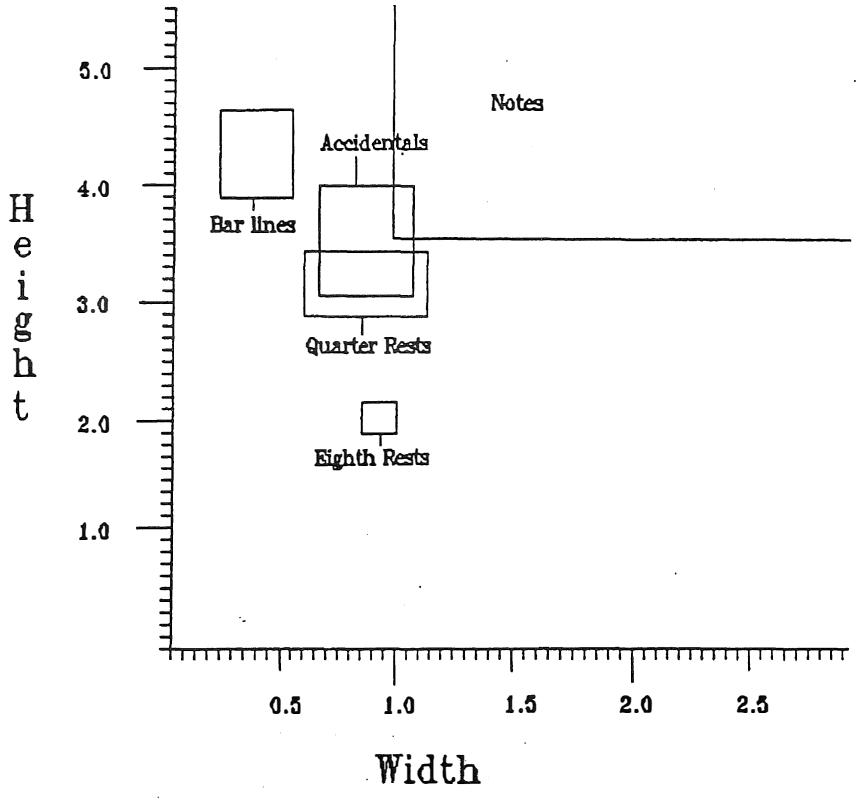

Figure 4. Classification of musical symbols in Width-Height space (see Fujinaga, 1988).

mats used in computer systems are intended more for particular tasks rather than for common use. Even if a particular format is widely spread and commonly applied, it is used for special tasks rather than for any purpose whatsoever. For example, "MIDI (Music Instrument Digital Interface) data format was established as a hardware and software specification which would make it possible to exchange information between different musical instruments or other devices such as sequencers, computers, lighting controllers, mixers, etc. This ability to transmit and receive data was originally conceived for live performances, although subsequent developments have had enormous impact in recording studios, audio and video production, and composition environments" (MIDI, 1990). Nevertheless, MIDI format, as performance oriented, is not a universal one. Thus, for instance, it is very difficult or even impossible to represent in MIDI format graphical features related to music notation. On the other hand, format used by notation programs are notation oriented and cannot be easily used as a universal format for music representation.

Several attempts have been made lately in order to define universal format of music notation, Dannenberg (1993), Field-Richards (1993), Haken, Blostein (1993). However, there is still lack of the commonly accepted universal format of music notation.

Because recognition confidence in MIDISCAN is not satisfactory enough for direct conversion of recognized music notation into MIDI format, it is necessary to edit acquired data, correct it and then convert into MIDI format. These tasks need acquired data to be stored in some form suitable for both editing and conversion. For this particular aim, a special format was developed. This format is called MNOD format (Music Notation Object Description). It plays the role of an intermediate format between printed music notation and playable MIDI format.

It was assumed that recognized music notation would be edited for correction in the form compatible with original score. This assumption allows for simultaneous displaying of original score and acquired data. It makes editing and checking correctness of acquired data as easy as comparing two notations which should be identical. All differences can be easily corrected, even if user is not familiar with music and music notation.

MNOD format applied in MIDISCAN for data representation meets all these requirements. Its main features give the possibility of data interpretation form both perspectives: notation oriented and performance oriented, which makes music data easily accessible for both purposes: editing and MIDI conversion. Unfortunately, MNOD format applied in MIDISCAN does not represent all commonly used notational symbols. Only symbols edited in MNOD editor (see section 3.4 for the list of processable notation objects) are represented in the format.

MNOD format is structured hierarchically. The levels in this hierarchy reflect data accessibility for the above purposes. MNOD format may be regarded as two different structures permeating each other.

The notation oriented structure may be seen in the following levels:

- score,
- page,
- stave on page,
- objects of music notation

while performance oriented structured may be outlined as below:

- score,
- part of score (applicable to the scores of part type),
- system,
- stave in system,
- vertical events,
- objects of performed music.

This comparison gives only general view on differences between those attempts. Extended discussion on this topic is out of the scope of this paper.

It is worth mentioning that levels of both structures differ in their meaning even if they are called similarly. E.g., notation structure reflects sequential organization of staves on page while performance structure organizes staves according to the systems of the score, regardless of their order on the page. Similarly, objects of music notation are seen differently in both structures. For example, in notation structure, notes must have such features as their position on the page, while their relative position to each other is unimportant. On the other hand, relative placement of notes is significant for performance structure, while their position on the page is of less importance.

The approach to music representation applied in MIDISCAN is flexible and easy to control: displaying data in graphical form on the screen, converting music data to MIDI format and independent music data processing.

Acquiring contextual information from recognized music notation and checking correctness of recognized notation is the aim of independent data processing applied in the program. This processing allows for locating of pickup/close-out measures, analyzing voice lines, verifying bar lines or change of key and time signature consistency, monitoring data integrity. In general, the possibility of independent music data processing considered in wider context creates a lot of research problems related to knowledge representation, which makes music representation interesting from a more general point of view.

5. Experimental results

MIDISCAN was developed and ported on PC 386 and compatible computers in WINDOWS environment. Hardware requirements are similar as for WINDOWS environment. Digitizing resolution of 300 dpi is suggested for acquired binary images of the A4 size. It gives TIFF file of the size approximately equal to 1MB. Both orientations of a page: portrait and landscape are accepted. Pages/files of bigger size can be effectively processed on computers with at least 8 MB RAM.

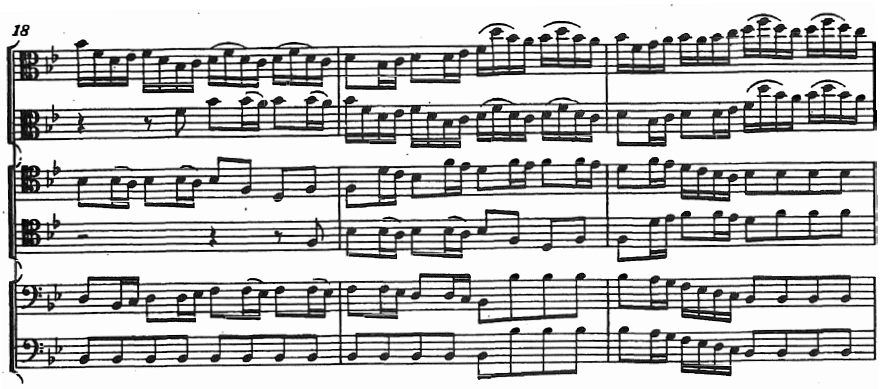

Figure 5. An example of a medium quality score (J.S. Bach, Brandenburg Concerto No. 6)

Processing time, which is different for different notations, depends on resolution of the image and on notation density.

For example, it took about 75 minutes to recognize first movement of Bach's Brandenburg Concerto no. 6. The experiment was done on PC 386 / 8 MB RAM, 33 MHz clock. The score consisted of 15 pages of A4 format, 18 staves per page, 6 staves per system, scanned at 300 dpi resolution. The printing quality of the score as well as music notation density were considered as medium. In Figure 5 an excerpt of this score is presented. The recognition efficiency for this score was considerably high. Three staves were incorrectly localized and, what was important, system detected all three errors and signalized them to the user. Estimated recognition rate exceeded 90%. It was calculated as the ratio of missing or mistaken objects or their features, to all objects.

Another experiment done for Beethoven's "Für Elise" gave similar result in speed processing (i.e. comparable recognition time per stave), but the rate of recognition was higher due to higher printing quality and lower density of tested score, see Figure 6 for an excerpt of this score. All 36 staves of this score were located accurately, score structure was also correctly detected, estimated recognition rate was over 95%.

More experiments done for different scores confirmed that context information implied form music notation, such as pickup and close-out measures and voice lines analysis and location, was correctly and comparably easily acquired.

MNOD to MIDI conversion was very fast and took about one minute for Bach's score and a few seconds for that of Beethoven's.

To compare similar parameters of other recognition systems see Blostein, Baird (1992), Itagaki (1992), Kato, Inokuchi (1992).

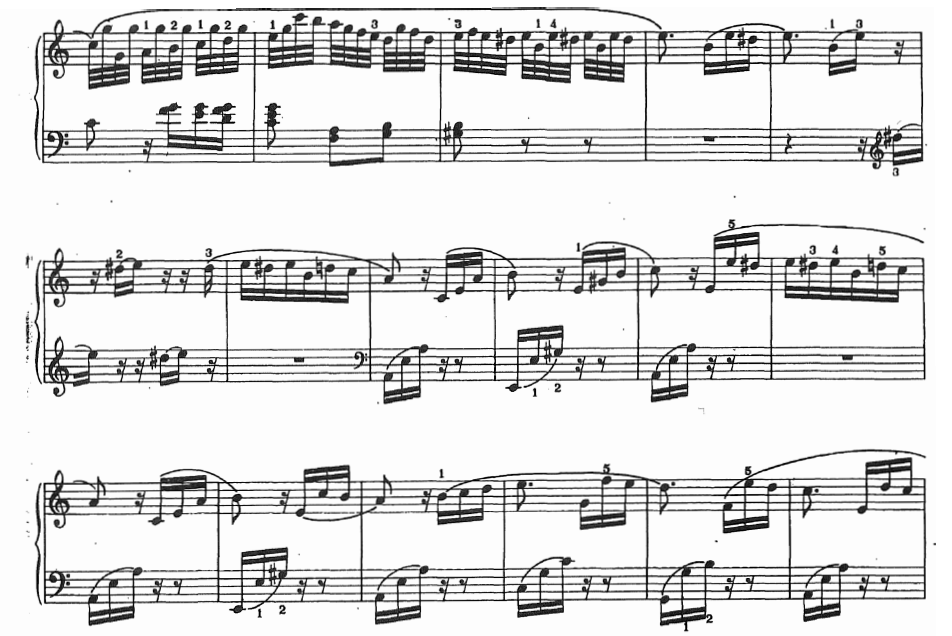

Figure 6. An example of high quality score (L. van Beethoven, *Fuer Elise*)

6. Conclusions

The paper describes MIDISCAN – a recognition system for printed music notation. Music notation recognition is a challenging problem in both fields: pattern recognition and knowledge representation. Music notation symbols, though simple and well characterized by their features, are arranged in sophisticated and confusing way in real music notation, which makes recognition task highly difficult and still open for new ideas, as for example, fuzzy sets application in skew correction and stave location. On the other hand, the aim of the system: conversion of acquired printed music into playable MIDI format, requires special representation of music data. This representation should be adequate for both: source – notation oriented, and target – performance oriented music data. Regarding further development of this system, the effort should be put on following tasks: improving recognition methods, extending class of recognized objects, improving and extending music representation format.

See also Aikin (1994), Homere (1994), Lindstrom (1994) for reviews of MIDISCAN.

Acknowledgments

The author wishes to thank Ewa Pawelec for development and implementation of recognition methods and Robert Zalewski for implementation of user interface of the program. Likewise, the technical and administrative support from ZH Computer Inc., Minneapolis, Minnesota and CPZH Inc., Warsaw, Poland is cordially acknowledged. Special thanks to Christopher Newell, who brought the idea of MIDISCAN and supported its development.

References

AIKIN, J. (1994) Musitek Midiscan: Sheet music recognition software. *Keyboard*, March 1994, 136-140.

BLOSTEIN, D., BAIRD, H.S. (1992) A critical survey of music image analysis. In: *Structured Document Analysis*, H.S. Baird, H. Bunke, K. Yamamoto (eds), Springer-Verlag, 405-434.

Computing in Musicology (1994) **9**, 1993-1994, 109-166.

DANNENBERG, R. (1993) Music representation issues, techniques, and systems. *Computer Music Journal*, **17**, 3, 20-30.

FIELD-RICHARDS, H.S. (1993) Cadenza: A music description language. *Computer Music Journal*, **17**, 4, 60-72.

FUJINAGA, I. (1988) *Optical music recognition using projections.* Master's thesis, McGill University, Montreal, Canada, September 1988.

HAKEN, L., BLOSTEIN, D. (1993) The Tilia music representation: extensibility, abstraction, and notation contexts for the lime music editor. *Computer Music Journal*, **17**, 3, 43-58.

HOMENDA, W. (1991) Fuzzy relational equations with cumulative composition operator as a form of fuzzy reasoning. *Proc. of the Int. Fuzzy Eng. Symp. '91*, Yokohama, November 1991, 277-285.

HOMENDA, W., PEDRYCZ, W. (1991) Processing of uncertain information in linear space of fuzzy sets. *Fuzzy Sets & Systems*, 44, 187-198.

HOMERE, S. (1994) Musitek Midiscan: Donnez la parole a vos partitions. *Keyboards Magazine.*

ITAGAKI, T., et al. (1992) Automatic recognition of several types of music notation. In: *Structured Document Analysis*, H.S. Baird, H. Bunke, K. Yamamoto (eds), Springer-Verlag, 466-476.

KATO, H., INOKUCHI, S. (1992) A recognition system for printed piano music using musical knowledge and constraints. In: *Structured Document Analysis*, H.S. Baird, H. Bunke, K. Yamamoto (eds), Springer-Verlag, 435-455.

LINDSTROM, B. (1994) Musitek Midiscan: optical music recognition is finally a reality. *Electronic Musician*, February 1994, 151-154.

MIDI 1.0, Detailed Specification (1990) Document version 4.1.1, February 1990.

ROSS, T. (1970) *The Art of Music Engraving and Processing.* Hansen Books, Miami.

STONE, K. (1980) *Music Notation in Twentieth Century: A Practical Guidebook.* W.W.Norton & Co., New York.