[Wit73] G. Wittlich. Project SCORE. *Computational Musicology Newsletter*, 1(1):6, 1973. Abstract of paper from *International Conference on Computers in the Humanities*, University of Minnesota, 1973.

[Øst88] B. Østenstad. Oppdeling av objektene i et digitalt notebild i klassifiserbare enheter. Rapport 31, Bildebehandlingslaboratoriet, Institutt for informatikk, Universitetet i Oslo, Oslo, NORWAY, October 1988. In Norwegian; 57 pages; ISBN 82-7476-034-4.

[Pen90] B. Pennycook. Towards advanced optical music recognition. *Advanced Imaging*, April 1990. 3-page magazine article.

[Pre70] D. S. Prerau. *Computer Pattern Recognition of Standard Engraved Music Notation.* PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts USA, September 1970. Thesis supervisor: Murray Eden; 240 pages.

[Pre71] D. S. Prerau. Computer pattern recognition of printed music. In *Proceedings, Fall Joint Computer Conference*, volume 39, pages 153–162, Montvale, NJ, November 1971. A.F.I.P.S. Press.

[Pre75] D. S. Prerau. Do-re-mi: A program that recognizes music notation. *Computers and the Humanities*, 9:25–29, 1975.

[Pru66] D. H. Pruslin. *Automatic Recognition of Sheet Music.* PhD thesis, Massachusetts Institite of Technology, June 1966. Sc.D. dissertation; thesis supervisor: Murray Eden; 94 pages.

[Rea79] G. Read. *Music Notation: A Manual of Modern Practice (2nd Edition).* Taplinger Publishing, New York, NY, 1979.

[Roa86] C. Roads. The Tsukuba musical robot. *Computer Music Journal*, 10(2):39–43, Summer 1986.

[Ros70] T. Ross. *The Art of Music Engraving and Processing (2nd Edition).* Hansen Books, Miami, FL, 1970.

[Rou88] D. Roush. Music formatting guidelines. Technical Report OSU-3/88-TR10, Department of Computer and Information Science, The Ohio State University, 1988.

[RT88] 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.

[SHM$^+$85] I. Sonomoto, T. Harada, T. Matsushima, K. Kanamori, M. Konuma, A. Uesugi, Y. Nimura, S. Hashimoto, and S. Ohteru. Automatic recognition system of printed music for playing keyboards. *TG*, MA84-22:17–22, 1985. In Japanese.

[Sto80] K. Stone. *Music Notation in the Twentieth Century: A Practical Guidebook.* W. W. Norton & Co., New York, NY, 1980.

[TA82] A. Tojo and H. Aoyama. Automatic recognition of music score. In *Proceedings, 6th International Conference on Pattern Recognition*, page 1223, Munich, W. Germany, 1982. Short English version of longer Japanese [AT82].

[Taw86] S. Tawada. Dance score input system for the representation of human body movement. B.s. thesis, Educational Center for Inf. Proc., Kyoto University, Kyoto 606 JAPAN, 1986. Thesis advisor: K. Hachimura; 59 pages; in Japanese.

[Tho88] E. Thorud. Analyse av notebilder. Rapport 28, Bildebehandlingslaboratoriet, Institutt for informatikk, Universitetet i Oslo, Oslo, NORWAY, August 1988. In Norwegian; 63 pages; ISBN 82-7476-030-1.

[Tøn86] S. Tønnesland. SYMFONI: System for notekoding. Technical report, Institute of Informatics, P.O. Box 1080 Blindern, N-0316 Oslo 3, Norway, November 1986. In Norwegian; generously illustrated; 90 pages.

[Gro90] J. Groever. A computer-oriented description of music notation. Technical report, MUSIKUS, Department of Music, University of Oslo, P.O.Box 1017, Blindern, N-0315 Oslo 3, NORWAY, 1990. Contact: Arvid Vollsnes arvid@ifi.uio.no.

[HO87] K. Hachimura and Y. Ohno. A system for the representation of human body movement from dance scores. *Pattern Recognition Letters*, 5:1–9, January 1987.

[Hut70] A. Hutchinson. *Labanotation.* Theater Art Books, New York, NY USA, 1970.

[IIHO91] T. Itagaki, M. Isogai, S. Hashimoto, and S. Ohteru. Automatic recognition of several types of musical notation. In H. S. Baird, H. Bunke, and K. Yamamoto, editors, *Structured Document Image Analysis.* Springer-Verlag, Heidelberg, 1991.

[Kas70] M. Kassler. An essay toward specification of a music-reading machine. In B. S. Brook, editor, *Musicology and the Computer*, pages 151–175. City University of NY Press, New York, NY, 1970.

[Kas72] M. Kassler. Optical-character recognition of printed music: A review of two dissertations. *Perspectives on New Music*, 11(1):250–254, Fall-Winter 1972.

[KI91] H. Kato and S. Inokuchi. A recognition system for printed piano music using musical knowledge and constraints. In H. S. Baird, H. Bunke, and K. Yamamoto, editors, *Structured Document Image Analysis.* Springer-Verlag, Heidelberg, 1991.

[LC85] Myung Woo Lee and Jong Soo Choi. The recognition of printed music score and performance using computer vision system. *Journal of the Korean Institute of Electronic Engineers*, 22(5):429–435, 5 September 1985. In Korean; English translation available from H. S. Baird.

[Mah82] J. V. Mahoney. Automatic analysis of musical score images. B. s. thesis, Dept of Computer Science and Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02129 USA, May 1982. Advisor: B. K. P. Horn; 53 pages.

[Mar87] N. Martin. Towards computer recognition of the printed musical score. B.sc. project report, Thames Polytechnic, Woolwich, London UK, May 1987.

[Mat85] T. Matsushima. Automated high speed recognition of printed music (Wabot-2 vision system). In *Proceedings, 1985 International Conference on Advanced Robotics*, pages 477–482. Japan Industrial Robot Association (JIRA), 3-5-8, Shiba Koen Minato-ku, Tokyo, 1985.

[MOH89] T. Matsushima, S. Ohteru, and S. Hashimoto. An integrated music information processing system: PSB-er. In *Proceedings, 1989 International Computer Music Conference*, pages 191–198, Columbus, Ohio, November 1989.

[NP73] G. Nelson and T. R. Penney. Pattern recognition in musical score - project no. m88. *Computers and the Humanities*, 8:50–51, 1973.

[NSI78] Y. Nakamura, M. Shindo, and S. Inokuchi. Input method of [musical] note and realization of folk music data-base. *TG*, PRL78-73:41–50, 1978. In Japanese.

[Oht84] S. Ohteru. A multi processor system for high speed recognition of printed music. *Natl. Conv. (Rec.)*, 1984.

[OIT79] M. Onoe, M. Ishizuka, and K. Tsuboi. Experiment on automatic music reading. In *Proceedings, 20th IPSJ National Conference*, volume 6F-5, 1979. In Japanese.

of the notation and the completeness of the recognition.

## References

[AC82] A. Andronico and A. Ciampa. On automatic pattern recognition and acquisition of printed music. In *Proceedings, International Computer Music Conference*, pages 245–278, Venice, Italy, 1982. Computer Music Association Publications.

[AT82] H. Aoyama and A. Tojo. Automatic recognition of printed music. *TG*, PRL82-5:33–40, 1982. In Japanese.

[BH91] D. Blostein and L. Haken. Justification of printed music. *Communications of the ACM*, 34(3):88–99, March 1991.

[Car89] N. P. Carter. *Automatic Recognition of Printed Music in the Context of Electronic Publishing*. PhD thesis, Univ. of Surrey, Depts. of Physics and Music, February 1989. 174 pages.

[CB91] N. P. Carter and R. A. Bacon. Automatic recognition of printed music. In H. S. Baird, H. Bunke, and K. Yamamoto, editors, *Structured Document Image Analysis*. Springer-Verlag, Heidelberg, 1991.

[CBM88] N. Carter, R. A. Bacon, and T. Messenger. The acquisition, representation and reconstruction of printed music by computer: A survey. *Computers and the Humanities*, 22:117–136, 1988.

[CBT88a] A. Clarke, B. M. Brown, and M. P. Thorne. Inexpensive optical character recognition of music notation: A new alternative for publishers. In *Proceedings, Computers In Music Research Conference*, Bailrigg, Lancaster LA1 4YW, U.K., 11-14 April 1988. Sponsored by Ctr for Res. into the Applications of Computers to Music, Dept. of Music, Univ. of Lancaster; 6 pages.

[CBT88b] A. T. Clarke, B. M. Brown, and M. P. Thorne. Using a micro to automate data acquisition in music publishing. *Microprocessing and Microprogramming*, 24:549–554, 1988.

[CBT89] A. T. Clarke, B. M. Brown, and M. P. Thorne. Coping with some really rotten problems in automatic music recognition. *Microprocessing and Microprogramming*, 29:547–550, 1989.

[Ede68] M. Eden. Other pattern-recognition problems and some generalizations. In P. Kolers and M. Eden, editors, *Recognizing Patterns*, pages 196–225. MIT Press, Cambridge, Massachusetts USA, 1968.

[FAP89] I. Fujinaga, B. Alphonce, and B. Pennycook. Issues in the design of an optical music recognition system. In *Proceedings, 1989 International Computer Music Conference*, Columbus, Ohio, 2-5 November 1989. 4 pages.

[FAPB89] I. Fujinaga, B. Alphonce, B. Pennycook, and N. Boisvert. Optical recognition of musical notation by computer. *Computers in Music Research Newsletter, No. 1*, 1989. 4 pages.

[Fuj88] I. Fujinaga. Optical music recognition using projections. Master's thesis, McGill University, Faculty of Music, Montreal, CANADA, September 1988. For an M.A. in Music Theory; 67 pages.

[Gro85] Mu Research Group. Automated recognition system for musical score. Bulletin 112, Science and Engineering Laboratory, Waseda University, 6-1 Nishiwaseda 1-chome, Shinjuku-ku, Tokyo 160, 1985. 28 pages.

phonic music for separating and isolating musical symbols. [FAP89] reports on a second implementation, on a Sun workstation. This music recognizer presents the user with a split screen, showing both the original music and the recognized score. The user then interactively corrects the recognized version. [FAP89] states that the initial target is simple polyphonic music with one note per stem. The next target will be the recognition of traditional piano music.

[RT88] describes a system for primitive extraction in hand-written sheet music (Sections 4.5 and 5.6). Few performance results for this system have been given. Extremely difficult input was used: sloppy handwritten music, digitized at only 100 dpi. Analysis results for a few short sections music were described. In these tests, a total of 17 solid note heads were recognized; 7 of these had mistaken pitch, and 9 extra blobs were detected. It is difficult to estimate how this method would compare to other methods when applied to higher-resolution, better-quality input. The authors state that "The results obtained using these techniques were quite good — certainly far above expectations."

Clarke *et al.* [CBT88a, CBT88b, CBT89] describe a partial implementation of a score-reading system written in Turbo C and running on an IBM PC (Section 5.7). Digitization occurs at around 200 dpi. A user-chosen threshold converts the scanned gray-level image into binary. Preliminary results have been reported but no details about testing are supplied. [CBT88b] states that "At present, the system will correctly recognize single line melodies of a subset of musical notation, with about 90% accuracy."

[Car89, CB91] describe a promising system for segmentation of musical symbols in images of music notation (Section 4.6 and 5.8). It produces a natural, stable segmentation under difficult imaging conditions, without an excess of ad hoc rules. A complete recognition system using these segmentation results is under development. The implementation is in C, running under Unix. The results currently available do not include figures on recognition accuracy. The test data consisted of nine images sampled at 400 dpi (4680 × 3344 pixels for an A4 size page) and a tenth image sampled at 200, 300 and 400 dpi. Various layouts and font sizes were used, including solo instrument parts, solo instrument with piano accompaniment, and orchestra score. The results shown in [CB91] demonstrate a clean separation of music symbols from staff lines. The system shows good tolerance of noise, limited rotation, broken print and distortion.

[KI91] is a sophisticated system for recognizing images of piano music (Sections 4.7 and 5.9). This system handles complex music notation, including two voices per staff with chords and shared note heads. Symbol recognition is fairly complete, including slurs and pedal markings. Grace notes are not recognized. The system is written in C, running on an APOLLO workstation. Scanning is at 240 dpi. Four works were selected for experimentation, with the indicated recognition rates: Beethoven's "Für Elise" (95.6%), Mozart's Turkish March (91.5%), a Chopin etude (87.1%), and a movement of a Beethoven sonata (83.3%). The recognition rate is calculated by counting the words that are modified or appended in the process of correcting the output of the system. These words correspond to information such as note pitch and duration, or the presence of a slur or pedal marking. These are impressive recognition rates, given the complexity

Examples of successful and unsuccessful recognition are given. [Mah82] reports on a partial implementation of a system to recognize primitive patterns in music; the inference of musical symbols from these was not attempted (Sections 4.4 and 5.3). His examples show the effect of removing the bare parts of lines (those portions where the measured line width lies within a given range); the successive removal of staff lines and stems is used to isolate the remaining symbols. The primitive-extraction system, written in Zetalisp, is capable of extracting roughly vertical or horizontal lines or shallow arcs and elementary 4-connected regions from a binary image. Interactive facilities are used for determining normalization factors, building and refining line and region descriptions, and constructing and classifying pattern objects. Small tests have been done, including a few measures of polyphonic guitar music, scanned at 250 dpi. The detection of curved lines (slurs and ties) was not tested. This method emphasizes human interaction in the "pre-calibration" stage used to construct symbol descriptions; thus the system is fine-tuned to the examples it was tested on. Even though good results were achieved on the small tests shown, no conclusions can yet be drawn about the generality of the method.

The vision system for the Wabot-2 robot [Oht84, Gro85, Mat85, SHM+85, Roa86, MOH89] can perform fast, accurate recognition of simple three-part organ scores (Sections 4.8.1 and 5.4). Images are 2000 by 3000 pixels, which provides about 250 dpi for a standard page of music. A page of music is processed in 10 or 15 seconds, using special correlation hardware and parallel processing to achieve such speed. Basic music symbols are recognized, including clefs, accidentals, time signatures, bar lines, notes, beams, rests, staccato and marcato marks. Symbols such as words, slurs, ties, expression marks, ornaments and tempo indications are not recognized. [Gro85] reports on recognition results for ten simple scores, with 15 second recognition time and nearly 100% recognition rate. This system has been successfully used in live demonstrations. It is not clear whether this system could be easily extended to handle more complex music notation. [LC85] describe a system that recognizes staff lines, bar lines, notes, chords and rests. Projection methods are used. Processing time for a 237x192 image was eight minutes on an Apple II. These efforts were hampered by poor imaging conditions. A recognition rate of over 90% was obtained on some inputs.

[Fuj88, FAP89, Pen90] describe music-recognition systems that use X and Y projections for symbol identification (Sections 4.8.2 and 5.5.2). [Fuj88] reports on an IBM PC implementation whose symbol extraction capabilities were well-tested on a variety of monophonic music taken from various publishers. (Seventeen monophonic works were tested; the code for staff-locating was tested on an additional 12 polyphonic works.) The music is scanned at 200 dpi; each page takes about two minutes of processing time on a PC (fifteen seconds per staff). A symbol recognition rate of 70% was achieved on new samples; 100% recognition was achieved on the development samples. Recognized symbols include four types of clefs, key signatures, half notes, quarter notes, beamed notes, flagged notes (the type of flag is not recognized), two classes of accidentals (flat and sharp/natural), quarter and eighth rests, duration dots and bar lines. The projection methods as implemented in [Fuj88] rely heavily on properties of monothereby providing top-down feedback for error correction and data disambiguation. This approach provides a promising method for subdividing a recognition system into intellectually manageable tasks. The modularized knowledge organization and control structure provide a good basis for scaling up from a small system to a large system.

### 10.2 Image Defects and Broken Characters

Robust methods must be developed for interpreting music notation despite image noise. [Mah82], p. 27, discusses problems with image noise, and says that "The worst case is when a primitive of character pattern, which is expected to be connected for extraction purposes, is actually broken up into two or more fragments in the image." His work does not address this problem. [Car89] mentions difficulties that arise from fragmentation of symbols due to poor print quality. Carter's processing methods (Section 5.8) are designed to minimize these problems. He states that "Severe break-up of symbols will, however, continue to be a problem for a topology-based approach and will probably necessitate the use of artificial intelligence based techniques in order to take advantage of higher level musical information." [Car89], p. 153.

## 11 Working systems and Experimental Results

Various systems for music recognition have been implemented. Comparing these systems is difficult because of the variety of input data used, and because it is difficult to judge how much each system is tuned to the particular examples for which results are reported. Thus we simply present a list of systems, approximately in chronological order. The system described in [Pru66] recognizes quarter notes and beamed note groups (Sections 4.1 and 5.1). Up to four-note chords are allowed, provided the note heads are physically adjacent, forming "note clusters." Other music symbols, such as rests, flags, hollow note heads, accidentals and clefs, are not treated. All tests were drawn from one musical publication. On this small experimental sample, good recognition results were achieved. [Pre70] describes a system that recognizes a more complete set of symbols (Sections 4.2 and 5.2). Work-in-progress towards this system was mentioned in [Ede68]. The system was only tested on four small examples drawn from the same piece (Mozart duets for two wind instruments, scanned at 225 dpi). [Pre71] reports a total of 137 components in all of the tests; these components were constructed from 527 fragments (Section 4.2). Good recognition rates were achieved, but the recognition system was tuned to these examples. Recognized symbols include clefs, accidentals, half quarter and eighth notes; flagged sixteenth notes are not recognized but multiple beams are permitted. No chords were recognized. [AC82] describes a system that recognizes clefs, key signatures, notes, rests and accidentals in simple monophonic music. Hand-chosen thresholds are used to binarize the images (force to bilevel). Staff-line removal is used for symbol segmentation (Section 4.3) and syntactic methods are used for symbol recognition (Section 7).

size, shading and position of each Labanotation symbol. X and Y projections of a reduced-resolution image are computed; prominent peaks in these projections are used to detect base lines and separation lines. Next, skeletonization is used to split the image into two images, one consisting of line components and the other consisting of solid blob-like component. The base and separation lines are located in the line-image, and then removed. The resulting line image and blob image contain candidate figures for Labanotation symbols. Touching and incomplete symbols are processed using knowledge of the notation.

## 10 Open Problems

Many open problem must be solved before optical score reading becomes reliable enough to accomplish many of the goals mentioned in Section 1.1. We briefly mention two problem areas: problems of system organization, and problems of image noise.

### 10.1 System Architecture: Scaling up a Prototype

In music reading, as in other applications, it is difficult to scale up a small working prototype into a large, complete system. For example, in syntactic methods, a grammar organization that is intellectually manageable when it contains 30 production rules may be come unmanageable if it is expanded to hundreds of production rules. It is hard to predict the difficulty of extending a system that can analyze monophonic music to one capable of analyzing polyphonic music. Thus, methods of structuring and organizing systems to be expandable are of great interest. In this section we summarize various approaches to the organization of a music-recognition system.

[RT88] describes the use of a rule-based system (Section 5.6). Various researchers have experimented with syntactic methods (Section 7). Also, multiple passes are often used to break the music-recognition task into smaller subtasks [AT82, RT88]. The projection methods (Section 5.5) have the advantage of great simplicity. They are able to recognize simple monophonic music efficiently. However, it may be difficult to extend these methods to polyphonic music, which contain more symbols that are connected or in vertical alignment. [Fuj88], p. 3, states that the restriction to monophonic music "is not critical since a complete OMR [Optical Music Recognition] system will contain a number of subprograms, each specifically designed to analyzed a certain type of score."

[KI91] describes a flexible control structure and data flow for a recognition system. A variety of recognition methods can be accommodated. These methods communicate their results via a working memory that contains hypotheses at various levels of abstraction, ranging from the pixel image at the bottom to the abstract interpretation at the top. Kato and Inokuchi argue that a top-down approach to recognition is useful: high-level knowledge about music notation can be used to constrain low-level pattern processing tasks. In the system described in [KI91], information in the working memory is generated primarily in a bottom-up fashion, but upper levels can reject the results produced from lower levels, [Tøn86] describes an extensive project at the University of Oslo to read sheet music from video camera images with variable lighting, scale, and orientation. Histogramming, skeletonization, projections and template matching are used, resulting in a system that can recognize staff lines and some of the most common music symbols. [Øst88] and [Tho88] describe a second music-reading project at the University of Oslo. In [Øst88], after staff-line removal, a polygonal approximation is constructed by tracing the contours of the music symbols. An analysis of concavities is performed to segment the symbols into constituent components. [Tho88] performs symbol classification using statistical and syntactic methods. Features used include height, width, area, perimeter, and number of holes. Feature analysis was combined with the use of syntactic rules about music symbols.

The music reading system of [NSI78] is aimed at constructing a database of Japanese folk music. This music is monophonic. Projection methods appear to be used. [AT82] reports on another Japanese score-reading effort; [TA82] is a one-page English summary of this work. The motivation for this work is the creation a music data base. The system is designed to recognize both monophonic and polyphonic music, including music with chords as well as music where two voices are printed together on a single staff. A series of passes are used to detect staff lines, to form a coarse segmentation by removing portions of staff lines, to perform fine segmentation, to classify segments into 10 categories, to perform symbol recognition using decision trees designed for each category, and finally, to perform a syntax check and interactive correction of erroneous or ambiguous interpretations. Preliminary experiments are reported to be quite promising.

Other publications available only in Japanese include [OIT79].

## 9 Dance Notation

Music notation is one of many graphical notations used for transmitting information. The problem of automatic score reading is related to the problem of reading other graphical notations. Some notations are more similar to music notation than others; perhaps dance notation is particularly closely related. [HO87] describes a recognition system for Labanotation, one of the common notations for dance [Hut70]; a more detailed account of this work, in Japanese, is given in [Taw86]. Labanotation is similar to music notation in that information is transmitted by symbols drawn on a background of lines, with one axis denoting the passage of time. A Labanotation score is written in vertical runs, with time increasing from bottom to top. Vertical “base lines” subdivide each run into columns, where each column represents the movement of a body part such as the head, an arm, or a leg. The base lines are somewhat analogous to staff lines in music. Horizontal “separation lines” provide synchronization points; these are roughly analogous to bar lines in music. Body movements are indicated by polygonal symbols drawn in the column corresponding to the body part. Duration of a movement is indicated by the vertical length of a symbol, direction is indicated by the shape of the symbol, and vertical motion is indicated by the shading of the symbol. The principal goal in [HO87] is to recognize the shape, above-right of, and above-left of) to relate terminal and nonterminal grammar symbols. The terminal grammar symbols are geometric figures such as white dots, black dots, and oriented lines of various lengths.

[Pre70] mentions grammars for music notation but restricts his work to the development of algorithmic implementations of syntax rules. Prerau makes a distinction between notational grammars and higher-level grammars for music. Notational grammars allow the computer to recognize important music relationships between the symbols of the music sample. The higher-level grammars are concerned with phrases and larger units of music. [Pre75], p. 27, says "to recognize music notation, however, a computer must find an algorithmic description of the music notation syntactical system." His research concentrated on algorithmic syntax rules, used to constraint the possible locations of various symbols. He concludes "The determination of this algorithmic description, a major phase of the solution of the recognition problem, ... may be the first detailed algorithmic description of the grammar of even a subset of the standard music notation symbol system." This algorithmic description is directed at recognition of music notation; other researchers have developed algorithmic descriptions of music syntax directed at generation of music notation (*e.g.* [Rou88, BH91]).

[Fuj88], p. 16, states that "Music notation grammar is context-free and LL(k); this is in effect what allows musicians (top-down parsers) to read the music as efficiently as they do." A small context-free grammar for simplified, monophonic music is presented; this ambiguous grammar does not contain any positional information other than left-to-right ordering. The author notes the limitations of the purely syntactic approach where context is not taken into account, and suggests that attributes could be used to introduce semantic considerations into the grammar.

[Mat85], p. 481, mentions the use of a musical grammar to correct errors such as missing beats or contradictory repeat signs. The following grammar constraints are given, to be applied to three-part organ music: each of the three parallel voices have the same total note duration; a fat double bar appears only at the end of each part; a treble or bass clef always appear right at the start of each staff; the time and key signatures almost never change within a system, and can be determined by majority rule; and, except for pickup measures, the number of beats in each bar should match the time signature.

## 8 Miscellaneous Papers

In spite of our best efforts, we were unable to locate copies of all known publications on this subject; for completeness, however, we wish to mention [Mar87], an undergraduate research project, [Wit73], a one-page abstract touching on the subject, and [NP73], a two-page summary of remarks.

Music image analysis is an international research field, and English translations or summaries are not always available. Here we offer brief comments on several foreign-language publications, with apologies for our inability to understand them fully.

tion of symbols that span measure boundaries. The final image interpretation is formed by combining the partial results from each measure. Good recognition results are obtained on complicated piano music, as discussed in Section 11.

## 6 Relative Positions of Symbols

In music notation, a two-dimensional arrangement of symbols is used to transmit information. Thus in addition to recognizing the identity of individual symbols, it is necessary to analyze the relative positions of symbols. A variety of methods have been devised for describing and testing relative symbol positions. Non-syntactic methods are reviewed here; syntactic methods will be discussed in Section 7. [Pre70] places the minimum and maximum x values for recognized symbols into a sorted list. This permits overlapping symbols to be easily identified. [Mah82] treats musical symbols as being made up of simple component primitives. Spatial relationships between musical symbols are expressed as simple relationships between their component primitives. For example, if the second note in a beamed note sequence is preceded by a sharp, then the important relationship is that between the sharp and the note head to its right. This is a simpler description than that which results by viewing the beamed note sequence as a single symbol which "surrounds" the sharp.

It is hoped that the use of simple relationships between component primitives will simplify syntactic description and analysis. Mahoney suggests describing relationships as an absolute distance combined with a relative position (left, right, above, below). The distance between two primitives can be defined in a variety of ways. For example, one could designate distinguished locations on each primitive, such as the center of a dot and the midpoint of a line, and measure distances between these distinguished points. Alternatively, one could measure the distance between the closest points on two symbols.

## 7 Syntactic Methods

Music notation consists of symbols related to each other in a two-dimensional way, and these two-dimensional relationships often carry information. The significance of these relationships must be captured in a syntactic description of music. Various methods have been suggested for extending grammatical methods which were developed for one-dimensional languages. While many authors suggest using grammars for music notation, their ideas are only illustrated by small grammars that capture tiny subsets of music notation.

[AC82] uses a high-level grammar for describing the organization of music notation in terms of music symbols. Lower level grammars are used to describe the structure of individual music symbols; these grammars are used for music symbol recognition. The high-level grammar is strict and very simple, describing a piece as a sequence of staves, where a staff starts with a clef, then a key signature then a sequence of measures. The lower level grammars are only briefly described in the paper. They use 5 adjacency operators (above, below, right of, placement of symbols. These complex notations may be ambiguous, so proper knowledge is required for their interpretation.

Music symbols differ greatly in size and position, frequency of appearance, importance and so on. Thus it is difficult to devise a single method for recognizing all symbols. It may be necessary to use a variety of recognition methods. A flexible control structure is required to make this possible. Kato and Inokuchi use a collection of processing modules that communicate by operating on a common working memory. The working memory represents information about the current bar of music at five levels of abstraction. The first layer is the pixel image. The second layer contains primitives, including stems, note heads, beams, flags, accidentals, duration dots, and rests. The third layer contains music symbols: notes and rests which are synthesized as combinations of primitives from the second layer. The fourth layer contains the meaning of each symbol, such as the pitch and duration of a note. The fifth layer contains possible interpretations of the bar as a whole; these are formed by time-order combinations of the hypotheses in the fourth layer.

The four processing modules are (1) primitive extraction, (2) symbol synthesis, (3) symbol recognition, and (4) semantic analysis. These processing modules are made up of one or more recognition and verification units. The primitive extraction module contains units for recognizing items such as stems, beams and note heads. Hypothesized primitives are removed from the pixel image; thus the order of execution of units influences the result. Execution of units occurs in a heuristically-determined order.

The operation of the processing modules is governed by a variable threshold that controls the strictness of matching. Tight thresholds mean that extracted primitives are faithful to the primitive model, whereas looser thresholds mean that regions whose shape are far from the primitive model are also extracted as primitives. Unacceptable hypotheses are rejected at higher layers, and sent back to lower layers for further processing. (For example, rejected primitives are restored in the pixel image.) Thus results are obtained quickly for high-quality images, but the analysis of noisy images takes much longer.

High-quality parts of the image are recognized first, with tight thresholds. Once these extracted primitives have been eliminated from the image, further recognition is performed with looser thresholds. Recognition consists of pattern processing (symbol recognition) and semantic analysis. The pattern processing has to cope with overlap between symbols, breaks in thin lines, and unexpected ink spots. The processing modules use knowledge about music notation to constrain the pattern processing task. For example, the distance between staff lines is used to restrict the size of symbols. The width of the staff lines is used as an indicator of the image quality, and is used to set some thresholds for symbol recognition.

Recognition proceeds one measure at a time. In the preprocessing stage, staff lines and bar lines are detected (Section 4.7). Next, subimages containing single measures are processed. This processing consists of staff-line elimination, recognition of attributive symbols (clefs, key signatures and time signatures), and recognition of note symbols. Finally, the postprocessing stage performs recognisymbol-image. (Complete template matching was judged to be too computationally expensive.) For example, three rows of pixels, near the top, middle and bottom of the symbol, are used to identify an accidental as a flat, a sharp or a natural. Individual notes in a beamed group of notes may be identified by examining top and bottom profiles of the beamed-note symbol.

Work on chord recognition is beginning. Chord recognition is complicated because note heads are not constrained to occur only at the end of a stem. Starting with a quarter-note or half-note chord that has been perfectly segmented, stem direction can be determined by checking a row at the top and bottom of the symbol: one of the rows should intersect a note head and the other row should intersect a stem. The note heads themselves are found by checking a vertical column of pixels that is offset from the stem. If the stem x location is in the middle of the chord, then two vertical columns of pixels are checked to look for note heads on either side of the stem. No mention is made of how beams or flags are handled.

This system is not yet complete. Many of the proposed techniques are not robust; noise sensitivity may turn out to be a significant problem. [CBT88a] mentions that "other problems that need to be solved include that of recognizing symbols that coalesce together, and the complications that spurious points or noise can cause during the recognition process."

### 5.8 Carter: LAG-based Symbol Segmentation

[Car89, CB91] discuss a comprehensive system for segmentation in images of music notation. Section 4.6 reviewed the use of the transformed LAG to separate staff lines from music symbols, and to describe objects which correspond to music symbols or connected components of music symbols. These segmentation results are interpreted by a recognition system; compared to the segmentation system, the recognition system is in an early stage of development. The objects resulting from the segmentation are classified according to bounding-box size, and according to the number and organization of their constituent sections. [CB91] notes that "overlapping or superimposed symbols will need to be separated out by a specific algorithm. This is similar to the not insignificant problem of character separation, but far more complex due to the 2-D organization of music notation."

### 5.9 Kato and Inokuchi: a Layered Working Memory

Kato and Inokuchi describe a sophisticated recognition system for printed piano music [KI91]. Musical knowledge is required to deal with the connections and overlaps between music symbols, and to handle ambiguities. A top-down approach is used, recognizing one measure of music at a time. The system is designed to handle both simple and complex notation. Simple, monophonic notations can be interpreted uniquely by means of simple rules. Complex notations have higher symbol density, with more connections, overlaps and complicated character. Thus it may not be realistic to attempt to recognize music at such low resolution. [RT88] states that the results obtained were "quite good." Indeed, the low-level processing does capture some of the important features of the poor-quality input. However, it is not clear that further processing of such low-quality segmentation results will succeed. To better illustrate the performance of this system, tests with good-quality machine-printed and hand-written music are needed.

The following primitives are recognized: circular blobs (for closed note heads), circles (open note heads), horizontal lines, non-horizontal line segments, and arcs. (Symbols such as clefs do not occur in any of the test examples.) The location and orientation data for each primitive are intended to form the input to a high-level visual expert system. Primitive identification is coded as several passes, with procedural Fortran code in the first passes and "knowledge-based" Prolog code in the last pass. Staff-line detection has been reviewed in Section 4.5. Note-head detection is extremely difficult in these handwritten images, and a general-purpose blob detector is often fooled. Thus note heads are searched for in constrained locations. Vertical lines, which might be note stems, are located. Then a thickness measure is used to test for wide spots at the ends of each potential stem. Finally, if there is a wide spot, it is accepted as a note head if it has a circularity measure greater than some threshold. The authors claim to have benefited from writing most of their code using a rule-based approach. Unfortunately, no comparison is made with other techniques that have been used for incorporating knowledge of music-notation into a recognition system.

### 5.7 Clarke: Score Reading on a Microcomputer

Work by Clarke *et al.* [CBT88a, CBT88b, CBT89] is directed at performing optical score reading on a microcomputer. Thus much of their effort is directed at dealing with the main-memory-size restrictions on IBM PCs, and at developing computationally inexpensive methods for symbol identification. Staff lines are identified and removed before the remaining symbols are classified. The staff lines are located by looking for long horizontal runs of black pixels. Then the neighborhood of each staff-line pixel is examined to determine whether a music symbol intersects the staff line at this point [CBT88a]. It is not clear how much this method has been tested; the authors claim that this "relatively simple algorithm has proved to be satisfactory in removing the stave lines from the image."

The image is processed one staff at a time, to accommodate the memory limitations on a PC. Staves are located by examination of a single column of pixels near the left end of the system. Large blank sections indicate gaps between staff lines, and are used to divide the image into individual staves. Complete staff separation is not always achievable — parts of symbols belonging to the staff above or the staff below may be included; these have to be ignored in processing.

For symbol recognition, an initial classification is obtained from the symbol height and width, as in Prerau's system (Section 5.2). Further classification is performed by examining the pixels in a few particular rows and columns of the contribution due to staff lines is subtracted from this X projection, and then the following features of the symbol are calculated: width, height, area, and number of peaks in the X projection.

These features are used in conjunction with syntactic knowledge for symbol classification. Examples of syntactic knowledge include (1) the first symbol in the staff projection is expected to be one of four clefs, (2) the next group of symbols, containing no horizontal gaps greater than a staff space, are expected to be a key signature, (3) within beamed note groups, notes and accidentals are the only expected symbols and (4) dots of prolongation only occur following notes and rests. In some cases localized projections are used to distinguish symbols; for example, a Y projection of the bottom staff space is used to distinguish between treble and bass clefs, and Y projections on either side of a note stem are used to detect flags and beams. Classification of other symbols relies on a width-height space similar to that used by Prerau (Section 5.2). Some classifications are difficult to perform reliably using projection methods. For example, time signatures are not recognized, and the distinction between a sharp and a natural is not made. However, this work demonstrates that projections provide an efficient means for performing initial classifications.

In [Fuj88] a series of ad-hoc tests are used for symbol recognition. [FAP89] reports on a more general and extensible treatment of symbols, based on classification in a feature space. Symbol recognition is done using features extracted from the projection profiles and their first and second derivatives. (No mention is made of noise problems being encountered when taking a second derivative of a projection profile.) Features include width, maximum height, area, aspect ratio and rectangularity. Classification using the k-nearest neighbor rule was found to be prohibitively time-consuming. Offline optimization is used to address this problem; for example, an attempt is made to calculate the most effective subset of features to use.

### 5.6 Roach and Tatem: Rule-based System for Handwritten Music

All of the systems that we have surveyed make some use of knowledge about music notation: the existence and properties of staff lines, note stems and note heads, the syntax of music notation, *etc.* The work of Roach and Tatem [RT88] proposes that such information should be represented in a rule-based system, and that the information should be applied starting with the earliest steps of symbol segmentation and recognition. Tatem and Roach describe the segmentation portion of a prototype system that processes hand-written music notation.

Unfortunately, the input chosen was of such poor quality and such low resolution that it is difficult to judge the effectiveness of the system. The experiments reported in [RT88] use sloppy hand-written sheet music as input. The music is digitized at 100 dpi, which means that the distance between staff lines is six rows of pixels. The staff lines themselves occupy one or two pixels, leaving only four or five pixels for the gap between neighboring staff lines. One of us (Blostein) has found that such resolution is barely satisfactory for displaying legible music on a terminal screen, even when hand-tuned bit maps are designed for each music

Next an image area containing only a staff nucleus is formed, and X and Y projections are used to find bar lines. Notes are recognized using X and Y projections from a small window around the symbol. Characteristic points in the projections are used for classification; a comparison is made with stored projections for known symbols. The examples contain chords and horizontal beams. Pitch and duration of notes is recognized. The authors state that the method is rather rotation-sensitive, so that recognition fails if the image is tilted.

**5.5.2 Fujinaga, Alphonce and Pennycook.** [Fuj88] and [FAP89] describe work that makes extensive use of X and Y projections both for segmentation and for symbol recognition. The system was initially designed on an IBM PC but is being transported to a Sun workstation [FAP89]. [Fuj88], p. 2, defines "The basic task of an OMR [Optical Music Recognition] system is to convert the score into a machine-readable format by means of an optical scanner; the digitized image is then analyzed to locate and identify the musical symbols." Thus, the emphasis of this thesis is on symbol recognition, not on higher-level analysis of 2-D arrangements of music symbols.

There is great variation in shape and size among music symbols. Thus X and Y projections suffice for identification of many music symbols even though they can only establish the approximate shape and size of symbols. The basic strategy employed by [Fuj88] is to locate symbols using projections or syntactic knowledge, and to then calculate local projections for detailed symbol classification. Heavy reliance on properties of monophonic music are made in this process. Fujinaga makes interesting comments regarding system development [Fuj88], p. 53: "Many different algorithms and threshold values were tried until a satisfactory recognition rate was achieved with the training samples...The most frustrating aspect of developing this system was the difficulty of monitoring progress. Because there are several steps involved before any decision is made about the symbols, it was extremely hard to locate problem areas. It was particularly difficult to determine whether misrecognitions occurred because of segmentation errors or because of classification errors." Section 4.8.2 describes how a global Y projection is used to roughly locate the staves, followed by localized Y projections which accurately determine the position of staff lines.

Next, an X projection of the staff is used to locate the individual musical symbols. (Only monophonic music is being analyzed.) An X projection of the entire staff is difficult to analyze due to interference from associated symbols such as expression marks, measure numbers, and lyrics. Instead, [Fuj88] uses a projection of the staff nucleus, the area between the top and bottom staff lines. While this projection cuts off symbols that protrude above or below the staff, it is sufficient for locating symbols. The staff lines themselves give a background projection value in the X projection; symbol-locations are identified whenever the X projection value exceeds the background value by one staff space. At this point a local Y projection is taken, covering the full height of the staff rather than just the staff nucleus. This Y projection is used to determine the vertical extent of the symbol; finally, an X projection is taken using these vertical bounds. The

(1) distortions occur when the page of music sags on the music stand, (2) some rotation may be present, (3) distance to the page may vary, and (4) the illumination is uneven. (Images captured using a flatbed scanner or drum scanner avoid some of these problems.) The image is subdivided and each region separately thresholded to allow for uneven illumination. After staff detection, the image is rotated and normalized to compensate for distortions introduced in scanning.

Musical symbols are recognized according to a two-level hierarchy, where the upper level is implemented in hardware and the lower level in software. Staff lines, note heads and bar lines, which can occur in many places in the image, belong to the upper level. These are searched for using hardware-implemented template-matching. The template matching is done using an AND operation rather than an EXCLUSIVE-NOR operation; that is, the number of coincident black pixels are counted. Eight standard templates for note heads are used. Each template comes in nine different sizes, ranging from 8 × 8 to 16 × 16 pixels. The correct template size is selected on the basis of the normalization parameters resulting from staff-line detection. The lower level of the hierarchy contains symbols whose possible locations are constrained by the recognition results for the upper level symbols; these symbols are found using software-implemented localized search. Lower level symbols include rests, stems, flags, repeat signs, staccato and marcato marks, accidentals, prolongation dots, clefs and time signatures.

Template matching to detect filled note heads leads to incorrect matches. These are eliminated at a later stage, using knowledge about the syntax of music notation. If this method were applied to more complex notation, the problem of spurious matches might become more serious. As it stands, excellent recognition results are achieved on organ scores containing relatively simple notation.

### 5.5 Symbol Recognition using Projections

Various researchers have used projections for symbol recognition. [Pru66] mentions that a function "thickness in points versus x coordinate" could be used to yield equivalent information to the transition information he stores for each symbol-trace. [NSI78] and [Tøn86] both use projections (Section 8). [Taw86, HO87] use projection methods for analyzing Labanotation, a dance notation (Section 9). We now discuss two projection-based methods in more detail.

**5.5.1 Lee and Choi.** [LC85] describes a microcomputer-based music recognition system that uses projection methods to first recognize staff lines, then recognize bar lines and finally recognize notes, including chords and rests. The images have severe noise problems, particularly near the image border; this is due to the imaging method and to lighting problems. Preprocessing is performed to reduce noise.

Staff lines are found in a Y projection. A threshold of 0.7 × (maximum projection value) is used to select projections strong enough to be candidate staff lines. These candidates are searched to find groups of five equally-spaced lines.

the allowable thickness range for the line. The "dot" primitives are not extracted until after line primitives have been processed and removed. [Mah82], p. 22, states that "It is clear from these examples that the actual process of breaking a PSMN [Printed Standard Music Notation] image up into its simple components requires having some knowledge about the structure of the notation. One could not, for instance, successfully extract all primitive symbols, as we have defined them, by simply extracting all lines and then extracting all regions." On the other hand, [Mah82], p. 24, says "There is no reliance on context for the recognition of primitives... All syntactic considerations are left to the analysis routine, whose sole task is to find specified relationships between already classified symbols and build corresponding objects."

Mahoney's processes are initially used in an interactive mode to develop object descriptions and to tailor predefined descriptions to new music samples for which the old descriptions do not work well. A "correct and redisplay" loop is used to refine the descriptions. All measures of distance are normalized on either the staff-line or staff-space thickness. Region masses are normalized on the mass of whole-note interiors (the white space inside the whole-note head). Sample line parameters for describing staff lines, ledger lines, beams, note stems and bar lines are: principal direction (horizontal, vertical); angle (permitted deviation from horizontal or vertical); thickness (lower and upper bound), length (lower and upper bound), maximum permitted gap. Sample region parameters for describing whole-, half-, and quarter-note heads, flags, sharps and dots are: mass (lower and upper bound), width (lower and upper bound), height (lower and upper bound), inclination angle (used only for selected symbols).

### 5.4 Wabot-2: Symbol Detection in a Two-level Hierarchy

In the early 1980s an impressive keyboard-playing robot was developed in Japan. [Gro85] provides a description of the whole system; [Roa86] is a short, easily available overview. For a detailed description of the vision system, see [Mat85]. [SHM$^+$85] is a reference suitable for readers of Japanese. [MOH89, IIHO91] describe extensions to form the PBS (Performance, Score, Braille) system. Among its other capabilities, the Wabot-2 robot has a vision system capable of interpreting images taken of sheet music placed on a music stand. For an anthropomorphic effect, the CCD camera is placed on the robot's shoulders; thus, while the robot plays the keyboard, vibrations prevent the CCD camera from being used for score reading. The sheet music must be read and interpreted before the robot begins to play. Very fast image interpretation is achieved in the Wabot design — approximately 10 to 15 seconds are needed to interpret one page of music. Special purpose hardware and parallel processing are used to achieve such high speed. For the simple scores used, the recognition rate is nearly 100%.

The robot plays three-part organ scores, containing relatively simple notation. There are three staves per system: the top staff is for the right hand, the middle staff is for the left hand, and the bottom staff is for the feet. The robot's video camera captures images of organ scores that have been placed on a music stand. The following attributes of the imaging must be taken into consideration:

of standard music notational symbols ... is that each type of music symbol is significantly different in overall size from almost all other types of music symbols." To exploit this, the bounding-box dimensions of each symbol are expressed in staff-space units. The height and width of the bounding box are used to look up a list of possible matches; this is done via a precomputed table containing the areas where each symbol can occur in a height/width space. (This height-width table was constructed by hand-measurement of the size of many samples of each type of notational symbol. The measurements obtained for each type of symbol form small regions in the height/width space; these regions are enlarged to accommodate printing errors and variations.) Typically there are three to five possible matches for each symbol, given the fairly small subset of music symbols being analyzed. Heuristic tests are used to distinguish symbols that overlap in the height/width space; these tests take advantage of the syntax, redundancy, position and feature properties of each music symbol type. [Pre70] states that this classification scheme is specific to one publisher, but could easily be adapted. [Pre71] presents examples of syntactic redundancy and positional redundancy, and also discusses representative symbol-discrimination tests.

### 5.3 Mahoney: Pattern Primitives and Composite Symbols

[Mah82] deals with the extraction of pattern primitives in music. Section 4.4 reviewed Mahoney's use of line removal for region isolation; here we discuss his primitive-recognition methods. The goal of Mahoney's work is to design an approach on the basis of which a real recognition system might be developed. Some approaches he suggests are probably infeasible in practice, but others may prove useful. Pattern primitives, such as note heads, stems, beams and flags, can be combined to form "composite music symbols," such as notes, chords, and beamed note sequences. This distinction between pattern primitives and composite symbols subdivides and simplifies the recognition task. The primitive lines accommodate the variable parameters of the composite symbols: a parameterized composite symbol such as a beamed note sequence is made up of (unparameterized) characters and parameterized lines. [Mah82], p. 24, states that "It is simpler to give a variable description for a beam than for a beamed note sequence, and it is easier to design flexible recognition procedures around simple descriptions."

Mahoney envisions a system that does not use context for the recognition of primitives. Knowledge about the structure of standard music notation is needed to infer musical symbols from the relationships between the various kinds of primitives; this topic is beyond the scope of [Mah82]. The pattern primitives Mahoney uses are lines, dots and characters. Classes of primitives are described using ranges of values for parameters. For example, lines typically have ranges for length, thickness and orientation. These ranges can be used to characterize the difference between stems, beams and staff lines. The thickness ranges are also useful for removing the bare portions of lines: in order to remove a staff line without leaving holes in the symbols superimposed on the staff line, Mahoney suggests removing the line only in places where the measured thickness is within

**4.8.2 Projection methods for staff-line identification.** Various authors have used projection methods for staff-line identification; some of these methods operate without removing the staff lines from the image. Here we concentrate on the work of Fujinaga *et al.* [Fuj88, FAP89, FAPB89]. Other systems using projections are mentioned in Sections 4.7 and 5.5. In Fujinaga's work, a Y projection of the entire page of music is used to locate staves. The mean of the entire projection is used as a threshold value; then groups of five peaks are sought. Staff lines are rarely perfectly horizontal. In a Y projection, individual staff-line peaks cannot be resolved if the staff is skewed so that delta-y for a single staff line is greater than than half of a staff space. ([CB91] performed projection experiments and states that a rotation of less than 30 minutes of arc can cause the peaks of the Y projection to merge.) Nevertheless, the staff as a whole projects to a region of high values, and can be distinguished from the relatively empty space between staves.

Fujinaga accepts a cluster of five or more peaks as a staff; extra peaks can result from ledger lines, horizontal beams, or skew. A minimum-staff-separation parameter is used: a cluster of five or more peaks must be separated from other clusters of peaks by a certain distance to be recognized as a staff. This method successfully locates most staves, except for percussion staves using only a single staff line. An underlying assumption is that a horizontal line can be used to separate neighboring staves and their associated symbols. This is true in most music printed for single monophonic instruments, but it is not true for dense orchestral scores. Fujinaga's method does not solve the problem of separating staves that occupy overlapping y intervals. Once the area occupied by a staff has been located, localized Y projections are used to accurately locate the staff lines: series of Y projections are taken starting at the right margin, moving leftward until five clear peaks appear.

## 5 Symbol Classification

After staff lines, have been identified and/or removed from the image, the next major processing step is to classify music symbols. A great variety of techniques have been applied to this problem.

### 5.1 Pruslin

[Pru66] uses contour tracing to describe connected binary image regions that remain after removal of thin horizontal and vertical lines. Classification depends both on trace properties as well as on inter-trace measurements. A method for performing template matching using contour traces is developed.

### 5.2 Prerau: the Height/Width Symbol Space

Prerau recognizes a subset of music symbols using simple measures. Relative symbol size is used for an initial classification. [Pre75], p. 27, states "A property histograms of run lengths are calculated. The maximum in the 0-pixel histogram is interpreted as the staff spacing, and the maximum in the 1-pixel histogram is interpreted as the staff-line width. Once the staff size has been established, the staff lines themselves are located. Run-lengths in 10 evenly-spaced columns are used to get accurate local estimates of staff-line spacing and width. Next, small rectangular sections of the staff are analyzed, near the left and right ends of the staff. Short horizontal runs are eliminated; this eliminates most music symbols, but beams and portions of note heads and clefs remain. Next an X projection is used to estimate the locations of the ends of the staff, and a Y projection is used to obtain an accurate height estimate. If the edge of the staff is not found, the window is moved further toward the margin of the page. Bar lines are found using similar methods, tailored to the recognition of piano music. Rectangular masks are placed on the right-hand staff (top staff), the left-hand staff, and between the staves. Then short vertical runs are eliminated, and hypothesized bar-line locations are extracted from an X projection. The existence of a bar line is established if all three masks hypothesize a bar line at the same X location.

In this system, the distance between staff lines is used to restrict the size of symbols, and the width of the staff lines is used to set thresholds for symbol recognition. Staff lines are eliminated from the image before recognition of music symbols begins. The staff lines are tracked from the left, based on initial estimates of their location. A staff line is eliminated wherever the staff width is below a threshold. The methods used for symbol recognition are described in Section 5.9.

### 4.8 Staff-line Identification without Staff-line Removal

So far we have surveyed techniques for detecting and removing staff lines from an image of music notation. Some researchers have developed alternate analysis techniques that that involve staff-line identification, but not staff-line removal.

**4.8.1 Template matching in the presence of staff lines.** [Mah82], p. 21, suggests that template-matching could be applied to characters such as time-signature numerals without actually separating them from the background lines; no attempt is made to implement this.

The Wabot-2 system [Mat85] does perform template matching in the presence of staff lines (Section 5.4). Staff lines are detected and used to normalize the image, to determine the score geometry, and also to restrict the search area for music symbols. Staff lines are detected in hardware by a horizontal line filter. In order to tolerate skew, a short filter is used: the filter size can be between 8 and 80 pixels long. Where five equally-spaced lines are found, a staff is deemed to exist. Normalization parameters are calculated for each staff; these parameters include staff location, staff inclination, area covered by staff, and note-head size. In preparation for further processing, the image of each staff is normalized according to these parameters.

information, derived from the LAG, is used to determine whether a symbol has merged with a staff line.

The Line Adjacency Graph is formed directly from a vertical run-length encoding of a binary image. An individual vertical run of pixels is called a segment. A transformed LAG is formed by linking together neighboring segments to form sections. Sections are formed using a left-to-right scan, in which neighboring vertically-overlapping segments are linked. Junctions occur when a segment in one column overlaps several segments in an adjacent column; sections are terminated at these junctions. In the transformed LAG, each section is represented by a node in a graph, and junctions are represented by edges in the graph. The nodes in the transformed LAG should correspond to structural components of musical symbols. A rule limiting the rate of change in section thickness helps accomplish this. The rule states that the current section is terminated if its average height differs from the height of the next segment by more than a factor of 2.5. This rule ensures that staff lines and ledger lines are assigned to different sections than are portions of music symbols. Similarly, a note head is assigned to a different section than the note stem. Section formation is insensitive to small rotations: [CB91] shows consistent sections obtained from an image at two different orientations. The use of a LAG is also efficient, since subsequent processing operates on section data rather than on individual pixels. [Car89] found the LAG preferable to other common methods of data reduction and feature extraction, such as thinning to form skeletons. The LAG is equally effective describing blobs as well as lines.

Noise removal on the transformed LAG proceeds by removing isolated or singly connected sections with small area (5 pixels or less in a 400 dpi image). If removal of these noise regions turns a multi-way junction into a two-way junction, then the two remaining sections are merged provided their heights differ by less than a factor of 2.5.

The transformed LAG is searched for potential staff-line sections (filaments): sections that satisfy criteria related to aspect ratio, connectedness and curvature. Long beams may be included as filaments; these are filtered out using a histogram of filament thickness to determine a threshold for maximum staff-line thickness. Roughly collinear filaments are concatenated together into filament strings, thereby bridging the gaps introduced by superimposed music symbols. The occurrence of five horizontally overlapping and roughly equally-spaced filament strings is recognized to form a staff. After staff lines are identified, the transformed LAG is restructured: further merging of non-staff sections takes place, now that junctions with staff-line sections have been specially marked. At this point, musical symbols are effectively isolated from the staff lines. Connected non-staff-line sections are combined to form objects, which correspond to music symbols or to connected components of music symbols.

### 4.7 Kato and Inokuchi: Run-length Histograms and Projections

In [KI91], the spacing of staff lines is estimated by scanning 10 evenly-spaced columns in the image. Run-lengths are measured in each of these columns, and removal of complete staff lines, is performed to separate vertically-connected note heads. Section 5.3 contains further details of Mahoney's methods for primitive extraction. Mahoney concludes that this system "goes a long way toward correct isolation of characters in the image, but more work is needed for dealing with the more difficult cases of line-region overlap. This is a challenging problem."

### 4.5 Roach and Tatem: Staff-line Identification via Line Angle and Thickness

Roach and Tatem [RT88] discuss a knowledge-based system for segmenting images of music notation. Their processing of staff lines is reviewed here; symbol processing is reviewed in Section 5.6. Roach and Tatem worked with images of hand-drawn sheet music — the staff lines themselves were machine printed, but all other music symbols were hand drawn. Traditional general-purpose methods for line-detection, such as the Hough transform or line tracking, did not perform well on these images; the need to introduce domain specific knowledge was identified. In order to isolate musical symbols, only the bare sections of staff lines should be removed. Staff lines are detected using measures of line angle and thickness. A window is passed over the image to compute a line-angle for every black pixel. The line angle is measured from the center of the window to the furthest black pixel in that window; this furthest black pixel is chosen so that the path from it to the center does not cross any white pixels. To detect staff lines, a large window radius is used. This causes covered staff-line sections to be labeled with a horizontal line-angle despite the interference of the superimposed musical symbols. Once a line angle has been determined, a line-thickness can be measured. These two measurements, combined with adjacency information, are used to identify horizontal lines, and "questionable pixels" which occur at the intersection of a line and a figure. It would be interesting to perform a comparison of this algorithm with the LAG-based methods used by Carter (Section 4.6).

### 4.6 Carter: LAG-based Staff-line Identification

[Car89, CB91] discuss a comprehensive system for segmentation in images of music notation, using processing based on a Line Adjacency Graph (LAG). A common problem in staff-line identification is to distinguish the bare staff-line sections from the sections where the staff is intersected by a music symbol. Particularly difficult is to detect places where a thin portion of a symbol tangentially intersects a staff line. Examples of this include the highest part of a bass clef, as well as the intersection of the treble clef with the lowest staff line. Under these conditions, many other methods of staff-line processing create gaps in symbols, but Carter's LAG-based analysis successfully identifies such tangential intersections of symbols with staff lines. Other goals of this work are (1) to locate staff lines despite image rotation of up to 10 degrees, (2) to cope with slight bowing of staff lines, and (3) to cope with local variations in staff-line thickness. Region examples. Not surprisingly, gaps are left in symbols that are tangent to staff lines, and staff lines are not removed when they pass through small holes in symbols. The crossing of beams and staff lines is a problem: the staff-line removal results "in the suppression of some particulars that ... prevent the identification ... of the linking of the notes" ([AC82], p. 255).

### 4.4 Mahoney: Line Removal for Region Isolation

[Mah82] distinguishes between two types of line removal, removing real lines (bare staff-line sections) and removing ideal lines (complete staff lines). The goal in real line removal is to remove only those parts of the line that do not overlap other symbols; this is accomplished by removing only those portions of the line satisfying the line's allowed thickness range. This type of removal is generally desirable for staff lines. Ideal line removal involves removing the line everywhere along its length. This can be used to split adjacent symbols; for example, performing ideal removal of a staff line can be used to separate note heads that are located on adjacent spaces. In some cases, ideal stem removal splits note heads that are a half-step apart and on opposite sides of the stem. In other cases, a different region segmentation approach would be needed — one that is not provided in [Mah82]. Mahoney repeatedly uses the following strategy for symbol identification: first construct a set of candidates for one or more symbol types, then use symbol-type descriptors to select the matching candidates ([Mah82], pp. 39 ff). For example, to identify staff lines, staff-line candidates are constructed; these include all thin horizontal lines in the image. Next, the staff-line descriptor (specifying allowable thicknesses, lengths, and gap-lengths) is used to classify staff lines. Similarly, the ledger-line descriptor is used to classify ledger lines. Good extraction of staff lines and ledger lines is achieved even though the initial construction of horizontal lines contains unwanted segments, such as the tops and bottoms of half-note heads.

The treatment of vertical lines is similar, with descriptors for stems and bar lines used to classify the vertical lines. The note stems are removed to preserve beam continuity, and then beams are constructed and classified. Once the classification of lines is completed, symbol classification begins. First, loop interiors are found: this detects the whole- and half-note heads that do not have a staff line running through them. (Other regions are also detected; for example, the top half of a "3" used for fingering.) Then the bare staff lines and ledger lines are removed from the image (the lines are removed only where they are within the allowable thickness range, so line removal does not cause gaps in superimposed symbols).

The detection of loop interiors is repeated to find the loops on staff lines. Then the descriptors for half- and whole-note heads are used to classify these regions. (Erroneous regions, such as the top half of the "3" are left unclassified.) Candidate regions for other symbols are constructed by removing all staff lines, ledger lines, stems, bars and beams, and then finding connected regions. All other symbols, such as solid note heads, accidentals and flags, are found by matching symbol descriptors to this set of candidate regions. Special processing, such as normally be blank. Staff line identification is complicated by noise and distortion. [Pre70] notes that the five staff lines found on a piece of sheet music are not exactly parallel, horizontal, equidistant, of constant thickness, or even straight; scanning noise and quantization noise compound these problems. [Car89] notes that staff-line analysis techniques must be able to cope with staff-line inclination and curvature, as well as with the interfering effects of beams and other linear elements in the score. In some cases, staff lines may be obscured to a significant extent by multiple beams, particularly when these are horizontal. Thus standard image-processing techniques for line-finding often do not suffice for locating staff lines.

### 4.1 Pruslin

Two early MIT PhD theses, one by Pruslin [Pru66] and the other by Prerau [Pre70, Pre71, Pre75] addressed the removal of staff lines in images of sheet music. Both of these theses were reviewed in [Kas72]. [Pru66] preprocesses the music image by eliminating all thin horizontal and vertical lines, including many bare staff-line sections and stems. This results in an image of isolated symbols, such as note heads and beams, which are then recognized using contour-tracing methods. This drastic preprocessing step erases or distorts most music symbols other than quarter notes and beamed note groups, so extension of this work seems infeasible [Pre70].

### 4.2 Prerau: Contour Tracing

In his PhD thesis, Prerau [Pre70] describes a "fragmentation and assemblage" method for treating staff lines and isolating music symbols. In the fragmentation step, the system scans along the top and bottom edges of staff lines to identify parts of symbols lying between/above/below the staff lines; a new symbol fragment is begun whenever a significant change in slope is encountered. Fragments from a single symbol are separated by the gaps left from crossing staff lines. In the assemblage step, these symbol fragments are recombined. A simple connection rule is used in the assemblage step: two symbol fragments that are separated by a staff line are connected if they have horizontal overlap. As noted by [CB91], symbols which merge with staff lines do not always have horizontal overlap, and would be disconnected by Prerau's method. For example, the top portion of a bass clef would be disconnected, as would shallow slurs tangent to a staff line.

### 4.3 Andronico and Ciampa: Bare Staff-line Removal

[AC82] mentions an alternate treatment of staff lines, which attempts to remove staff lines only in bare sections where the staff lines are not crossing symbols. Three additional hypothetical staff lines (2 above and one below) are traced to remove ledger lines. Successive trials are used to search for staff lines, using a number of iterations proportional to the amount of background noise. The method used is not described, but the figures show creditable results on simple

Examples are given of the widely varying appearance of symbols such as the treble clef. Music symbols vary in orientation, appearance and positioning. Typical music symbols are much less regular in appearance and positioning than are the characters in printed text. Adjacent and overlapping symbol placements are used, further complicating the recognition process.

## 3 Thresholding and Noise Reduction

A common first operation in a music recognition system is thresholding to convert a gray-scale image into a binary image. Other forms of preprocessing are sometimes used for noise reduction. The early work of Prerau is typical: [Pre71] works with 512x512 images with 8 gray levels, scanned at 225 dpi . A threshold is chosen (apparently manually) to convert to a binary image. Prerau claims that since most points in the image are not near the threshold, the choice of threshold-level is not critical. In [CB91], the scanner performs automatic thresholding. A horizontal low-pass filter is used to remove short breaks in staff lines and symbols. In images scanned at 400 dpi, isolated or singly-connected black sections of less than 5 pixels are removed as noise. (The division of the image into sections is described in Section 4.6.)

[LC85] describes the use of preprocessing for noise reduction. A three-by-three mask is used to eliminate isolated black pixels and to fill in isolated white pixels. Also, a simple filter is used to control the amount of light in the CCTV camera image; details about this filter are not given. In the vision system for the Wabot-2 robot [Mat85] (Section 5.4), the image is subdivided and each region separately thresholded to allow for uneven illumination. The image is then rotated as required and normalized to compensate for distortions introduced in scanning.

## 4 Staff Lines

Staff lines play a central role in music notation. They define the vertical coordinate system for pitches, and provide a horizontal direction for the temporal coordinate system. The staff spacing gives a size normalization that is useful both for symbol recognition and interpretation. Almost universally, sizes and distances are measured in units that are normalized to the staff spacing. Many authors implicitly make an assumption that was explicitly stated by [Fuj88]: the size of musical symbols is linearly related to the staff spacing. This assumption has not been rigorously tested, but appears to hold at least approximately.

Recognition of staff lines is one of the first steps in most music recognition systems. Since recognition of music symbols is confounded by the existence of horizontal lines through the symbols, a common goal is to identify staff lines and remove the bare staff-line sections. [Pre70] identifies three ways in which staff lines interfere with symbol recognition: (1) the staff lines graphically connect symbols that would normally be disconnected, (2) the staff lines camouflage the contour of a symbol, and (3) the staff lines fill in symbol areas that would system capable of recognizing all music notation. [Pru66] states that a complete solution to the music recognition problem is "the specification of: which notes are present, what order they are played in, their time values or durations, and volume, tempo, and interpretation." This level of recognition suffices for only some of the applications listed in Section 1.1.

[Kas70] gives a musician's view of the desired I/O behavior of a music-reading machine. The proposed output language is somewhat clumsy and dated due to its emphasis on binary and octal codes. However, it is interesting to see a musician's list of the information that would be desired from a music-reading machine. In addition, Kassler's definition of a *scanning unit* may be useful. A scanning unit is composed of a subset of music symbols found on one staff and forming an unbroken X projection. The following music symbols are not included in scanning units: staff lines, beams, slurs, ties, brackets, text, and crescendo or decrescendo signs. Thus the scanning units are generally quite small, for example, a clef, one or more key-signature accidentals, a time signature, note heads in a chord. These scanning units are used to delimit the extent of parameterized symbols such as slurs and beams.

### 2.1 Music symbols: Primitives, Parameterized Symbols, and Characters

The definition of a "music symbol" varies, although all authors agree that staff lines are symbols in their own right, separate from all other symbols that appear on the page. Some authors (*e.g.* [Pre70]) define music symbols as all four-connected regions that remain after staff lines have been removed. Thus a beamed note sequence is called a single symbol. Other authors (*e.g.* [Mah82, KI91]) consider such symbols to be composed of pattern primitives such as stems, beams and note heads.

Many authors [Pre70] distinguish between *characters*, which are size invariant, and other malleable symbols such as beams and slurs, which have a parameterized shape. Traditional character recognition methods such as template matching can be applied to music characters, but not to parameterized symbols. [Mah82] distinguishes between the music symbols that describe what is to be played and the music symbols that describe how things are to be played. The recognition of the "what" symbols (notes, clefs, key signatures, and so on) forms the basic music recognition problem. Extending this to the "how" symbols (which occur in great variety, often in the form of text phrases) compounds the basic music recognition problem with that of reading printed alphanumeric text. Note that recognition of the "what" symbols would be sufficient for many of the applications listed in Section 1.1.

### 2.2 Recognition of Music Symbols

A good introduction to the difficulties of symbol recognition in music is given in [Fuj88], Chapter 4. This thesis provides a well-illustrated introduction to music printing methods — typography, engraving, lithography, and modern methods.

**digitizing resolution** The spatial resolution of the document scanner during image acquisition, usually expressed in units of *dots per inch* (dpi). (Some authors prefer *pixels per inch* (ppi).)

**Fig. 2.** Illustration of some terms for musical notation.

## 2 Problem Statement

Common music notation does not have a unique, precise definition. Four particularly useful publications in English that attempt to codify printing standards for music notation are [Gro90, Rea79, Ros70, Sto80]. All of these admit that, in practice, composers and publishers often feel free to adapt old notation to new uses, and invent new notation, as they see fit. There are in fact national "dialects" of music notation, and musical works use many different levels of notational complexity. Thus it may not be possible to devise a single recognition see also ([Fuj88], pp. 4-6). Alphanumeric entry of a music-description language is common, but this method is slow and error prone. Music editors with graphical user interfaces can be used; this reduces errors and speeds up entry, particularly if MIDI input devices can be used to enter pitch and rhythm information directly. Attempts have been made to recognize music from audio input; some success has been obtained with monophonic music, but extension to polyphonic music seems very difficult.

### 1.3 Terminology

Here is a summary of commonly-occurring terminology:

**staff line** A long thin horizontal line which defines a coordinate system for music notation. Typically five parallel staff lines are drawn to form a staff, but only one or two staff lines may be used for percussion music.

**staff-line sections** The *covered* sections of a staff line are those sections where other music symbols intersect the staff line; the remaining sections of the staff line are *bare*.

**staff** The staff lines plus all associated symbols, including music symbols, lyrics and textual annotations.

**staff space** The distance between the staff lines within a single staff. The staff space provides a normalized unit of measurement for expressing distances.

**system** A set of staves that are played in parallel; in printed music these staves are connected by braces, and bar lines may be drawn through from one staff to the next. A page of an orchestra score may contain only a single system. (Some authors prefer the term *paragraph.*)

**staff nucleus** The area of a staff that contains the staff lines and the musical symbols (we introduce this term for lack of any existing term). Many music recognition systems restrict their attention to symbols in the staff nucleus. In order to avoid missing symbols on ledger lines, the staff nucleus can be defined to extend vertically one or two staff spaces above the top staff line and one or two staff spaces below the bottom staff line.

**voice** A musical line. A voice may correspond to a single instrument; a piano part is usually notated as two and sometimes more voices. Several voices may be printed together on one staff: in an orchestra score, the Flute 1 and Flute 2 voices are printed on the same staff (with opposite stem direction), but they are printed separately to make the individual instrumental parts.

**monophonic** Music consisting of a single voice, where this voice contains no chords.

**X projection** A projection of an image onto the X axis, that is, downwards forming a horizontal distribution. The result is a vector whose $i$th component is the sum of all black pixels in the $i$th column of the image. (Some authors refer to this as a *vertical projection*, since the projection is in the vertical direction.)

**Y projection** A projection of an image onto the Y axis. The result is a vector whose $i$th component is the sum of all black pixels in the $i$th row of the image. (Some authors refer to this as a *horizontal projection.*)

sic recognition is much less well formalized, and furthermore depends strongly on the application (Section 1.1). As a result, agreement is elusive on uniform standards for success: existing music recognition systems are able to extract information sufficient for some applications but not for others. Most work through 1990 has concentrated on locating staves and isolating and recognizing symbols. Outstanding problems include effective algorithms to interpret the resulting 2-D arrangement of symbols, and precise formalisms for representing the results of interpretation.

### 1.1 Goals and Applications

Automatic recognition of machine-printed music has been undertaken for a variety of reasons, and as a result, technical goals vary also. For example, for the analysis of musical style, it may be sufficient to extract the pitch, duration, and simultaneity of all notes. It is harder to produce parts from a score (or *vice versa*), since all musical symbols, not only the notes, must be recognized and associated correctly with voices. The following list of applications for printed music recognition is compiled from various authors' lists, including [Kas70, Pre75, Fuj88].

One important class of applications concerns *editing of scores* for reprinting, revision, and preparation of performance materials:

1. adapt existing works to other instrumentations: for example, reduce full scores to piano scores;
2. read various works in old editions and produce a new printing;
3. make critical editions of musical compositions given different printed versions of the 'same' composition;
4. transpose a music sample to some other key;
5. produce parts from a given score or a score from given parts;
6. read in a newly engraved piece of music and proofread it for syntactic and other errors;
7. convert existing scores to Braille to aid blind musicians;
8. print newly written music automatically (if the recognition program were extended to the recognition of handwritten music notation); and
9. produce audio versions of a given written composition; the computer can be used as a combination musician and instrument.

Another class of emerging applications concerns *collecting databases*:

1. create indices of themes and other music features;
2. analyze musical structure and style;
3. test theories of music; and
4. evaluate algorithms for the automatic analysis or composition of music.

### 1.2 Non-optical Input Methods

In the absence of optic music reading capabilities, non-optical input methods have been used. [CBM88] contains an extensive survey of music input methods;

# A Critical Survey of Music Image Analysis

*Dorothea BLOSTEIN*[1] *and Henry S. BAIRD*[2]

[1] Dept Computing & Inf. Science,*** Queen's University,
Kingston, Ontario CANADA K7L 3N6
[2] AT&T Bell Laboratories, 600 Mountain Avenue, Murray Hill, NJ 07974 USA

The research literature concerning the automatic analysis of images of printed and handwritten music notation, for the period 1966 through 1990, is surveyed and critically examined.

## 1 Introduction

Printed and handwritten music notation is intended to document musical information in a legible, archival form. Both recognition and generation of music notation can of course be modeled as mappings between the printed notation and the information it represents (Figure 1).

**Fig. 1.** Recognition and Generation of Music Notation.

While many important details of the appearance of machine-printed music notation are effectively standardized, the information to be recovered during mu-

*** This work was supported in part by the Natural Sciences and Engineering Research Council of Canada.