"ENGRAVE - An Expert System that Understands and Generates Musical Notation"

by

Joseph Edward Tatem

Thesis submitted to the Faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of MASTER OF SCIENCE in Electrical Engineering

APPROVED:

[Signatures]

July, 1987
Blacksburg, Virginia
C.2

LD
56.55
V0.35
1987
T375
C.2
ENGRAVE -- AN EXPERT SYSTEM THAT UNDERSTANDS AND
GENERATES MUSICAL NOTATION

by

Joseph Edward Tatem

Committee Chairman: John W. Roach

Electrical Engineering

(ABSTRACT)

ENGRAVE is an expert system that performs the work of a music engraver. The system takes a digital image of a conductor's score as input and produces engraved-quality copies of the individual parts.

ENGRAVE consists of a vision system and a graphics system. The vision system contains a low-level section that recognizes features and a visual expert system that understands their meanings. Under the current computer vision paradigm, low-level processing is general-purpose while domain knowledge is used only in "high-level" routines. We hypothesize that domain knowledge is necessary at the low level as well for a vision system to work reliably. The visual expert uses a frame-based approach to explain the objects found in music.
graphics system, we make use of the concept of "spatial planning" to generate correct musical notation. We show by example that spatial planning can be applied to other domains.
Acknowledgments

I would like to take this opportunity to thank those who have been involved in the project and have supported me while I was working on it.

John Roach was a most able and competent advisor. He was both demanding and understanding with just the right mix. Somehow, John knew when I needed a push and when I needed a break and was nearly always available at odd times and on short notice to discuss problems and ideas. Much of the success of this project is due to John’s insistence on thoroughness and quality work.

Special thanks to Jim Sochinski for making his knowledge of musical notation readily available and for helping to provide the detail necessary to produce a good system. This research had its beginning when Jim inquired into the possibility of building such a system as ENGRAVE.

My girlfriend, then fiance, then wife, and now soon to be the mother of my child, was a constant and much-needed support and shoulder during both hard times
and good times. Thanks for sticking with me, Dale.

I would like to thank Andy Tatem, Mike Stone, and Chris Philipoom for putting me up and for putting up with me for "a couple of weeks."

My parents went above and beyond the call of parental duty to provide financial support, sage advice and prayer. I deeply appreciate their sacrifice to help me finish. Also, many thanks to the members of the home fellowship that meets in their house for your unflagging support.

Most importantly, I would like to thank my Lord Jesus Christ who gave me the strength to "finish my course."
Table of Contents

Chapter 1 Introduction......................................................1

Chapter 2 Literature review..............................................9
  2.1 Printing.................................................................9
  2.2 Recognition..........................................................11
  2.3 General information................................................12

Chapter 3 Low Level Vision..............................................20
  3.1 Implementation.......................................................24
  3.2 Line Angle and Thickness.........................................24
  3.3 Labelling the Background.........................................30
  3.4 Horizontal Lines......................................................35
  3.5 Horizontal Line Deletion..........................................37
  3.6 Figure Segmentation................................................41
  3.7 Summary........................................................................45

Chapter 4 Visual Expert System.........................................54
  4.1 Implementation.........................................................57
  4.2 An Example...............................................................59
  4.3 Conclusion.................................................................66

Chapter 5 Spatial Planning for Musical Output....................72
  5.1 Planning Ahead..........................................................73
  5.2 Constraint Guidance..................................................74
  5.3 Description...............................................................77
  5.4 An Example...............................................................80
  5.5 Beat Units.................................................................94
  5.6 Rendering.................................................................97
  5.7 Constraint Guidance In Another Domain........................104
  5.8 Literature Review......................................................116
  5.9 Conclusions.............................................................118

Chapter 6 Discussion......................................................122
  6.1 Vision.......................................................................122
  6.2 Graphics.................................................................124
  6.3 Conclusion...............................................................125

REFERENCES.........................................................................126

Appendix A Visual Expert Rules.........................................131
  A.1 Rules about Note Durations..........................................131
  A.2 Rules about Measures................................................132
  A.3 Rules about Notes.......................................................134
    A.3.1 TO-FILL Rules for Notes........................................134
    A.3.2 IF-NEEDED Rules for Notes.....................................135
  A.4 Rules about Staves...................................................138
<table>
<thead>
<tr>
<th>Appendix</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>Spatial Planner Rules</td>
<td>139</td>
</tr>
<tr>
<td></td>
<td>B.1 Constraint Guidance Rules</td>
<td>139</td>
</tr>
<tr>
<td></td>
<td>B.2 Rendering Rules</td>
<td>140</td>
</tr>
<tr>
<td>C</td>
<td>Diagrams of Musical Characters</td>
<td>154</td>
</tr>
<tr>
<td>D</td>
<td>NOTES</td>
<td>171</td>
</tr>
<tr>
<td>E</td>
<td>Spatial Planner Graphics</td>
<td>175</td>
</tr>
<tr>
<td></td>
<td>E.1 Character Symbols</td>
<td>175</td>
</tr>
<tr>
<td></td>
<td>E.2 Beaming</td>
<td>175</td>
</tr>
<tr>
<td></td>
<td>E.3 Ties and Slurs</td>
<td>176</td>
</tr>
</tbody>
</table>
List of Figures

1) Block Diagram of the ENGRAVE System......................2
2) Copy of Handwritten Input to the ENGRAVE System........3
3) An Example of ENGRAVE's output..........................5
4) Output from [Hiller and Baker, 1957].......................13
5) Output from [Boker-Heil, 1972].............................14
6) Output from SMUT [Byrd, 1977].............................15
7) Bach's "Art of the Fugue" as displayed by SMUT...........16
8) Tenor part from Figure 7 as displayed by ENGRAVE.........17
9) Output from [Smith, 1973]..................................18
10) Edited output from Mockingbird............................19
11) Example thresholded image.................................25
12) Example thresholded image.................................26
13) Example thresholded image.................................27
14) Example thresholded image.................................29
15) Computation of line angle................................31
16) Line angle images corresponding to Figure 11............32
17) Line angle images corresponding to Figure 12............33
18) Line angle images corresponding to Figure 13............34
19) Horizontal lines ... in Figure 11.........................38
20) Horizontal lines ... in Figure 12.........................39
21) Horizontal lines ... in Figure 13.........................40
22) Musical figures remaining ... Figure 11...................42
23) Musical figures remaining ... Figure 12...................43
24) Musical figures remaining ... Figure 13...................44
25) The final, thinned image ... Figure 11....................46
26) The final, thinned image ... Figure 12....................47
27) The final, thinned image ... Figure 13....................48
28a) A sloppily written input..................................50
28b) Horizontal lines detected in the image..................51
28c) Horizontal staff lines removed from the image.........52
28d) The final thinned result..................................53
29) Example of a note frame...................................60
30) Diagram of input data for the visual expert system...61
31) A more difficult measure to understand....................68
32) A plan and the notation that it represents...............75
33) Skeleton plan for Figure 8................................82
34) Measure 10 has been moved to the second line...........83
35) Notes with longer duration take up more space...........85
36) All of the measures ... now fit on the line...............87
37) The final configuration......................................88
38) A beat-unit..................................................95
39) The same beat-unit ... in the skeleton plan...............96
40) Beat-units and sub-beat-units created.....................98
41) Beat-units combined.........................................99
42) Combining level-1 sub-beat-units.........................100
43) Final configuration..............................101
44) Conventions for spacing..........................103
45) An example of ENGRAVE’s output..................105
46) An example showing ties and slurs................106
47) An example showing beat-units and beaming........107
48) A plan ... aircraft carrier controller’s screen...109
49) The blocks found in the controller’s screen.......111
50) Initial placement of objects......................113
51) Resolution of objects off the screen...............114
52) Final screen......................................115
C-1) Whole note........................................155
C-2) Closed notehead..................................156
C-3) Open notehead.....................................157
C-4) Sharp.............................................158
C-5) Natural............................................159
C-6) Flat...............................................160
C-7) Staff line..........................................161
C-8) Repeat.............................................162
C-9) Repeat (beginning)................................163
C-10) Flag..............................................164
C-11) Treble clef.......................................165
C-12) Bass clef.........................................166
C-13) Figure "3" for time signatures..................167
C-14) Figure "4" for time signatures..................168
C-15) Quarter rest......................................169
C-16) Eighth rest........................................170
D-1) NOTES Example....................................174
E-1) Intersection of two degenerate conics..........178
E-2) The geometry for calculating slurs and ties.....180
List of Tables

1) Summary of results from Figure 30.....................67
2) Summary of results from Figure 31.....................69
3) Initially, all of the measures ... of equal size....89
4) Measure 5 expanded to its minimum size.............91
5) All measures are reduced by the same amount... ...92
6) All measure are reduced again.........................93
Chapter 1

Introduction

ENGRAVE is an expert system that understands musical notation. Its function is that of an engraver who extracts individual parts from a conductor's score. It is an expert system in that it captures the expertise of an engraver who must be able to read a score and generate notationally correct parts.

ENGRAVE's input is a grey-tone digital image of a handwritten musical score (one page at a time). Low-level image processing routines extract primitive features from the input image. These features are lines, circles, blobs, and curved lines. Descriptions of them are made available to a visual expert system which interprets these features to determine the pitch and duration of each note as well as some articulation information. The visual expert represents the separate musical parts in a music language called NOTES. NOTES is interpreted by a graphical expert that produces high quality graphical output on a laser printer or other high-resolution graphics output device. See Figure 1 for a block diagram of the system. Figure 2 shows part of a page from a
Figure 1
Block Diagram of the ENGRAVE system.
Figure 2
Copy of handwritten input to the ENGRAVE system.
handwritten score (the input for ENGRAVE) and Figure 3 shows the output of the C-part in the same piece of music.

A primary motivation behind the development of ENGRAVE is economics. Music copying and engraving is an art requiring the services of highly trained experts. The texts for copyists and engravers reveal the intricacies of producing readable work. For example, Ross [1970] gives almost three hundred examples of beams between two eighth notes. That these professionals are members of unions only compounds the situation. A publisher has a choice between having his work done in the United States or sending it overseas to take advantage of cheaper foreign labor. In Korea, printing shops are run like assembly lines where each person makes only one kind of figure. Printing performed in this manner is relatively inexpensive, but one must pay a price in the time that it takes to send the information between Korea and the United States.

In the vision phase, ENGRAVE uses knowledge about musical notation to interpret the input images. Traditionally, vision systems have used domain knowledge
at high levels while low-level vision processing has remained general purpose. During the initial development of ENGRAVE, the plan was to use general-purpose image processing techniques to detect musical primitives. A high-level expert system that operated on these features was to contain all of the knowledge about musical notation. Many different approaches were tried at the low level with little success. Finally, the decision was made to use knowledge about musical notation during low-level processing. The results obtained from this approach were significantly better than any other approach that was tried. As a result of this experience, we hypothesize that low-level knowledge is necessary for a vision system to perform successfully. ENGRAVE supports this claim.

ENGRSAVE uses spatial planning to produce its graphical output. Spatial planning is a knowledge-based technique that uses constraints and production rules. Unlike its predecessors, ENGRAVE is not simply a clever algorithm that prints music. Algorithmic approaches are limited in capability, and systems that use them cannot be easily modified or extended. ENGRAVE’s capabilities match or exceed those of its predecessors. More importantly, any notation that ENGRAVE cannot currently
produce can be added with ease. This is true for two reasons. First, spatial planning is a natural approach for musical output that captures the structure of the problem. Any modifications to the system will involve only a few rules and will not result in changes to the overall structure of the system. Secondly, ENGRAVE can be easily modified since rule-based systems are highly modular.

Not only is spatial planning a natural approach for printing music, it can be applied to other domains that involve placement and interactions of objects in space. Examples of this class of problems include production plant layout and formatting graphics screens for human-computer interaction. By applying spatial planning to a large problem, we have shown it to be a feasible approach for real-world problems.

The ENGRAVE system may be logically divided into three parts: low-level vision, a visual expert, and a graphics expert. Each part will be treated separately in its own chapter. In Chapter 3, we discuss the low-level processing that supports the visual expert and the use of domain knowledge at this level. The low-level system
detects primitives that are passed to the visual expert discussed in Chapter 4. This portion of the system makes use of frames and production rules as a representation of knowledge about musical planning and its application to musical output. A central element of the system is a musical representation language called NOTES. Although a full description of NOTES is not within the scope of this thesis, it is described briefly in Appendix D.

Most of the system is written in PROLOG [Roach and Fowler, 1983] with the exception of the low-level processing which is written in RATFOR [Kernighan and Plauger, 1976; Krusemark and Elliott, 1981] and runs on the GIPSY image processing system [Haralick et al, 1981]. An interface between PROLOG and Extended CORE graphics [SIGGRAPH, 1979; Ehrich, 1983] was developed for producing graphical output.
Chapter 2

Literature review

This chapter is an overview of previous work reported in the literature on printing and recognizing music. Each subsystem of ENGRAVE is discussed in a separate chapter, so discussion of theoretical background is reserved for the beginning of each chapter.

2.1 Printing

Music printing programs have existed at least as early as 1965 when Hiller and Baker [1965] reported on a system that used a music typewriter for input. Figure 4 shows a later system by Boker-Heil [1972] is shown in Figure 5. This system as well as many of the others in existence uses an alphanumeric language as input. Generally, these languages are difficult to read and to enter. This can lead to input errors. A natural way for humans to enter data is needed.

Besides input language, both of the systems mentioned have two major downfalls -- spacing and graphics. The graphics in both systems are crude to the
point that some figures are not recognizable. Although it is not the most difficult problem posed by this domain, it is an important consideration since we would like attractive output. Proper spacing aids a performer by providing a visual clue to note duration. If a piece is properly spaced, a performer will not have to expend extra effort deciphering the notation [Roemer, 1973, p. v]. Gomberg [1977] proposed a data structure and an algorithm to provide proper spacing but neither showed any output nor indicated that he had implemented his techniques.

The SMUT system [Byrd, 1974] produces the best results of the early systems. Figure 6 (from [Byrd, 1977]) shows some output. Output from a more recent version of SMUT is shown in Figure 7 (taken from [Hofstadter, 1979]). Figure 8 shows the output of ENGRAVE on the same part.

Other printing work has been reported in connection with musical editors. These systems generally do not include all aspects of musical notation but leave some tasks to the user. An early musical editor that produced some fine output is described in [Smith, 1973] and an
example can be seen in Figure 9. The MOCKINGBIRD system [Maxwell and Ornstein, 1984] provides the best graphics the author has seen but leaves many notational details for the user. Figure 10 (from [Roads, 1981]) shows some output after editing to specify spacing, voicing and beaming.

Some graphics work has been done for music transcription systems [Moorer, 1977; Piszczalski and Galler, 1977]. The emphasis in these systems is on interpreting a waveform rather than on graphical output.

2.2 Recognition

All previous work relating to the automatic recognition of musical notation has dealt with printed music only. The best known of these systems is DO-RE-MI [Prerau, 1971; 1975]. DO-RE-MI is a pattern recognition system that classifies the objects in an image according to their size. This technique works well for printed music where figures are uniformly printed. In handwritten music, this technique is not reliable since the appearance of a given type of note may vary widely within a single piece.
2.3 General Information

For information about the rules of conventional musical notation, [Roemer, 1973] provides motivation for correct notation and is oriented towards copyists. [Stone, 1980] is intended to be a standard for unusual notations that occur in modern music but also covers some of the basic conventional notation. A book for engravers, [Ross, 1970], is a clear and well-organized source presented in a form that is easy to convert to a computer program.
Figure 4
Output from [Hiller and Baker, 1957].
Figure 6
Output from SMUT [Byrd, 1977].
Figure 7
Bach's "The Art of the Fugue"
as displayed by SMUT.
TRIO FOR WOODWINDS
FOR MADELEINE AND DURIS MILHAUD

I.

OBOE

CLARINET IN Eb

BASSOON

POCHISS. RIT. A TEMPO

POCO Stringendo PIU CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.

POCO CRESC.
Figure 10
Edited output from Mockingbird.
Chapter 3
Low Level Vision

Essential to any vision system is some kind of early processing that can provide information in a usable form to higher level routines. In ENGRAVE, low-level routines detect primitives that are passed on to the visual expert. Our primitives are the commonly occurring shapes in music: circular blobs (for closed noteheads), circles (open noteheads), horizontal lines, line segments (not horizontal), and arcs. The location and orientation data for each primitive are made available to the visual expert system through a file. Currently, computer vision is thought of as being divided into two areas of study: low-level and high-level computation. Under the current paradigm, low-level vision consists of general-purpose operators that extract information from images. High-level processing interprets this information using knowledge about its domain [Ballard and Brown, 1982, pp. 2-6].

During the initial phase of development, ENGRAVE was designed under the current paradigm from computer vision. Many of the standard techniques for line detection were
tried with little success. These techniques could only be made to work consistently for a small class of images. This is particularly true of handwriting where there is large variation between one person’s handwriting and another’s. When the system parameters are adjusted for one set of images, we find that the operators do not work well for another set. It is difficult and often impossible to adjust the parameters so that a system will work reliably for all classes of images in a single domain. Eventually, we concluded that general-purpose techniques would not work and that explicit use must be made of domain knowledge. Concerning the segmentation of curves, Rosenfeld and Johnston speculate that "if the user has special knowledge about his class of curves, he might well be advised to devise his own segmentation procedure taking this knowledge into account" [Rosenfeld and Johnston, 1973]. An expert system for low-level image segmentation is described in [Nazif and Levine, 1984]. The knowledge in their system is about low-level image processing. We propose not only the use of image processing knowledge, but of domain knowledge as well. When domain knowledge was used at the low-level in ENGRAVE, the results obtained were remarkably better than any results obtained under the current computer vision
Paradigm.

The horizontal staff lines in a musical score provided a particularly difficult problem which was solved by incorporating knowledge about musical notation into the low-level process. Staff lines intersect most figures in a piece of music. Should the intersection pixels be labelled as part of the staff line or as part of a figure? In reality, these pixels are members of both the staff and the intersecting figure. Most labelling schemes, however, can only label these pixels with one label. Prerau [1971] simply removes the staff lines from the image thereby causing notes and other figures to be broken into pieces. These pieces can be easily reassembled because of the regularity of printed music. With handwritten music, we cannot make use of such regularity. Instead, we make use of knowledge about the domain first to detect horizontal lines and then to remove them intelligently so as not to lose any information.

The Hough transform has been used successfully in many applications to detect straight lines, circles, and ellipses [Duda and Hart, 1972; Tsuji and Matsumoto,
1978]. However, the Hough transform has only been used to detect a few figures at a time. On a single page of a conductor's score, there are more than 50 staff lines in addition to the lines for note stems, bar lines, etc.

In this application, the Hough space is too noisy for us to be able to detect all of the figures reliably. Less prominent figures are obscured by the stronger ones. Watson et al. [1984] have developed a line tracker that functions well in a similar domain. The tracker uses grey-tone values to determine whether to assign a pixel to a line or to start a new line. Because of the many line intersections, the line tracker tends to turn corners or to break notes and other musical figures apart.

Closed noteheads can be detected as convex blobs. A spoke filter [Minor and Sklansky, 1981] was tried, but it detected the line intersections as blobs as well as the noteheads. In some cases, the intersections were stronger than the notehead. Haralick [1974] gives a circularity measure that is simple and works well when used with a threshold. The region detection portion of Watson's line tracker also works well detecting these noteheads.
The input image is a grey-tone digitization of one page of a score. This image is converted to a binary image using the double-adaptive threshold (DAT) of Watson et al. [1984]. Figures 11 through 13 show some samples of an image after thresholding.

After thresholding, some preliminary processing is performed to determine line angle and thickness. Using this information and knowledge of our domain, horizontal lines are detected and removed from the image. Once the staff lines are removed, we are left with an image where the notes and other figures are all isolated. These figures are broken into their component line segments. All of the information from this low-level system is passed to the high-level expert through several files.

3.2 Line Angle and Thickness

Line direction and thickness are important properties that need to be computed. Line direction gives essential information to help determine horizontal line
Figure 11
Example thresholded image from the handwritten input. An eighth note beamed to a half note. The bottom of the eighth note tail coincides with the staff line.
Figure 12

Example thresholded image from the handwritten input. A quarter note followed by a half note. The top of the half note coincides with the top staff line.
Figure 13

Example thresholded image from the handwritten input. Quarter rest, natural, quarter note with dot above it. Note break in staff line next to rest; also, the natural has one disconnected tail part, a staff line through its middle, and its top horizontal line coincides with a staff line.
pixels and thickness can aid in the identification of figures that cannot be otherwise identified. For example, many closed noteheads in our images are so small that they cannot be detected reliably by the circularity measure. Our eyes, though, can discern their presence (See Figure 14). These figures can be properly identified later by observing that one end of the stem is thicker than the other.

To compute this information, an odd-sized circular window is passed over the image. If the center pixel is black, then the window is scanned to find the black pixel farthest from the center. The angle of this point and distance from the center define a vector that approximates the line direction at the center pixel. To compute line thickness, the window is scanned along a line perpendicular to the direction vector until a white pixel or the edge of the window is encountered.

In many cases, a straight line between the farthest pixel and the center will cross white space. If this occurs, the angle will not correspond to the line direction. To prevent this from occurring, some pixels are deleted from the window before computing the angle.
Figure 14

Example thresholded image from the handwritten input. Using a magnifying glass, we were able to determine that this result was not unfaithful to the original. Digitization of original page was 1200 x 800 (horizontally, therefore 100 scan lines per inch). The blob to the left was meant as a sharp, the figure to the right is a quarter note. The quality of this input is not typical.
Any black pixel that (1) is not the central pixel, and (2) is adjacent to a white pixel that is closer to the center than itself, is deleted. This process is repeated until no more pixels are deleted. Figure 15 illustrates this process. After deletion, the line angle and thickness are completed.

To detect horizontal lines, the radius of the window should be set to a large value -- at least half the width of one staff in the image. This will force pixels at the intersection of a staff line and a figure to have a horizontal line angle. Figures 16 through 18 show the line angle images computed from the examples.

3.3 Labelling the Background

Many figures such as half notes and accidentals have "holes" that if detected can be useful in identifying the figures. To this end, the image background is segmented by a simple labelling process as follows:

- If an unlabelled white pixel has no labelled white 8-neighbors, then give it a unique label.
- If a labelled white pixel has a differently labelled 8-neighbor, then label the original pixel with its neighbor's label.
Figure 15
Computation of line angle.
Figure 16
Line angle images corresponding to Figure 11.
Figure 17
Line angle images corresponding to Figure 12.
Figure 18

Line angle images corresponding to Figure 13.

34
This process is continued until no more labelling is done.

The difference between the "holes" for which we are looking and the other white areas in the image is that the holes are significantly smaller and they are located inside of musical figures.

3.4 Horizontal Lines

The most important figure to deal with in music is the horizontal line. Nearly every figure in a piece of music intersects a staff line. Our strategy is to detect horizontal lines and then remove them in a manner that does not break up the other figures. We use angle information and pixel adjacency to determine whether a pixel is a line pixel. The rules are as follows:

A black pixel is marked as a horizontal line pixel if
- it is to the left or right of a line pixel; or
- it is horizontal and it 8-adjacent to a line pixel.

If a pixel is horizontal and has no marked adjacent pixels then mark it with a new label.

A white pixel is a horizontal line pixel if it has a marked pixel to its left and to its right. (This rule allows for the possibility that a line
may have a 1-pixel break.)

Often, it is impossible to tell by its neighbors whether or not a pixel is part of a horizontal line. A pixel may even be both part of a horizontal line and part of some other figure. These pixels are given a special label that indicates that these pixels are "questionable." That is, it is uncertain whether these pixels are part of a horizontal line or not. A list of these pixels is stored in the output file for use at higher levels. Here are the rules for questionable pixels:

- A pixel is questionable if it is not horizontal and is above or below a line pixel.

- A pixel is questionable if it is above a questionable pixel and below another questionable pixel.

- A questionable pixel above a line pixel is relabelled as part of that line if none of the 8-connected pixels above it are black.

- A questionable pixel below a line pixel is relabelled as part of that line if none of the 8-connected pixels below it are black.

As pixels are labelled, they are added to a list of pixels that all have been marked with the same label. A list is also kept of questionable pixels.
Some pixels may be elements both of a line and the questionable list. These will generally be pixels that are both part of a line and some other figure.

After all the line pixels are identified, there usually are adjacent regions that have been labelled differently. With the exception of the pixels that have been labelled "questionable," these regions are part of the same line and are merged.

A least-squares fit is used to determine the equations of the lines and these coefficients are made available in a file along with the list of questionable pixels. We can identify staff lines by the fact that they are almost as long as the width of a page and that the other horizontal lines are much shorter. Figures 19 through 21 show the line pixels that were detected in the original images.

3.5 Horizontal Line Deletion

Once the horizontal lines have been detected, they are removed from the image leaving the other figures
Figure 19
Horizontal lines, staff lines,
extracted for image in Figure 11.
Figure 20
Horizontal lines, staff lines, extracted for the image in Figure 12.
Figure 21
Horizontal lines, staff lines, extracted for image in Figure 13.
intact. This is not done by deleting line pixels, but by marking the pixels that should remain. The rules for marking follow:

Mark a black pixel if
- it is not a line pixel;
- it is not horizontal but the pixel above or below it is horizontal;
- it is a line pixel and is above or below a marked pixel; or
- it has marked pixels to its left and to its right.

When this process is completed, the musical figures will have been isolated. Figures 22 through 24 show the results of line deletion.

3.6 Figure Segmentation

Once the figures have been isolated, they are segmented into their components and this information is made available to the visual expert. First, the figures are thinned using the algorithm given by Arcelli and DiBaja [1981] in order to obtain chain codes. Their thinned figures are segmented at junction points. Junction points are defined as pixels where three or more segments having the same label meet. After segmentation, we are left with the chain codes of all of the arcs in
Figure 22
Musical figures remaining once the horizontal staff lines have been intelligently removed from the image in Figure 11.
Figure 23
Musical figures remaining once the horizontal staff lines have been intelligently removed from the image in Figure 12.
Figure 24

Musical figures remaining once the horizontal staff lines have been intelligently removed from the image in Figure 13.
Each individual arc is further segmented by breaking the arcs at curvature maxima as described in Rosenfeld and Johnston [1973]. Figures 25 through 27 show the results of thinning and segmentation.

Descriptions of all the features detected are output to a file. This file is loaded by the expert system which uses it to identify the figures.

3.7 Summary

The results obtained using these techniques were quite good -- certainly far above expectations. In most cases, the system was able to correctly isolate and segment the figures, and enough information was made available to identify them correctly. In portions of the score where handwriting was sloppy, some information was lost. Figures 28a through 28d show successive results of processing a sloppy image.

Originally, our plan was to use general purpose image processing techniques as early processing for a visual expert system. It was found, however, that general techniques could not be made to work well.
Figure 25
The final, thinned image corresponding to the image in Figure 11. Other information including where blobs (noteheads) exist and their extent, candidate horizontal lines connecting music figures, etc. are passed to the high level expert.
Figure 26
The final, thinned image corresponding to the image in Figure 12. Other information including where blobs (noteheads) exist and their extent, candidate horizontal lines connecting music figures, etc. are passed to the high level expert.
Figure 27
The final, thinned image corresponding to the image in Figure 13. Other information including where blobs (noteheads) exist and their extent, candidate horizontal lines connecting music figures, etc. are passed to the high level expert.
Domain knowledge had to be incorporated at the low level to achieve satisfactory results. This is an important result since vision systems typically do not explicitly incorporate knowledge during low-level processing. Traditionally, world knowledge is used only for "high-level vision" and low-level processing is done using general purpose mathematical models.

Much of the lower-level knowledge in this system is procedural -- not declarative. Where it was used, the rule-based approach gave the system the flexibility it needed to do its job. Several of the processes in this low-level system were implemented under the old general-purpose paradigm. The system would benefit from also making these processes rule-based and flexible. The importance of this approach is not that we have achieved the ultimate system but that a dramatic improvement resulted from adding world knowledge to low-level processing. This suggests that a vision system must incorporate world knowledge at all levels to work successfully.
Figure 28a
A sloppily written input. A couple of slurred dotted eighth-sixteenth note pairs. The second sixteenth note has a sharp accidental to its left. The second tail on each sixteenth note coincides with a staff line; the visual cue showing that it is a sixteenth note is therefore obliterated.
Figure 28b

Horizontal lines detected in the image
Figure 28c

Horizontal staff lines removed from the image.
Figure 28d

The final thinned result. The open noteheads on the second dotted eighth-sixteenth will be easily eliminated by the high level expert system. Determining that the second note of each couple is a sixteenth note, however, can only come from contextual inference and background knowledge.
Chapter 4

Visual Expert System

After low-level processing has extracted the primitive features from a piece of music, this information is passed to a visual expert system. The expert system, written in PROLOG, uses knowledge about musical notation to identify each of the figures in the score based on the primitives and the spatial relationships between them. These primitives are the lines, holes and blobs that are detected by the low-level system. These features are made available to the visual expert through several files which are loaded by PROLOG. For each primitive, there is information about its type, its location and its orientation. The goal of the system is to determine pitch and duration for each figure in the image. A "figure" is a collection of primitives that have some relationship to each other. Usually, these primitives are connected. When pitch and duration have been determined, the information is output in a musical representation language called NOTES (briefly described in Appendix D).

The visual expert's problem is to classify the
various figures in handwritten music in a manner consistent with the rules of musical notation. This has been referred to as a "consistent labelling problem", and tree-searching methods have been proposed to solve these problems [Haralick and Elliott, 1980]. Music presents a complex domain and a very large search space. Standard search techniques, even with clever speed-ups, are still blind and may spend time investigating unlikely possibilities. Use of frames provides a means of limiting the search space and directing the search to produce a more focused problem solution.

ENGRAVE builds a hierarchy of frames that corresponds to the structure of musical notation. Lines contain measures and measures contain notes. Slots of higher level frames in the hierarchy may contain the names of lower level frames. For example, line frames have a list of measures as one of their slots. Communication among levels of the hierarchy is achieved through production rules. These rules are "attached" to the slots of the frames and to the frames themselves. One or more "IF-NEEDED" rules are associated with each slot and are invoked when a value is needed for that slot. When an IF-NEEDED rule succeeds, the value
that was computed is added to the slot. "TO-FILL" rules are associated with a frame and contain the slots that are required to be filled for a frame. TO-FILL rules automatically fire when a frame is created. Both TO-FILL and IF-NEEDED rules are triggered "on demand." No values are computed until they are needed. Once they are computed, they are stored in slots so that they need not be computed again. The visual expert is patterned after CENTAUR [Aikins, 1980], a lung disease expert that uses frames with attached rules to make diagnoses about lung diseases based on data from tests.

Each measure frame must satisfy a constraint that the sum of the durations of the notes in a measure must add to the number of beats in the measure. If this constraint is not satisfied, the system will backtrack and try new rules for various figures. Each time the constraint is checked for a measure, an evaluation function is run. If the sum of the note durations for that measure is equal to the number of beats in the measure, then the evaluation function returns zero. Otherwise, the evaluation function returns a value that indicates the degree that the constraint is violated. When a set of notes is suggested that does not satisfy
the constraint, ENGRAVE will backtrack until the constraint is satisfied or until all solutions for the measure have been exhausted. If no solution is found that completely satisfies the constraint, the solution with the lowest value returned from the evaluation function (the solution that is "closest" to being correct) is used for the measure.

The system provides the capability for the application of constraints to any frame. These constraints are checked when the TO-FILL rule for a frame has completed running. While ENGRAVE only contains one constraint, other constraints may be added to measures and to other objects as the knowledge base grows.

4.1 Implementation

The frames and production rules in this system are implemented in PROLOG using the techniques described in [Chester, 1981]. Production rules are all of the form:

```
(RULE ... args ...) IF (condition-1)
  (condition-2)
  ...
  (condition-m)
  (action-1)
```
To retrieve values from a frame slot, we use a Prolog clause called RETRIEVE. RETRIEVE first tries to get the value directly from the slot. If the slot does not contain a value, then RETRIEVE calls the IF-NEEDED rules that are attached to the slot. If there are no rules attached to the slot or if they fail when they are called, then RETRIEVE will return the default value for the slot. It is expected that the IF-NEEDED rule will add the value it computes to the slot.

The goal of the system is to find the figures in a piece of music and to label them correctly using their features as information. Frames play an important part in this process because they specify the features needed to give a figure a certain label. In addition, the presence of a frame suggests plausible combinations of features and thereby provides a focus for program execution. The program begins by checking for important distinguishing features. If one or more of these features is found associated with a figure, this suggests a class of possible objects for the figure and causes a frame to be
instantiated. Now the system begins to fill out the other slots in the frame. Each frame is assigned a label based on the features that are detected. In cases where there is ambiguity or where a frame is not completely filled out, there may be several possible labels for a frame. When there is more than one possible label, these labels are resolved using global constraints. Figure 29 shows a frame for a dotted eighth note.

4.2 An Example

This section shows the processing sequence of ENGRAVE as it operates on a one-measure image. A diagram of the image is shown in Figure 30. The rules used in this expert system are listed in Appendix A.

After loading the primitives file, the program is started simply by creating a staff frame. This triggers the TO-FILL rule for staves which tries to fill the "lines" slot and "measures" slot for the staff. Since neither slot has a value, the IF-NEEDED rules for these slots will fire.

The rule that fills the "lines" slot of a staff
(note G45 notehead G12) ; blob for a notehead
(note G45 stem G15) ; stem connected to notehead
(note G45 flags (G22)) ; one flag connected to stem
(note G45 dots (G13)) ; one dot in the vicinity of
; and following the notehead
(note G45 duration 2.666) ; Durations are represented by
; the denominator of the
; note-type, in this case,
; a 1/8 note has been modified
; by the dot.

(note G45 pitch (C 5))
;
(blob G12 50 103 23) ; notehead
(curve G15 ((51 106)(52 106)(52 107)(52 108)...(51 122))) ; stem
(curve G22 ((52 122)...(40 130))) ; flag

Figure 29
Example of a note frame
Figure 30

Diagram of input data for the visual expert system.
looks for five staff lines that are adjacent (no staff lines between them). Staff lines are distinguished during low level processing from other horizontal lines by the fact that staff lines are very much longer. In this example, there are only five staff lines in the image, so finding them is trivial. In a larger image, ENGRAVE would start with the five highest staff lines in the image.

When we say that a rule "looks for" a staff's lines, we mean that it simply performs a RETRIEVE operation. This is how "computation on demand" works. When the value of a slot is needed, it is simply computed. If the value has been computed, it is simply returned. If no value has been computed, the IF-NEEDED rule for the slot is triggered and computes the value. In our example, the TO-FILL rule attempts to retrieve the value of the lines slot. Since there is no value for the slot, the IF-NEEDED rule fires and makes a conclusion about which primitives are the staff's lines.

The TO-FILL rule for staves also looks for the measures that are part of the staff. The IF-NEEDED rule for measures repeatedly creates a measure frame until it can find no more measures in the staff. Each time a
measure frame is created, the TO-FILL rule for measures is run. This rule looks for a measure’s bar lines and its notes and then checks the notes against a constraint. In our example, ENGRAVE labels the primitives 1 and 25 as bar lines.

The first primitive that is found is the hole numbered 13. The hypothesis is made that this primitive is the notehead of the current note. The "notehead" slot in the note frame now contains a reference to hole 13. The system knows there are only a few kinds of musical figures that have open noteheads: whole notes, half notes, sharps and naturals. The first primitive that the TO-FILL rule looks for is a stem. If there is a stem, then we know that the note is a half note. Otherwise, the system must go on and try another rule. In this case, the note does have a stem: the figure numbered 16. The system needs only to determine whether there are any dots modifying the duration of the note. None are found and the IF-NEEDED rule for dots fails and ENGRAVE must include that this note is a half note with the usual two beat duration.

The next primitive that ENGRAVE finds is a blob (3).
This is most likely a note with a closed notehead (such as a quarter note or an eighth note) or a whole or half rest. The TO-FILL rule, if it finds a blob that has not already been labelled, assigns the blob as the notehead and then retrieves the note's duration. This triggers an IF-NEEDED rule for note duration. If a note has a closed notehead, ENGRAVE looks for a stem. The notehead numbered 3 does have a stem so we know that the note is not a rest. At this point, the system has to look for flags and or dots to determine the duration of the note. The IF-NEEDED rule for flags fails because there are no flags attached to the stem. The rule for dots concludes that there is a dot (7). Therefore, ENGRAVE concludes that this note is a dotted quarter note (1-1/2 beats).

The blob numbered 8 is the primitive used for the next note. The reasoning process proceeds exactly as above until ENGRAVE looks for flags. This note has a flag (15) attached to its stem. From the figure, it may not seem that 15 and the note's stem are connected but ENGRAVE has knowledge to account for the fact that some of the connecting pixels were included as part of the staff line. There is a rule that understands that these two features are connected because they both are
connected to a staff line and they are placed in a relationship that suggests a stem and a flag.

None of the remaining primitives in the image have the correct relationships to each other to be labelled as musical figures. (One exception is the tie formed by 10 and 11 -- no knowledge of ties is currently present in the system.) Each of these figures receives a NIL label indicating that no interpretation has been assigned.

At this point the notes are checked against a constraint. The constraint requires that the sum of the durations of the notes in a measure must equal the number of beats in the measure. Since this measure is in 4/4 time (a fact specified by the user), the sum of the note durations must add up to four beats. This measure contains a dotted quarter note (1-1/2 beats), an eighth note (1/2 beat), and a half note (2 beats). The sum of these durations is indeed four beats so the constraint is satisfied. At this point all of the notes in the current measure are known and the TO-FILL rule for the measure is satisfied. The rule for the staff is also satisfied since there are no more measures in this image. On a larger image, the TO-FILL rule for a staff would create a new
measure frame and proceed to process the next measure.

The final conclusions in this example are summarized in Table 1. ENGRAVE has built a small frame structure showing a line that contains a single measure. The measure contains three notes. The first note is a dotted quarter note (1-1/2 beats) with a pitch of "A". The second is an eighth note "E" and the third is a half note "E". Figure 31 shows a messier measure. ENGRAVE's conclusions are summarized in Table 2.

4.3 Conclusion

ENGRAVE can understand most conventional notation if it is reasonably legible. Even when the handwriting is somewhat sloppy, as in Figure 31, ENGRAVE can understand the notation. In Figure 31, ENGRAVE mistakes the natural before the second note for a sharp. This is due to the fact that the key signature is not known to the system. ENGRAVE can tell when a horizontal line is being obscured by a staff line. In Figure 30, the system determines that the second note is an eighth note even though the flag is closer to another note. Because ENGRAVE knows that staff lines obscure horizontal lines and because the system
Table 1
Summary of results from Figure 30

<table>
<thead>
<tr>
<th>Frame</th>
<th>Slot</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Staff A</td>
<td>Lines</td>
<td>20, 21, 22, 23, 24</td>
</tr>
<tr>
<td></td>
<td>Measures</td>
<td>B</td>
</tr>
<tr>
<td>Measure B</td>
<td>Staff</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Bar-lines</td>
<td>1, 25</td>
</tr>
<tr>
<td></td>
<td>Notes</td>
<td>C, D, E</td>
</tr>
<tr>
<td>Note C</td>
<td>Measure</td>
<td>B</td>
</tr>
<tr>
<td>Dotted quarter note</td>
<td>Notehead</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Stem</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Dot</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>Duration</td>
<td>1 1/2 beats</td>
</tr>
<tr>
<td></td>
<td>Pitch</td>
<td>A natural</td>
</tr>
<tr>
<td>Note D</td>
<td>Measure</td>
<td>B</td>
</tr>
<tr>
<td>Eighth note</td>
<td>Notehead</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>Stem</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>Flag</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>Duration</td>
<td>1/2 beat</td>
</tr>
<tr>
<td></td>
<td>Pitch</td>
<td>E natural</td>
</tr>
<tr>
<td>Note E</td>
<td>Measure</td>
<td>B</td>
</tr>
<tr>
<td>Half note</td>
<td>Notehead</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>Stem</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>Duration</td>
<td>2 beats</td>
</tr>
<tr>
<td></td>
<td>Pitch</td>
<td>E natural</td>
</tr>
</tbody>
</table>

Unaccounted for: 10, 11
A more difficult measure to understand.
<table>
<thead>
<tr>
<th>Frame</th>
<th>Slot</th>
<th>Value(s)</th>
<th>Frame</th>
<th>Slot</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Staff A</td>
<td>Lines</td>
<td>1, 2, 3, 4, 5</td>
<td>Note E</td>
<td>Measure</td>
<td>B</td>
</tr>
<tr>
<td></td>
<td>Measures</td>
<td>B</td>
<td></td>
<td>Notehead</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Accidental</td>
<td>note D</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Stem</td>
<td>16</td>
</tr>
<tr>
<td>Measure B</td>
<td>Staff</td>
<td>A</td>
<td></td>
<td>Flag</td>
<td>17+18+20</td>
</tr>
<tr>
<td></td>
<td>Bar lines</td>
<td>6, 35</td>
<td></td>
<td>Duration</td>
<td>1/2 beat</td>
</tr>
<tr>
<td></td>
<td>Notes</td>
<td>C, E, F, G</td>
<td></td>
<td>Pitch</td>
<td>D sharp</td>
</tr>
<tr>
<td>Note C</td>
<td>Measure</td>
<td>B</td>
<td>Note F</td>
<td>Measure</td>
<td>B</td>
</tr>
<tr>
<td>Quarter rest</td>
<td>Upper leg</td>
<td>7+8</td>
<td></td>
<td>Notehead</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>Lower leg</td>
<td>9</td>
<td></td>
<td>Stem</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>Middle bar</td>
<td>none</td>
<td></td>
<td>Flag</td>
<td>17+18+20</td>
</tr>
<tr>
<td></td>
<td>Duration</td>
<td>1 beat</td>
<td></td>
<td>Duration</td>
<td>1/2 beat</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Pitch</td>
<td>D sharp</td>
</tr>
<tr>
<td>Note D</td>
<td>Measure</td>
<td>B</td>
<td>Note G</td>
<td>Measure</td>
<td>B</td>
</tr>
<tr>
<td>Sharp</td>
<td>Center</td>
<td>11</td>
<td></td>
<td>Notehead</td>
<td>28, 29</td>
</tr>
<tr>
<td></td>
<td>Left leg</td>
<td>10</td>
<td></td>
<td>Stem</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>Right leg</td>
<td>12</td>
<td></td>
<td>Duration</td>
<td>2 beats</td>
</tr>
<tr>
<td></td>
<td>Upper bar</td>
<td>none</td>
<td></td>
<td>Pitch</td>
<td>D sharp</td>
</tr>
<tr>
<td></td>
<td>Lower bar</td>
<td>none</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Modifies</td>
<td>note E</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Unaccounted for: 19, 21, 22, 25, 26
knows the kinds of notes that should have flags, ENGRAVE is able to correctly determine the duration of both the second and the third notes. ENGRAVE is also able to use its knowledge to determine that the two curves at the beginning of the measure in Figure 31 form a quarter rest in spite of the fact that the middle bar is obscured by the staff line.

In the present implementation, the visual expert system is separate from the low-level system. Once the low-level system has run, it produces files for the visual expert and control is passed to the expert system. As mentioned in Chapter 3, knowledge is used in the low-level system as well as in the visual expert system. Ideally, these two levels should be integrated. In this way, all of the knowledge in the vision system could be available to every level of the system. In addition, when a high level rule fails, the system could backtrack into the lower levels of the system if necessary. Some work is needed to build such a system. First, a control structure is needed to handle communication between the different levels of the system. Secondly, knowledge has different meaning at different levels. A method must be determined for representing
knowledge at these different levels and for converting between them.
Chapter 5
Spatial Planning for Musical Output

An important kind of planning involves the management of objects in space and their interactions and is called spatial planning. Spatial planning is a resource allocation problem. It is different from the usual robot planning in Artificial Intelligence, however, because it deals with geometric problems. Typically, A.I. planners have dealt with dependencies of tasks and with task sequencing, not with reasoning about geometry. Spatial planning must also address the interactions of objects in space and the methods needed to place them properly.

Creating printed music is a spatial planning problem characterized by a large solution space. To get an idea of the potential size of the problem see Figure 3. Figure 3 shows a simple piece of music that has 139 notes and 50 measures. Each note has a wealth of information associated with it including location, pitch, duration, stem direction, and spacing information. All of these properties must be computed for every note, measure and staff so that each object is properly placed on a
sheet of paper. Ross gives over three hundred rules just for beaming two eighth notes together [Ross, 1970, p. 104]. Because of the size of the space, some planning must take place in order to avoid having to redo large amounts of work. This planning activity must be guided by knowledge of musical notation and by knowledge of how musical figures interact in space.

This chapter reports the investigation of spatial planning for printing music. The output stage of ENGRAVE takes a computer representation of a piece of music and generates a high quality output. ENGRAVE is a hierarchical, spatial planner that uses a process called "constraint guidance" to plan the spatial layout and correct notation of a piece of music.

5.1 Planning ahead

Before producing a manuscript, a copyist must plan ahead to insure that all of the measures fit on a page and that they line up properly on the staff [Roemer, 1973]. Engravers plan by actually drawing sketches on their plates to mark the positions of the notes [Ross, 1970]. Notice in Figure 3 that the last measure on each
line (other than the last) ends exactly at the end of the line. In addition, the notes have differing amounts of space between them depending on their duration -- longer notes take more space. To determine the placement of notes on a line so that they are properly spaced and still fill the line, ENGRAVE plans ahead using a technique called "constraint guidance."

ENGRAVE's equivalent to the engraver's rough sketch is a hierarchical plan. The hierarchy represents the hierarchical structure of music. At the highest level are pages. To print a page, one must print the staff lines that it contains. Likewise, to print a staff line, its measures must be printed. ENGRAVE computes the structure of the plan and computes only the details that are needed to make sure that the plan will work. Once this information is determined, all of the other parameters needed to print the music can be calculated when the music is actually printed. See Figure 32 for an example of a plan the notation that it will eventually produce.

5.2 Constraint guidance

Spatial planning can be viewed as a constraint
A plan and the notation that it represents.
satisfaction process. A good indication of this can be seen by noting the similarity between Gaschnig's [1974] formalization of constraint satisfaction problems and Eastman's [1973] characterization of the spatial planning problem.

ENGRAVE uses constraints to guide its spatial planning process. A constraint is defined as a relation between plan variables [Stefik, 1981] or a requirement that must hold true of the variables. In ENGRAVE, the constraints are the required relationships between figures that, when satisfied, will result in correct musical notation. We use the term "constraint guidance" because the constraints are the goals of the system. Because the constraints constitute goals, it can be said that the constraints guide the spatial planning process.

Constraint guidance is a generate and test procedure. (Stefik suggests that all planning is generate and test.) ENGRAVE starts with an initial plan and tests the plan to see whether it satisfies the constraints. If any constraint is violated, a corrective action is taken that modifies the plan, and the cycle is repeated until all constraints are satisfied.
Every constraint has a corrective action associated with it that is taken if the constraint is not satisfied. Each of these constraint-action pairs forms a forward-chaining production rule with each having the form IF (NOT constraint) THEN corrective-action. Using this format, it is possible to build a knowledge base about any domain that requires resource allocation. Later, we will demonstrate this by showing that constraint guidance can be used to format screens for human-computer interaction.

5.3 Description

Execution of ENGRAVE proceeds in three stages: initialization, constraint guidance and rendering. During the initialization stage, a skeleton plan is built that the constraint guidance stage uses. Measures are assigned to staff lines at a rate of ten measures per line. The constraint guidance stage uses this skeleton plan for the constraint guidance process. When no more constraints are violated, the rendering stage computes the values necessary to display the music as indicated in the plan. Constraint guidance and rendering are knowledge based
processes driven by rules.

There are really two kinds of rules in ENGRAVE. They are called rendering rules and constraint guidance rules after the stage of the system that uses them. The main difference is that constraint guidance rules guide the planning process and rendering rules compute the details of a particular plan.

Constraint guidance rules are the constraint-corrective action pairs mentioned above. These rules are concerned with planning how to lay out the piece of music on a sheet of paper. They deal with the higher levels of the plan hierarchy -- mainly staff lines and measures. The constraint guidance rules are used by ENGRAVE listed below:

1) If there exist two adjacent measures containing a whole measure rest or a multiple-measure rest, then combine them into a single measure containing a multiple measure rest.

2) If the last page in the sequence contains more than its maximum number of lines then create a new page following it.

3) If there exists a page with more than its maximum number of lines, then move the last line on the page to the next page.

4) If, for the last line in the piece, the sum of the minimum sizes of its measures exceeds the amount of space available for the line then
create a new line following it.

5) If there exists a line such that the sum of the minimum sizes of its measures exceeds the amount of space available for that line then move the last measure down to the next line.

6) If there exists a line, not the last line, such that the sum of the sizes of its measures is less than the amount of space available on the line then expand the measures uniformly to fill the line.

7) If there exists a line such that the sum of the minimum sizes of its measures is less than or equal to the space available but the sum of the actual sizes is greater than the space available then reduce all of the measures uniformly so as to make the sum of their sizes equal to the space available.

8) If there exists a measure whose size is less than its minimum size then expand the measure to its minimum size.

9) If there exists a beat-unit that is not full and there exists a flagged note that is adjacent to one of the notes in the beat-unit but is not a member of any beat-unit itself then add the note to the beat-unit. (Note: A beat-unit is a group of notes that are beamed together and treated as a unit. Typically, the notes in beat-unit will sum to one beat. By "flagged note" we mean that the note has duration less than a quarter note and will have a tail or a beam attached to its stem.)

10) If there exists a flagged note that is not a member of any beat-unit then create a beat-unit with that note as its only element.

11) If there exist two adjacent sub-beat-units that are of the same level and are members of the same beat-unit then combine the two sub-beat-units.
These rules are sufficient to insure that, when the constraint guidance process is finished, the plan places all of the staff lines and all of the measures correctly and that all notes smaller than a quarter note are grouped properly into beat-units.

Rendering rules are mainly used in the rendering stage of ENGRAVE. They are used to compute properties such as stem direction and note locations that can be determined once a plan has been computed. Some of these rules are used during constraint guidance to compute properties (such as measure size) which are needed for planning and which may change as the plan changes. Rendering rules are invoked to compute a property, or frame slot, when the value is needed but is not available.

5.4 An Example

To illustrate how the constraint guidance procedure works, we will trace through a small example. The piece of music is shown in Figure 8 as ENGRAVE produces it.

First, the initialization section produces a
skeleton plan: ten measures to a line. Diagrammatically, the skeleton plan is shown in Figure 33. Because the initialization section blindly assigns ten measures to each line, some adjustment may be necessary. This is where constraint guidance comes in. Notice that the sum of the minimum sizes of the measures in the first line is 210.75 spaces. (A space is the width of a quarter note.) Since lines are 160 spaces long on the paper being used, the first line is much too full. If we tried to print the line with so many measures it would be completely unreadable because there is not enough room on the line for all of the notes in the measures. This is accounted for by one of the constraints that ENGRAVE uses to plan the layout of this piece of music. When constraint guidance begins, the first rule that fires is rule 5 which contains a constraint concerning the sum of the minimum sizes of the measures in a line. When the rule fires, it moves measure 10 down to the beginning of the next line. The new plan is shown in Figure 34.

Before we go on, we need to discuss briefly the way that a measure’s minimum size is computed. Any two adjacent figures in a piece of music must be at least one space apart for the music to be legible.
Figure 33
Skeleton plan for Figure 8.
<table>
<thead>
<tr>
<th>line:</th>
<th>measure</th>
<th>minimum size:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 2 3 4 5 6 7 8 9</td>
<td>16.2 25.2 34.65 45.3 31.8 10.2 26.4 10.2 4.2</td>
</tr>
<tr>
<td></td>
<td>10 11 12 13 14</td>
<td>6.6 6.6 16.2 26.4 30.6</td>
</tr>
</tbody>
</table>

Figure 34
Measure 10 has been moved to the second line.
Additionally, the distance between notes varies with the duration of the notes. Each note has an amount of space following it that is a function of its duration. Consequently, the minimum size is determined from the duration of the smallest note in the measure. Consider, for example, the measure shown in Figure 35. The smallest note in this measure is the first (a 32nd note) so the distance between it and the next note can be no less than one space. The third note is twice as long as the first two but it does not, as we might suppose, get twice the space. Instead, it gets one and one half times the space of the shorter note. As it turns out, music engravers do not use a strict proportionality but tend to "skimp" on their space in order to fit more notes into a line. Using this "one-and-one-half" method, we can compute the minimum amount of space needed for a measure, adding one space for the distance between the bar line and the first note.

Going back to the constraint guidance process and Figure 34, we find the sum of the minimum sizes of the measure in the first line still exceeds the amount of space available for the line so constraint guidance rule number 5 fires again moving measure 9 to the next line.
Figure 35
Notes with longer durations take up more space.
Rule 5 will continue to fire until measure 5 is finally moved to the second line. The plan now looks like Figure 36. Now the second line is in violation of the line size constraint. Since it is currently the last line in the piece, rule 4 will fire creating a new line with no measures in it. Because no measures were moved by rule 4, the second line is still too full and rule 5 fires moving measure 14 down to line 3. Now, the line constraint is satisfied for all lines and the plan looks like Figure 37.

At this point, we know that all of the measures will fit on their lines. However, the measures must also fit exactly. That is, the bar line of the last measure must line up exactly with the end of the staff. The constraint guidance process continues.

Since it is desirable to have all of the measures in a line be nearly the same size, they are initially assigned actual sizes such that they are all the same size and the sum of their sizes is the length of the line. In the case of line 2, each of the nine measure is given 17.78 spaces. When we look at Table 3, however, we see that some of the measures are smaller than their
Figure 36

All of the measures in the first line now fit on the line.
Figure 37
The final configuration that fully satisfies the line constraint.
Initially, all the measures on a line are of equal size.

<table>
<thead>
<tr>
<th>measure</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>minimum size</td>
<td>31.8</td>
<td>10.2</td>
<td>26.4</td>
<td>10.2</td>
<td>4.2</td>
<td>6.6</td>
<td>6.6</td>
<td>16.2</td>
<td>26.4</td>
</tr>
<tr>
<td>size</td>
<td>17.78</td>
<td>17.78</td>
<td>17.78</td>
<td>17.78</td>
<td>17.78</td>
<td>17.78</td>
<td>17.78</td>
<td>17.78</td>
<td>T = 160</td>
</tr>
</tbody>
</table>
minimum sizes. This causes rule 8 to fire and adjust the
size of measure 5. In Table 4, we now have measure 5
correctly sized but the sum of the measure sizes is
larger than the line size. This causes rule 7 to fire and
all of the measure sizes are reduced.

To save some computation, only those measures that
are larger than their minimum size are considered
reducible. There are six reducible measures in this
example, so the amount of reduction is \((174-160)/6 = 2.33\).
If any reduction would result in a measure being smaller
than its minimum size, the measure is left at that size.
The result of this reduction is shown in Table 5.

Because of measure 62, the sum of the measure sizes
is still too large. Rule 7 fires again. This time there
are five reducible measures and the amount of reduction
is \((178 - 160)/5 = 3.6\). When this reduction is applied to
the measure, we get the result shown in Table 6 and all
of the measures fit properly on the line. This procedure
is applied to every line in the piece until all measures
are correctly sized. The only exception is when there are
only a few measures on the last line (as in this
example). In this case, no actions are taken and the
<table>
<thead>
<tr>
<th>measure</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>minimum size</td>
<td>31.8</td>
<td>10.2</td>
<td>26.4</td>
<td>10.2</td>
<td>4.2</td>
<td>6.6</td>
<td>6.6</td>
<td>16.2</td>
<td>26.4</td>
</tr>
<tr>
<td>size</td>
<td>31.8</td>
<td>17.78</td>
<td>17.78</td>
<td>17.78</td>
<td>17.78</td>
<td>17.78</td>
<td>17.78</td>
<td>17.78</td>
<td>17.78</td>
</tr>
</tbody>
</table>

T = 174
Table 5

All measures are reduced by the same amount but not below their minimum size.

<table>
<thead>
<tr>
<th>measure</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>minimum size</td>
<td>31.8</td>
<td>10.2</td>
<td>26.4</td>
<td>10.2</td>
<td>4.2</td>
<td>6.6</td>
<td>6.6</td>
<td>16.2</td>
<td>26.4</td>
</tr>
<tr>
<td>size</td>
<td>31.8</td>
<td>15.45</td>
<td>26.4</td>
<td>15.45</td>
<td>15.45</td>
<td>15.45</td>
<td>15.45</td>
<td>16.2</td>
<td>26.4</td>
</tr>
</tbody>
</table>

T = 178.05
Table 6
All measures are reduced again.

<table>
<thead>
<tr>
<th>measure</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>minimum size</td>
<td>31.8</td>
<td>10.2</td>
<td>26.4</td>
<td>10.2</td>
<td>4.2</td>
<td>6.6</td>
<td>6.6</td>
<td>16.2</td>
<td>26.4</td>
</tr>
<tr>
<td>size</td>
<td>31.8</td>
<td>11.85</td>
<td>26.4</td>
<td>11.85</td>
<td>11.85</td>
<td>11.85</td>
<td>11.85</td>
<td>16.2</td>
<td>26.4</td>
</tr>
</tbody>
</table>

T = 160
measure size set to the minimum size.

5.5 Beat Units

Rules 9, 10, and 11 deal with groupings of notes called "beat-units." A beat-unit is a group of notes connected by a beam. See, for example, Figure 38. Each note is an eighth note or a shorter note. The "sub-beat-unit," a construct that was devised for ENGRAVE, indicates grouping within a beat-unit. The beams on a beat-unit indicate the sub-beat-units. In every beat-unit there is a "major beam" connecting all of the notes in the beat-unit. If any of the notes are shorter than an eighth note, there are also "minor beams" to indicate the duration of each note. This will become clearer as we go through the example.

When ENGRAVE creates its skeleton plan, no beat-units are indicated in the plan. All notes are attached directly to their measures. The part of the plan for Figure 38 is shown diagrammatically in Figure 39.

Each of the notes is a flagged note indicating that its duration is that of an eighth note or less. Rule 10
Figure 38
A beat-unit.
Figure 39
The same beat-unit as it is represented in the skeleton plan.
will create beat-units or sub-beat-units to change the plan to look like Figure 40. One level is added to the plan for each flag that a note would require if it were notated by itself. Rule 9 will group the beat-units together to form a single beat-unit as shown in Figure 41. Rule 11 combines adjacent sub-beat-units of the same level. When we combine at sub-beat-unit level 1, we get Figure 42. All of the notes at level 1 are adjacent, so they get combined into a single sub-beat-unit. Combining at level 2, we get Figure 43. The plan now contains the beaming structure of our example. The beat-unit itself will get one beam. Underneath the major beam will be another beam, corresponding to the sub-beat-unit at level 1, which also spans the entire beat-unit. Below the second beam there are two beams connecting notes 1 and 2 and connecting notes 4 and 5. These beams correspond to the sub-beat-units at level 2.

5.6 Rendering

Once all of the constraints have been satisfied, the constraint guidance process is finished and the rendering phase begins. During this phase, all of the properties are computed that are needed to print the layout as
Figure 40
Beat-units and sub-beat-units created.
measure

beat units

sub-beat-unit lul 1

sub lul 2

notes

Figure 41
Beat-units combined.
Figure 42
Combining level-1 sub-beat-units.
planned during constraint guidance.

Since the nodes in a plan represent objects, it is natural for them to be represented as frames. Rendering rules are "if-needed" rules that are attached to slots and that fire when a slot is accessed which has no value. In this way, all slot values are computed on demand. As a result, the program needs only to display each object in the plan. As each object is accessed all of the values needed to display the object are computed if they have not already been computed.

When a key signature or a time signature must be displayed, there are rendering rules that determine the proper spacing between them. These rules are used whenever a change of key, time, or clef occurs in the music and at the beginning of the piece. Figure 44 shows a diagram of proper spacing between each of these kinds of figures.

Slot values that are computed by rendering rules are properties that are not essential to the planning process but that are needed to display the figure. Some of these properties are stem length, pitch, and key signature. A
Figure 44
Conventions for spacing.
list of the rendering rules can be found in Appendix A.

ENGRAVE is capable of producing most conventional music notation. Figures 45 through 47 show some of its capabilities. An evaluation of other systems that produce musical notation can be found in Chapter 2. We feel that ENGRAVE compares favorably with these systems and that it has an added advantage because of the flexible nature of its representation. ENGRAVE does not currently output multi-line parts or music for voice as these capabilities require new rulesets. Appendix E contains the details of the ENGRAVE's graphics including the methods used to compute ties, slurs and beam slants.

5.7 Constraint Guidance in a Another Domain

Constraint guidance can be applied to any domain that involves the placement of objects in space. To illustrate its use, we will present a second example of constraint guidance as applied to the formatting of graphics screens for human-computer interaction. We will use the aircraft carrier controller's screen discussed in [Reilly and Roach, 1984]. Instead of two screens, we will assume that we have one large screen on which we place
Figure 46
An example showing ties and slurs.
Figure 47

An example showing beat-units and beaming.
all of the objects. These objects are: a plan view of the aircraft carrier showing the flight path and the locations of the planes, a vertical perspective for displaying the aircraft altitudes (the stack), a horizontal perspective of the final landing phase (the glide path), the marshall status board containing textual information about the aircraft, and a message area for system messages and user input. We want to place each of these objects on the screen while following good visual design techniques so that no information is obscured.

As in music, we start with a skeleton plan. In this simplified version of the domain, two plans will seldom be very different. The major blocks will be ordered in the plan according to their relative importance as specified by the user. A three-level hierarchy exists in this example because each block on the screen is composed of several parts. Figure 48 shows a typical plan. Initially, the nodes in the plan will not have location values. The constraint guidance system will compute and manipulate these location and size.

Constraints for this example involve the concepts of
A plan for an aircraft carrier controller's screen.

Figure 48
visual design and the requirement that information must not be obscured. We will use the following set of rules:

1) If the highest priority block has not been placed, then place it in the center of the screen.

2) If the stack has not been placed, then place it to the left or right of the carrier.

3) If the second highest priority block has not been placed, then place it so the flight path points to it.

4) If the glide path has not been placed, then place it so that it points towards the aircraft carrier.

5) If the marshall status board has not been placed, then place it in the largest free block of space.

6) If the message area has not been placed, then place it in the largest free block of space.

7) If two opposite sides of a block intersect an object or are off the screen, then scale the block down so as to clear at least one side.

8) If a side of a block is off the screen, then move the block away from that edge of the screen.

9) If a side of a block obscures another object, then move the block away from that object.

The blocks in this example are shown separately in Figure 49. There are five planes in the flight path. Three of these have their characteristics displayed on the marshall status board and one is in the final landing phase. Assume that the blocks are ranked in the following
Figure 49
The blocks found in the controller's screen.
order of importance: plan view, marshall status board, stack, glide path, and message area. Rule 1 will place the plan view in the center of the screen. Rule 2 places the stack directly to the left of the carrier. The second highest priority block, the marshall status board, is placed by rule 3 below and to the right of the plan view. Rule 4 places the glide path above and to the right of the plan view. Finally, the message area is placed in the free area to the right of the screen by rule 6. The result of these placements is shown in Figure 50. All of the block have now been placed so rules 1 through 6 will no longer trigger. Rules 7, 8 and 9 deal with the blocks that overlap or go off the screen. Rule 8 will move the marshall status board upward so that it is entirely on the screen. A plane is obscured however, by this move. Before this is dealt with, rule 8 will trigger once more to move the glide path onto the screen. The result is shown in Figure 51. Now rule 9 triggers to move the marshall status board to the right and uncover the fifth plane. The glide path still overlaps the plan view but no information is obscured so the screen is left as it is. Figure 52 shows the final screen.

We have given a simple example to show how
Figure 50
Initial placement of objects.
Figure 51
Resolution of objects off the screen.
Figure 52
Final screen
Constraint guidance can be applied to another domain. In reality, the rules for visual design are not nearly as well-defined as in music and they are not as rigid [Roach et al., 1982; Roach et al., 1986]. A realistic system for visual design would have to capture this flexibility and the subjectivity of the domain. Spatial planning is central to the realization of such a system.

5.8 Literature Review

Planning is an important part of problem solving. Work on planning systems refers back to some of the earliest work in Artificial Intelligence. GPS [Ernst and Newell, 1969] and STRIPS [Fikes and Nilsson, 1971] are especially notable. Both of these systems were essentially linear planners and could only plan at one level of detail. Multiple levels of abstraction were provided by hierarchical planners, the first of which was LAWALY [Siklossy, 1972; Siklossy & Dreussi, 1983]. Other hierarchical planners are ABSTRIPS [Sacerdoti, 1974], NOAH [Sacerdoti, 1977], and MOLGEN [Stefik, 1981]. It is worth noting that MOLGEN used a technique called "constraint posting" to handle some of problems caused by interacting subgoals.
Constraint satisfaction is also an important topic in Artificial Intelligence literature. In some domains, it is desirable to define constraints that limit the search space of a system. This is especially true of resource allocation problems [Wilkins, 1984]. ISIS [Fox, 1983], a job shop scheduling system, makes use of constraints to build schedules for machines and products. As noted above, MOLGEN used constraints to reduce sub-goal interaction in planning tasks. DEVISER [Vere, 1983] is a time-based planner that uses constraint propagation to adjust start and end times for tasks in a plan. While TEX [Knuth, 1979] is not an AI system, Knuth's "glue" is a kind of constraint.

Eastman's [1973] spatial planning system planned the layout of furniture in a room. The used a digitized representation of space and a set of procedural operators. Grason [1970] used a graph theoretic approach for arranging rooms in a building. In both cases, the objects that are being arranged had simple shapes and could be arranged using a straightforward state space search. We feel that we have been able to build on this foundation to allow for spatial planning using complex
The emphasis in this work has been on planning in spatial domains rather than on musical output. There are some musical output systems, however, that merit attention. Perhaps the best known of these systems is SMUT [Byrd, 1974]. This system can handle most conventional music notation and can print multiple voices. Other systems are mentioned in Chapter 2. All of these systems use algorithmic methods to format their output. We feel that much flexibility and power can be gained by using heuristic methods from Artificial Intelligence instead of algorithmic methods.

5.9 Conclusions

We have described a system that produces quality musical output. The main thrust of this chapter however, was not to describe how to produce correct musical output but to describe constraint guidance by showing how it operates in a musical domain. We have endeavored to do this by discussing the principles behind constraint guidance and by stepping through some examples.
There are some limitations to constraint guidance: its dependence on an initial skeleton plan and the lack of ability to relax constraints. Constraint guidance as we have implemented it relies on an initial skeleton plan to begin its operation. This makes ENGRAVE a generate-and-test system in a sense. One of the lessons of DENDRAL [Lindsay et al., 1980, p. 148] is that a good generate-and-test system must have a good generator. It is important therefore, that a skeleton plan be a good "first cut." This will reduce the amount of extra work that the system must do. We could have had ENGRAVE initially assign all of its measure to one line. Had we done this, ENGRAVE would have spent a lot of time creating new lines and moving individual measures to the next line. This could be a very serious performance problem for a large piece of music. A system that uses constraint guidance ought to have a good set of heuristics for generating a skeleton plan.

It is possible that, in some domains, over-constrained situations could occur. In our implementation this would cause ENGRAVE to run forever. In an over-constrained problem, there will always be a violated constraint. It is desirable to be able to detect
over-constrained situations and intelligently relax constraints to arrive at a best solution [Fox, 1983]. Barring this, the rule set must be designed such that there is always a solution that satisfies all of the constraints.

We believe that constraint guidance is applicable to three dimensional domains. The chief requirement is that objects can be decomposed into sub-parts. This allows for the multiple levels of detail in a plan. It is not required that the decomposition conform rigidly to a tree structure but there must be a distinction between levels of detail. A good example is a factory floor layout. Many objects on a factory floor have parts that can be configured in different ways. For example, a robot station may have a controller that must be located near the robot and a parts bin that must be within reach of the robot arm. Additionally, machines may be combined into larger entities such as an assembly line using constraints about their location on the line. We believe that this kind of spatial planning problem is a suitable application for constraint guidance.

In summary, ENGRAVE is a spatial planning system
that uses a technique called constraint guidance to plan the spatial layout of music. This technique is useful not only for musical layout, but for other spatial domains as well. We believe that constraint guidance extends existing spatial planning techniques to provide more utility and power.
Chapter 6
Discussion

6.1 Vision

The first section of ENGRAVE is a vision system that can process handwritten musical notation. In the first attempt to build the system, we tried using general-purpose image processing techniques to detect features. These features were to be the input to an expert system that used musical knowledge to understand these features. The expert system worked quite well but it depended on an unreliable low-level system for its input. Much time was spent changing parameter values for the low-level operators to get better results. However, a set of parameters can only be tuned to work for a small set of images and consistent results cannot be guaranteed in general. The solution to this problem was to use domain knowledge at the low level. When this was tried, there was a remarkable improvement in the results. This suggests an important conclusion: domain knowledge must be employed at all levels for a vision system to be successful.
There are several issues that merit further consideration. The system as it stands is still separated into a low-level section and an expert system. I feel that these two levels need to be integrated into a single system. This would enable full control over knowledge at all levels. In this way, failure of a high-level rule could result in backtracking into low-level rules causing a reconsideration of pixel relationships. When we build such an integrated system, we will find that several levels of knowledge exist. Knowledge about image-processing, low-level relationships, musical notation and control exist simultaneously in the same system. This has been discussed in [Clancey, 1982] and a framework is proposed to represent different levels of knowledge and their relationships. The use of frames [Minsky, 1975] is another method that is used to explicitly handle different levels of knowledge.

Another issue is the flow of information between levels. While ENGRAVE was being developed, it was found that information from low-level processes was useful at the higher levels even though no conclusions could be made earlier using the same information. Low-level
information, however, after has a different meaning at higher levels. The problem that needs to be addressed is how information is passed between levels and what forms it should have to be most effectively used at any level.

6.2 Graphics

ENGRAVE uses spatial planning to produce its output. This methodology is appropriate for domains that involve placement of objects in space. This work and previous work have only dealt with two dimensional domains. An interesting and useful effort would be to extend these ideas to three dimensions.

One of the worst shortcomings of the spatial planner is speed. The system is slow partly because it is implemented in an interpreted language and because it must repeatedly search a large database. (There are approximately 250 nodes in the plan for the full piece shown in Chapter 4.) One of the lessons of DENDRAL is that the efficiency of the generator is important in generate-and-test systems [Lindsay et al, 1980]. In our case, this points to the corrective actions of the production rules. What is needed for improvement is
corrective actions that correct a situation more completely. Ideally, when an unsatisfied constraint is detected, the corresponding action should fully satisfy the constraint. Whether this is possible is an open question, but there certainly is room for improvement.

6.3 Conclusion

The development of ENGRAVE provided a good vehicle to study some interesting questions in artificial intelligence, particularly expert systems. The use of world knowledge for low-level vision computing and the use of spatial planning for graphical output are the two most important concepts that came out of this study.

While ENGRAVE, as it stands now, is not ready for commercial use, it would require only a few additions to reach that point. Development of a music editor is the largest effort that would be required. More effort is also needed to make the system execute faster. However, with the state of affairs in the music publishing industry, a batch job that ran overnight still would be an economic improvement over current practices.
REFERENCES


D. Chester, "Implementing Production Systems in Prolog," Internal Report, Department of Computer and Information Sciences, University of Delaware, 1981.


M. Stefik, "Planning with Constraints (MOLGEN: Part 1),"


Appendix A

Visual Expert Rules

This appendix and Appendix B contain listings of the rules used by ENGRAVE. The most important product of an expert system is the knowledge that has been codified. Therefore, these rules are listed in English so that someone else may make use of the knowledge.

A.1 Rules about note durations.

Note durations are expressed as the reciprocal of the mathematical value of the note’s duration. Hence the duration of a quarter note would be expressed as 4. The duration of a dotted quarter note (one and one-half beats) would be 2.6667 (or 8/3).

If a note has a closed notehead, then its duration is computed using the formula $2(F+D+2)/3D$. Where $F$ is the number of flags and $D$ is the number of dots associated with the note.

If a note has an open notehead and there are two curves that intersect the notehead and one curve is to
the left of the notehead and the other curve is to the right of the notehead and one of the curves rises higher than the other then the note is a natural and its duration is NIL.

If a note has an open notehead, its duration is computed by the formula \( \frac{2^D T}{3^D} \) where \( T \) is the type of note (2 for a half note, 1 for a whole note) and \( D \) is the number of dots that modify the note.

If a note has no notehead and there are two curves such that both intersect the middle line of the staff and one runs from the middle line higher and one runs from the middle line lower and both nearly vertical then the note is a quarter rest and has pitch R (for rests).

All other notes have duration NIL

A.2 Rules about measures

To compute the notes of a measure, recursively create a note frame until the create operation fails. Then add the list of notes to the NOTES slot of the measure.
If a measure has a previous measure, then the first bar-line is the second bar-line of the previous measure.

If a measure has no previous measure, then its first barline is the leftmost curve that is nearly vertical and intersects all of the staff lines.

To compute the second bar-line of a measure, find a curve that is nearly vertical and intersects all of the staff lines and has no similar curve between it and the first barline.

If a measure is in the top line on the page, then the top boundary of the area covered by the measure is the top of the image.

If a measure has another staff line above then its top boundary is half the distance between the top line of its staff and the bottom line of the staff above it.

If a measure is in the bottom line of the page, the measure's bottom boundary is the bottom of the image.
If a measure has another staff line below it then its bottom boundary is half the distance between bottom line of its staff and the top line of the staff below it.

The left boundary of a measure is the column coordinate of the measure's first bar line.

The right boundary of a measure is the column coordinate of the measure's second bar line.

A.3 Rules about notes

A.3.1 TO-FILL rules for notes

If there is an unused hole in the image, then treat it as a notehead and retrieve the note's duration and pitch.

If there is an unused blob in the image, then treat it as a notehead and retrieve the note's duration and pitch.

If there is an unused curve in the image, then assign it to a note and retrieve the note's duration and
A.3.2 IF-NEEDED rules for notes.

If a note has an open or a closed notehead and there is a set of connected nearly vertical curves one of which intersects the notehead, then conclude that these curves are the note’s stem.

If a note has a closed notehead and the note has a stem and there is a curve that intersects the stem on the opposite end from the notehead, then add the curve to the list of flags for the note.

If a note has a closed notehead and the note has a stem and there is a curve that intersects a staff at one end and the note’s stem intersects that same staff line and there are no curves between the stem and the curve, then add the curve to the note’s list of flags.

If a note has a closed notehead and the note has a downward pointing stem and there is a curve that is below and to the right of the note’s stem and there are no curves between the stem and the curve being considered,
then add the curve to the note's list of flags. (This allows for stems that are drawn too short. One end of the flag should be almost directly below the stem.)

If a note has a closed notehead and the note has an upward pointing stem and there is a curve that is above and to the left of the note's stem and there are no curves between the stem and the curve being considered, then add the curve to the note's list of flags.

A note's staff is the staff that contains the measure that contains the note. (This one is a little obvious but it is a necessary piece of knowledge.)

If a note has a stem and the end of the stem opposite its intersection with the notehead, then the note has a downward pointing stem.

If a note has a stem and the end of the stem opposite its intersection with the notehead, then the note has a downward pointing stem.

If a note has a stem which does not intersect the notehead and the stem is above the notehead, then the
note has an upward pointing stem.

If a note has stem which does not intersect the notehead and the stem is below the notehead, then the note has a downward pointing stem.

Note: In this implementation, a note may have only one dot or none.

If a note has a downward pointing stem and there is a dot in the measure which is to the right and above the notehead, then add this dot to the list of dots for this note.

If a note has an upward pointing stem and there is a dot in the measure which is to the right and below the notehead, then add this dot to the list of dots for this note.

If there is a dot in the same measure as a note, and the dot is directly to the right of the notehead and there are no other noteheads between the dot and the notehead, then add this dot to the list of dots for this note.
A.4 Rules about staves.

To fill a stave frame find its lines and find its measures.

The lines of a staff consist of five "staff-lines" that are equal distances apart and that have no other staff lines between them.

The left end of a staff is the average of the left end coordinates of all the staff lines.

The right end of a staff is the average of the right end coordinates of all the staff lines.

To find the measures of a staff, create a measure frame repeatedly until the TO-FILL rule fails.
Appendix B

Spatial Planner Rules

B.1 Constraint Guidance Rules

1) If there exist two measures such that:
   - they are adjacent; and
   - they both contain only a whole rest;
   then combine them into a single multiple-measure rest.

2) If there exists a line such that:
   - it is the last line; and
   - the sum of the minimum sizes of the measures is greater than the line size;
   then create a new line after this one.

3) If there exists a line such that:
   - it is not the last line; and
   - the sum of the minimum sizes of the measures is greater than the line size;
   then move the last measure of the line down to the next line.

4) If there exists a line such that:
   - it is not the last line; and
   - the sum of the measures is less than the line size;
   then expand all the measures in the line by \((L-S)/n\) where \(L\) is the line size, \(S\) is the sum of the measures and \(n\) is the number of measures in the line.

5) If there exists a line such that:
   - the sum of the measure sizes is greater than the line size; and
   - the sum of the minimum sizes is less than or equal to the line size;
   then find the least complex measure in the line with \((\text{size} > \text{min})\) and reduce it by \(\text{min}(\text{sum_sizes} - \text{linesize}, \text{size} - \text{min})\).

6) If there exists a measure such that its size is less than its minimum size
   then expand the measure by \((\text{min} - \text{size})\).
7) If there exists a beat-unit such that it is not full and there exists a note such that:
   - it is a flagged note; and
   - it is adjacent to one of the notes in the beat-unit; and
   - it is not in another measure or beat-unit;
then add that note to the beat-unit.

9) If there exists a note such that:
   - it is flagged; and
   - it is not already in a beat-unit;
then create a beat-unit with that note as its only element.

10) If there exist two sub-beat-units such that:
   - they are the same level; and
   - they are adjacent;
then they can be combined.

B.2 Rendering Rules

If the pitch of a note is not defined (in the NOTES language), the pitch is the same as the note before it.
(Implied tie)

If the NOTES duration value for a note is 1 and the note is not a rest, then the note is a whole note.

If the NOTES duration value for a note starts with a 2 and the note is not a rest, then the note has an open notehead.

If the NOTES duration value for a note starts with 4 or greater and the note is not a rest, then the note has a closed notehead (as in quarter notes).

If a note is a rest and has a duration value of 1, then the note is a whole rest.

If a note is a rest and has a duration value that starts with 2, then the note is a half rest.

If a note is a rest and has a duration value of 4, then the note is a quarter rest.
If a note is a rest and has a duration value of 8 or greater, then the note is an eighth note rest (or smaller).

The vertical coordinate for rests is the same as the center line of the staff.

Notes in the treble clef are offset vertically from the center of the staff by

\[(3.5 \times O \times S) - 17\text{ spaces}\]

where \(O\) is the octave number for the note and \(S\) is the offset from C for the letter name of the note.

Notes in the bass clef are offset vertically from the center of the staff by

\[(3.5 \times O \times S) - 11\text{ spaces}\]

where \(O\) is the octave number for the note and \(S\) is the offset from C for the letter name of the note.

The sequence number (distance in spaces from C) for A is 2.5.

The sequence number for B is 3.

The sequence number for C is 0.

The sequence number for D is 0.5.

The sequence number for E is 1.

The sequence number for F is 1.5.

The sequence number for G is 2.

Full measure rests (whole rests) are placed in the center of the measure.

The first note in a measure always starts one unitsize from the start of the measure.

If a measure contains a front repeat (left-handed), then the first note in a measure is placed 3 spaces away from the bar line.
If a measure is the first in its line and the line contains a time signature at its beginning, then the first note in the measure is placed 2-1/4 spaces after the time signature.

If a measure is the first in a line, and the line contains a key signature at its beginning, then the first note in the measure is placed 2-1/2 spaces past the last accidental in the key signature.

If a measure is the first in its line, and the line contains no key or time signatures at its beginning, the first note in the measure is placed 1-1/2 spaces from the beginning of the line.

If a measure has a time signature change, then the first note in the measure is placed 2-1/4 spaces from the time signature.

If a measure has a key signature change, then the first note in the measure is placed 2-1/2 spaces from the last accidental in the key signature.

If a measure has no time or signature changes, then the first note in the measure is placed 1-1/2 spaces from the first bar line of the measure.

If a note is not the first note in its measure, its horizontal coordinate depends on measure size and the location of the last note as follows:

\[ \frac{P \times U \times (S - E) + L}{(M - E)} \]

where:

- \( P \) is the space allocated to the last note; and 
- \( U \) is the "unitsize" of the measure; and 
- \( S \) is the size of the measure; and 
- \( E \) is the amount of space needed in the measure for signature changes and repeats; and 
- \( M \) is the minimum size of the measure.

If a note has an accidental sign, the note should be moved 1 space to the right and the sign should be placed 1 space before the note.
The "space" property of a note (the amount of space the note gets within a measure) is computed by:

\[
S \cdot \frac{T}{V}
\]

where:

- \(S\) is the spacing value of the note from the measure's profile; and
- \(T\) is the type of the note (4 for quarter notes); and
- \(V\) is the reciprocal of the note's duration in beats.

Note that this space is computed as if the measure were at minimum size.

The stem direction for notes in a beat unit is the same as the stem direction of the note in the beat unit that is farthest from the center line of the staff.

If a note is part of a beat-unit, then its stem direction is the same as that of the beat-unit.

If a note's horizontal location is below the center line of the staff, then its stem direction is "up".

If a note's horizontal coordinate is above or on the center line of the staff, then its stem direction is "down".

If a note is part of a beat-unit, then its stem is long enough to intersect with the major beam of the beat-unit.

If a note is more than one ledger line outside of the staff, then the stem is long enough to intersect the middle line of the staff.

If a note has more than two flags, then the stem length is computed (in spaces) by:

\[
F + 1.5 \text{ where } F \text{ is the number of flags.}
\]

In general, stems are one octave (3-1/2 spaces) long.

If a note is not a whole note, and not a rest, then the note has a stem.
If a note is dotted and is on a line, then the first dot is 1/2 space above and 1-1/2 spaces to the right of the notehead.

If a note is dotted and is on a space, then the first dot has the same vertical coordinate and is 1-1/2 spaces to the right of the notehead.

Dots for the same note are placed one space apart horizontally and all have the same vertical coordinate.

If a note is above or below the staff, then the number of ledger lines is computed by:

\[ D - 2 \]

where \( D \) is the absolute value of the offset of the note from the center line of the staff in spaces.

The slope of a beat-unit is the linear regression slope of the line formed by the noteheads in the beat-unit.

For the primary beam, the note closest to the middle line of the staff determines the location of the beam.

For sub-beat-units, the point on the beam is one line lower/higher than the primary beam for each level past 2.

If a note is the only element of a sub-beat-unit and is the last note of its beat-unit, then it has a fractional beam that points to the left.

If a note is the only element of a sub-beat-unit and is the first note of its beat-unit, then it has a fractional beam that points to the right.

If a note is the only element of a sub-beat-unit and is in an odd position in its beat-unit, then it has a fractional beam that points to the left.

If a note is the only element of a sub-beat-unit and is in an even position in its beat-unit, then it has a fractional beam that points to the right.

If a beat-unit or sub-beat-unit does not have a fractional beam, then the X-coordinates of the endpoints of its beam are the same as the X-coordinates of the stem ends of the first and last note in the unit.
If a beat-unit has more than one note, then it has a beam.

If a note has flags, then the first flag is located at the end of the stem and subsequent flags are each located one space closer to the notehead.

If a note has an accidental, it is located 1-1/4 spaces to the left of the notehead and has the same Y-coordinate.

The key signature of a line is the same as the key signature of its first measure.

If a line is the first in the piece, then print its key signature when the line is printed.

If a line has a different key than the previous line, then we print its key signature.

No key signature is printed for lines other than the first unless the key has changed since the last line.

No key signature is printed if the line is in the key of C.

The clef of a line is the same as the clef of its first measure.

The time signature of a line is the same as the time signature of its first measure.

If a line is the first on its page, then its X position is

\[
\text{120} \quad \frac{-}{n+1}
\]

in spaces where \(n\) is the number of lines on the page.

A line's column coordinate is computed by

\[
\text{120} \quad \frac{L}{n+1}
\]

in spaces where \(L\) is the location of the previous location and \(n\) is the number of lines on the page.

If a measure is the last in its line, the line is not the
last in the piece, and the next measure does not have a key change, then the measure’s bar line is placed at the end of the line. (Note: the bar line that is associated with a measure is the one at the end of the measure.)

If a measure contains a rest for its entire duration, and it is not the first measure of the line, then the location of the bar line is computed from the location of the previous bar line added to the size of the measure.

A bar line’s horizontal coordinate depends on measure size and the location of the last note in the measure as follows:

\[
\frac{P \times U \times (S - E) + L}{(M - E)}
\]

where:

- \( P \) is the space allocated to the last note; and
- \( U \) is the "unitsize" of the measure; and
- \( S \) is the size of the measure; and
- \( E \) is the amount of space needed in the measure for signature changes and repeats; and
- \( M \) is the minimum size of the measure.

The vertical coordinate of a bar line is the same as the center line of the staff.

If a measure is the first in its line and has a starting repeat, then its starting point is the same as the horizontal coordinate of the repeat sign.

If a measure is the first in its line, then its starting point is 3/4 spaces past the last sign in the lines time signature.

Otherwise, a measure’s starting point is the same as position of the second bar line of the previous measure.

In a given measure, if there exists a note such that:
- it is not the first note in the measure; and
- it has an accidental; and
- its predecessor is of less than or equal duration; then the measure’s unitsize is three times the size of a horizontal space.

otherwise it is twice the size of a horizontal space.
If a measure is in the last line on a page and the sum of
the minimum sizes of the measures on the line takes up less
than 1/3 of the space on the line, then set the size of the
measure equal to the minimum size of the largest measure on
the line.

The size of a measure is computed by the space available
for a line divided by the number of measures in the line.

If a measure is the first or last measure in the piece
(These measures are often incomplete), then the size of the
measure is computed by

\[
\frac{D \cdot S \cdot L}{Num \cdot N}
\]

where

- \(D\) is the denominator of the time signature; and
- \(Num\) is the numerator of the time signature; and
- \(S\) is the sum of the durations of the notes in the
  measure (this is equal to the time signature for
  complete measures); and
- \(L\) is the amount of space in the line; and
- \(N\) is the number of measures in the line.

If a measure is the first in the piece, then its time
signature is the same as the time signature of the piece.

If a measure has a time signature change, then its time
signature is the same as the new time signature.

If a measure is not the first in the piece and does not
have a time signature change in it, then its time signature
is the same as that of the previous measure.

If a measure is the first in the piece, then its key
signature is the same as the key signature of the piece.

If a measure has a key signature change, then its key
signature is the same as the new key signature.

If a measure is not the first in the piece and does not
have a key signature change in it, then its key signature
is the same as that of the previous measure.

If a measure is the first in the piece, then its clef is
the same as the clef of the piece.

If a measure has a clef change, then its clef is the same as the new clef.

If a measure is not the first in the piece and does not have a clef change in it, then its clef is the same as that of the previous measure.

If a clef is the clef at the beginning of a line, then it is located 2-1/2 spaces past the beginning of the line. Its vertical coordinate is the same as the center line of the staff.

If a clef is part of a measure (because of a clef change), then it is located 2-1/2 spaces to the left of the previous bar line.

If a key signature is part of a measure (a key change) then the horizontal position of its first sign is 1-1/2 spaces past the bar line.

If a key signature is part of a line, then the horizontal location of the first sign is 2 spaces past the clef sign.

If a time signature is the in the same measure as a key signature, then the time signature is positioned 2-1/4 spaces past the last sign of the key signature.

If a time signature is part of a line, then its horizontal is 3-1/4 spaces past the clef sign.

If a time signature is part of a measure and there is no key signature in the measure, then its horizontal position is 1-3/4 spaces past the bar line.

Note: In the following, "front repeat" refers to the starting repeat figure in a repeated section of music and "end repeat" refers to the terminating repeat figure.

If a front repeat is in the first measure of a line, then its horizontal position is 2-3/4 spaces past the time signature.

If a front repeat is in the same measure as a time signature change, then its horizontal position is 2-3/4 spaces past the time signature.
If a front repeat is in the same measure as a key change, but the measure has no time change, then the horizontal position of the repeat is 2-3/4 spaces past the last sign in the key signature.

If a front repeat is in a measure that is the first in its line, and there are no other figures at the start of the line (clefs, etc.), then the horizontal location of the repeat is the same as the start of the staff line.

If a front repeat is in a measure that is not at the start of a line and the measure has no other key, time, or clef changes, then the horizontal position of the repeat is the same as that of the previous bar line.

If an end repeat occurs in a measure preceding a clef change, then locate the repeat 2-1/2 spaces before the clef sign.

If an end repeat occurs in a measure with no clef change following, then the horizontal location of the repeat is the same as the location of the measure’s bar line.

If a line does not contain a repeat in the first measure, then the amount of space on the line for notes is lessened from the normal length of staff line by 6-1/2 spaces plus one space for each sign in the key signature.

If a line has a new key signature, a new time signature, and a repeat in the first line, then the space on the line for notes is lessened from the normal length of a staff line by 10 spaces plus one space for each sign in the key signature.

If a line has a new key signature and a new time signature but no repeat in the first measure, then the space on the line for notes is lessened from the normal length of a staff line by 6-1/2 spaces plus one space for each sign in the key signature.

If a line has a new key signature and a repeat in the first measure then the space on the line for notes is lessened from the normal length of a staff line by 7-3/4 spaces plus one space for each sign in the key signature.

If a line has a new time signature and has a repeat in the first measure then the space on the line for notes is lessened from the normal length of a staff line by 6
spaces.

If a line has a new key signature only, then the space on the line for notes is lessened from the normal length of a staff line by 4-1/2 spaces plus one space for each sign in the key signature.

If a line has a new time signature only, then the space on the line for notes is lessened from the normal length of a staff line by 2-1/2 spaces.

If a line has a repeat in the first measure and no changes in key or time, then the space on the line for notes is lessened from the normal length of a staff line by 1-1/2 spaces.

If a line has no changes and therefore no special figures to allow space for then the space on the line for notes is equal to the normal length of a staff line.

If a line has a different key signature than the line preceding it then both the clef and the key signature must be printed at the beginning of the line.

If a line has a different clef than the line preceding it then both the clef and the key signature must be printed at the beginning of the line.

If a line has a different time signature than the line preceding it then the time signature and clef must be printed at the beginning of the line.

The minimum size of a measure is determined by the following formula:

\[ S \times U + E \]

where:

- \( S \) is the sum of the durations of the notes in the measure; and
- \( U \) is the unitsize for the measure; and
- \( E \) is the amount of extra space needed for signatures, clefs, etc.

Allow 3 spaces in a measure for a front repeat.

If a front repeat is followed by a time signature, then
there are 2-3/4 spaces between them.

If a front repeat is followed by a key signature, then there are 2-3/4 spaces between them.

If a front repeat follows a bar line, then they have the same location.

Allow 2-1/2 spaces for a time signature.

If a time signature is followed by a key signature, then there are 2-1/4 spaces between them.

If a time signature follows a bar line, then there are 1-3/4 spaces between them.

Allow 4 spaces plus one space for each sign but one for a key signature.

Allow one space if the first note in a measure has an accidental.

If the next measure has a clef change, then allow 5 spaces to print it in the current measure.

Allow 1-1/2 spaces if the measure has an end repeat.

If two notes are tied or slurred together and the notes are not on the same line, then create two "slur" nodes for the figure. One node is from the first note to the end of the line and the other is from the beginning of a line to the second note.

If two notes on the same line are slurred or tied together, then create "slur" node for the slur or tie.

If a tie is a short tie and the notes are on a line, then the left end point is 3/4 spaces to the right of the notehead of the first note and 1/4 space above or below the notehead (in the direction of the tie).

If a tie is a short tie and the notes are on a space, then the left end point is 1/2 space to the right of the first notehead and 3/4 spaces above/below it.

If a tie is a short tie and the notes are on a line, then the right end point is 3/4 spaces to the left of the second notehead and 1/4 space above/below it.
If a tie is a short tie and the notes are on a space, then the right end point is 1/2 spaces to the left of the second notehead and 3/4 spaces above/below it.

If a tie is a short tie and the notes are on a line, then the peak is located halfway between the noteheads and 1/2 space above or below them, depending on the direction of the tie.

If a tie is a short tie and the notes are on a space, then peak of the tie is located halfway between the noteheads and 1 space above or below them.

If a tie is not a short tie and has both notes on a line, then the left endpoint is located 3/4 spaces to the right of the first notehead and 3/4 spaces above or below it (in the same direction as the tie).

If a tie is not a short tie and has both notes on a space, then the left endpoint is located 1/2 spaces to the right of the first notehead and one space above or below it.

If a tie is not a short tie and has both notes on a line, then the right endpoint is located 3/4 spaces to the right of the second notehead and 3/4 spaces above or below it.

If a tie is not a short tie and has both notes on a space, then the right endpoint is located 1/2 space to the right of the second notehead and one space above or below it.

If a tie is not a short tie and has both notes on a line, then the peak of the tie is located halfway between the noteheads of the tied notes and 1-1/2 spaces above or below them.

If a tie is not a short tie and has both notes on a space, then the peak of the tie is located halfway between the noteheads of the tied notes and 2 spaces above or below them depending on the direction of the tie.

A curve is a short tie if it is a tie and the first and last notes are less than 3 spaces apart.

If a slur or a tie has any note between the notes it connects with a downward stem, then the slur or tie is concave down.
If the left note of a slur is on a line and the stem direction of the note is the same as the direction of the slur, then the left end point of the slur is 1-1/4 spaces directly above or below the notehead (in the opposite direction of the stem).

If the left note of a slur is on a space and the stem direction of the note is the same as the direction of the slur, then the left end point of the slur is one space directly above or below the notehead (in the opposite direction of the stem).

If the left note of a slur has an upward stem, and the slur is concave down, then the left end point of the slur is 2 spaces to the right of the notehead and 1/2 space above the top of the stem.

If the right note of a slur is on a line and the stem direction of the note is the same as the direction of the slur, then the right end point of the slur is 1-1/4 spaces directly above or below the notehead (in the opposite direction of the stem).

If the right note of a slur is on a space and the stem direction of the note is the same as the direction of the slur, then the right end point of the slur is one space directly above or below the notehead (in the opposite direction of the stem).

If the right note of a slur has an upward stem, and the slur is concave down, then the right end point of the slur has the same Y coordinate as the notehead and is 1/2 space above the top of the stem.

If a slur is concave down, then the peak of the slur is one space above the highest obstacle between the left and right notes of the slur.

If a slur is concave up, then the peak of the peak of the slur is one space below the lowest obstacle between the left and right notes of the slur.
Appendix C

Diagrams of Musical Characters

This appendix diagrams of the graphical characters used by ENGRAVE to produce musical notation.
Figure C-1
Whole Note
Figure C-2
Closed Notehead
Figure C-3
Open Notehead
Figure C-5
Natural
Figure C-6
Flat
Figure C-8
Repeat
Figure C-9
Repeat (beginning)
Figure C-11
Treble Clef
Figure C-12
Bass Clef
Figure C-13
Figure '3' for time signatures
Figure C-14
Figure '4' for time signatures
Figure C-15
Quarter Rest
Figure C-16
Eighth Rest
Input to the printing system is a piece of music expressed in NOTES -- a language for representing music on a computer. We describe NOTES briefly to acquaint the reader with it and to aid in the understanding of the system.

NOTES is a list-structured language suitable for use with PROLOG or LISP. It has the advantage that it is also reasonably easy for human beings to read.

A score is represented as a list. The first elements of the list are the composer, the title and initializations such as the key signature and the time signature. The elements following are lists for each movement or section. Each movement is further decomposed into parts for each instrument. It is the parts that contain the important information for printing.

To represent a note, we must first represent both pitch and duration. A note is a list of at least two character strings. The first string represents
Duration is a sequence of digits followed by zero or more "d"'s which may be followed by an optional "t" and more digits. The first digits form an integer that represents the type of note. A half-note would be represented by a "2" and a quarter-note by a "4". The optional "d"'s are used for dotted notes -- one "d" for each dot. The "t" shows that the note is part of an n-tuplet group with the integer following being n. Pitch is represented by a three- or four-character string. The first character is the letter name of the note. The second is the octave that the note is in and the third and fourth characters, if present, represent an accidental. If the pitch is simply the letter "R", then the note is a rest. Any other strings following the pitch will be modifiers that represent articulation markings. Where possible, these are literal indicators of the figure.

Measures are simply lists of notes. At the head of the list is the word "meas" followed by any modifiers to the measure such as a repeat or a key change. See Figure D-1 for an example of a few measures expressed in NOTES.

The following is the NOTES representation of the
piece shown in Figure 8.

(part tenor

(meas (4 F3)(4 EN3)(4)(4 G3))
(meas (4)(4 F#3)(8)(8 G#3)(4 A3))
(meas (2)(8)(16 G#3)(16 F#3)(4 G3))
(meas (4)(8 A3)(8 E2)(8 A3)(16 Bb3)(16 A3)
(8 GN3)(8 A3))
(meas (8 B3)(8 D4)(8 G3)(8 A3)(8 B3)(8 G3)
(8 E3)(16 FN3)(16 G3))
(meas (4 A3)(2 G3)(4 F3))
(meas (8)(16 E3)(16 D3)(4 A3)(4)(4 G3))
(meas (4 A3)(4 R)(2 R))
(meas (1 R))
(meas (2 R)(2 B3))
(meas (2 A3)(2 C4))
(meas (4d BN3)(8 C#4)(2 D4))
(meas (4d)(16 C#4)(16 BN3)(2 C4))
(meas (8 D4)(8 E4)(8 D4)(8 CN4)(8 BN3)(8 A3)
(8 B3)(8 D4))
Figure D-1
NOTES Example
Appendix E
Spatial Planner Graphics

E.1 Character Symbols

Many of the symbols in musical notation are characters. That is, they have a constant shape and orientation. These characters were hand digitized and made available as a sequence of points. A figure is specified by a list of triples \((X, Y, OP)\) where \(X\) and \(Y\) are spatial coordinates relative to the center of the figure and \(OP\) is either "move" or "line" indicating whether the operation should be MOVE_ABS_2 or LINE_ABS_2. When the location of a figure is calculated, these values are added to \(X\) and \(Y\) for each point and the proper operation is performed. Drawings of these figures are in Appendix C.

E.2 Beaming

Among the non-character symbols are the beams between adjacent eighth notes and smaller notes. These beams vary in length and slope for different combinations of notes. The slope of the beam must follow the general
slope of the notes it connects. We use linear regression to compute the slope of the notes in a beat-unit and use this slope as the slope of the beam. The stem directions of all the notes in the beat-unit are determined by the stem-direction that the note farthest from the middle of the staff would have if it were not beamed. The rule for locating the beam dictates that no stem in the beat-unit should have a length of less than a sixth (2 1/2 spaces).

We position the beam one octave (up or down according to stem-direction) from one of the notes in the beat-unit. This note is chosen so that all of the other notes in the beat-unit have acceptable stem lengths. For example, if the notes have up stems and the beam has a positive slope, then the note chosen is the highest note in the beat-unit with ties going to the first note. See Figure 47 for some examples of beaming.

E.3 Ties and Slurs

Ties and slurs are modeled as conic sections using the techniques outlined in [Faux and Pratt, 1979, pp. 17-36]. We find that the intersection of two conic sections, $S_1 = 0$ and $S_2 = 0$ specifies a family of
conics given by

\[(1 - q)S_1 + qS_2 = 0\]

In particular, let \(S_1 = l_1l_2 = 0\) and \(S_2 = l_3^2 = 0\) be degenerate conics where \(l_1, l_2\) and \(l_3\) are straight lines. The equation

\[(1 - q)l_1l_2 + ql_3 = 0 \quad (1)\]

specifies a family of conics that are tangent to \(l_1\) and \(l_2\) at their intersections with \(l_3\). See Figure 2-1.

For a tie or slur we let the point \((x_1, y_1)\) be the left-most point of the curve and let \((x_2, y_2)\) be the right-most point. We specify some peak by \((x_3, y_3)\). For ties, this peak is the midpoint of the two noteheads. For slurs, it is slightly higher than the highest obstacle that must be cleared. The height is a function of the distance of the obstacle from the midpoint of \(l_3\). The closer the obstacle is to the end of the curve, the smaller the height becomes. To specify the curve uniquely, let \(l_4\) be a line through \((x_3, y_3)\) and perpendicular to \(l_3\) (specified by \((x_1, y_1)\) and \((x_2, y_2)\)). Let \((x,y,y_b)\) be the
Figure E-1
The intersection of two degenerate conics
intersection of $l_3$ and $l_4$. Now let $(x_t, y_t)$ be another point of $l_4$ such that $(x_3, y_3)$ is midway between $(x_b, y_b)$ and $(x_t, y_t)$. If $l_1$ is a line through $(x_1, y_1)$ such that $l_1$ and $l_2$ intersect at $(x_t, y_t)$ then we have specified our geometry as shown in Figure E-2. The equation of $l_3$ is:

$$l_3 = a_3x + b_3y + c_3$$

$$= (y_2 - y_1)x + (x_1 - x_2)y + x_2y_1 - x_1y_2$$

Since $l_4$ is perpendicular to $l_3$, its slope is

$$m_4 = \frac{-1}{m_3} = \frac{-x_2 - x_1}{x_1 - x_2} = \frac{x_1 - x_2}{y_1 - y_1} = \frac{x_1 - x_2}{y_2 - y_1}$$

and its equation is

$$l_4 = a_4x + b_4y + c_4$$

$$= (x_1 - x_2)x + (y_1 - y_2)y$$

$$+ x_3(x_2 - x_1) + y_3(y_2 - y_1).$$

Solving the intersection of $l_3$ and $l_4$, we obtain

$$x_b = \frac{b_3c_4 - b_4c_3}{a_3b_4 - a_4b_3}$$

$$y_b = \frac{a_4c_3 - a_3c_4}{a_3b_4 - a_4b_3}$$

Then

$$x_t = 2x_3 - x_b$$

and
Figure E-2
The geometry for calculating slurs and ties
\[ y_t = 2y_3 - y_b \]

by our geometry. Using these values, we find that

\[ l_1 = a_1 x + b_1 y + c_1 = (y_t - y_1)x + (x_t - x_1)y + x_1 y_1 - x_1 y_t \]

and

\[ l_2 = a_2 x + b_2 y + c_2 = (y_2 - y_t)x + (x_t - x_2)y + x_2 y_t - x_t y_2 \]

Substituting into (1), we get

\[ (1 - q)l_1 l_2 + q l_3^2 = Ax^2 + Bxy + Cy^2 + Dx + Ey + F \quad (2) \]

after some manipulation where

\[ A = a_1 a_2 (1 - a) + a_3 q \]
\[ B = (a_1 b_2 + a_2 b_1)(1 - q) + 2a_3 b_3 q \]
\[ C = b_1 b_2 (1 - q) + b_3^2 q \]
\[ D = (a_1 c_2 + a_2 c_1)(1 - q) + 2a_3 c_3 q \]
\[ E = (b_1 c_2 + b_2 c_1)(1 - q) + 2a_3 c_3 q \]
\[ F = c_1 c_2 (1 - q) + c_3^2 q. \]

To produce this curve, we take advantage of the fact that it is relatively flat. Starting at \( x_1 \) we increment \( x \) and compute \( y \) until we reach \( x_2 \).

Rearranging (2) to solve for \( y \) we get

\[ Cy^2 + (Bx + E)y + (Ax^2 + Dx + F) = 0 \]
and

\[-(Bx + E) \pm (Bx + E)^2 - 4C(Ax^2 + Dx + F)\]

\[y = \frac{\text{--------------------------}}{2C}\]

and when \(C = 0\)

\[Ax^2 + Dx + F\]

\[y = \frac{\text{-------------}}{Bx + E}\]

This geometry creates an ellipse so we use the positive square root when the curve is concave down and the negative square root when the curve is concave up.

The geometry we have used works quite well in most cases. Figure 46 shows an example of the ties and slurs that can be produced.