Page Turning — Score Automation for Musicians

Honours Project
November 1999

John R. McPherson
Supervised by Dr Tim Bell

Abstract

This report examines various aspects of the use of computers for the purpose of displaying sheet music for musicians. The main focus is on the layout of the displayed music and methods that can be used to simulate physical page turning to provide new information for the musician. The techniques that may be used to update the field of view depend on factors such as the amount of information to be displayed at any one time, from which direction the new information should come, various monitor attributes such as size, brightness and resolution, and how the transition should be made between the current view and the next view.

A small feasibility study was conducted on a group of musicians to determine which of the page turning concepts under consideration should undergo further evaluation for potential benefit.

The results of this study suggest that a few of the identified techniques could be withdrawn from consideration. However, individual preferences appear to have a large effect in the usability of the different methods.

Contents

Chapter 1

Introduction

As computers become more accepted by all areas of society, they are being increasingly used in non-traditional areas such as fine arts, particularly music. Younger people in particular are comfortable with using computers, and are beginning to expect to use them in a variety of ways. Currently, the main role of computers in music is to aid in composition by storing the composer's ideas, and for playback (sequencing) of these works.

This report investigates the main issues involved in a system designed to assist musicians during practice or performance by displaying their music on a computer monitor rather than on paper. The emphasis in this report is on how the musical score is presented to musician.

A system like this should offer advantages over currently accepted methods for it to be of use. Possible advantages that have been identified are as follows.

- Hands-free display and page turning of the music is possible, allowing the musician to play uninterrupted. This is also an issue in orchestras, where some of the musicians must stop performing to turn the page.
- The display of the music is now silent. This may be very desirable for example during performance or recording.
- Assistance in determining where the musician is up to in the score through animation techniques.
- A computer-based system is more flexible in terms of offering different methods to suit individual preferences. These methods may include resizing the displayed music, modifying attributes such as brightness, contrast and colour, the animation techniques used, if any, and the actual page turning method used. An example is that a pianist's eyes
    - are often much closer to the sheet music
    - normally read two staves of music (one for each hand) rather than one

  when compared to a cellist, which may influence the preferred settings.
- A computerised page turning system offers the more general advantages of a digital representation, such as 'Digital Library' storage and retrieval

features, no deterioration of image quality over time, and ease of sharing and transmitting data.

The main objective of this report is to investigate relevant factors to, identify problems with and propose possible solutions to the task of displaying music to aid musicians in the performance and practice of music. This means the focus of the project will be what the music will look like and how the transition from one page of sheet music to the next will be displayed.

The project also involves designing and carrying out an experiment that implements various ideas for a page turning system. The system was tested on a small group of professional musicians, and feedback will be sought on design issues such as preferred methods for scrolling the music, and animation used for assisting the musician to easily locate his or her current position. The purpose of this experiment is to determine which, if any, of the tested methods are shown to hold potential, and require further study, and which methods can be discounted.

An overview of page turning ideas and some of the current obstacles to this application is given in chapter 2. These issues include:

- reading from paper compared with reading from monitors
- the effect that display hardware has on the quality of the
- what a page turning system could achieve
- the use of animation techniques, and the effect they have on readability.

Examples of these perceptual issues include the effect of clarity and brightness on legibility. CRT monitors are now of sufficient quality that the above qualities are surpassed. Many standard monitors today are capable of refresh rates of 75 Hz, with graphics cards capable of over 100 frames per second. Some of the current problems preventing practical use of page turning systems are due to the limitations of monitors, as the remaining hardware is powerful enough to render fast-moving graphics smoothly. Some of these are physical problems, such as the available screen area, and the size and weight or the monitor, for a traditional CRT display. On the other hand, current liquid crystal display (LCD) technology that allows smaller portable screens have problems with resolution and refresh rates, with fast movement of graphics being difficult to perceive.

This leads on to discussion of current systems that are relevant to page turning in chapter 3. Some of these implementations use methods and ideas that can be adapted for page turning purposes, while others are needed in their entirety. This includes the prevalent music data formats, which are described, and their features and limitations with respect to page turning are discussed. Tied in with the digital music formats is the applications that interpret and generate the data. Some of these applications have features desirable in a page turning system, such as the ability to zoom and resize the displayed music. However, the biggest problem is that these applications are all written either for the purpose of entering or typesetting music, and not with playback at the forefront of the design. Also, briefly discussed is another potential way to convert printed music into digital form — Optical Music Recognition (OMR). This is likely to become more widespread in the future as the technology improves. There are also other technical areas that need to be explored, for example:

- systems designed to convert audio signals into a digital form, allowing recognition of the notes played
- systems to convert printed music into a suitable digital form
- allowing a musician to make annotations to the music for practice or performance purposes.

The other main area that needs improvement is the feedback mechanism from the user to the system. The two fields investigated here are pitch recognition and tempo matching, and both are still under research. Score following is the part that has to match the internal music representation with the input and decide where the current position is, and this is quite a difficult task.

Chapter 4 discusses the need for the preliminary evaluation study, the design decisions made, and the implementation of the necessary software. These decisions include hardware considerations and animation methods. Some background is given regarding animation used to aid in reading music. The main aim was to investigate the usefulness of various methods for updating what the musician sees on the display, so that those of value could be examined in detail to determine which are most desirable. This meant implementing a program that displayed music, and used different page turning techniques. The implementation, and its functionality, are explained.

Following this, chapter 5 discusses the evaluation study, and the results are interpreted. Improvements that may be made are offered, and what was learnt is described. Particularly, explanations are offered for the reasons behind the users' comments regarding what they did and did not like about the demonstrated techniques.

Chapter 2

Page Turning Description and Problems

While much research has been carried out in the past examining issues related to reading text from monitors [1], such as anti-aliasing, speed and accuracy of reading, and fatigue, very little research has currently been done for the reading of music. One relevant area was investigated by Picking [2], who examined the effects of animation on reading static (that is, not moving) music, and the accuracy and usability of different animation methods as perceived by a group of musicians.

However, little research has been done in the area of page turning. The main area of uncertainty is the policies that can be used to show musicians a whole piece of music, when only a small part of it can be displayed at any time.

Other than the method used for page turning, other features are possible that are useful to the users. For example, demonstrating the correct notes for a bar or line of music to a musician may dramatically improve his or her learning.

2.1 Page Turning methods

One possible method for showing more music than can be displayed is to stream the music as one long horizontal line. Potential problems with this include the limited amount of music displayed, as it may be that some musicians like to see upcoming music far in advance of where they are currently at.

Displaying the music vertically may be a better option, as the reader can see more of the music. This may open up problems of how the musician keeps track of the current position in the score if the music is constantly moving. For both of these methods, some form of scrolling must be used to add new information as it is required and to remove old information when it is no longer needed. Again, little research has been done into the effects of reading music from monitors as the music is moving. The movement may not be constant — it may be that some musicians find faster movement followed by stationary periods more readable.

Another possible method for updating the displayed music is to not use movement, but rather overwrite past music with future music. This means that the musician must keep track of where on the monitor both the current position and future position are, although visual cues can help the musician quickly locate old, current and future music. How much music should be displayed at any one time is a decision that must be made, although this is limited by the size of the monitor and the desired size of score. It may be that musicians feel most comfortable working with the same amount of information that would be printed on a single page of music, or two side-by-side sheets of music, as this is the form they are used to. Then again, individual preference may come into it, with less that one page, or between one and two pages worth of music on the monitor. This method also needs to replace the old music with new. The timing of this needs be decided — for example, should the music be updated line by line as each line is finished, or should updates occur less frequently but update more of the music?

One page turning method that is not discussed in detail in this report is using some sort of fish-eye view. This is really a specialised version of the horizontal streaming method above, and may need to be looked at in more detail in the context of this method.

A. Forsberg *et al* [3] describe a tablet for pen-based input of music and they had the problem of displaying more music than could be shown on the small display. Their partial solution was to to have music already entered 'shrink', so that the scale of the music became smaller closer to the left hand edge of the tablet. Their name for this was the 'perspective wall metaphor', which is attributed to Mackinlay *et al* [4], and gives the user more context to work with. Figure 2.1 shows an example of this as used by Forsberg. This could be modified for page turning purposes to have upcoming music gradually increase in size, so that the current one or two bars are the largest.

Figure 2.1: A demonstration of a fish-eye view

2.2 Eyesight Issues

For scrolling, there are well-defined [5] properties for perception of motion. For example, for the eye to perceive motion as smooth, it must see a minimum of around 20 frames per second, while monitors based on cathode ray tubes (CRT) must have a display rate of at least 50 frames per second for their inherent flicker to not be distracting to the human eye. CRTs typically use interlacing to remove create the appearance of non-flicker. Interlacing refers to refreshing only every second line of the monitor, so the first pass updates the odd lines and the second pass updates the even lines. This means that the effective update is 60 fps when the actual picture being created may only be updating at 30 fps. Liquid crystal displays also typically use at least 60 fps.

Sloboda [6] describes in detail how musicians tend to read music, especially pianists who must read information from two lines of music at the same time. He also discusses the effect of both the style of music and the experience of the musician on the amount of music needed for look ahead to provide fluent performance. For example, the style of music may affect whether a pianist reads across the upper stave of a bar then the lower stave, or jumps up and down between them throughout the bar. He also suggests that a musician needs to be able to read at least seven notes ahead of the current position. Also, an experience musician reading music will look further ahead if the music consists of 'standard' chord progressions and 'normal' harmonic progressions, rather than unexpected or unusual notes.

How the music is presented is also of importance. The typesetting of the music, and the amount of whitespace between symbols is an art, and how computer programs should get this right has been investigated [7]. Other characteristics can have an effect on readability. For example, "...brightness conveys shape more effectively than hue, but hue provides more accurately distinguishable display levels than does brightness." [8]

The use of animation has already been shown to offer benefits in reading music and text from monitors. Picking [2] describes a small study where musicians are asked to sight read music from a computer monitor while the music is played to them — however, they must record where in the music three deliberate mistakes were made. Four different methods were used to highlight the current position in the score as it was played, and the results suggest that the more animated methods were preferred by the musicians. However, this study involved only one screenful of music, so the problem of page turning was not addressed.

2.3 Disadvantages of Using Page Turning

One potential disadvantage with a page turning system is the effect of reading from a display for a long period of time. Techniques may have to be developed that do not cause irritation to a viewer. Current and future PC graphics accelerators offer good improvement in quality for very little cost [9], and may help overcome any potential problem. Other developments include liquid-crystal displays that use newer, active matrix technology offer better brightness and less blurring than passive LCDs [10].

Another problem with current monitors is their screen area. 2 side-by-side A4 sheets is 297 millimetres high by 421 millimetres wide, which works out to a diagonal of just over 20 inches. Monitors above this size are not yet common. Also, the physical bulk of current monitors may make them unsuited for page turning. Portable displays, such as liquid crystal displays, may not be quite sharp enough for page turning purposes, although as mentioned above, this area is constantly changing.

Another disadvantage is the reliance on a computer during an important event, such as a performance. Hardware, software and power failure are all possibilities that can cause grief at inopportune moments. There is also the overhead the musician has of learning how to use the system.

Chapter 3

Systems Relevant to Page Turning

This chapter examines current concepts and implementations that have potential uses in designing and implementing a page turning system.

Some of the more popular music notation software applications are evaluated, and the various file formats that are used to store musical information, and their relative merits, are described in section 3.2. In particular, the current dominant file format, MIDI, is described in detail. This section also describes the problems faced by music editing programs, such as the lack of a formal set of rules for musical notation [11]. Individual rules describing the placement of an notational element on the page are not difficult, but the interaction between rules, such as precedence, are not well-defined — they are often subjective, with music copyists learn by example. This problem becomes more complicated when it is realised that composers sometimes deliberately 'break' these rules, perhaps because there is simply no better way to direct the musician(s) how to play the score.

Another application area that needs to understand music notation is Optical Music Recognition (OMR). This technique involves interpreting the symbols in a picture, such as a scanned image of sheet music, and recreating the information in a format that encapsulates the implied audio content.

One desirable feature of page turning systems that is not yet readily available is the ability to listen to the performed music and synchronise with the musician. Section 3.4 describes recent work in score following, and techniques for synchronising the page turning with audio input are explored.

3.1 Existing Systems

Currently, most software packages available for home and professional users are designed to aid in the composition and sequencing of music, not to help the performance or practise of it. The features that are relevant to page turning are discussed here, and those which are relevant but not suited are described.

Also discussed here are some prototype page turning systems that have been developed.

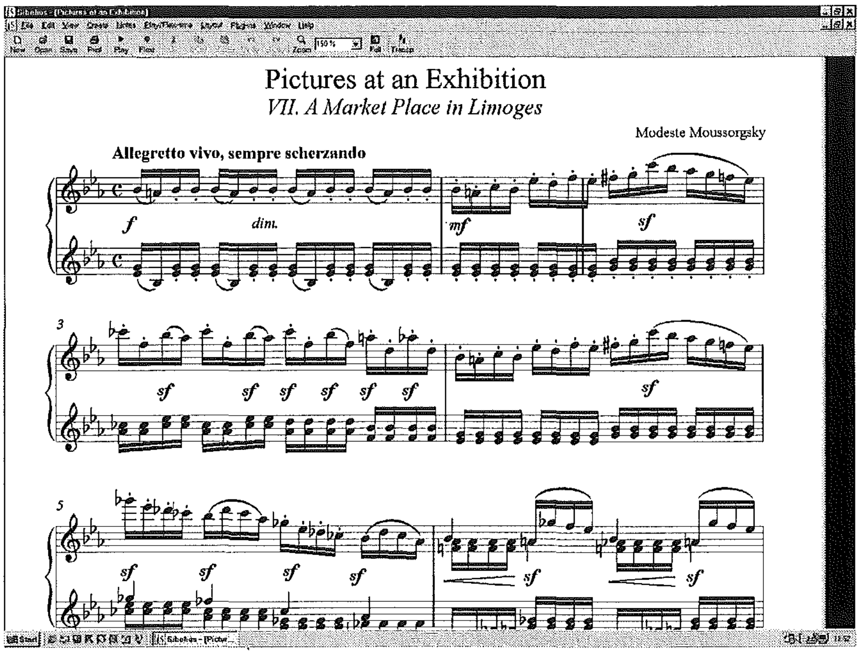


Figure 3.1: Screenshot of the Sibelius music typesetting program

3.1.1 Notational Software

Solero

Solero, by Sunhawk[1], is a commercial system for designed for displaying and playing downloaded sheet music. Copyrighted music may be downloaded for a small fee that also covers royalties. The system uses MIDI (MIDI is discussed in section 3.2.1) for output. One particular nice feature about this software is that it highlights each note as it is played. This software appears to cater primarily to people wanting to learn their favourite 'pop' songs, and this highlighting feature may improve learning speed. The two main disadvantages of this software are the page turning method, and the fact that closed, proprietary file formats are used. The page turning method consists of the music jumping up when the end of each line is reached, with a new line of music appearing at the bottom of the display. As discussed in chapter 5 of this report, this is probably not as useful as other possible methods.

Sibelius

Sibelius, made by Sibelius Software Limited[2], is available for both Windows and Macintosh PCs, and is designed for entering and typesetting music. Data is entered manually, either by using a mouse or by using MIDI-capable equipment, such as an electronic keyboard. Sibelius plays music back through MIDI, and animates the score with a vertical blue line that moves along the current stave as the notes are played. If the next note to be played is not currently displayed,

[1]Sunhawk have software available for free download from their web site, `www.sunhawk.com`.
[2]`http://www.sibelius.com`

the view will jump so that it is. This means that the display may jump twice during the same stave if it is both off the bottom of the screen, and has some notes to the right of the screen. As for Solero, this is could be done using a better method.

Sibelius displays the music better than most of the other programs, taking advantage of higher resolutions, and the use of textures that seem to make the music easier to read. It uses its own file format, but can also export to MIDI format. It can also import from MIDI, although this is limited by problems inherent in the format. This is discussed later in the discussion about the MIDI format.

Lime

Lime[3] is free to download, fully-functioning, for evaluation purposes. It has very little animation during playback, consisting of an option to grey notes that have been played. Also, the music must be scrolled manually if the user wants the display to be updated.

Lime is designed to be a notation editor, and, like the others, uses MIDI for composition and playback, although input can be done through the mouse and keyboard. Lime is relatively small, and quite flexible in terms of allowing a user to manually alter the score, although the default rules get the notation correct most of the time. Although it uses a native file format, it can import and export both MIDI and NIFF files. Incidentally, Lime was written by the authors of [7].

Rosegarden

Rosegarden[4] is a completely free open-sourced notational editor and sequencer for `Unix` systems. Input can be through the use of both a MIDI keyboard and a mouse. Music is displayed as one long horizontal stave, and during playback this can be greyed out. The user interface is starting to look dated, but an updated version is in progress, at `http://cvs.gnome.org`.

3.1.2 Development and Experimental Systems

The Digital Music Stand [12] was a conceptual music stand designed in 1995 aimed primarily for use by orchestral musicians. Although not implemented, care was taken to design a tool of maximum benefit to the users, with constant communication with members of the Pittsburgh Symphony Orchestra. The tool would encompass orchestral- and sectional- wide communication, with synchronous automated page turning and a centralised digital library to store the music. It would also have tools for use by each individual for practice such as a metronome, a tuner and pen-based on-screen annotations. However, because it was not implemented, the issue of page turning methods was not looked at in detail, although their design decisions settled on using a fold-able display consisting of two 9" by 12" (22.8 by 30.5 cms) high-resolution screens.

[3]Lime is available from `http://datura.cerl.uiuc.edu`
[4]Available from `http://www.bath.ac.uk/~masjpf/rose.html`

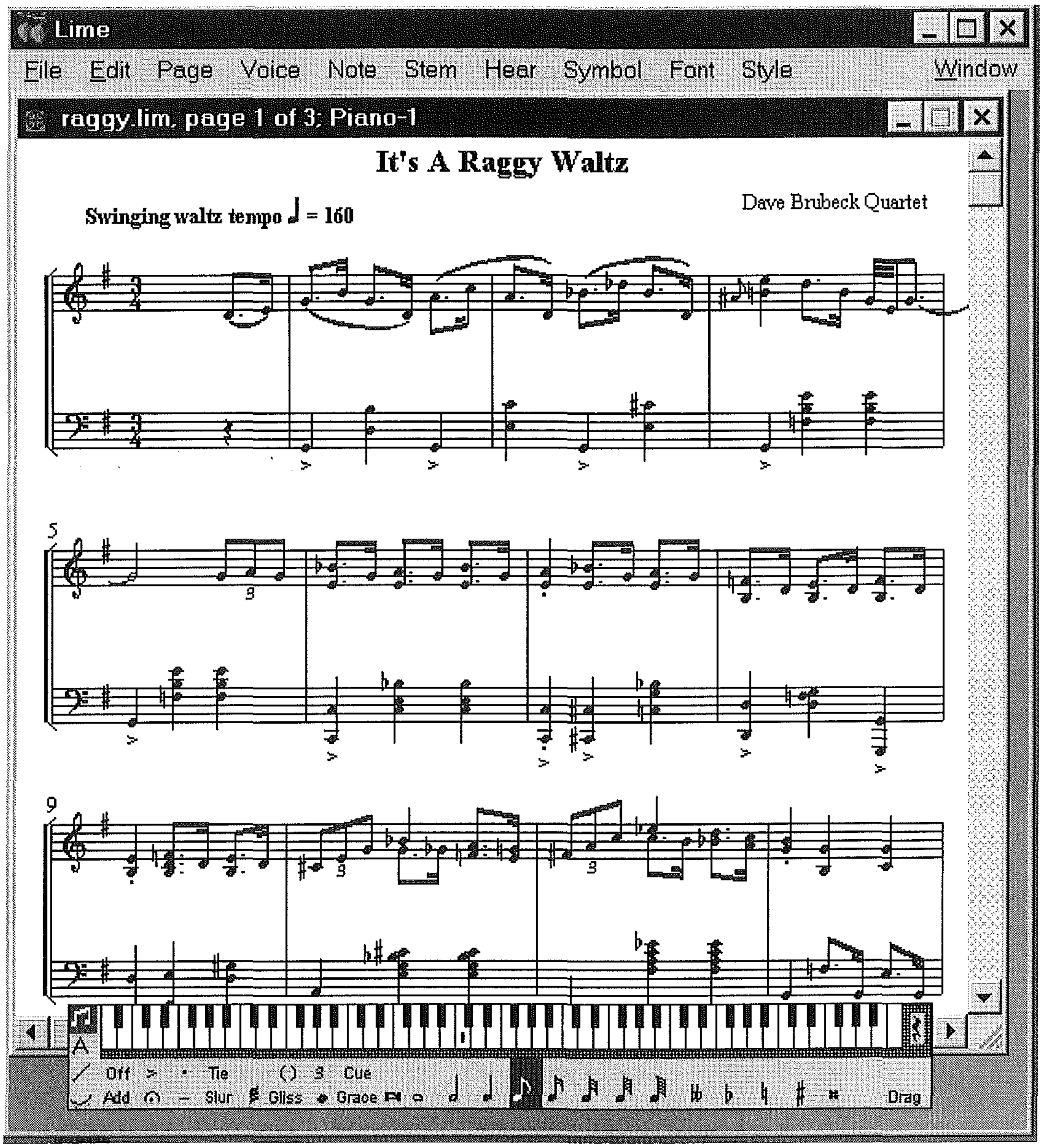


Figure 3.2: Screenshot of the Lime program

Another system with the same goals in mind was designed by Bellini *et al* [13]. Their system, Music Object Oriented Distributed System (MOODS)[5], is object oriented for all aspects, such as data representation, and distributed communication. A prototype system has been developed, and public demonstrations have been given of it being used by groups of up to five musicians. Each musician has a digital stand (Distributed Lectern Interactive Object Oriented, or DLIOO), and the stands are linked to a central server that stores the music. The page turning method used to display the score for the musician consists of horizontally splitting the display area into two, with new music appearing in whichever portion does not contain the current position. Unfortunately, apart from stating that scrolling the music is undesirable, no rationale was given for this design decision. Timing and synchronisation between the performance and the display system is manually, with an observer following the score and adjusting the tempo for the system if they start to drift apart.

3.2 Data Representation

A page turning system needs to permanently store some information about each music score to achieve the desired features stated earlier. In particular, the format that the data is stored in should provide enough information to allow the page turning system to:

- typeset the score correctly
- contain extra information about how the score is 'broken up' for page turning purposes – for example, page breaks if that is appropriate for the updating method
- store information about the audio content of the score
- allow the page turning system to synchronise between the audio and graphical representations.

Another desirable feature of general musical data representation formats (that is, not just with respect to page turning) is the ability to display and convey music even though it is syntactically 'illegal', as notationally incorrect scores are often produced [14]. Figure 3.3 is one example given by Donald Byrd [15] showing breaches of convention by famous musicians. This example is from Johannes Brahms's Intermezzo opus 117, number 1. The major typesetting *faux pas* is that the first five notes in both staves are all played simultaneously. The point here is that despite sounding at the same time, these notes have four different horizontal positions on the score. Other 'features' it shows are:

- the first group of three notes in the lower clef have stems up instead of stems down
- the slur over these three notes is over the beam rather than under the note heads
- accidentals for two notes are not on the immediate left of them, but rather on the left of the preceding note.

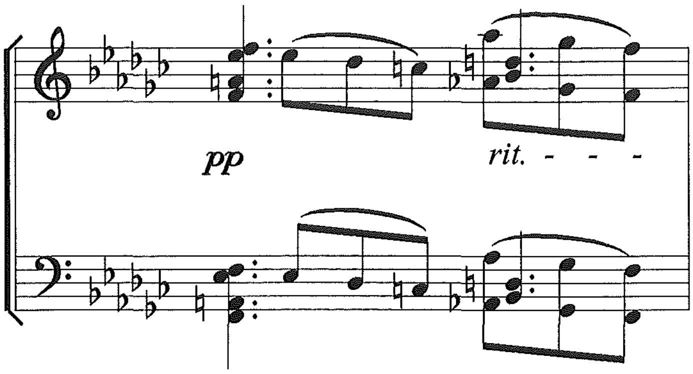


Figure 3.3: An example of notational rules broken for the sake of readability

Creating digital representations of a score is often a time-consuming task, as the music normally has to be manually entered through the use of an electric keyboard and then touched up to include markings such as phrasing, fingering and dynamics. Some of the more popular music notation software applications are discussed here, with the focus on which features are desirable for page turning, and which features are lacking. An overview is given here of some relevant file formats as well as methods for creating and viewing these files.

Current data representation methods are described, covering the major data file formats MIDI, NIFF and SMDL, their relative merits, and how they could be used in a page turning system. For example, the NIFF format provides enough information to describe both the graphical and audio representations of the music, while MIDI provides an accurate audio representation but very little information about the graphical layout of the music. An advantage of these formats is that they are open, non-proprietary standards.

3.2.1 MIDI

Currently, one of the most widely used formats for current systems is the Musical Instrument Data Interface (MIDI). This standard is widely supported in electronic keyboards, digital synthesizers, and computer sound cards, and allows two or more of these interfaces to connect for message-passing purposes. MIDI itself is not responsible for the audio output – it is up to the receiving hardware how a message is interpreted and what action is taken. This means that MIDI can also be used for other purposes such as controlling lighting effects. However, the majority of MIDI implementations are designed to create audio output, such as a synthesiser built into an electronic keyboard. Also, some sound cards use *wave-table synthesis* to create more realistic musical instrument sounds. [16].

Overview of the MIDI protocol

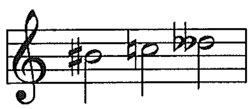
MIDI is an event-driven system that passes messages usually consisting of two or three bytes. These messages are includes information such as the pitch and volume of the note. The pitch is represented as 0 being the lowest note and 127

[5] `http://aguirre.dsi.unifi.it/~moods`

being the highest note. Compare this to a piano keyboard, which has 88 keys. 'Middle C' (the centre key on a piano) corresponds to a MIDI pitch value of 60.

For example, a *note on* event consists of 3 bytes – the first byte giving the pitch, the second byte giving the volume (called *velocity*) and the third byte giving the *channel* used. If the receiving equipment is some sort of synthesizer, then this message will cause a tone to be played. Sometime later there will be another event to stop this note, either a *note off* event, or a note on event with the velocity set to zero.

There are sixteen channels available, and an instrument (also called a voice) is assigned to each channel. For example, channel 0 may be assigned to a Grand Piano. On MIDI equipment that has the 'General MIDI' capabilities, channel 10 will only be used for percussion instruments.



C natural
B sharp D double flat

Figure 3.4: The same note, part 1

Middle C

Figure 3.5: The same note, part 2

This gives rise to a serious problem, at least as far as using MIDI information to imply graphical layout is concerned; this is that there is no guarantee that the note will be put on the correct line of the stave, as the pitch information provided by a MIDI signal is only for the acoustic signal. For example, figure 3.4 shows notes are three different notations of exactly the pitch, on a single stave, while figure 3.5 shows the same note drawn on both the treble and bass staves. For a pianist, a note drawn on the lower stave would normally be played with the left hand, while the upper stave indicates the right hand.

In both examples, the notes all represent the same physical note on a musical instrument, but depending on the context of the music may make a large difference to the way the score is read.

Another example is that an accidental is sometimes shown on the score even when it is not needed, just to remind the musician, or if the composer feels it may not be obvious. Again, MIDI does not encapsulate this.

To give an example of how MIDI data is passed between MIDI equipment (that is, in real time) consider the note described in figure 3.5. This note is Middle C, and is a minim (which lasts for two beats).

The data produced by a MIDI keyboard may look like this:
0x90 0x3C 0x40
< *delay* >
0x90 0x3C 0x00.

The first byte is a status byte, where the 0x9n represents a note on event, and the 0 means channel 1. The second byte (the first data byte) represents a pitch. 0x3C (60 in decimal) represents middle C. Finally, the third byte (data byte 2) represents the velocity when the message is a note on message.

Notes:

- n describes which channel. Both events must have the same channel.
- The timing of the second event is entirely dependent on how long the note is depressed. If the music is played at 200 beats per minute, then the second event would occur $\frac{2}{200}$ths of a minute later (0.6 seconds).

Real-time issues using MIDI

MIDI data is sent serially, with each byte having start and stop bits, and exactly 3,125 bytes of data can be sent in 1 second, which corresponds to 320 milliseconds per byte [17]. Computer-generated MIDI data normally has very consistent lengths for each performed note, whereas humans can not physically create such accuracy. This means that when recording or importing from human-created MIDI, quantisation is used to round the note lengths to sensible values, as far as the computer is concerned. However, the problem with this is that there is no way to tell if the quantisation was correct, and all note lengths were interpreted correctly.

Although quite a lot of the score representation can be obtained from the information in a MIDI file, there is not enough to ensure consistency with the original. For example, there is no way to tell if a sounded note was a long note that was played staccato, or merely if it was a very quick note. Figure 3.6 shows two bars that may sound very similar when performed. The first bar contains a short note, followed by rests. The second bar contains one note, with the dot below it denoting *staccato*, which is literally Italian for attack, implying that the note is played quickly. From the timing information only, as MIDI would provide, it is impossible to determine which is the correct representation.

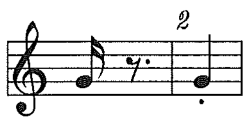


Figure 3.6: Two notes that may be performed similarly

MIDI by itself is not a good format to use for page turning. However, because of its popularity and success for the audio representation of music, it is possible to build a system that incorporates MIDI, using a separate representation scheme to show the graphical layout of the music. The MIDI events occur at the beginning and end of each note played can be used to determine the current position in the displayed score, so that co-ordination between the two representations is relatively straight-forward to implement.

3.2.2 NIFF

The Notation Interchange File Format (NIFF) [14] is designed so that both the page format and the audio representation can be interpreted from the file.

The NIFF standard was produced in co-operation with the publishers of the major music software applications (originally Encore, Finale, Score, Midiscan, NoteScan, SightReader and Coda, and later Mosaic, Overture, Cakewalk and Nightingale). The formal specification was released in August 1995 after about sixteen months of development. Two main reasons were given for getting competitors to work together to develop a common format. Firstly, the popularity of the World Wide Web, which rapidly increased the ability to share music between users, especially across different platforms. The second reason was the increasing accuracy and popularity of optical music recognition for automation of music data entry. A common format was desirable so that the OMR results could then be used in any application, rather than separate OMR software being developed for each.

Overall, NIFF has good potential for use in a page turning system due to its ability to accurately describe both the audio and notational representations of a music score.

3.2.3 SMDL

Standard Music Description Language (SMDL) was ratified by the International Standards Organisation (ISO) in 1996. It conforms to the ISO's Standard Generalised Mark-up Language standard, and is designed to describe music at a very high level for data exchange between music applications. However, the format does not directly describe the graphical layout of the page, instead using the familiar concept of links to external sources. For example, a SMDL file may link a MIDI file, or a portion of a MIDI file, with an image, say an X11 bitmap. In theory, this mapping could occur at a very low level, for example on a note-by-note basis. The idea behind this is to allow not just the audio and graphical representations of music to be stored, but to be much more flexible in linking any type of multimedia data to music.

However, partly because of this lack of a firm graphical representation, and partly due to commercial interests, this format is not yet widely used.

3.2.4 Other Methods

There are some programs designed for describing the graphical content of a score only. One that provides professional-looking typesetting is MusicTeX [18] [19], which has been superseded by MusixTeX. MusicTeX is an add-on package for TeX and LaTeX for typesetting music. Figures 3.4, 3.5 and 3.6 were prepared using MusixTeX.

3.3 Optical Music Recognition

There is currently quite some effort being put into systems that can scan sheet music and produce digital representations that encapsulate the graphical and performance information, such as pitch, note lengths, rests, and directions such as stacatto, slurs and dynamics. OMR offers a way to digitise sheet music on a large scale, and the accuracy of this method will improve in the future.

There are quite a few commercial OMR systems on the market. One commercial OMR system is NoteScan[6], written by Cindy Grande, who helped develop the NIFF format [14]. Midiscan, by Musitek[7], uses an interesting method to overcome shortcomings in the OMR. After sheet music is scanned in and recognised, Midiscan shows both the original and its interpretation line by line, to help users find errors.

One non-commercial OMR system is an online version of CANTOR [20] developed at the New Zealand Digital Library[8] that allows users to submit images of scanned sheet music, and receive a digital format of the result of OMR processing.

3.4 User Feedback for Synchronisation

A page turning system requires some input from the user for synchronisation. This information is needed to determine when the display needs to update both any animation as well as the page turning method itself.

As mentioned earlier, most current applications use the MIDI events for synchronisation as well as for audio output. However, this has some problems as far as a page turning system is concerned. Firstly, these systems are designed only for playback. Real-time input is not nearly as consistent in terms of note lengths, so ambiguities may arise, and notes may be missed accidentally. Secondly, recognition of analog audio waves is difficult due to the variation in sound quality made by different instruments. Thirdly, how to actually use the feedback from the musician to determine the position in the score is also difficult.

This section gives a brief overview of the current research areas that may be used to help synchronise a musician's performance with the digital representation of the score.

3.4.1 Pitch Recognition

Pitch recognition refers to the process of recognising an audio (analog) signal as a specific pitch or tone. This then allows software to follow the score by tracking the played notes and comparing them to the expected notes. Pitch recognition is difficult because many musical instruments produce harmonics, making it hard to distinguish the actual note or notes played. The unique combination of harmonics produced by different instruments is partly what allows the human ear to distinguish between, say, a brass instrument such as a trumpet and a woodwind instrument such as an oboe.

It is even harder to determine which combination of notes are being played when a polyphonic instrument such as a piano is used, as these instruments use strings (which create more harmonics than, say, brass instruments). However, commercial systems currently exist that use digital signal processing to create midi signals from audio in real-time. In particular, a great deal of effort has been put into creating pitch-to-MIDI converters for the electric guitar, probably

[6] http://www.musicwareinc.com/prod14.htm

[7] http://www.musitek.com

[8] http://www.nzdl.org

because of the large number of the guitar enthusiasts. For example, MidiAxe[9] and Axon Ax100[10] are both systems that convert notes played on an electric guitar into MIDI signals for use by MIDI equipment.

Another system that uses pitch recognition, although not in real-time, is the *MELody inDEX* (MELDEX) [21], which is available[11] on the World Wide Web. Given a short audio extract of monophonic (single harmony) music, it will perform pitch recognition, and then perform pattern matching on a digital library of tunes. Its pitch tracking algorithm is based on a method developed in the late 1960s by Gold and Rabiner [22] for speech. One interesting aspect of the MELDEX system is the method used to perform the pattern matching. While using the intervals between notes is an option, the default method is to use the change in pitch from the previous note, with `D` representing descending, `U` representing ascending, `R` representing a repeated note, and `*` representing the first note. For example, the first few notes of 'Happy Birthday To You' would be represented as `*RUDUD`. This has several major advantages: firstly, the accuracy of the pitch recognition as well as the performed music is no longer critical, and secondly, pattern matching on this type of data is faster than other methods.

3.4.2 Tempo Matching

One solution offered to the difficulties of implementing digital signal processing is tempo matching. This means using the temporal information of the audio rather than the pitch to decide where in the score the musician is up to. This has the advantage of being able to interpret audio that is a mix of more than one instrument, something that becomes increasingly difficult for pitch tracking.

An example system is by M. Goto and Y. Muraoka [23]. Their implementation worked particularly well for music played at a regular tempo, such as music having a drum-kit maintaining the beat. When it was tested on a variety of 'popular' songs, their system correctly kept time for twenty-seven out of the thirty songs. The system had some drawbacks however; probably the major drawback was the amount of processing required to do this task, with the system consisting of a Sun SPARCSTATION 10 connected by SCSI to a large parallel computer (a Fujitsu AP1000). Another drawback was the requirement for the piece of music to have a near constant tempo. Most classical works are performed with quite marked variations in tempo in somewhere in the score. Many contemporary pieces are also performed with tempo changes, although often this are not deliberate.

Tempo matching shows good potential for use in synchronisation in a page turning system.

3.4.3 Score Following

Score following refers to matching the notes being played to the location of the notes on the musical score. Most of the systems currently in use for score following are implemented using MIDI signals. However, it is more desirable for an implementation to use the audio signal from real instruments, rather than the digital signals from electronic instruments. The requirement of using

[9] `www.midiaxe.com`

[10] `www.midi-classics.com/p9753.htm`

[11] part of the New Zealand Digital Library, available at `http://www.nzdl.org/meldex`

an electronic instrument would severely limit the usefulness of the system for musicians wanting to practice or perform on their 'real' instruments.

Some of the problems associated with score following are as follows:

- The computational load required for the score following algorithms that perform real-time pattern matching between the input and the score is often very high.
- Errors that occur in either the pitch recognition system or in the performance itself must be identified. In other words, the system must be tolerant to variations in the actual input from the expected input.
- Similarly, slight variations in timing of the performance should not affect the system's ability to continue.
- Some pieces of music require improvisation by one or more of the performers, and the system should be able to handle this.

Despite these problems, score following systems are only going to become more accurate and usable with time.

3.5 Advantages of Digital Representation

As mentioned in the previous section, digital libraries such as the MELDEX system [21] [24] allow content-based retrieval, where the user supplies a short sample of the piece they are trying to recall. Of course, these systems also allow retrieval on other information such as title, composer and lyrical content. Indexing systems can allow much quicker retrieval when managing large collections of scores.

This also means that music can be stored on a central server, and only accessed when required. As mentioned earlier, an example of this is the Solero software by Sunhawk, which allows users to download music into the browser for a small fee, typically US$5–10, that includes royalties. Other advantages that are not specific to music include the ability for quick and lossless data transfer.

Apart from data access, digital representations are much easier to manipulate; for example, convert to other file formats required by other applications, or the ability to customise a system for a user's personal preferences, such as scaling, or changing the colours used.

Chapter 4

Experiment Design and Implementation

To evaluate the different page turning methods that have been identified, a preliminary usability study was carried out.

The aims of the study were to:

- determine the end-users' personal preferences and feedback for page turning methods, and determine the variability, if any, of these preferences between individuals.
- get more information about the readability of music displayed on monitors.
- get more information about the usefulness of animation in a page turning setting.

The reason for investigating these aims is to help prepare future research by focusing on the one or two page turning methods that are most likely to be accepted by the end users, the musicians.

Because there are currently no systems available that allow different page turning methods, a system had to be designed.

Picking's experiment [2] studied the effect and perceived usefulness of various animation methods on the reading ability of musicians. However, his study was only done using extracts of music that could be displayed on the monitor *in their entirety*, so the effect of interactions between animation and page turning, and between animation and scrolling, was not observed. Specifically, the study aims to help reduce the uncertainty in potential methods used for page turning.

This uncertainty includes the following ideas.

- New music appears horizontally or vertically.
- Animations cues, such as greying out, or a current bar or note marker.
- If scrolling is used, the overall rate of scrolling is determined by the performance tempo, but within this it may be better to have 'faster' scrolling followed by periods of no movement, or it may be better to have a constant rate of movement.

- Replacing part of the currently displayed music that has already been played with the new music. This may be make it difficult to keep track of the current position.

- Splitting the display vertically, using the traditional idea of two currently displayed pages at any time.

Current notational and typesetting software, such as Sibelius, Encore, Lime and Cakewalk, all use jumping methods to display music during playback. For this reason, jumping methods were included as part of the experiment to compare the effectiveness of the 'established' methods against other possible techniques.

4.1 Display Issues

Possible hardware options for a page turning system include using a standard monitor, connecting two monitors and graphics adapters, using some sort of projection, and using a Liquid Crystal Display, either flat-screen or laptop display.

The prototype system should allow the size of the displayed music to be modified. The graphical user interface allowed the display resolution to be quickly changed, which effectively meant the music could quickly be zoomed in or zoomed out.

The system was designed to use a standard 17" monitor. The reasons for this are the availability of these monitors, issues involving the latency of LCD displays, and the nature of the study. For a follow-up experiment, where the system is being used to look at the usability in much more detail, more consideration should be given to the other display possibilities.

For pianists, an electric keyboard should be used due to the difficulty in placing a monitor at the right height on a piano. This problem will continue to exist until monitors are built that are small enough yet of an adequate quality for page turning purposes.

4.2 Animation Issues

Picking has demonstrated animation techniques used to help musicians follow the score while reading from a monitor. However, his study did not account for two things that need to be investigated for animation in a page turning system. Firstly, musicians may find it harder to keep their place in a larger piece of music, especially if the size means that page turning must be used. Picking acknowledges this as being outside the scope of his study. Secondly, now that the musicians are actually playing as well as reading, some of them, particularly pianists, will need to look away from the monitor at some stage due to the nature of their instrument.

To evaluate animation techniques that help the musician recover the current position on the monitor quickly, a number of methods can be used. One is to only highlight the current note (or bar or stave), as was used in Picking's study, to see if the length of the music does make a difference.

Other methods that may be more helpful with longer scores include greying out played music — either note-, bar- or even stave-at-a-time — to help the musician quickly locate the current position on the score, and animating a real-life page turn from right-to-left, if this makes sense for the chosen page turning method.

4.3 Implementation

The were generated using Lime, as the image quality produced from the software was vastly superior to scanned sheet music. Another advantage with using notational software is that MIDI output is available, allowing samples to be played to the musician as a learning tool, as described above. The pictures are displayed as bitmaps rather than rendered from the digital representation of the score due to the complexity of rendering systems. It was felt that this was adequate for the purposes of this evaluation study. The graphical language `Tcl/Tk` was used due to its suitability for prototyping and evaluating graphical interfaces.

The *Wizard of Oz* method, where the experimenter mimics the expected behaviour of the fully-functioning system, was used to keep the displayed music synchronised with the performer. This mimics the score-following functionality that is expected in a proper page turning system.

The study involved the musicians sight-reading from the monitor, and filling out a questionnaire comparing their perceived ease of reading for each method with the ease of reading from paper. The scale used ranged from one to five, with one representing much harder, two representing harder, three representing the same, four representing easier, and five representing much easier.

It was explained to the musicians that how well they sight-read was not important at all. What was important was whether the system helped them in performing the music in front of them. This was an attempt to reduce any preference for later techniques because the piece was becoming more familiar. To further minimise this possibility, there were at least two pieces for each musical instrument, with the exception of the bass. This was because the piece chosen for the bass was sufficiently long that all of the methods could be evaluated without any part of the score becoming too familiar.

The page turning methods that they were asked to evaluate were:

1. 'Single Page'. This method displayed one page of music, and lines of music that had already been played were eventually replaced with new music, so that when the musician got to the bottom of the screen, the music was read from the top again. This method had the option of greying out the old lines of music as they were completed.

2. 'Double Pages'. As the name suggests, the musician saw the music in the form of two pages, side by side. This method updated the left-hand-side staves of music when the musician was reading from the right-hand-side, and vice versa. This method also had the ability to grey out the old staves.

3. 'Horizontal Scrolling'. This method involves one continuous line of staves, taking up the width of the screen and scrolling from the right. This method has the ability to modify the scrolling speed.

4. 'Vertical Scrolling'. This method appears to 'join' the traditional sheets of paper, top to bottom so as to look continuous. This method also allows the scrolling speed to be changed.

One final point about the displayed music is that a decision was made to write out repeats in full, rather than have the system repeat the music. This was because the aim of this study was to get an idea of the relative strengths and usability of each method. Another reason is that the different methods would implement repeats and other directions that 'jump' around the score in different fashions. For example, horizontal scrolling would probably have no choice but to rewrite the music, while double pages and vertical scrolling may take action depending on the size of the repeat and the distance of the jump.

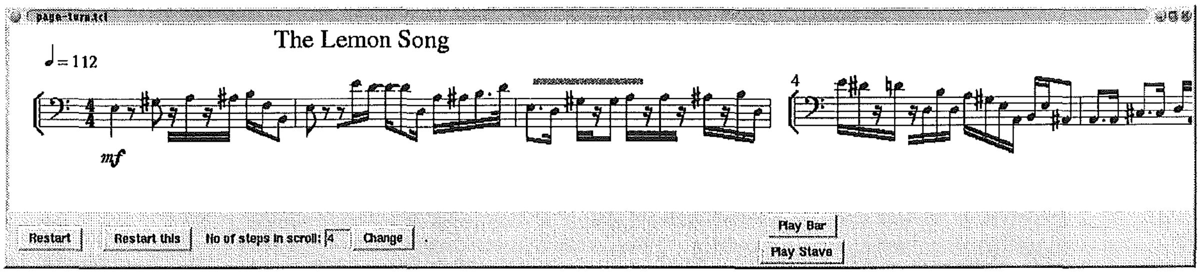


Figure 4.1: Screen-shot of Evaluation Program, 'Horizontal' Method

Figure 4.1 is a screen-shot of the Horizontal Scrolling method, showing the bass solo from Led Zeppelin's 'The Lemon Song'. In this method, the music moves to the left so that the bar under the cursor is always the current bar (apart from the initial few bars before that point).

Figure 4.2 shows a screen-shot of a piano piece (the 3rd Movement of Beethoven's 'Moonlight Sonata') displayed using the Single Page method. The musician has already played the first three staves and is currently playing the fourth. The third stave was greyed out when it was completed, and at the same time the second stave was updated to show new music. Similarly, when the musician had just finished playing the second stave, it became grey and the first stave was updated to show the next line after the bottom of the page is reached. This continues in a circular fashion until the end of the piece is reached. (Incidentally, the two obscured notes on the second stave will 're-appear' when the third stave is updated.)

Figure 4.2: Screen-shot of Evaluation Program, 'Staves' Method

Chapter 5

Experimental Findings

The evaluation study described in the previous section was carried out on a small group of musicians from the Christchurch Polytechnic School of Jazz in September 1999. Of the six participants, three were piano players, two were bass players (both used electric bass for the study) and the other was a trumpet player. A discussion of the results is given here, and general observations are made about what was lèarnt. Also, possible improvements are discussed.

5.1 Results

The numeric results given in the table below measure the users' comparisons between reading from the display and reading from paper, where 1 represents 'reading from the display is much harder', and 5 represents 'reading from the display is much easier'.

Instr.	Pages		Vertical		Horizontal		Overall
	Single	Double	Jump	Scroll	Jump	Scroll	Clarity
bass	3	2	4	4	1	3	3
bass	5	4	4	5	1	4	4
piano	1	3	1	1	3	3	2
piano	3	2	1	3	1	3	3
piano	2	1	1	2	2	2	2
trumpet	4	-[a]	2	5[b]	2	2	3
	ave: 2.5	2.4	2.2	3.3	1.7	2.8	2.8

[a]This method was not available for the B♭ instrument, the trumpet
[b]The trumpet player had use of a foot pedal

5.2 Discussion

Unfortunately, with this few participants there is insufficient data to significantly analyse any difference between the categories. Despite this, some insights can be drawn from the data to aid in deciding which of the methods show potential and which do not. Some observations that can be made from the results are as follows.

1. There was a general trend of the three pianists agreeing with each other for

each question, with a maximum difference in assigned scores of two. The pianists also consistently rated each method lower than the non-pianists, with the exception of horizontal jumping, which none of the musicians preferred. Also, both bass players preferred vertical movement over horizontal movement for both jumping and scrolling, especially when the music was made full-sized. These points would suggest that the instrument played has an effect on the usability of the demonstrated methods. This is more likely to be due to kind of music that an instrument is suited for, especially if the instrument can play more than one note at a time.

2. The three non-pianists rated vertical scrolling either 'easier' or 'much easier' to use than paper. The main reason given was the ability to zoom the music to full screen, making the score larger than printed sheet music. Every participant considered Jumping either as hard as or harder than smooth scrolling for both Vertical movement and Horizontal movement. When asked which method was preferred, the most popular were Vertical Scrolling and Single Page. These results indicate that jumping is too difficult to follow while playing, and can be given less emphasis for use in page turning in favour of the other methods.

3. The pianists commented that the double pages would have been rated higher if the music had been larger. One of the bass players also said that the music was difficult to read due to its size for this method. This suggests that the 17" monitor was too small for evaluating this method — recall that two A4 sheets side by side have a diagonal of just over 20".

4. The trumpet player tried some of the methods using a foot-pedal to send feedback to the system, rather than the experimenter controlling it. The trumpeter did not find either of the Horizontal methods easier than reading from paper, but did find Vertical Scrolling to be much easier. This suggests that if the user is expecting movement, because he or she caused it, the system is much easier to use. However, not every musician has the opportunity to provide feedback through a physical device such as a pedal, so this method should be considered alongside other feedback methods, rather than as a possible replacement.

5. One of the pianists commented on the lack of fingering information for the difficult piano piece (3rd movement of Moonlight Sonata). It was felt that this notation would have made a difference for sight-reading, and this suggests that performance directions such as fingering would have helped.

6. Generally, the clarity of the music on the monitor was found to be similar to the clarity of paper, with three saying it was the same, two saying it was slightly worse, and one saying it was slightly better. Two people said that brightness of the monitor, in particular the whitespace, was distracting. One of them felt there was too much white space, while other felt that the screen itself generated too much light.

One interesting point is that the musicians did not find the animation (a small red bar marker) useful — they found that they did not really notice it. This seems to reinforce Picking's finding that this method was judged less useful than several other animation methods by musicians. Three of the musicians also said that any movement at all became distracting.

5.3 Possible Improvements

Displaying the music on a lightly coloured texture rather than plain white may help alleviate potential problems with brightness, and may make it less tiring to look at the monitor for a long period of time. This is the method used by the Sibelius software mentioned previously, allowing users to choose a colour and a texture from a set of predefined selections.

For a proper evaluation, the system should probably use a proper rendering system rather than displaying static bitmapped images. Free, open-sourced notational software such as `Rosegarden` may offer a good starting point for developing such a system.

The monitor used in the experiment may have been slightly too small. The use of a monitor over 20" may make a difference, especially for the double pages method. Evaluating page turning methods on displays other than CRTs, such as LCD, may also show up trends that are not apparent here.

Different animation methods, such as animating a page flip, or using a fish-eye view as described in section 2.1, could be evaluated. Different animation methods may change the usability of some of the page turning techniques.

Chapter 6

Conclusion

Most of the components required for a successful page turning system are currently available. Computer hardware can now produce smooth, fast-moving '3D' graphics, and this capability is more than adequate for page turning requirements, even if there are relatively heavy calculations occurring. This means that the major obstacles to implementing a page turning display system are design decisions that must be made on factors that are currently not well understood.

The evaluation study carried out gave some insight into determining musicians' preferences for various methods of page turning. The various methods included techniques for displaying music in horizontal and vertical fashions, with various scrolling methods, as well as methods that overwrite parts of the display with new music. The results suggest that the musicians did not find reading music from the monitor significantly more difficult than reading music from paper, although it appears that attributes such as monitor size, the background colour of displayed music, and the fidelity of the digital image may affect the musicians' perception of usability.

Another interesting result of this study showed that the hardest page turning method to use was the method used by most of the major commercial notation software packages, although it must be remembered that they were not designed for performance during playback.

However, the musicians generally differed in their evaluations for each of the other methods, although the pianists tended to agree with each other, and the non-pianists agreed with each other. Overall, these results seem to suggest that while some methods are more suitable for some instruments, individual preferences vary. However, one advantage of a computerised system is the ability to allow each user to choose a combination of preferred options.

Also discussed were the possible digital representations of music. MIDI, the most commonly used protocol for computer music interaction, is generally not well suited as a data format for page turning purposes, mostly due to its discrete real-time description of music, and related to this, its limitations for giving an accurate notational representation. Other data formats were described and their relative merits with respect to page turning were discussed.

In summary, interactive computing in the arts is now realistic. Hardware considerations are now minor; designing applications that are easy to use as well as powerful and accurate is the next step. Page turning systems that are flexible for each user and can take advantage of all of the benefits that digital representation of music offers are a realistic goal.

Acknowledgements

I would like to thank the members of the Computer Science Computer-Music Research Group for their many suggestions, and ideas. Tim Bell, who supervised me, corrected many of my mistakes and kept offering improvements and insights. Jane McKenzie was very helpful in checking and correcting writing style.

I would like to thank Neill Pickard, the director of the Christchurch Polytechnic School of Jazz, for his permission and assistance regarding the evaluation study carried out, and also the students who gave up their time to participate and provide valuable feedback.

Bibliography

[1] A. Dillon. Reading from papers versus screens: a critical review of the empirical literature. *Ergonomics*, 35(10):1297 – 1326, 1992.

[2] Richard Picking. Reading music from screens vs paper. *Behaviour and Information Technology*, 16(2):72–78, 1997.

[3] Andrew Forsberg, Mark Dieterich, and Robert Zeleznik. The Music Notepad. In *UIST 98*, pages 203 – 210, San Francisco, CA, USA, 1998. ACM.

[4] J. D. Mackinlay, G. G. Robertson, and S. K. Card. The Perspective Wall: Detail and context smoothly integrated. In *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, pages 173 – 179.

[5] W. Hibbard, H. Levkowitz, J. Haswell, P. Rheingans, and F. Schroeder. Interaction in perceptually-based visualization. In Dr Georges Grinstein and Dr Haim Levkowitz, editors, *Perceptual Issues in Visualization*, IFIP Series on Computer Graphics, pages 23–32. Springer, 1995.

[6] John A. Sloboda. *The Musical Mind.* Oxford University Press, 1985.

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

[8] Penny Rheingans and Chris Landreth. Perceptual principles for effective visualizations. In Dr Georges Grinstein and Dr Haim Levkowitz, editors, *Perceptual Issues in Visualization*, IFIP Series on Computer Graphics, pages 59 – 74. Springer, 1995.

[9] Philipp W. Peppler and Steve Stephens. Image generation systems in virtual training platforms. *Communications of the ACM*, 42(9):85 – 88, September 1999.

[10] Joel Pollack. Active matrices brighten liquid-crystal displays. *Optoelectronic World,* supplement to *Laser Focus World*, 34(5):15 – 17, May 1998.

[11] Gary M. Rader. Creating printed music automatically. *Computer*, 29(6):61, June 1996.

[12] C. Graefe, D. Wahila, J. Maguire, and O. Dasna. Muse: Digital stand. In *Conference proceedings on Human factors in computing systems*, page 436, 1996.

[13] Pierfrancesco Bellini, Fabrizio Fioravanti, and Paolo Nesi. Managing music in orchestras. *IEEE Computer*, 32(9):26 – 34, September 1999.

[14] Cindy Grande and Alan Belkin. The development of the notation interchange file format. *Computer Music Journal*, 20(4):33–43, 1997.

[15] Donald Byrd. Music notation software and intelligence. *Computer Music Journal*, 18(1):17–20, 1994.

[16] Linux Documentation Project. *Sound-HOWTO*, 1.20 edition, March 1999. `http://metalab.unc.edu/LDP/`.

[17] Paul Messick. *Maximum MIDI: Music Applications in C++*. Manning, 1997.

[18] Michel Goossens, Sebastian Rahtz, and Frank Mittelbach. *The LaTeX Graphics Companion*. Addison-Wesley, 1997.

[19] Daniel Taupin. MusicTeX: using TeX to write polyphonic or instrumental music. In J. Zlatuška, editor, *EuroTeX'92: Proceedings of the 7th European TeXConference*, pages 257 – 272, Prague, Czechoslovakia, September 14–18 1992. Brno: Masarykova Universita.

[20] David Bainbridge and Tim Bell. An extensible Optical Music Recognition system. In *Nineteenth Australasian Computer Science Conference*, February 1996.

[21] Rodger J. McNab, Lloyd A. Smith, David Bainbridge, and Ian H. Witten. The New Zealand digital library MELody inDEX. *D-Lib Magazine*, May 1997.

[22] B. Gold and L. Rabiner. Parallel processing techniques for estimating pitch periods of speech in the time domain. *Journal of the Acoustical Society of America*, 46:442 – 448, 1969.

[23] M. Goto and Y. Muraoka. A beat tracking system for acoustic signals of music. In *Proceedings of the second ACM international conference on Multimedia '94*, page 365, 1994.

[24] David Bainbridge, Craig G. Nevill-Manning, Ian H. Witten, Lloyd A. Smith, and Rodger J. McNab. Toward a digital library of popular music. In *DL '99: Proceedings of the fourth ACM conference on Digital Libraries*, pages 161 – 169.