The Design of a Pen-Based Musical Input System

Jamie Anstice[1], Tim Bell[1], Andy Cockburn[1], Martin Setchell[2]
[1]Department of Computer Science
[2]School of Music
University of Canterbury
Christchurch, New Zealand
jamie, tim, andy@cosc.canterbury.ac.nz
m.setchell@music.canterbury.ac.nz

Abstract

Computerising the task of music editing can avoid a considerable amount of tedious work for musicians, particularly for tasks such as key transposition, part extraction, and layout. However, the task of getting the music onto the computer can still be time consuming and is usually done with the help of bulky equipment. This paper reports on the design of a pen-based input system that uses easily-learned gestures to facilitate fast input, particularly if the system must be portable. The design is based on observations of musicians writing music by hand, and an analysis of the symbols in samples of music. A preliminary evaluation of the system is presented, and the speed is compared with the alternatives of handwriting, synthesiser keyboard input, and optical music recognition. Evaluations suggest that the gesture-based system could be approximately three times as fast as other methods of music data entry reported in the literature.

1. Introduction

For many writers, a word-processor is inextricably linked with the writing process. The benefits offered by word-processors are numerous, and include the ease of editing (both content and representation), the ability to store and retrieve documents, and the support for sharing documents with others. Many writers would find it hard to carry out their work without computer support.

Computers offer benefits to written music that are analogous to those of word-processors. Computers can also process music in a variety of ways that would be tedious if carried out manually, particularly tasks like key transposition, part extraction, and layout. Several computer systems support musical input, but the process is tedious and time-consuming [12].

In this paper we describe our work on *Presto*, a computer system for pen-based musical input. Our aim is to develop fast and efficient mechanisms for musical input that require only minutes to learn. The system is intended for composers, arrangers, music editors, and performers. It is particularly valuable where portability is an issue. Currently the preferred means of music input is to use a synthesiser keyboard in combination with a computer keyboard and mouse, which could hardly be considered portable. In comparison, the traditional manuscript paper and pen can be used anywhere, including in rehearsals and while on tour. A direct application of a pen-based music input system is for the "Muse" digital music stand [5], a design proposal for an aid for musicians. Muse has an LCD display for music, along with a variety of aids for performance and rehearsal, such as automatic page turning, annotation using a stylus, and co-ordination with other musicians' music displays. Pen-based editing of music would be a natural addition to the system.

Our work is presented as follows. Section 2 reviews current mechanisms for musical input, and describes recent work on pen-based and gestural user interfaces. To ensure that *Presto*'s input language is both fast and learnable, Section 3.1 reports on the relative frequency of musical symbols across a variety of musical scores, and Section 3.2 analyses the music writing techniques used by musicians. Section 4 derives a design proposal for *Presto* from the analysis, and describes a story-board evaluation of the proposal.

2. Background

Hardware devices for human input and output have traditionally been limited to a screen for output, and a keyboard and mouse for input. This limitation is rapidly diminishing, and it is now common for entry-level computers to be able to support many input and output devices, including microphones, loud-speakers, pen-based tablets, musical keyboards, and so on.

In this Section we review current mechanisms for computer input of music, and we examine novel pen-based input mechanisms that we believe have the potential to improve human-input of music.

2.1. Computer Input of Music

When compared to English text, written music is a highly expressive notation. Although the number of symbols in text and music are roughly equivalent, written music is a two-dimensional medium while text is linear — music can express several concurrent streams in one physical area. English text consists of the alpha-numeric characters and punctuation, and words follow each other in a linear stream. Music consists of an alphabet of symbols such as crotchets (quarter notes), quavers (eighth notes), and sharps, but the semantics of the symbols depend on their horizontal *and* vertical locations. Much of the difficulty of musical input to computers is derived from music's two-dimensional semantics.

Several computer systems support musical input. Once the music is held in a computer format it can be processed in a variety of ways. Normally the resultant output of these systems is a graphical representation of the musical score and audio playback of the music. Current techniques for musical input are reviewed below: A thorough review of these techniques is presented by Anstice [1].

Direct Keyboard Entry is achieved through ASCII music representation languages such as DARMS [6] and MusicTex [17]. Users must learn a precise syntax, and any limitation in the language's grammar constrains its flexibility. The cognitive mapping from ASCII notation to musical score is weak.

Direct Manipulation Music Input. The graphical user interfaces provided by most current music systems, such as Lime [11] and Finale [4] combine a keyboard and pointing device with a bit-mapped display to allow direct manipulation of musical symbols. Users create musical objects using key-selections and menus, and the properties of the objects can be adjusted through direct manipulation. Editing musical scores is well supported by these systems (the users directly manipulate the objects of interest), but initial input is frustrated by the abstracted mapping between musical objects and keystrokes.

Musical Keyboard and Direct Manipulation. Most modern music systems allow users to enter music by playing a music keyboard connected to the computer. This allows fluid entry of lines of music. There are two major problems with this style of musical input. First, if the music contains multiple lines of music, either the system must disambiguate between the lines, or (more commonly) the user must play each line independently. Second, recognising the musical rhythm is difficult, particularly when multiple lines of music must be correlated. Recognition errors must be corrected using the direct manipulation editing facilities.

Optical Music Recognition aims to minimise human involvement in music input. The musical score is scanned to a bitmap image, and the computer system attempts to parse the bitmap. Error rates in current systems are relatively high, and although initial input of the musical score is fast, the subsequent human editing of the score is slow and tedious [12]. This technique is also dependent on a printed version of the musical score being available. Bainbridge and Carter [3] provide a comprehensive review of Optical Music Recognition systems.

Soundtrack Analysis is a current area of research which involves computerised transcription from recorded music. Monophonic translation has been achieved [14], but there are substantial difficulties in polyphonic translation [13].

2.2. Pen-Based Input

User demands for portability have resulted in small computers that have little space for a keyboard. Small keyboards are cumbersome and slow to use for text entry [18], so alternatives are an active area of research. Pen-based systems for data-entry are rapidly developing, driven by their popularity with users [8]. They consist of a tablet, which is normally an LCD screen (of any size), and an electronic pen or stylus which is used to write on the tablet.

Pen-based input mechanisms range from those that make no attempt to recognise the users' free-hand marks to those that constrain the user to tapping characters on soft-keyboards [15]. Wang's FreeStyle system [7], for instance, allows users to annotate documents in free-hand using an electronic pen. Most pen-based systems, however, use some form of recognition to convert the user's marks and gestures into computer text which is more legible than hand-written text and consumes less computer memory. The style of marks that the systems can recognise is the primary distinction between pen-based systems.

Cursive hand-writing recognition is the most ambitious style of recognition. There is great variation between individuals' hand-writing, so systems that attempt to recognise cursive hand-writing must be trained to recognise each user's style. Systems such as the Apple Newton [2] can be trained to achieve over 90% accuracy, and although the error rate remains frustrating for extensive text input, users are surprisingly enthusiastic about relatively poor performance [8].

Boxed character recognition. To ease computer recognition, some systems constrain the users to printing discrete characters within specific regions on the screen. These systems display a small box (or series of boxes) into which users print characters. The character entry boxes slide along the screen as the user enters characters. This technique vastly reduces the potential variability at each recognition step (from the set of words to the set of characters), but there is still substantial similarity between certain characters, such as “u” and “v”. Some systems require users to re-enter characters that are incorrectly recognised (with the risk that subsequent recognition may also be incorrect), while others support a “Tap-correction” in which the most likely set of matches are displayed, and the user selects the desired character [9].

UniStrokes. Recognition errors reduce writing speed and disrupt writing flow. To ease these problems, Unistrokes systems [10] use an abstract alphabet in which all characters are denoted by a single pen stroke: each character is recognised when the pen leaves the tablet surface. The Unistrokes abstract alphabet is designed to reduce the similarity between the character symbols, and yet to maintain a partial mental mapping between the symbols and the characters they represent. Unistrokes also allows “Heads up” text entry, escaping the need for the user to look at the tablet surface while entering data. This is achieved by having all unistroke symbols entered into a small static region of the tablet. As soon as the pen leaves the tablet, the symbol that the user wrote is recognised, the recognised character is presented in an output region that shows the linear stream of characters, and the symbol mark is deleted allowing the next symbol to be written on the same region of the tablet.

The primary disadvantage of unistrokes is that users must learn an abstract alphabet. This effort is minimised by the partial mapping from symbols to characters, and Goldberg and Richardson [10] report that the complete alphabet is normally learnt in about 10 minutes. Unistrokes supports an active reference sheet for learning assistance.

Marking Menus. Marking menus [16] are pie menus which pop-up under the pen-tip after a short delay (about half a second). Users then select the desired option by dragging the pen into the appropriate region of the pie menu. With experience, users learn the direction to particular menu options, and then selections can be made by flicking the pen in the appropriate direction without the menu being displayed. To support the large number of options required for text-entry, T-cube [18] uses hierarchical marking menus, with keyboard shortcuts.

2.3. The Potential of Pen-Based Music Input

None of the music input methods reviewed in Section 2.1 allows input at a speed that is comparable with hand-written music on paper. Hewlett and Selfridge-Field [12] describe a study of music input times for Haydn’s Symphony No. 1. Using input through a combination of music and computer keyboards, the initial input took four hours and twenty minutes with subsequent editing taking nine hours and twenty minutes, a total of thirteen hours and forty minutes. Using the Optical Music Recognition system *SightReader* the initial input took thirteen minutes, but the subsequent editing took nine and three-quarter hours, a total of ten hours and seven minutes.

We performed an experiment in which three musicians copied a third of one movement of the same score as the one in Hewlett and Selfridge-Field’s study. Averaging over the three subjects, and extrapolating the time taken, the handwritten input took only 42% of the time that the Optical Music Recognition input method took. Thus handwritten input appears to be at least twice as fast as other methods of input. Currently the disadvantages (with pen and paper) are the lower quality and the lack of flexibility. A pen-based computer system would remove these disadvantages.

Our goal is to use pen-based input devices to radically improve the speed of music input.

3. Analysis of Music Symbols and Music Writing

Goldberg and Richardson [10] identified three primary criteria for their pen-based text input system, Unistrokes: ease of learning, high distinction between the input symbols (to ease recognition), and fast writing speed. These design criteria are equally applicable to musical input as they are for textual input.

Learning can be assisted by a natural mapping between the input symbol and the intended musical symbol. Writing speed can be assisted by allocating the simpler symbols to the most frequently occurring musical characters. This Section analyses the frequency of musical symbols and the “natural” mechanisms that composers use when writing music. The data collected is used to derive a design proposal for pen-based musical input.

3.1. Musical Symbols

The musical symbols in a single page sample from seven compositions were examined. These compositions were selected to provide a diverse range of styles in written music. The composers included Praetorious, Bach, Schumann and Farquhar, and the instrumentation ranged from a quintet to a full orchestral score. Table 1 provides an ordered summary of the 22 most frequently occurring musical symbols in the seven scores.

Table 1 shows that notes account for 51% of the symbols on the page, and are far more frequent that any other individual symbol. Of the 1242 quavers and shorter notes, only

Symbol	Name	Number	%
♪	Quaver & shorter	1242	23%
♩	Crotchet	1069	20%
𝅗𝅥	Minim	445	8%
.	Durational dot	392	7%
⌒	Slur or tie	333	6%
♯	Sharp	249	5%
♫	Single beam	196	4%
♮	Natural	188	4%
♭	Flat	166	3%
𝄽	Crotchet rest	148	3%
Text	Textual symbol	141	3%
𝄻	Bar rest	133	2%
𝄾	Quaver rest	107	2%
𝅮	Tail	103	2%
♬	Double beam	90	2%
>	Accent	74	1%
𝄐	Fermata	65	1%
𝅝	Semibreve	54	1%
	Mixed beam	49	1%
𝄼	Minim rest	6	0.1%
𝄿	Semiquaver rest	5	0.1%
𝄺	Multi-bar rest	4	0.1%

Table 1. Frequency of musical objects in a sample of printed instrumental music.

103 had tails. The others were marked by beams, with an average of 3.7 notes per beam. Annotations to notes, such as dots, slurs and ties, accidentals and beams, are the next most frequent, accounting for 29% of the symbols. Rests account for less than 6% of the symbols.

3.2. Music Writing

'Natural' music writing methods were analysed using video protocol analysis. Ten of the subjects were postgraduate and senior undergraduate music students, and one a University music lecturer. All subjects carried out three music writing tasks which were selected to illuminate any differences between music writing techniques caused by cognitive aspects of the task.

1. Copying — subjects copied three pieces of music differing in degree of complexity, from printed musical scores. Although it is possible to copy the music with minimal cognitive processing (people with no musical experience could be expected to complete the tasks), we were interested to see how the complexity of the music affected the musical input times.

Object	Piece 1	Piece 2	Piece 3
Black Heads	105	68	65
Hollow Heads			7
Stems	105	68	53
Beams	54	36	16
Accidentals	15	11	13
Bar Lines	6	5	7
Clef Signs	3	2	4
Dots	1	11	5
Quaver Tail		1	2
Slurs		15	
Crescendo		1	
Textual Object		11	
Total Objects	289	229	184
% Head, Beam, Stem	91%	75%	82%
Average Time	241 sec	334 sec	352 sec
Max Time	396 sec	502 sec	483 sec
Min Time	127 sec	221 sec	264 sec
Average Time per Object	0.83	1.46	1.9

Table 2. Timing for copying music in task 1.

2. Harmonising — subjects were given a melody line (from Bach's chorale "Freu' dich sehr, O meine Seele") and were asked to write a simple four-part harmonisation. This task is cognitively demanding, and only experienced musicians could be expected to complete it.

3. Writing from memory — subjects were asked to write a short melody from memory ("God save the Queen" was suggested).

A complete analysis of these tasks is presented in [1]. We summarise the observations in two categories, first the time taken to input the data, and second the techniques used to write the musical symbols.

Timing. There was a clear correlation between the complexity of the task and the time to complete it. Even in the copying task, which could be carried out mechanically without cognitive processing, the average time to enter each musical object increased with complexity. The number of musical objects, therefore, has a lesser effect on data-entry time than the density of the objects: Table 2 shows that in the copying tasks the average time per object for the first piece was approximately half that in the third. It also shows that overall the first piece was input much more quickly than the third, despite containing many more objects.

Most of the time in task 2 was spent in thought. Only 24 notes had to be added to the music to create the harmony, but the average time for the task was 315 seconds, producing an average time per note of 13 seconds.

Writing Styles. There was no noticeable variation between the mechanisms that each of the subjects used for the three tasks: their music writing mechanisms were the same for copying, harmonising, and "composing." There were, however, major difference between the individuals' music writing styles. Variations in the writing styles of notes, stems, and beams (the most frequent musical objects) are discussed below—see Anstice [1] for a full review.

All the subjects started writing note groups (such as quavers and semi-quavers) by drawing the note-heads first. There was variation in the techniques used to add stems and beams. Seven of the subjects added the stems to all of the notes (left to right), and added the beam (if necessary) as a final step. Figure 1 illustrates this process. Three of the subjects, however, added the stems to the left-most and right-most notes first, then added the beam, and then added the enclosed stems, as illustrated in Figure 2. One subject drew the stems on the left-most note, extending it into a beam, and finishing the single pen-stroke as the stem of the right-most note. Stems of the enclosed notes were added as a final step. This process is illustrated in Figure 3.

Figure 1. Common method for drawing quaver groups.

Figure 2. Less common method.

Figure 3. Rare method.

4. *Presto*: Design Proposal

Our studies of printed music and how people write music have been used to design a pen-based music data entry system. The system is called *Presto*, because it is primarily intended to facilitate rapid input.

The design of Presto has involved a trade off between learnability, recognition accuracy, and speed. Learnability is affected by the simplicity of the mapping from gestures to musical objects, and the number of gestures that must be learned. The accuracy of recognition depends on the distance between symbols, and how simple they are. The speed of input also depends on how simple the objects are, and will be best if the shortest gestures are used for the most frequent symbols. Since the primary motivation for a musician to move from pen and paper to a computer is usually to save time and avoid repetitious work, we have opted for a trade-off that favours speed. The learnability and recognition accuracy have not suffered unduly because the bulk of musical notation is covered by a relatively small vocabulary (Table 1), and so there are few symbols to learn, and they can be quite different to each other to give accurate recognition.

In our initial design for Presto, only the more common symbols have gestures. In Section 3.1 we observed that notes, dots, accidentals, slurs and ties, beams and rests account for about 86% of the symbols on a page, and so gestures are provided for this subset. The small vocabulary of common symbols is backed up by making everything available through marking menus, so even rare symbols will have a gesture, which is simply the menu selection gesture. This is not mnemonic, but can be learned, particularly if one symbol that is normally rare is common in the piece of music being edited.

Our study of musicians' writing techniques have indicated that it is desirable to allow partially complete notation to be entered even if it is musically incorrect. For example, in the process of drawing the beamed notes shown in Figure 1, noteheads are drawn without stems, and in the second stage there are four crotchets, which could cause the number of beats in a bar to be exceeded.

We have chosen to use shorthand gestures for Presto because there is considerable redundancy in conventional music writing. For example, a crotchet note requires some time to colour in the head, and add a tail of the correct height. These repetitious tasks are best done by the computer, so a simpler gesture is used which is instantly converted under the pen to the correct symbol. The gestures have been designed to be mnemonic to make learning as simple as possible, although this is secondary to the goal of making input efficient.

The gestures to be implemented in the prototype of Presto are shown in Table 3. This shortest gesture, a dot, corresponds to the most common object, a filled notehead with a stem. The pitch of a note is indicated by where the gesture is made on the stave. The direction of a stem can either be chosen automatically by the system, or alternative rules (such as always up) can be specified by the user in advance. The stems can also be changed afterwards. Different note durations are constructed mainly by modifying a crotchet. For example, a horizontal line through a stem halves the duration of a note (users should think of it as a beam), while a

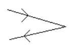
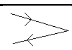
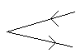
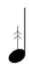
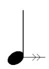
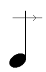
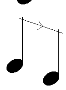
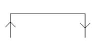
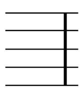
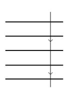
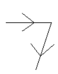
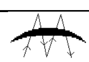
Musical Symbol	Gesture	Effect	
		Filled note	Draw a dot to get a filled note with automatic stem generation
		Filled note with stem	Draw a stem to place a filled note and give stem direction
		Minim	Start on the pitch of the note, draw right, then left
		Doubles value	Start drawing on note or rest
		Halves value	Start drawing on note or rest
		Raise note	Flick pen from note upwards
		Lower note	Flick pen downwards from note
		Add dot	Flick pen left from note or rest
		Add tail	Draw line over one stem
		Add beam	Join stems to add beam
		Add slur or tie	Draw from first note to last note
		Add barline	Draw from top to bottom of staff
Rest		Pop-up rest menu	
		Delete Objects	Scrub up-down-up-down (down-up- down-up also works) over an object to delete it

Table 3. Main gestures in the *Presto* system.

right-left gesture doubles the duration.

The prototype system is primarily intended for evaluating the effectiveness of the gestures for input, so we have not designed a very sophisticated system for editing the music. The main editing operation available is deletion, which is essential, particularly while users are learning the system. This is provided by a vertical "scrubbing" gesture. Other editing, such as moving, transposing pitch, changing stem direction, and adding ornaments, will be done via menus. The objects to be acted on by a menu selection will be selected by drawing a closed curve around them.

The proposed gestures have been evaluated by a storyboard simulation.[1] One of the authors wrote out 28 bars (six parts) of the Haydn Symphony that was used in the previous experiments. The first 13 bars took 84% of the time taken by the subjects copying by hand, and the last 15 bars took only 77% as the writer became accustomed to the notation. We expect that the task will be a little easier in the implementation where the writer will have (fast) visual feedback, and audio feedback is also likely to be valuable. Standard user interface mechanisms for increasing the efficiency of user input (such as copy and paste) will also be available in the prototype.

These figures indicate that although the Presto notation is faster than conventional notation, musicians can write remarkably quickly despite the amount of redundancy in conventional notation. Extrapolating from these results, the Presto system appears to be about three times as fast as the Optical Music Recognition results reported by Hewlett and Selfridge-Field [12]. Techniques for music data entry have improved since Hewlett and Selfridge-Field reported their results, but a three-fold improvement is needed to be faster than the Presto system. Also, Optical Music Recognition systems require high quality input, where as the Presto system can be used with low quality scores, and also for composition and transcription, where the original does not yet exist.

5. Conclusions

The Presto system has been designed with the goal of being a fast pen based input system. The statistics gathered in Section 3.1 show that relatively few objects make up the majority of symbols on a page, which has suggested a system with relatively few gestures, combined with menus to insert the less common symbols. Because there are few gestures they are easy to learn, and recognition is simple.

A storyboard evaluation of the system has confirmed that the gestures are faster than conventional writing, although only by a margin of about 16 to 23%, which is partly a reflection of how quick writing music by hand can be, and is also explained by the overhead of the cognitive processing required for any music writing task. The cognitive load for the Presto system may even be lower than conventional writing once users are experienced with it, because it removes the need to consider details like stem directions and the location of beams. The quality of the output from the digital system will be superior to handwriting at the speeds we performed our observations, and the output is a lot more flexible.

A preliminary evaluation of the Presto system indicates that it is as much as three times as fast as alternative methods for entering musical data onto a computer. Since it is relatively simple to learn, its main drawback is likely to be that it requires pen-based input hardware.

[1] An attempt at a Wizard of Oz experiment failed because the gestures could be completed much faster than the simulated output could be put in place.

References

[1] J Anstice. Design of a pen-based music input system, 1996. Masters Thesis in Preparation.

[2] Apple Computer Inc. *Newton: Apple MessagePad Handbook*. Apple Computer. 1 Infinite Loop, Cupertino, CA 95014–2084, 1995.

[3] David Bainbridge and Nicholas Carter. *Handbook on Optical Character Recognition and Document Image Analysis*, chapter 5. World Scientific Publishing Company (in publication), 1995.

[4] Coda Music Technology, 610 Bury Drive, Eden Prairie, MN 55346-1718. *Finale 3.5*, 1996.

[5] Orya Dasna, Christopher Graefe, Justin Maguire, and Derek Wahlia. muse Digital Music Stand for Symphony Musicians. *Interactions*, 3(3), May – June 1996.

[6] Raymond F Erikson. "The Darms Project": A status report. *Computers and the Humanities*, 9, 1975.

[7] E Francik, SE Rudman, D Cooper, and S Levine. Putting innovation to work: Adoption strategies for multimedia communication systems. *Communications of the ACM*, 34(12):53–63, 1991.

[8] C Frankish, R Hull, and P Morgan. Recognition accuracy and user acceptance of pen interfaces. In *Proceedings of CHI'95 Conference on Human Factors in Computing Systems* Denver, May 7–11, pages 503–509, 1995.

[9] D Goldberg and A Goodisman. Stylus user interfaces for manipulating text. In *Proceedings of the Fourth ACM Symposium on User Interface Software and Technology.*, pages 127–135. ACM Press, 1991.

[10] D Goldberg and C Richardson. Touch-typing with a stylus. In *Proceedings of INTERCHI'93 Conference on Human Factors in Computing Systems* Amsterdam, April 24–29, pages 80–87. Addison-Wesley, 1993.

[11] Lippold Haken and Dorothea Blostein. The Tilia Music Representation Language: Extensibility, Abstraction and Notation Contexts for the Lime Music Editor. *Computer Music Journal*, 17(3), 1993.

[12] WB Hewlett and E Selfridge-Field. How practical is optical music recognition as as input method? In WB Hewlett and E Selfridge-Field, editors, *Computing and Musicology: An International Directory of Applications*, pages 159–166. Thomson-Shore, 1993–1994.

[13] J. A. Moorer. On the Transcription of Musical Sound by Computer. *Computer Music Journal*, 1(4), 1977.

[14] M. Piszczalski and B. A. Galler. Computer Analysis and Transcription of Performed Music. *Computers and the Humanities*, 13:195–206, 1979.

[15] A Sears, D Revis, J Swatski, R Crittenden, and B Shneiderman. Investigating touchscreen typing: The effect of keyboard size on typing speed. *Behaviour and Information Technology*, 12(1):17–22, 1993.

[16] MA Tapia and G Kurtenbach. Some design refinements and principles on the appearance and behaviour of marking menus. In *Proceedings of ACM Conference on User Interface Software and Technology, Pittsburgh, November 14–17.*, pages 189–195, 1995.

[17] Daniel Taupin. *MusicTeX — Using TEX to write polyphonic or instrumental music*, June 1995.

[18] D Venolia and F Neilberg. T-cube: A fast, self-disclosing pen-based alphabet. In *Proceedings of CHI'94 Conference on Human Factors in Computing Systems* Boston, April 24–28, pages 265–270. Addison-Wesley, 1994.