Computers and the Humanities **35:** 95–121, 2001.

The Challenge of Optical Music Recognition

DAVID BAINBRIDGE[1] and TIM BELL[2]
[1]*Department of Computer Science, University of Waikato, Hamilton, New Zealand (E-mail: d.bainbridge@cs.waikato.ac.nz);* [2]*Department of Computer Science, University of Canterbury, Christchurch, New Zealand (E-mail: t.bell@cosc.canterbury.ac.nz)*

Abstract. This article describes the challenges posed by optical music recognition – a topic in computer science that aims to convert scanned pages of music into an on-line format. First, the problem is described; then a generalised framework for software is presented that emphasises key stages that must be solved: staff line identification, musical object location, musical feature classification, and musical semantics. Next, significant research projects in the area are reviewed, showing how each fits the generalised framework. The article concludes by discussing perhaps the most open question in the field: how to compare the accuracy and success of rival systems, highlighting certain steps that help ease the task.

Key words: optical music recognition, musical data acquisition, document image analysis, pattern recognition

1. Introduction

Optical Music Recognition (OMR) – a computer system that can "read" printed music – has much promise: a clarinetist could scan a tune and have it transposed automatically; a soloist could have the computer play an accompaniment for rehearsal; a music editor could make corrections to an old edition using a music notation program; or a publisher could convert a piece to Braille with very little work. OMR has been the focus of international research for over three decades, and while numerous achievements have been made, there are still many challenges to be faced before it reaches its full potential.

OMR addresses the problem of *musical data acquisition*, the key impediment to many computer music applications. It is not, however, the only data entry method for music. The most common method for music data entry in current use combines synthesiser keyboard entry and computer keyboard entry. The musical keyboard is typically used to enter the notes by playing each voice in isolation, either in time with a metronome or using the computer keyboard to enter rhythmic information. The computer keyboard and mouse are then used to correct any mistakes and to add other notation such as lyrics, slurs, and dynamics. Music data entry in this form demands a level of skill from the keyboard player, and adding the remaining notation is time-consuming.

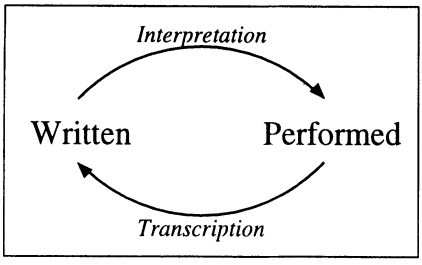

Figure 1. Forms of music representation.

Even if current commercial software could be refined to record information such as dynamics from the synthesiser keyboard (and the performer was able to play these accurately), the computer keyboard stage is still required, since there are many more features to printed music than the way it is finally played – for instance, clefs, key signatures, time signatures, titles, and lyrics. Also, deciding on line breaks, page breaks, even the spacing and grouping of notes within a bar, is a skilled task, and is generally seen as a craft rather than some process that can be defined by a set of rules and automated (Ross, 1970; Read, 1974; Heussenstamm, 1987).

OMR greatly simplifies the task of musical data acquisition; however we are not arguing that OMR should replace musical keyboard entry. In the particular circumstance where the music is already available in printed form, OMR can substantially accelerate the process of musical data acquisition. OMR, therefore, is an attractive supplement. Not only does the technique reduce the chance of human error made in the transcription, but it can also capture much of the "extra" information that the former method requires the user to laboriously add after the notes have been played.

The most likely scenario is one where OMR is used to process the majority of symbols on the page, followed by an editing stage using a standard music editor, where the musical and computer keyboards are used to correct mistakes and omissions. This has the added benefit of greatly reducing the musical keyboard skills required. Of course, OMR is not perfect, and the editing can be a significant part of the time taken for OMR-based data entry; however, OMR systems are steadily improving, and the number of music-based projects utilising OMR steadily increasing (Carter, 1992a). Given the vast body of printed music, OMR could radically reform computer applications in music.

To help clarify the limits in musical data acquisition by computer, let us consider the larger context of music representation. Figure 1 shows the relationship between written and played music. Written music is converted to performed music by *interpretation*, and performed music is converted to written music by *transcription*. Both are non-trivial operations. A performer may study a written work for months before finally presenting their interpretation. The performer will draw upon a range of information, such as knowledge about the composer, and the technical limits

of the instrument, as well as considering the mood and feeling of the work. To transcribe an imagined performance into a written form, a composer reverses this process, deciding what notation would indicate the desired effect. A similar process is undertaken when a recording that has never before existed in written form – such as a jazz improvisation, or a folk tune – is transcribed.

Only limited success has been achieved by computer techniques that imitate these human processes. A computer application that requires the translation from written music to an audio equivalent or *vice versa* will inevitably suffer from a loss of information. Such a situation occurs in synthesiser keyboard data entry, and explains the strong reliance on post editing. Conversely, having a computer play exactly what is written in a score can lead to a mechanical sounding rendition. Better results are achievable if the computer music application stays on one side of the diagram in Figure 1: a MIDI keyboard can be very expressive in sequencing work; and an OMR system is ideal for editing the written page.

The structure of this paper is as follows. First we analyse the graphical properties of music notation that make processing printed music challenging. This includes individual musical features, the complex two-dimensional spatial relationships that exist between them, the superimposing of staff lines with many important musical objects, and the fact that music is sometimes typeset in an ambiguous manner. Next we give an historical account of OMR research, outline a general framework that decomposes the problem of OMR into key stages, and summarise recent trends in the field. The paper concludes by discussing the difficulty of comparing rival OMR systems, highlighting certain steps that help ease the task.

2. Properties of Music

In principle OMR is an extension of Optical Character Recognition (OCR); however, the problems to be faced are substantially different. Examples are provided below that demonstrate the complexities involved. First we consider the problems posed by *individual* musical features, then we broaden our view and look at the relationships that exist *between* musical features.

2.1. Individual musical features

The graphical properties of musical objects are significantly different to those of printed text. This point is illustrated in Figure 2. Text utilises the vertical dimension of the page in a simple way, spacing lines of text in an orderly fashion. Each line itself conforms to a base-line stretching horizontally across the page. Music extends the use of the vertical dimension. Within a line of music, the y-axis is also used to convey pitch; the same shape, therefore, can be translated to different vertical positions. The translation example in Figure 2 shows three crotchet notes that are graphically identical, but drawn at different positions indicating different pitches. The issue of translation is additionally complicated by chords, since individual note

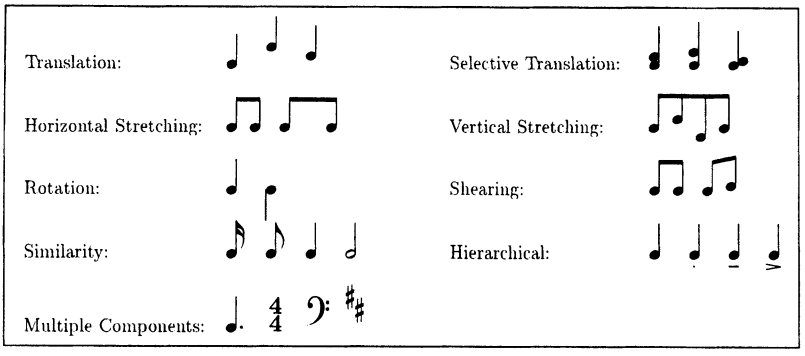

Figure 2. Properties of music not found in printed text.

heads are no longer constrained to be at the end of stems. Examples of this selective translation are also shown in the figure. In principle there are an infinite number of variations in the placement of note heads in a chord, although in practice not all are used. Even so, the number of permutations is so high it is not possible to use a finite set of templates to match these shapes.

Other requirements of music lead to the same musical event having more than one graphical representation. In some circumstances, the difference is a simple alteration such as stretching. This can occur horizontally, as in the beamed note example in Figure 2, caused mainly by the vertical alignment of notes between staves that are played at the same time. Shapes may also be stretched vertically, a good example being a beamed note where the note stems have been either lengthened or shortened to connect with the beam. More complex alterations include rotation, which is affected by where the note is placed on a staff, or whether the staff is carrying two voices at that point; and shearing, a consequence of using vertical displacement to indicate pitch.

Another important difference between text and music is that the graphical appearance of most glyphs in text is quite different, while in music many shapes are graphically similar, and the minor differences convey crucial information. In Figure 2 the example for similarity shows a progression of notes where each successive shape is slightly different to the last, yet each change doubles the duration of the note. A similar situation also occurs with hierarchical symbols in music. By hierarchical we mean optional symbols such as dynamic markings that can occur in addition to the main musical feature, to convey extra information. In Figure 2, the hierarchical example shows the same basic crotchet note altered to become a staccato note, a stressed note, and an accented note, by the addition of an extra component below the note head. This additionally serves to illustrate the frequent use, in music, of multiple components. Also included in Figure 2 are more varied examples of shapes that use multiple components. This contrasts with Western text, where multiple components are relatively rare, and typically consist of a main shape with a supporting smaller mark. Most common is the use of a dot, as in 'i',

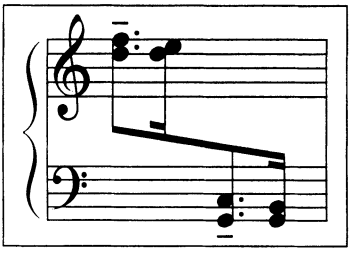

Figure 3. A musical feature can include many variations of representation.

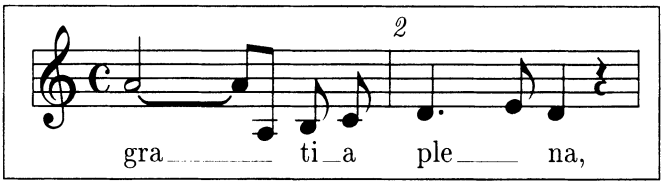

Figure 4. Music relies on many complex two-dimensional spatial relationships.

'j', ':', ';', '?', and '!', with some Roman alphabet based languages augmented by diacritical marks, such as 'é', 'è', and 'ö'. Such exceptions are normally handled by OCR systems using ad hoc methods, whereas OMR must tackle this problem head on.

To compound these difficulties for an OMR system, the transformations do not occur in isolation. One musical shape may vary all these properties simultaneously. Figure 3 shows one example where the construction of the note includes stretching, shearing, rotation, selective translation, and hierarchical symbols.

2.2. THE LAYOUT OF MUSICAL FEATURES

The differences between music notation and text go further than this variance in the atomic characters. Text is predominantly one-dimensional in layout, whereas music makes full use of the two-dimensional space. For example, a clef at the start of a line of music affects subsequent notes on that staff until another clef is encountered; and the syllables to lyrics written beneath a staff are associated with particular notes. These points are illustrated in Figure 4.

Complex two-dimensional relationships are characteristic of Document Image Analysis problems, an active area of research (Baird et al., 1992; Bunke et al., 1997). Figure 5 shows some examples of document types that have been processed successfully by computer. To recognise such documents the general strategy is to first segment the image, then to form the larger objects that are present in the image, from the segmented components. This is accomplished by using some computer encoded representation of knowledge about the notation that a person who is skilled in the notation takes for granted once learnt. Document Image Analysis forms a rich source of techniques for OMR.

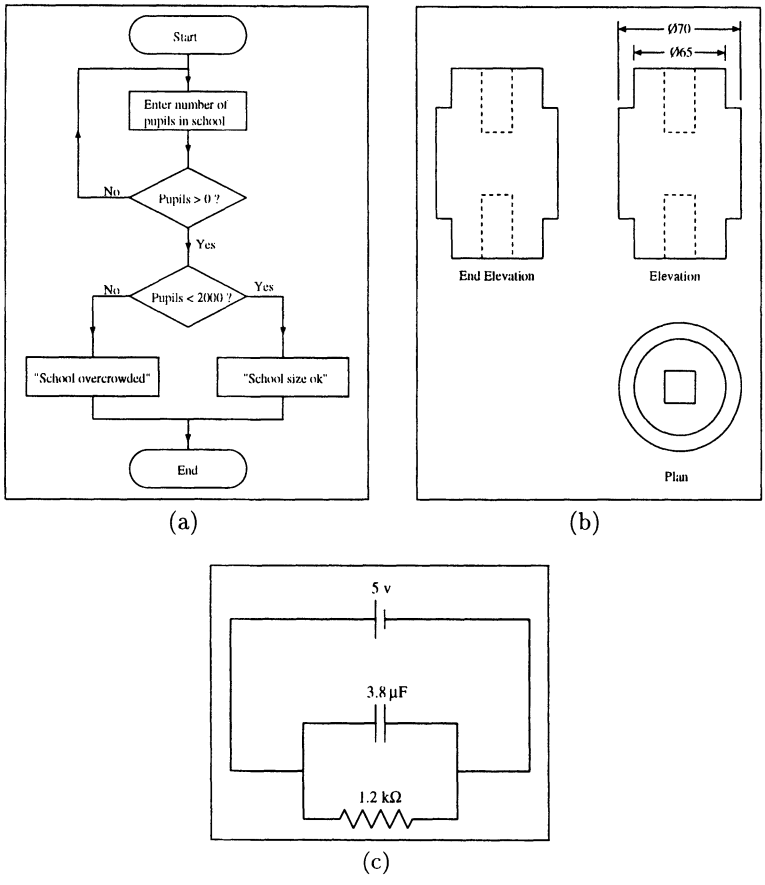

Figure 5. Examples of computer processed documents (a) flowchart (b) technical drawing (c) circuit diagram.

2.2.1. *Ambiguity*

A consequence of music relying on two-dimensional relationships is the issue of order. When considering a single voice of music, there is generally a unique and understood order in which the music must be read, otherwise the music would be ambiguous. However, it is sometimes *written* in an ambiguous manner. In such situations, the breach of convention is normally a sacrifice intended to simplify layout, and an experienced musician can still determine the meaning of the music and correctly pick the order – typically because the alternative interpretations would be nonsense.

An example of ambiguity is shown in Figure 6. After study, it becomes apparent that the notation shows both the left and right hands starting on the second staff, alternatively playing notes in an ascending arpeggio, with the notation crossing over to the first staff part way through. However, because there are no rests

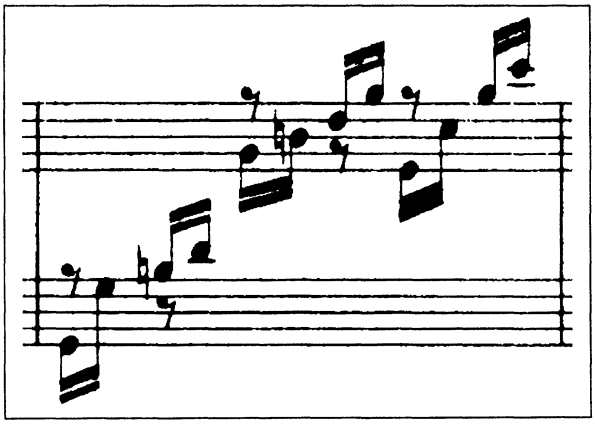

Figure 6. An example of music notation that is written ambiguously to improve layout; the bar is played as an arpeggio using alternate left and right hands, despite being drawn on staves which normally indicate separate left and right hands.

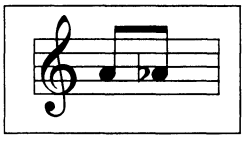

Figure 7. A seemingly simple piece of music relies on two-dimensional layouts and relationships that are complex and intricate for a computer to process.

prior to the inscribed notes in the top staff another, and initially less convoluted, interpretation is that the notes to both staves should start at the same time.

Just because a correct order exists, this does not mean it will be easy for a computer to determine. For example, in Figure 7, consider which of the two musical shapes should be processed first: the beamed notes or the accidental? Logically the accidental and second note form a subgroup, but the recognition system will more naturally link the two notes as a group, since they are physically linked, and then treat the accidental in isolation.

2.3. SUPERIMPOSED MUSICAL FEATURES

A unique aspect of music that does not occur in the Document Image Analysis examples shown in Figure 5, is the *superimposing* of shapes. In music notation, notes are intentionally superimposed on staves to convey pitch, and slurs, ties and crescendo/diminuendo "hairpins" sometimes cross over bar lines, making it difficult for computer algorithms to distinguish musical features. Similar problems of occlusion occur in other fields, such as computer automated cell counting in Biology (Russ, 1992) where two cells on the microscope slide may overlap, partially obscuring each other. Researchers report that superimposition is a severe problem in image processing, and one that proves difficult to compensate for. A

great deal of the complexity in OMR systems is due to superimposition, especially symbols printed on the staff.

2.4. Music notation includes text

So far, the discussion has concentrated on the unique musical shapes found on the staff. Music, though, includes many other shapes off the staff, such as bar numbers, dynamics, fingering information, lyrics, and other text and text-like characters. Thus, OMR is really a *superset* of OCR.

To fully process the text in music, it is insufficient to convert each letter to its ASCII form, and then group words and lines together. The position of the text on the page is important, and must be used to associate its meaning with the correct part of the music. To complicate things further, it is conventional for dynamic markings to be written in Italian and abbreviated (***f***, ***mf***, ***p***, and so on), and foreign languages are common, even if the music is intended for English speaking musicians. This prevents the use of conventional OCR lexicons, but there is potential to use specialised word lists.

3. History

The first published OMR work was carried out at Massachusetts Institute of Technology in the mid 1960s by Pruslin (1966; Kassler, 1972). Pruslin's system recognised a subset of CMN, primarily musical notes. It lacked features such as clefs, time signatures, and rests, but permitted complex stem-sharing chords. Prerau (1970, 1971; Kassler, 1972; Prerau, 1975) originally intended to extend Pruslin's work, but unfortunately he discovered that Pruslin's staff-removal (used to tackle the issue of superimposing) distorted most musical features other than note clusters and beams. He therefore had to develop an alternative approach.

Prerau observed that, unlike text characters, a musical feature could usually be identified by the dimensions of its bounding box. Because the same musical feature could appear larger or smaller in a different piece of music, the dimensions of the bounding boxes were normalised with respect to the height of the staff. A survey of different styles of music was undertaken and a database built. The heuristic classification algorithm developed used the database to quickly reduce the number of possible musical features an unknown shape might be: often the database would return a single match, and thus the object was classified. In situations where the bounding boxes for musical features in the database were not unique, additional checks were designed to distinguish the shapes.

In contrast to Pruslin's work, Prerau's system recognised clefs, rests, certain time signatures, and accidentals. However, it could not handle chords. Many systems following Prerau's work have made significant use of his observation about musical features having distinctive bounding boxes, leading to efficient musical feature classification algorithms.

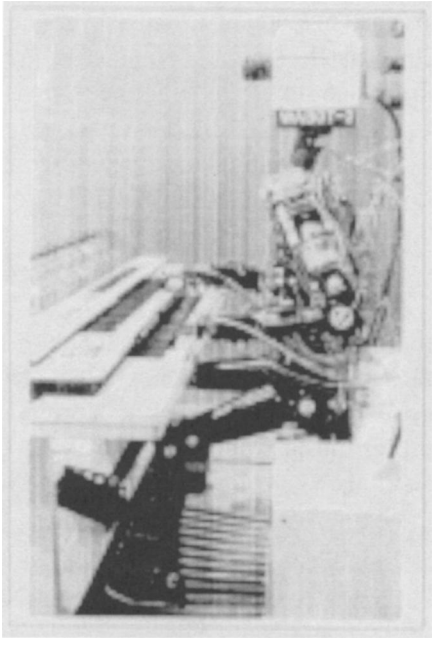

Figure 8. The Wabot-2 (reproduced from the Bulletin of Science and Engineering Research Laboratory, Waseda University (Matsushima et al., 1985)).

Early work was limited by technology, and merely acquiring image data was a significant part of these projects. Prerau, for example, used a flying spot scanner to achieve a scan resolution of approximately 225 dots per inch (dpi), whereas today 300 dpi, 400 dpi, and 600 dpi scanners are common and more reliable. It was not until the early 1980s that the next significant development occurred. In a major undertaking by Waseda University (Japan), a robotic organist (Figure 8) was built (Matsushima et al., 1985; Roads, 1986). Demonstrated at the 1985 International Exposition, the robot would play the organ using its mechanical hands and feet. The robot could play an unknown piece of sheet music placed in front of its electronic eye (a video camera), or accompany someone singing a tune it already knew. Visitors to the exposition were invited to converse with the robot, and select a piece to sing. The robot would then keep in time with the singer, even if they varied their speed, and would automatically transpose key if the person accidentally changed the key they were singing in.

The real-time processing necessary was achieved by using dedicated hardware: 17 16-bit computers and 50 8-bit computers with fibre optic data-links. The project took $3\frac{1}{2}$ years, and involved four professors and some 50 students. Had commercial equipment and services not been donated, it is estimated that the project would have cost US$2,000,000 (Roads, 1986).

The OMR component of the Wabot-2 was the first system to recognise a comprehensive set of CMN, including chords. The real-time processing requirement led to a 'brute force' approach for many OMR problems. A key component

of the system was a frame-buffer that was used to store a normalised version of the page. Significant use was made of template matching, rendered practical by being realised in hardware, which processed the image stored in the frame-buffer.

More recently the cost of hardware suitable for OMR has come within the budget of many research centres. A reasonable configuration would consist of a flatbed scanner with 200-400 dpi resolution, connected to a workstation with appropriate memory and disk space.[1] The fact that no hardware item is peculiar to OMR has been beneficial to the growth of the subject. Since 1985 there have been over 20 research projects.

The increased interest has substantially helped further understanding of the problems in accomplishing OMR, and the first commercial packages that recognise the core of CMN are beginning to emerge. Also, some OMR research projects have looked into other formats such as medieval music (McGee et al., 1991; Carter, 1992b), and handwritten formats (Roach et al., 1988; Bulis et al., 1992; Wolman et al., 1992; Yadid et al., 1992; Anstice, 1996). Work with both formats is, understandably, harder since the clarity and regularity of the graphics is considerably reduced. For example, in handwritten music a note head is more likely to be physically separated from its note stem, and will more typically consist of a short stroke rather than the neat oval found in engraved scores. Results for these formats are promising, but work on these topics lags behind the success of printed CMN in much the same way that handwritten OCR research follows in the wake of printed OCR.

4. An Overview of OMR

Optical music recognition is a complex problem, rendered manageable through decomposition. Figure 9 shows how OMR can be simplified into four smaller tasks: *staff line identification*, *musical object location*, *musical feature classification*, and *musical semantics*. Musical feature classification is in turn decomposed into *primitive detection* and *primitive assembly*, and optional *image enhancement* is possible after staff line identification and musical object location.

The framework is not the only decomposition possible, although there is good reason to base work on this model. All of the existing OMR work fits the model (Bainbridge et al., 1997b), although this fact is often superficially obscured by the use of different terminology, and choices in system structuring that combine or overlap particular stages in the model. In the future this framework may not prove to be the best solution for OMR, since developments in programming paradigms may prompt a different structuring; however, it is still useful since the model unifies the different strategies that exist today, allowing us to compare and contrast different OMR systems. Using the model, we can identify the strengths or weaknesses of a given OMR system, and discuss the merits of competing algorithms.

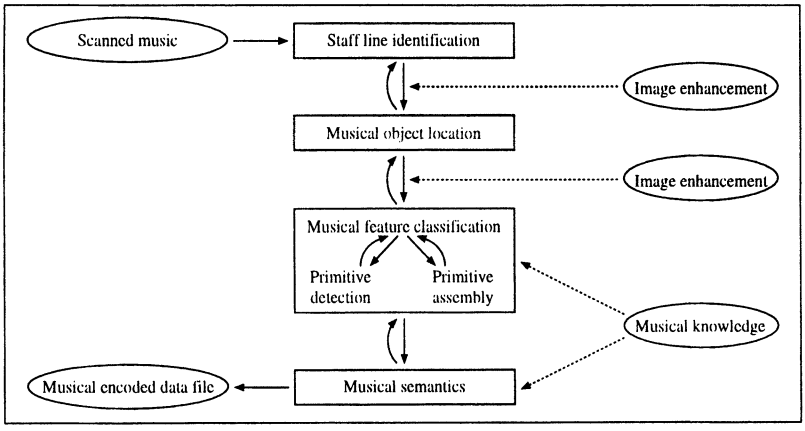

Figure 9. A general framework for OMR.

Below, the four principal stages of the model are described in turn. This leads on to a discussion of recent trends, and some concluding remarks. For a fuller and more historic review of OMR work see Blostein and Baird (1992), and Selfridge-Field (1994).

4.1. Staff line identification

Before an OMR system can start recognising the shapes contained in an image, it must establish the size of the music notation being used. Staff lines are a reliable feature contained within a page, from which the staff height can be deduced, and consequently the size of the music notation. All reported OMR systems detect staff lines as the initial stage.

Since staff lines in a scanned image are not guaranteed to be level (or even straight!) detecting staff lines is a trickier task than might first be imagined. To deal with these complications, staff line detection algorithms need to make some assumptions about the image. Two common assumptions are that staff lines cover a large proportion of the image, and that (excluding noise) their thickness will be one of the smallest found.

The most widely used method for detecting staff lines is based on *horizontal projections* (Blostein et al., 1992). A horizontal projection maps a two-dimensional bitmap into a one-dimensional histogram by recording the number of black pixels in each row of the bitmap in the corresponding entry in the histogram. Under such a projection, staff lines appear as distinct peaks in the histogram that can be detected easily. Numerous variations exist, but the basic technique remains the same. Figure 10 shows a representative excerpt of music that will be used to illustrate the various stages of the general OMR framework. Its horizontal projection is shown in Figure 11.

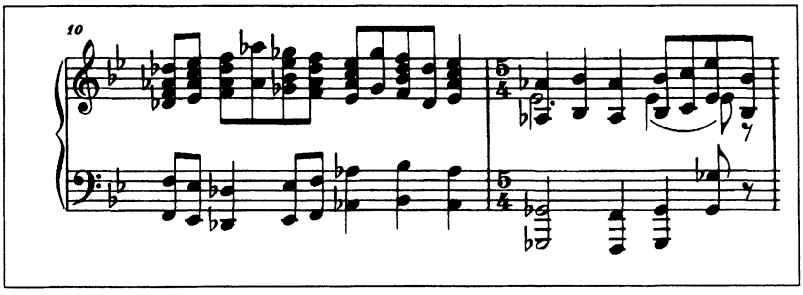

Figure 10. A representative excerpt of music, taken from Mussorgsky's "Pictures at an exhibition," published by Schott of Vienna, Austria (Page 2, Line 4).

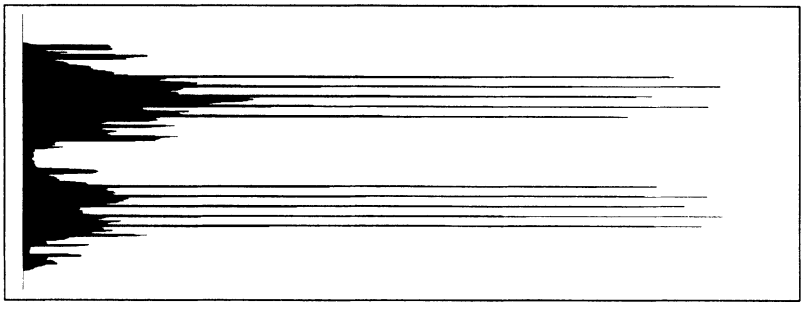

Figure 11. The horizontal projection of the example excerpt when scanned carefully.

The music in Figure 10 was scanned carefully so the staff lines were as level as possible. It would be naïve to assume all music is scanned so carefully. The horizontal projection of a more typical scan of the example excerpt (staff lines at +0.4°) is shown in Figure 12. Notice how the distinct peaks in a projection deteriorate.

Instead of calculating the projection for the whole page, existing systems usually apply the basic algorithm to smaller regions, selected on the left-hand side and right-hand side of the image. This will locate short sections of staff line that are less affected by the angle at which the page was scanned. The segments can then be logically connected one-for-one, where the average gradient of these lines determines the angle of skew. Staff identification is achieved by grouping the individually detected staff lines into sets of five.

An alternative strategy for identifying staff lines is to use vertical scan lines (Carter, 1989; Glass, 1989; Kato et al., 1990; Reed, 1995). There is more scope for variation here. In broad terms, these algorithms use the vertical scan lines to cut through the image, gathering data on black (or white) cross-sections. The data is then searched for a dominant feature, which corresponds to the staff line thickness (or the distance between staff lines). In addition, consistency checks, such as the identified staff lines occur at regular gaps, are often included to confirm the selected staff lines.

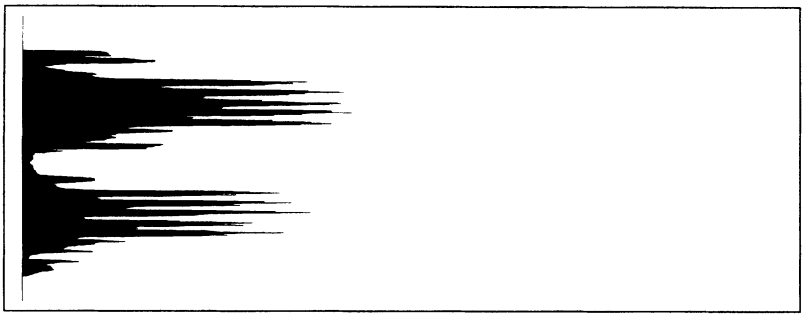

Figure 12. The horizontal projection of the example excerpt from a more typical scan.

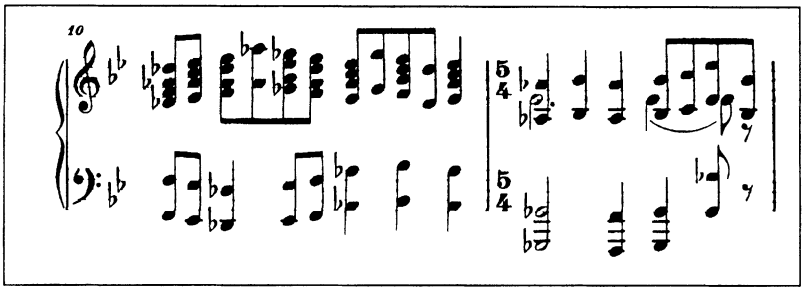

Figure 13. The example excerpt with staff lines removed.

4.2. Object Location

Once the staff lines have been identified, the individual musical objects must be located. This can be accomplished by either *ignoring* or *removing* the staff lines.

Because it is known where the staff lines are, the computer can search between the staff lines for musical objects, effectively ignoring the staff lines. When using this approach, the key task is deciding when one object stops, and the next one starts. Ignoring staff lines in conjunction with musical feature classification simplifies this decision. A search for objects between the staff lines is started, and when an object is encountered, the musical feature classification stage is invoked. The result of the match determines the extent of the symbol encountered, and directs the search for further patterns. A drawback to the approach is that as the set of music notation processed becomes more complex, so too does the control of the matching algorithm.

A more popular method is to remove the staff lines, since this is not dependent on the complexity of the music notation. The *de facto* algorithm for this task was first described by Clarke et al. (1988). In this algorithm, a staff line is removed piecemeal. The algorithm follows along the line, replacing it with white pixels unless there is evidence of activity on either side of the line. The check for the existence of a musical object generally searches a region no more than two pixels away from the top or bottom of that part of the staff line. Figure 13 shows a typical result of applying the algorithm.

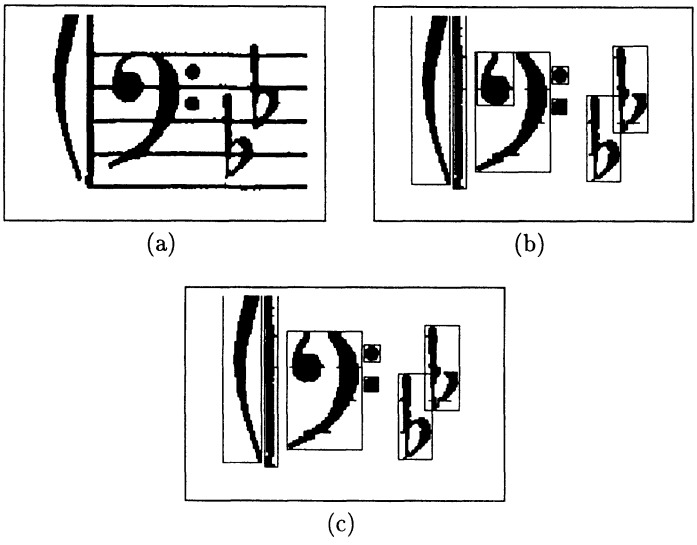

Figure 14. Locating musical objects (a) the original image (b) isolated shapes broken up by staff removal (c) broken shapes reconstructed.

Despite care being taken to keep objects whole, the algorithm does cause some fragmentation, especially to objects that blend tangentially with staff lines, such as a bass clef or a minim note head. Examples of this can be seen in Figure 14b, where the bass clef is fragmented in two and the second flat is broken at the top.

A simplistic solution to this problem is to join two shapes if they are situated close together, and both have an edge of their bounding box that is flush with the same staff line. The algorithm is not ideal because in certain situations it will incorrectly join objects that should remain separate, and at other times, objects that should be joined are left separate. The result of applying this imperfect algorithm to Figure 14b is shown in Figure 14c. In this small example the algorithm correctly joins up the bass clef curl without introducing any erroneous conglomerate shapes.

By assuming that fragmentation only occurs when staff lines are removed, this simple algorithm makes no allowance for objects that become broken due to other causes, such as scanning a work that includes faint pen-work and blemishes in the original. Thus the algorithm described is not only imperfect, but insufficient. Dealing with fragmented objects is an important problem that a realistic OMR system must solve and is the focus of recent research (Coüasnon et al., 1994; Ng, 1995; Ng et al., 1995, 1996; Bainbridge, 1997).

As well as fragmented objects, a scanned piece may also include two or more musical objects that are joined. A distinction is made here between two classes of joined objects: objects that legitimately become joined, such as slurs and hairpin crescendos crossing bar lines; and objects that should not have been touching, such

as an accidental or the end of a slur being drawn too close to a note head. To aid clarity, the first class of joined objects will be called *superimposed*, whilst the term *touching* will refer to the latter scenario.

Strictly speaking, touching objects should never arise. Texts on the topic of music notation (Ross, 1970; Read, 1974; Heussenstamm, 1987) stress the importance of a clear, distinct layout when positioning objects, and having objects that touch reflects poor craft-work. However, reality is less than ideal, and touching objects frequently occur in printed music. Dealing with touching objects, therefore, is an important problem that a realistic OMR system must solve.

Once the musical features have been isolated (and some attempt has been made to separate touching objects and to correct any fragmentation) it is a simple step to locate individual objects. The image is processed left to right, top to bottom, systematically considering every pixel. If a pixel is found to be black, then it is used as a seed for a flood fill algorithm (Foley et al., 1990) that also removes the shape being filled, as the algorithm proceeds. By keeping a list of all the flood-filled shapes, individual musical shapes are correctly located.

4.3. Musical feature classification

With the musical objects located, one might believe the problem of OMR has been reduced to that of OCR. Unfortunately, as Figure 2 illustrated, this is not the case. The shapes in music are inherently more complex than those in text. Rather than attempting to classify such intricate shapes as a whole, a trend in successful OMR systems is to work at a sub-symbol level. First, the simpler geometric *primitive* shapes that make up symbols (such as note heads, stems, beams and accidentals) are detected using pattern recognition techniques such as template matching (Witten et al., 1994), the Hough transform (Boyle et al., 1988), and projections (Pavlidis, 1982); then the primitives are assembled into musical features, guided by musical knowledge. Such a decomposition is often found at the heart of Document Image Analysis systems (Baird et al., 1992).

Notice how this strategy naturally deals with many of the "complications" that music has compared with text. For example, a musical feature with multiple, disjoint components (such as the key signature for A) now needs no special treatment, since it is merely an object made up of three primitive shapes (sharps in this case) where the primitive shapes are close but do not actually touch; and objects that are similar (such as a crotchet and a quaver note), are readily distinguished by the number of individual primitives detected in that region (in this case both have a stem and a filled-in note head, but only the quaver has a tail attached to the stem).

Once primitives have been found the challenge is to construct musical features from them. Techniques from the area of Document Image Analysis that have been successfully adapted and applied to OMR to solve the musical feature classification stage, are: Definite Clause Grammar (Coüasnon et al., 1994, 1995; Bainbridge,

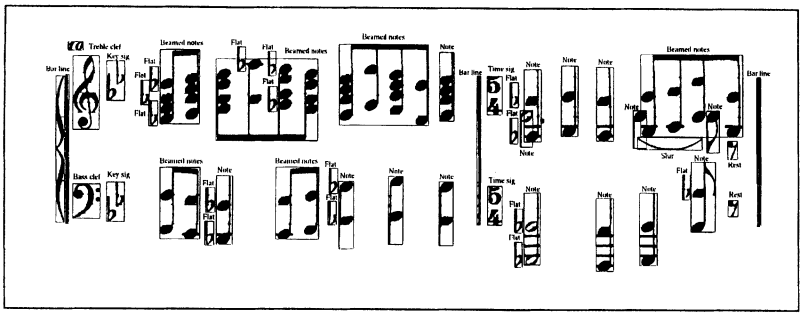

Figure 15. The example excerpt of music with classified musical features.

1997), Graph Grammar (Fahmy et al., 1991, 1992; Reed, 1995), Musical Knowledge with Constraints (Kato et al., 1990), and a Decision Tree Classifier (Baumann et al., 1992). All achieve musical feature classification by decomposing symbols into primitive shapes.

The process of musical feature classification is illustrated in Figure 15, where a labelled bounding box has been drawn around each classified musical feature. At this point, an OMR system has determined what all the shapes are graphically. Consequently a certain level of music recognition has been accomplished. From this point, it is possible to generate a "clean" version of the page, where the primitive shapes (for example note heads, and stems) are replaced by perfectly formed geometrical shapes (for example ellipses, and lines), and if these primitive objects are correctly grouped and made available in a standard drawing editor, then the musical features can be manipulated to correct any imperfections. This, in essence, describes the editorial enhancement application cited at the start of the article.

However, at this stage the music has not been completely recognised. For example, only diatonic transposition is possible since accidentals are not associated with notes, and even the clef that governs a note has not been established. Audio playback is not possible for the same reason. Although the OMR system has graphically recognised the music, it does not yet musically "understand" the piece.

4.4. Musical Semantics

The final stage in an OMR system is to extract the musical semantics from the graphically recognised shapes, and store it in a musical data structure. Essentially, this involves interpreting the spatial relationships between the detected primitives found in the image. For example, a treble clef affects the register of notes on its staff – a note head drawn on the bottom line of the staff is played as the note E, and so on. More subtle interplay occurs when the same graphical shape can mean different things in different situations. For instance, to determine if an object between two notes is a slur or a tie, the pitch of the two notes must be considered; and a dot

```
Staff System 1:
Staff 1:
{4Db 4F 4Ab 5Db}(0.5),{4Eb 4Ab 5C 5Eb}(0.5),
{4F 4Ab 5Db 5F}(0.5),{4Ab 5Ab}(0.5),
{4Gb 4Bb 5Eb 5Gb}(0.5),{4F 4Ab 5Db 5F}(0.5),
{4Eb 4Ab 5C 5Eb}(0.5), .....

Staff 2:
{2F 3F}(0.5),{2Eb 3Eb}(0.5),{3Db 2Db}(1.0),
{2Eb 3Eb}(0.5),{2F 3F}(0.5),
{2Ab 3Ab}(1.0),{2Bb 3Bb}(1.0),{2Ab 3Ab}(1.0) | ....
```

Figure 16. Musical semantics of the example excerpt.

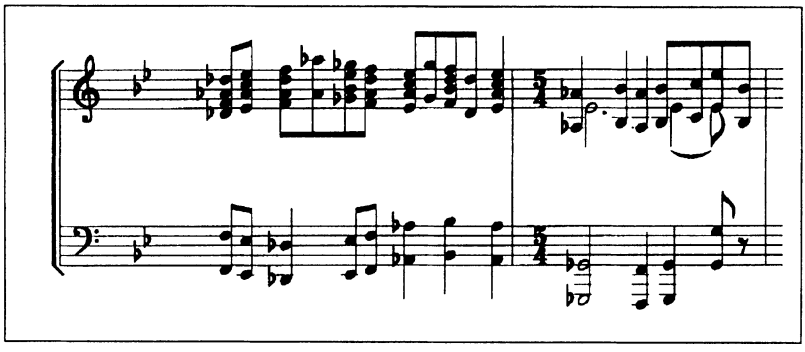

Figure 17. The example excerpt reconstructed in a musical editor.

above a note head changes its performance character principally by reducing its duration whereas a dot *to the right of* a note head increases the note's duration.

How this stage is accomplished is often omitted in the literature, since the operations are specific to a particular output format. In broad terms, this phase of an OMR system is characterised by (possibly) multiple passes over a graph-like data structure, creating links, deleting links, or modifying the attributes stored at nodes due to the effect of one musical feature – say a key signature – on other musical features, such as notes.

The musical semantics of the example excerpt is shown in Figure 16 using an internal text format. Unfortunately a standard format that represents an interpreted page of music is yet to be established. For now, the best that can be achieved is a flexible musical data-structure that is easily traversed to generate specific musical file formats. Figure 17 shows the example excerpt reconstructed from the Tilia file format using the Lime music editor (Haken et al., 1993).

Recent developments for a standard musical file format are promising. A consortium of interested parties (music software firms, academic researchers, and music publishers) formed a working group to address the issue, resulting in a

specification for a format known as NIFF[2] (Notation Information File Format). Also, an ANSI committee has produced SMDL[3] (Standard Music Description Language) which is derived from the ISO standard HyTime (Hypermedia/Time-based Structuring Language). Only time will tell if these standards will be adopted by the computer music community.

4.5. The end point of OMR

At the start of this article, applications for OMR such as accompaniment, editing and transposition were cited. These examples illustrate that the level of musical detail required from an OMR system is dependent on its intended use.

For basic audio playback, it is sufficient to recognise only the pitches and durations of musical features. This is all that is required for many simple applications and is relatively easy to achieve. Items such as the title and fingering information are superfluous, and depending on how crude the playback can be, dynamic markings may also be ignored. Alternatively, an OMR system for editorial enhancement must recognise all the graphical shapes on the page. In this situation, typography is the primary goal, hence it is not strictly necessary for the OMR system to understand the musical significance of the graphical shapes, only what the primitive shapes are, and where they are located.

These two examples represent extreme situations. Other applications require a mixture of graphical recognition and musical understanding. The ultimate goal of OMR work is a system capable of graphically recognising *and* musically understanding all the information present in a page of music, and consequently requires both the musical feature classification and the musical semantics stages to be implemented.

5. Recent Trends

In this section we discuss how recent systems fit into the framework described in Section 4. Although some methods only implement a subset of CMN, the range of symbols recognised is increasing and is now reasonably comprehensive, so the framework does not seem to impose undue limitations on what can be recognised.

Work by Carter (1989; Carter et al., 1990, 1992; Carter, 1992a, 1993, 1994a, 1994b) is based upon a transformed Line Adjacency Graph (LAG) (Pavlidis, 1982). A transformed LAG of a binary image is formed by first generating a vertical run-length encoding of the image (Figure 18a), then comparing the run-length encodings of adjacent columns. Runs of black that are of similar length and position in adjacent columns are continually amalgamated to form a single node in the graph, until no suitable segment exists (Figure 18b). To detect staff lines and primitive shapes, the graph is traversed, searching for particular configurations of nodes. The system is tolerant of skew up to 10 degrees.

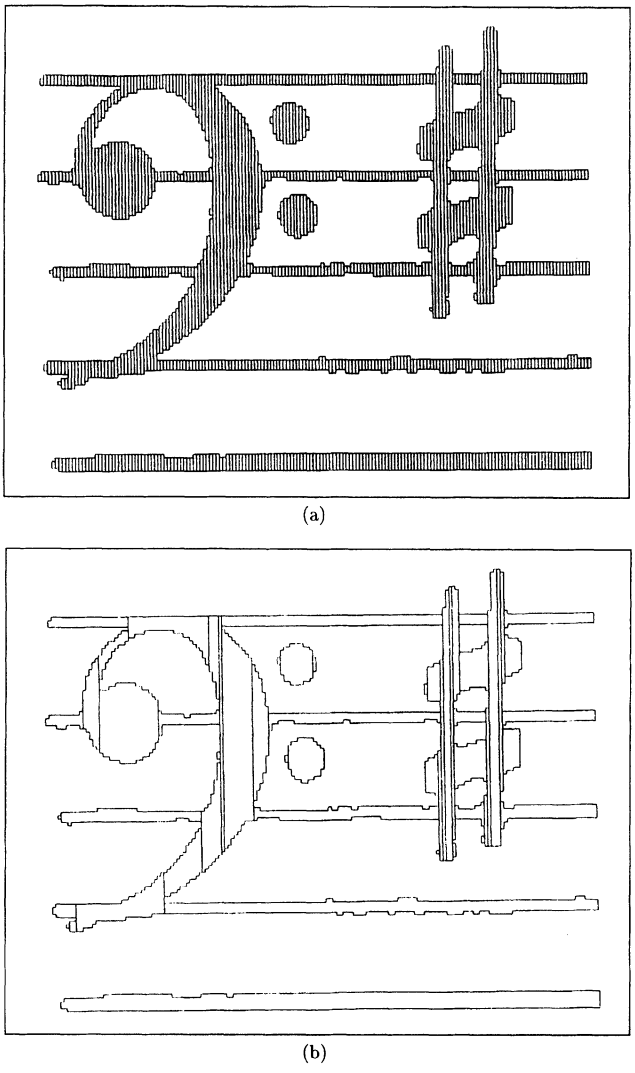

Figure 18. Generating a transformed line adjacency graph (a) the segments formed by vertical run-length encoding (b) the segments formed by amalgamating similar vertical run-length encodings that are adjacent. (Reproduced from Carter's Ph.D. thesis (Carter, 1989)).

Developed by Fujinaga, an interactive OMR system aimed at the first three stages of the general framework has steadily evolved at McGill University since 1988 (Alphonce et al. 1988; Fujinaga, 1988; Fujinaga et al., 1989a, b, c, 1991a, b, 1992; Fujinaga, 1992, 1997). The system integrates a recogniser, editor, and learner. The recogniser utilises projection methods (Fujinaga, 1988) and a k-nearest neighbour classification algorithm (Cover et al., 1967) to detect musical features. The result is viewed using a music notation editor, where any corrections made by the user are passed on to the learning module. Here, the weights of the feature vectors used for classification are recomputed, aided by a genetic algorithm, to

give an improved match next time. Even with the genetic algorithm, this is an expensive operation, and consequently the system is designed to perform the learning operation during processor idle-time.

With Fujinaga's design, the graphical classification of shapes is naturally extensible. When a musical feature is encountered for the first time it will be misclassified. This is corrected by the operator during the interactive phase. As more examples of the new shape are encountered and corrected, the system will gradually learn how to distinguish the shape automatically.

Kato and Inokuchi use a computational model from Artificial Intelligence: a top-down approach that uses musical knowledge with constraints to recognise piano music (Kato et al., 1990). The system comprises five ordered layers: image, primitive, symbol, meaning, and goal; where the first four layers correspond to staff line identification, primitive detection, primitive assembly and musical semantics in the general framework, and the final layer (the goal) 'holds' the system together. Each layer can generate hypotheses that subsequent layers must either verify or refute, using a working memory model to handle communication between the layers. Based on this approach, Baumann and Dengel devised a top-down strategy that applies a decision tree classifier, and encoded musical rules to recognise piano music (Baumann et al., 1992).

Coüasnon et al. (1994, 1995; Coüasnon, 1997) devised an OMR system based on a grammar. The primary role of the grammar is to specify the taxonomy of musical features (primitive assembly), but it also controls the joining and segmentation of shapes as well as defining a sequential order to process objects on the staves – effectively specifying how to "read" music. The grammar forms a flexible component, making it easy to alter what constitutes valid notation.

Fahmy and Blostein (1991, 1992) concentrate on the primitive assembly and musical semantics stages of an OMR system, using an attributed programmed graph grammar (Bunke, 1982) as the basis of their work. Traditionally a graph grammar translates an input graph into an output graph that reflects a higher level of understanding of the document. The work by Fahmy and Blostein differs slightly in that the starting point is a collection of isolated nodes that represent the primitive shapes detected in the image. The graph grammar processes these nodes, forming links that represent interactions between primitives, consequently generating a connected graph as output.

The principal aim of the work carried out by Bainbridge (1994a, b, c, 1995a, b; Bainbridge et al., 1996; Bainbridge, 1997; Bainbridge et al., 1997a, 1998) is a fully extensible OMR system. This is achieved by combining a specially designed programming language for primitive detection, a user-configurable knowledge-base for primitive assembly, and an object oriented interface for musical semantics. In doing so, the design is capable of processing not only an extensible set of shapes within one notation, but a variety of notations, such as CMN, plainsong notation, and tablature.

The specially designed programming language, called PRIMELA,[4] eliminates the need for repetitive descriptions, and consequently the code is concise – comparing C++ to PRIMELA, it was found that, on average, 928 lines of C++ code were needed per primitive type processed, but only 56 lines of PRIMELA code. Grammar rules in the knowledge-base provide a flexible medium in which the valid taxonomy of musical features can be expressed. Finally, the object oriented interface provides a mechanism that can be tailored to encode the semantics of a specific musical notation.

6. Comparing OMR Systems

Perhaps the hardest task in OMR is comparing competing systems. Different systems place different restrictions on the type of music that can be processed, with incompatible sets of musical features being the most common difference. Even if two OMR systems process comparable collections of music notation, the numerous end-points to OMR (audio playback, graphical reconstruction, and others that lie in-between) weaken many comparisons. And should two systems have compatible end-points, further complications arise due to the multitude of file formats available for the final representation.

To make comparable measurements of OMR performance we need a corpus of scanned music. Attempts to compile one, however, are plagued by copyright issues. The corpus built by the principal author (Bainbridge, 1997) categorises images using typographical criteria, since this was found to be of more relevance than musical styles such as folk music and jazz. Categories include: orchestrated score, miniature score, accompaniment, monophonic, percussion, and computer typeset, where an image can belong to more than one category. In the future it is hoped that the corpus can be released on CD-ROM.

Another challenge is to determine how the accuracy of an OMR system is to be calculated. There are different types of errors, most of which result in either omission, commission, or substitution of symbols. Which is more serious? Some published results quote percentages based on the number of correctly processed shapes out of all possible shapes; others quote percentages based on the number of shapes in the image the system was capable of recognising. And what constitutes a shape? Do the three sharps in the A major key signature count as one object or three? Reed demonstrates the problem with an incisive example in which he presents, from the same set of data, two different calculations that yield accuracy levels of 11% and 95% respectively (Reed, 1995).

We believe the difficulty results from trying to summarise the accuracy of an OMR system as one number. Given the decomposition of OMR into smaller more manageable pieces, it makes sense to quote accuracies rates in terms of these key stages: staff line identification, primitive detection, primitive assembly, and musical semantics.

For the first three items, accuracy calculations are straightforward – although care still needs to be taken in *how* these values are calculated. For the final item, musical semantics, we suggest counting edit operations using a musical editor. This idea was first suggested by Carter (Bainbridge et al., 1997b), but as Carter himself points out, this would commit the measurement to one particular software package, warts and all.

Here we refine the idea to editing operations using a hypothetical music editor. No rigorous definition exists for the operations of the hypothetical editor, rather, if we can justify a straightforward editing operation which requires only a few keyboard[5] presses and/or clicks of the mouse, then we assign a cost of one to the operation. More complicated operations must be built up from these atomic steps. Some examples of simple operations are:

- adding a note,
- "cutting" a group of musical features,
- "pasting" a group of musical features,
- altering the pitch of a note, and
- changing the key signature on a staff which then automatically updates any affected notes.

For example, when there is a series of triplets in a piece of music, a common habit in CMN is to drop the ornamental '3' after the first few occurrences. Unless the OMR system employs some form of global analysis of the detected timing information, these omissions result in mistakes that carry through to the reconstructed score. In the music editing package Finale, the conversion of three quavers beamed together into a triplet is convoluted, requiring many operations. By comparison, in the hypothetical editor we could imagine grouping the notes in question using the mouse, and then selecting a "make triplet" option from the user interface. Such a correction requires only a few keyboard and mouse presses and therefore carries the operational cost of one.

This example highlights the difference between the functionality provided by existing music notation software, and the functionality required by an OMR package. Existing editors have not been specifically designed for OMR applications, which is something that will need to change in the future. Grande Software boast the first commercial system with integrated editor and OMR software (Selfridge-Field, 1994), but their description of the product talks about the OMR package being "interfaced" to the notation program rather than being developed in parallel. What the hypothetical editor allows us to assume is a working environment where the OMR system and the editor are fully integrated. If such an integrated tool existed, the next release of the editor can always be updated to incorporate new functionality as dictated by repetitive OMR processing errors.

For completeness, we now summarise the reported accuracy rates of prominent OMR systems. Generally, authors express the number of correctly processed shapes as a percentage of the shapes the OMR system *should* have processed:

the Wabot-2 project (Matsushima et al., 1985) reports an accuracy rate of "nearly 100%" for ten simple scores of organ music; Kato and Inokuchi (1990) report accuracy rates of 83–95% (an average of 89%) for four test scores of varying complexity; Modayur et al. (1995) report an overall accuracy rate of 96% for 74 images (however these images are small – the total surface area of the 74 images is approximately equivalent to two A4 pages); Reed (1995) reports accuracy rates in the range 78–100% (an average of 95%) for nine images of printed CMN; and Bainbridge (1997) reports an overall accuracy rate that exceeds 96% (staff detection 100%, primitive detection 96.8%, primitive assembly 97.5%, and musical semantics 98.6%) for eleven test images.

7. Conclusion

In this article we have highlighted the challenge posed by optical music recognition and described significant research projects in this area. The article also shows a generalised framework (Figure 9) for OMR – which existing work fits into – that emphasises the key stages in the process, although there are diverse methods in the literature for implementing each stage.

Perhaps the most open question in OMR is how to compare the success and accuracy of rival OMR systems. Here we recommend, as important steps in this process, the need for a corpus of musical images and a more detailed breakdown of an OMR system's accuracy rates, with errors measured by the number of editing operations required in an abstract music editor.

Notes

[1] An A4 page scanned at 300 dpi occupies around 1 Mbyte uncompressed.
[2] For information about NIFF, e-mail James Tauber at `jtauber@jtauber.com`
[3] For further details, contact the member body of ISO in your own country.
[4] PRIMELA is an acronym for PRIMitive Expression LAnguage.
[5] Either computer keyboard or musical keyboard.

References

Alphonce, B., B. Pennycook, I. Fujinaga and N. Boisvert. "Optical Music Recognition: A Progress Report". In *Proceedings of the Small Computers in the Arts*, 1988, pp. 8–12.

Anstince, J. *A Pen-Input Computer Music Editing System*. M.Sc. thesis, Department of Computer Science, University of Canterbury, Christchurch, NZ, 1996.

Bainbridge, D. and T. Bell. "An Extensible Optical Music Recognition System". In *Proceedings of the Nineteenth Australasian Computer Science Conference*. Melbourne, 1996, pp. 308–317.

Bainbridge, D. and T. Bell. "Dealing with Superimposed Objects in Optical Music recognition". In *Proceedings of the Sixth International Conference on Image Processing and Its Applications*. Dublin, 1997, pp. 756–760.

Bainbridge, D. and N. Carter. "Automatic Reading of Music Notation". In *Handbook on Optical Character Recognition and Document Image Analysis*. Ed. H. Bunke and P.S.P. Wang. Singapore, World Scientific, 1997, pp. 583–603.

Bainbridge, D. and S. Inglis. "Music Image Compression". In *Proceedings of the IEEE Data Compression Conference*. Ed J. Storer and M. Cohn. Snowbird, Utah: IEEE Computer Society Press, pp. 208–218.

Bainbridge, D. *A Complete Optical Music Recognition System: Looking to the Future*. Technical report tr-cosc 06.94. Christchurch, NZ: Department of Computer Science, University of Canterbury, March 1994.

Bainbridge, D. *Optical Music Recognition: Progress Report 1*. Technical report tr-cosc 05.94. Christchurch, NZ: Department of Computer Science, University of Canterbury, November 1994.

Bainbridge, D. "Playing Musical Computers". *New Zealand Science Monthly*, 5 (November 1994), 10–11.

Bainbridge, D. "Optical Music Recognition: A Generalised Approach". In *Proceedings of the Second New Zealand Computer Science Research Students' Conference*. Waikato, NZ, April 1995, pp. 23–30.

Bainbridge, D. *Optical Music Recognition: Progress Report 2*. Technical report tr-cosc 02.95. Christchurch, NZ: University of Canterbury, January 1994.

Bainbridge, D. *Extensible Optical Music Recognition*. Ph.D. thesis, Department of Computer Science, University of Canterbury, Christchurch, NZ, 1997.

Baird, H.S., H. Bunke and K. Yamamoto, Ed. *Structured Document Image Analysis*. Berlin: Springer-Verlag, 1992.

Baumann, S. and A. Dengel. "Transforming Printed Piano Music into MIDI". In *Advances in Structural and Syntactic Pattern Recognition* (*Proceedings of International Workshop on Structural and Syntactic Pattern Recognition*), volume 5 of *Series in Machine Perception and Artificial Intelligence*. Ed. H. Bunke. Bern: World Scientific, 1992, pp. 363–372.

Blostein, D. and H.S. Baird. "A Critical Survey of Music Image Analysis". In *Structured Document Image Analysis*. Ed. H.S. Baird, H. Bunke and K. Yamamoto. Berlin: Springer-Verlag, 1992, pp. 405–434.

Boyle, R. and R. Thomas. *Computer Vision: A First Course*. Oxford: Blackwell Scientific, 1988.

Bulis, A., R. Almog, M. Gerner and U. Shimony. "Computerized Recognition of Hand-Written Musical Notes". In *Proceedings of the International Computer Music Conference*. San Jose, California, 1992, pp. 110–112.

Bunke, H. and P.S.P. Wang, Ed. *Handbook of Character Recognition and Document Image Analysis*. Singapore: World Scientific, 1997.

Bunke, H. "Attributed Programmed Graph Grammars and Their Application to Schematic Diagram Interpretation". *IEEE Pattern Analysis and Machine Intelligence*, 4(6) (Novemver 1982), 574–582.

Carter, N.P. and R.A. Bacon. "Automatic Recognition of Music Notation". In *Proceedings of the International Association for Pattern Recognition Workshop on Syntactic and Structural Pattern Recognition*. Murray Hill, New Jersey, June 1990, p. 482.

Carter, N.P. and R.A. Bacon. "Automatic Recognition of Printed Music". In *Structured Document Image Analysis*. Ed. H.S. Baird, H. Bunke and K. Yamamoto. Berling: Springer-Verlag, 1992, pp. 456–465.

Carter, N.P. *Automatic Recognition of Printed Music in the Context of Electronic Publishing*. Ph.D. thesis, Departments of Physics and Music, University of Surrey, Guildford, UK, February 1989.

Carter, N.P. "A New Edition of Walton's Façade Using Automatic Score Recognition". In *Advances in Structural and Syntatic Pattern Recognition* (*Proceedings of International Workshop on Structural and Syntactic Pattern Recognition*), volume 5 of *Series in Machine Perception and Artificial Intelligence*. Ed. H. Bunke. Bern: World Scientific, 1992, pp. 352–362.

Carter, N.P. "Segmentation and Preliminary Recognition of Madrigals Notated in White Mensural Notation". *Machine Vision and Applications*, 5(3) (1992), 223–230.

Carter, N.P. *A Generalized Approach to Automatic Recognition of Music Scores*. Technical Report STAN-M-87. California, USA: Department of Music, Stanford University, December 1993. 77 pp.

Carter, N.P. "Conversion of the Haydn Symphonies into Electronic form Using Automatic Score Recognition: A Pilot Study". In *Proceedings of the IS&T/SPIE's Symposium on Electronic Imaging: Science and Technology; Conference 2181 – Document Recognition*. Ed. L.M. Vincent and T. Pavlidis. San Jose, California, February 1994, pp. 279–290.

Carter, N.P. "Music Score Recognition: Problems and Prospects". *Computing in Musicology: An International Directory of Applications*, 9 (1994), 152–158.

Clarke, A.T., B.M. Brown and M.P. Thorne. "Inexpensive Optical Character Recognition of Music Notation: A New Alternative for Publishers". In *Proceedings of the Computers in Music Research Conference*. Luncaster, UK, April 1988, pp. 84–87.

Coüasnon, B. and J. Camillerapp. "Using Grammar to Segment and Recognize Music Scores". In *International Association for Pattern Recognition Workshop on Document Analysis Systems*. Kaiserslauter, Germany, October 1994, pp. 15–27.

Coüasnon, B., P. Brisset and I. Stephan. "Using Logic Programming Languages for Optical Music Recognition". In *The Third International Conference on the Practical Application of Prolog*. Paris, April 1995, pp. 115–134.

Coüasnon, B. *Segmentation and Recognition of Documents Guided by A Priori Knowledge: Application to Musical Scores*. Ph.D. thesis, IRISA, France, 1997.

Cover, T.M. and P.E. Hart. "Nearest Neighbor Classification". *IEEE Transactions of Information Theory*, 13 (1967), 21–27.

Fahmy, H. and D. Blostein. "A Graph Grammar for High-Level Recognition of Music Notation". In *Proceedings of First International Conference on Document Analysis and Recognition*, volume 1. Saint Malo, France, 1991, pp. 70–78.

Fahmy, H. and D. Blostein. "Graph Grammar Processing of Uncertain Data". In *Advances in Structural and Syntactic Pattern Recognition (Proceedings of International Workshop on Structural and Syntactic Pattern Recognition)*, volume 5 of *Series in Machine Perception and Artificial Intelligence*. Ed. H. Bunke. Bern: World Scientific, 1992, pp. 373–382.

Foley, J.D., A. van Dam, S.K. Feiner and J.F. Hughes. *Computer Graphics: Principles and Pratice*. Reading, Massachusetts: Addison-Wesley, 1990.

Fujinaga, I., B. Alphonce and B. Pennycook. "Issues in the Design of an Optical Music Recognition Systems". In *Proceedings of the International Computer Music Conference*. Ohio State University, November 1989, pp. 113–116.

Fujinaga, I., B. Alphonce, B. Pennycook and N. Boisvert. "Optical Recognition of Music Notation by Computer". *Computers in Music Research*, 1 (1989), 161–164.

Fujinaga, I., B. Pennycook and B. Alphonce. "Computer Recognition of Musical Notation". In *Proceedings of the First International Conference on Music Perception and Cognition*. 1989, pp. 87–90.

Fujinaga, I., B. Alphonce and K. Hogan. "Optical Music Recognition: Progress Report". In *Proceedings of the International Computer Music Conference*. Montreal, Canada, 1991, pp. 66–73.

Fujinaga, I., B. Pennycook and B. Alphonce. "The Optical Music Recognition Project". *Computers in Music Research*, 3 (1991), 139–142.

Fujinaga, I., B. Alphonce, B. Pennycook and G. Diener. "Interactive Optical Music Recognition". In *Proceedings of the International Computer Music Conference*. San Jose, California, 1992, pp. 117–120.

Fujinaga, I. *Optical Music Recognition Using Projections*. M.Sc. thesis, McGill University, Montreal, Canada, 1988.

Fujinaga, I. "An Optical Music Recognition System that Learns". In *Enabling Technologies for High-Bandwidth Applications*, volume SPIE 1785. Ed. J. Maitan. 1992, pp. 210–217.

Fujinaga, I. *Adaptive Optical Music Recognition*. Ph.D. thesis, Department of Theory, Faculty of Music, McGill University, Montreal, Canada, 1997.

Glass, S. *Optical Music Recognition*. B.Sc thesis, Department of Computer Science, University of Canterbury, Christchurch, NZ, 1989.

Haken, L. and D. Blostein. "The Tilia Music Representation: Extensibility, Abstraction, and Notation Contexts for the Lime Music Editor". *Computer Music Journal*, 17(3) (1993), 43–58.

Heussenstamm, G. *The Norton Manual of Music Notation*. New York: W. W. Norton, 1987.

Kassler, M. "Optical Character Recognition of Printed Music: A Review of Two Dissertations". *Perspectives of New Music*, 11(2) (1972), 250–254.

Kato, H. and S. Inokuchi. "A Recognition System for Printed Piano Music Using Musical Knowledge and Constraints". In *Proceedings of the International Association for Pattern Recognition Workshop on Syntactic and Structural Pattern Recognition*. Murray Hill, New Jersey, June 1990, pp. 231–248.

Matsushima, T., T. Harada, I. Sonomoto, K. Kanamori, A. Uesugi, Y. Nimura, S. Hashimoto and S. Ohteru. "Automated Recognition System for Musical Score – the Vision System of WABOT-2". In *Bulleting of Science and Engineering Research Laboratory*, volume 112. Waseda University, Tokyo, Japan, September 1985, pp. 25–52.

McGee, W.F. and P. Merkley. "The Optical Scanning of Medieval Music". *Computers and the Humanities*, 25(1) (1991), 47–53.

Modayur, B.R., V. Ramesh, R.M. Maralick and L.G. Shapiro. *MUSER: A Protoype Musical Score Recognition System Using Mathematical Morphology*, Technical Report. Seattle, USA: Intelligent Systems Laboratory, Electrical Engineering Department, University of Washington, June 1995.

Ng, K.C., R.D. Boyle and D. Cooper. "Automated Optical Musical Score Recognition and Its Enhancement Using High-Level Musical Knowledge". In *Proceedings of the XI Colloquium on Musical Informatics*. Bologna, Italy, November 1995, pp. 167–170.

Ng, K.C. and R.D. Boyle. "Recognition and Reconstruction of Primitives in Music Scores". *Image and Vision Computing*, 14 (1996), 39–46.

Ng, K.C. *Automated Computer Recognition of Music Scores*. Ph.D. thesis, University of Leeds, Leeds, UK, 1995.

Pavlidis, T. *Algorithms for Graphics and Image Processing*. Rockville, Maryland: Computer Science Press, 1982.

Prerau, D.S. *Computer Pattern Recognition of Standard Engraved Music Notation*. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, September 1970.

Prerau, D.S. "Computer Pattern Recognition of Printed Music". In *Proceedings of the Fall Joint Computer Conference*. Montvale, New Jersey: AFIPS Press, November 1971.

Prerau, D.S. "Do-Re-Mi: A Program that Recognizes Music Notation". *Computers and the Humanities*, 9(1) (January 1975), 25–29.

Pruslin, D. *Automatic Recognition of Sheet Music*. Sc.D. dissertation, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, June 1966.

Read, G. *Music Notation: A Manual of Modern Practice*. London: Victor Gollancz, 1974.

Reed, T. *Optical Music Recognition*. M.Sc. thesis, Department of Computer Science, University of Calgary, Canada, September 1995.

Roach, J.W. and J.E. Tatem. "Using Domain Knowledge in Low-Level Visual Processing to Interpret Handwritten Music: An Experiment". *Pattern Recognition*, 21(1) (1988), 33–44.

Roads, C. "The Tsukuba Mustical Robot". *Computer Music Journal*, 10(2) (1986), 39–43.

Ross, T. *The Art of Music Engraving and Processing: A Complete Manual Reference and Text Book on Preparing Music for Reproduction and Print*. Miami: Hansen Press, 1970.

Russ, J.C. *The Image Processing Handbook*. Boca Raton, Florida: CRC Press, 1992.

Selfridge-Field, E. "Optical Recognition of Music Notation: A Survey of Current Work". *Computing in Musicology: An International Directory of Applications*, 9 (1994), 109–145.

Witten, I., A. Moffat and T. Bell *Managing Gigabytes: Compressing and Indexing Documents and Images*. New York: Van Nostrand Reinhold, 1994.

Wolman, J., J. Choi, S. Asgharzadeh and J. Kahana. "Recognition of Handwritten Music Notation". In *Proceedings of the International Computer Music Conference*. San Jose, California, 1992, pp. 125–127.

Yadid, O., E. Brutman, L. Dvir, M. Gerner and U. Shimony. "RAMIT: Neural Network for Recognition of Musical Notes". In *Proceedings of the International Computer Music Conference*. San Jose, California, 1992, pp. 128–131.