Working Paper Series
ISSN 1170-487X

Musical Image Compression

by David Bainbridge and Stuart Inglis

Working Paper 97/25
November 1997

© 1997 David Bainbridge &
Stuart Inglis
Department of Computer Science
The University of Waikato
Private Bag 3105
Hamilton, New Zealand

Musical Image Compression

David Bainbridge & Stuart Inglis

Department of Computer Science
University of Waikato
Hamilton, New Zealand

{*d.bainbridge, s.inglis*}*@cs.waikato.ac.nz*

1 Introduction

Optical music recognition aims to convert the vast repositories of sheet music in the world into an on-line digital format [Bai97]. In the near future it will be possible to assimilate music into digital libraries and users will be able to perform searches based on a sung melody in addition to typical text-based searching [MSW+96]. An important requirement for such a system is the ability to reproduce the original score as accurately as possible. Due to the huge amount of sheet music available, the efficient storage of musical images is an important topic of study.

This paper investigates whether the "knowledge" extracted from the optical music recognition (OMR) process can be exploited to gain higher compression than the JBIG international standard for bi-level image compression. We present a hybrid approach where the primitive shapes of music extracted by the optical music recognition process—note heads, note stems, staff lines and so forth—are fed into a graphical symbol based compression scheme originally designed for images containing mainly printed text. Using this hybrid approach the average compression rate for a single page is improved by 3.5% over JBIG. When multiple pages with similar typography are processed in sequence, the file size is decreased by 4–8%.

Section 2 presents the relevant background to both optical music recognition and textual image compression. Section 3 describes the experiments performed on 66 test images, outlining the combinations of parameters that were examined to give the best results. The initial results and refinements are presented in Section 4, and we conclude in the last section by summarizing the findings of this work.

2 Background

In processing a scanned piece of music, an OMR system extracts specific knowledge about the layout and structure of the page. Moreover, the process breaks the page down into primitive shapes that are similar in idea to the isolated shapes—also known as marks—of a textual image. For textual images, previous work has shown that processing the data at this symbol level yields better compression than working at the pixel level [Ing97, WBE+94]. In this paper we ask the question, can a similar exploitation of symbols be used in the context of musical images?

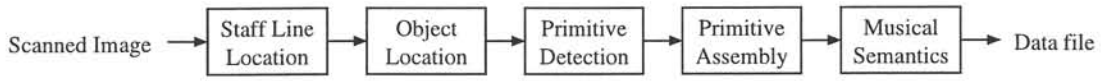


Figure 1: Key stages in the OMR process.

2.1 Textual images

If an image contains mainly printed text, these images are historically called *textual images*. Textual images are commonly generated during the building of a digital library. Pages from books and periodicals are scanned and stored electronically where they can be compressed and indexed for later retrieval.

If scanned images are stored naively, they typically require about 1 Mbyte of space. Because textual images contain large expanses of white areas and repeated characters, the image can be stored more efficiently. Ascher and Nagy [AN74] first introduced a method for compressing textual images by extracting characters from an image, and building a library that contains a single exemplar of each character. The library is compressed using standard bi-level image compression techniques, and the indices into the library are compressed using standard text compression techniques. This compression process is lossy, as only a single "e" will occur in the reconstructed image. To make the process lossless, the difference between the original and the reconstruction must also be encoded—the difference is known as the *residue* image.

In this paper we will be using the TIC textual image compression system [Ing97] that was first introduced by Witten *et al.* in both its lossy and lossless forms [WMB94].

2.2 Optical music recognition

Figure 1 shows the main steps to the OMR process. First the staff lines are located. The most common approach is to form a histogram by counting the number of pixels per horizontal line (one on the left-hand side and the other on the right) and then detect any distinct peaks (Figure 2a). Using the peak information, the skew angle can be determined and the image rotated to correct the skew.

The individual musical features are located next (Figure 2b). The feature location process is complicated as staff lines are superimposed over many important musical features. All reported algorithms perform this segmentation task with varying degrees of imperfection [Car89, MB91, CBS95, BC97]. A popular technique, first described by Clarke *et al.* [CBT88], removes a staff line incrementally by considering one vertical slither of a staff line at a time. Working from left to right, the area either above or below the staff line is checked for evidence of musical objects. If no evidence is found, then the slither of staff line is removed. The check for the existence of a musical object is performed locally, searching a region no more than two pixels away from the top or bottom of that part of the staff line. Consequently musical features that blend tangentially into a staff line, such as bass clefs and minim notes, become fragmented and subsequent processing stages in the OMR process must be tolerant to these defects. An example of this occurs in Figure 2b, where the bass clef at the

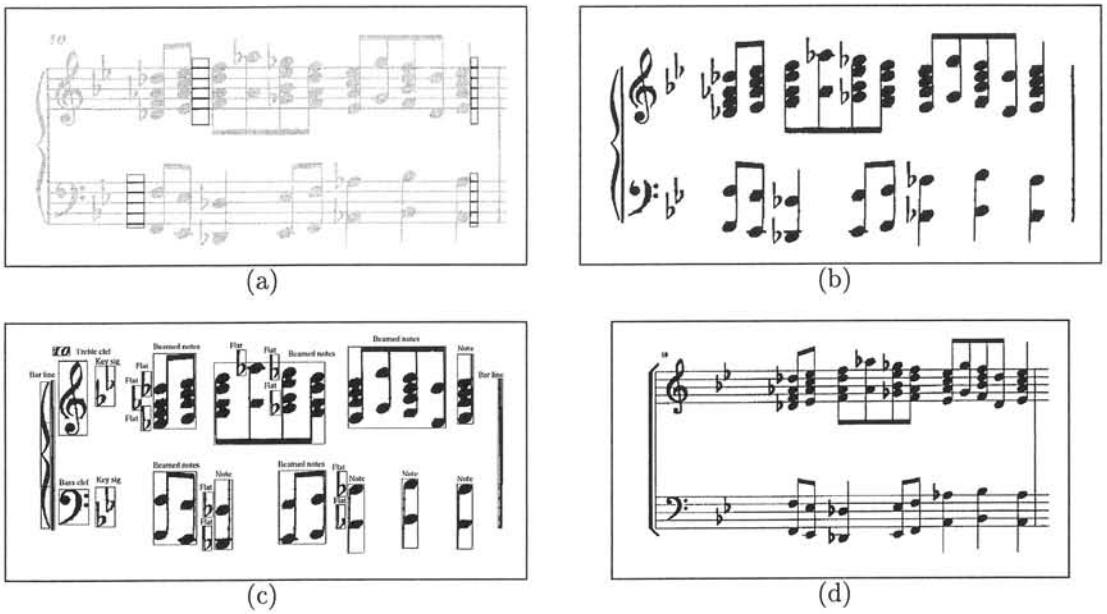


Figure 2: The OMR knowledge extraction process (a) locating of the staff lines (b) locating the musical features (c) assembling primitive shapes back into musical features (d) reconstructing the score from the output file using a music editor.

start of the lower staff is broken in two.

The image produced by object location is now similar to lines of text; however, the pattern recognition task contrasts with optical character recognition since the shapes are more intricate and variable. The solution is to decompose the task into the recognition of basic primitive shapes—note heads, beams, dots, and others—and then assemble them back into musical features (Figure 2c).

The final step extracts the musical semantics of the score. This produces a graph where the intricate two-dimensional relationships of music are mapped onto the assembled musical features, storing information such as pitch and duration as attributes and forming links between them. It is then a simple matter to traverse this graph to generate the desired musical file format—for instance, an audio based format such as MIDI or a musical editor based format such as Tilia [SF97] (Figure 2d).

The size of the generated file is much smaller than the original image. For example, one experiment performed by the authors saw images of typically 1 Mbyte reduce to approximately 1500 bytes in the Tilia format; however, mistakes made in any one of the OMR stages lead to erroneous data in the output file. And even if a file was 100% correct, the format only encapsulates the *essence* of the music. It is unlikely that the reconstructed score will use the same musical fonts or page layout. In short, the reconstructed score will never be a pixel perfect replica of the original. This is apparent in Figure 2 where the reconstructed score (Figure 2d) is musically equivalent to the original (Figure 2a) but typographically quite different.

Figure 3 shows a compromise between compression rate and knowledge extraction.

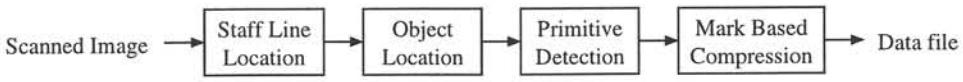


Figure 3: Altering the OMR process to compress images.

The first three stages to the OMR process are used to extract the primitive shapes contained in the musical image as before; but this data is now passed to the symbol based compression scheme TIC described above. Using such a system it becomes possible to store the musical image in a more condensed format than the original, yet still be able to reconstruct the image pixel perfect.

This idea was originally proposed by the first author [Bai97]; no other work in this area is known. The term MIC (Musical Image Compression) will be used subsequently to refer to this compression scheme.

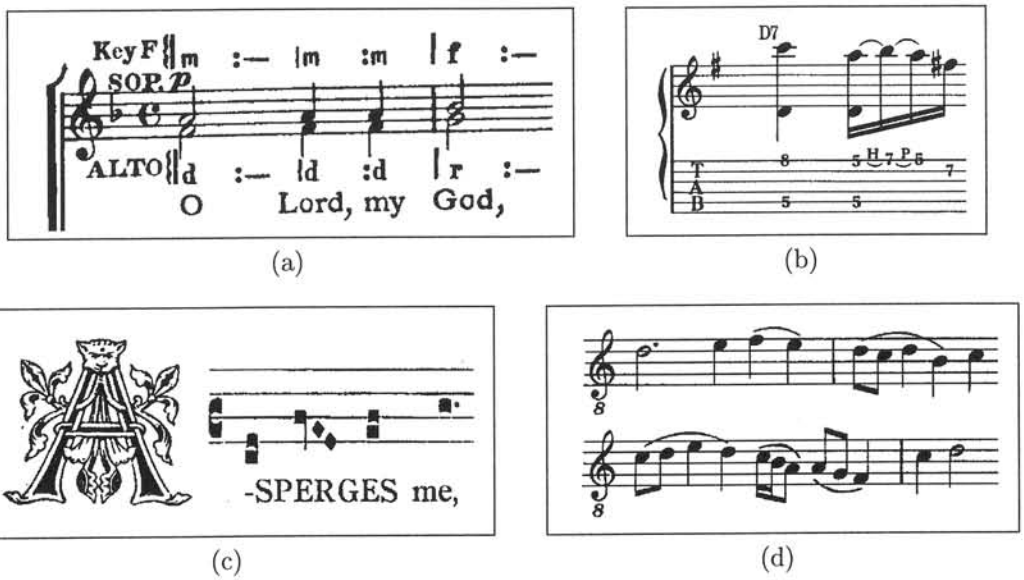
2.3 Musical primitives for mark based compression

To realize the compression scheme shown in Figure 3, the primitive detection stage must be modified to generate shapes suitable for the mark based compression scheme. However, there are typographical differences between music and text that need addressing. For instance, compared to text, there is no clear horizontal line of marks, and the width and height of primitives are more diverse—most notably staff lines which typically span the image but are only a few pixels high.

This study of differences led to four parameters that control how the primitives detected by the OMR process are generated for TIC. They are: grouping, primitive ordering, order tolerance, and size bounding. *Grouping* musical primitives by area mimics the idea of lines of text. For example, the primitive detection stage could generate all the primitives that fall inside the first staff, followed by the shapes that lie between the first and second staff, followed by the second staff and so on. Within one such rectangular area, the primitives could be *ordered*. For example, "order xy" would list all primitives on their x value, and for any primitives that shared the same x position, their y values would be used. Additionally, this ordering could be subject to a *tolerance*, where a certain amount of deviation in the first value can occur before the second value is used to disambiguate its position in the list. Finally, the size of the marks generated could be restricted by *size bounding*. Any primitive larger than the prescribed amount is divided into smaller pieces.

3 Experimentation

The image corpus that was used for experimentation was a musical image corpus created for OMR work [Bai97]. The collection contains 66 works—191 images in all—taken from a diverse selection of music notation, including sol-fa, guitar tablature and plainsong, in addition to the popular Western style of notation known as Common Music Notation (CMN), examples of which are shown in Figure 4. The number of pages in each work range from one to ten and the scan resolution used was 300 dots



(a) (b) (c) (d)

Figure 4: Examples of music notation from the corpus (a) sol-fa (b) guitar tablature (c) plainsong (d) common music notation.

per inch.

The corpus is further divided according to typesetting criteria. Categories include: orchestrated score, miniature score, accompaniment, monolinear music (no chords), individual instrument, hymn, percussion, computer typeset and high quality handwritten music, where one corpus entry may fit into more than one category.

The main experiment compressed the first page of each corpus entry. The attributes used were:

- *Grouping*—staff, staff system, page.
- *Ordering*—xy, yx, none.
- *Order tolerance*—0, 1, 2, 3, 5, 10.
- *Size bounding staff lines in the x-dimension*—100, 200, 300, none.

The second experiment focused on compressing multiple files. For this, three pieces (each six pages long) were studied and processed using the best set of parameters identified by the main experiment.

One final modification made to the OMR process was the number of different primitive shapes recognized. For complete OMR it is necessary to correctly identify all musical primitives. However, since the OMR process removes the primitives detected, after processing a few of these shapes, the remaining primitives become graphically isolated and therefore can be easily extracted as marks. In this experiment the detected primitives were: treble clef, bass clef, vertical lines, beams, and filled-in note heads.

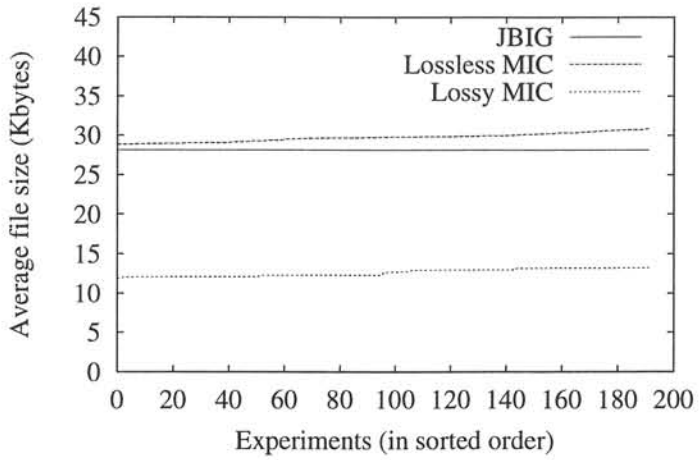


Figure 5: The results of the main experiment: 192 variations of 4 attributes averaged over 66 images calculated using lossy and lossless versions of TIC.

4 Results

The values of the four attributes were enumerated through their 192 combinations. For each parameter combination, the OMR format was converted to the TIC format using the set of parameters. The TIC system compressed each of the 66 images using these options. Once all 192 combinations were processed, they were sorted in order of compression performance. Figure 5 shows the results of the main experiment. The average file size for JBIG is shown as a horizontal line. JBIG required 28,143 bytes on average to compress the images.

The best parameter selection for lossless compression was: *size bounding* of 300, *grouping* by staff, *tolerance* of 0 and *ordering* of none. This set of parameters decreased the average file size from 30,868 bytes to 28,859 bytes, a reduction of 2009 bytes or 6.5%. However, the average file size is larger than JBIG by 716 bytes or 2.5%. For lossy compression the average file size over the 66 image corpus was 11,780 bytes—an improvement of 58.1% over JBIG. The best parameter selection for the lossy method was: *size bounding* of 300, *grouping* by staff system, *tolerance* of 10 and *ordering* of none.

To ascertain which parameters were important, the data from Figure 5 was analyzed using Correlation Feature Selection (CFS) which is based on Minimum Description Length principles [HS96]. CFS selected attributes *size bounding* and *grouping* as having significant influence.

Figure 6 shows the data from the main experiment again, filtered using these two attributes. The parameters *ordering* and *tolerance* were set to none and 0 respectively as these were the values used by the top parameter combination of the 192 variations. The figure shows that the best configuration divides staff lines in the x-dimension into 300 pixel blocks and groups all primitives by rectangular areas based on the staff areas in the page.

Size bounding	*Grouping*	*Bytes*
300	staff	28,859
300	system	28,943
200	staff	28,950
200	system	29,038
100	staff	29,285
100	system	29,381
none	staff	29,820
300	page	29,847
none	system	29,902
200	page	29,962
100	page	30,347
none	page	30,738

Figure 6: The main experiment filtered by *size bounding* and *grouping* with *ordering* set to none and *tolerance* set to 0.

4.1 Refinements

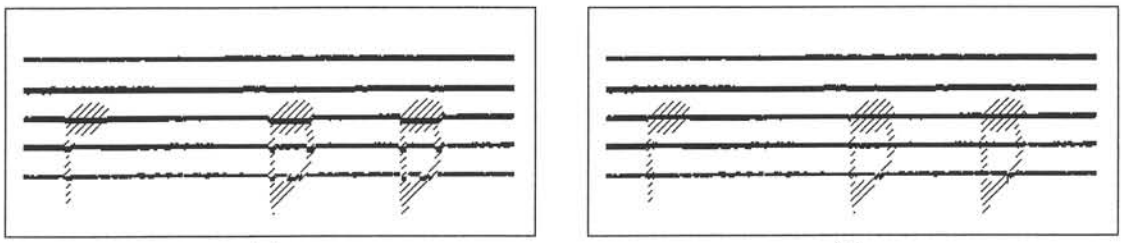
In the previous section the input to the conversion between the OMR system and the TIC system had been unaided. After generating the initial results, we examined the mark file, looking for possible refinements.

4.1.1 Cropping

During the conversion from the OMR system to TIC, marks were divided into smaller pieces. Often this process resulted in marks that had surrounding white space. If these extra pixels were removed, the average file size reduced to 28,656 bytes. This is a reduction of 200 bytes or 0.7%. Compression was achieved as fewer pixels were encoded and the white perimeter of the mark was removed. The removal of the white perimeter meant that the common pixel contexts were all black, which increased prediction accuracy.

4.1.2 Staff line extraction

An analysis of the symbol libraries showed that the reuse of 300 pixel long staff line components within an image was not as high as expected. Extracted staff lines include noise because it is difficult to determine their exact position when they pass through musical features. The algorithm used, known as "track", has been empirically found to be best for OMR [Bai97]. However, it is prone to small vertical displacements in the staff line position as it passes through a musical feature (Figure 7a). Since the staff lines are horizontally wide and vertically thin, a small vertical displacement can disproportionately effect the component matching process inside TIC. An alternative algorithm, known as "wobble" is less prone to vertical errors (Figure 7b). When this algorithm was used for compression, the result was a reduction in the average file size to 27,512 bytes. Thus, the total corpus size is 2.2%



(a) (b)

Figure 7: Comparison of two staff line extraction algorithms (a) track (b) wobble.

smaller than when compressed with JBIG and the average compression improvement is 3.5%.

4.2 Multiple pages

Multiple pages in a score usually have homogeneous typography. This is because publishers use the same software or publishing equipment to produce the score. A mark-based approach like TIC is suited for processing multiple pages, as the library rapidly adapts to contain representative samples of the common objects in the image.

Figure 8 shows the compression improvement that is possible using each scheme as multiple pages are processed. The test images were generated by concatenating the pages together (top to bottom) into one large bitmap. For example, to analyze the results after four pages, the first four pages were merged together into one large bitmap, and this bitmap was compressed using both JBIG and MIC. Figure 8a shows a logarithmic improvement over the first six pages. MIC gives a 7.5% improvement while JBIG compression actually decreases by 0.6%. Figures 8b and 8c show similar improvements for MIC over multiple pages. For both these figures, JBIG shows no improvement.

5 Summary

This paper introduces a new field for compression research: the compression of scanned musical images. We combined an optical music recognition system with a symbol-based textual image compression system to give higher compression than JBIG.

The total compressed size of the 66 musical images was reduced by 2.2% using the hybrid method presented (an average compression of 3.5% per file). When a series of six images from the same score was compressed, the MIC technique increased compression by 4–8%. JBIG did not gain any compression by seeing multiple pages. Nor does it have the option of producing lossy images, which are 58.1% better compressed than JBIG.[1]

[1] We would have liked to compare our work with JBIG-2 since this includes a lossy form, however the standard is yet to be finalized, precluding such a comparison.

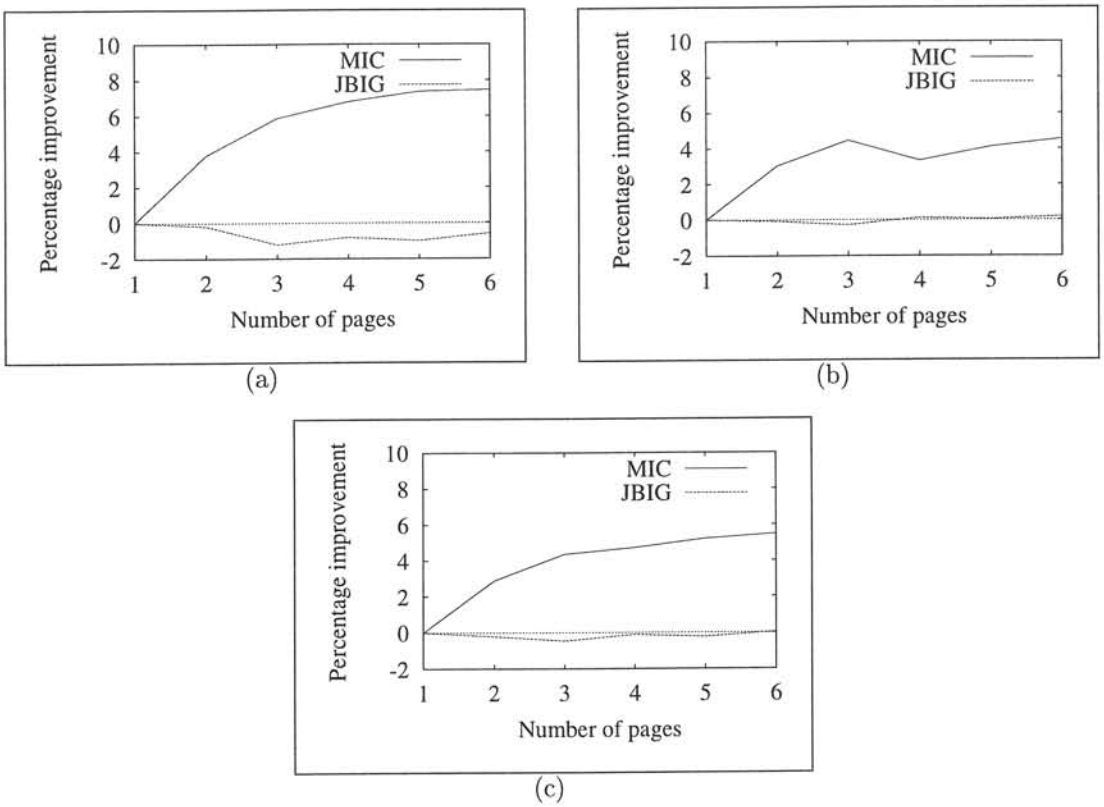


Figure 8: Compression improvement using multiple pages (a) "The Heart Asks Pleasure First" (b) "Great Balls of Fire" followed by "Wake Up, Little Susie" (c) "Bohemian Rhapsody."

The work has also shown that the best technique for staff line extraction in an OMR system is not necessarily the best technique for musical image compression. It seems that because OMR is a one-way process, small errors and problems can be overlooked during the segmentation stage, but when we need a reversible (compression/decompression) process, different techniques are better suited.

Finally, because the performance of the textual image compression system is dependent on the quality of the component segmentation, it seems that this technique of musical image compression could be used as an independent gauge of accuracy for the OMR community.

References

[AN74] R. N. Ascher and G. Nagy, *A means for achieving a high degree of compaction on scan-digitized printed text*, IEEE Transactions of Computers **23** (1974), no. 11, 1174–1179.

[Bai97] D. Bainbridge, *Extensible optical music recognition*, Ph.D. thesis, Department of Computer Science, University of Canterbury, NZ, 1997.

[BC97] D. Bainbridge and N. Carter, *Automatic reading of music notation*, Handbook on Optical Character Recognition and Document Image Analysis (H. Bunke and P. S. P. Wang, eds.), World Scientific, Singapore, 1997, pp. 583–603.

[Car89] N. P. Carter, *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.

[CBS95] B. Coüasnon, P. Brisset, and I. Stephan, *Using logic programming languages for optical music recognition*, The Third International Conference on the Practical Application of Prolog (Paris), April 1995, pp. 115–134.

[CBT88] A. T. Clarke, B. M. Brown, and M. P. Thorne, *Inexpensive optical character recognition of music notation: A new alternative for publishers*, Proceedings of the Computers in Music Research Conference (Lancaster, UK), April 1988, pp. 84–87.

[HS96] M. Hall and L. A. Smith, *Practical feature subset selection for Machine Learning*, Proceedings of the Australian Computer Science Conference (University of Western Australia), February 1996.

[Ing97] S. J. Inglis, *Document image compression*, Ph.D. thesis, Department of Computer Science, University of Waikato, Hamilton, NZ, 1997, (*in preparation*).

[MB91] P. Martin and C. Bellissant, *Low-level analysis and recognition of music drawing images*, Proceedings of First International Conference on Document Analysis (Saint-Malo, France), vol. 1, 1991, pp. 417–425.

[MSW+96] R. J. McNab, L. A. Smith, I. H. Witten, C. L. Henderson, and S. J. Cunningham, *Towards the digital music library: tune retrieval from acoustic input*, Proceedings of Digital Libraries '96, 1996, pp. 11–18.

[SF97] E. Selfridge-Field (ed.), *Beyond Midi: The handbook of musical codes*, MIT Press, Cambridge, Massachusetts, USA, 1997.

[WBE+94] I. H. Witten, T. C. Bell, H. Emberson, S. Inglis, and A. Moffat, *Textual image compression: Two-stage lossy/lossless encoding of textual images*, Proceedings of the IEEE **82** (1994), no. 6, 878–888.

[WMB94] I. H. Witten, A. Moffat, and T. C. Bell, *Managing Gigabytes: Compressing and indexing documents a nd images*, Van Nostrand Reinhold, NY, USA, 1994.