

## DEALING WITH SUPERIMPOSED OBJECTS IN OPTICAL MUSIC RECOGNITION

D Bainbridge and T C Bell

Universities of Waikato and Canterbury, New Zealand

### INTRODUCTION

Optical Music Recognition (OMR) involves identifying musical symbols on a scanned sheet of music, and interpreting them so that the music can either be played by the computer, or put into a music editor. Applications include providing an automatic accompaniment, transposing or extracting parts for individual instruments, and performing an automated musicological analysis of the music.

A key problem with music recognition, compared with character recognition, is that symbols very often overlap on the page. The most significant form of this problem is that the symbols are superimposed on a five-line staff. Although the staff provides valuable positional information, it creates ambiguity because it is difficult to determine whether a pixel would be black or white if the staff line was not there. Symbols also overlap because they can have multiple components, such as a note with a slur in front of it. If the music is engraved and reproduced accurately then these should not overlap, but in practice it is inevitable that some will end up touching.

The other main difference between music recognition and character recognition is the set of permissible symbols. In text, the alphabet size is fixed, and—to aid readability—each symbol is intentionally different. Conversely, in music notation there is no standard “alphabet” of shapes, with composers inventing new notation where necessary, and music for particular instruments using specialised notation where appropriate. Furthermore, different shapes in music notation are often *intentionally* similar in appearance, where the small differences substantially change the musical characteristic of the shape. To counter this problem we have designed a system that is extensible in the music notation it can process. This aspect of our project is described in Bainbridge and Bell (1). The focus of this paper is on techniques we have developed to deal with superimposed objects.

For most OMR systems, four stages can be identified: *staff line identification*, *musical object location*, *musical feature classification*, and *musical semantics*. The first three stages are greatly affected by the problem of superimposed symbols, and this paper focuses on ways in which they can deal with this issue.

### STAFF LINE IDENTIFICATION

The staff line identification stage involves looking for the horizontal lines on which objects are superimposed. A popular approach that accomplishes this is to detect peaks in a horizontal projection of the image—see Blostein and Baird (2) for a survey. These lines, however, will not necessarily be straight, since distortions in the printing of the page or the scanning process may produce skewed, bent, or broken lines. Existing algorithms are tolerant of these deviations, but the final representation of the staves is normally a collection of straight lines. This is inaccurate since it fails to record the imperfections.

We have discovered that the accuracy of such algorithms can be dramatically improved with the addition of a post-processing step. Working from left to right, the post-processing step (known as *wobble*) attempts to follow a staff line by taking slithers that are one pixel wide. Based on the previous slither of staff line detected, the next slither is searched for in a nearby vertical location. The slither found can be slightly higher or lower than the previous slither, hence the ability to follow a staff line that wobbles (or bows or bends). The process is shown in Figure 1.

More precisely, the starting point for a new slither is the *top* and *bottom* of the previous staff line slither (Figure 2), which is then widened by one pixel both above and below; this is done to allow the next slither of staff line to deviate away from the previous position. Next, all the slithers that at least start between the new values of *top* and *bottom* are found, and the centre of the slither that is closest to the centre of the previous slither is chosen. Since it is possible that the slither found is due to an object passing through the staff line at that point, the final step is to check that the chosen slither is short enough to be only a staff line. If this is not the case, then the previous values for *top* and *bottom* are used as the extremities of the new slither.

To evaluate the effect of this new post-processing step, 15 A4 pieces of music were selected from a diverse range of sources (see Bainbridge (3)), scanned, and their staff lines extracted by hand. For the purposes of testing, these hand-edited staff lines are referred to as the “ideal” staff lines. Table 1 compares a straight line approximation with the *wobble* process. A high value in the “missed” column indicates an algorithm that does not always follow the ideal staff lines. A high value in the “extra” column is

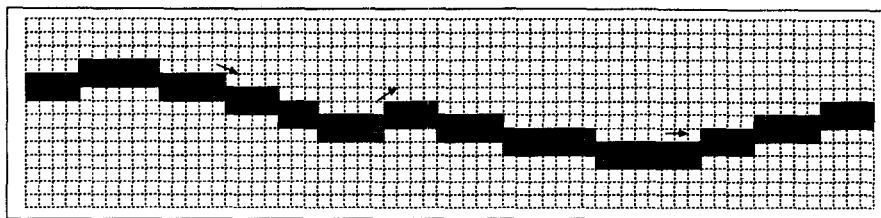


Figure 1: Overview of the *wobble* process.

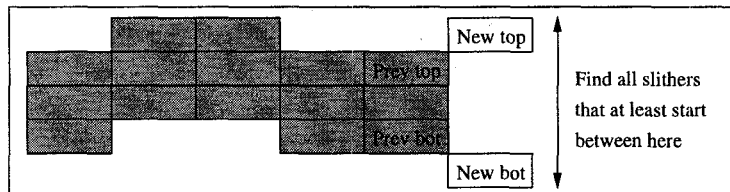


Figure 2: How the limits for potential new slithers are based on the previous slither of staff line. The gray pixels represent a staff line.

caused by an algorithm mistaking other parts of the image as the ideal staff line. For the 15 test images, the accuracy of staff line identification rose from 50% to 94% when the *wobble* post-processing step was included.

Even with hand-edited staff lines it is impossible to generate an image that represents the true staff lines since superimposed objects obscure the upper and lower limits of the line, requiring “educated guesses.” Extreme occurrences of this are when beams fall on a section of staff line, or long slurs of low curvature cross a staff line. Consequently, a comparison of the hand-generated “ideal” staff lines with staff lines generated by the computer will always include a residual difference.

## OBJECT LOCATION

Once the staff lines have been identified, the superimposed musical objects must be located. The most popular approach in OMR systems is to *remove* the staff lines. In this approach, pixels on the staff line are deleted only if there is no evidence that they are a part of an object that continues above or below the expected position of the line.

The *de facto* test for this was first described by Clarke *et al.* (4). Traversing the length of each staff line (say from left to right), the template shown in Figure 3 is used to determine if there is an object superimposed on the staff line at any given point. For a vertical slither of staff line starting at pixel A, if pixel B is black and at least one of the pixels X, Y or Z is black, then this is deemed sufficient evidence of an object, and the slither of staff line remains. A mirror image of the template is used for the bottom half of the staff line.

Even with this careful staff line removal algorithm, an object can become fragmented when a thin part of its shape,

typically a line, blends tangentially into a staff line. Common examples where this occurs are minim note heads and bass clefs.

An algorithm described by Martin and Bellissant (5) addresses this deficiency. It is similar to that described by Clarke *et al.* except that a more intricate computation is used to decide when to remove a slither of staff line. Instead of using a template to decide if an object passes through a particular point on a staff line, a set of *chords* are taken. Here the term chord is used in its mathematical sense, to mean a straight line joining the ends of an arc. The terminology is slightly inaccurate, since we are not strictly dealing with circles, but with the irregular shapes that cross staff lines. Nevertheless, the intention is the same, since the algorithm computes the shortest distance between two points on the perimeter of the shape. An example of this mathematical type of chord is shown in Figure 4.

In the algorithm, chords are computed over the range  $[-90^\circ, +90^\circ]$ , and assembled into a histogram based on angle. A histogram with one distinct peak lying at approximately  $0^\circ$  is caused by a point on the staff line that has no object passing through it, and therefore, can be safely removed. A histogram with two or more distinct peaks is the result of an object passing through, or nearby, that point on the staff line, and is consequently left in the image.

Although Martin and Bellissant report reduced fragmentation, some did still occur; however, one can see how this trend of increasing the sophistication of the superimposed object test can be extended to deal with more forms of fragmentation. Unfortunately the price to pay for reduced fragmentation is increased computational time.

Rather than remove the staff lines, we have developed an

TABLE 1 - Comparison of staff line identification algorithms.

Algorithm	Number of ideal pixels	Missed	Extra	Total number of incorrect pixels	Error
Straight line approximation	3827438	1832010	66387	1898397	49.60%
Post-processed by <i>wobble</i>	3827438	89605	57791	14739	3.85%

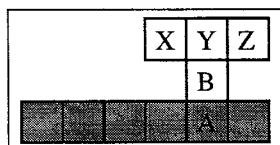
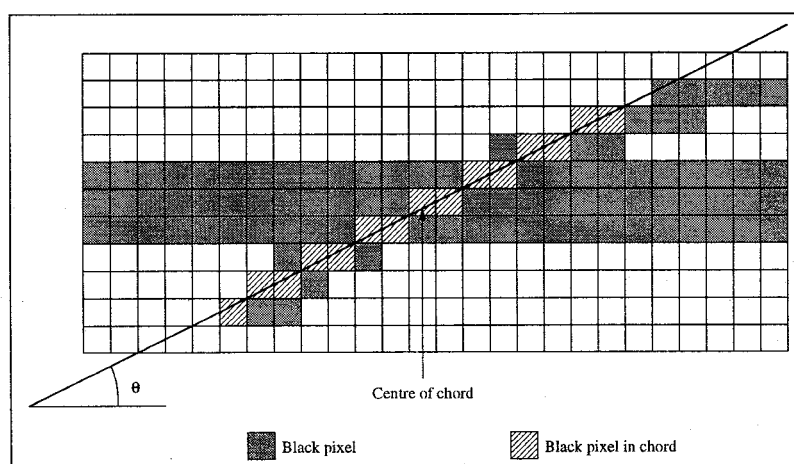
Figure 3: The template used by Clarke *et al.* for removing staff lines. The gray pixels represent the top edge of a staff line.

Figure 4: An example (mathematical) chord.

alternative, faster algorithm that follows the objects that *cross* the staff lines. The algorithm starts by tracking between staff lines to find shapes. When one is found, a modified flood-fill algorithm that crosses over staff lines is used to locate the entire object. When the algorithm reaches a staff line, the surrounding pixels are checked to determine whether or not the object exists on the other side. If it does, the flood-fill procedure is directed to continue on the other side of the staff line. The decision to cross a staff line is a reshaped form of the heuristic used in the removal algorithm to decide if a slither of staff line should be deleted, hence an equivalent staff line crossing algorithm exists for any removal based algorithm.

To evaluate the performance of this new algorithm, the same 15 pieces of music as before were processed to locate the superimposed musical objects. The results are shown in Figure 5. For the simple template test, the new algorithm is only slightly faster; however, for the more sophisticated, but computationally expensive, chord based check the algorithm is four times faster. This is because the crossing based algorithm only needs to apply the test for superimposed objects when an object

comes near a staff line, unlike the removal based version which applies the test at every point on the line. When an expensive test is used, the saving in time is significant.

### PRIMITIVE DETECTION

Once the staff lines have been removed or effectively ignored by crossing over them, the black pixels left should be a part of musical objects. The next stage in the OMR process is to detect the primitive shapes—such as note heads and note stems—that exist in these isolated shapes. In previous projects a variety of pattern recognition techniques have been used to solve this problem: template matching, projections, and morphology to name a few (2). In keeping with our design requirement of extensibility, we make no commitment to one particular technique, but instead we offer a collection of pattern recognition routines available through a specially designed programming language for primitive detection called PRIMELA. A PRIMELA description is written for each type of primitive shape to be recognised with the pattern recognition routine most appropriate for the task selected.



TABLE 2 - Overall accuracy rate for the core CMN primitives in the 11 test images.

Primitive	Number of primitives	Number of primitives missed	Number of shapes incorrectly identified as primitive	Overall accuracy rate
Treble clef	67	1	0	98.51%
Bass clef curl	39	1	0	97.44%
Alto clef curl	8	0	0	100.00%
Time signature	23	0	0	100.00%
Sharp	219	13	0	94.06%
Flat	92	3	2	94.57%
Natural	57	10	0	82.46%
Double sharp	2	0	0	100.00%
Tail up	215	3	4	96.74%
Tail down	131	6	0	95.42%
Vertical line	3224	37	14	98.42%
Beam	791	63	7	91.15%
Filled-in note head	2830	2	21	99.19%
Hollow note head	174	32	7	77.59%
Semibreve note head	2	0	0	100.00%
Rectangle rest	119	1	0	99.16%
Crotchet rest	80	9	0	88.75%
Quaver rest	281	4	6	96.44%
Semiquaver rest	14	0	0	100.00%
Dot	334	12	10	93.41%
Slur	244	14	0	94.26%
Total/Weighted average	8946	211	71	96.85%

symbols that followed the rules for music engraving. The reality is that the images are bent and distorted, have poorly formed symbols, and symbols that touch each other when they should not. Because the symbols are superimposed on a staff line, these problems are especially challenging, particularly when a symbol is fragmented or two symbols overlap when they should not.

By careful removal or avoidance of staff lines, and allowing for them to deviate from a straight line, the "damage" done to symbols by staff line removal can be reduced. The next stage, matching the objects, is then designed to allow for the kinds of imperfections than can be expected both in the original image and after staff line removal, resulting in low error rates, which are necessary to make OMR a feasible technology.

## REFERENCES

1. Bainbridge, D. and Bell, T., 1996, Proc. of the Nineteenth Australasian Computer Science Conf. (Melbourne) 308-317.
2. Blostein, D. and Baird, H. S., 1992, Structured Document Image Analysis (Baird, H. S., Bunke, H., and Yamamoto, K., eds.) Springer-Verlag, Berlin, 405-434.
3. Bainbridge, D., 1997, "Extensible Optical Music Recognition," PhD thesis, Department of Computer Science, University of Canterbury, Christchurch, NZ.
4. Clarke, A. T., Brown, B. M., and Thorne, M. P., 1988, Proc. of the Computers in Music Research Conf. (Lancaster, UK) 84-87.
5. Martin, P. and Bellissant, C., 1991, Proc. of First International Conf. on Document Analysis, 1 (Saint-Malo, France) 417-425.
6. Heussenstamm, G., 1987, "The Norton Manual of Music Notation," W. W. Norton, New York.