Optical Music Recognition with Wavelets

Susan E. George
University of South Australia, Australia

INTRODUCTION: SUPER-IMPOSED OBJECTS

The aim of optical music recognition (OMR) is to "recognise" images of music notation and capture the "meaning" of the music. When OMR is successful it will be able to automatically extract a logical representation of printed or handwritten music captured in an image. There are a variety of reasons why OMR is required. Chiefly, it is convenient for swift input of music notation and might be subsequently edited, performed, used as a search or other. There are many stages before that final high-level interpretation can be made and recognition of the primitive symbols contained in the notation is primary. One of the biggest challenges in OMR is the super-imposition of music notation symbols – notes and other – upon stave lines in the music image. This article examines a general-purpose knowledge-free method in the wavelet transform, to deal with super-imposition in images of typeset music.

Super-imposition arises when notes and other music symbols are placed upon the stave line (or staff), making "ink" of the symbol overlap with "ink" of the stave line. There are various reasons why isolating the super-imposed object is so difficult within OMR. Firstly, image capture may have introduced perturbations; stave lines in the image are rarely parallel, horizontal, equidistant, of constant thickness, or even straight. Secondly, other symbols within the music (such as beams, phrase marks and others) can be mistaken for stave lines and hence lines mis-located. Thirdly, there is only one piece of "ink" for objects that are super-imposed upon the lines and stave lines have to be extracted, leaving the symbol intact — or conversely the symbol has to be segmented from the stave line, having identified its location within the stave.

The OMR field has taken two basic approaches to dealing with super-imposed objects. These approaches are (i) the removal of stave lines and (ii) the recognition/segmentation of music symbols. Neither of these methods has met with complete success. When focusing upon the segmentation, difficulties specific to music notation arise, including (i) the variant size of symbols (e.g., phrase markings, slurs) rendering template matching inappropriate and (ii) the various ways of typesetting a particular musical sound are potentially infinite, and again template matching would not suffice for all the possibilities.

Since the 1960s there have been various approaches to dealing with super-imposed objects. All of these approaches are knowledge-based techniques, in that they are assuming some information is known about the format of music images in order to locate the stave lines or isolate symbols. Blostein and Baird (Blostein & Baird, 1992) present a critical survey of problems and approaches to music image analysis. Here we summarise some of the approaches to OMR from the initial attempts in the 1960s.

- Pruslin (1967) - remove thin horizontal lines (by thresholding the length of continuous vertical runs of pixels, assuming that a figure for "line thickness" was known)
- Prerau (1975) – remove stave lines and restore those parts that coincide with symbols (using a contour trace from the edge of each stave line to locate the start of symbols that might be placed on those lines).
- Nakamura et al. (1979) – remove stave lines using line tracker, calculating a least-squares fit for stave lines computing threshold to erase the stave lines and achieve segmentation.
- Andronico and Ciampa (1982) - remove just the exposed stave lines.
- Aoyama and Tojo (1982) – detect stave lines by using the horizontal histogram of black pixels within each block of low resolution scan lines and using a threshold to find sets of five peaks. Coarse segmentation was then undertaken, removing obvious, exposed sections of stave line by examining the vertical run-lengths involved. Fine segmentation detected black and white noteheads by searching for overlapping pixel runs.
- Mahoney (1982) - isolate symbols from stave lines, interpolating between end-points where there were gaps.
- Matsushima et al. (1985) – detect stave lines by a short bar-like filter that operated down equi-spaced columns in the image, simultaneously with the scanning process.
- Roach and Tatem (1988) – detect stave lines with a line-tracking algorithm calculating line direction and thickness at each pixel in a grey-scale image by

passing a window of appropriate proportions over the image. They have also considered the specific challenges of handwritten music.

- Brainbridge and Bell (1997) – determine presence of stave lines by following the wobble in groups of pixels and remove with heuristics.
- Lin and Bell (Lin et al., 2000) - propose colour printing and scanning technology to help deal with the issue of super-imposed objects.
- Choudhury, DiLauro et al. (2001) – remove staff lines using projection profiles, remove text by heuristics, vertical components of stems and barlines and horizontal components of note heads, recognising the remaining symbols with K-NN classifier combing the glyphs again.
- Droettboom et al. (Droettboom et al., 2002) - describe the use of classifiers for symbols recognition, providing tools for the creation of a simple heuristic classifier, a template-based image matching and a k-nearest neighbour learning classifier.

All these approaches have met with some success and various limitations. Perhaps the biggest limitation is that they are knowledge-based, requiring specific knowledge about music to isolate the symbols, or follow stave lines. Implicitly, knowledge-based techniques are constrained by the knowledge that they may draw upon. For example, a symbol finding algorithm can only find symbols that it knows about, and should a new symbol be encountered within the music notation, the recognition is likely to fail. Similarly, a stave line tracking algorithm is constrained by heuristics about how stave lines generally appear, or in what ways they may be corrupted. When knowledge is explicitly utilised, at some point, a novel input will be encountered that falls outside the scope of the knowledge-based method (this is especially the case when dealing not with "printed" documents, but handwritten ones). Hence the importance and appeal of general-purpose knowledge-free methods.

BACKGROUND: WAVELETS

Wavelets are a relatively recent mathematical approach extending some of the principles of Fourier analysis for the study of periodic signals, decomposing a signal into its component parts. Wavelet theory is well suited for complex signals, where the frequency structure changes throughout the signal (i.e., non-periodic signals). Since very few signals or images are truly periodic, wavelet techniques are ideal. Wavelets permit the signal to be viewed so that the large-scale fluctuations are emphasised (with the small detail as noise), or such that the small fluctuations are emphasised (with the larger scale fluctuations as background). Interest in wavelets is also stimulated by the fact that some wavelets may be implemented in an extremely computationally efficient manner.

The wavelet is defined by the "mother wavelet" from which other wavelets in the "family" can be generated. All members of a particular family share the same basic wavelet shape that is shifted or dilated (*i.e.*, they are made tall and skinny or short and fat). The specific parameters are translation ('b') and contraction ('a'). Members of an individual wavelet family y ^{a,b} (x) can be defined by (1):

$$\psi^{a,b}(x) = |a|^{-1/2}\,\psi\left(\frac{x-b}{a}\right) \qquad (1)$$

Equation 1 generates a one-dimensional function that can be plotted. There are a number of families of wavelets including Harr, Daubechies, Coifmanns, Mexican Hat, Meyer and Morlet. The coefficients in the wavelet transform are an important part of the wavelet expression since (combined with the generating mother wavelet) they compact the signal and can be used to approximate a function. Instead of having to store every value of the function, it is only necessary to store the coefficients, perhaps at various levels of decomposition, and from these the signal can be obtained. Two-dimensional wavelets can be applied to two-dimensional signals; that is, the wavelets are applied to images.

In terms of general image processing, wavelets decompose images into their high and low pass components in row and column-wise directions, thus filtering an image into its component parts. There are four combinations of low pass filters (H1 and H2) and high pass filters (G1 and G2) in the vertical (H1 and G1) and horizontal (H2 and G2) directions. These decompositions are typically represented in quadrant diagrams where the top left quadrant contains an approximation of the original image, the top right contains the horizontal components, the bottom left the vertical components and the bottom right the diagonal components.

When considering wavelet image decomposition, it is worth observing that the music image possesses certain properties, including: (i) the music stave lines – which present a strong horizontal component to the image signal and which are (generally) regular across the image page, (ii) other music notation symbols - which present a more vertical component and are more local perturbations within the image; wavelets are particularly useful for such local irregularities within an image, and some wavelets may be more suitable than others for the regular/irregular patterns that occur within a music image.