VOICE SEPARATION IN POLYPHONIC MUSIC: A DATA-DRIVEN APPROACH

Anna Jordanous
Music Informatics
University of Sussex

ABSTRACT

Much polyphonic music is constructed from several melodic lines - known as voices - woven together. Identifying these constituent voices is useful for musicological analysis and music information retrieval; however, this voice-identification process is time-consuming for humans to carry out. Computational solutions have been proposed which automate voice segregation, but these rely heavily on human musical knowledge being encoded into the system. In this paper, a system is presented which is able to learn how to separate such polyphonic music into its individual parts. This system uses a training corpus of several similar pieces of music, in symbolic format (MIDI). It examines the note pitches in the training examples to make observations about the voice structures. Quantitative evaluation was carried out using 3-fold validation, a standard data mining evaluation method. This system offers a solution to this complex problem, with a 12% improvement in performance compared to a baseline algorithm. It achieves an equal standard of performance to heuristic-based systems using simple statistical observations: demonstrating the power of applying data-driven techniques to the voice separation problem.

1. INTRODUCTION

A common compositional device in music is to construct a piece by interweaving several melodic lines. In musicology, these melodic lines are often referred to as voices [3]. [1] Each voice can be considered independently as a melodic pattern which is complete and interesting in its own right. Several related voices, combined together to form one piece of polyphonic music, can generate additional harmonic qualities to enhance the voices.

Fugues provide a perfect example of this compositional technique in action, being constructed solely of a number of different melodic voices. J. S. Bach was a fundamentally important composer in the history of fugue composition; in particular his highly influential work *The Well-Tempered Clavier* comprises 48 fugues.

In analysis of music such as Bach fugues, the musicologist identifies individual voices to facilitate more advanced analysis of the melodic content such as finding thematic patterns common to different voices. The musical score usually gives the musicologist much help in identifying each voice, as each voice is notated slightly differently (the direction of the note stems often indicates which voice each note belongs to).

Identifying each voice is a considerably harder task if these notational clues are not present. In such cases the musicologist needs to examine musical detail within the piece, such as the note pitches and rhythms [7, 12], using this in conjunction with their knowledge of voice structure in that compositional style. Voice identification can often be a painstaking and time-consuming process.

Can a computer learn how to extract the constituent voices from a piece of music by similar examination of musical detail? Given minimal human assistance and a training set of similar music with the voices already identified, I propose that patterns of voice movement can be identified and learnt. Such patterns represent learned knowledge of compositional style that can assist the computer in identifying individual voices in other music of a similar style. Computers should complete this task markedly faster than a human, due to processing speeds available.

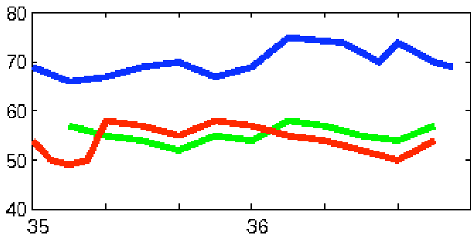
2. RELATED WORK

In recent years a number of different solutions have been proposed for the task of voice segregation [2, 4, 8, 9, 10, 11]. Prior to 2004, voice identification was considered supplementary to the primary task of transcription to notation from musical input [2, 9] but has recently become a problem of interest in its own right [3, 4, 8, 10, 11].

Previous work has imposed human musical knowledge on the system in the form of rules and heuristics [2, 4, 8, 9, 11], rather than enabling the system to learn how voices are structured. In other words, the program is told exactly how to solve the voice segregation problem rather than allowing it to learn how to piece together the voices. These pieces of work use perceptually-motivated rules [7, 12] and have been successful to a certain degree; however, the computer is not learning these rules and developing its own knowledge, but merely utilising the knowledge provided by human investigation. Higher-level voice-leading principles provide a heuristical guide to typical routes that voices take throughout the course of the piece. For extreme styles of music, these principles may fail, particularly in more contemporary music that challenges the rules

[1] This terminology is not used solely in vocal music, as voices (in this context) are also used in instrumental music.

(a) Bars 35-36 of Fugue no. 6 in Dm BWV 851



(b) How the voices move in these two bars

Figure 1. In Fugue no. 6 in Dm, bar 35-6: the middle and lower voices cross even though all voices are present.

of classical harmony and structure.

Kirlin and Utgoff [10] provide the sole prior example of using machine learning to tackle this problem. Their system, *VoiSe*, uses very limited training data, only training on carefully selected sections of one piece (between 4-8 bars). It is unclear how *VoiSe* is able to generalise over a particular genre or composer's style.

Many systems to date [8, 9, 10, 11], have attempted to tackle the voice segregation problem by considering the entire musical score, in a linear fashion. Chew and Wu's reductionist approach [4] provides an alternative. Their voice separation method uses a *contig* approach taken from computational biology techniques, which identifies points where a number of different voice fragments are present (contigs) and uses these contigs to gradually piece the voices together. One heuristic is key to their approach:

> "Because voices tend not to cross, when all voices are present, one can be certain of the voice ordering and assignment." [4] (p. 4)

This heuristic is central to the success of the contig method, but it is flawed: examples exist where all voices are present and do cross (Figure 1). In such cases, Chew and Wu's method cannot be completely accurate. However their general approach is worth further investigation.

3. VOICE SEGREGATION MODEL

3.1. Guiding principles governing this solution

This paper presents an artificially intelligent system inspired by human attempts to solve the voice segregation problem, but not *controlled* by human knowledge. The system learns to identify voices using statistical analysis of the voice structuring of other similar pieces of music. This approach is inspired by how a musicologist examines the structure of fugal voices in Bach's work in their musical education, to learn about Bach's voice-writing style.

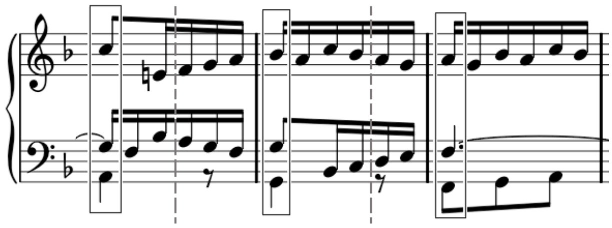


Figure 2. The system in action. *Marker points* are highlighted with boxes, and the dotted line divides the notes into *windows* centred around these marker points.

Breaking the piece of music down into smaller sections is sensible. Inspired by Chew and Wu [4], the system looks for areas where the voice structure is more obvious. It then works outwards from those local points, to piece together the route that each voice takes through the piece.

3.2. Implementation Details

The system was implemented in Matlab, making use of the MIDI toolbox for Matlab [6] to process MIDI files containing the training and test corpora. [2] A MIDI file is returned by the system such that the MIDI channel marks the voice that each note belongs to. The lowest voice is in channel 1, the second lowest in channel 2 and so on. The system learns from a training corpora of music files, examining how likely each possible MIDI pitch is to occur for each voice, and how likely each transition between pitches is to occur.

The voice identification algorithm for a given piece is:

PRE-PROCESSING: Transpose the piece into the normalised key of C so that the training data is not skewed harmonically

STEP 1: Find marker points: points in time where each voice is present and the pitches are distributed far apart

STEP 2: For each marker point: define windows centred around each marker point, which extend out to meet halfway between each marker point (see Figure 2)

STEP 3: For each window: work outwards from the marker to the window edges. Allocate each note x to a voice v using the probabilities learnt in training to maximise the cost function:

$$max_v[P(V(x) = v|V(n) = v) + (0.5 * P(V(x) = v))] \quad (1)$$

where $V(x)$ is the voice allocated to note x and note n is the previous note in voice v.

If at any point, more than one synchronous note is allocated to one voice, give priority to the highest scoring note

STEP 4: Similarly allocate voices to the notes from the first marker, backwards, to the beginning of the piece

STEP 5: Similarly allocate voices to the notes from the last marker, forwards, to the end of the piece

POST-PROCESSING: Transpose the fugue back to its original key (i.e. reverse the pre-processing step)

[2] All MIDI files were sourced from the Humdrum database, available at http://kern.humdrum.net (accessed January 2008)

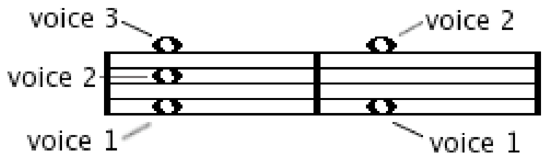


Figure 3. The baseline algorithm used for evaluation.

4. EVALUATION

Quantitative evaluation was carried out using 3-fold validation [13]. [3] Performance was measured using standard information retrieval statistics: precision (the percentage of notes allocated to a voice that correctly belong to that voice), recall (the percentage of notes in the voice that are successfully allocated to that voice) and F-measure (which reflects a balance of precision and recall). [4]

The system was compared to a baseline algorithm that used pitch ordering for allocation: at each timepoint, it allocated the lowest note to the lowest voice, the next lowest note to the next lowest voice and so on. If less notes are sounding than voices (i.e. one of the voices is silent), then it allocated voices from the bottom voice up, with upper voices unallocated (silent). Figure 3 shows this.

J. S. Bach's famous collection of fugues, *The Well Tempered Clavier*, supplied the corpus for the first experimentation. All 26 three-voice fugues were tested (see Table 1), as were all 19 four-voice fugues (Table 2) from this set.

Method	Voice	Precision	Recall	F-measure
My system	v3	90.53%	88.84%	89.48%
My system	v2	82.92%	83.40%	83.01%
My system	v1	92.44%	92.80%	92.44%
Baseline	v3	97.27%	63.28%	76.01%
Baseline	v2	67.62%	74.89%	70.80%
Baseline	v1	80.09%	99.15%	88.37%

Table 1. Voice identification: Bach three-voice fugues. Voice 3 is the highest voice and voice 1 is the lowest voice.

An interesting but under-explored avenue in previous work is how systems perform on other styles of music. So the second set of training data consisted of all 17 string quartets composed by Beethoven (see Table 3). [5]

Performance was compared to related systems [4, 8, 10, 11] where possible (see Table 4). In [10, 11] soundness and completeness scores are presented, correlating to precision and recall, respectively. [8] give precision scores only. The average voice consistency measures in [4] were used as completeness/recall scores (as in [11]).

Method	Voice	Precision	Recall	F-measure
My system	v4	79.85%	80.35%	79.73%
My system	v3	69.90%	67.45%	68.44%
My system	v2	69.31%	70.08%	69.35%
My system	v1	81.23%	83.04%	80.80%
Baseline	v4	94.97%	40.30%	54.92%
Baseline	v3	52.48%	49.66%	50.89%
Baseline	v2	52.99%	66.26%	58.42%
Baseline	v1	70.58%	99.43%	81.47%

Table 2. Voice identification in Bach four-voice fugues. Again voice 1 is the lowest voice, voice 4 the highest.

Method	Voice	Precision	Recall	F-measure
My system	violin1	79.84%	69.47 %	71.86%
My system	violin2	59.79%	57.91%	58.68%
My system	viola	60.86%	59.38%	60.00%
My system	cello	71.55%	72.02%	71.70%
Baseline	violin1	80.08%	51.41%	62.07%
Baseline	violin2	64.39%	57.03%	60.29%
Baseline	viola	63.52%	67.07%	65.06%
Baseline	cello	66.91%	90.68%	76.54%

Table 3. Voice identification: Beethoven String Quartets.

System	Precision	Recall	F-measure
This study	**80.88%**	**80.85%**	**80.86%**
Chew & Wu	n/a	88.98%	n/a
Kirlin & Utgoff *	88.65%	65.57%	75.38%
Madsen & Widmer	95.94%	70.11%	81.02%
Karydis et al	93.19%	n/a	n/a

Table 4. Comparison of average performance between my system and previous work (on Bach fugues). * Kirlin and Utgoff's system [10] was the only system not to be tested on Bach fugues, but on sections of Bach's Ciaccona.

5. DISCUSSION OF RESULTS

My system performed noticeably better than the baseline algorithm [6] at identifying each voice in Bach's fugues, averaging 80.5% F-measure compared to the baseline's 68.7%. Reasonable F-measures (average 64.4%) were also recorded when identifying each instrumental part in the Beethoven string quartets.

The system performed more strongly on the Bach fugues than on the Beethoven music. This is likely to be due to the greater variety in compositional style between differ-

[3] In 3-fold validation, the training corpus is divided into three sets. Over three training runs, two sets are used for training and one for testing, till all three sets have been used as test data.

[4] Evaluation of voice identification systems is non-trivial and worthy of discussion in its own right at greater length; however for ease of comparison between different systems I adopt this strategy for now.

[5] String quartet music occasionally requires instruments to play more than one note simultaneously; however in general each piece in the corpus was separable into almost entirely monophonic voicings.

[6] The design of the baseline algorithm clearly favoured recall in lower voices and precision in the higher voices. However the average F-measure score reflects a balance of overall precision and recall.

ent Beethoven string quartet compositions, compared to Bach fugues which have a more uniform style. Voice separation in Beethoven string quartets was expected to be a harder task than for Bach fugues. Evaluation showed this to be the case, with lower F-measure, precision and recall scores. [7] With average F-measure scores of over 60% for voice segregation in the Beethoven corpus, though, it is pleasing to see some potential in how the system generalises to work on music other than Bach fugues; an under-explored aspect in previous work.

The functionality of this system requires the existence of a set of relevant training examples. This is synonymous with a human needing experience of similar music before attempting to identify the voices in a new piece of music. If pieces by that composer, of a similar genre, do not exist for training, then more general training examples can be supplied to the system and the system will still be able to make a reasonable attempt at voice segregation. To illustrate this, voice separation carried out on Mozart's *Fugue in C minor*, (K. 546: mvt. 2), using Bach's fugues to train on, scored a mean F-measure of 75%.

This voice segregation system matches the standards of previous systems, despite having less human knowledge encoded in its operation. Its learning approach produced similar results to the systems driven by human-devised heuristics [4, 8, 11] (as demonstrated by the F-measure scores in Table 4). Better results were achieved than for *VoiSe* [10], the only other system incorporating learning; though a fair comparison cannot be made until *VoiSe* is tested on a comparable repertoire to the other systems.

6. FUTURE WORK

Reliance on human knowledge, although minimised compare to other systems, is still present in the system presented here. For this system to demonstrate artificial intelligence further, the data mining approach could be increased substantially. More observations from the training data could be incorporated, such as timing information, or observations about the nature of marker points. Ideally the system would identify for itself what is musically important for tracing the route of the voices through the piece.

More complex statistical tools could also be utilised. Currently the system only considers a history of one note previous to the current note, when allocating the current note to a voice. It would be interesting to apply Hidden Markov Models here, so that more of the previously allocated notes can be used to assist in voice allocation.

This paper has focussed exclusively on finding voices exactly as the composer has written them. However the written voicings do not always correspond to the melodic lines that we perceive when listening to music [1, 3, 5]. This system could be used with good effect to detect such higher-level voices, given appropriate training examples.

[7] Even when the training set was restricted to movements of a similar type, there was no noticeable improvement in performance.

7. CONCLUSIONS

Machine learning has provided a solution to the problem of dividing polyphonic music into its individual voices. From a small set of observations from training data, the system presented in this paper can identify the route that a voice takes through a piece. It is able to identify constituent voices of a polyphonic piece of music with good precision and recall; an average F-measure of 80% was recorded for Bach fugues and 64% for Beethoven string quartets (which vary more in style than Bach fugues).

Performance in the fugal voice-separation task was equal to that of more heuristically guided systems [8, 11] and surpassed that of a baseline algorithm which allocated voices purely on relative pitch positioning.

While improvements could be made to this system to enhance the knowledge it gains from data mining and further minimise the human knowledge it uses, this work represents an advance in the application of computational methods to the voice separation problem. It offers an alternative approach to that of encoding human-imposed heuristics and rules. There is much potential for further exploration of artificial intelligence techniques to the voice separation problem and to music analysis in general.

8. ACKNOWLEDGEMENTS

This work has benefitted from comments from Nick Collins, Chris Thornton, Chris Darwin and three reviewers.

9. REFERENCES

[1] A. Bregman. *Auditory Scene Analysis: The Perceptual Organization of Sound*. MIT Press, Cambridge, MA, 1994.

[2] E. Cambouropoulos. From MIDI to traditional musical notation. In *AAAI Workshop on Artificial Intelligence and Music*, USA, 2000.

[3] E. Cambouropoulos. 'Voice' separation: theoretical, perceptual and computational perspectives. In *ICMPC*, Italy, 2006.

[4] E. Chew and X. Wu. Separating Voices in Polyphonic Music: A Contig Mapping Approach. In *Computer Music Modeling and Retrieval: Revised Papers*, Esbjerg, Denmark, 2005.

[5] D. Deutsch. *Grouping mechanisms in music*, pages 299–348. Academic Press, San Diego, USA, 1999.

[6] T. Eerola and P. Toiviainen. *MIDI Toolbox: MATLAB Tools for Music Research*. University of Jyväskylä, Jyväskylä, Finland, 2004. http://www.jyu.fi/musica/miditoolbox (accessed January 2008).

[7] D. Huron. Tone and Voice: A Derivation of the Rules of Voice-Leading from Perceptual Principles. *Music Perception*, 19(1):1–64, 2001.

[8] I. Karydis, A. Nanopoulos, A. Papadopoulos, and E. Cambouropoulos. VISA: The Voice Integration/Segregation Algorithm. In *ISMIR*, Austria, 2007.

[9] J. Kilian and H. Hoos. Voice separation - a local optimisation approach. In *ISMIR*, France, 2002.

[10] P. Kirlin and P. Utgoff. VoiSe: Learning to Segregate Voices in Explicit and Implicit Polyphony. In *ISMIR*, UK, 2005.

[11] S. Madsen and G. Widmer. Separating voices in MIDI. In *ISMIR*, Canada, 2006.

[12] D. Temperley. *The Cognition Of Basic Musical Structures*. MIT Press, Cambridge, MA, 2001.

[13] I. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Elsevier, San Fransisco, USA, 2005.