Visualization in Audio-Based Music Information Retrieval

Matthew Cooper,[*] Jonathan Foote,[*] Elias Pampalk,[†] and George Tzanetakis[††]
[*]FX Palo Alto Laboratory
3400 Hillview Avenue, Building 4
Palo Alto, California 94304 USA
{foote, cooper}@fxpal.com
[†]Austrian Research Institute for Artificial Intelligence (OFAI)
Freyung 6/6 A-1010
Vienna, Austria
elias.pampalk@ofai.at
[††]Department of Computer Science (also Music)
University of Victoria
PO Box 3055, STN CSC
Victoria, British Columbia V8W 3P6 Canada
gtzan@cs.uvic.ca

Music information retrieval (MIR) is an emerging research area that explores how music stored digitally can be effectively organized, searched, retrieved, and browsed. The explosive growth of online music distribution, portable music players, and lowering costs of recording indicate that in the near future, most of the recorded music in human history will be available digitally. MIR is steadily growing as a research area, as can be evidenced by the international conference on music information retrieval (ISMIR) series (soon in its sixth year) and the increasing number of MIR-related publications in *Computer Music Journal* and other journals and conference proceedings.

Designing and developing visualization tools for effectively interacting with large music collections is the main topic of this overview article. Connecting visual information with music and sound has fascinated composers, artists, and painters for a long time. Rapid advances in computer performance have enabled a variety of creative endeavors to connect image and sound, ranging from simple direct renderings of spectrograms popular in software music players to elaborate real-time interactive systems with three-dimensional graphics. Most existing tools and interfaces that use visual representations of audio/music such as audio editors treat audio as a monolithic block of digital samples without any information regarding its content. The systems described in this overview are characterized by the fact that they attempt to visually represent higher-level information about the content of music. MIR is a new field, and visualization for MIR is still in its infancy; therefore we believe that this article provides a comprehensive overview of the current state of the art in this area and will inspire other researchers to contribute new ideas.

Computer Music Journal, 30:2, pp. 42–62, Summer 2006

Background

There has been considerable interest in making music visible. Many artists have attempted to realize the images elicited by sound (Walt Disney's *Fantasia* being an early, well-known example). Another approach is to quantitatively render the time or frequency content of the audio signal, using methods such as the oscillograph and sound spectrograph (Koening, Dunn, and Lacey 1946; Potter, Kopp, and Green 1947). These are intended primarily for scientific or quantitative analysis, although artists like Mary Ellen Bute have used quantitative methods such as the cathode ray oscilloscope toward artistic ends (Moritz 1996). Other visualizations are derived from note-based or score-representations of music, typically MIDI note events (Malinowski 1988; Smith and Williams 1997; Sapp 2001).

The idea of representing sound as a visual object in a two- or three-dimensional space with properties related to the audio content originated in psychoacoustics. By analyzing data collected from user studies, it is possible to construct perceptual spaces that visually show similarity relations between

single notes of different musical instruments (Grey 1975). Using such a *timbre space* as control in computer music and performance was explored by Wessel (1979). This idea has been used in the Intuitive Sound Editing Environment (ISEE), in which nested two- and three-dimensional visual spaces are used to browse instrument sounds as experienced by musicians using MIDI synthesizers and samples (Vertegaal and Bonis 1994). The Sonic Browser is a tool for accessing sounds or collections of sounds using sound spatialization and context-overview visualization techniques where each sound is represented as a visual object (Fernström and Brazil 2001). Another approach is to visualize the low-level perceptual processing of the human auditory system (Slaney 1997). An interesting visualization that combines traditional audio editing waveform representations and pitch-based placement of notes is used in the Melodyne software by Celemony (available online at www.celemony.com/cms/).

The main goal of this article is to provide an overview of visualization techniques developed in the context of music information retrieval for representing polyphonic audio signals. One of the defining characteristics that differentiate the techniques described in this article from most previous work is that the techniques described here use sophisticated analysis algorithms to automatically extract content information from music stored in digital audio format. The extracted information is then rendered visually. Visualization techniques have been used in many scientific domains (e.g., Spence 2001; Fayyad, Grinstein, and Wierse 2002); they tend to take advantage of the strong pattern-recognition abilities of the human visual system to reveal similarities, patterns, and correlations in space and time. Visualization is more suited for areas that are exploratory in nature and where there are large amounts of data to be analyzed. MIR is a good example of such an area. The concept of browsing is also central to the design of interfaces for MIR. Browsing is defined as "an exploratory, information seeking strategy that depends upon serendipity . . . especially appropriate for ill-defined problems and exploring new task domains" (Marchionini 1995).

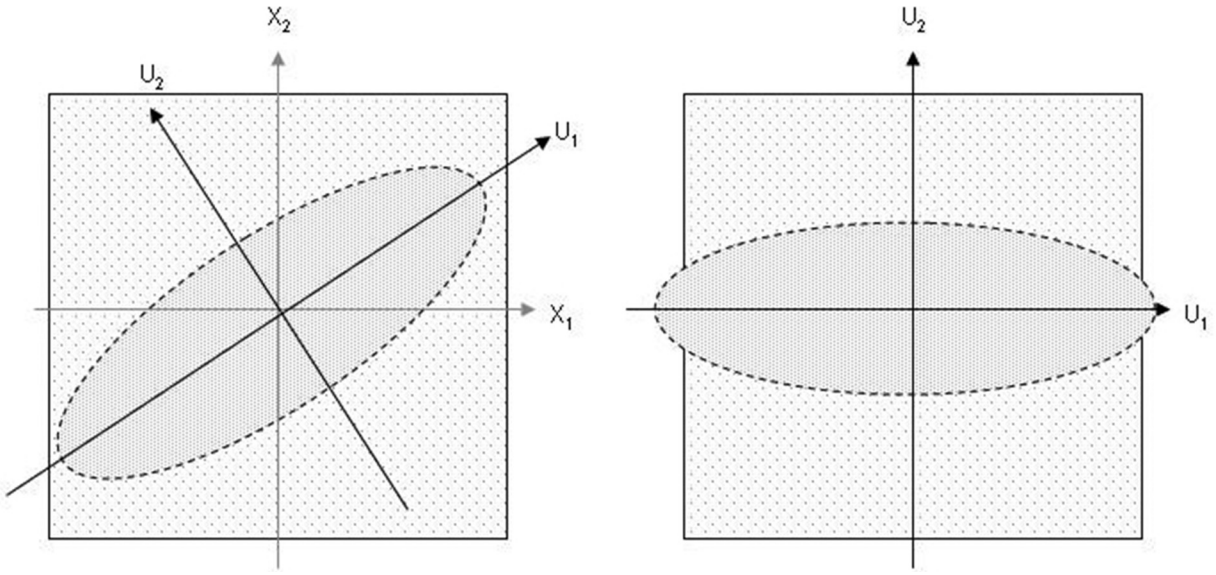
Techniques for visualizing music in the context of MIR can be roughly divided into two major categories: techniques for visualizing a single file or piece of music, and techniques for visualizing collections of pieces. The systems described in this article are representative of the possibilities afforded by MIR visualization.

Parameterizing the Audio

The first step in any visualization of an audio signal is to convert the audio to a parametric window-based feature and typically include Mel-Frequency Cepstral Coefficients (MFCCs), spectral features from the Short-Time Fourier Transform (STFT), or subspace representations from principal component analysis. The window size may be varied, although robust analysis typically requires resolution on the order of 20 Hz, that is, 20 windows per second. For most visualization techniques, the actual parameterization is not crucial as long as "similar" sounds yield similar parameters. Psychoacoustically motivated parameterizations have also been explored. Although this article occasionally describes specific parameterizations, it does so only for completeness and clarity. All the described visualization techniques can use other parameterizations.

In many of the visualization techniques described herein, it is necessary to reduce the dimensionality of the parameterization of the audio signal so that the information can be mapped for example to spatial dimensions or color. Principal components analysis (PCA)—or more generally, the Karhunen-Loeve transform (Jolliffe 1986)—can be used for this purpose. PCA is a dimensionality reduction technique where a high-dimensional set of feature vectors is transformed to a set of feature vectors of lower dimensionality with minimum loss of information. The extraction of a principal component amounts to a variance maximization rotation of the original feature space. In other words, the first principal component is the axis passing through the centroid of the feature vectors that has the maximum variance and therefore explains a large part of the underlying structure of the feature space. The next principal component tries to maximize the variance not explained by the first. In this

Figure 1. The left depicts two-dimensional data concentrated in an ellipse. The right shows the data rotated according to its two principal axes, computed using principal components analysis.



manner, consecutive orthogonal components are extracted.

Figure 1 (left) shows a scatter plot of two-dimensional data points. The data are concentrated in an ellipse that is indicated by a dotted line. PCA rotates the original data axes to maximize the variance accounted for by each dimension in the resulting subspace. In practice, PCA is computed using the singular-value decomposition (SVD; Strang 1988) of the rectangular data matrix. As shown in Figure 1 (right), projection of the data on to the rotated axis U_1 accounts for more variance in the data than projection onto the original axis X_1. Here, U_1 is the direction that accounts for the most variance in the original data and is the singular vector corresponding to the largest singular value. The axis U_2 is the axis orthogonal to U_1 that accounts for as much of the remaining variance in the original data as possible. The subsequent axes in the low-dimensional subspace are calculated similarly.

More specifically, the principal components are linear combinations of the original feature vectors v that can be arranged as columns of a matrix V. To compute the principal components, we first calculate the feature vector covariance matrix C:

$$C(k,l) = \frac{1}{I}\sum_i (V_i(k) - \overline{V}(k))(V_i(l) - \overline{V}(l)) \qquad (1)$$

where $\overline{V} = 1/I\sum_i V_i$, and its singular value decomposition (SVD) is given by

$$C = U\Sigma W^T \qquad (2)$$

Here, U and W are orthogonal matrices, and Σ is a diagonal matrix. The principal components of V are the columns of U, and the corresponding singular values are contained in Σ.

To perform dimensionality reduction from n dimensions to m dimensions, where $m < n$, the principal components corresponding to the m largest singular values are chosen. The collection of pieces of music over which the covariance matrix **C** is calculated is important and provides context-sensitivity for PCA-based visualizations. For example, if the feature vectors from only the specific song to be analyzed are used for the computation of the covariance matrix, the resulting PCA will reflect only the variance of that particular file. On the other hand, if a larger collection of pieces of music is used for the computation of the covariance matrix, the resulting PCA will reflect the variances over the entire collection. Therefore, the same piece of music can have different PCA-based visualizations depending on which feature vectors are used to calculate the covariance matrix.

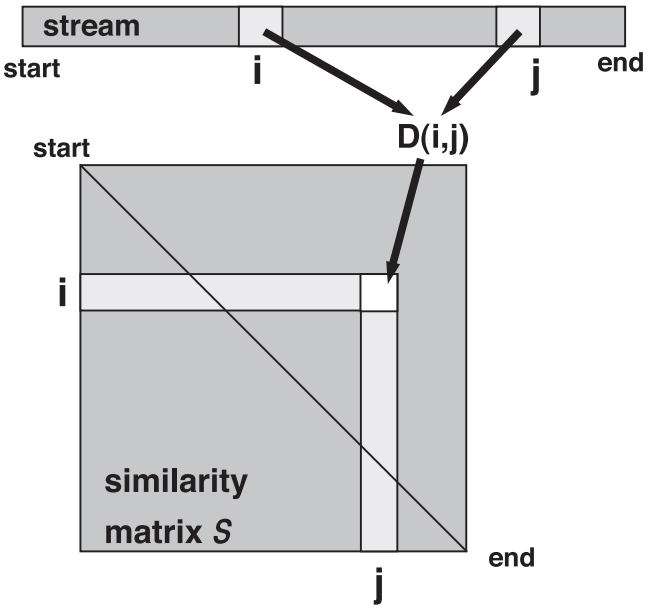


Figure 2. Diagram of the similarity matrix embedding.

Visualizing a Single Musical Piece

Music is a complex human artifact, and there are many possible ways one could try to describe it. The techniques described in this section attempt to visually represent aspects of a piece of music, such as structure, rhythm, self-similarity, and similarity to other pieces and styles.

Similarity Matrix

Musical pieces generally exhibit some degree of coherence or similarity over their full durations. With the exception of more avant-garde compositions, structure and repetition are general features of nearly all music. For example, the coda often resembles the introduction, and the second chorus generally sounds like the first. On a shorter time scale, successive bars are often repetitive, especially in popular music. The similarity matrix is a general method for visualizing musical structure via its acoustic self-similarity across time, rather than by absolute acoustic characteristics.

Similarity-matrix analysis is a technique for studying the global structure of time-ordered media streams (Foote and Cooper 2003). An audio file is visualized as a square, as shown in Figure 2. Time runs from left to right as well as from top to bottom. In the square matrix, the brightness of point (i, j) is proportional to the audio similarity between instants i and j in the source audio file. Similar regions are bright, dissimilar regions are dark. Thus, there is always a bright diagonal line running from top left to bottom right, because each audio instant is maximally similar to itself. Repetitive similarities, such as repeating notes or motifs, show up as checkerboard patterns: a note that occurs twice will give four bright areas at the corner of a square. The two regions at the off-diagonal corners are the "cross-terms" resulting from the first note's similarity to the second. Repeated themes are visible as diagonal lines parallel to—and separated from—the main diagonal by the time difference between repetitions.

The similarity matrix contains the quantitative similarity between all pairwise combinations of audio windows, and different audio parameterizations can be used depending on the application. Represent the B-dimensional feature vector computed for N windows of a digital audio file by the vectors $\{\upsilon_{1,\ldots,}\upsilon_N\} \subset \mathrm{R}^B$. Given a similarity measure $d: R^B \times R^B \rightarrow R$, the resulting similarity data is embedded in a matrix **S** as illustrated in Figure 1. The elements of **S** are

$$\mathbf{S}(i,j) = d(v_i, v_j) \tag{3}$$

Throughout, $\mathbf{S}(i, j)$ denotes the element of the ith row and jth column of the matrix **S.**

A common similarity measure is the cosine distance. Given feature vectors v_i and v_j (representing windows i and j, respectively), then

$$d_{\cos}(v_i, v_j) = \frac{< v_i, v_j >}{\| v_i \| \| v_j \|} \tag{4}$$

This measure is large if the vectors are similarly oriented in the feature space. Normalizing the inner product removes the dependence on magnitude (and hence energy, given spectral features). In practice, we typically zero-mean the data to give the highest range to the distance measure. To build a non-

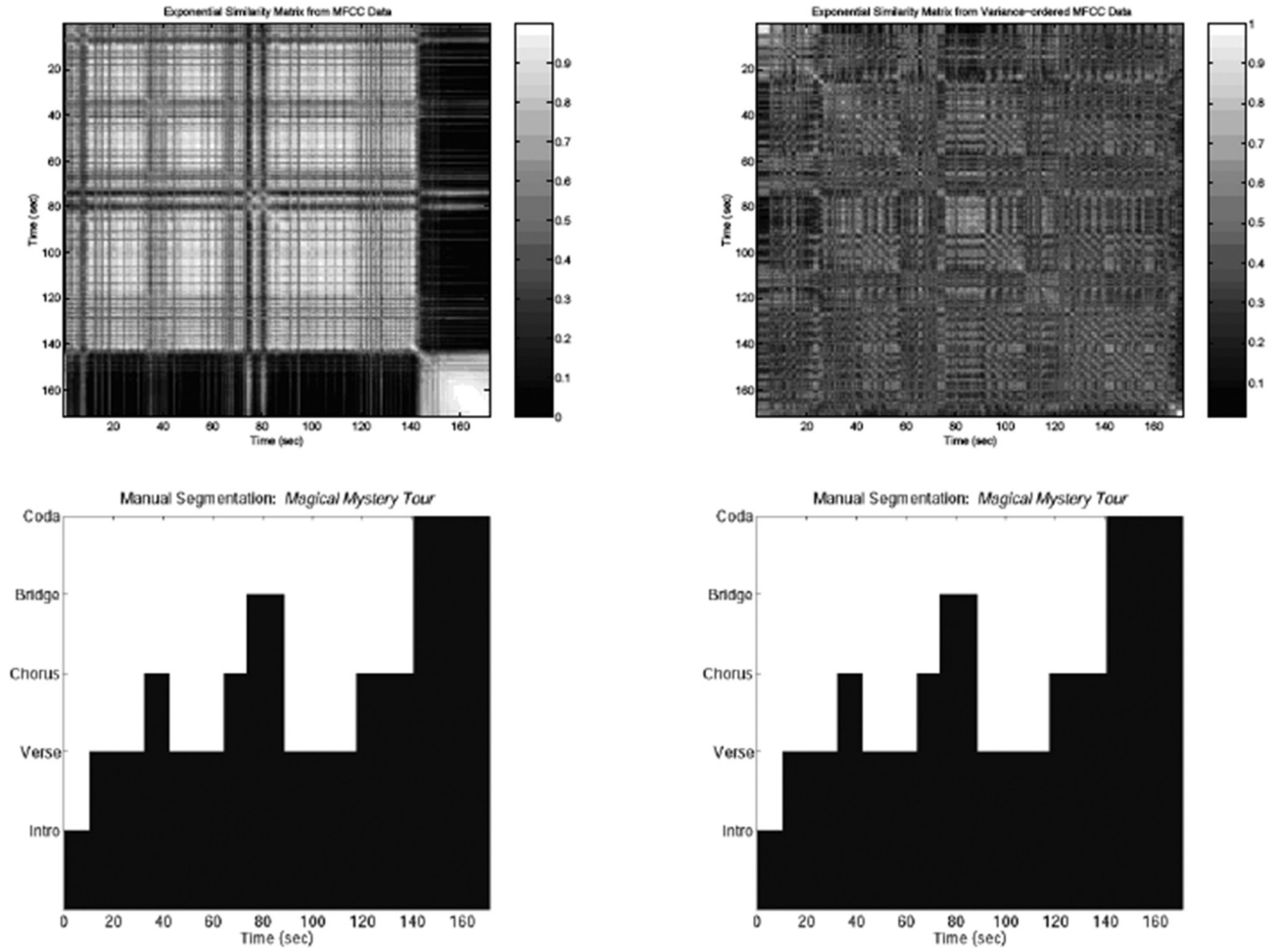


Figure 3. Similarity matrices computed from 45 MFCCs (top left) and the seven MFCCs with greatest variance (top right) after low-pass filtering for The Magical Mystery Tour. *The bottom row shows visualizations of the manual segmentation for reference.*

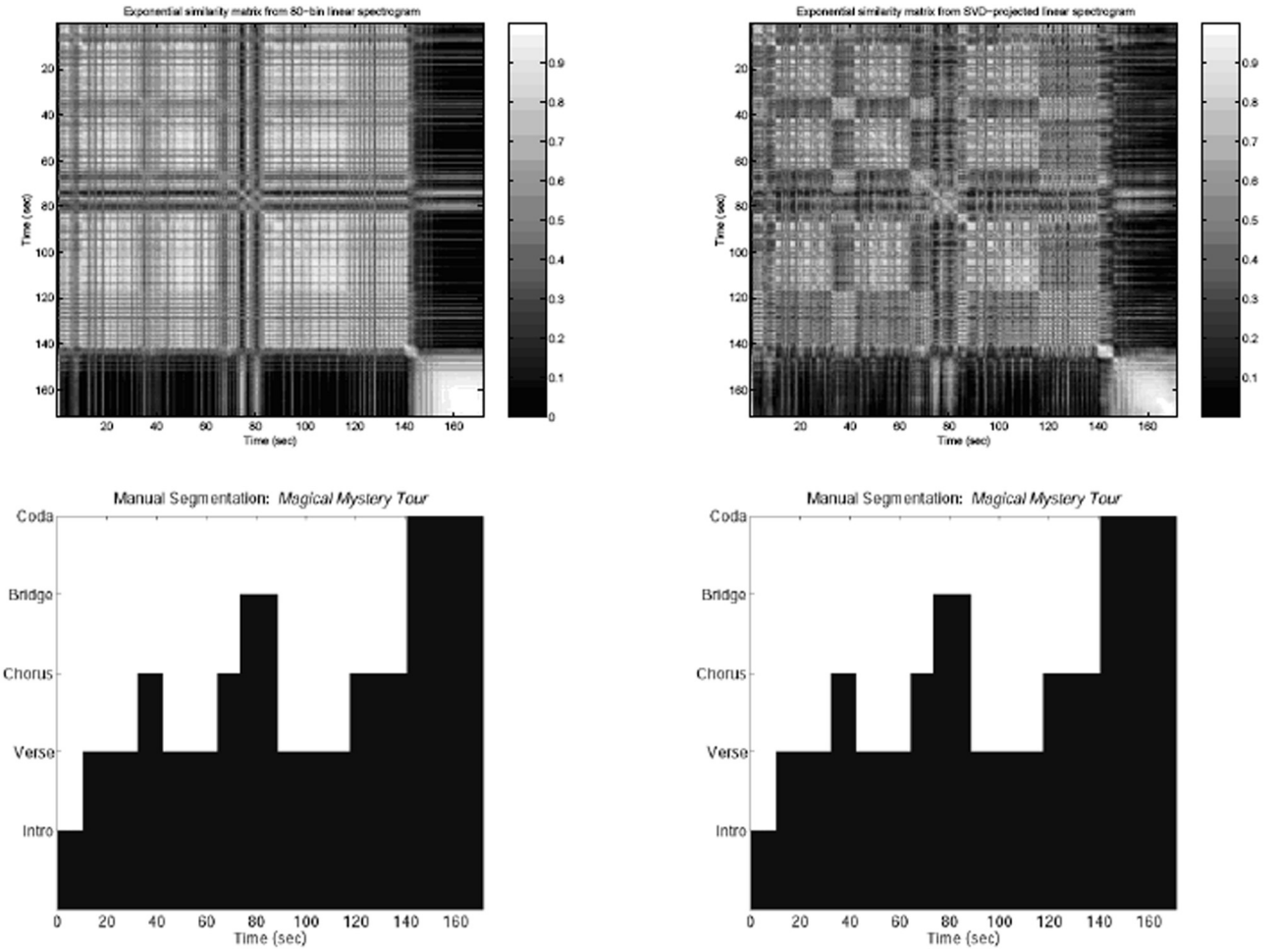
negative matrix, we also employ the following exponential similarity measure:

$$d_{\exp}(v_i, v_j) = \exp\left(\frac{d_{\cos}(v_i, v_j) - 1}{d_{\cos}(v_i, v_j) + 1}\right) \quad (5)$$

To visualize an audio file, an image is constructed so that each pixel at location (i, j) is given a gray value proportional to the similarity measure described above.

We now review a visualization of a popular song. The piece analyzed is *Magical Mystery Tour* by The Beatles. The 22-kHz digital audio file is divided into non-overlapping 1,024-sample windows at 20 Hz. We calculate the 1,024-point magnitude spectrum and 45 MFCCs from each audio frame. The upper-left panel of Figure 3 shows the similarity matrix computed from the MFCC features using Equation 5. The upper-right panel shows the similarity matrix computed from the seven MFCCs with the largest variances. The seven coefficients are normalized to unit variance before calculating the similarity matrix using the similarity measure of Equation 5. The upper-left panel of Figure 4 shows the matrix computed from the full spectrogram data. The upper-right panel shows the matrix computed using the spectrogram data projected into a subspace composed of its first seven principal components and scaled to unit variance. The bottom panels show visualizations of the manual segmenta-

Figure 4. Similarity matrices computed from the full spectrogram data (top left) and the SVD-projected spectrograms (top right) after low-pass filtering for The Magical Mystery Tour. *The bottom row shows visualizations of the manual segmentation for reference.*



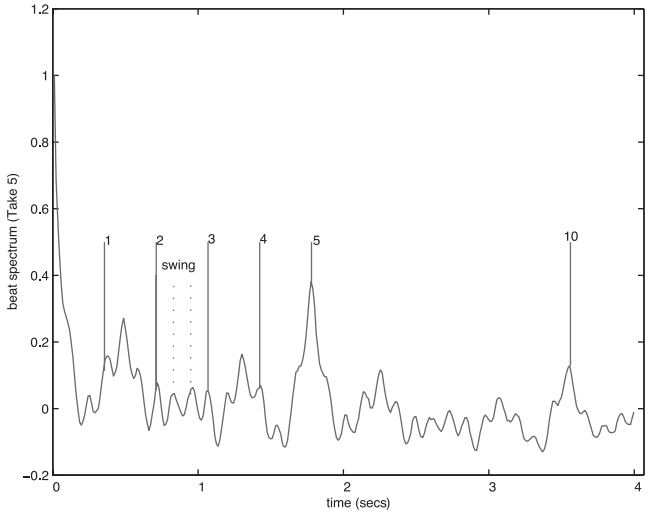
tion of Table 1. The *y*-axis in the visualizations indicates the cluster of each segment; the *x*-axis shows time.

In both Figures 3 and 4, the most visible structure is the song's coda, from 141 sec to 171 sec, which is distinct from the song's verse, chorus, and bridge elements. Its dissimilarity from the rest of the piece is quantified by the dark regions of low cross-similarity in the bottom-most rows and rightmost columns of the similarity matrices. As expected, the reduced-dimension features show improved discrimination in the corresponding similarity matrices. Overall, the PCA-projected spectrogram features provide the best visualization of the piece's structure.

Table 1. Manual Segmentation of *The Magical Mystery Tour*

Segment	*Boundaries (sec)*
Intro	0–10
Verse (Voc.)	11–21
Verse (Voc. and Inst.)	22–32
Chorus	33–42
Verse (Voc.)	43–53
Verse (Voc. and Inst)	54–64
Chorus	65–73
Bridge	74–88
Verse (Voc.)	89–102
Verse (Voc. and Inst.)	103–117
Chorus	118–141
"Outro"	141–171

Figure 5. Beat spectrum of the jazz composition Take Five.



Beat Spectrum and Beat Spectrogram

Both the periodicity and relative strength of rhythmic structure can be derived from the similarity matrix. The term *beat spectrum* is used to describe a measure of self-similarity as a function of the lag (Foote and Uchihashi 2001). Peaks in the beat spectrum at a particular lag l correspond to audio repetitions at that temporal rate. The beat spectrum $B(l)$ can be computed from the similarity matrix using diagonal sums or autocorrelation methods. A simple estimate of the beat spectrum can be found by diagonally adding the similarity matrix **S** as follows:

$$B(l) \approx \sum_{k \subset R} \mathbf{S}(k, k+l) \tag{6}$$

Here, $B(0)$ is simply the sum along the main diagonal over some continuous range R, $B(1)$ is the sum along the first superdiagonal, and so on. A more robust estimate of the beat spectrum is the autocorrelation of S:

$$B(k, l) \approx \sum_{i,j} \mathbf{S}(i, j)\mathbf{S}(i+k, j+l) \tag{7}$$

Because $B(k,l)$ is symmetric, it is only necessary to perform the sum over one variable to yield a one-dimensional result $B(l)$. This approach works surprisingly well for most kinds of musical genres, tempos, and rhythmic structures.

Figure 5 shows the beat spectrum computed from the first ten seconds of Paul Desmond's jazz composition *Take 5,* performed by the Dave Brubeck Quartet. Besides being in an uncommon time signature (5/4), this rhythmically sophisticated work requires some interpretation. First, note that there is no obvious periodicity at the actual beat tempo (denoted by solid vertical lines in the figure). Rather, there is a marked periodicity at five beats and a corresponding sub-harmonic at ten. Jazz aficionados know that "swing" is the subdivision of beats into non-equal periods rather than "straight" (equal) eighth notes. The beat spectrum clearly shows that each beat is subdivided into near-perfect triplets. This is indicated with dotted lines spaced one-third of a beat apart between the second and third beats. A clearer visualization of "swing" would be difficult to achieve by other means.

The beat spectrum can be analyzed to determine tempo and more subtle rhythmic characteristics. Peaks in the beat spectrum give the fundamental rhythmic periodicity (Foote and Uchihashi 2001). Strong off-beats and syncopations can be then deduced from secondary peaks in the beat spectrum. Because the only necessary signal attribute is repetition, this approach is more robust than other approaches that look for absolute acoustic features such as energy peaks.

There is an inverse relationship between the time accuracy and the beat spectral precision. Technically, the beat spectrum is a frequency operator and hence does not commute with a time operator. Thus, beat spectral analysis, like frequency analysis, exhibits a tradeoff between spectral and temporal resolution.

The *beat spectrogram* is used to analyze rhythmic variations over time. Like its namesake, the beat spectrorgram visualizes the beat spectrum over successive windows to show rhythmic variation over time. Time is on the x-axis, with lag time on the y-axis. Each pixel is colored with the scaled value of the beat spectrum at that time and lag, so that peaks are visible as bright horizontal bars at the repetition time. Figure 6 shows the beat spectrogram of a 33-second excerpt of the Pink Floyd song *Money.* Listeners familiar with this classic-rock chestnut may know the song is primarily in the 7/4

Figure 6. Beat spectrogram of Pink Floyd's Money *showing the transition from 4/4 time to 7/4 time around 10 seconds (on the* x*-axis).*

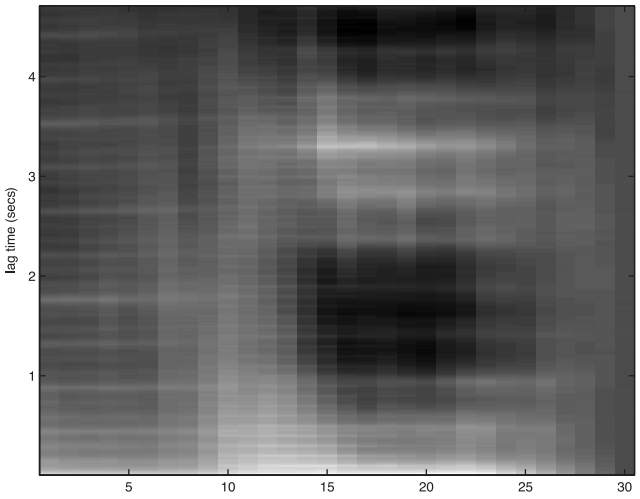
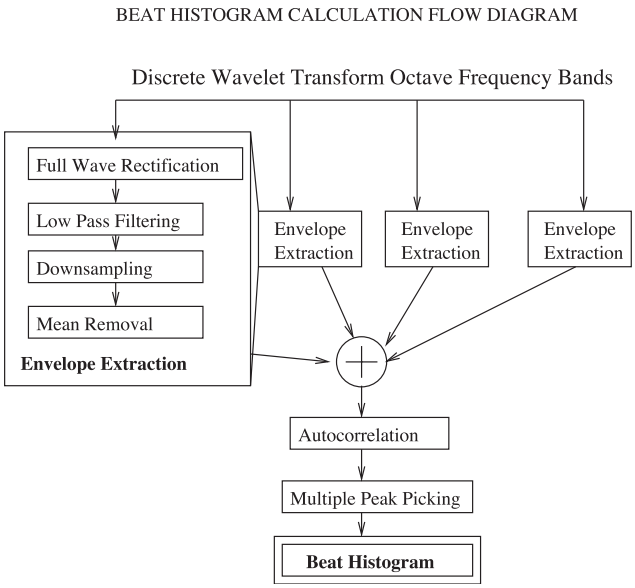


Figure 7. Flow diagram of beat histogram calculation.



time signature, save for the bridge (middle section), which is in 4/4. The excerpt shown begins at 4 min 55 sec into the song, and it clearly shows the transition from the 4/4 bridge back into the last 7/4 verse. To the left are strong beat spectral peaks on each beat, particularly at two and four beats (the length of a 4/4 bar), along with an eight-beat subharmonic. Two beats occur in slightly less than a second, corresponding to a tempo slightly faster than 120 beats per minute (120 BPM). This is followed by a short two-bar transition. Then, around 10 sec (on the *x*-axis) the time signature changes to 7/4, clearly visible as a strong seven-beat peak with the absence of a four-beat component. The tempo also slows slightly, visible as a slight lengthening of the time between peaks.

Beat Histograms

The *beat histogram* (BH) is similar to the beat spectrum in that it visualizes the distribution of various beat-level periodicities of the input signal. However, the method of calculation is different. The BH is calculated using periodicity detection in multiple octave channels that are computed using a discrete wavelet transform (DWT). Figure 7 shows a schematic diagram of the calculation. The signal is first decomposed into a number of octave frequency bands using the DWT. Following this decomposition, the time-domain amplitude envelope of each band is extracted separately. This is achieved by applying full-wave rectification, low-pass filtering, and downsampling to each octave frequency band. After removal of the mean, the envelopes of each band are then added together, and the autocorrelation of the resulting sum envelope is computed. The dominant peaks of the autocorrelation function correspond to the various periodicities of the signal's envelope. These peaks are accumulated over the whole sound file into a beat histogram, where each bin corresponds to the peak lag,namely, the beat period in BPM.

Rather than adding one, the amplitude of each peak is added to the beat histogram. That way, when the signal is very similar to itself (strong beat) the histogram peaks will be higher. In Tzanetakis and Cook (2002), six numerical features that attempt to summarize the BH are computed and used for classification. Figure 8 shows a BH for a piece of rock music (*Come Together* by the Beatles). (Notice the peaks at 80 BPM—the main tempo—and 160 BPM.) The *x*-axis corresponds to beats per minute, and the *y*-axis corresponds to the degree of self-similarity for that particular periodicity or *beat strength* (Tzanetakis, Essl, and Cook 2002). Many other algorithms for tempo and beat detection have

Figure 8. Beat histogram example for a piece of rock music (30 sec clip from Come Together *by The Beatles).*

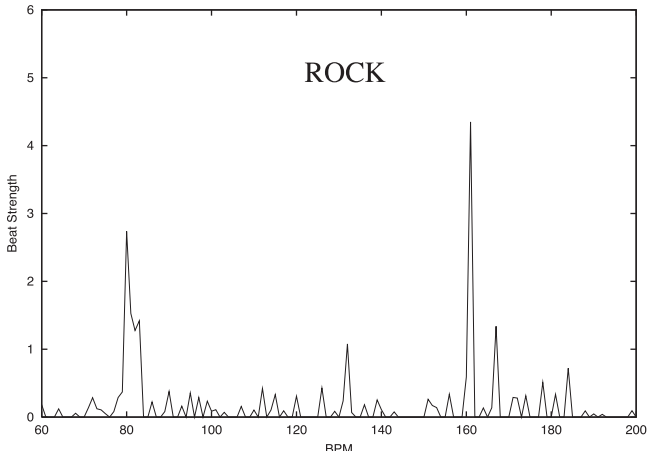
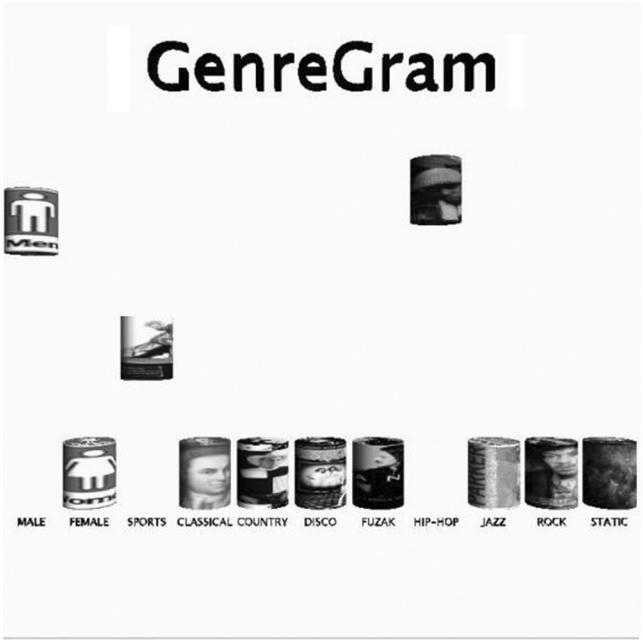


Figure 9. GenreGram, a dynamic real-time visualization of music classification.



been proposed in the literature and could be used as front ends to similar visualizations to the beat histogram and beat spectrum.

Real-Time Audio Classification Display

The *GenreGram* is a dynamic, real-time audio display for showing automatic genre classification results. More details about this process can be found in Tzanetakis and Cook (2002). The classification is performed using supervised learning where a statistical model of the feature distribution for each class is built during training with labeled samples. Once the classifier is trained, it can then be used to classify music it has not encountered before. Although it can be used with any audio signal, it was designed for real-time classification of live radio signals. Each genre is represented as a cylinder that moves up and down in real time based on a classification confidence measure ranging from 0.0 to 1.0. Each cylinder is texture-mapped with a representative image for each genre.

In addition to demonstrating real-time automatic musical genre classification, the GenreGram provides valuable feedback both to users and algorithm designers. Different classification decisions and their relative strengths are combined visually, revealing correlations and classification patterns. Because in many cases the boundaries between genres are fuzzy, a display like this is more informative than a single one-or-nothing classification decision. For example, both male speech and hip-hop are activated in the case of a hip-hop song, as shown in Figure 9. Of course, it is possible to use GenreGrams to display other types of audio classifications, such as instruments, sound effects, and birdsongs.

Mapping Time-Varying Timbre to Color

The basic idea behind *timbregrams* (Tzanetakis and Cook 2000a) is to map audio files to sequences of vertical color stripes in which each stripe corresponds to a short slice of sound (typically 20 msec to 0.5 sec). Time is mapped from left to right. The similarity of different files (context) is shown as overall color similarity, while the similarity within a file (content) is shown by color similarity within the timbregram. For example, a file that has an ABA structure, where section A and section B have different sound textures, will have an ABA structure in color also. Although it is possible to manually create timbregrams, they are typically created using PCA over automatically extracted feature vectors. Unlike approaches that directly map frequency content to

color such as Comparisonics (www.comparisonics.com), timbregrams allow any parametric audio representation to be used as a front end.

Two main approaches are used for mapping the principal components to color to create timbregrams. If an indexed image is desired, then the first principal component is divided equally, and each interval is mapped to an index of a colormap. Any standard visualization colormap such as grayscale or thermometer can be used. This approach works especially well if the first principal component explains a large percentage of the variance of the data set. In the second approach, the first three principal components are normalized so that they have equal means and variances. Although this normalization distorts the original feature space, in practice it provides more satisfactory results as the colors are more clearly separated. Each of the first three principal components is mapped to coordinates in a RGB or HSV color space. In this approach, a full-color timbregram is created.

There is a tradeoff between the ability to show small-scale local structure and global overall similarity depending on the quantization levels and the amount of variance in color range allowed. For example, by allowing many quantization levels and a large color-range variance, different sections of the same audio file can be visually distinguished. If only an overall color is desired, fewer quantization levels and smaller variation should be used.

It is important to note that the similarity in color depends not only on the particular file (content) but also the collection over which the PCA is calculated (context). That means that two files might have timbregrams with similar colors as part of collection A and timbregrams with different colors as part of collection B. For example, a string quartet and orchestral piece will have different timbregrams if viewed as part of a classical music collection but similar timbregrams if viewed as part of a collection that contains files from many different musical genres.

Timbregrams can be arranged in two-dimensional tables for browsing. The table axis can be either computed automatically or manually created. For example, one axis might correspond to the year of release, and the other might correspond to the automatically extracted tempo of the song. In addition, timbregrams can be superimposed over traditional waveform displays (see Figure 10) and texture mapped over objects in a timbre space, described later.

Figure 11 shows the timbregrams of six sound files. The three files on the left column contain speech and the three on the right contain classical music. It is easy to visually separate music and speech even in the grayscale image. It should be noted that no explicit class model of music and speech is used and the different colors are a direct result of the visualization technique. The bottom-right sound file (opera) is light purple and the speech segments are light green. In this mapping, light and bright colors correspond to speech or singing (Figure 11, left). Purple and blue colors typically correspond to classical music (Figure 11, right).

Timbregrams of pieces of orchestral music are shown in Figure 12. From the figure, it is clear that the fourth piece from the top has an AB structure, where the A part is similar to the second piece from the top and the B part is similar to the last piece from the top. The A part is light pink (in color) or light gray (in grayscale), and the B part is dark purple (in color) or dark grey (in grayscale). This is confirmed by listening to the corresponding pieces in which A is a loud, energetic movement where the entire orchestra is playing and B is a lightly orchestrated flute solo.

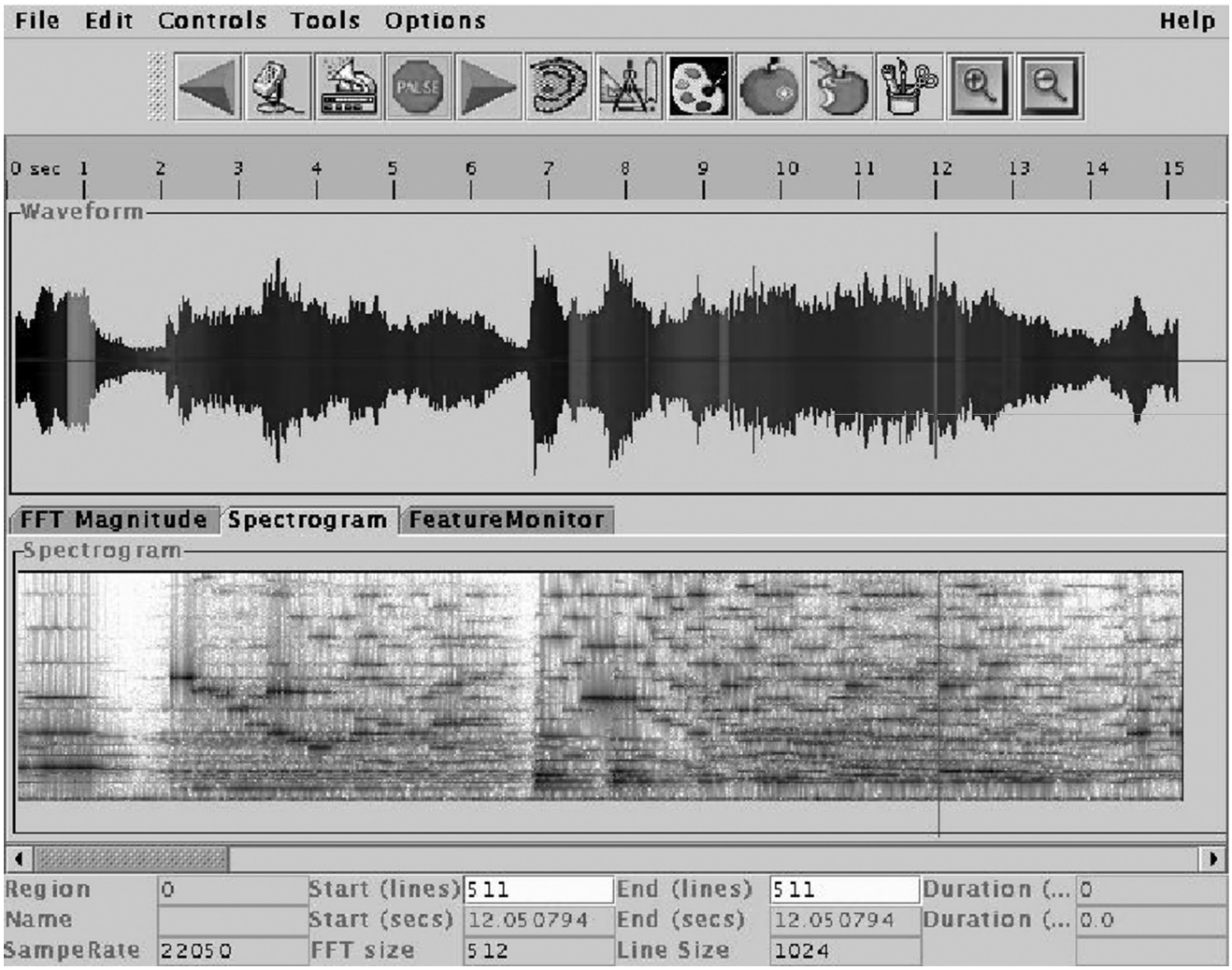
Visualizing Music Collections

Managing the increasing size of digital music and sound collections is challenging. Traditional tools such as the file browser provide little information to assist this process. In this section, some approaches to visualizing large music collections for browsing and retrieval are described.

Timbre Spaces

The Timbre Space Browser (Tzanetakis and Cook 2000b) maps each audio file to an object in a two- or three-dimensional virtual space. The main properties that can be mapped are the *x*, *y*, and *z* coordi-

Figure 10. Timbregram superimposed over a waveform.



nates for each object. In addition, a shape, texture image or color, and text annotation can be provided for each object. Standard graphical operations such as display zooming, panning, and rotating can be used to explore the browsing space. Data model operations such as section pointing and semantic zooming are also supported. Selection specification can also be performed by specifying constraints on the browser and object properties. For example, the user can ask to select all the files that have positive *x* values, triangular shapes, and red color. Principal curves, originally proposed in Hastie and Stuetzle (1989) and used for sonification in Herman, Meinicke, and Ritter (2000) can be used to move sequentially through the objects.

Figure 13 shows a two-dimensional timbre space of sound effects. The icons represent different types of sound effects such as walking (dark squares) and various other types of sound effects (white squares) such as tools, telephones, and door-bell sounds. Although the icons have been assigned manually, the x and *y* coordinates of each icon are calculated automatically based on audio features. This way, files that are similar in content are visually clustered together, as can be seen from the figure where the dark walking sounds occupy the left

Figure 11. Timbregrams of speech (left) and music (right). (The left column is light green, and the right column is dark purple.)

Figure 12. Timbregrams of orchestral music pieces.

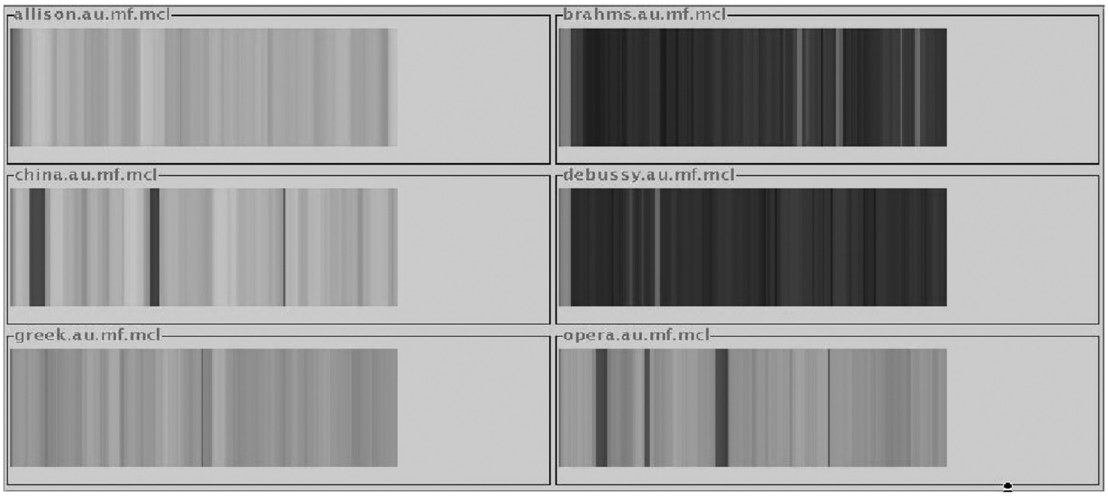


Figure 11

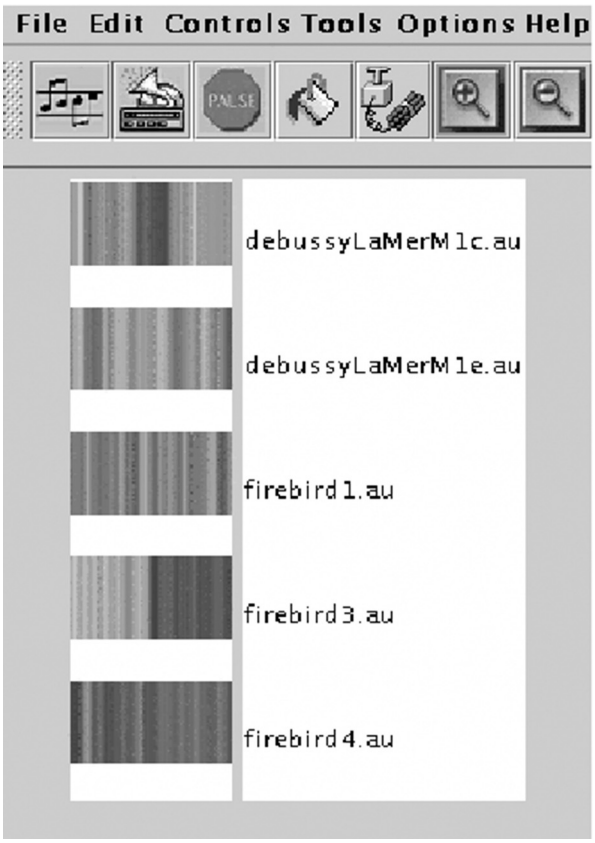


Figure 12

side of the figure while the white sounds occupy the right side.

Figure 14 shows a 3-D timbre space of different pieces of orchestral music. Each piece is represented as a colored rectangle. The *x, y,* and *z* coordinates are automatically extracted based on music similarity and the rectangle coloring is based on timbregrams. These figures contain fewer objects than typical configurations for clarity of presentation on paper. Audio collections with sizes that typically range from 100 to 1,000 files/objects can easily be accommodated with timbre spaces.

Music Similarity via Self-Organizing Maps and Smoothed Data Histograms

Islands of Music is a graphical interface to music collections (Pampalk 2001; Pampalk, Rauber, and Merkl 2002a). Similar pieces of music are automatically clustered into groups and visualized as islands. On an island, mountains and hills represent subgroups of similar pieces. Land bridges connect related islands. The music is arranged such that similar pieces and groups are close to each other on the map.

Islands of Music is based on the self-organizing map (SOM) neural network algorithm (Kohonen

Figure 13. Two-dimensional timbre space of sound effects.

Figure 14. Three-dimensional timbre space of orchestral music pieces.

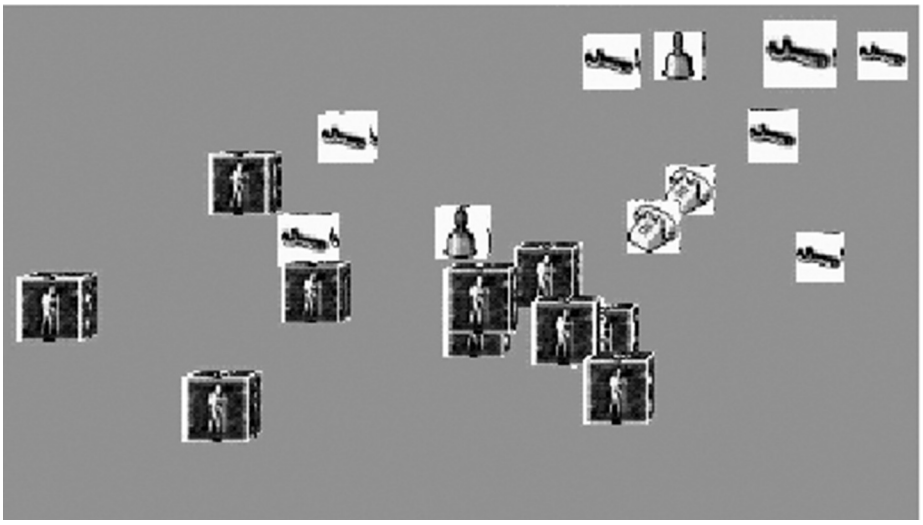


Figure 13

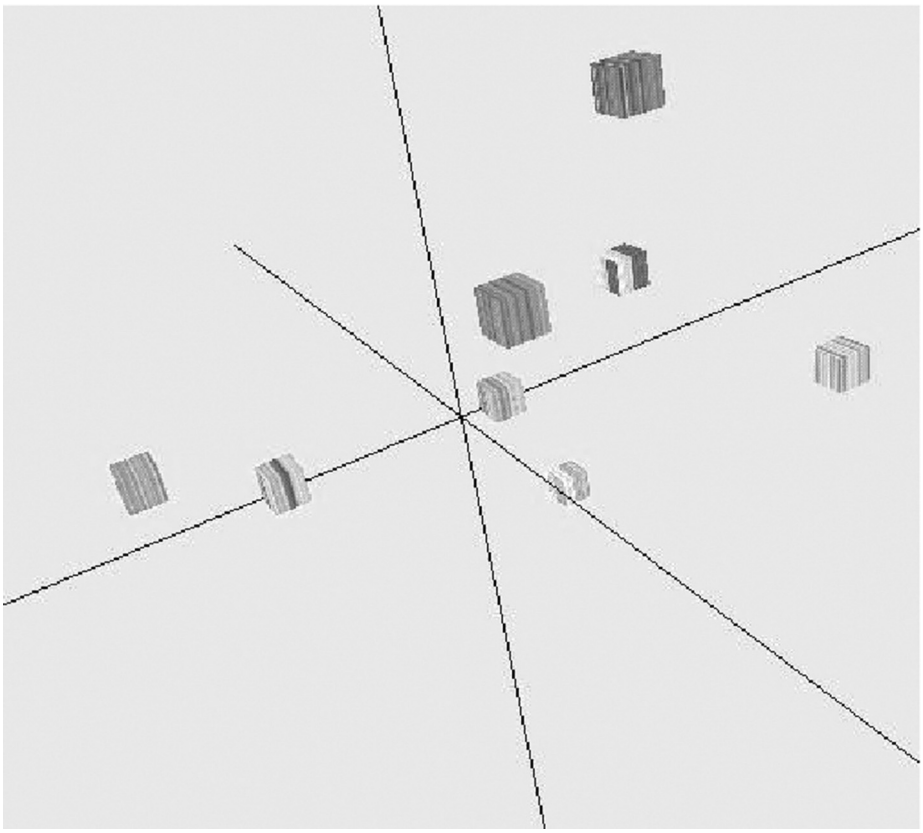


Figure 14

2001) combined with the smoothed data histogram (SDH) visualization (Pampalk, Rauber, and Merkl 2002b). The SOM consists of units arranged on a fixed grid. Each unit represents a prototypical piece of music. The pieces in the collection are mapped to the prototype that is most similar (also known as the best-matching unit). During the training process, the units are adapted to better represent the pieces mapped to them, with the constraint that units close to each other on the grid represent similar music.

The SDH visualization shows the distribution of the music collection on the map. It is a rough and robust estimate of the corresponding probability density function. In particular, for each piece, the weighted contribution of the n closest units are accumulated in the histogram bins. One of the results is that the land bridges between related islands become more apparent. The SDH informs the user in which areas there is a high density of pieces ("islands") and which areas are populated sparsely (the "sea"). The color mapping used has the following range: dark blue (deep sea), light blue (shallow sea), yellow (beach), light green (grass), dark green (forest), gray (rocks), and white (snow).

The most critical component of the Islands of Music interface is the similarity measure that is used. Several measures are available (e.g., Pampalk 2004); however, none of these performs comparably to similarity ratings by a human listener.

An example for Islands of Music is shown in Figure 15. (The same map in two dimensions is shown in Figure 16.) The snow-covered mountain on the lower left represents more aggressive music from Papa Roach and Limp Bizkit (a mix of metal, punk, and rap). Following the land bridge to the mountain toward the right leads to less aggressive music, including *Living in a Lie* by Guano Apes, *Not an Addict* by K's Choise, and *Adia* by Sarah McLachlan (all of which are slow songs sung by women and sound quite similar); this area also includes songs such as *Yesterday* by the Beatles, *California Dreaming* by The Mamas and The Papas, and *House of the Rising Sun* by The Animals. The third snow-covered mountain (lower right) represents classical music such as *Für Elise* by Beethoven. Other pieces on the same island include orchestra pieces and slow love songs.

A frequently asked question is what the x-axis and y-axis represent. If the mapping were linear, the two dimensions would correspond to the first two principal components of the data. However, the main advantage of the SOM compared to the PCA is its ability to map the data nonlinearly. Thus, it is not possible to directly label the axes. Nevertheless, different options to explain the regions of the map are available such as "weather charts" (Pampalk, Rauber, and Merkl 2002a). The idea is to visualize a third dimension (temperature, air pressure, strength of bass beats, etc.) on top of a map.

An example visualizing the distribution of bass beats is shown in Figure 17. Note that instead of the grayscale used here, we usually use a color scale ranging from blue (low values) to red (high values). The classical music island has the weakest bass beats, the two other mountains described previously have about average bass beats, and most of the islands in the upper region have very strong bass beats. For example, very strong bass beats can be found on the upper-left island, which contains mainly music from Bomfunk MC's (a mix of hip-hop, electro-funk, and house).

In the previous example, only 77 pieces were mapped. The same approach can be used for larger collections, as shown in Figure 18. This simply requires a zoom function. Alternatives include using growing hierarchical SOMs to organize music (Rauber, Pampalk, and Merkl 2002). To allow the user to fully benefit from hierarchical organizations, it would be useful to have an automatic summarization of individual music pieces and sets of pieces, for example, a short sequence of music that is typical for a whole island.

Figure 18 shows how an organization for a larger collection might look like. The highlighted island contains mainly music from Bomfunk MC's. The island a bit to the lower left contains mainly Red Hot Chili Peppers, and the mountain on the opposite side of the map (lower-left corner) contains mainly classical pieces.

Figure 19 shows the organization of 3,298 pieces from the magnatune.com collection. On the Web

Figure 15. Islands of Music.

Figure 16. Flat view of Figure 15 with song labels added.

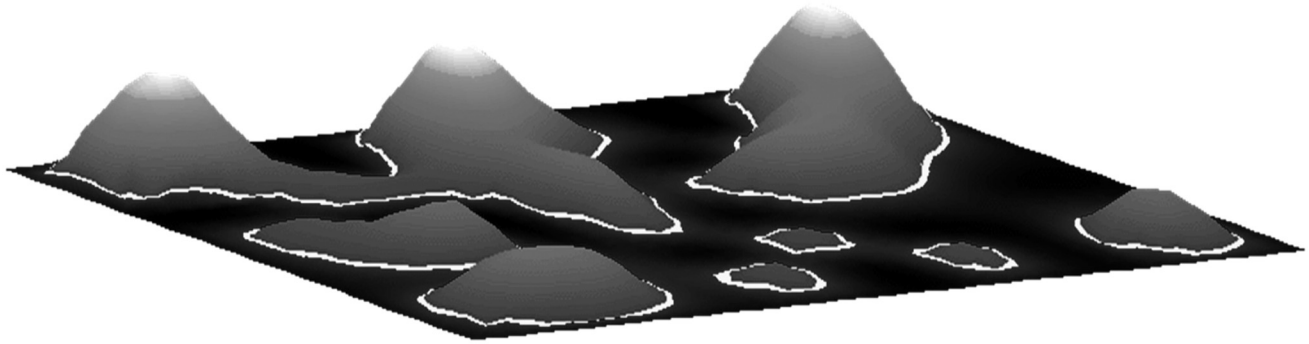


Figure 15

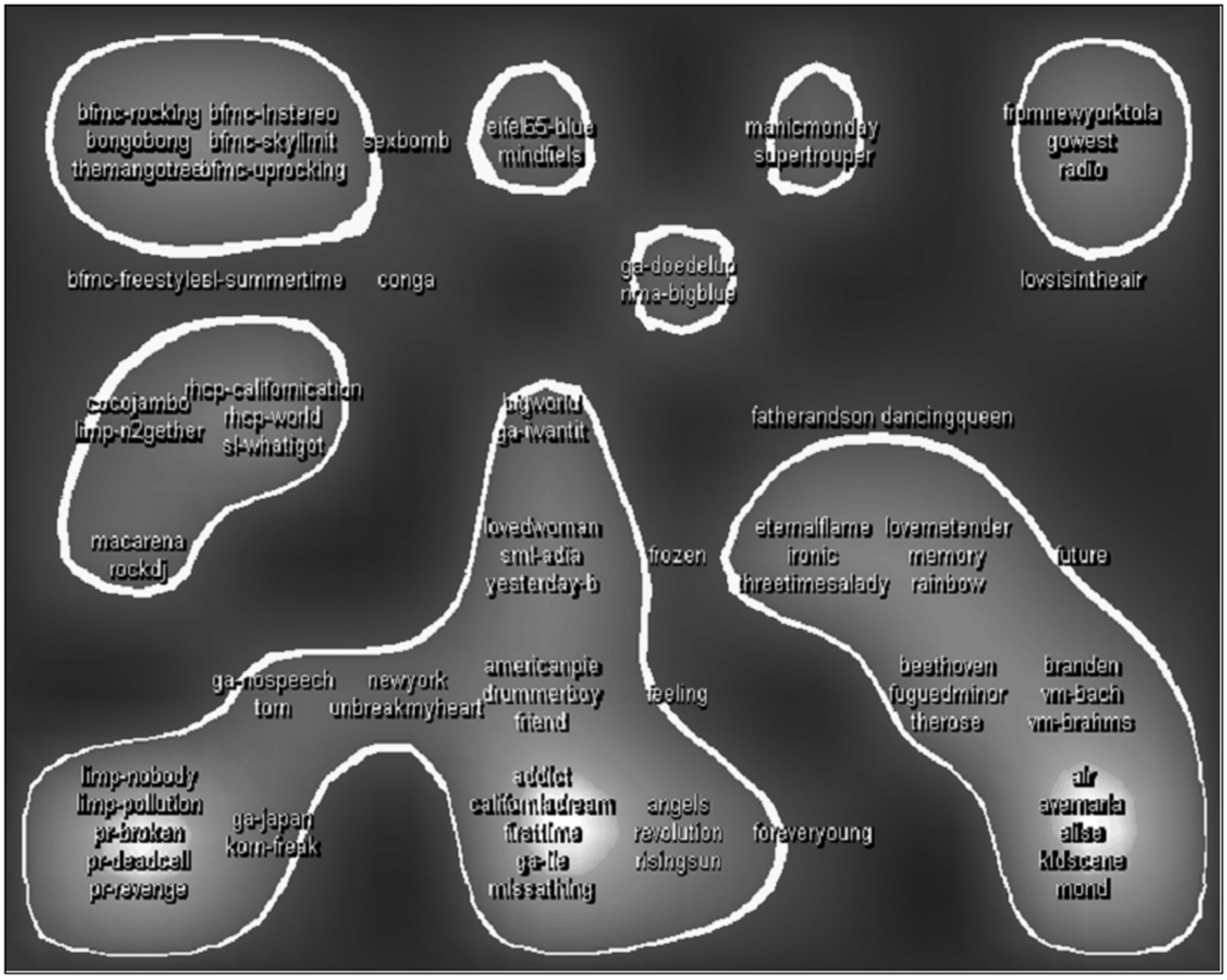
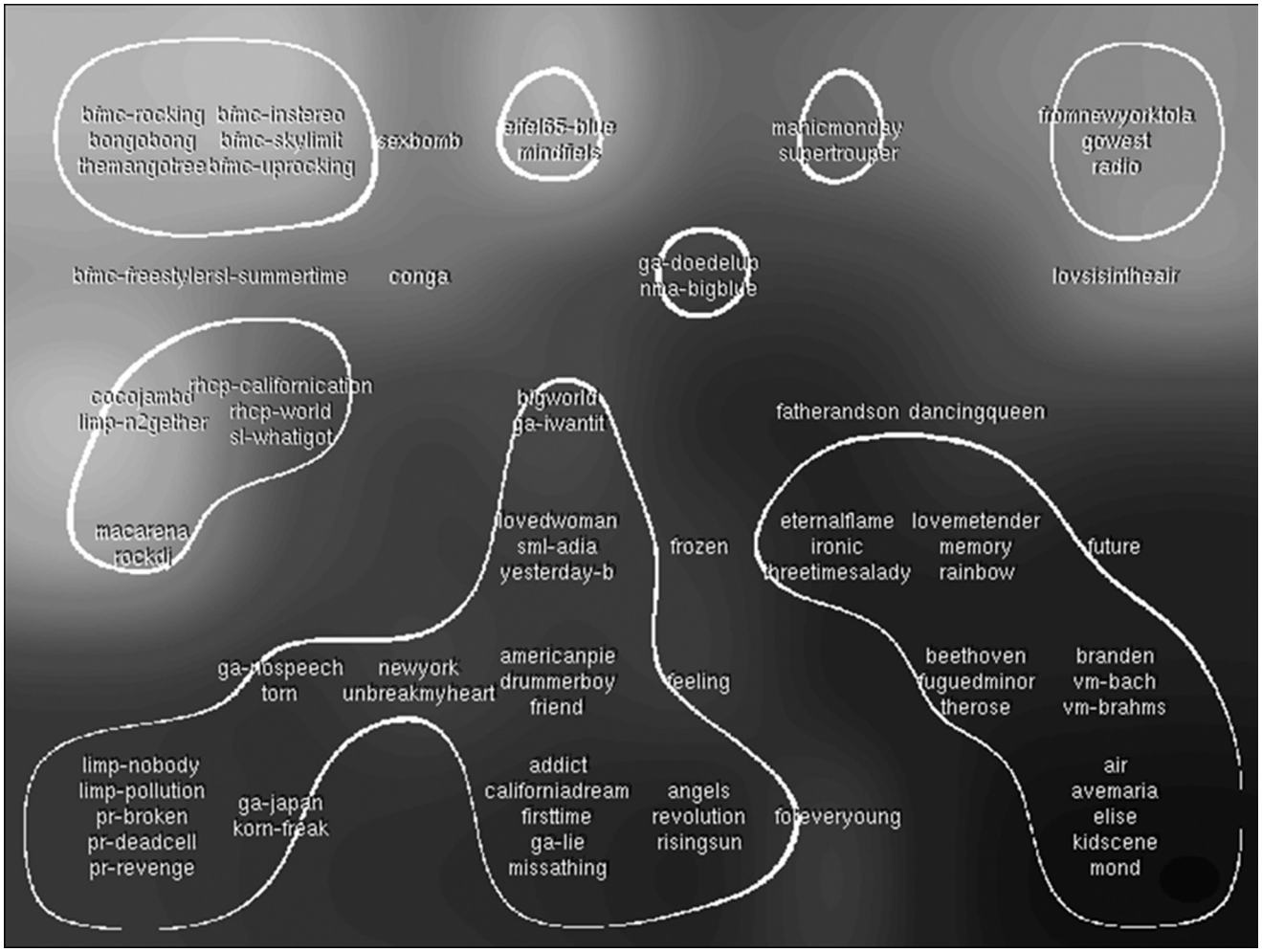


Figure 16

Figure 17. Weather charts, revealing areas with strong bass beats.

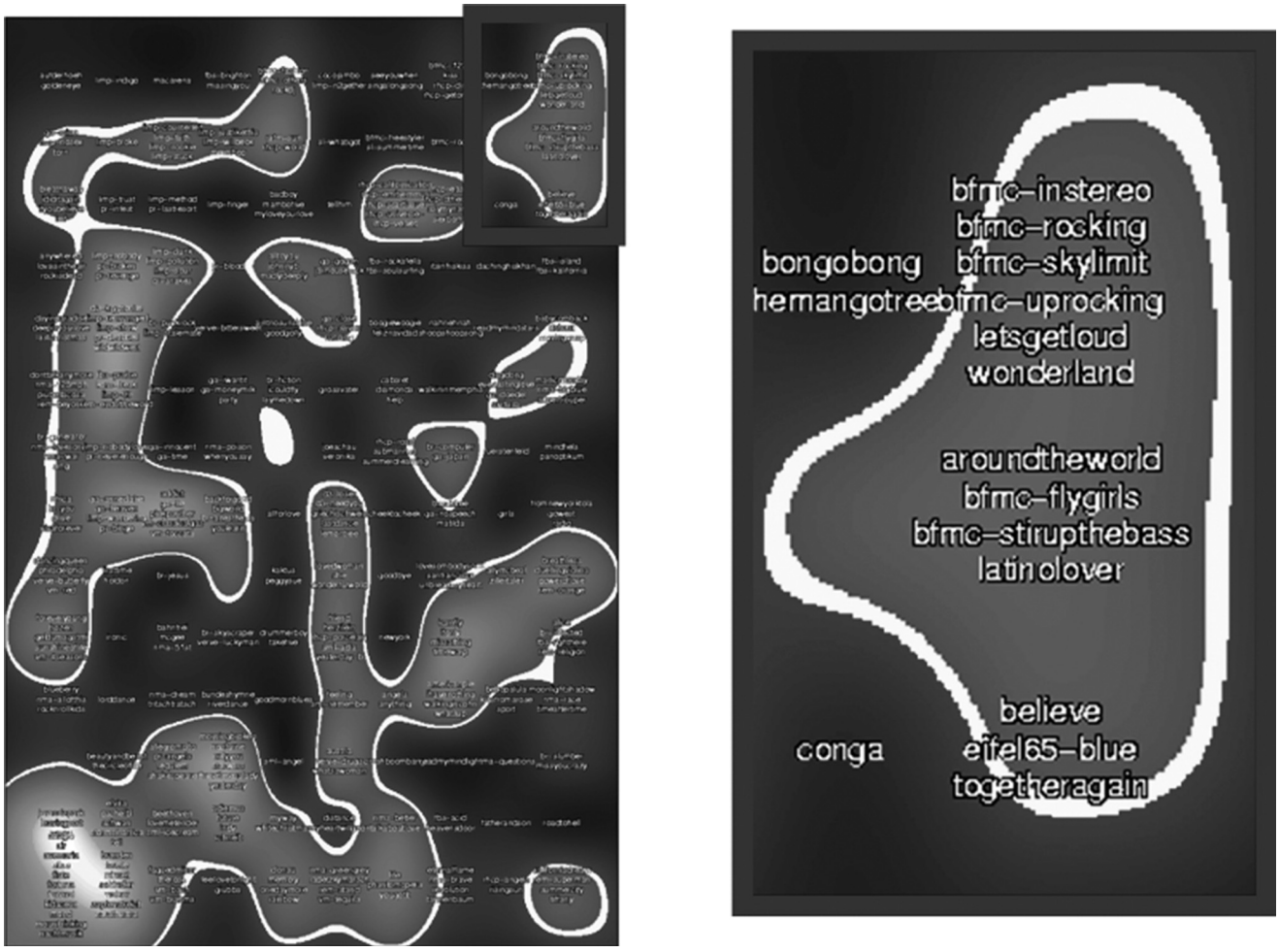


site, the pieces are organized into eleven genres, with most pieces labeled as classical music. Figure 19 shows the distribution of the genres on the SOM. Note that most genres are not an isolated and well-defined cluster but rather spread out over the whole map and have significant overlap with most other genres. In particular, world music is significantly spread out. The most compact cluster is punk music, which is located opposite from classical music.

Combining Different Views

So far, the assumption for the Islands of Music visualization was that there is one overall similarity measure according to which the pieces are organized. However, music similarity has many dimensions, such as tempo, rhythm, instrumentation, lyrics, cultural context, harmony, melody, and so on. Each of these dimensions defines a specific view of the music collection. These views can be combined using aligned-SOMs (Pampalk, Dixon, and Widmer 2004) to allow the user to smoothly and gradually change focus between the views.

Figure 18. Islands of Music with 359 pieces.



Aligned-SOMs are comprised of many individual SOMs of the same size stacked on top of each other. Each SOM represents a specific view, for example, derived from mixing 20 percent timbre and 80 percent rhythmic similarity. Neighboring SOMs represent similar views (i.e., similar mixing weights). During training of the SOMs, an additional constraint is enforced to ensure that each piece of music is located in the same area on neighboring SOMs.

Figure 20 shows an aligned-SOM combining a view based on rhythmic properties and a view based on MFCCs (describing spectral characteristics related to timbre). In the upper part of the screenshot is the Islands of Music view. For example, in the current view, classical piano pieces are located in the lower left and pieces from Papa Roach are in the upper right. Below are the codebooks, which are only of interest to the researcher studying specific characteristics of the similarity measures used. Below these codebooks is a slider that allows the user to focus more on rhythmic similarity (by moving the slider to the left) or timbre-based similarity (by moving the slider to the right). When the slider is moved, the pieces on the map are slowly rearranged to adjust to the new definition of similarity. As the pieces move, also the islands slowly move, sink, or emerge.

Figure 19. Visualization of genre distributions in the magnatune.com collection.

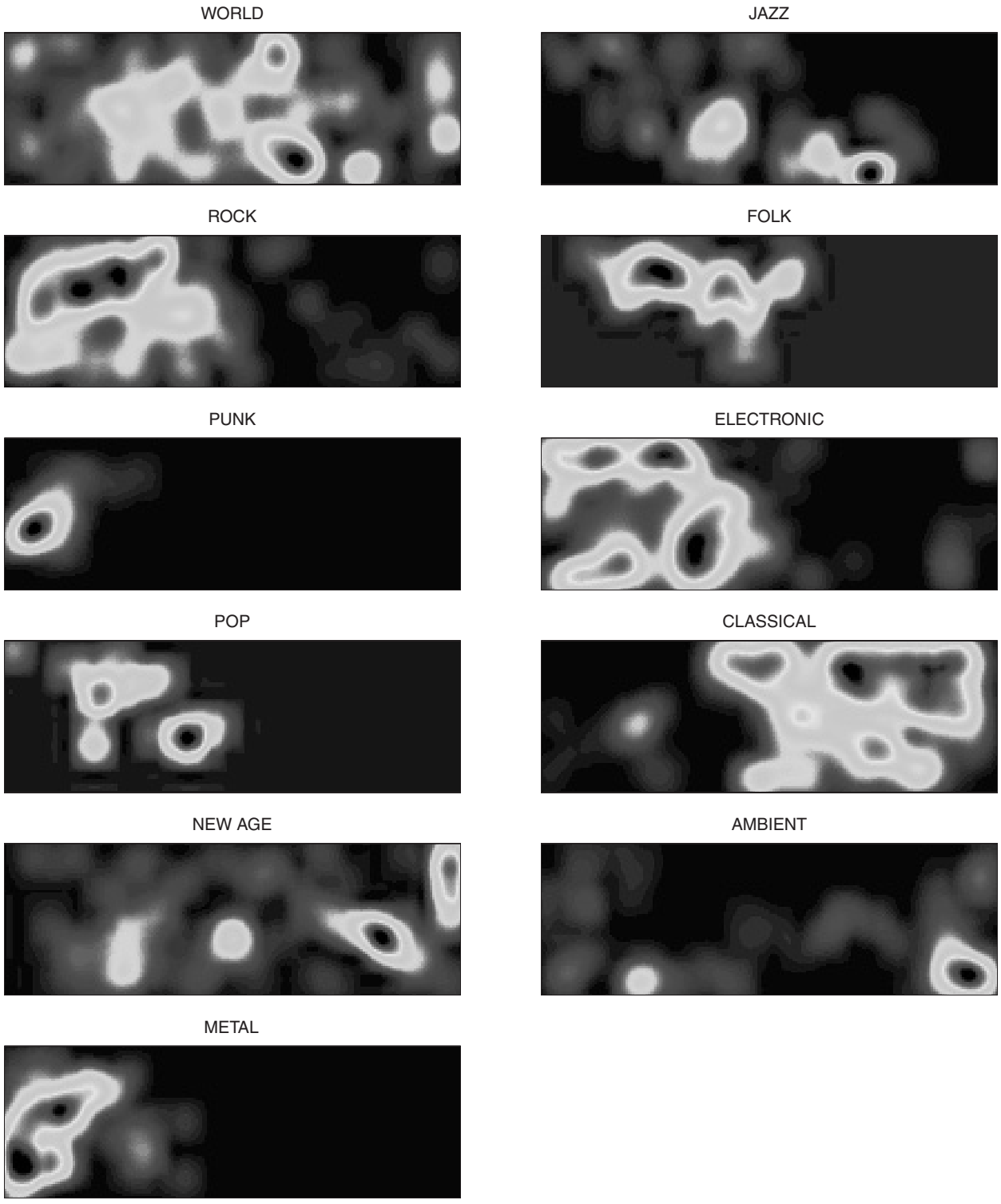
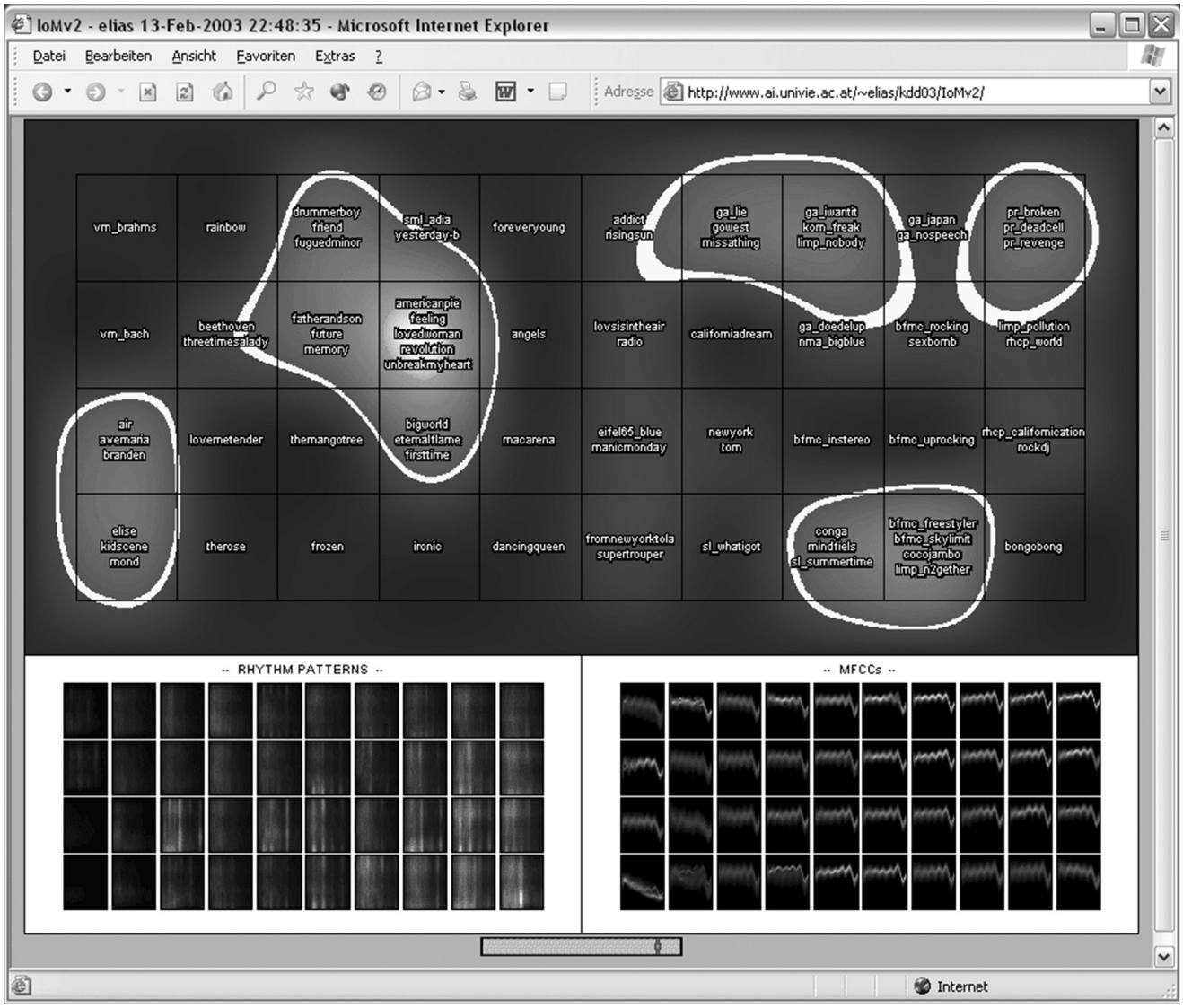


Figure 20. Combining different views using aligned-SOMs.



Conclusions

A variety of audio visualization techniques have been proposed in the context of MIR for representing music pieces and collections of them visually. These techniques rely on state-of-the-art audio signal processing and machine-learning techniques that automatically extract content information from audio signals that is subsequently mapped to visual attributes. By visualizing content information, one can take advantage of the strong pattern-recognition abilities of the human visual system to identify structure and patterns of audio signals. For example, the ABA structure of some piece of music can be immediately recognized in a visualization such as a similarity matrix or a timbregram but requires a few minutes of listening to be identified aurally.

A related field that is also still in its infancy is the visualization of symbolic data and connections to common music notation. The explicit visualization of pitch information and its integration with timbre and rhythm visualization are another area of future research. Evaluation is one of the biggest challenges in any type of visualization and typically requires extensive user studies. The field of MIR is new, and therefore related visualization techniques are still mostly an academic curiosity, and their evaluation has been mostly informal. It is our hope that these ideas will serve as seeds for highly interactive visual interfaces for exploring large collections of music in the future and more large-scale evaluation experiments.

References

Fayad, U., G. Grinstein, and A. Wierse, eds. 2002. *Information Visualization in Data Mining and Knowledge Discovery.* Burlington, Massachusetts: Morgan Kaufman.

Fernström, M., and E. Brazil. 2001. "Sonic Browsing: An Auditory Tool for Multimedia Asset Management." *Proceedings of the International Conference on Auditory Display:* 132–135.

Foote, J., and M. Cooper. 2003. "Media Segmentation Using Self-Similarity Decomposition." *Proceedings of SPIE* 5021:167–175.

Foote, J., and S. Uchihashi. 2001. "The Beat Spectrum: A New Approach to Rhythm Analysis." *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME).* Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Grey, J. M. 1975. "An Exploration of Musical Timbre." Ph.D. thesis, Department of Psychology, Stanford University.

Hastie, T., and W. Stuetzle. 1989. "Principal Curves." *Journal of the American Statistical Association* 84 (406): 502–516.

Herman, T., P. Meinicke, and H. Ritter. 2000. "Principal Curve Sonification." *Proceedings of the 2000 International Conference on Auditory Display.* New York: Association for Computing Machinery.

Jolliffe, I. 1986. *Principal Component Analysis.* New York: Springer.

Koenig, W., H. K. Dunn, and L. Y. Lacey. 1946. "The Sound Spectrograph." *Journal of Acoustical Society of America* 18:19–49.

Kohonen, T. 2001. *Self-Organizing Maps.* Berlin: Springer.

Malinowski, S. 1988. "The Music Animation Machine." Available online at www.well.com/user/smalin/mam.html.

Marchionini, G. 1995. *Information Seeking in Electronic Environments.* Cambridge: Cambridge University Press.

Moritz, M. 1996. "Mary Ellen Bute: Seeing Sound." *Animation World* 1(2):29–32.

Pampalk, E. 2001. *Islands of Music.* Master's thesis, Vienna University of Technology.

Pampalk, E. 2004. "A MATLAB Toolbox to Compute Similarity from Audio." *Proceedings of the 2004 International Conference on Music Information Retrieval.* Barcelona: Universitat Pompeu Fabra, pp. 254–257.

Pampalk, E., S. Dixon, and G. Widmer 2004. "Exploring Music Collections by Browsing Different Views." *Computer Music Journal* 28(2):49–62.

Pampalk, E., A. Rauber, and D. Merkl 2002a. "Content-Based Organization and Visualization of Music Archives." *Proceedings of ACM Multimedia.* New York: Association for Computing Machinery, pp. 570–579.

Pampalk, E., A. Rauber, and D. Merkl. 2002b. "Using Smoothed Data-Histograms for Cluster Visualization in Self-Organizing Maps." *Proceedings of the International Conference on Artificial Neural Networks.* Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, pp. 871–876.

Potter, P., G. Kopp, and H. Green. 1947. *Visible Speech.* New York: Van Nostrand.

Rauber, A., E. Pampalk, and D. Merkl. 2002. "Using Psycho-Acoustic Models and Self-Organizing Maps to Create a Hierarchical Structuring of Music by Sound Similarities." *Proceedings of the 2002 International Conference on Music Information Retrieval.* Paris: IRCAM, pp. 71–80.

Sapp, C. S. 2001. "Harmonic Visualizations of Tonal Music." *Proceedings of the 2001 International Computer Music Conference.* San Francisco, California: International Computer Music Association, pp. 423–430.

Slaney, M. 1997. "Connecting Correlograms to Neurophysiology and Psychoacoustics." Paper presented at the XI International Symposium on Hearing, Grantham, UK, 1–6 August.

Smith, S. M., and G. Williams. 1997. "A Visualization of Music." *Proceedings of Visualization '97.* New York: Association for Computing Machinery, pp. 499–502.

Spence, R. 2001. *Information Visualization.* Boston: Addison-Wesley.

Strang, G. 1988. *Linear Algebra and its Applications.* Orlando, Florida: Harcourt Brace Jovanovich.
Tzanetakis, G., and P. Cook. 2000a. "Audio Information Retrieval (AIR) Tools." *Proceedings of the 2000 International Conference on Music Information Retrieval.* Plymouth, Massachusetts: University of Massachusetts at Amherst.
Tzanetakis, G., and P. Cook. 2000b. "3D Graphics Tools for Sound Collections." Paper presented at the 3rd International Conference on Digital Audio Effects, Verona, Italy, 7–9 December.
Tzanetakis, G., and P. Cook. 2002. "Musical Genre Classification of Audio Signals." *IEEE Transactions on Speech and Audio Processing* 10(5):293–302.
Tzanetakis, G., G. Essl, and P. Cook. 2002. "Human Perception and Computer Extraction of Beat Strength." Paper presented at the 5th International Conference on Digital Audio Effects, Hamburg, Germany, 26–28 September.
Vertegal, R., and E. Bonis. 1994. "ISEE: An Intuitive Sound Editing Environment." *Computer Music Journal* 18 (2): 21–29.
Wessel, D. 1979. *Low-Dimensional Control of Musical Timbre.* Paris: Centre Georges Pompidou.