Pivot Vector Space Approach for Audio–Video Mixing

Philippe Mulhem
National Center of Scientific Research, France

Mohan S. Kankanhalli and Ji Yi
National University of Singapore

Hadi Hassan
Applied Science University, Jordan

An audio-mixing artist usually adds the musical accompaniment to video. Employing such artists is expensive and not feasible for a home video presentation. Our automatic audio–video mixing technique is suited for home videos. It uses a pivot vector space mapping method that matches video shots with music segments based on aesthetic cinematographic heuristics.

Audio mixing is an important aspect of cinematography. Most videos such as movies and sitcoms have several segments devoid of any speech. Adding carefully chosen music to such segments conveys emotions such as joy, tension, or melancholy. It also acts as a mechanism to bridge scenes and can add to the heightened sense of excitement in a car chase or reflect the somber mood of a tragic situation. In a typical professional video production, skilled audio-mixing artists aesthetically add appropriate audio to the given video shots. This process is tedious, time-consuming, and expensive.

With the rapid proliferation in the use of digital video camcorders, amateur video enthusiasts are producing a huge amount of home video footage. Many home video users would like to make their videos appear like professional productions before they share it with family and friends. To meet this demand, companies such as Muvee Technologies (http://www.muvee.com) produce software tools to give home videos a professional look. Our work is motivated by similar goals.

The software tool available from Muvee lets a user choose a video segment, audio clip, and mixing style (for example, music video or slow romantic). The Muvee software automatically sets the chosen video to the given audio clip incorporating special effects like gradual transitions, the type of which depends on the chosen style. If a user chooses an appropriate audio and style for the video, the result is indeed impressive. However, a typical home video user would lack the high skill level of a professional audio mixer needed to choose the right audio clip for a given video. It's quite possible to choose an inappropriate audio clip (say the one with a fast tempo) for a video clip (one that's slow with hardly any motion). The result in such a case would certainly be less than desirable.

Our aim is to approximately simulate the decision-making process of a professional audio mixer by employing the implicit aesthetic rules that professionals use. We have developed a novel technique that automatically picks the best audio clip (from the available database) to mix with a given video shot. Our technique uses a pivot vector space mixing framework to incorporate the artistic heuristics for mixing audio with video. These artistic heuristics use high-level perceptual descriptors of audio and video characteristics. Low-level signal processing techniques compute these descriptors. Our technique's experimental results appear highly promising despite the fact that we have currently developed computational procedures for only a subset of the entire suite of perceptual features available for mixing. Many open issues in the area of audio and video mixing exist and some possible problems in computational media aesthetics[1] need future work.

Aesthetic aspects

We initially tackled the problem of mixing music and moving images together by searching the existing artistic literature related to movies and cinematography. According to Andrew,[2] movies comprise images (still or moving); graphic traces (texts and signs); recorded speech, music, and noises; and sound effects. Prince highlights Aaron Copland's categorization of different roles of music in movies:[3]

- setting the scene (create atmosphere of time and place),
- adding emotional meaning,
- serving as a background filler,

1070-986X/03/$17.00 © 2003 IEEE

Published by the IEEE Computer Society

Table 1. Corresponding pairs of video and audio aesthetic features.

Video Feature	Extractable/Used	Audio Feature	Extractable/Used
Light type	No/no	Rhythm	Yes/no
Light mode	Yes/no	Key	No/no
Light falloff	Yes/yes	Dynamics	Yes/yes
Color energy	Yes/yes	Dynamics	Yes/yes
Color hue	Yes/yes	Pitch	Yes/yes
Color saturation	Yes/yes	Timbre	No/no
Color brightness	Yes/yes	Dynamics	Yes/yes
Space screen size	No/no	Dynamics	Yes/yes
Space graphic weight	No/no	Chords and beat	No/no
Space general shape	No/no	Sound shape	No/no
Object in frame	No/no	Chord tension	No/no
Space texture	Yes/no	Chords	No/no
Space field density/frame	No/no	Harmonic density	No/no
Space field density/period	No/no	Melodic density	No/no
Space field complexity/frames	No/no	Melodic density	No/no
Space graphic vectors	No/no	Melodic line	No/no
Space index vectors	No/no	Melodic progression	No/no
Space principal vector	Yes/no	Sound vector orientation	No/no
Motion vectors	Yes/yes	Tempo	Yes/yes
Zooms	Yes/no	Dynamics	Yes/yes
Vector continuity	Yes/no	Melodic progression	No/no
Transitions	Yes/no	Modulation change	No/no
Rhythm	No/no	Sound rhythm	No/no
Energy vector magnitude	No/no	Dynamics	Yes/yes
Vector field energy	Yes/no	Sound vector energy	No/no

- creating continuity across shots or scenes, and
- emphasizing climaxes (alert the viewer to climaxes and emotional points of scenes).

The links between music and moving images are extremely important, and the juxtaposition of such elements must be carried out according to some aesthetic rules. Zettl[4] explicitly defined such rules in the form of a table, presenting the features of moving images that match the features of music. Zettl based these proposed mixing rules on the following aspects: tonal matching (related to the emotional meaning defined by Copland), structural matching (related to emotional meaning and emphasizing climaxes defined by Copland), thematic matching (related to setting the scene as defined by Copland), and historical-geographical matching (related to setting the scene as defined by Copland). In Table 1, we summarize the work of Zettl by presenting aesthetic features that correspond in video and music. For instance, in the third row of Table 1, the light falloff video feature relates to the dynamics musical feature. The table also indicates extractable features (because many video and audio features defined by Zettl are high-level perceptual features and can't be extracted by the state of the art in computational media aesthetics), as well as we present the features that we use in our work.

Video aesthetic features

Table 1 shows, from the cinematic point of view, a set of attributed features (such as color and motion) required to describe videos. The computations for extracting aesthetic attributed features from low-level video features occur at the video shot granularity. Because some attributed features are based on still images (such as high light falloff), we compute them on the key frame of a video shot. We try to optimize the trade-off in accuracy and computational efficiency among the competing extraction methods. Also, even though we assume that the videos considered come in the MPEG format (widely used by several home video camcorders), the features exist independently of a particular representation format.

Light falloff

Light falloff refers to the brightness contrast between the light and shadow sides of an object and the rate of change from light to shadow. If the brightness contrast between the lighted side of an object and the attached shadow is high, the frame has fast falloff. This means the illuminated side is relatively bright and the attached shadow is quite dense and dark. If the contrast is low, the resulting falloff is considered slow. No falloff (or extremely low falloff) means that the object is lighted equally on all sides.

To compute light falloff, we need a coarse background and foreground classification and extraction of the object edges. We adapt a simplified version of the algorithm in Wang et al.[5] that detects the focused objects in a frame using multiresolution wavelet frequency analysis and statistical methods. In a frame, the focused objects (in home video, this often means humans) have more details within the object than the out-of-focus background. As a result, the focused object regions have a larger fraction of high-valued wavelet coefficients in the high frequency bands of the transform. We partition a reference frame of a shot into blocks and classify each block as background or foreground. The variance of wavelet coefficients in the high-frequency bands distinguishes background and foreground. The boundary of the background-foreground blocks provides the first approximation of the object boundary.

The second step involves refining this boundary through a multiscale approach. We perform successive refinements at every scale[5] to obtain the pixel-level boundary. After removing the small isolated regions and smoothing the edge, we calculate the contrast along the edge and linearly quantize the values. The falloff edge often has the highest contrast along the edge, so we select the average value in the highest bin as the value of light falloff in this frame.

Color features

The color features extracted from a video shot consist of four features: saturation, hue, brightness, and energy. The computation process is similar for the first three as follows:

1. Compute the color histogram features on the frames, set of intraframes: if we use the hue, saturation, and intensity (HSI) color space, the three histograms $hist_H$, $hist_S$, and $hist_{Brightness(B)}$ are respectively based on the H, S, and I components of the colors. We then obtain the dominant saturation, hue, and brightness in a shot.

2. Choose the feature values V_H, V_S, and V_B that correspond respectively to the dominant bin of each of $hist_H$, $hist_S$, and $hist_B$. All these value are normalized in [0, 1].

The values V_H, V_S, and V_B define a shot's hue, saturation, and brightness. The aesthetic color energy feature relates to the brightness, saturation, and hue features and is defined as $(V_H + V_S + V_B)/3$, which scales to the range [0, 1].

Motion vectors

To measure the video segments' motion intensity, we use descriptors from Pecker, Divakaran, and Papathomas.[6] They describe a set of automatically extractable descriptors of motion activities, which are computed from the MPEG motion vectors and can capture the intensity of a video shot's motion activity. Here we use the max2 descriptor, which discards 10 percent of the motion vectors to filter out spurious vectors or very small objects. We selected this descriptor for two reasons: The extraction of motion vectors from MPEG-1 and -2 compressed video streams is fast and efficient. Second, home videos normally have moderate motion intensity and are shot by amateur users who introduce high tilt up and down so that camera motion isn't stable. So, if we use the average descriptors, the camera motion's influence will be high. If we use the mean descriptor, the value will be close to zero, which will fail to capture the object's movement. Interestingly, max2 is also the best performing descriptor.

Aesthetic attributed feature formation

The descriptions discussed previously focus on features extraction, not on the attributed feature definitions. However, we can determine such attributed features. We collected a set of 30 video shots from two different sources: movies and home videos. We used this data set as the training set. A professional video expert manually annotated each shot from this training set, ascribing the label high, medium, or low for each of the aesthetic features from Table 1. Next, we obtained the mean and standard deviation of the assumed Gaussian probability distribution for the feature value of each label. We subsequently used these values, listed in Table 2, for estimating the confidence level of the attributed feature for any test shot.

Table 2. The mean and variance for the video and audio attributed features.

Video Feature	Attribute	m	s	Audio Feature	Attribute	m	s
Light falloff	High	0.3528	0.1323	Dynamics	High	0.7513	0.0703
	Medium	0.1367	0.0265		Medium	0.5551	0.0579
	Low	0.0682	0.0173		Low	0.3258	0.0859
Color energy	High	0.6515	0.1026	Dynamics	High	0.7513	0.0703
	Medium	0.4014	0.0819		Medium	0.5551	0.0579
	Low	0.1725	0.7461		Low	0.3258	0.0859
Color hue	High	0.7604	0.0854	Pitch	High	0.4650	0.0304
	Medium	0.552	0.0831		Medium	0.3615	0.0398
	Low	0.1715	0.1005		Low	0.0606	0.0579
Color brightness	High	0.8137	0.0954	Dynamics	High	0.7513	0.0703
	Medium	0.4825	0.1068		Medium	0.5551	0.0579
	Low	0.2898	0.0781		Low	0.3258	0.0859
Motion vector	High	0.6686	0.0510	Tempo	High	0.808	0.1438
	Medium	0.4683	0.0762		Medium	0.3873	0.0192
	Low	0.2218	0.0361		Low	0.0623	0.0541

Audio aesthetic features

Music perception is an extremely complex psycho-acoustical phenomenon that isn't well understood. So, instead of directly extracting the music's perceptual features, we can use the low-level signal features of audio clips, which can provide clues on how to estimate numerous perceptual features. In general, we found that perceptual label extraction for audio clips is a difficult problem and much more research is needed.

Low-level features

We describe here the required basic features that are extracted from an audio excerpt.

Spectral centroid (brightness). The spectral centroid is commonly associated with the measure of a sound's brightness. We obtain this measure by evaluating the center of gravity using the frequency and magnitude information of Fourier transforms. The individual centroid $C(n)$ of a spectral frame is the average frequency weighted by the amplitude, divided by the sum of the amplitude:[7-9]

$$C(n)=\frac{\int_0^\infty \omega\left|F_n(\omega)\right|^2 d\omega}{\int_0^\infty \left|F_n(\omega)\right|^2 d\omega}$$

where $F_n(\omega)$ represents the short-time Fourier transform of the nth frame, and the spectral frame is the number of samples that equals the size of the fast Fourier transform.

Zero crossing. In the context of discrete-time signals, a zero crossing is said to occur if two successive samples have opposite signs. The rate at which zero crossings occur is a simple measure of the frequency content of the signal. This is particularly true of the narrowband signals. Because audio signals might include both narrowband and broadband signals, the interpretation of the average zero-crossing rate is less precise. However, we can still obtain rough estimates of the spectral properties using a representation on the short-time average zero-crossing rate, as defined below:

$$ZCR = \frac{1}{N}\sum_m \left|\operatorname{sgn}[s(m)] - \operatorname{sgn}[s(m-1)]\right| w(m)$$

where, $\operatorname{sgn}(x) = \{1$ if $x \geq 0$, and -1 if $x \leq 0$,

and $w(m) = \{0.5(1 - \cos(2\pi \frac{m}{N-1}))$ if $0 < m < N - 1$, and 0 otherwise

Note that $w(m)$ is the Hamming window, $s(n)$ is the audio signal, and N is the frame length.

Volume (loudness). The volume distribution of audio clips reveals the signal magnitude's temporal variation. It represents the subjective measure, which depends on the human listener's frequency response. Normally volume is approximated by the root mean square value of the signal magnitude within each frame. Specifically, we calculate frame n's volume by

$$v(n)=\sqrt{\frac{1}{N}\sum_{i=0}^{N}S_n^{\,2}(i)}$$

where $S_n(i)$ is the ith sample in the nth frame of the audio signal, and N is the frame length. To measure the temporal variation of the audio clip's volume, we define two time domain measures based on the volume distribution. The first is the volume standard deviation over a clip, normalized by the maximum volume in the clip. The second is the volume dynamic range, given by

$$VDR(v)=\frac{\max(v)-\min(v)}{\max(v)}$$

Perceptual features extraction

We can relate the low-level audio features described previously with Table 1's perceptual labels required for our matching framework.

Dynamics. Dynamics refers to the volume of musical sound related to the music's loudness or softness, which is always a relative indication, dependent on the context. Using only the audio signal's volume features isn't sufficient to capture music clip dynamics because an audio signal could have a high volume but low dynamics. Thus, we should incorporate the spectral centroid, zero crossings, and volume of each frame to evaluate the audio signal's dynamics. We use a preset threshold (which we empirically choose using a training data set) for each feature to decide whether the audio clips' dynamics is high, medium, or low.

Tempo features. One of the most important features that makes the music flow unique and differentiates it from the other types of audio signal is temporal organization (beat rate). Humans perceive musical temporal flow as a rhythm related to the flow of music with the time. One aspect of rhythm is the beat rate, which refers to a perceived pulse marking off equal duration units.[10] This pulse is felt more strongly in some music pieces than others, but it's almost always present. When we listen to music, we feel the regular repetition of these beats and try to synchronize our feelings to what we hear by tapping our feet or hands. In fact, using certain kinds of instruments like bass drums and bass guitars synchronizes the rhythm flow in music.

Extracting rhythmic information from raw sound samples is difficult. This is because there's no ground truth for the rhythm in the simple measurement of an acoustic signal. The only basis is what human listeners perceive as the rhythmical aspects of the musical content of that signal. Several studies have focused on extracting the rhythmic information from the digital music representations such as the musical instrument digital interface (MIDI), or with reference to a music score.[11] Neither of these approaches is suited for analyzing raw audio data. For the purpose of our analysis, we adopted the algorithm proposed by Tzanetakis.[12] This technique decomposes the audio input signal into five bands (11 to 5.5, 5.5 to 2.25, 2.25 to 1.25, 1.25 to 0.562, and 0.562 to 0.281 KHz) using the discrete wavelet transform (DWT), with each band representing a one-octave range. Following this decomposition, the time domain envelope of each band is extracted separately by applying full wave rectification, low pass filtering, and down sampling to each band. The envelope of each band is then summed together and an autocorrelation function is computed. The peaks of the autocorrelation function correspond to the various periodicities of the signal envelope. The output of this algorithm lets us extract several interesting features from a music sample. We use DWT together with an envelope extraction technique and autocorrelation to construct a beat histogram. The set of features based on the beat histogram—which represents the tempo of musical clips—includes

- relative amplitudes of the first and second peaks and their corresponding periods,
- ratio of the amplitude between the second peak divided by the amplitude of the first peak, and
- overall sum of the histogram (providing an indication of overall beat strength).

We can use the amplitude and periodicity of the most prominent peak as a music tempo feature. The periodicity of the highest peak, representing the number of beats per minute, is a measure of the audio clips' tempo. We normalized the tempo to scale in the range [0, 1]. From the manual preclassification of all audio clips in the database and extensive experiments, we realized a set of empirical thresholds to classify whether the audio clips have a high, medium, or low tempo.

Perceptual pitch feature. Pitch perception plays an important role in human hearing, and the auditory system apparently assigns a pitch to anything that comes to its attention.[13] The seemingly easy concept of pitch in practice is fairly complex. This is because pitch exists as an acoustic property (repetition rate), as a psychological percept (perceived pitch), and also as an abstract symbolic entity related to interval and keys.

The existing computational multipitch algorithms are clearly inferior to the human auditory system in accuracy and flexibility. Researchers have proposed many approaches to simulate human perception. These generally follow one of two paradigms: place (frequency) theory or timing periodicity theory. In our approach, we don't look for accurate pitch measurement; instead we only want to approximate whether the level of the polyphonic music's multipitch is high, medium, or low. This feature's measurement has a highly subjective interpretation and there's no standard scale to define the pitch's highness. For this purpose, we follow the simplified model for multipitch analysis proposed in Tero and Matti.[14] In this approach, prefiltering preprocesses the audio signal to simulate the equal loudness curve sensitivity of the human ear and warping simulates the adaptation in the hair cell models. The frequency of the preprocessed audio signal is decomposed into two channels. The autocorrelation directly analyzes the low frequencies (below 1 KHz) channel, while a half-wave rectifier first rectifies the high frequencies (above 1 KHz) channels and then passes through a low pass filter. Next, we compute the autocorrelation of each band in a frequency domain by using the discrete Fourier transform as $\mathrm{corr}(\tau) = \mathrm{IDFT}\{|\mathrm{DFT}\{s(\tau)\}|^2\}$.

The sum of the two channel's autocorrelation functions represents the summary of autocorrelation functions (SACF). The peaks of SACF denote the potential pitch periods in the analyzed signal. However, the SACF include redundant and spurious information, making it difficult to estimate the true pitch peaks. A peak enhancement technique can add more selectivity by pruning the redundant and spurious peaks. At this stage, the peak locations and their intensity estimate all possible periodicities in the autocorrelation function. However, to obtain more robust pitch estimation, we should combine evidence at all subharmonics of each pitch. We achieve this by accumulating the most prominent peaks and their corresponding periodicities in a folded histogram over a 2,088-sample window size at a 22,050-Hz sampling rate, with a hop window size of 512 samples. In the folded histogram, all notes are transposed into a single octave (array of size 12) and mapped to a circle of fifths, so that the adjacent bins are spaced a fifth apart, rather than in semitones. Once the pitch histogram for an audio file is extracted, it's transformed to a single feature vector consisting of the following values:

- bin number of maximum peaks of the histogram, corresponding to the main pitch class of the musical piece;
- amplitude of the maximum peaks; and
- interval between the two highest peaks.

The amplitude of the most prominent peak and its periodicity, can roughly indicate whether the pitch is high, medium, or low. To extract the pitch level, we use a set of empirical threshold values and the same procedures as for tempo features extraction.

Audio aesthetic attributed feature formation

We collected a set of music clips from different music CDs, each with several music styles. Our data set contained 50 samples (each music excerpt is 30 seconds long). In our experiments, we sampled the audio signal 22 KHz and divided it into frames containing 512 samples each. We computed the clip-level features based on the frame-level features. To compute each audio clip's perceptual features (dynamics, tempo, and pitch as mentioned previously), a music expert analyzes the audio database and defines the high, medium, and low attributes for each feature, in a similar manner as with video shots. We eventually obtain the mean and variance of the Gaussian probability distribution to estimate the confidence level of the ternary labels for any given music clip. Table 2 lists the mean and variance parameters for each feature.

Pivot representation

We aim to define a vector space **P** that serves as a pivot between the video and audio aesthetics representations. This space is independent of any media, and the dimensions represent the aesthetic characteristics of a particular media. Pivot space **P** is a space on IR^p and is defined with the p set of aesthetic features in which the music and videos are mapped. The initial spaces **V** (for video) and **M** (for music) are respectively spaces

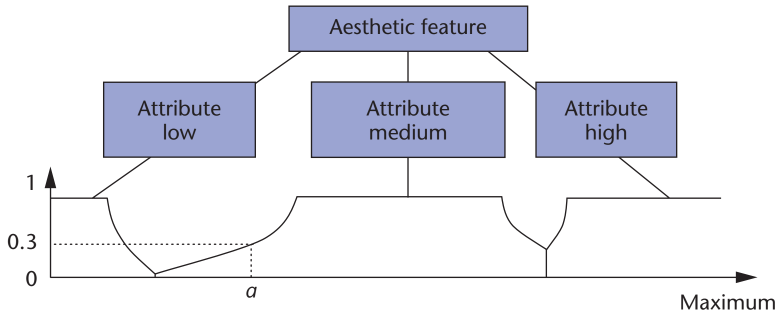

Figure 1. Fuzzy linguistic variables for aesthetic features.

on IR^v and IR^m, with v being the number of tuples (video feature, description) extracted for the video, and m the number of tuples (audio feature, description) extracted for the music excerpts.

Media representation of base vector spaces V and M

Once we define the aesthetic features, we consider how to represent video and audio clips into their aesthetic spaces **V** or **M**. In the two spaces, a dimension corresponds to an attributed feature. Instances of such attributed features for video data include brightness_high, brightness_low, and so on. One video shot is associated with one vector in the **V** space. Obtaining the values for each dimension resembles handling fuzzy linguistic variables, with the aesthetic feature playing the role of a linguistic variable and the attribute descriptor acting as a linguistic value,[15] as presented in Figure 1. In this figure, we represent sharp boundaries between fuzzy membership functions in a manner that removes correlation between them. The x-axis refers to the actual computed feature value and the y-axis simultaneously indicates the aesthetic label and the confidence value. Figure 1 shows an attributed feature value a, which has the medium label with a fuzzy membership value of 0.3.

Using a training collection for each linguistic value obtains the membership function. As described previously, we assume that each attributed feature follows a Gaussian probability distribution function, that is, we compute the mean μ_ij and standard deviation σ_ij on the samples so the probability density function $f_{\mu\_ij, \sigma\_ij}(x)$ becomes available for each aesthetic feature i and attribute j in (low, medium, high). We next translate each Gaussian representation into a fuzzy membership function, which can compute labels for the video or musical parts. Because we consider three kinds of attributes for each feature—namely high, medium, and low—one membership function represents each attribute, as shown in Figure 1. The following steps define the membership functions for M^{ij} for each aesthetic feature i and attribute j in (low, medium, high):

1. Thresholding the Gaussian distributions to ensure that the membership function fits into the interval [0, 1].

2. Forcing the membership function for the low attributes to remain constant and equal to the minimum of 1 and the maximum of the Gaussian function for values smaller than the mean μ.

3. Forcing the membership function for the high attributes to remain constant and equal to the minimum of 1, and the maximum of the Gaussian function for values greater than the mean μ.

4. Removing cross correlation by defining strict separation between the fuzzy membership functions of the different attributes for the same feature.

Formally, the membership functions $M^{i\ \mathrm{Small}}$ defined for the linguistic values that correspond to low attributes is $M^{i\ \mathrm{Small}}(x) = \min(1, f_{\mu\_i\mathrm{Small}, \sigma\_i\mathrm{Small}}(\mu))$ for $x \in [0, \mu]$; $\min(1, f_{\mu\_i\mathrm{Small}, \sigma\_i\mathrm{Small}}(x))$ for $x \in]\mu, y]$, y so that $f_{\mu\_i\mathrm{Small}, \sigma\_i\mathrm{Small}}(y) = f_{\mu\_i\mathrm{Medium}, \sigma\_i\mathrm{Medium}}(y)$; and 0 otherwise. We assume for $M^{i\ \mathrm{Small}}$ that no negative values occur for the base variable.

The membership function $M^{i\ \mathrm{Medium}}$ for attributes corresponding to medium is defined as $M^{i\ \mathrm{Medium}}(x) = \min(1, f_{\mu\_i\mathrm{Medium}, \sigma\_i\mathrm{Medium}}(x))$ for $x \in]y, \mu]$, y so that $f_{\mu\_i\mathrm{Small}, \sigma\_i\mathrm{Small}}(y) = f_{\mu\_i\mathrm{Medium}, \sigma\_i\mathrm{Medium}}(y)$; $\min(1, f_{\mu\_i\mathrm{Medium}, \sigma\_i\mathrm{Medium}}(x))$ for $x \in [\mu, z]$, z so that $f_{\mu\_i\mathrm{Medium}, \sigma\_i\mathrm{Medium}}(z) = f_{\mu\_i\mathrm{High}, \sigma\_i\mathrm{High}}(z)$; and 0 otherwise.

For the attribute that corresponds to high, the membership function $M^{i\mathrm{High}}$ is $M^{i\mathrm{High}}(x) = \min(1, f_{\mu\_i\mathrm{High}, \sigma\_i\mathrm{High}}(x))$ for $x \in]y, \mu]$, y so that $f_{\mu\_i\mathrm{Medium}, \sigma\_i\mathrm{Medium}}(y) = f_{\mu\_i\mathrm{High}, \sigma\_i\mathrm{High}}(y)$; $\min(1, f_{\mu\_i\mathrm{High}, \sigma\_i\mathrm{High}}(\mu))$ for $x \geq \mu$; and 0 otherwise.

So, for each feature i (in the video or music space), we compute the $M^{i\ \mathrm{Low}}$, $M^{i\ \mathrm{Medium}}$, or $M^{i\ \mathrm{High}}$ fuzzy membership functions using the previous equations. We then express the values obtained after a video shot or music excerpt analysis in the aesthetic space using the linguistic membership function value. So the dimension space v (respectively m) of **V** (respectively **M**) equals three times the number of video (respectively music) aesthetic features.

We represent one video shot as a point S_i in the space **V**. Each of the coordinates $s_{i,j}$ of S_i is in [0,1], with values close to 0 indicating that the corresponding aesthetic attributed feature doesn't properly describe Shot_i, and values close to 1 indicating that the corresponding aesthetic attributed feature well describes Shot_i. Similar remarks hold for the music excerpts represented as E_k in the space **M**. Table 2 presents the mean and variance obtained for each attributed feature that we used.

Pivot space mapping

The mapping going from the IR^v (respectively IR^m) to the IR^p space is provided by a $p \times v$ (respectively $p \times m$) matrix $\mathbf{T}_v$ (respectively T_m) that expresses a rotation or projection. Rotation allows the mapping of features from one space to another. Several features of the IR^v space might be projected in one feature of IR^p; it's also possible that one feature of IR^v might be projected onto several features of IR^p. The mapping is single stochastic, implying that the sum of each column of $\mathbf{T}_v$ and $\mathbf{T}_m$ equals 1. This ensures that the coordinate values in the pivot space still fall in the interval [0, 1] and can be considered fuzzy values. The advantage of the mapping described here is that it's incremental. This is because if we can extract new video features, the modification of the transformation matrix $\mathbf{T}_v$ preserves all the existing mapping of music parts. We directly extrapolated the transformation matrices $\mathbf{T}_v$ and $\mathbf{T}_m$ from Table 1 and define links between video- and music-attributed aesthetic features and pivot-attributed aesthetic features. Figure 2 shows the mapping process.

The fundamental role of the pivot space is allowing the comparison between video- and music-attributed aesthetic features. We compute the compatibility between the music described with V1 and the music described as M1 as the reciprocal of the Euclidian distance between V'1 and M'1. For instance, the cosine measure (as used in vector space textual information retrieval[16]) isn't adequate because we don't seek similar profiles in terms of vector direction, but on the distance between vectors. The use of Euclidean distance is meaningful; when the membership values for one video shot and one music excerpt are close for the same attributed feature, then the two media parts are similar on this dimension, and when the values are dissimilar, then the attributed feature for the media are different. Euclidean distance holds here also because we assume independence between the dimensions.

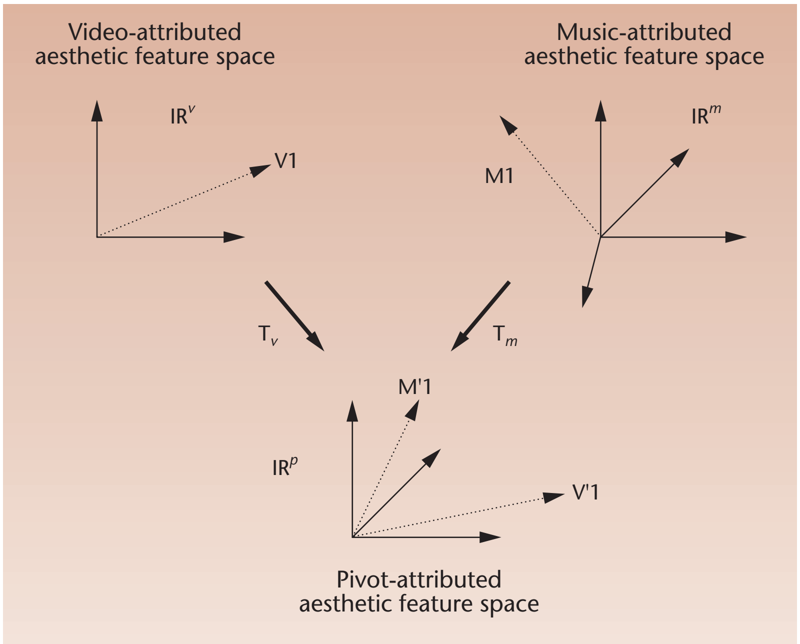

Figure 2. Pivot vector representation.

In our work, the dimension p is 9, because we assign one dimension of **P** for the three attributes high, medium, and low related to the following:

- Dynamics (related to light falloff, color energy, and color brightness for video and dynamics for music). Without any additional knowledge, we only assume that each of the attributed features of dynamics in the pivot space are equally based on the corresponding attributed video features. For instance, the high dynamics dimension in **P** comprises one-third each of the high light falloff, color energy, and color brightness.
- Motion (related to motion vectors of video and tempo of music).
- Pitch (related to the color hue of video and pitch of music).

In the following, we'll represent a music excerpt or a video shot in the pivot space using a 9-dimensional vector corresponding respectively to the following attributed features: low_dynamics, medium_dynamics, high_dynamics, low_motion, medium_motion, high_motion, low_pitch, medium_pitch, and high_pitch.

We now illustrate the mapping with one example taken from the file LGERCA_LISA_1.mpg that belongs to the MPEG-7 test collection. The selected shot, namely L01_39, is between the frame 22025 and 22940. Table 3 (next page) presents the extracted features and the mapping into

Table 3. Initial feature values and attributed features for the video shot L01_39 in the video space.

Features	Light Falloff	Color Hue	Color Brightness	Color Energy	Motion Vector
Feature value	0.658	0.531	0.643	0.413	0.312
High falloff	1.0				
Medium falloff	0.0				
Low falloff	0.0				
High hue		0.0			
Medium hue		1.0			
Low hue		0.0			
High brightness			0.0		
Medium brightness			1.0		
Low brightness			0.0		
High energy				0.0	
Medium energy				0.0	
Low energy				0.701	
High motion					0.0
Medium motion					0.638
Low motion					0.0

Table 4. Attributed features for the video shot L01_39 in the pivot vector space.

Low Dynamics	Medium Dynamics	High Dynamics	Low Motion	Medium Motion	High Motion	Low Pitch	Med Pitch	High Pitch
0.701	0.0	0.0	0.0	0.638	0.0	0.0	1.0	0.0

the video space. Table 4 shows the mapping into the pivot space.

Media sequence matching

From a retrieval point of view, the approach presented in the previous section provides a ranking of each music excerpt for each video shot and vice versa, by computing the Euclidean distances between their representatives in the pivot space. However, our aim is to find the optimal set of music excerpts for a given video where the compatibility of the video and music (as defined in Table 1) determines optimality.

One simple solution would be to only select the best music excerpt for each shot and play these music excerpts with the video. However, this approach isn't sufficient because music excerpt duration differs from shot duration. So, we might use several music excerpts for one shot, or have several shots fitting the duration of each music excerpt. We chose to first define the overall best match value between the video shots and the music excerpts. If we obtain several best matches, we take the longest shot, assuming that for the longer shots we'll have more accurate feature extraction. (Longer shots are less prone to errors caused by small perturbations.) Then we use media continuity heuristics to ensure availability of long sequences of music excerpts belonging to the same music piece.

Suppose that we obtained the best match for Shot_i and the music excerpt l from music piece k, namely $M_{k,l}$. Then we assign the music excerpts $M_{k,m}$ (with $m < l$) to the part of the video before Shot_i, and the music excerpts $M_{k,n}$ ($n > l$) to the parts of the videos after Shot_i. Thus, we achieve musical continuity. If the video is longer than the music piece, we apply the same process on the remaining part(s) of the video, by placing priority on the remaining parts that are contiguous to the already mixed video parts, ensuring some continuity. The previous description doesn't describe the handling of all the specific cases that could occur during the mixing (for example, several music parts might have the same matching value for one shot), but it gives a precise enough idea of the actual process.

Experiments

We tested our approach on 40 minutes of edited home videos (109 shots) taken from the

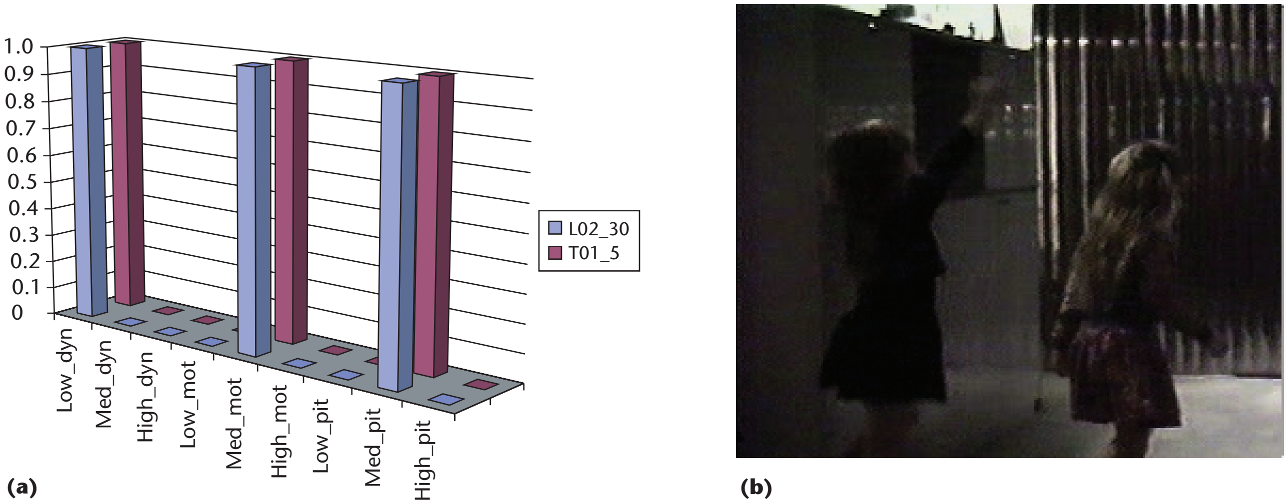

Figure 3. (a) Matching between the video L02_30 and the music T01_5. (b) A sample frame from the video.

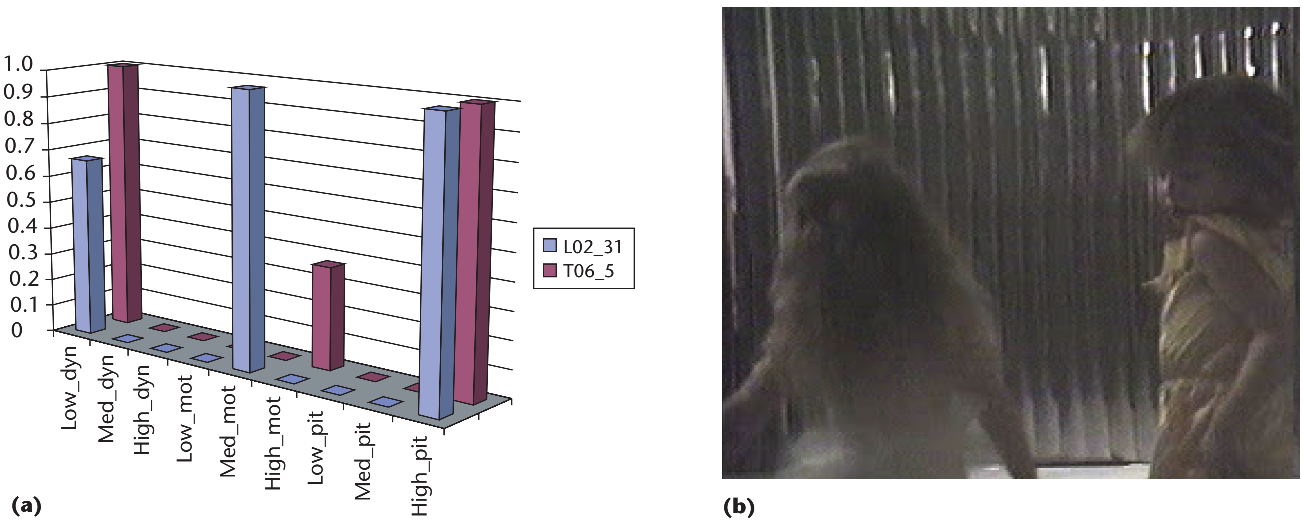

Figure 4. (a) Matching between the video L02_31 and the music T06_5. (b) A sample frame from the video.

MPEG-7 test collection (LGERCA_LISA_1.mpg from the CD no. 31 and LGERCA_LISA_2.mpg from the CD no. 32) and considered 93 minutes (186 excerpts of 30 seconds each) of music composed of instrumental rock, blues, and jazz styles. We defined the video excerpts according to the sequence list provided for the MPEG-7 evaluation.

Elemental shot-excerpt mixing results

We first present the matching of one music excerpt to one video shot. We chose three shots of the LGERCA_LISA_2.mpg (from frames 16044 to 17652), namely L02_30, L02_31, and L02_32. The first shot presents two girls dancing indoors with a dim light. Because the girls are dancing, there's motion, but no close-ups, so the motion is medium. The video's pitch (related to the hue of the colors presents in the shot) is also medium because the girls' clothes have some colors. Figure 3 presents nine dimensions of the pivot space for L02_30 and the best matching obtained with the music excerpt T01_5, extracted from the music piece "Slow and Easy" (from the music album *Engines of Creation* by Joe Satriani, Epic Records, 2000). This music is a medium- to slow-tempo rock piece, with medium dynamics and medium pitch. As Figure 3 shows, the matching is perfect and the distance is 0.

Figure 4 presents the features of the shot L02_31 in the pivot space. This shot is also dark, but less than L02_32, which is why the low_dynamics dimension has a value equal to

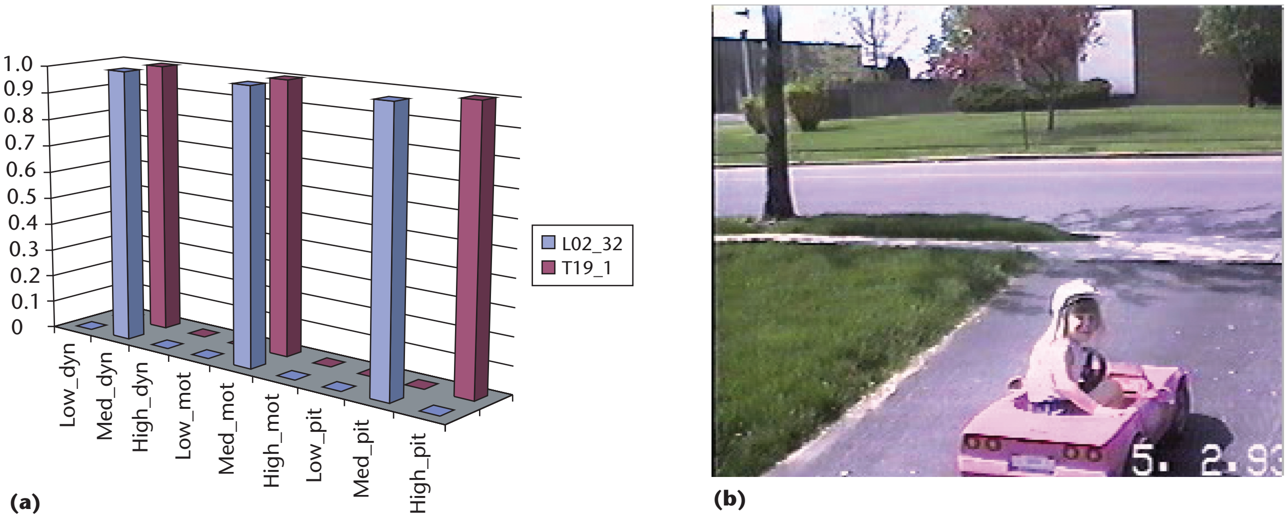

Figure 5. (a) Matching between the video L02_32 and the music T19_1. (b) Sample frame from the video.

0.66. In this shot the two girls dancing are closer to the camera, generating much more motion. Also, the dresses' colors are more visible, generating high_motion and high_pitch. The minimum distance value obtained is 0.071 for the music excerpt T6_5 (medium tempo rock, "Motorcycle Driver" from the album *The Extremist* by Joe Satriani, Relativity Records, 1992) with high pitch and low energy. The matching isn't perfect because the motion doesn't match, but this match is the best one we obtained.

Consider now the shot L02_32, taken outdoors. In this case, the images are brighter than in the previous two cases, and the feature med_dynamics equals 1. There isn't much motion because the focus point (a girl driving a small pink car) is far away. The pitch is also medium because there are only natural colors. The shot L02_32 best matches at a distance of 1.414 with the music excerpt T19_01 ("Platinum" from the eponymous album of Mike Oldfield, Virgin Records, 1979), as presented in Figure 5. This music excerpt is of medium tempo, with medium dynamics and high pitch.

Full audio–video mixing results

We matched a sequence of shots, which correspond to a "Kids Dancing on Stage and After Play" segment in the video LGERCA_LISA_2.mpg. We numbered the shots from L02_44 to L01_51. The segment shows children coming onto a stage, and then a prize nomination. This sequence lasts 2 minutes and 33 seconds. We obtained the best match for shot L02_48 (vector [0, 1, 0, 1, 0, 0, 0, 1, 0]) with the music excerpt T19_1, described previously; the match is perfect. The shot L02_48 has small motion activity, medium bright colors, and medium hue colors.

According to our rule to ensure media continuity, we mixed the same music for shots L02_49, L02_50, and L02_51. We then considered shot L02_47 (vector [0.95, 0, 0, 0, 1, 0, 0, 0, 1]), mixed with the music excerpt T06_1 (medium tempo rock, "Motorcycle Driver," from *The Extremist* album by Joe Satriani, Relativity Records, 1992) with a vector of (1, 0, 0, 0, 1, 0, 0, 0, 1). The distance between their respective vectors is 0.041. This shot contains not very bright images and not much motion but a lot of high hue colors, and we mixed it with medium tempo rock music with high pitch and low energy.

Because shot L02_47 is longer than the music excerpt, we mix it with music part T06_2. We continue to mix shots going backward from shot L02_47. Shot L02_46 (vector [0.97, 0, 0, 0, 1, 0, 0, 1, 0]) contains less high hue colors because it focuses on a girl wearing black clothes. We mixed it with its best music match, T05_2 (medium rock, "S.M.F," from *Joe Satriani* by Joe Satriani, Relativity Music, 1995), with a distance value of 0.024. By back propagating the music, we can mix the remaining shots L02_44 and L02_45 with music excerpt T05_1, which is the preceding part of T05_2 in the same song. Figure 6 shows the mixing obtained.

Conclusions

Our novel audio–video mixing algorithm picks the best audio excerpts to mix with a video

Web Extras

Our experiments obtained aesthetically pleasing mixing results. One example each of elemental (elem.mpg and elem.wmv) and full (full.mpg and full.wmv) audio–video mixing is available for viewing on *IEEE MultiMedia*'s Web site at http://computer.org/multimedia/mu2003/u2toc.htm.

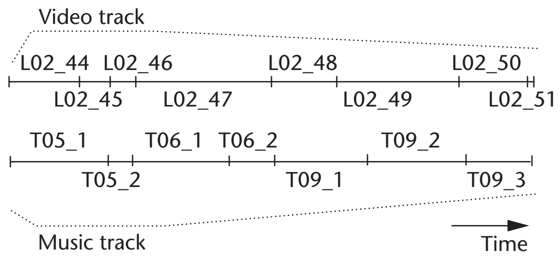

Figure 6. Mixing of the video shots from L02_44 to L02_51. The upper line presents the video shots sequence along the time axis and the lower line shows the musical audio tracks associated with the shots according to the timeline.

clip according to perceptual cues of both video and audio. Our work represents initial steps toward developing automatic audio–video mixing that rivals that of a skilled human mixing artist. Many interesting and challenging problems remain for future study.

We provided computational procedures for only a subset of the video features, but we need computational procedures for all video and audio descriptors. Table 1 lists a total of 43 attributed features, but only 16 of them are extractable, and we've only used 10 of them so far. Future research will develop procedures for all the attributed features so we can use the heuristics of Table 1 for audio–video mixing. While video processing seems relatively easier, hardly any corresponding work has occurred for music. The literature on digital audio processing is overwhelmingly skewed toward speech processing and scant work exists on nonspeech audio processing.

We essentially used the mixing heuristics as given in Zettl.[4] Perhaps better mixing heuristics are possible, and we need to understand better the aesthetic decision process of mixing artists.

Table 1 doesn't explicitly address music genre. It's obvious that while perceptual features of two audio clips of two different genres might be similar, their appropriateness for a particular video clip might differ. Because we use a Euclidean pivot space, it's possible to define clustering methods to make the matching faster when considering compatible music and videos. For instance, if we define different genres of music, it's possible in the IR^p space to define a vector that's the center of the vector mass of each genre. We would then base the matching first on genre representatives, and once we obtained the best matching genre for a particular video, we could limit the aesthetic matching to that musical genre's vectors. The influence of music genre would improve the mixing algorithm.

If we incorporate musical genre into the mixing framework, then we'll need automatic genre classification to process large audio collections. This appears to be a challenging problem.[12]

After matching at the video shot and music segment level, we presented a heuristic procedure based on media continuity for the overall mixing. We could improve this procedure by developing some overall measures of optimality over and above media continuity. Thus, a second-best match for a particular shot might lead to the overall best match for the whole clip, which isn't possible with our algorithm. As a result, we need to formally pose the audio–video mixing as an optimization problem.

We could introduce special effects such as cuts and gradual transitions in the video to better match and synchronize shots with the matched audio. Thus, we could define an appropriate mixing style based on the attributed features of the mixed media.

Using gradual transitions is a well understood concept for videos, but not much work has occurred around aesthetically pleasing gradual transitions for blending two disparate audio clips.

We aim to work on many of these problems in the future. **MM**

References

1. C. Dorai and S. Venkatesh, "Computational Media Aesthetics: Finding Meaning Beautiful," *IEEE MultiMedia*, vol. 8, no. 4, Oct.–Dec. 2001, pp. 10-12.
2. J. D. Andrew, *The Major Film Theories—An Introduction*, Oxford Univ. Press, 1976.
3. S. Prince, *Movie and Meaning—An Introduction to Film*, 2nd ed., Allyn and Beacan, 2000, pp. 224-231.
4. H. Zettl, *Sight Sound Motion: Applied Media Aesthetics*, 3rd ed., Wadsworth Publishing, 1999.
5. J.Z. Wang et al., "Unsupervised Multiresolution Segmentation for Images with Low Depth of Field," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 1, Jan. 2001, pp. 85-90.
6. K. Peker, A. Divakaran, and T. Papathomas, "Automatic Measurement of Intensity of Motion Activity of Video Segments," *Proc. SPIE*, M.M. Yeung, C.-S.

Li, and R.W. Lienhart, eds., vol. 4315, SPIE Press, 2001, pp. 341-351.

7. E. Scheirer and M Slaney, “Construction and Evaluation of Robust Multifeature Speech/Music Discriminator,” *Proc. IEEE Int’l Conf. Acoustics, Speech, and Signal Processing* (ICASSP 97), IEEE Press, 1997, pp. 1331-1334.
8. E. Wold et al., “Content-Based Classification, Search and Retrieval of Audio,” *IEEE MultiMedia*, vol. 3, no. 2, Summer 1996, pp. 27-37.
9. S. Rossignol et al., “Features Extraction and Temporal Segmentation of Audio Signals,” *Proc. Int’l Computer Music Conf.* (ICMC 98), Int’l Computer Music Assoc., 1998, pp. 199-202.
10. W. Jay and D.L. Harwood, *Music Cognition*, Academic Press, 1986.
11. E.D. Scheirer, “Using Music Knowledge to Extract Expressive Performance Information from Audio Recording,” *Readings in Computational Auditory Scene Analysis*, D.F. Rosenthal and H.G. Okuno, eds., Lawrence Erlbaum, 1998.
12. G. Tzanetakis, G. Essl, and P. Cook, “Automatic Musical Genre Classification of Audio Signals,” *Proc. Int’l Symp. Music Information Retrieval* (ISMIR 01), 2001, pp. 205-210, http://ismir2001.indiana.edu/proceedings.pdf.
13. A. Bregman, *Auditory Scene Analysis*, MIT Press, 1990.
14. T. Tero and K. Matti, “Computational Efficient Multipitch Analysis Model,” *IEEE Trans. Speech and Audio Processing*, vol. 8, no. 6, Nov. 2000, pp. 708-715.
15. G.J. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic, Theory and Applications*, Prentice Hall, 1995.
16. G. Salton, *The SMART Retrieval System-Experiments in Automatic Document Processing*, Prentice Hall,1971.

Philippe Mulhem is director of the Image Processing and Applications Laboratory (IPAL) in Singapore, a joint laboratory between the French National Center of Scientific Research, the National University of Singapore, and the Institute for Infocomm Research of Singapore. He is also a researcher in the Modeling and Multimedia Information Retrieval group of the CLIPS-IMAG laboratory, Grenoble, France. His research interests include formalization and experimentation of image, video, and multimedia documents indexing and retrieval. Mulhem received a PhD and an HDR from the Joseph Fourier University, Grenoble.

Mohan S. Kankanhalli is a faculty member at the Department of Computer Science of the School of Computing at the National University of Singapore. His research interests include multimedia information systems (image, audio, and video content processing and multimedia retrieval) and information security (multimedia watermarking and image and video authentication). Kankanhalli received a BTech in electrical engineering from the Indian Institute of Technology, Kharagpur, and an MS and PhD in computer and systems engineering from the Rensselaer Polytechnic Institute.

Ji Yi is pursuing an MS at the National University of Singapore. Her research interests include digital video processing and multimedia systems. Yi received a BS in computer software from Nanjing University, China.

Hadi Hassan is a lecturer at the University of Applied Science and Technology, Amman, Jordan. His research interests include multimedia signal processing, digital audio, and image retrieval. Hassan received a BSc in electronic engineering from the University of Technology, Iraq, an MSc in computer control engineering from Baghdad University, and a PhD in image processing and pattern recognition from Shangai Jiaotong University.

Readers may contact Mohan S. Kankanhalli at the National University of Singapore, School of Computing, Singapore 119260; mohan@comp.nus.edu.sg.