MUSIC GENRE VISUALIZATION AND CLASSIFICATION EXPLOITING A SMALL SET OF HIGH-LEVEL SEMANTIC FEATURES

Giorgio Prandi, Augusto Sarti, Stefano Tubaro

Dipartimento di Elettronica e Informazione,
Politecnico di Milano
I-22100, Como, Italy
{prandi,sarti,tubaro}@elet.polimi.it

ABSTRACT

In this paper a system for continuous analysis, visualization and classification of musical streams is proposed. The system performs visualization and classification task by means of three high-level, semantic features extracted computing a reduction on a multidimensional low-level feature vector through the usage of Gaussian Mixture Models. The visualization of the semantic characteristics of the audio stream has been implemented by mapping the value of the high-level features on a triangular plot and by assigning to each feature a primary color. In this manner, besides having the representation of musical evolution of the signal, we have also obtained representative colors for each musical part of the analyzed streams. The classification exploits a set of one-against-one three-dimensional Support Vector Machines trained on some target genres. The obtained results on visualization and classification tasks are very encouraging: our tests on heterogeneous genre streams have shown the validity of proposed approach.

1. INTRODUCTION

In the last years, due to the large diffusion of digital audio contents, the need for analysis and classification tools which enable a simple cataloging, exploration and fruition of large audio databases has considerably grown. This need, particularly felt by final listeners, has been driven not only by the success of portable digital audio readers as iPod, Zen and Zune, but also by the explosion of streaming applications. In general, especially in the latter case, the content navigation is still performed using traditional modalities, by exploiting meta-tags, meta-descriptions and, somewhat, collaborative filtering. Content-driven navigation paradigms are still little exploited, and mainly as helper in hybrid navigation techniques. Unfortunately, in particular for heterogeneous streams, tags are not sufficient to describe the audio content: the user may be interested to know what currently happens on a particular stream, exploiting a simple semantic description of the related audio characteristics. Moreover, the user may be interested to see the characteristics of two or more signals, to choose and listen what he considers more interesting with respect to the description. In this connection, a graphical representation of the state of the audio signals may be useful to control and compare many streams, for example a large number of Internet radios.

In this paper a real-time analysis, visualization and classification system for audio streams is proposed. As shown in Figure 1, the system is composed of four blocks or modules: the low-level feature extraction block receives the music stream and performs a continuous extraction of sets of low-level features from consecutive small parts of the signal. Each resulting low-level feature vector $\bar{\mathbf{v}}_L$is fed into the next high-level feature extraction module, which implements three high-level semantic features by means of Gaussian Mixture Models (GMMs). Each feature is mapped on a single GMM, which performs a non-linear reduction of the low-level feature vector to a single scalar number. The three values related to the three semantic descriptors make up the high-level feature vector $\mathbf{v}_H$, which is used both for visualization and genre classification. In particular, the classification technique uses a set of binary Support Vector Machines (SVMs) [1] to detect the genre of the currently analyzed part of the stream.

The main strengths we identify in our system are the following:

- *Simplicity*: the system structure and the operations performed by each block are relatively simple;
- *Use of a small set of high-level semantic features* which enables:
  - a simple and intuitive *visualization* of the evolution of the characteristics of the audio signal through a proper rendering of the feature space;
  - the *genre classification* related to the actual content of the stream;

The paper is organized as follows. In Section 2 the works related to our system proposal are briefly described and discussed. Section 3 explains the tasks of low-level and high-level feature extraction. The Section 4 presents the method we used to show the evolution of the characteristics of the analyzed stream, and Section 5 describes the classification system. The experimental results are discussed in Section 6. Finally, Section 7 draws some conclusion remarks and describes current work-in-progress activities on improving the performance of the system.

2. RELATED WORKS

Our work can be considered as belonging to genre classification and music content visualization research areas: in particular, it tries to use a unique, semantic description of the signal to visualize and classify audio streams in real-time. In this section we list a set of works that are related to our study.

The problem of content-based music classification and visualization has been addressed in the literature under different points of view. In general, music genre classification systems tries to detect the genre of musical tracks given a specific taxonomy. Although the problem is still unsolved, many solutions have been proposed to study and improve classification performance. Given

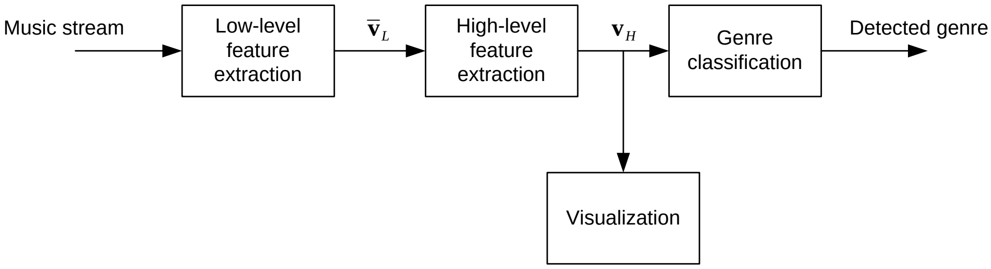


Figure 1: Proposed genre visualization and classification system

a hierarchical genre taxonomy, Tzanetakis and Cook [2] evaluate the musical genre classification task describing the audio signal through three feature sets related to timbre, rhythm and pitch of the song texture and using using different statistical pattern recognition classifiers, including GMMs. In our system Gaussian Mixture Models are used to analyze the low-level features to extract high-level descriptors, but we use SVMs to perform real-time detection of the music genre. SVM-based classifiers are exploited for example in [3] and [4]: the former, using a hierarchical representation of genres, performs detection with a multi-layer classification approach. The comparison with other methods using different pattern recognition approaches shows the superior performance of the SVM-based technique. In the latter, the classification is performed through a mixture of SVM-based experts, i.e. sets of classifiers which concentrate on different aspects of the sound. With respect to the low-level feature extraction task, our work is based on the technique proposed by Barbedo and Lopes in [5], where a classification system based on four low-level features is proposed. The classification engine detects the genre of the song by measuring distances between 12-component summary vectors, computed by considering mean, variance and main peak prevalence of each low-level feature value in a segment. The concept of high-level features, which is exploited in our system, has been used in some classification works, also for MIDI music [6]. Such descriptors are referred to rhythm, melody, harmony and other high-level characteristics of the song. In general, these characteristics have a proper, specific representation which can be difficult to quantify and to use for an homogeneous visualization. In our work, instead, the definition of different high-level features is made through a unique, visualization-oriented technique which quantify in a proper value interval the presence or the absence of a particular sound characteristic.

Under the point of view of the graphical representation of audio data, in the last years effective techniques to visualize large collections of music have been studied and proposed in the literature. In general, these techniques are based on some textual and/or content related characteristics of the songs. For example, in [7] a visualization system which uses content-based descriptors and meta-information is presented. The engine exploits Self-organizing map techniques to show plots in which similar songs are grouped together in “islands of music”. A similar approach is used in [8], where a system for organizing large music collection is described: a set of low-level audio features are extracted and aggregated to form high-level descriptors which are used to cluster and visualize music in topographic maps. In [9] the authors present Mood Cloud, an application to predict and visualize in real-time the mood of the song, subdivided in five categories. The visualization is made using a bar-graph based approach in which the bars are resized with respect to the mood probability. A bar-graph based approach is also used in [10]: this work describes a configurable system for studying music classification systems in real-time; the graphical representation is used to make results more readable and interpretable. In our system we use visualization techniques which are more similar to bar-graphs exploited in some of the reviewed works. However, instead to have a number of moving bars we are able to see the evolution of the audio stream in a more intuitive way by following the motion of a point in a 2-dimensional triangle plot or by observing the color associated to the musical status of the stream.

3. FEATURE EXTRACTION

This section is subdivided in two parts. In the first one, the extraction technique for low-level features, based on work in [5], is briefly summarized. The second part introduces the specific high-level descriptors used in our work, the method followed for their generation and the low-to-high level dimensionality reduction approach.

3.1. Low-level feature extraction

The low-level feature extraction block processes a monophonic input stream to produce a related low-level compact description. In particular, the signal waveform (in our experiments sampled at 44100 Hz, using a resolution of 16 bit per sample) is analyzed considering small, 1 sec long segments. For each segment, a further subdivision is performed, by means of 21.3 ms long, 50% overlapped frames. For each frame, the following low-level features [5] are computed:

- *Spectral roll-off* (SRO): gives the frequency under which there is the most part of the spectral energy (in our case, 95%);
- *Perceptual loudness* (PL): captures a measure of the loudness as perceived by the human auditory system;
- *Bandwidth* (BW): quantifies the bandwidth of the audio signal;
- *Spectral flux* (SF): captures the dynamic of the signal, by quantifying the quadratic difference between the logarithms of the magnitude spectra of consecutive analysis frames.

The previous process leads to the computation of about 92 four-dimensional feature vectors for the current segment. To have a more compact representation of the audio content, a summary feature vector $\bar{\mathbf{v}}_L$ belonging to the segment under analysis is computed by extracting, for each feature, the following descriptors:

- *Mean*: the mean of the feature over the entire segment;
- *Variance*: the variance of the feature over the entire segment;
- *Main peak prevalence*: a measure which quantifies the prevalence of the main peak of the feature with respect to its mean value. The value of the main peak prevalence p of the feature v_L in the current segment is given from:

$$p_{v_L} = \frac{\max\left[v_L\left(i\right)\right]}{1/I\sum_{i=1}^{I} v_L(i)} \tag{1}$$

  where i is the frame index, and I is the total number of frames contained in the audio segment.

Each resulting value is normalized using a proper coefficient (in our case we use the maximum values of the descriptors detected in the audio signals used for the generation of the high level features). Thus, the 12-dimensional vector $\bar{\mathbf{v}}_\mathbf{L}$ containing the normalized mean, variance, and main peak prevalence of SRO, PL, BW and SF is sent to the high-level feature extraction module.

3.2. High-level feature extraction

In the high-level feature extraction module a non-linear 12- to 3-dimension reduction process is applied to the summary vector $\bar{\mathbf{v}}_L$ by computing three scalar semantic descriptors values from the 12 original values of the summary vector itself. In our work, the reduction process is performed through the usage of properly trained Gaussian Mixture Models, which implement the concept of high-level features. In the following, we explain the process needed to generate a GMM related to a specific high-level descriptor and to compute the value of the descriptor itself with respect to a given summary vector.

3.2.1. Feature generation

Generating a high-level feature means to properly train the related Gaussian Mixture Model, exploiting audio signals strictly related to the meaning of the descriptor. In our work we have chosen to implement the following high-level features:

- *Classicity* (CL): a timbric feature, which tells if the current segment presents a classical sound;
- *Darkness* (DK): a feature which gives a measure of the darkness (slow changing signal with low energy on high frequencies) of the sound waveform related to the segment.
- *Dynamicity* (DY): a feature which tells if the current segment contains highly dynamic music;

For each feature, the generation process is composed of two steps, namely *configuration* and *training*:

Configuration In the configuration step, it can be decided which are the low-level features to use to train the GMM. For example, we may consider all the 12 descriptors related to the summary vector or only a subset strictly related to the meaning of the feature or to the specific goals of visualization and classification processes. In some cases, an automatic feature selection may be useful to support the configuration process. Note that, of course, this phase affects the dimensionality of the GMM.

Training Once the subset of low-level features has been selected, the training phase allows to build the GMM exploiting a set of audio signals that show characteristics belonging to the semantic meaning of the current high-level feature. The low-level extraction process applied to training streams is the same as described in Section 3.1, but only the descriptors selected in the configuration step are taken into account. The possibly reduced summary vectors are then used to generate the GMM, exploiting the Figueiredo-Jain algorithm [11] initialized with an appropriate number of components (10 may be sufficient for our purposes).

In our work, we have used 640 sec of audio signal to train each feature. The training signals have been selected by following the rules depicted in Table 1.

Feature	Training set selection rules
CL	As timbric descriptor, the *classicity* feature has been trained using classical music of post-renaissance ages.
DK	The *darkness* feature has been trained with low-pass, slowly-changing signals belonging to electronic genre.
DY	The *dynamicity* feature has been trained using some dance-style and pop patterns which present a high level of fast and substantial spectral changes.

Table 1: Rules followed to select the proper training set for each high-level feature.

3.2.2. Feature value computation

The value of a high-level feature with respect to the current analyzed segment is computed by evaluating the likelihood of the low-level vector on the GMM which implements the feature itself. In general, from the vector $\bar{\mathbf{v}}_\mathbf{L}$, the vector $\bar{\mathbf{v}}_\mathbf{L}^*$ which contains only the low-level descriptor coefficients decided during the generation of the feature is created and sent to the related GMM model for likelihood evaluation. Given the likelihood $L(\bar{\mathbf{v}}_\mathbf{L}^*; \mathcal{M})$ of $\bar{\mathbf{v}}_\mathbf{L}^*$ with respect to the current Gaussian Mixture $\mathcal{M}$, the value v_H of the high level feature is extracted as:

$$v_H = \log\left(1 + L(\bar{\mathbf{v}}_\mathbf{L}^*; \mathcal{M})\right) \tag{2}$$

The result is also normalized to have a maximum feature value of 1.

By repeating this procedure for each high-level feature, at the end of the iteration the three high-level descriptor values are used to make up the 3-dimensional high-level feature vector $\mathbf{v}_H$ related to the current segment, which will be sent to visualization and classification modules.

4. VISUALIZATION

The visualization engine maps the values of the high-level features in an appropriate visualization space. In our test we use two types of representation: triangular plot and color. Let us consider the generic high-level vector $\mathbf{v}_H(j)$ specified as

$$\mathbf{v}_H(j) = [v_{H,CL}(j) \quad v_{H,DK}(j) \quad v_{H,DY}(j)] \tag{3}$$

where $1 \leq j \leq J$ is the segment index, J is the total number of segments currently analyzed and $v_{H,CL}(j)$, $v_{H,DK}(j)$, $v_{H,DY}(j)$ are respectively the values of CL, DK and DY features related to the segment j. Assuming $j = 1$ the index of the last, currently analyzed segment, the triangular plot shows the point associated to the weighted feature vector $\tilde{\mathbf{v}}_H$ defined as

$$\begin{aligned} \tilde{\mathbf{v}}_H &= [\tilde{v}_{H,CL}(j) \quad \tilde{v}_{H,DK}(j) \quad \tilde{v}_{H,DY}(j)] = \\ &= \sum_{j=1}^{K} \frac{K-j+1}{\sum_{q=1}^{K} q} \mathbf{v}_H(j) \end{aligned} \tag{4}$$

where K is the maximum value between 30 and J. This low-pass filtering allows to regularize the evolution track of the signal. The plot is performed by computing a convex combination of the normalized components $\hat{\tilde{v}}_{H,CL}$ and $\hat{\tilde{v}}_{H,DK}$, defined as follows:

$$\begin{aligned} \hat{\tilde{v}}_{H,CL} &= \frac{\tilde{v}_{H,CL}}{\tilde{v}_{H,CL} + \tilde{v}_{H,DK} + \tilde{v}_{H,DY}} \\ \hat{\tilde{v}}_{H,DK} &= \frac{\tilde{v}_{H,DK}}{\tilde{v}_{H,CL} + \tilde{v}_{H,DK} + \tilde{v}_{H,DY}} \end{aligned}$$

Note that the triangular representation imposes $\hat{\tilde{v}}_{H,DY} = 1 - (\hat{\tilde{v}}_{H,CL} + \hat{\tilde{v}}_{H,DK})$. The position $\mathbf{t}_H = [t_{H,x} \quad t_{H,y}]$ of the point on the 2-d plane related to the current $\tilde{\mathbf{v}}_H$ is computed as:

$$\begin{aligned} t_{H,y} &= \hat{\tilde{v}}_{H,DK} \sin \pi/3 \\ t_{H,x} &= \hat{\tilde{v}}_{H,CL} t_{H,y} \cot \pi/3 \end{aligned}$$

Although the triangular plot gives a visual representation of the current position of the stream with respect to the value of the three high-level features, using colors may be a more intuitive way to show the current feel of the music signal. By mapping each of the three high-level features on a specific primary color, the color vector $\mathbf{c}_H$ related to the weighted feature vector $\tilde{\mathbf{v}}_H$ is defined as specified in Equation 5. We have associated the red (R) to DY, the green (G) to CL and the blue (B) to DK.

$$\mathbf{c}_H = [\text{R} \quad \text{G} \quad \text{B}] = [\tilde{v}_{H,DY} \quad \tilde{v}_{H,CL} \quad \tilde{v}_{H,DK}] \tag{5}$$

However, to simplify the display of the results, the color graphs presented in this paper are defined using the normalized coefficients:

$$\hat{\mathbf{c}}_H = [\text{R} \quad \text{G} \quad \text{B}] = \left[\hat{\tilde{v}}_{H,DY} \quad \hat{\tilde{v}}_{H,CL} \quad \hat{\tilde{v}}_{H,DK}\right] \tag{6}$$

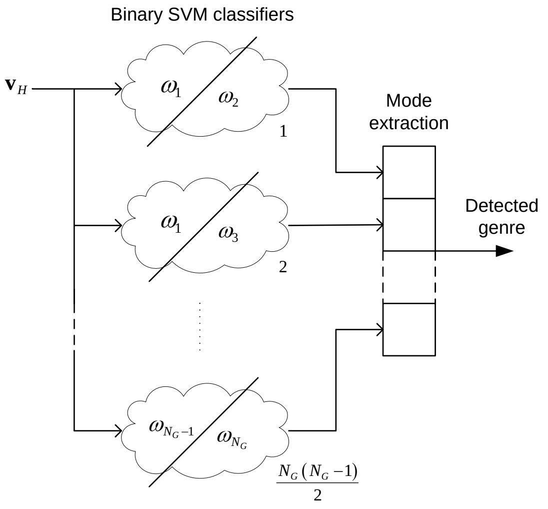


Figure 2: Block diagram of the frame-based classification module

5. GENRE CLASSIFICATION

The genre classification module uses a set of one-against-one Soft Margin SVMs to recognize the genre of the current audio segment form which the vector $\mathbf{v}_H$ has been generated. A single one-against-one SVM is a binary classifier: it tells what of the two considered classes a specific feature vector is belonging to. In our case, given a number N_G of genres to recognize, that is therefore the number of considered classification classes, $N_G(N_G - 1)/2$ binary SVMs have to be defined, one for each couple of classes. The high level feature vector $\mathbf{v}_H$ is classified by each SVM; then, a max-wins algorithm finds the final class of the current vector by selecting the most recurring class from the results given by the binary classifiers.

The general scheme of the frame-based classification module is shown in Figure 2: the vector $\mathbf{v}_H$ is fed into each of the $N_G(N_G - 1)/2$ binary classifiers, which work in parallel (in the block diagram, the generic class ω_g, with $1 \leq g \leq N_G$, is associated to the genre g); the result of each classification is then used to detect the mode of the current classification task performed by SVMs; the class corresponding to the mode of the classification results is given as the detected genre for the current segment.

If two genres g_A and g_B have the same number of occurrences in the classification results, the system detects as current genre the winner of the head to head classification g_A vs. g_B. If the same situation is detected on more than two genres, the system chooses the winner randomly between the genres that have the same, maximum number of occurrences.

5.1. Training of SVM classifiers

In our system, the training of each SVM binary classifiers has been performed by exploiting the Radial Basis Function kernel defined as

$$K(\mathbf{x}_n, \mathbf{x}_m) = e^{-\lambda \|\mathbf{x}_n - \mathbf{x}_m\|^2}, \quad \lambda > 0 \tag{7}$$

where $\mathbf{x}_n$ and $\mathbf{x}_m$ are two data instances, computing 80/20 cross-validation and performing a grid search on the penalty parameter C [1] and on the kernel parameter λ to find the best classifier configuration with respect to classification accuracy. The high-level training vectors are generated from homogeneous genre streams by following the techniques explained in Section 3. Feature vectors belonging to two different genres are then used to train the related SVM. For each genre, the training high-level feature vectors have been extracted from a total of 640 sec of audio data.

6. EXPERIMENTAL EVALUATION

In this section the results of our tests are presented and discussed. After having explained the experimental setup, in the Section 6.2 the graphic representations obtained using a set of test signals of different music genres are described. The Section 6.3, instead, presents a first evaluation on classification performance of our system.

6.1. Experimental setup

The main test phase for the visualization task has been conducted by considering a single, 15 min (900 segments) test stream composed of five pieces of music belonging to Dark Ambient (*da*), Baroque (*br*), New Age (*na*), Dance (*dn*) and Solo Piano (*sp*) genres. The same music genres have been used for the evaluation of the classification task but considering them in five separated 7 min (420 segments) homogeneous genre streams.

In general, we have used the following two high-level feature configurations, with respect to the *Feature generation* step described in Section 3.2.1:

- M_1: all the low-level features belonging to the feature summary vector are used to train and test the high-level features;
- M_2: the assignment of low-level features to high level descriptors is partial, as specified in Table 2. In particular, for CL, which is a timbric feature, we have chosen to select all the 12 low-level descriptors; for DK, we have chosen features related to the description of spectral changing speed and signal bandwidth; for DY all the indicators of "movement" has been selected. The resulting dimensionality for the GMMs is 12 for CL and 9 for DK and DY.

Feature	Low-Level features assigned
CL	All the low-level features.
DK	SRO: all values; BW: all values; SF: all values.
DY	SRO: variance and main peak prevalence; PL: variance and main peak prevalence; BW: variance and peak prevalence; SF: all values.

Table 2: Rules followed to perform the association between low- and high-level features for the M_2 testing configuration.

6.2. Visualization results

The visualization results obtained using the stream described in experimental setup section with M_1 configuration are shown in Figure 3. In particular, in Figure 3(a) the triangular plot describes the audio signal evolution with respect to the three high-level feature used. The clusters associated to the five music genres are clearly visible. The tracks between clusters represent the transitions between one genre to another, and are caused by the low-pass filtering technique used in visualization algorithm. As can be noticed, there are four genres on the center/right of the triangle: in these cases there is a significant contribution, although in different measures, of darkness and classicity features. The dynamicity feature instead is very important for Dance genre. In particular, in this case the value of the other descriptors is almost zero. By mapping the high-level feature vector on colors exploiting the relation (6), we have obtained, for each genre, the average results shown in Figure 3(b).

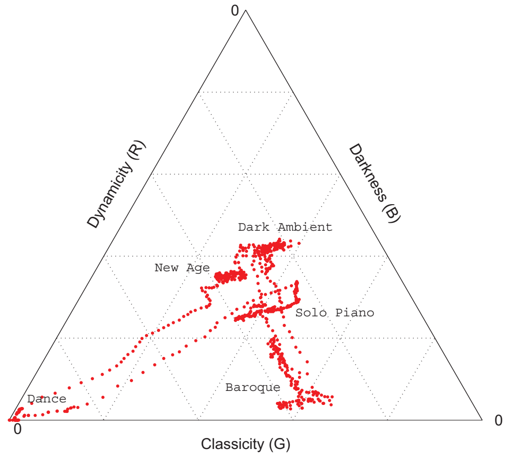
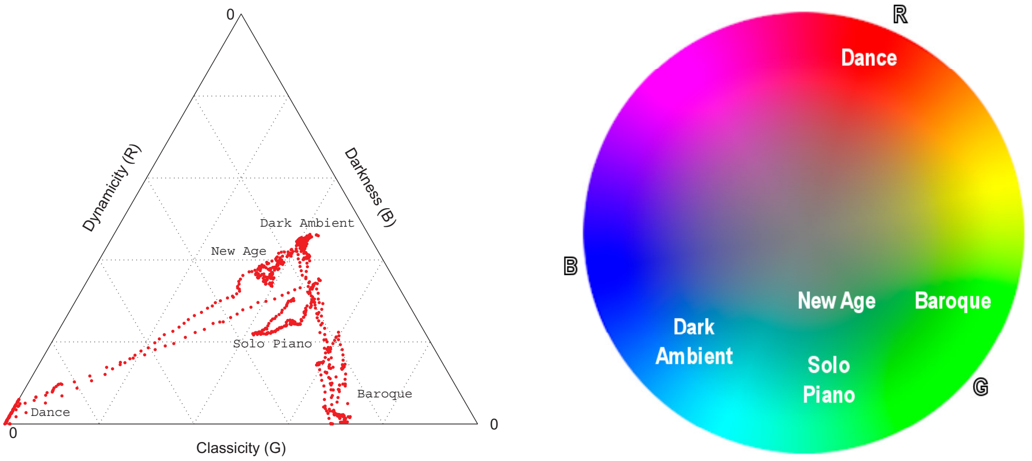


Figure 4: Triangular plot visualization of high-level features obtained through the analysis of the heterogeneous music stream by adopting the configuration M_2.

By adopting the configuration M_2 on the same heterogeneous stream we have obtained the plot shown in Figure 4. Not considering some low level features in the configuration of DK and DY leads to a slightly more clear definition of clusters and trajectories. In this case DK and DY do not consider a complete description of the timbre, but only a partial one which takes into account only the low-level features which are related to the semantic meaning of the high-level feature. This leads to have DK and DY more uncorrelated from the entire representation of the timbre of the signals. In addition, a general translation to the left edge of the triangle can be noticed, caused by the increasing values of the DY feature for all the genres. These phenomena are clearly visible also in Figure 5, where the audio signal evolution related to the famous italian pop-genre track "Almeno tu nell'universo" has been drawn. We can observe that the M_2 configuration may conduct to a more compact but still significant representation of the audio signal, with a general left-side translation due to the increasing importance of DY.

6.3. Classification results

To test the classification performance, we used the set of five homogeneous genre streams described in Section 6.1. The classification performed on each single vector $\mathbf{v}_H$ allows to assess the representativeness of the high level vectors with respect to the considered music genres. Tables 2(a) and 2(b) present accuracy results respectively for M_1 and M_2 configurations, arranged as confusion matrices. As reference, in Table 2(c) we have reported the accuracy values obtained using a more traditional classification system



(a) Resulting triangular plot for mixed genre stream test.

(b) Resulting average genre colors for mixed genre stream test.

Figure 3: Visualization of high-level features obtained through the analysis of the heterogeneous music stream by adopting the configuration M_1.

based on the scheme in Figure 2 but trained and tested using low-level summary vectors directly.

As can be noticed, the classification results for *da*, *br*, *dn* and *na* in M_1 with respect to the low-level based classification system are good, especially for *da* and *br*: in this case the Darkness and Classicity high-level features work very well to enable a correct detection of the genre. By considering the configuration M_2 we can notice the good results obtained on *da* and *na*; also the accuracy of *dn* raises with respect to M_1 configuration but remains under the result of the traditional classifier. The performance on *br* suffers the increasing value of the Dynamicity in its high-level feature vectors: this leads in some cases to confuse the Br genre with *na*.

The result of Solo Piano detection is very insufficient in all the cases and our system perform even worse of the traditional one. The overall unsatisfactory performance is related to the particular behavior and characteristics of the Piano Solo songs, which causes the feature vectors to span over a large portion of the feature space.

(a) Confusion matrix for classification accuracy on audio segments using M_1 configuration.

	da	*br*	*dn*	*na*	*sp*
da	66.26	0.97	0	14.08	18.69
br	3.21	91.11	0	5.68	0
dn	0	0	82.47	17.53	0
na	17.76	1.93	4.63	63.32	12.36
sp	57.69	14.53	0	5.13	22.65

(b) Confusion matrix for classification accuracy on audio segments using M_2 configuration.

	da	*br*	*dn*	*na*	*sp*
da	75.21	2.38	0.17	22.07	0.17
br	3.49	81.20	0.17	15.14	0
dn	0	0.21	88.68	11.11	0
na	9.27	1.93	1.93	86.49	0.39
sp	78.63	5.98	4.27	2.14	8.97

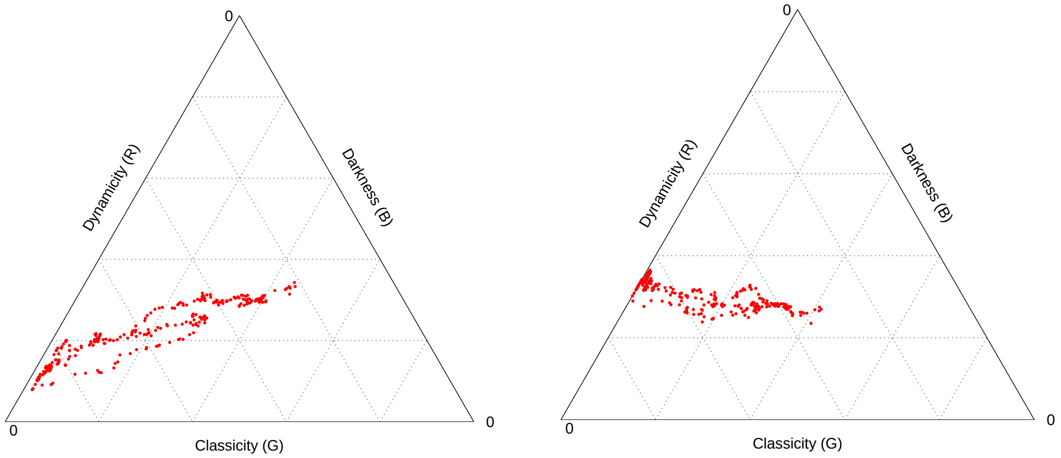
(c) Confusion matrix for classification accuracy on audio segments using a classic system based on low-level features.

	da	*br*	*dn*	*na*	*sp*
da	65.20	11.71	0.17	19.35	3.57
br	10.32	89.68	0	0	0
dn	0.41	0.61	98.78	0.20	0
na	10.04	7.72	1.93	79.54	0.77
sp	55.98	4.27	3.42	0	36.32

Table 3: Percentage accuracy results for all the tested configurations.

7. CONCLUSIONS

In this paper a new system for music genre visualization and classification has been proposed. The system exploits a small number of high-level semantic features which are used to show the current characteristics and evolution of the music stream to the user in a user-friendly way, in our case by using colors or simple triangular plots. Moreover, the same high-level feature set is used to perform genre classification on the stream. The preliminary results presented in the paper have confirmed the validity of the proposed method on both visualization and classification. In particular, in some cases, the classification task outperforms a standard SVM-based classifier based on low-level features. We are currently studying problems, issues and results with wider data sets and high-level features and we are trying to improve the genre classification by applying some filtering techniques to the high-level feature vectors and to genre-detection results. For enhancing the association between features and colors, we are considering to follow indications given from studies as [12].



(a) Resulting triangular plot for "Almeno tu nell'universo", using M_1 configuration.

(b) Resulting triangular plot for "Almeno tu nell'universo", using M_2 configuration.

Figure 5: An example of a single-track evolution computed on "Almeno tu nell'universo" by M. Martini using the two configurations M_1 and M_2.

8. REFERENCES

[1] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[2] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on speech and audio processing*, vol. 10, no. 5, pp. 293–302, 2002.

[3] C. Xu, NC Maddage, X. Shao, F. Cao, and Q. Tian, "Musical genre classification using support vector machines," in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03)*, 2003, vol. 5.

[4] N. Scaringella and D. Mlynek, "A mixture of support vector machines for audio classification," *1 st Music Information Retrieval Evaluation Exchange (MIREX)*, 2005.

[5] J.G.A. Barbedo and A. Lopes, "Automatic genre classification of musical signals," *EURASIP Journal on Applied Signal Processing*, vol. 2007, no. 1, pp. 157–157, 2007.

[6] C. McKay and I. Fujinaga, "Automatic genre classification using large high-level musical feature sets," in *Proceedings of the International Conference on Music Information Retrieval*, 2004, vol. 525, p. 30.

[7] E. Pampalk, S. Dixon, and G. Widmer, "Exploring music collections by browsing different views," *Computer Music Journal*, vol. 28, no. 2, pp. 49–62, 2004.

[8] F. Morchen, A. Ultsch, M. Nocker, and C. Stamm, "Databionic visualization of music collections according to perceptual distance," in *Proc. of the 6th International Conference on Music Information Retrieval (ISMIR05), London, UK*, 2005.

[9] C. Laurier and P. Herrera, "Mood Cloud: A Real-Time Music Mood Visualization Tool," *Computer Music Modeling and Retrieval*, 2008.

[10] K. West, J.S. Downie, X. Hu, and M.C. Jones, "Dynamic visualization of music classification systems," in *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM New York, NY, USA, 2008, pp. 888–888.

[11] M.A.T. Figueiredo and A.K. Jain, "Unsupervised learning of finite mixture models," *IEEE Transactions on pattern analysis and machine intelligence*, pp. 381–396, 2002.

[12] R. Bresin, "What is the color of that music performance," in *proceedings of the International Computer Music Conference-ICMC*, 2005, pp. 367–370.