Department of
Computational
Perception

Improvement of automatic music detection in TV productions using rhythm features

February 29, 2008

Arnaud Moreau
a.moreau@gmx.net
0325440
E 066 932

Contents

1 Introduction

This paper is the final report of an internship performed at the department of computational perception at the Johannes Kepler University Linz under the supervision of Klaus Seyerlehner and Professor Gerhard Widmer. In a project proposed to the department of computational perception the Austrian National Broadcasting Corporation (ORF) wants to automatically determine the amount of music in the sound track of a TV production, no matter if it is played in the foreground or background, in order to determine royalty fees. This is far more difficult than pure speech/music discrimination. Klaus Seyerlehner and Gerhard Widmer have already developed a novel, very powerful feature (Continuous Frequency Activation - CFA) described in [1]. They want to find out if their results can be improved by analyzing the rhythmical content in the audio data.

1.1 Task Description

The steps guiding the internship are defined as follows:

- Incorporate the CFA Feature [1] into the HR Framework.
- Design rhythmic descriptors which improve classification performance.
- Test and evaluate the new features using the HR Framework.
- Test and evaluate the use of spectral flux as a feature.

1.2 Related Work

A lot of research has been done so far concerning the separation of pure speech and pure music audio data (referred to as speech/music discrimination). It is obvious that the problem adressed in [1] is more complicated than pure music/speech discrimination, because music is generally mixed with speech or environmental sounds in TV productions. Work concerning speech/music discrimination can be found in [2], [3] and [7]. But there have also been attempts to use audio information in videos in order to solve video related problems like segmentation [4] or indexing [5], [6].

Concerning the rhythm aspect of music, the spectral flux feature is widely used in speech/music discrimination [3], [2] or content-based audio classification like in Lu et. al [8] who consider 5 classes: silence, music, background sound, pure speech and non-pure speech. Rhythm analysis meaning autocorrelation algorithms has also been used for speech/music discrimination [11] or music information retrieval [9], [10].

2 Methods

In this section the HR framework and newly developed algorithms used to achieve the goals of this work are described.

2.1 The Framework

The HR Framework has been developed in MATLAB (The MathWorks Inc.) at the department of computational perception and provides tools to carry out experiments in the domain of music detection. Machine learning tasks (classification) are carried out by WEKA [12], an open source machine learning environment, that is interfaced via MATLAB. Before using the HR Framework the system paths (audio database, WEKA, etc.) have to be specified in `hr_default_params.m`. For testing the framework the program `labelmusic.m` is a good starting point. One can choose a classifier from a list and perform music detection on a set of input audio files. The reference label file (if there is one) and the result labels from the classifiction are visualized underneath the wavform.

2.1.1 Generating training data

The code shown below demonstrates the generation of training data in the HR framework. `hr_getallfeatures.m` returns the list of available features, it can also be used to change feature extraction parameters like window sizes. If not all features are desired, their respective fields have to be deleted manually from the structure `feat`. The trainingdata is extracted from the training database using the function `hr_generatetraining.m` and a file in WEKA's arff format is created automatically.

```
feat = hr_getallfeatures;
openvar('feat');
hr_generatetraining(feat, true, 0, 0, 34*3, 0, 'CFA_SF_RHY.arff');
```

2.1.2 Building an audio classifier

The classifier is generated by using the function `hr_createaudioclassifier.m`, its name can be specified in the argument. Again the features have to be specified by first calling `hr_getallfeatures.m` and then deleting the undesired ones. The type of classifier is chosen using `hr_createclassifier.m`. All classifier algorithms available in WEKA are supported in addition to Gaussian Mixture Models (GMM) using the NETLAB Toolbox. After training using the generated arff data file from section 2.1.1 the confusionmatrix and the percentage of correctly classified instances of the trainingset is displayed. The newly created classifier is going to show up in the GUI after calling `hr_saveaudioclassifier.m`.

```
ac = hr_createaudioclassifier('CFA_SF_RHY_SMO');
ac.featdesc = hr_getallfeatures;
openvar('ac.featdesc');
ac.classifier = hr_createclassifier(...
    'weka.classifiers.functions.SMO');
[ac.classifier, pred] = hr_trainclassifier(ac.classifier,...
    'CFA_SF_RHY.arff', 'CFA_SF_RHY.arff');
confusionmat = hr_confusionmatrix(pred, 2)
sum(diag(confusionmat))/sum(sum(confusionmat))
hr_saveaudioclassifier(ac);
```

2.1.3 Evaluating an audio classifier

The experiments in this paper are carried out using the ”classify batch” function in `labelmusic.m`. The thereby generated label files can be compared to the reference label files using the program `evaluateQuality.m`.

```
labelmusic;
evaluateQuality;
```

2.2 Incorporating CFA

The novel CFA feature is integrated into the framework by creating the feature extraction file `hr_cfa.m`. Little technical changes are performed on the original feature extraction algorithm provided by Klaus Seyerlehner. The first issue concerns the size of the analysis windows. The CFA feature is designed to work on segments of 2.6 s length. The Framework however differentiates 2 sizes of analysis windows. The small window size (usually 1024 samples = 46.44 ms at 22,050 Hz sampling frequency) is used to extract one feature vector that is aggregated in the large window using mean and standard deviation. Those aggregated values are then used for classification. For the CFA feature the parameters in `hr_getallfeatures.m` are set as follows:

small window size 2048 samples (92.88 ms)

small hop size 512 samples (23.22 ms)

large window size 52,736 samples (2.39 s)

large hop size 25,600 samples (1.16 s)

The large window and hop size results from using 100 frames to form the large window. The parameters for feature extraction are then set to compute mean values (no standard deviation), where the feature extraction procedure assignes the same value to every 100 frames in one large window. The new feature extraction algorithm is then registered to the functions `hr_extractfeatures.m` and `hr_check_feat_params.m`.

The following changes are applied to the CFA feature extraction algorithm as described in [1]:

- The eight largest peak values are summed to quantify the overall "peakiness" of the activation function.
- The final CFA feature value is divided by the frame size (usually 100) to be able to compare results from different frame sizes.

2.3 The Beat Spectrum

The idea is to use the Beat spectrum described in [9] to measure the rhythmical content in the audio. First a parameterization of the downsampled audio data is computed (11 kHz), resulting in a sequence of feature vectors. In this case a simple log power spectrogram (window size 1024 samples and hop size 256 samples) is used. One could also use MFCCs or other psychoacoustically relevant representations. For each block of 200 feature vectors a similarity matrix using the cosine distance given by

$$s_{cos}(\mathbf{x_i}, \mathbf{x_j}) = \frac{\mathbf{x_i}^\mathbf{T} \mathbf{x_j}}{\sqrt{\mathbf{x_i}^\mathbf{T} \mathbf{x_i}} \sqrt{\mathbf{x_j}^\mathbf{T} \mathbf{x_j}}} \tag{1}$$

is computed. The normalization term is used to achieve independence from energy. In Fig. 1 such a similarity matrix is presented where the music in the background has a strong beat and is covered by applause noise. This can be seen as white stripes parallel to the main axis. On this similarity matrix the autocorrelation using 2-dimensional FFT is computed and summed along one dimension resulting in the beat spectrum. In Fig. 2 the periodicity of the above mensioned example can clearly be seen in the beat spectrum.

2.4 The new rhythm features

From the beat spectrum the maxima (peaks) are extracted in order to compute the standard deviation (first feature) and the maximum of all peaks (second feature). This results in a 2-dimensional feature vector for a 4.78 s audio block. Fig. 2 demonstrates the extracted peaks. The second feature value would in that case correspond to the height of the fifth peak.

The feature values computed from a larger audio excerpt are shown in Fig. 3. The second subplot shows the reference labels (1 for music and 0 for

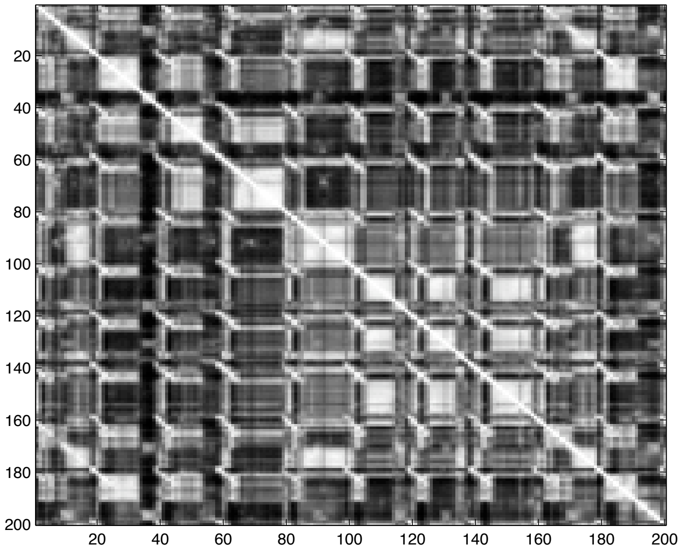

Figure 1: This figure shows the similarity matrix of the intro music of the talk show "Barbara Karlich Show", which is covered by applause. The periodicity due to the strong beat can be seen in the stripes parallel to the main diagonal axis.

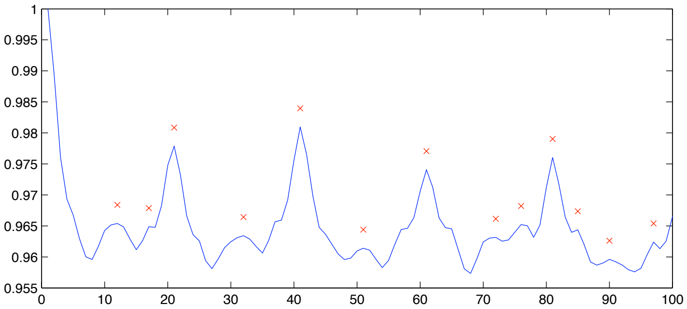

Figure 2: The periodicity of this music excerpt can be seen by the peaks in the beat spectrum. The extracted maxima (red) are also displayed.

no music) and the third and forth subplot display the rhythm feature values (standard deviation of peaks and maximum peak, respectively). When rhythm is present the maximum peak value should be near 1 and otherwise tends to be lower. The standard deviation on the contrary is low when there equally distributed peaks and high when there is noise.

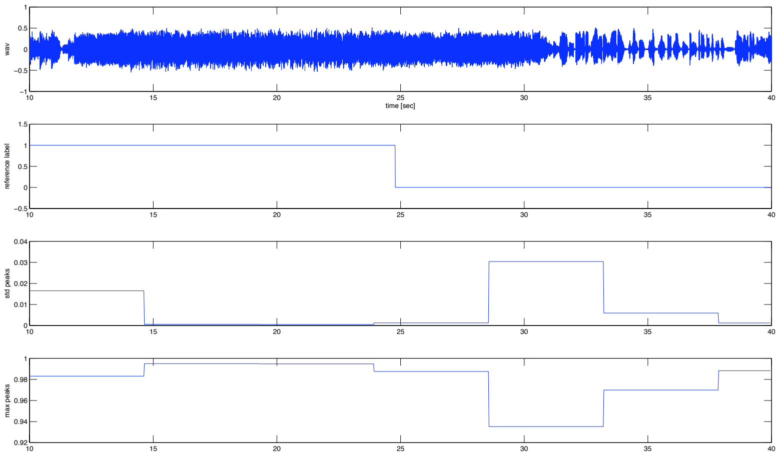

Figure 3: Audio excerpt from "Barbara Karlich Show.wav" (from 10 s to 40 s) together with rhythm feature values and reference labels.

2.5 Spectral Flux

The computation of the spectral flux is demonstrated in the following code snippet:

```
function [ sf ] = hr_sf( wav, sfp )

    X = spectrogram(wav,sfp.win,sfp.win-sfp.hop);
    dX = diff(abs(X),2);
    HdX = (abs(dX) + dX)/2;
    sf = sum(HdX,1);

    sf = filter(ones(1,sfp.frame_size)/sfp.frame_size,1,sf);

end
```

The spectral flux has been implemented as proposed in [13]:

$$SF(n) = \sum_{k=1}^{N} H(|X(n,k)| - |X(n-1,k)|) \tag{2}$$

N denotes the number of frequency bins, $H(x) = \frac{x+|x|}{2}$ is the half-wave rectifier function and $X(n, :)$ is the fourier transformed audio window at time n. A running average filter has been applied to eliminate the high variability of spectral flux values.

As stated in [2] music has a higher rate of change and goes through more drastic frame-to-frame changes than speech does, so the spectral flux values are higher for music than for speech. This can also be seen in Fig. 4 where an audio excerpt from "Barbara Karlich Show" together with reference labels and the spectral flux values are shown. The ascent of the spectral flux values at the beginning are due to the long window size of the running average filter. The spectral flux values remain at the same height after the music stops because the audience applause continues. Another example from "ZIB" is shown in Fig. 5. The higher values at 300 s are also due to applause noise.

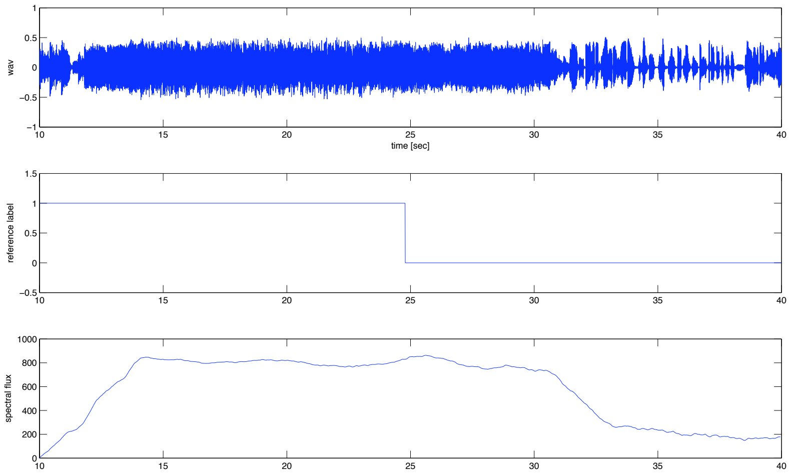

Figure 4: Audio excerpt from "Barbara Karlich Show.wav" (from 10 s to 40 s) together with spectral flux values and reference labels.

2.6 Classifier

In all experiments carried out to measure the accuracy of the newly developed features the same classifier, namely SMO is used. It implements John Platt's sequential minimal optimization algorithm for training a support vector classifier described in [14].

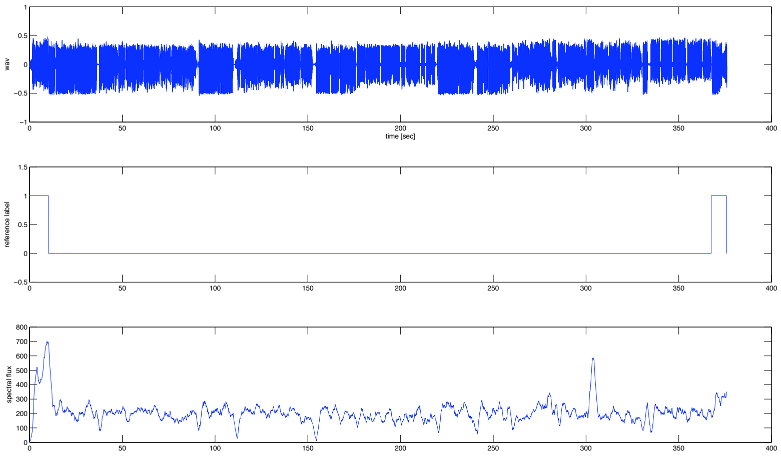

Figure 5: "ZIB.wav" together with spectral flux values and reference labels.

3 Results

The results are presented in Tab. 1 and Tab. 2 (with and without postprocessing, respectively). The second column shows the percentage of music in the TV shows of the test set. The following columns show the results evaluated on different feature subsets (CFA alone, CFA and spectral flux, CFA and rhythm features and spectral flux, respectively). The last row lists the absolute average error, computed as

$$\frac{1}{10}\sum_{i=1}^{10}|real_i - estimated_i|. \tag{3}$$

4 Discussion

The first thing that jumps to the eye is that the results from CFA alone are better than reported in [1]. This could be due to the slight changes in the feature extraction process (see 2.2) or the different classifier. Of couse an average error of 2.89% is a really high baseline for improvements. The most important question that has to be investigated is: Can automatic music detection in audio streams of TV productions be improved by using rhythm features? The answer according to the presented result tables is yes. Adding spectral flux to the feature set reduces the average error by 0.10%, adding the new rhythm features subtracts another 0.20%. Apparently adding spectal flux together with the new rhythm features does not improve the results anymore.

Title	% real	% est. CFA	% est. +SF	% est. +Rhy	% est. +SF+Rhy
Alpen Donau Adria	57.08	51.08	52.42	53.93	53.86
Barbara Karlich Show	7.51	7.43	6.59	7.43	6.38
Da wo es noch Treue gibt	62.90	63.74	63.83	64.07	63.65
Frisch gekocht	10.01	6.20	6.20	6.69	6.20
Gut beraten Österreich	8.77	6.99	6.99	6.99	6.99
Heilige Orte	54.34	49.22	49.85	49.82	49.86
Heimat fremde Heimat	29.72	23.52	24.01	24.43	23.92
Hohes Haus	17.50	12.52	12.52	13.24	12.55
Julia	80.36	79.33	79.88	79.77	79.86
ZIB	4.92	2.74	2.74	2.90	2.74
absolute average error (in %)		3.20	2.99	2.62	2.86

Table 1: Results with postprocessing

Title	% real	% est. CFA	% est. +SF	% est. +Rhy	% est. +SF+Rhy
Alpen Donau Adria	57.08	49.21	49.76	50.21	49.76
Barbara Karlich Show	7.51	8.29	7.79	8.42	7.37
Da wo es noch Treue gibt	62.90	62.07	62.17	62.24	62.04
Frisch gekocht	10.01	8.58	7.81	9.34	8.06
Gut beraten Österreich	8.77	8.85	8.95	9.05	9.10
Heilige Orte	54.34	48.66	48.93	49.54	49.13
Heimat fremde Heimat	29.72	23.75	23.82	23.89	23.79
Hohes Haus	17.50	14.16	14.17	14.49	14.42
Julia	80.36	77.57	78.03	77.66	77.86
ZIB	4.92	4.76	4.76	4.91	4.76
absolute average error (in %)		2.89	2.78	2.57	2.75

Table 2: Results without postprocessing

4.1 Smoothing

The following smoothing routines have been used when generating Tab. 1:

```
silence = 0;
settings.threshold = 0.5;
settings.region_size = 5;

labels_smooth = hr_postproc_iterative_smooth(labels, settings, ...
    silence);

settings.min_music = 10;
settings.min_speech = 10;
settings.speechfirst = true;
labels_smooth = hr_postproc_removeshort(labels_smooth, settings, ...
    silence);
```

The first method uses a window of 5 s size to change labels in regions smaller than the window according to neighboring values. The second method removes short regions smaller than 10 s. In Fig. 6 the result of this postprocessing algorithm is demonstrated. Without postprocessing the percentage of music is slightly overestimated, but with the strong reduction of "music" regions the error increases.

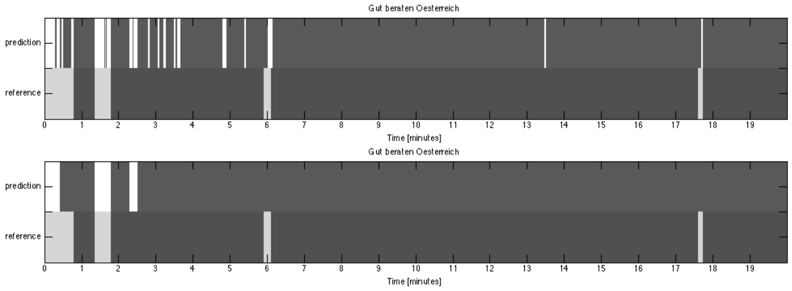

Figure 6: The TV show "Gut beraten Österreich" labelled without (first subfigure) and with postprocessing (second subfigure). The first line of either subplot shows the predicted labels, the second line shows the reference labels. Dark regions mean "no music".

4.2 Future work

The quest for new powerful features is still not finished. Other representations of audio such as chroma-vector have not been investigated so far, as they could carry strong information about the presence of music or harmony in an audio stream. Further experiments could be carried out to find more suitable classification methods or parameters thereof. The newly developed rhythm features based on the beat spectrum could further be investigated to test different computation methods: parameterization of audio data, block size, or completely other features extracted from the beat-spectrum. Also postprocessing methods can be further refined.

5 Conclusions

The most obvious conclusion that can be drawn from the results presented in section 3 is that the CFA feature from [1] is already so powerful, that it leaves only little space for improvement. Nevertheless rhythm analysis does not harm nor destroy the power of CFA, on the contrary - the usefulness has been shown in this paper.

References

[1] Klaus Seyerlehner, Gerhard Widmer, Tim Pohle, and Markus Schedl. Automatic music detection in television productions. In *Proceedings of the Int. Conf. on Digital Audio Effects (DAFx-07)*, Bordeaux, France, 2007.

[2] E. Scheirer and M. Slaney. Construction and evaluation of a robust multifeature speech/music discriminator. In *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '97)*, volume 2, page 1331, Washington, DC, USA, 1997. IEEE Computer Society.

[3] M. Kashif Saeed Khan, Wasfi Al-Khatib, and Muhammad Moinuddin. Automatic classification of speech and music using neural networks. In *Proc. of the 2nd ACM International Workshop on Multimedia Databases ACM-MMDB 2004*, pages 94–99, Washington DC, USA, November 2004. ACM.

[4] Massimo De Santo, Gennaro Percannella, Carlo Sansone, and Mario Vento. Classifying audio of movies by a multi-expert system. In *Proc. of the 11th International Conference on Image Analysis and Processing (ICIAP 2001)*, pages 386–391, Palermo, Italy, September 2001.

[5] Kenichi Minami, Akihito Akutsu, Hiroshi Hamada, and Yoshinobu Tonomura. Video handling with music and speech detection. *IEEE MultiMedia*, 05(3):17–25, 1998.

[6] Kenichi Minami, Akihito Akutsu, Hiroshi Hamada, and Yoshinobu Tonomura. Enhanced video handling based on audio analysis. In *Proc. of the IEEE International Conference on Multimedia Computing and Systems '97*, pages 219–226, Ottawa, Canada, June 1997.

[7] J. Mauclair and J. Pinquier. Fusion of descriptors for speech/music classification. In *Proc. of the 12th European Signal Processing Conference (EUSIPCO '04)*, Vienna, Austria, 2004.

[8] Lie Lu, HongJiang Zhang, and Stan Z. Li. Content-based audio classification and segmentation by using support vector machines. *Multimedia Systems*, 8(6):482–492, 2003.

[9] Jonathan Foote and Shingo Uchihashi. The beat spectrum: A new approach to rhythm analysis. *ICME*, 00:224, 2001.

[10] Jonathan Foote, Matthew D. Cooper, and Unjung Nam. Audio retrieval by rhythmic similarity. In *Proc. of the 3rd International Conference on Music Information Retrieval (ISMIR 2002)*, Paris, France, October 2002.

[11] Roman Jarina, Noel O'Connor, Seán Marlow, and Noel Murphy. Rhythm detection for speech-music discrimination in mpeg compressed domain. In *Proceedings of the lth International Conference on Digital Signal Processing (DSP 2002)*, 2002.

[12] Ian H. Witten and Eibe Frank. *Data Mining: Practical machine learning tools and techniques.* Morgan Kaufmann, San Francisco, 2005.

[13] Simon Dixon. Onset detection revisited. In *Proc. of the Int. Conf. on Digital Audio Effects (DAFx-06)*, pages 133–137, Montreal, Quebec, Canada, Sept. 18–20, 2006.

[14] John C. Platt. Fast training of support vector machines using sequential minimal optimization. *Advances in kernel methods: support vector learning*, pages 185–208, 1999.