Sampo Vesa. 2007. The effect of features on clustering in audio surveillance. In: Proceedings of the AES 30th International Conference on Intelligent Audio Environments. Saariselkä, Finland. 15-17 March 2007. 10 pages.

The Effect of Features on Clustering in Audio Surveillance

Sampo Vesa

Helsinki University of Technology, Espoo, Finland

Correspondence should be addressed to Sampo Vesa (sampo.vesa@tml.hut.fi)

ABSTRACT

The effect of the choice of features on unsupervised clustering in audio surveillance is investigated. The importance of individual features in a larger feature set is first analyzed by examining the component loadings in principal component analysis (PCA). The individual sound events are then assigned into clusters using the self-tuning spectral clustering and the classical K-means algorithms. A weighted version of the original set is used, where the weights have been optimized by a genetic algorithm (GA) for maximally error-free clustering. The weighted feature set expectedly outperforms the original feature set and its PCA-reduced version. Insight into the importance of individual features is also gained.

1. INTRODUCTION

Audio surveillance refers to techniques for monitoring the environment based on sound. Audio surveillance is based on a signal model which assumes that the observed microphone signal contains more or less stationary background noise and additive short-duration audio events. Typically, the sound stream is segmented online and interesting events are stored to the disk. The analysis of audio events can take place either online or offline at a later time, depending on the application. Audio surveillance has applications in, e.g., security [1] [2] and telemedicine [3].

The analysis of the sound events collected in audio surveillance is conducted by pattern recognition techniques. Depending on the application, supervised or unsupervised recognition is carried out. The former is suitable when the concern is on detecting particular sound classes, such as speech, screaming or gunshots, from the sound stream. A set of representative sounds of each class is needed for training the system. In this study, the unsupervised approach is adopted, i.e., there are not predefined classes and the goal is to group similar sounds together into clusters. However, in the current study speech sounds are not included, since they are assumed to be detected separately by voice activity detection (VAD) techniques. Ruling out speech sounds was necessary due to the diversity they exhibit, which turned out to be problematic in automatic clustering.

The goal of this study is to investigate the effect of the chosen features on clustering of sound events. Similar studies on environmental sound recognition (ESR) exist [4] [5], but in that application area, the analyzed signals are longer recordings of, e.g., crowd or traffic noise, and discrete sound events are not analyzed separately. Also, the classification is usually done in a supervised manner into discrete sound categories, which is the case also in [4] and [5]. In previous studies on unsupervised audio surveillance, MFCC features [1] and FFT features [2] have been used. The selection of features in unsupervised audio surveillance is briefly addressed in [6], where a set of ten parameters is chosen from a larger candidate set by analysis of covariances. The performance of the K-means clustering algorithm compared to manual clustering is evaluated by cluster compactness and the discriminative performance of single features is assessed.

The system described in this paper consists of a sound event detection and storing stage, in which a continuously recorded sound signal is analyzed online (using an algorithm described in [6]) and the sound events the spectrum of which deviates from the background sound are stored to the hard disk. The individual sound events are later analyzed offline and clustered together using the basic K-means algorithm [7] and the self-tuning spectral clustering algorithm [8], the latter of which, to the knowledge of the author, has not been used in audio surveillance previously. PCA-based factor loading analysis with Varimax rotation [9] is conducted to investi-

gate the mutual correlation between features, as well as the amount of data variance they explain. Performance of the clustering is investigated using a manually labeled data set collected in an office (see Sec. 5). Finally, the effect of features on unsupervised clustering performance is measured with different feature sets. Evaluation of the performance is based on a manual labeling of the classes. An evolutionary algorithm is used to compute optimal weights for the largest feature set that result in best clustering (with K-means) in terms of the labeling.

2. DETECTION OF AUDIO EVENTS

The first stage in audio surveillance is to segment the discrete sound events from a continuous stream of audio. It is usually assumed that the sound environment is relatively sparse, i.e., consisting of discrete sound events that do not significantly overlap in time. By following an idea from video surveillance, the sound environment can be divided into *background* and *foreground* [10], the former of which consists of a relatively stationary sound, such as air conditioning or hum of traffic, while the latter consists of discrete sound events.

In order to be able to detect the individual sound events, a model for the background sound is needed, as well as a criterion by which significant deviations from the background can be detected. Again, following the ideas of video surveillance [10], the background can be modeled using a time-adaptive mixture of Gaussians [11] [2]. By using more than one Gaussian, multimodal distributions can be modeled and the background model is able to capture recurring deviations in the background.

In this work, a simplified background model [6] is used. The spectrum of the background is approximated by time-averaging the power spectrum and considering large enough deviations from the background as sound events. Härmä et al. [6] present two different detectors based on an estimate of the background noise power spectrum[1]

$$\begin{aligned}\langle S(n,t)\rangle &= (1-\gamma)|S(n,t)|^2+\gamma\langle S(n,t-1)\rangle \\ &\quad \forall n=0,\ldots,N-1 \end{aligned} \tag{1}$$

where $\langle\cdot\rangle$ denotes smoothing over time, realized by a first order IIR[2] in this case, $S(n,t)$ is the complex spectrum at discrete frequency n and frame index t and $\gamma\in[0,1]$ is a forgetting factor that determines the amount of smoothing.

The first detector is designed to detect transients and loud onsets. It is realized as the full-band difference to noise [6]

$$\begin{aligned}\mathrm{T1}_{dB} &= 20\log_{10}\frac{\sum_{n=0}^{N-1}|D(n,t)|}{\sum_{n=0}^{N-1}\langle S(n,t)\rangle} \\ &= 20\log_{10}\frac{\sum_{n=0}^{N-1}\left||S(n,t)|^2-\langle S(n,t)\rangle\right|}{\sum_{n=0}^{N-1}\langle S(n,t)\rangle}\end{aligned} \tag{2}$$

where $D(n,t)=|S(n,t)|^2-\langle S(n,t)\rangle$ is the difference spectrum between the background and short-time spectra. The background noise spectrum $\langle S(n,t)\rangle$ is not updated during a sound event. The threshold was set to 35 dB in [6].

The second detector is used for detecting narrow-band sound events. This detector compares the maximum difference spectrum peak to its average [6]

$$\mathrm{T2}_{dB}=20\log_{10}\left[\max(D(n,t))-\sqrt{\frac{1}{N}\sum_{n=0}^{N-1}D(n,t)^2}\right] \tag{3}$$

A threshold of 35 dB was also used for this detector in [6]. In this work, a threshold of 20 dB was found to be suitable for both detectors.

Detecting and storing of sound events was implemented using Mustajuuri [12], a real-time audio processing system. In order to avoid excessive use of mass storage, a sampling rate of 16 kHz was used. Some heuristics are needed to get well-segmented sound events. The two detection functions of Eqs. (2) and (3) typically fluctuate below and above the thresholds, so a mechanism is needed that avoids gaps in the sound events. This accomplished by saving a few extra frames after the value of the detection function has dropped below the respective threshold. It is also necessary to always append at least one extra frame preceding the event, because otherwise the starts of transients are likely to be missed [6].

3. THE FEATURES

[1] The Eqs. (1), (2) and (3) are based on [6], but modified by the author.

[2] Also known as the "leaky integrator".

The next part of the audio surveillance system is the feature extraction stage, in which descriptive features will be calculated from each sound event. There exists a plenty of features that have been used in audio classification problems (see, e.g., [13], [14]). An important part of designing a system for (audio) classification is to decide, which features will be used. In the case of supervised classification, the performance of the classification system can be easily evaluated, since the true class labels for each sample are known. In unsupervised classification, the situation becomes more complicated, since true class labels are not necessarily available. In that case, some numerical measures for clustering performance are needed. If class labels exist, one can construct error measures based on how the different classes are spread into the clusters.

The features considered in this study are listed in Table 1. Most of the features are frequency-domain features, describing different aspects of the frequency distribution of the signal. The pitch describes the fundamental frequency f_0 of the signal and is calculated using the YIN algorithm [15]. Coherence, a byproduct of pitch calculation, describes how strong the fundamental frequency is. The last three features (described in more detail in [16]) are calculated in the time domain and describe the characteristics of the signal waveform. More detailed descriptions of the features listed in Table 1 can be found from the references listed in the third column.

All of the features are calculated from the segmented sound events in overlapping frames. Since the segmentation by the online detection algorithm is rather coarse, a finer segmentation based on short-time energy is performed. If multiple events are found in the fine segmentation, the one with highest energy is chosen, since that event has most likely caused the detection in the first place. The purpose of the fine segmentation is to avoid the background sound and unimportant low-energy sound events from affecting the calculated features.

In the case of the MFCCs, the mean of each coefficient calculated from individual frames of the fine-segmented event is used in the final feature vector, as well as the mean of the first order differences between frames, termed the ΔMFCCs. With all other frequency-domain features and the ZCR, the mean and variance of the feature sequence calculated from frames is used. For the pitch and coherence, the median is used instead of the mean in order to avoid sudden erroneous pitch jumps from affecting the feature. The last three time domain features (LoHAS, LoLAS and AHA) are calculated exactly as described in the appendix of [16]. Each feature is normalized by subtracting the mean and dividing by standard deviation.

4. CLUSTERING

The final step in this audio surveillance system is clustering. Two different clustering methods were compared: self-tuning spectral clustering and classical K-means clustering. Spectral clustering algorithms are methods which attempt to cluster similar data points together using eigenvectors of matrices derived from the data [19]. Typically, the affinity matrix between data points is used. An algorithm described by Ng et al. [19], Ng-Jordan-Weiss (NJW) algorithm, is a basic version of a spectral clustering algorithm. The self-tuning spectral clustering algorithm [8], also known as the Zelnik-Perona (ZP) algorithm, is an extension to the NJW algorithm, which can handle multi-scale data and choose the scaling parameter and the optimal number of clusters automatically. The classic K-means clustering algorithm [7] recovers maximally compact clusters by minimizing a cost function of sums of distances of data vectors from closest cluster centers.

5. EVALUATION

The system was evaluated using a set of 226 sound events recorded during a weekend in April 2006 (from Friday evening to Monday morning) in the office room of the author. There are no speech events in the data set. We assume here that speech events are treated separately in audio surveillance, by detecting speech activity and analyzing speech content by appropriate methods.

Table 2 lists the 24 different feature classes and the corresponding number of events, as derived in manual clustering (labeling). Close attention has been paid to whether or not the sound source was in the same room as the microphone. Outside events are indicated by "(far)" in Table 2. It is seen that most of the events are transient like sounds (doors, "thumps" and "clonks") and sounds of cars passing by the nearby road. Most of the transient sounds are most likely caused by guards on shift and people leaving/entering the work place. The total number of events is very low, which is quite consistent with the fact that the office is empty during weekends. There are also

Name	Abbreviation	Reference(s)
Mel-frequency cepstral coefficients	MFCC	[17] [13]
Delta mel-frequency cepstral coefficients	ΔMFCC	[13]
Spectral centroid	SC	[13]
Bandwidth	BW	[13]
Spectral roll-of frequency	SRF	[13]
Spectral flatness	SF	[18]
Delta spectral magnitude	DSM	[13]
Root-mean square value	RMS	
Zero-crossing rate	ZCR	[13]
Pitch	-	[13]
Coherence	-	
Length of high amplitude sequence	LoHAS	[16]
Length of low amplitude sequence	LoLAS	[16]
Area of high amplitude	AHA	[16]

Table 1: The features used in this study.

many clear outlier event classes that occur less than 10 times.

5.1. Analysis of factor loadings

To gain an insight into the mutual dependencies between features, the PCA factor loading matrix with Varimax rotation [9] is calculated using the entire feature set of 48 features. The elements of the loading matrix indicate correlation between features and the principal components (PCs) [4] [5]. The principal components (columns of the matrix) are ordered in the order of explained variances, so that the first components explain most of the variance in the data. If a feature has a high loading value on an important PC, the feature explains larger amount of data compared to a feature loading a less important PC. Also, if two features load the same component, the features are mutually correlated.

The absolute values of the loadings are shown in Fig. 1, with loadings smaller than 0.4 set to zero (white) for clarity. The mutual correlation between MFCC 1, SC, BW, SRF, SF and ZCR is evident, since all features load the first PC. The first five of the aforementioned features are correlated because they all describe the coarse shape of the spectrum. Even though a time-domain feature, the zero-crossing rate is also correlated with the spectral centroid and thus loads the first PC as well.

The 2nd PC is loaded by the RMS and AHA features. Both describing the energy content of the signal, they clearly carry significantly orthogonal information compared to the first PC, which was related to spectral shape. The amplitude descriptors (LoHAS and LoLAS) seem to load the 3rd principal component, again providing information not found in the other features.

The variances of the spectral features SC and BW, and SRF and SF, load the 4th and 5th PCs, respectively. The reason for the correlation of the variances of SC and BW is likely the fact that the BW feature tells how much the spectrum is spread around its centroid, which is the SC feature. The correlation between the variances of SRF and SF is harder to explain intuitively. Nevertheless, these features seems to carry information not found in the other considered features.

The coherence seems to account for more variance in this data set, compared to the pitch. This is most likely due to the fact that most sounds in the data set are transient-like impact sound (see Table 2), and do not have a clear fundamental frequency.

It should be kept in mind that one can not directly predict the performance of different features in clustering from the PCA loading matrix, since the loadings only tell about which features are mutually correlated and which features account for most of the variance in the data. Accounting for a large amount of variance does not necessarily imply that the feature performs well in classification.

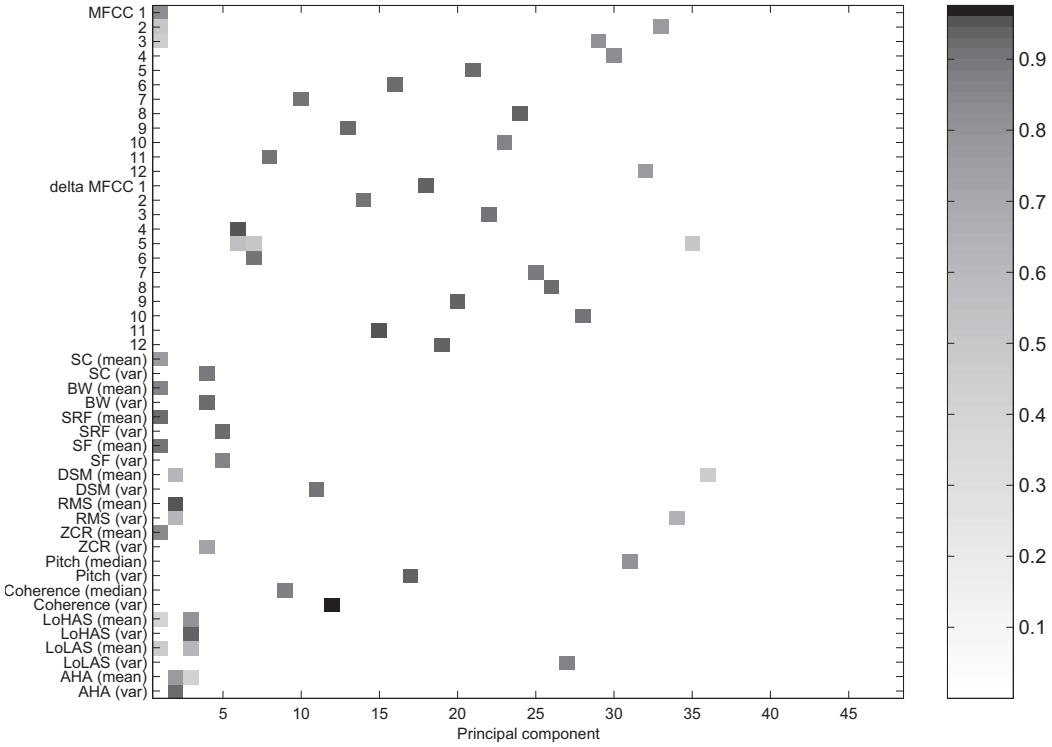

Fig. 1: Factor loading matrix for all considered features.

5.2. Evaluated feature sets

Since the choice of features is crucial for the performance of a pattern recognition system, one would like to have the optimal set of features for the problem in hand. The performance of different feature sets derived from the features listed in Table 1 is therefore investigated. Four cases are considered:

- **Entire feature set**
  All 48 features (using the statistics calculated from features listed in Table 1, as described in Sec. 3).
- **Static MFCC features only**
  The 12 MFCC features.
- **PCA-transform of entire feature set**
  A 22-dimensional feature set (explaining 90% of the variance) derived from the principal component analysis (PCA).

5.3. Quantitative evaluation

The different combinations of the four feature sets and two clustering methods were evaluated quantitatively. The number of clusters C was chosen to be equal to the number of classes M in all cases, i.e., $C = M = 24$. Thus in ideal case, each cluster would contain members of one class only.

The performance of the clustering in terms of the manually derived labels (Table 2) is evaluated by calculating

Index	Class description	Number of events in class
1	door (far)	35
2	thump (far)	25
3	car (far)	22
4	clonk (far)	21
5	damped thump (far)	21
6	unidentifiable (far)	20
7	door	14
8	clonk	10
9	double thump (far)	10
10	general LF (far)	10
11	keyboard	7
12	sweep	7
13	screech (far)	5
14	general click	3
15	thump and keyboard	3
16	chair	2
17	thump	2
18	thump HF	2
19	zipper	2
20	click	1
21	coughing	1
22	general click (far)	1
23	impulsive	1
24	thump HF (far)	1

Table 2: The manual class labels used in this study. The total number of events was 226. 'LF' stands for low-frequency, 'HF' for high-frequency, and '(far)' indicates that sounds from that class do not originate from the room of recording.

two statistics. The first of these is the clustering error, which is defined as the average fraction of event assigned to other but dominant class in a cluster

$$Error = \frac{1}{C}\sum_{i=1}^{C}\frac{N_{i,tot} - N_{i,max}}{N_{i,tot}} \quad (4)$$

where C is the number of clusters, $N_{i,tot}$ is the total number of events in cluster i, and $N_{i,max}$ is the largest number of events in cluster i with mutually identical class labels. In the ideal case, when each cluster contains members of one class only, the error will be zero.

The second statistic measures how much events labeled to a single class are spread across clusters. This is accomplished by

Clustering method	Feature set	Error	Spread
Self-tuning spectral clustering	All features	.52	.19
	MFCC	.56	.23
	PCA	.55	.20
K-means clustering	All features	.38	.16
	MFCC	.45	.16
	PCA	.40	.16

Table 3: The performance of different clustering methods and different feature sets, when the number of clusters is 24.

$$Spread = \frac{1}{M}\sum_{i=1}^{M}\frac{C_i}{C} \quad (5)$$

where M is the number of manually derived classes, C is the number of clusters and C_i is the number of clusters with at least one event having label i.

5.4. Results

Table 3 shows the performance of the two clustering algorithms (self-tuning spectral clustering and K-means) with the three different feature sets (see Sec. 5.2). It is clear that in this application, K-means clustering performs better than self-tuning spectral clustering. It seems that one does not need a complicated clustering algorithm for this kind of data. It is evident that using the original feature set or its PCA-reduced version gives equally low *Error* and *Spread* values with both clustering algorithms. Discarding features other than the static MFCCs slightly degrades the clustering performance with both algorithms, as information on the spectral envelope only is utilized.

6. OPTIMIZING FEATURE WEIGHTS

Since the feature sets presented in Sec. 5.2 are were chosen in a heuristic manner, it would be interesting to know, which features are really useful for the clustering. Therefore, an optimal feature set was derived by finding feature weights (in the range of $[0,1]$) that maximize the clustering performance in terms of the manual class labels. Features with low weights could be discarded without affecting the clustering performance.

The feature weights were optimized using a genetic algorithm (GA), due to their effectiveness in high-dimensional search spaces [20]. An approach similar

to that of [21] was used, where the simple genetic algorithm (SGA) is used for the optimization. All of the calculations are performed using the Genetic Algorithm Toolbox ver. 1.2 (see Sec. 8 for web address), with the default settings of the sample SGA script (`sga.m`), except that the population size was set to 10 times the dimension of the feature vector, i.e., 480. The algorithm was run for 150 iterations, at which point the optimization had reasonably converged and the weights did not change significantly anymore.

The first considered objective function that was minimized is the sum of the clustering error and the spread, as defined in Eqs. (4) and (5), plus a penalty term

$$Obj_1 = Error + Spread + Penalty \tag{6}$$

where the penalty term is otherwise equal to Eq. (4), but classes which have a clustering error of zero, i.e., only one class, are not included. The penalty term avoids the creation of large clusters with too many classes.

An alternative objective function is the sum of the number of classes in each cluster

$$Obj_2 = \sum_{j=1}^{C} \sum_{i=1}^{M} I_{i,j} \tag{7}$$

where C is the number of clusters, M is the number of classes and $I_{i,j}$ is 1 when there is at least one event of class i in cluster j.

6.1. Results

Table 4 shows the feature weights, derived by the GA using one of the two objective functions (Obj_1 and Obj_2). Weights over 0.5 are shown in bold. One can see that the MFCCs seem to be important for classification in both cases. However, all of the MFCCs are not weighted as much as others and the weights differ between the objective functions. Most notably, in the case of Obj_2, almost all of the ΔMFCCs are given weights over 0.5. This is most likely due to the fact that with Obj_2, the optimization tries to more aggressively force the events with the same class label into an own cluster, which promotes the use of temporal information as well. The importance of temporal features has been observed before in the context of audio and music classification [14].

The variances of RMS and Pitch seem to be important with both objective functions. These features contain information that is not found in the spectral features and it might be that they are important because of that. It is interesting to note that the variances of these features are more informative than the means. This phenomenon, that second order statistics convey more information than the features themselves, has been observed before in the context of speech and music discrimination [22]. Similar phenomenon can be found among the amplitude descriptors (LoHAS, LoLAS and AHA), where the variance of AHA seems to be the most important.

Interestingly, the means of BW and SRF seem to be more important with the second, more aggressive objective function. It is hard to deduce an intuitive explanation for this. It might also be that those features, being correlated with the first MFCC, can have either low or high weights, without affecting the clustering performance. The same might be true with the mean of ZCR, which is given much lower weight with Obj_2.

Figs. 2 and 3 how the different classes are distributed across clusters, when using an optimized (Obj_2) and non-weighted version of the 48-dimensional set of all considered features. In the original matrix, each element (i, j) contains the number of sound events with manual class label i present in cluster j. For plotting Figs. 2 and 3, the *rows* of the matrix are normalized to have sum equal to one, so the distribution of a certain event class across clusters can be seen. The rows and columns of the matrix are ordered by the sizes of the classes and the clusters, respectively.

Figs. 2 and 3 show that with the optimized feature set, the events are not spread into as many classes as is the case with the non-weighted feature set. However, this comes with the expense of the classes also being much larger. It might be, though, that with the current set of features, regardless of their weighting, it is not possible to differentiate between the largest transient classes, such as the “thump” and “clonk” classes from outside the room. These two classes could also possibly have been merged, due to their similarity. The car sounds have been quite nicely separated into two classes by the weighted feature set. Some sounds that occur only once, such as the mouse click, have been put into a large cluster with other sounds when the weighted feature set is used.

Table 5 shows the clustering results with the optimized weights. By comparing the results with Table 3 it can be observed that both of the weighted feature sets give lower values of *Error* and *Spread*, compared to the feature sets of Sec. 5.2. Again, the K-means clustering is superior to

the self-tuning spectral clustering algorithm. The better performance of features optimized with Obj_2 compared to those optimized with Obj_1 is also evident when using K-means clustering.

Table 6 shows the results when features having weights less than 0.5 are completely discarded. This results in feature vectors of dimensions 19 and 25, for the objective functions Obj_1 and Obj_2, respectively. When K-means clustering is used, the performance is slightly worse than with the weighted feature sets (see Table 5). In the case of self-tuning spectral clustering, the performance is slightly better.

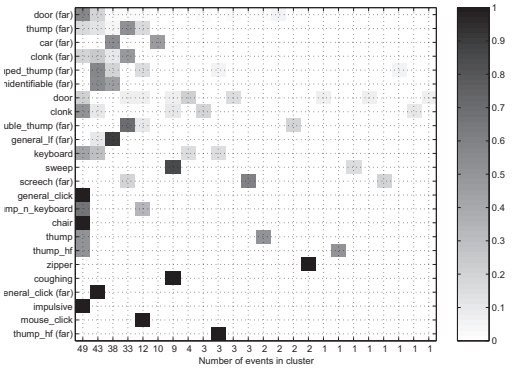

Fig. 2: Distribution of manually labeled classes across clusters (K-means a feature set of dimension 48 optimized by GA with Obj_2, number of clusters $C = 24$).

7. CONCLUSIONS

This work had a goal of assessing the effect of chosen feature set on the performance of unsupervised clustering in an audio surveillance application. Basically, the goal was to evaluate the importance of features, when the goal is classify the sound events in a similar way as humans do. A high-dimensional feature set was compared with a PCA-reduced version, as well as with the full feature set with feature weight optimization.

It was found that in order to force the sound events into different clusters by perceptually derived class label, features that contain temporal information are important and have to be given more weight than for many of the static

Feature	Weight (Obj_1)	Weight (Obj_2)
MFCC 1	**0.96**	**0.88**
2	**0.99**	0.20
3	**0.65**	0.11
4	**0.87**	**0.72**
5	0.45	**0.53**
6	0.17	0.41
7	0.42	0.12
8	0.06	0.26
9	**0.67**	0.11
10	**0.54**	**0.66**
11	**0.53**	**0.74**
12	0.07	0.28
ΔMFCC 1	**0.95**	0.49
2	**0.83**	**0.90**
3	0.14	**0.97**
4	0.38	**0.74**
5	0.28	**0.51**
6	**0.91**	**0.84**
7	0.21	**0.51**
8	0.06	**0.94**
9	**0.91**	**0.96**
10	**0.99**	**0.91**
11	**0.82**	**0.81**
12	0.01	**0.98**
SC (mean)	0.23	0.43
SC (var)	0.01	0.27
BW (mean)	0.34	**0.99**
BW (var)	0.09	0.18
SRF (mean)	0.06	**0.91**
SRF (var)	0.17	0.48
SF (mean)	**0.63**	**0.56**
SF (var)	0.33	0.12
DSM (mean)	0.02	0.02
DSM (var)	**0.61**	**0.79**
RMS (mean)	0.24	0.25
RMS (var)	**0.73**	**0.87**
ZCR (mean)	**0.73**	0.24
ZCR (var)	0.31	**0.94**
Pitch (median)	0.47	0.30
Pitch (var)	**0.96**	**0.87**
Coh. (median)	0.01	**0.77**
Coh. (var)	0.24	0.01
LoHAS (mean)	0.10	0.15
LoHAS (var)	0.20	0.02
LoLAS (mean)	0.17	0.08
LoLAS (var)	0.03	0.11
AHA (mean)	0.33	0.11
AHA (var)	**0.77**	**0.83**

Table 4: The optimal weights for each feature.

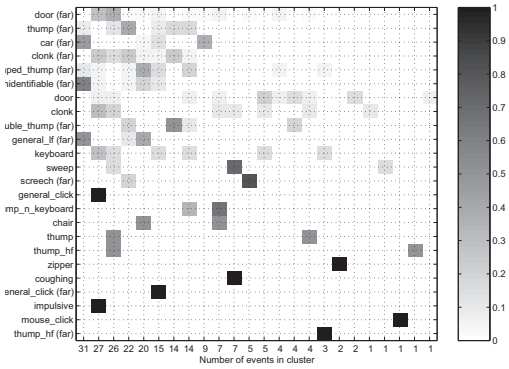

Fig. 3: Distribution of manually labeled classes across clusters (K-means with non-weighted feature set, number of clusters $C = 24$).

Clustering method	Objective function	Error	Spread
Self-tuning spectral clustering	Obj_1	.50	.20
	Obj_2	.48	.21
K-means clustering	Obj_1	.31	.14
	Obj_2	.24	.12

Table 5: The performance of different clustering methods for the feature set with optimized feature weighting.

Clustering method	Objective function	Error	Spread
Self-tuning spectral clustering	Obj_1	.47	.20
	Obj_2	.45	.20
K-means clustering	Obj_1	.33	.15
	Obj_2	.32	.15

Table 6: The performance of different clustering methods for the feature set with features having weights < 0.5 discarded.

and spectral features. However, the coarse shape of the spectral envelope also plays an important role.

It was also found that the K-means clustering algorithm performs better than the ZP algorithm in all cases. It might be that the ZP algorithm, which is originally intended for computer vision applications, is not suitable for audio — at least in this particular study. Most likely the differences in topologies of the feature (data) spaces cause different clustering algorithms to be effective in different applications.

Future work on this subject includes finding better criteria for assessing the clustering performance. New features could also be developed which in particular could be used to differentiate between different transients. It is evident that this can not be accomplished by using static or averaged spectral features alone. A larger data set should also be used in the analysis to get clearer clusters.

8. ACKNOWLEDGMENTS

The author wishes to thank Dr. Aki Härmä and Dr. Tapio Lokki for helpful discussions and comments. A set of Matlab functions for factor loading analysis [3] by Jürgen Kayser was used. The (self-tuning) spectral clustering experiments were conducted using Matlab code[4] by Lihi Zelnik-Manor. Optimization of the feature weights was conducted using the Genetic Algorithm Toolbox version 1.2[5]. This work was funded by the HeCSE graduate school.

9. REFERENCES

[1] R. Radhakrishnan, A. Divakaran, and P. Smaragdis. Audio Analysis for Surveillance Applications. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA'05)*, New Paltz, NY, October 2005.

[2] C. Stauffer. Automated Audio-Visual Analysis. Technical Report AIM-2005-026, MIT Artificial Intelligence Laboratory, 2005.

[3] M. Vacher, D. Istrate, J.-F. Serignat, and N. Gac. Detection and Speech/Sound Segmentation in a Smart Room Environment. In *Proceedings of the 3rd International Conference on Speech Technology and Human-Computer Dialogue*, Cluj, Romania, May 2005.

[3] http://psychophysiology.cpmc.columbia.edu/mmedia/Kayser2003a/Appendix.html

[4] http://www.vision.caltech.edu/lihi/Demos/SelfTuningClustering.html

[5] http://www.shef.ac.uk/acse/research/ecrg/gat.html

[4] D. Mitrovic, M. Zeppelzauer, and H. Eidenberger. Analysis of the Data Quality of Audio Features of Environmental Sounds. *Journal of Universal Knowledge Management*, 1(1):4–17, 2006.

[5] D. Mitrovic and M. Zeppelzauer and H. Eidenberger. Towards an Optimal Feature Set for Environmental Sound Recognition. Technical Report TR-188-2-2006-03, Vienna University of Technology, 2006.

[6] A. Härmä, J. Skowronek, and M. F. McKinney. Automatic Surveillance of the Acoustic Activity in Our Living Environment. In *Proceedings of the IEEE International Conference on Multimedia & Expo (ICME 2005)*, Amsterdam, The Netherlands, July 2005.

[7] S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Academic Press, 1999.

[8] L. Zelnik-Manor and P. Perona. Self-Tuning Spectral Clustering. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 1601–1608. MIT Press, Cambridge, MA, 2005.

[9] H. Harman. *Modern Factor Analysis*. University of Chicago Press, 2nd edition, 1967.

[10] C. Stauffer and W. E. L. Grimson. Adaptive Background Mixture Models for Real-Time Tracking. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR99)*, Fort Collins, CO, USA, June 1999.

[11] M. Cristani, M. Biecgo, and V. Murino. On-Line Adaptive Background Modelling for Audio Surveillance. In *Proceedings of the 17th International Conference on Pattern Recognition (ICPR'04)*, Cambridge, The United Kingdom, August 2004.

[12] T. Ilmonen. Mustajuuri - An Application and Toolkit for Interactive Audio processing. In *Proc. ICAD 2001*, pages 284–285, Espoo, Finland, 2001.

[13] D. Li, I. K. Sethi, N. Dimitrova, and T. McGee. Classification of General Audio Data for Content-Based Retrieval. *Pattern Recognition Letters*, 22(5):533–544, 2001.

[14] M. F. McKinney and J. Brebaart. Features for Audio and Music Classification. In *Proceedings of the International Symposium on Music Information Retrieval (ISMIR'2003)*, Baltimore, USA, October 2003.

[15] A. de Cheveigne and H. Kawahara. YIN, a Fundamental Frequency Estimator for Speech and Music. *The Journal of the Acoustical Society of America*, 111(4):1917–1930, 2002.

[16] D. Mitrovic. Discrimination and Retrieval of Environmental Sounds. Technical Report TR-188-2-2005-16, Vienna University of Technology, 2005.

[17] S. Davis and P. Mermelstein. Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4):357–366, 1980.

[18] O. Izmirli. Using a Spectral Flatness Based Feature for Audio Segmentation and Retrieval. In *Proc. ISMIR 2000*, pages 284–285, Plymouth, MA, USA, 2000.

[19] A. Y. Ng, M. I. Jordan, and Y. Weiss. On Spectral Clustering: Analysis and an Algorithm. In *Advances in Neural Information Processing Systems*, pages 849–856. MIT Press, Cambridge, MA, USA, 2002.

[20] J. Yang and V. Honavar. Feature Subset Selection Using a Genetic Algorithm. *IEEE Intelligent Systems*, 13:44–49, 1998.

[21] B. Minaei-Bidgoli and W. F. Punch. Using Genetic Algorithms for Data Mining Optimization in an Educational Web-based System. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2003)*, Chicago, IL, USA, July 2003.

[22] E. Scheirer and M. Slaney. Construction and Evaluation of a Robust Multifeature Speech/Music Discriminator. In *Proc. ICASSP '97*, pages 1331–1334, Munich, Germany, 1997.