University of Marburg at TRECVID 2008: High-Level Feature Extraction

Markus Mühling[1,2], Ralph Ewerth[1,2], Thilo Stadelmann[1,2], Bing Shi[2],
and Bernd Freisleben [1,2]

[1] SFB/FK615, University of Siegen, D-57068 Siegen, Germany
[2] Dept. of Math. and Computer Science, University of Marburg, D-35032 Marburg, Germany
{muehling, ewerth, stadelmann, shib, freisleb}@informatik.uni-marburg.de

Abstract

In this paper, we summarize our results for the high-level feature extraction task at TRECVID 2008. Our last year's high-level feature extraction system was based on low-level features as well as on state-of-the-art approaches for camera motion estimation, text detection, face detection and audio segmentation. This system served as a basis for our experiments this year and was extended in several ways. First, we paid attention to the fact that most of the concepts suffered from a small number of positive training samples while offering a huge number of negative ones. We tried to reduce this unbalance of positive and negative training samples by sub-sampling the negative instances. Furthermore, we increased the number of positive training samples by creating image variations. Both methods improved the detection results significantly, while the sub-sampling approach achieved our best result (8.27% mean inferred average precision). Second, we incorporated two further feature types: Hough features and audio low-level features. Finally, we supplemented our approach using cross-validation in order to improve the high level feature extraction results. On the one hand, we applied cross-validation for feature selection, on the other hand we tried to find the best sampling rate of negative instances for each concept.

1. Structured Abstract

In this section, the results of our participation in the high-level feature extraction task are presented in the form of the requested structured abstract. In Section 2, we describe the extracted low-level features plus additional mid-level features, which are the result of state-of-the-art algorithms in the field of camera motion estimation [5], text detection [6], face detection [17], and audio segmentation. The components of our system are discussed in detail in Section 3. The experimental results are presented in Section 4. Section 5 concludes the paper.

The high-level feature extraction experiments were evaluated by the TRECVID team [16] using the inferred average precision measure suggested by Aslam et al. [2].

"What approach or combination of approaches did you test in each of your submitted runs?"

The following six runs of category "A" were submitted:

- A_Marburg1: Baseline, TRECVID 2008 training set with merged annotations from active learning and MCQ-ECT-CAS;
- A_Marburg2: Baseline plus using only every fourth negative sample for training;
- A_Marburg3: Baseline plus Hough and audio low-level features;
- A_Marburg4: A_Marburg3 plus cross-validation for different sampling rates of negative instances;
- A_Marburg5: A_Marburg3 plus image variations of positive samples;
- A_Marburg6: A_Marburg3 plus cross-validation for feature subsets.

"What, if any significant differences (in terms of what measures) did you find among the runs?"
"Based on the results, can you estimate the relative contribution of each component of your system/approach to its effectiveness?"

In a first experiment, we reduced the number of negative training instances by a simple sub-sampling method. This sub-sampling approach considered only every fourth negative sample and improved the results of our last year's baseline system significantly (from 5.91% to 8.27% mean inferred average precision). It achieved our best run for high level feature extraction in terms of mean inferred average precision.

Furthermore, we supplemented our low-level feature set with Hough and audio low-level features. This run (A_Marburg3) using the extended feature set showed a slight performance decrease (from 5.91% to 5.76% mean inferred average precision). Based on the previous system, we performed three further experiments. First, we applied sub-sampling of negative instances in combination with stratified threefold cross-validation in order to find the best sampling rate. Again we achieved clearly better results compared to the reference system (5.76% vs. 8.04% mean inferred average precision). Second, we increased the number of positive training samples by creating image variations of positive key frames and thus improved the results to 7.39% mean inferred average precision. Third, we applied stratified threefold cross-validation to find the best feature subset for each concept. Interestingly, the use of this cross-validation could not improve the detection results.

“Overall, what did you learn about runs/approaches and the research question(s) that motivated them?”

The experiments revealed that the approaches trying to reduce the unbalance between positive and negative training samples improved the high-level feature extraction results significantly. The sub-sampling of negative instances not only accelerated the process of building the concept model but most notably leads to clearly better results in terms of mean inferred average precision. Only the concept “Classroom” could not profit from this add-on in both related experiments. Likewise, the increase of positive training samples by creating image variations had a positive impact on the overall detection results. Particularly, the result of the concept “Mountain” was strongly boosted by this approach and achieved our best result for this concept. Furthermore, the experiments showed that the use of cross-validation for models built on different feature subsets brought no performance gain.

2. Feature Extraction

Our video analysis system automatically extracts several low-level as well as mid-level features. Compared to our last year’s system we additionally extracted audio low-level features and Hough features. In section 2.1 we describe our visual features, followed by the audio features in section 2.2.

2.1 Visual Features

Several visual features are extracted for each video shot. The frame in the middle of a shot was used as a key frame. If a key frame contains black bars, these top and bottom regions of the image are automatically detected and removed in a preprocessing step. The removal of black bars is realized by zooming into the image. The following low-level features are extracted from a key frame: color moments, color correlograms, Hough features, texture features and Gabor wavelet features. In addition, several mid-level features are extracted automatically from the entire shot by utilizing camera motion estimation [5], face detection [17] and text detection [6]. In the following, the extracted features are briefly described.

Color moments: Color moments are extracted at two different granularities. The first three global color moments are computed for the whole key frame. Corresponding values are extracted for each region of a 3 x 3 grid in HSV (Hue, Saturation, Value) color space. The i-th pixel of the j-th color channel of an image region is represented by c_{ij}. Then, the first three color moments are defined as:

$$mean_j = \frac{1}{N} \cdot \sum_{i=0}^{N-1} c_{ij} \qquad (1)$$

$$stdev_j = \sqrt{\frac{1}{N} \cdot \sum_{i=0}^{N-1} (c_{ij} - mean_j)^2} \qquad (2)$$

$$skew_j = \sqrt[3]{\frac{1}{N} \cdot \sum_{i=0}^{N-1} (c_{ij} - mean_j)^3} \qquad (3)$$

Texture features: The gray-scale image co-occurrence matrices m_k are constructed at 8 orientations. We use these matrices to extract the following values representing the global texture:

$$energy_k = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (m_{kij})^2 \qquad (4)$$

$$contrast_k = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (i - j)^2 \cdot m_{kij} \qquad (5)$$

$$entropy_k = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} m_{kij} \cdot log(m_{kij}) \qquad (6)$$

$$homogeneity_k = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\frac{m_{kij}}{1+|i-j|}, \qquad (7)$$

where N is the number of gray values and m_{kij} is the value of the co-occurrence matrix m_k at position (i, j).

Color autocorrelograms: Color correlograms describe the spatial relationship between colors, whereas autocorrelograms are limited to identical colors. An autocorrelogram expresses the probabilities of colors to re-occurr in a certain distance. We preferred small distances (1, 4, 7, 10, 13, 16 and 19 pixel), so that local spatial correlations of identical colors are represented by the correlogram. Colors are described in HSV color space. By choosing a smaller number of bins representing the brightness component we get more independent of illumination changes. In total, each color correlogram results in a 350-dimensional feature vector.

Hough features: The Hough transform is a feature extraction method to detect parametrizable geometrical objects in binary gradient images. We used the probabilistic Hough transform provided by the OpenCV library [15] to detect lines in edge images. The binary gradient images are the result of the robust canny edge detection algorithm. The results of the Hough transform are exploited to build two-dimensional histograms based on the orientation and length of the detected lines. Altogether we obtain ten histograms, one global histogram and one for each region of a 3 x 3 grid. Using 6 bins for length respectively orientation, we totally obtain a 360-dimensional feature vector.

Gabor wavelet features: Gabor wavelet features are extracted for eight orientations and five frequencies. The functions to compute the wavelet coefficients can be expressed as follows [8]:

$$\begin{aligned} g_{\theta,\lambda,\varphi,\sigma,\gamma}(x,y) &= e^{-\frac{x'^2+\gamma^2 y'^2}{2\sigma^2}}\cos(2\pi\frac{x'}{\lambda}+\varphi) \\ x' &= x\cos\theta + y\cos\theta \\ y' &= -x\sin\theta + y\cos\theta \end{aligned} \qquad (8)$$

A Gabor wavelet is controlled by five parameters: orientation θ, wave length λ, phase φ, radius σ of the Gaussian function, and the aspect ratio γ. The radius of the Gaussian function is chosen proportionally to the wave length, and the aspect ratio is fixed to 1. Gabor energies of a pixel for the different orientation and spatial-frequency combinations are obtained by a superposition of the phases 0 and π/2 using the L2-Norm. The resulting 40 Gabor energies per pixel are summarized in a Gabor histogram describing the whole image. By distinguishing ten energy classes, we obtain 400 histogram feature values. We further computed the average result of each Gabor energy filter for each region of a 4 x 4 grid. Thus, the total number of Gabor wavelet features amounts to 1040 values.

Camera motion features: Motion vectors embedded in MPEG videos are employed to estimate camera motion at the granularity of P-frames, according to the approach presented in [5]. The following camera motion types are distinguished: translation along the x-axis, respectively y-axis, rotation around the x-axis, respectively y-axis and z-axis, and zoom. The distribution of the values for a shot concerning the different camera motion types are described by using the following statistical values: mean, median, minimum, maximum, standard deviation, and skewness. In addition, the percentages of a shot concerning the different camera motion types pan, tilt and zoom are considered, so that we finally get a 39-dimensional camera motion vector.

Text features: A robust text detection approach [6], which can automatically detect horizontally aligned text with different sizes, fonts, colors and languages, is applied at the granularity of I-frames. First, a wavelet transformation is applied to the image and the distribution of high-frequency wavelet coefficients is considered to statistically characterize text and non-text areas. Then, the k-means algorithm is used to classify text areas in the image. The detected text areas undergo a projection analysis in order to refine their localization. We use the detected text areas to derive the following features per shot: the number of appearing text elements, the average text position, the average text frame coverage, and the average number of text elements per frame.

Face features: Frontal and profile faces are detected in each video frame using the face detector provided by the OpenCV library [15]. The face detection approach is an implementation of the approach suggested by Viola and Jones [17] with Lienhart's extensions [10]. The Adaboost-based approach of Viola and Jones was chosen since it is a very fast approach that nearly operates in real-time on today's computers and thus can even be applied to every single frame of a sequence. Since their approach usually reports many detections of slightly different sizes and positions, an average rectangle is computed based on the reported detections, in case that the number of detections exceeds a threshold. A tracking procedure also based on the OpenCV library is used to assemble face appearances of the same person in subsequent frames of a shot using the optical flow computation of Bouquet [3], which is an extension of the Lukas-Kanade [13] algorithm. The extension processes image pyramids to enable the estimation of fast movements as well. For each shot, the number of face sequences, the number of detected faces, front faces respectively profile faces, the average frontal or profile shot size, respectively, the mean number of detection hits for frontal faces or profile faces, respectively, and the percentage length of a shot, where a person appears, are considered as mid-level features.

2.2 Audio Features

This year we incorporated audio low-level as well as mid-level features in our concept detection system. To further analyze the audio data, the extracted low-level audio features were fed into a content-based audio classification and segmentation system based on the approach of Lu et al. [12].

Audio low-level features: We extracted two sets of low-level audio features: The first set serves as the basis for the mid-level features described below. It is specifically tailored to facilitate audio type classification and contains the following quantities, extracted from non-overlapping 25ms frames [12]: 8th-order mel-frequency cepstrum coefficients (MFCCs), zero-crossing rate, short time energy, sub-band energy distribution, brightness and bandwidth, spectrum flux, band periodicity , a measure of frame noisiness and the position of the cepstral peak. The second low-level feature set is composed to give a more general view of the audio content of a shot and to facilitate the recognition of e.g. single sounds directly in our concept detection system. It comprises 20 MFCCs with their first order derivatives, 10 line spectral pairs and a measure of pitch. These 51 features were each summarized per shot in a histogram comprising 10 bins, resulting in a 510-dimensional audio low-level feature vector.

Audio mid-level features: The audio type classification system produces mid-level features on a per-second (sub-clip) basis in the form of acoustic class labels and related probabilities for “silence”, “speech”, “pure speech”, “non-pure speech”, “music”, “background” and “action” sounds (an error label, “undefined”, may also be produced). The low-level features are therefore aggregated per second, normalized and then concatenated to form one feature vector per sub-clip, which is processed by a hierarchical tree of support vector machines, if it was not previously classified as silence by a threshold based classifier. Figure 1 shows this classification tree, which is trained on more than 32 hours of audio samples including, among others, the TIMIT data for clean speech [11] and the NOIZEUS [7] corpus. Five-fold cross-validation on a subset of 15000 feature vectors was used to find the best parameter settings for each two-class support vector machine with a RBF (radial basis function) kernel using the libSVM library [4]. Finalizing the classifier’s decision, short silence periods within speech are labeled as “pause” by a heuristic decision function. A second algorithm based on the work of Ahmadi and Spanias [1] processes energy, zero-crossing rate and cepstral peak low-level features to add “voiced”- and “unvoiced” speech labels to the mid-level features.

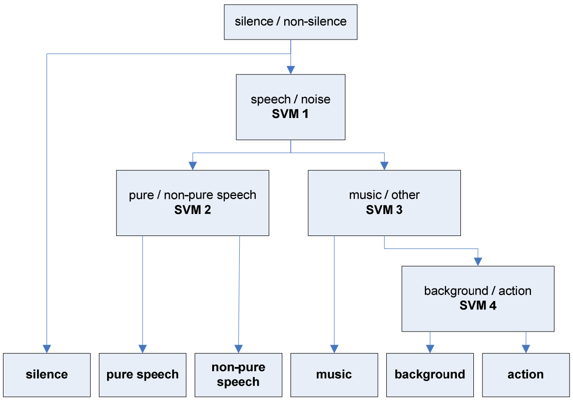

Figure 1: Scheme of the hierarchical audio type classifier: A single feature vector per sub-clip serves as input; output is a single acoustic class label and its corresponding probability.

All 11 mid-level features are then processed to describe the audio-content of a video shot by statistical values: mean, median, minimum, maximum, standard deviation, and skewness of the per-frame label-probabilities are calculated. Furthermore, the percentage of each audio type label with respect to the shot length is calculated. Finally, these percentages and the distribution properties of the probabilities are fed into the further learning algorithm as the final audio mid-level features, resulting in a 77-dimensional feature vector.

3. High-Level Feature Detection System

The goal of the proposed system is to learn models for the high-level semantic features based on the extracted audiovisual low-level and mid-level features described in section 2. In our baseline system (figure 2) we concatenated the multi-modal low-level and mid-level features in an early fusion scheme and fed them directly into a support vector machine with a radial basis function kernel using the implementation provided by the libSVM library [4].

To reduce the unbalance of positive and negative training samples, which concerns nearly all concepts, we applied the following two approaches. First, we reduced the number of negative instances by sub-sampling. Second, we increased the number of positive samples by creating image variations of positive key frames, where the number of these variations was dependent of the number of available positive training samples for a concept. Our goal was to obtain a number of positive training samples which was between 1200 and 1500. The number of applied variations varied depending on the number of available positive samples per concept. If the number of positive samples was above 1000, then no additional positive training samples were created. The following variations (in total up to 16 dependent on the concept) were applied:

- the brightness of the keyf rame was normalized;
- the key frame was smoothed with a Gaussian filter;
- it was zoomed into the image, i.e. an region of size ¾*width and ¾*height was cropped from the middle of the keyframe and scaled up to the original keyframe width and height;
- several rotated variations of the keyframe and the zoomed keyframe were created: with an rotation angle between -15° and +15° and a step size of 5°.

The features that were extracted from the whole shot and not only from the keyframe were left unchanged for the new instances. We obtained up to 16 additional training instances for each positive training sample.

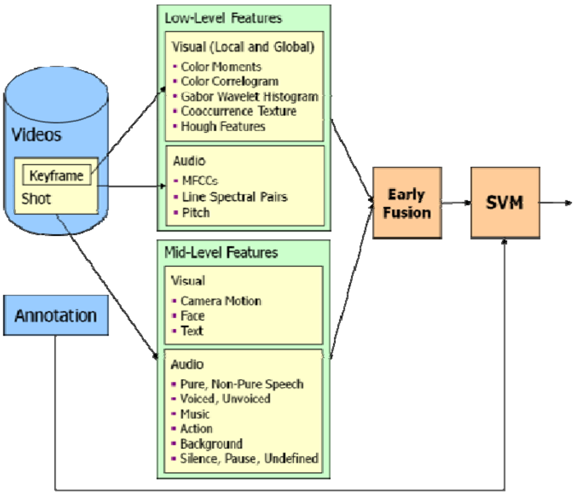

Figure 2: Overview of our baseline system.

Moreover, we used stratified three-fold cross-validation to find the best sampling rate for reducing the negative instances and to get the best subset of features for each concept. Inside the cross-validation process, we used average precision scores for evaluation.

4. Experimental Results

In this section, we present our results for the high-level feature extraction task. We submitted six runs of category "A". The MDC decoder was used for MPEG decoding [9] in our experiments. Our first run A_Marburg1 corresponds to the baseline system of the last year's challenge.

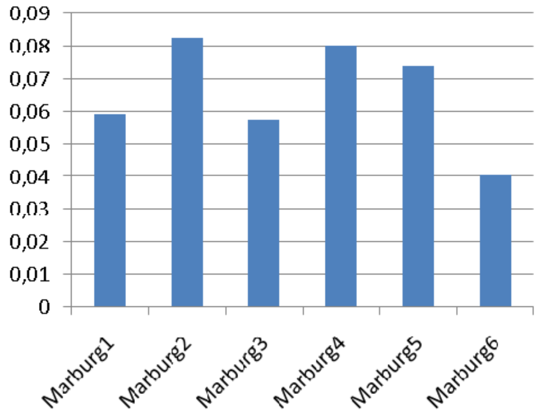

Figure 3: Overview of the results of our six runs in terms of mean inferred average precision.

In a first experiment (A_Marburg3) we extended this baseline system by a sub-sampling method, which considers only every fourth negative sample and thus reduces the number of negative training instances. This simple sub-sampling approach improved the results of our last year's baseline system significantly (from 5.91% to 8.27% mean inferred average precision) and achieved our best run for high-level feature extraction in terms of mean inferred average precision.

Furthermore, we enlarged our low-level feature set by adding Hough and audio low-level features, as described in Section 2. This run (A_Marburg3) using the extended feature set showed a slight performance decrease (5.91% vs. 5.76% mean inferred average precision). Based on this system, we performed three further experiments. We observed that most of the concepts suffer from a small number of positive training samples while offering a huge number of negative ones. Therefore, two of the following experiments reduce this unbalance of positive and

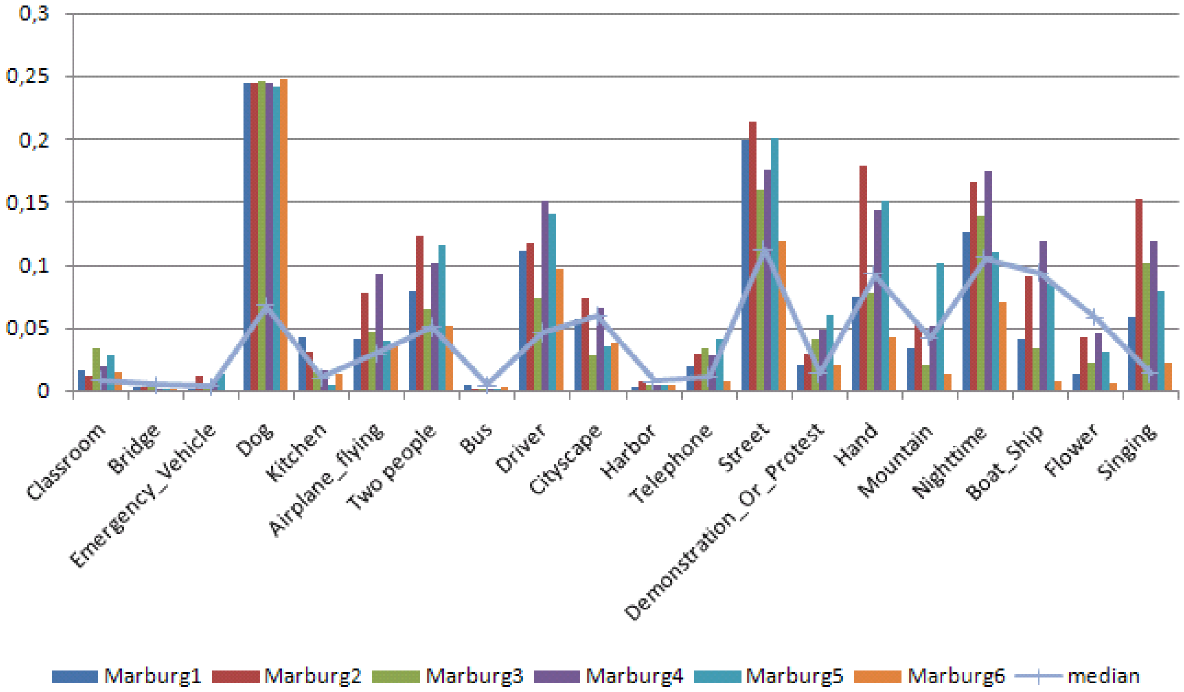

Figure 4: Comparison of our submitted runs concerning all evaluated high-level features. The median values refer to all submitted high-level feature extraction runs.

negative training samples. Once more, we applied sub-sampling of negative instances, but this time in combination with stratified threefold cross-validation in order to find the best sampling rate. Again we achieved clearly better results compared to the reference system A_Marburg3 (5.76% vs. 8.04% mean inferred average precision). Only the concept "Classroom" did not profit from the sub-sampling approach in both related experiments. Second, we increased the number of positive training samples by creating image variations, as described in Section 3. This run (Marburg5) achieved clearly better results as well (7.39% mean inferred average precision).

Particularly the result of the concept "Mountain" was strongly boosted by this approach and achieved our best result for this concept.

Finally, we applied stratified threefold cross-validation to find the best feature subset for each concept. Due to lack of time, we first performed cross-validation for each of our feature sets. Thereafter, we regarded the top four feature sets and validated all possible subsets. Using the concept models based on the best feature subset did not improve the detection results. The results decreased from 5.76% to 4.07% mean inferred average precision. Figure 3 gives an overview of the results of all our submitted runs.

5. Conclusions

In this paper, we presented our experiments for the high-level feature extraction task. Our high-level feature extraction of the last year was extended in several ways. First, we incorporated two new feature types, namely Hough and audio low-level features. These features did not contribute to an improvement of the results in terms of mean inferred average precision. Second, we tried to reduce the unbalance of positive and negative training samples, which most of the concepts exhibited. Both – the reduction of negative training instances by sub-sampling and the generation of positive samples by creating image variations – clearly improved the results. Interestingly, the use of cross-validation for performance estimation of models built on different feature subsets did not improve the detection results. This performance loss probably happened due to over-fitting.

In total, our best run based on sub-sampling negative instances obtained a mean inferred average precision of 8.27%.

6. Acknowledgements

This work is financially supported by the Deutsche Forschungsgemeinschaft (SFB/FK 615, Teilprojekt MT)

7. References

1. Ahmadi, S. and Spanias, A.S. Cepstrum-Based Pitch Detection Using a New Statisctical V/UV Classification Algorithm, *IEEE Trans. on Speech and Audio*, Vol. 7, No. 3, May 1999, pp. 333-338.

2. Aslam, J.A., Pavlu V., and Yilmaz, E. Statistical Method for System Evaluation Using Incomplete Judgments, Proceedings of the 29th ACM SIGIR Conference, Seattle, 2006.

3. Bouguet, J.-Y. Pyramidal Implementation of the Lucas Kanade Feature Tracker. In *OpenCV Documentation, Intel Corporation, Microprocessor Labs*, 1999.

4. Chang, C.-C. and Lin, C.-J. LIBSVM: A Library for Support Vector Machines. http://www.csie.ntu.edu.tw/~cjlin/libsvm

5. Ewerth, R., Schwalb, M., Tessmann, P., and Freisleben, B. Estimation of Arbitrary Camera Motion in MPEG Videos. In Proc. of the 17th International Conference on Pattern Recognition, Vol. 1. Cambridge, United Kingdom, 2004, pp. 512-515.

6. Gllavata J., Ewerth R., and Freisleben B. Text Detection in Images Based on Unsupervised Classification of High-Frequency Wavelet Coefficients. In *Proceedings of 17th Int. Conference on Pattern Recognition*, Vol. 1, Cambridge, UK, 2004, pp. 425-428.

7. Hu, Y. and Loizou, P. Subjective Comparison of Speech Enhancement Algorithms. In *Proc. Int. Conference on Acoustics, Speech, and Signal Processing*, Vol. 1, Toulouse, France, 2006, pp. 153-156.

8. Kruizinga P. and Petkov N. Non-linear operator for oriented texture, *IEEE Trans. on Image Processing*, 8 (10), 1999, pp. 1395-1407.

9. Li, D. and Sethi, I. MPEG Developing Classes. http://www.cs.wayne.edu/~dil/research/mdc/docs

10. Lienhart, R., Liang, L., and Kuranov, A. A Detector Tree of Boosted Classifiers for Real-time Object Detection and Tracking. In *Proceedings of IEEE Int'l Conf. on Multimedia & Expo,* , Vol. *2*, Baltimore, Maryland, USA, 2003, pp. 277-280.

11. Linguistic Data Consortium, The DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus, 1990, available online (09.02.2007): http://www.ldc.upenn.edu/Catalog/readme_files/timit.readme.html.

12. Lu, L., Zhang, H.-J., and Li, S. Z. Content-based audio Classification and Segmentation by using Support Vector Machines. In *Multimedia Systems 8*, 2003, pp. 482-492.

13. Lucas, B. and Kanade, T. An Iterative Image Registration Technique with an Application to Stereo Vision. In *Proc. of the Int'l Joint Conf. on Artificial Intelligence, Vancouver, Canada,* 1981, pp. 674-679.

14. Mühling, M., Ewerth, R., Stadelmann, T., Zöfel C., Shi B., and Freisleben, B. University of Marburg at TRECVID 2007: Shot Boundary Detection and High Level Feature Extraction. In Online Proceedings of TRECVID Conference Series 2007: http://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html

15. OpenCV, Open Computer Vision library, http://sourceforge.net/projects/opencvlibrary, 20.10.2008

16. Smeaton, A. F., Over, P., and Kraaij, W. Evaluation campaigns and TRECVid. In *Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, Santa Barbara, California, USA, 2006, ACM Press, New York, NY, pp. 321-330

17. Viola, P. and Jones, M. J. Robust Real-Time Face Detection. In International Journal of Computer Vision 57(2), Kluwer Academic Publishers, Netherlands, 2004, pp. 137-154.