Technische Universität Chemnitz at TRECVID Instance Search 2014

Marc Ritter[1], **Manuel Heinzig**[1], **Robert Herms**[2], **Stefan Kahl**[2], **Daniel Richter**[1], **Robert Manthey**[1], **and Maximilian Eibl**[2]

[1] Junior Professorship Media Computing, and
[2] Chair Media Informatics at Technische Universität Chemnitz, D-09107 Chemnitz, Germany

Abstract. This contribution describes our first appearance at the TRECVid Instance Search task (Over et al., 2014; Smeaton et al., 2006). Therefore, we try to verify our approach by introducing an extensible system architecture in order to process both subtasks of interactive and automatic runs using basic audiovisual concepts. The first approach incorporates an easy-to - the creation of a graphical user interface for faster assessment and evaluation by using well-known visual MPEG–7 descriptors in combination with the audio track to distinguish indoor and outdoor scenes with respect to a given query. In contrast, our automatic runs are mainly based on statistical assumptions about the distribution of shots and reverse shots around the appearance of the query samples in the video collection. All runs make use of an adaptable and easy-to-use keyframe extraction scheme that is based on the distribution of shot lengths and greatly reduces the number of frames to be processed by the entire indexing and retrieval system.

1 Structured Abstract

1. *Briefly, list all the different sources of training data used in the creation of your system and its components.*

- For training issues, we solely used the given master shot reference, and the audio and video tracks of the first video with ID 0 from the provided *BBC EastEnders* video footage.

2. *Briefly, what approach or combination of approaches did you test in each of your submitted runs?*

- Within the first interactive run I_E_TUC_MI_1, we are using MPEG-7 Dominant Color in combination with audio-based indoor/outdoor detection and a semantic shot composition that is based on around 1.1 million extracted keyframes.
- All other runs are based on a Probabilistic Run-length weighted Neighborhood Algorithm (PRNA) that is built on probabilistic assumptions about the occurrences of instances and thus shrinks the keyframe pool to around 6,700 available frames.
- Our second interactive run I_E_TUC_MI_2 combines the PRNA with a semantic shot composition based on the advanced dominant color descriptor.
- The previous configuration is applied to the fully automatic run in F_E_TUC_MI_3.
- A last run F_E_TUC_MI_4 validates the PRNA within a shot composition approach for similar shots in a specific environment using basic MPEG-7 descriptors like dominant colors and color layout.

3. *What if any significant differences (in terms of what measures) did you find among the runs?*

- We present an adaptable and easy-to-use keyframe extraction scheme in order to reduce the large amount of 42 million frames to 1.1 million keyframes that were used for indexing or instance comparison at I_E_TUC_MI_1.
- A further reduction to barely 6,700 keyframes was achieved by using our proposed PRNA approach without affecting our results significantly at I_E_TUC_MI_2.
- As expected, and in terms of MAP, there is a significant difference between both interactive and fully automatic runs.
- The results of the runs with PRNA are promising within Precision at rank 30 (P30). Since these probabilistic methods depend on initial starting points, the score drops heavily afterwards due to a lack of occurrences.

Correspondence to: Marc Ritter
marc.ritter@informatik.tu-chemnitz.de

4. *Based on the results, can you estimate the relative contribution of each component of your system/approach to its effectiveness?*

- The reduction scheme of extracting representative keyframes via preprocessing or even PRNA is crucial to an efficient further processing.
- The user-defined database functions allow a fast comparison of the descriptors even on mid-tier computing architectures.
- The usability of our interactive GUI seems appropriate to improve the results while allowing a fast rejection of false positives.

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

- The *Dominant Color Descriptor* applied on small image blocks of 48×48 pixel is vulnerable against color noise in such large data sets with more than 75 million descriptor entries. The Euclidean distance measure appears to be insufficient for a reliable application. Complementary features are necessary to improve Precision and Recall.
- The PRNA method seems to be an usable heuristic for finding a set of new shots containing an instance based on some detected samples in the direct or indirect neighborhood, especially to boost the top 30 result entries.
- Tiny keyframes are effective for matching video footage of the same source that has been lossy encoded.
- In our opinion, the inclusion of the audio tracks seems promising despite we could not measure the direct performance gain within the current setup.

The remainder of the paper is organized as follows: Section 2 provides a general view about the basic concepts and more common components of our system architecture and the underlying workflow for both run types. The specific algorithms that were used within the system, are described in Section 3. Remarks regarding the official evaluation results are given in Section 4 followed by some conclusions in Section 5.

2 System Architecture

In the following, we give an overview about the core components of our system (cf. to Section 2.1) that are crucial to accomplish the instance search task. The proposed keyframe extraction scheme and necessary preprocessing steps that are directly applied to the original video footage and sample queries of the topics are discussed in Section 2.2. To cope with the large amount of data, a database is essential (Section 2.3) to store any extracted descriptors and to calculate distances with respect to the search query inline in order to deliver the most similar instances (Section 2.4).

2.1 Overview

The basic scheme of our system appears similar to classic approaches to Image Retrieval and other systems previously developed in the context of TRECVid, such as (Gupta et al., 2012; Mukai et al., 2011; Natsev et al., 2010).

We present two different approaches to tackle the different requirements of interactive and fully automatic runs. Therefore, we focus on the illustration of the complex workflow of our system for the interactive run in more detail. In contrast, the presentation of the system for the automatic runs appears rather short while solely consisting of a concatenation of methods being described in the subsequent section.

2.1.1 Interactive Run

Within the interactive runs (see Figure 1), we firstly preprocess both query data and the video collection with different steps to extract and build a solid and consistent base of data (refer to Section 2.2). Visual descriptors are extracted at the position of the requested object and the size of its smallest bounding box area. Additionally, we investigate the audio track of the sample video in order to automatically estimate whether it belongs to an indoor or outdoor concept. Human operators were enabled whether to make use of or neglect this information at the beginning of a run leading to distinct retrieved datasets.

In contrast to the single images of the queries, the video footage consists of more than 41,760,000 frames. A reduction step is introduced that splits each video of the test collection into representative units of keyframes decreasing the total number of frames to be processed any further to 2.6 percent of the original volume resulting in 1,145,775 frames in total. This yields to a significantly decreased amount of occupied disc space from 286 GB in the video domain to 64 GB of directly accessible images. In order to locate the requested objects in different sizes and positions, we subdivide each frame by a grid at different resolutions, where basic visual features are extracted (compare to Section 3) from each cell.

While the extracted descriptors are stored in a database, distances between the descriptors and an incoming query are calculated using distance measures that depend on the underlying descriptor. We achieve a great speed-up by utilizing the capabilities of user-defined functions from PostgreSQL that allow us to perform any similarity computation directly in the database. During the next step, the best results are retrieved in ascending order, one keyframe per shot being stored in an image file. These files are successively loaded in chunks of six images into a graphical user interface that quickly allows to assess the absence or presence of an object. In compliance to the rules of interactive runs, we save the period of time that has already been expired since the start of the search whenever a positive result is retrieved. Finally, the shot IDs of the positive evaluated keyframes are extracted and stored with their evaluation time stamps into the TREC XML result file.

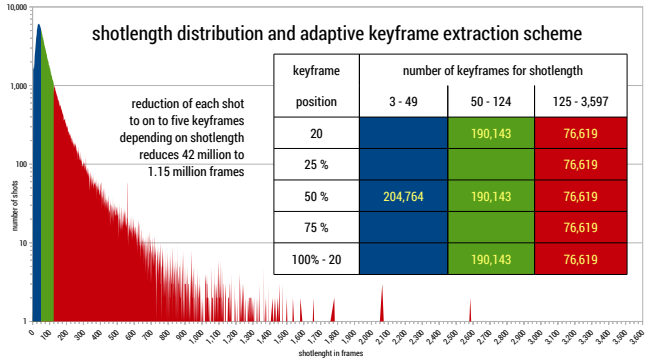

Figure 2. Distribution of the shot lengths with an overview about the locations of the extracted keyframes measured in frames with respect to the beginning of a shot.

2.1.2 Automatic Runs

In order to accomplish the task for fully automatic system execution, we apply a probabilistic algorithm (refer to Section 3.2) that follows basic assumptions while being concerned with a repetitive and therefor dependent occurrence of instances in direct neighborhood of a known instance sample. Therefrom, we match features extracted from the given master shot references with the ones provided alongside the query object. This can be referred to as an approach to duplicate detection using the information from the query video. Furthermore, we were able to combine this approach with a shot composition and Advanced Dominant Color descriptors, defined in Section 3.4, to infer a grouping of scenes rather than shots.

2.2 Preprocessing and Keyframe Extraction

Our different approaches for feature extraction demand an abundant preprocessing on the given data. The underlying video collection consists of 244 MPEG-4 video files where each contains four omnibus episodes of around 30 minutes plus short additional video sequences like advertisements. The first step is concerned with splitting the data collection into the 471,526 automatically determined shots according to their starting and ending points that are given in the *master shot reference table*. This task can be easily accomplished by utilizing FFMPEG[1] via command line while being accompanied by a deinterlacing procedure that is based on the built-in YADIF filter, and a correction of the pixel aspect ratio to squared pixels by stretching the anamorphic images. Besides extracting the video to a full image size of $1{,}024 \times 576$ pixel, we extracted a reduced version with 456×256 pixel, and created an audio-only version at 16 kHz mono in 16 bit PCM format.

To further reduce the information that needs to be processed by our image processing approaches, we decided to extract representative frames from each shot that we refer to as keyframes. The amount of frames extracted from each shot is determined by its shot length. Figure 2 shows the distribution of the shot lengths provided by the *master shot reference table*. We found the following trivial selection scheme to work nicely: Single keyframes are selected from the middle of short clips lasting less than two seconds, and two additional frames at the beginning and the end when lasting up to five seconds. In order to adapt to camera panning in longer shots or a change in the background, for example by closing or opening a door, another two additional frames are extracted at 25 percent from both shot boundaries. Beyond, we introduce a safety margin of 20 frames from the shot boundaries in order to mitigate predictable side effects of imprecisely located shot boundaries that are inherent to the automatically determined *master shot reference table*, and that might prove crucial to our approach that clearly neglects any spatiotemporal information of intermediate frames. Besides, all keyframes were saved in JPEG-format with highest quality settings. In order to prevent any statistical corruptions in the latter feature extraction process by black borders or other artefacts at the margins of the images, we crop each image by default at its full resolution by 8 pixels in each direction.

The anamorphic equalization of pixels of the test collections also forces us to apply these operations on the query images to be capable of retrieving similar distortion-free instances. Another reason is the capability to derive the audio-based indoor/outdoor-concept of a given sample shot of the query image. This is achieved by a conversion of the 120 video samples of all 30 topics from WEBM- to MPEG-4 format while applying deinterlacing and stretching operations as described above.

Since our PRNA method relies on the assumptions that potential instances could be found in the direct spatiotemporal neighborhood of the sample query, we extract the first frame of each shot with an image size of 32×18 pixel in uncompressed bitmap format from all available data sources.

2.3 Database

The need of calculating results for many querys and the requirement to order them for a proper submission led to the approach of storing all values in a database with the aim to avoid a repeated recalculation. For this purpose, we decided to use PostgreSQL as our central database repository with its sophisticated capabilities to efficiently process large amounts of data. The native support of array data types provides an easy and convenient way to store the results of the extracted visual MPEG-7 features like Dominant Color or Color Layout Descriptors (cf. to Section 3.1).

We calculate the distances between the descriptors of the query and each block of the video corpus directly in the database via user defined functions (UDF). Following this approach grants us direct access to the data without any intermediate steps or data transfer that usually slows down any

[1] http://www.ffmpeg.org, 10/17/2014

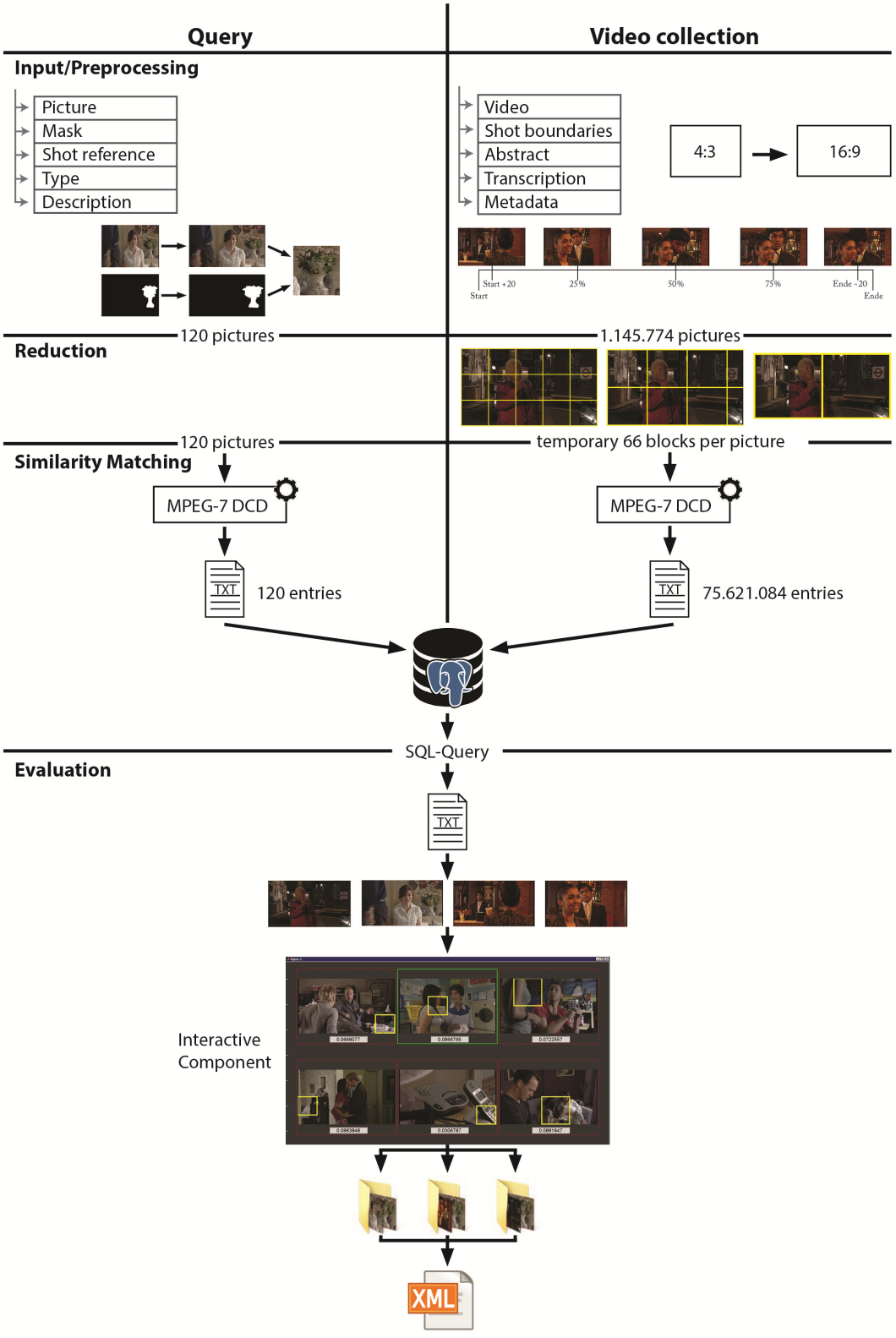

Figure 1. Basic workflow of our system at the example of the MPEG-7 Dominant Color Descriptor used within our interactive experiments.

process. Besides distributed calculation in the cloud and GPGPU computing, we consider this method as a fast solution to reliably exhaust multi-core processing systems. Since the calculation between the queried descriptor and the stored descriptors can be considered independent within our grid-based approach, the calculation process can be speeded up by a core-wise parallel subdivision of the overall amount of data.

We use PostgreSQLs native programming language PL/PgSQL (The PostgreSQL Global Development Group, 2014) to reimplement the well-known MPEG-7 distance measures from *Caliph & Emir* (Lux, 2009) for the Dominant Color and Color Layout Descriptors, respectively. Setting the volatility parameter to *immutable* enables a parallel execution with read-only access per table resulting in a massive speed improvement (Quad-Core system with hyperthreading resulting in eight cores in total) of more than ten thousand times faster in comparison to a calculation with the Java libraries of *Caliph & Emir* or multiple thousands compared to *MPEG-7 FexLib* (Bastan et al., 2010).

2.4 Querying

Before a query is processed, the previously described steps are applied. These data units are further processed by a Java program that includes the *Caliph & Emir Image Retrieval library* (Lux, 2009) to extract the *MPEG-7 based dominant color* features. To not hinder calculation and to prevent transfer failures to happen, the resulting features are firstly stored by writing them into a plain text file. Another Java tool parses the contents of this file into the database with help of the appropriate JDBC-Driver. To minimize the transfer overhead, this is done by splitting the stream of datasets into chunks of 100 units per SQL query. Once the database is filled, calculation is initiated via a multi-threaded program in Java, in which every thread builds its own database connection and calls the previously implemented UDF distance functions on a fraction of the data stock. The results of this calculation are then stored into another table. A set of results is now fetched from the database using SQL-SELECT statements in order to deliver the required 1,000 different shots.

Within the interactive runs, the calculated outcome was intellectually verified for correctness using a self-built graphical user interface (see *Interactive Component* in Figure 1). The classification results are arranged in a folder structure that is easily readable by other programs or humans. To satisfy the stipulation of the underlying task, the evaluated folder structure is then parsed and converted into an XML file.

3 Algorithmic Approaches

Our portfolio of algorithmic approaches comprises visual feature extraction procedures using dominant colors (Section 3.1), probabilistic assumptions about the structure and occurrences of queries and instances in the video footage (Section 3.2), and an incorporation of the audio track with the aim to distinguish indoor and outdoor scenes (Section 3.3). It is complemented by methods that group adjacent shots into higher-level scenes (Section 3.4).

3.1 Visual Feature Extraction

To counter the problem of objects appearing in a variety of different sizes, we extract features not only from full images, but also from the rectangular cells of a grid structure. A basic assumption is that within a tile an object might be represented at the same zoom level as in its query picture. As a consequence, the size of those blocks is based on the approximate dimensions of the smallest queries. In the past years, the query instances were based around a minimum size of 35 to 40 pixel in at least one dimension. Hence, we decided to use an enlarged block size of 48 x 48 pixel.

According to this scheme, the shots that originated from the preprocessed videos are scaled down, e.g. when we want to make use of a 5×3 block pattern, we first need to create a rescaled copy of the full image containing 1024×576 pixels to just 240×144 pixels. The opportunities that objects vary in their size, pose or point of view with respect to the image query, their representations are taken into consideration by extracting three different scaled grids, thus building a three layered pyramidal structure with resolutions of 240×144, 168×96 and 96×48 pixels.

With this block building approach, an object might overlap at the edges of a tile and therefore might be split up into several parts, making it much harder to detect. We eliminate this issue by shifting the grid for half the size of a subsection once in vertical as well as in horizontal direction. In this way, we end up with a representation of 66 different blocks at multiple resolutions for each image from where arbitrary features might be extracted per block.

The descriptor of our choice is the well-known *Dominant Color Descriptor* originating in ISO 15938, also known as MPEG-7 standard (Ohm et al., 2001). It extracts a maximum of eight dominating colors from a given image, weights those colors by providing their individual degree of influence via a percentage value and calculates a spatial distribution throughout the image. We use the implementation of the *Caliph & Emir* (Lux, 2009) image processing library that still neglects the spatial distribution using standard parameters in its most recent revision. All image processing operations are utilized within efficient multi-threaded video processing chains of the AMOPA framework (Ritter, 2014; Ritter and Eibl, 2011).

3.2 Probabilistic Run-length weighted Neighborhood Algorithm

This approach is based on two assumptions. The first one is that longer shots have a higher probability of containing a

	Indoor			Outdoor			Correct
Γ	P	R	F_1	P	R	F_1	(%)
−11.0	0.849	0.776	0.811	0.429	0.550	0.482	72.27
−11.5	0.907	0.776	0.836	0.503	0.742	0.599	76.76
−12.0	0.890	0.640	0.745	0.387	0.742	0.509	66.41

Table 1. Experimental results of different log-energy thresholds (Γ) for the indoor outdoor classification (Precision, Recall and F_1).

searched instance than shorter shots. Let Π denote the target instance (shot number of the query) of a given sample shot in the test collection. The second assumptions states that there is a higher probability that similar object instances are more likely to be contained in the neighborhood Ω around Π, whereas the probability $P \propto \Delta(\Pi,\omega)^{-1}$ decreases while enlarging the distance between Π and a specific location $\omega \in \Omega$.

Hence, the approach takes all shots in the neighborhood around sample shots known to contain a searched instance, weighting them by distance (number of intermediate shots) and run-length to create an ordered list.

As mentioned before, a successful application of this approach requires prior knowledge about some shots containing a searched instance. In our runs, we used the four shots given as example for each topic. Because information about the originating sequences of the query shots were not given, it was mandatory to retrieve them. This was accomplished by utilizing lower-level methods of duplicate detection theory. Accordingly, we employed a binary comparison of tiny bitmap pictures from the topic sample clips with all previously extracted bitmap pictures of the shots from the dataset (cf. Section 2.2). This method benefits from some speed-ups by including the meta data from each video to preselect a certain set of shots, or by loading this preselected set of shots into RAM and executing all byte-by-byte-comparisons in memory. Due to the compression differences between the query format and the video footage, the sets did not contain any complete identical pictures. That's why the differences of each two bytes at the same positions were calculated and accumulated for the entire file. When this accumulated value achieved a new maximum in comparison to the best already calculated sum of another file or exceeded a certain threshold (we used the length of the file multiplied by eight), the system could move forward to the next file. This method allowed us to identify the shot numbers of all 120 topic examples in the entire video footage in less than 200 seconds, in sum within a range of 0.16 to 27.52 seconds per shot (3 comparisons per millisecond in average) on some mobile *Core i5* laptop.

We found that two topics only contained example shots from the development set. Thus we were not able to identify shots for those cases in the test set. Heuristically, we used a list of the longest shots in descending order to fill up the result lists to 1,000 shots.

3.3 Indoor/Outdoor Detection on Audio

The indoor and outdoor classification of shots aims to increase the precision in an interactive run. For example, a human operator might reason that it is likely that the cell phone instance search task might always be located outdoors what could lead to additional constrains in the SQL statement while retrieving the results from the database. In the initial analysis of the development set a lower background noise in most of the indoor shots has been perceived. This is reflected in the log-energy feature between the spoken utterances of the actors. However, there are many shots that last for less than 2 seconds including just speech with no opportunity to discriminate between indoor and outdoor.

We assume that a scene, consisting of more than one shot, is either indoor or outdoor, and there is at least one shot in a scene lasting long enough to analyze non-speech signals. In this context, our approach for the indoor/outdoor classification works as follows:

1. The indoor/outdoor classification of shots based on the log-level feature and functional.

2. The indoor/outdoor classification on scene-level depends on the number of indoor shots.

3. If one shot meets the indoor requirements, all other shots in the same scene are considered as indoor.

The provided dataset for development in the challenge consists of 1,997 Shots. To evaluate our approach 512 shots were intellectually grouped into 25 indoor scenes (392 Shots) and 10 outdoor scenes (120 Shots). We defined a training set that consists of the rest of the samples whereas only appropriate samples were selected. We just used the ones that last longer than two seconds in order to fulfill the requirements of our approach, and thus to experimentally investigate the appropriate threshold of the log-energy feature for the discrimination process. Consequently, the number of training samples varied in each conducted experiment.

We used the functionals minimum, range, mean, and standard deviation of log-energy and evaluated different classifiers. The Support Vector Machine (SVM) of LIBSVM (Chang and Lin, 2011) within WEKA (Hall et al., 2009) performed best. The experimental results of our approach are shown in Table 1. It is obvious that the best F_1 score for the classes indoor and outdoor is at −11.5 of the log-energy yielding to a correctness of 76.76%. The choice of samples for training based on the threshold is crucial as it can be seen at the difference of 10.35% of correctness between the thresholds −12.0 and −11.5. Finally, we selected the model with the best correctness result and applied it within our architecture on the actual test set of the challenge.

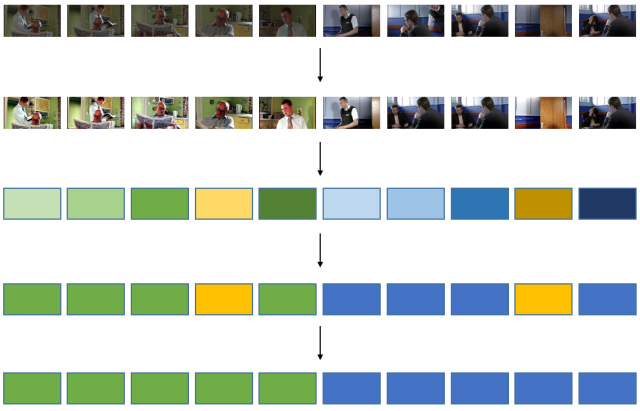

Figure 3. Row 1: Sequence of keyframes. Row 2: Brightened keyframes using histogram equalization. Row 3: Sequence of characteristic colors extracted by k-means clustering. Row 4: Normalization of the color channels to values 0, 128, and 255. Row 5: Single outliers have been eliminated resulting in an emphasized scene-to-scene transition.

3.4 Shot Composition and Advanced Dominant Color

The slightly different approach using Advanced Dominant Color (ADC) is used to find semantic linked shots in direct temporal connection. It was used to group related shots to one scene. It is based on the assumption that related shots which were filmed at the same location have a quite similar overall coloring. This assumption is backed by the observation that the different but quite limited locations in a daily soap are all designed in a special way to facilitate the recognition for the audience, whereas the provided data set of the BBC soap *EastEnders* seems to meet this prerequisites.

Figure 3 illustrates our approach. ADC is based on one Dominant Color (DC) which is extracted from an image by *k-means clustering* (MacQueen, 1967), but with exactly one cluster. Therefor the most dominant color is extracted. This process is limited to especially colorful colors meaning that in RGB *color space* one of the three color channels is quite different from the others in its value. This is also supported by a color histogram spread on each channel. However, the spread on the red channel is somehow limited to prevent a red cast which would make the final result less useful. The resulting RGB values of the DC are discretized to the three values 0, 128 and 255 leading to only 27 distinct final colors.

However, in some shot/reverse shot-situations these overall colors may differ too much yielding to different ADCs. There may be some different reasons for accidentally distinct ADCs in a series of successive shots that could be eliminated by using a smoothing technique.

4 Results

We submitted four different runs. Two interactive with a sophisticated approach and two automatic based on rather simple estimations. Both runs returned rather mediocre results placing us in the lower-middle tier in comparison to other participants this year.

4.1 Interactive Runs

As expected by the usage of the Dominant Color Descriptor, resulting images had a color scheme similar to the queried pictures. Despite downsampling the input, the results were adequate. However, the setup did not necessarily find correct object representations, as toning of two different objects can also match. To a certain point, problems like this should have been absorbed by the spatial coherency value of the descriptor, but the provided implementation seemed to struggle calculating those values. Hence, the system had no information about the structural behaviour of the picture, and object detection was purely depending on the colors. We finished with a *Mean Average Precision* of 0.037 for I_E_TUC_MI_1 and 0.034 for I_E_TUC_MI_2.

Due to our reduction and usage of in-database calculation the runtime of the complete system was approximately half the runtime of the initial video collection, which is a notable insight, since we only used a single *Dual Quad-Core* Xeon 5472 system for computation.

4.2 Automatic Runs

Both fully automatic runs were based on the Probabilistic Run-length weigthed Neighborhood Algorithm (PRNA) described in Section 3.2 and the approaches concerning shot composition and Advanced Dominant Color described in Section 3.4. They were applied to the identified sample shots, and by constraining its outputs to the shots identified by the scene detection based on the ADC the SC for these example shots in I_E_TUC_MI_3 or on MPEG-7 descriptors on I_E_TUC_MI_4.

In contrast to the approach described above, the result lists were not filled up by the list of the 1,000 longest shots, and this default list was only returned when there were no other results. The aim was to improve the precision over the recall based pure PRNA approach. In the evaluation, we achieved a MAP of 0.015 for run _E_TUC_MI_3 and 0.017 for run I_E_TUC_MI_4 over the 27 evaluated topics. In contrast, run I_E_TUC_MI_3 scored the highest mean (Precision at total relevant shots) of 0.047 among our submitted runs. Overall, we were ranked in the lower mid-field in the fully automatic runs.

5 Conclusions

We introduced an extensible system architecture to process both subtasks of interactive and automatic runs using basic audiovisual concepts, and an adaptable keyframe extraction schemes based on a shot length distribution to vastly reduce the amount of data to process. We demonstrated an effective

way of MPEG-7 descriptor distance measure calculation by using internal PostgreSQL database functionality.

A lightweight and easy to use UI allows faster judgement and evaluation. We combined well-known visual MPEG-7 descriptors with the audio track to distinguish indoor and outdoor scenes with respect to a given query for our interactive runs. Designing a framework representing an extensive image processing workflow and collaborative evaluation proved to be very effective despite mostly mediocre results. The overall performance of this approach can be improved by using additional visual descriptors, and by making use of more sophisticated machine learning strategies. We fully expect better results by further development.

In contrast, our automatic runs are mainly based on statistical assumptions about the distribution of shots and reverse shots around the appearance of the query samples in the video collection. Although hardly any image processing algorithms were implemented, those runs exceeded our expectations in terms of precision at least for the top 30 results (P30). We aim for a combination of those statistical features and additional visual MPEG-7 descriptors together with SIFT and bag-of-words models in our future work.

Using semantic features as additional ranking constraints is a promising way of improving both visual feature extraction through search space optimization/minimization, and overall relevance of the resulting set of shots. The search of a given instance in large corpora will benefit from the extraction and use of non-visual features and metadata. In order to extend our framework in the future, we aim to pursue this multimodal approach.

Acknowledgements. This work was partially accomplished within the project ValidAX — Validation of the AMOPA and XTRIEVAL frameworks (grant no. VIP0044), and localizeIT (grant no. 03IPT608X) funded by the Federal Ministry of Education and Research, Germany as well as the project Chrooma+ supported by the Sächsische Aufbaubank within the European Social Fund in the Free State of Saxony, Germany. Any programme material is copyrighted by BBC.

References

Bastan, M., Cam, H., Güdükbay, U., and Ulusoy, Ö.: Bilvideo-7: an MPEG-7- compatible video indexing and retrieval system, IEEE MultiMedia, 17, 62–73, doi:10.1109/MMUL.2010.5692184, 2010.

Chang, C.-C. and Lin, C.-J.: LIBSVM: a library for support vector machines, ACM Transactions on Intelligent Systems and Technology (TIST), 2, 27, 2011.

Gupta, V., Varcheie, P. D. Z., Gagnon, L., and Boulianne, G.: Content-based video copy detection using nearest-neighbor mapping, in: Proceedings of the International Conference on Information Science, pp. 918–923, Centre de recherche informatique de Montreal (CRIM), doi:10.1109/ISSPA.2012.6310685, 2012.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H.: The WEKA data mining software: an update, SIGKDD Explorations, 11, 10–18, http://www.cs.waikato.ac.nz/~ml/publications/2009/weka_update.pdf, 2009.

Lux, M.: Caliph & Emir: MPEG-7 photo annotation and retrieval, in: Proceedings of the 17th ACM International Cconference on Multimedia, pp. 925–926, doi:10.1145/1631272.1631456, 2009.

MacQueen, J.: Some methods for classification and analysis of multivariate observations, in: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics, pp. 281–297, University of California Press, Berkeley, Calif., http://projecteuclid.org/euclid.bsmsp/1200512992, 1967.

Mukai, R., Kurozumi, T., Kawanishi, T., Nagano, H., and Kashino, K.: Content-Based Copy Detection, in: Proceedings of the International Conference on Information Science, NTT Communication Science Laboratories, 2011.

Natsev, A., Smith, J. R., Hill, M., Hua, G., Huang, B., Merler, M., Xie, L., Ouyang, H., and Zhou, M.: TRECVID-2010 Video Copy Detection and Multimedia Event Detection System, in: Proceedings of the International Conference on Information Science, IBM Research, 2010.

Ohm, J.-R., Cieplinski, L., Kim, H. J., Krishnamachari, S., Manjunath, B., Messing, D. S., and Yamada, A.: The MPEG-7 Color Descriptors, in: Introduction to MPEG-7: Multimedia Content Description Interface, edited by B. S. Manjunath, Philippe Salembier, T. S., The MPEG-7 Multimedia Description Standard, 2001.

Over, P., Awad, G., Michel, M., Fiscus, J., Sanders, G., Kraaij, W., Smeaton, A. F., and Quenot, G.: TRECVID 2014 – An Overview of the Goals, Tasks, Data, Evaluation Mechanisms and Metrics, in: Proceedings of TRECVID 2014, NIST, USA, http://www-nlpir.nist.gov/projects/tvpubs/tv14.papers/tv14overview.pdf, 2014.

Ritter, M.: Optimization of algorithms for video analysis: A framework to fit the demands of local television stations, in: Wissenschaftliche Schriftenreihe Dissertationen der Medieninformatik, edited by Eibl, M., vol. 3, pp. i–xlii,1–336, Universitätsverlag der Technischen Universität Chemnitz, Germany, http://nbn-resolving.de/urn:nbn:de:bsz:ch1-qucosa-133517, 2014.

Ritter, M. and Eibl, M.: An Extensible Tool for the Annotation of Videos Using Segmentation and Tracking, in: Design, User Experience, and Usability. Theory, Methods, Tools and Practice, edited by Marcus, A., vol. 6769 of *Lecture Notes in Computer Science*, pp. 295–304, Springer Berlin Heidelberg, doi:10.1007/978-3-642-21675-6_35, 2011.

Smeaton, A. F., Over, P., and Kraaij, W.: Evaluation campaigns and TRECVid, in: MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval, pp. 321–330, ACM Press, New York, NY, USA, http://doi.acm.org/10.1145/1178677.1178722, 2006.

The PostgreSQL Global Development Group: PostgreSQL 9.3.5 Documentation - PL/pgSQL - SQL Procedural Language, http://www.postgresql.org/docs/9.3/static/plpgsql-overview.html, 2014.