On-line Video Synchronization Based on Visual Vocabularies

Vítězslav Beran, **Pavel Zemčík**, **Adam Herout**

Department of Computer Graphics and Multimedia
Faculty of Information Technology
Brno University of Technology
Božetěchova 2, 612 66 Brno, CZ
e-mail: {beranv, zemcik, herout}@@fit.vutbr.cz

Submitted: 24/02/2010
Accepted: 10/03/2010
Appeared: 20/03/2010

Abstract: This paper presents a procedure for on-line visual-content-based video synchronization. The motivation of the presented work is the rare evidence of on-line video processing systems employed in video classification or summarization applications although many off-line solutions for video analysis exists. In some applications, the video streams go through broadcast systems that delay the original video and also distort the original signal. The system that would be able to automatically detect such delays, transmission errors, distortions, or broadcast failure is highly required. The presented solution employs visual vocabularies that allow signing the video frames by bag-of-words. The synchronization procedure is based on searching for similar frames from different video streams. This paper also overviews the state-of-the-art techniques required for visual vocabulary building and discusses the convenient properties of techniques for real-time on-line systems.

Keywords: On-line video processing, visual vocabulary, video synchronization

1. INTRODUCTION

The main task of video synchronization systems is to detect and validate the time offset between similar video streams. The motivation of the presented work is lack of the on-line video processing solutions. Having two or more video streams, the goal is to find their time shift in real-time. The video analysis should be done on-line and based mostly on visual content.

The presented approach is focused on visual content analysis. The previous approaches, analyzing the visual video content on-line in real-time, are based mostly on global features are poorly robust to geometrical transformations that led presented research to employ local features and visual vocabularies.

The visual vocabulary method was introduced by Sivic and Zisserman (2006). Motivated by text retrieval techniques where the document is represented as a set of *textual* words, the method represents the image content using *visual* words. Images are analyzed and local features are extracted. Having the visual vocabulary, the local image features are translated into pre-trained visual words. The image content is then represented as a set of visual words (also known as set-of-words, bag-of-words or image signature). The work of Sivic and Zisserman (2006) utilized widely used SIFT transformation by Lowe (2004) as local features. The visual vocabulary training was based on naive k-means algorithm. When introduced, the visual vocabularies were used in image and object retrieval applications. Similarly to text retrieval, the database of images is indexed by inverted file approach. Search for a query image in database results in immediate returning of a ranked list of documents (key frames, shots, etc.) similarly to search in text documents.

Later works utilized more types of local image features such as corner-like detector with full affine adaptation (Mikolajczyk and Schmid (2004)) or detection of stable regions (MSERs by Matas et al. (2002)). When the local features are finally described, the performance of the visual vocabulary approach correlates to performance of particular image feature extraction technique. Needs for large and more discriminative vocabularies lead the research to find faster clustering methods. Two significant methods were developed: hierarchical k-means (Nister and Stewenius (2006)) and approximated k-means (Philbin et al. (2007)). The developed methods allow creation of vocabularies with size about 1M of visual words with reasonable time and computational cost.

The later research experimenting with different translation schemas when translating the local image features into bag-of-words showed that the discriminative power of the vocabulary could be improved not only by extending the vocabulary size. Instead of the standard approach, where one local feature is translated in just one visual word, a single image feature could be assigned to several visual words. The approach is known as *soft-weighting*. The examples of different soft-weighting results are published by Philbin et al. (2008) and Jiang et al. (2007). Further improvements in precision and number of retrieved candidates were done by Chum et al. (2008), where authors employed *minimal hashing* algorithm and later also utilized the *geometrical* information from local features (Chum et al. (2009)). The methods dealing with NDID (Near-Duplicate Image Detection) or NDSD (Near-Duplicate Shot Detection) are mainly focused on searching the most similar samples in the image database. In this case both global image features (color histograms with locality sensitive hashing) and local features are used to search for near-duplicates in the large image database (Chum et al. (2007)).

The paper firstly introduces the overview of two stages of the video synchronization system. The following Section contains an overview of state-of-the-art local image feature extraction and description methods explaining their crucial attributes for real-time systems. The process of building the visual vocabulary is described in Section 3 together with vocabulary searching methods. The possibilities of visual words weighting when bag-of-words are constructed are discussed in Section 4. Sections 6 and 7 describe the experiments - the used data and the explored transformations and discusses the results.

2. IMAGE LOCAL FEATURES

Real-time applications, such as on-line video synchronization introduce specific demands to the commonly used techniques. The attributes of image local feature extraction methods are stability, repeatability, and robustness to several types of transformations or distortions (Mikolajczyk et al. (2005). The characteristic of feature descriptor is its discriminative power. Usually, the more powerful the feature extraction and description methods are, the higher is their time cost. For the real-time applications dealing with consecutive video-frame processing, the methods performance could be decreased at the expense of execution time increase. The computational cost demands also deriving the size of the visual vocabulary.

The local image features can be constructed on two fundamental image structures: corners or regions. The detected local image features are expected to be invariant to geometric and illumination changes. Different detectors emphasize different aspects of invariance, resulting in keypoints of varying properties and sampled sizes. See examples of local features in Figure 1.

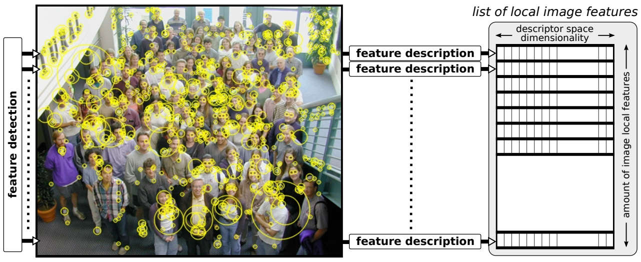
list of local image features
descriptor space dimensionality
feature detection
feature description
feature description
feature description
amount of image local features

Fig. 1. Image local feature detection and description.

One subset of detectors analyses the local changes in image intensity. The scale-adopted Harris function (Mikolajczyk and Schmid (2004)) is sensitive to corner-like structures. The Hessian (Mikolajczyk et al. (2005)) function, Difference of Gaussian (Lowe (2004)) or Laplacian of Gaussian (Lindeberg (1998)) detect the blob-like structures. The Harris and Hessian function are extended by characteristic scale detection which is where some characteristic function (e.g. Laplacian of Gaussian) attains a maximum over a scale. To obtain the affine invariance of the detectors, the affine adaptation process is included that iteratively adapts the region shape by maximizing the intensity gradient isotropy over the elliptical region. Later published results by Bay et al. (2006) show that the Hessian function can be effectively approximated by block filters. The SURF detector is based on effective platform computation of Haar-wavelets on integral images. The authors also introduced new descriptors utilizing the same platform.

The approach known as *FAST corners* (Rosten and Drummond (2006)) employs machine learning to construct a corner detector that outperforms all know approaches in the speed point of view. The FAST itself is neither invariant to scale nor to shear. When full affine invariance is necessary, the techniques mentioned above (characteristic scale selection and affine adaptation) can be applied. Unfortunately, they significantly slow down the process. In the presented application, only slight geometrical transformations are supposed so the fully affine invariance is not needed. The detected corners are described using a *gradient location and orientation histogram* (GLOH) designed by Mikolajczyk and Schmid (2005).

The region-based detectors are represented by *maximally stable extremal regions* (MSER) developed by Matas et al. (2002). The candidate region is a connected component of an appropriately thresholded image. The *extremal* refers to the property that all pixels inside the MSER have either higher or lower intensity than all the pixels on its outer boundary. The process of selecting the threshold optimizes the region stability to be maximal. Table 1 shows the precisions and speeds of some image local feature extractors. These preliminary experiments were evaluated on Kentucky (and Oxford) dataset (see Section 6 for more details) using kd-tree approach dataset (see Section 3) for vocabulary searching. The image retrieval precision is used to measure the system performance. The number of features is only orientation number because the amount of extracted feature depends not only on image size but also on parameters of the extractor. Also the method's speeds depends on the amount of extracted features. The speed is measured in seconds per frame. In the on-line real-time video processing systems, the speed factor is the most crucial. The SURF detector offers convenient balance between speed, robustness, precision and discriminative power. The presented system is based on the SURF detector and descriptor in its extended version that calculates 128-dimensional feature vectors.

Table 1. Comparison of local image feature extractors.

	SIFT	SURF	FAST+GLOH
precision	66.9%	56.9%	32.8%
# of features	1000	500	400
speed[s]	7.2	1.0	0.3

3. VISUAL VOCABULARY

The idea of visual vocabulary, firstly used in *Video Google* by Sivic and Zisserman (2006), brings the techniques from natural language processing and information retrieval area. The document (image) is represented as an unordered collection of words (bag-of-words model). In computer vision, the (visual) words might be obtained from the feature vectors through a quantization process. The objective is to use vector quantization to descriptors to translate them into clusters' labels which represents the visual words.

The visual vocabulary is built during the training stage. A part of the data (training data) is used to divide the descriptor space into clusters. Each cluster is labeled; it has its own identification number. The vocabulary is a list of cluster centers and identifiers. The clustering procedure based on k-means algorithm contains the search step, when the sample should be assigned to the nearest. The later research introduced several solutions to avoid time consuming naive sequential search. Figure 2 schematically shows the different approaches described in detail below.

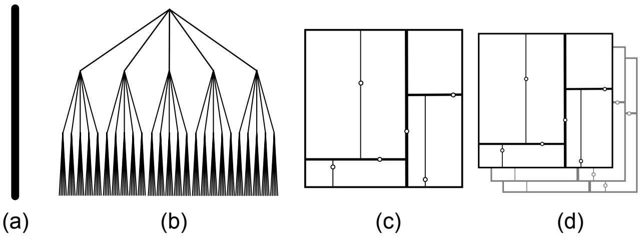

Fig. 2. Clustering strategies - (a) naive sequential, (b) hierarchical, (c) kd-tree and (d) random forest.

When the size of the resulting vocabulary is small ($k < 10^3$), the *naive k-means* algorithm can be used (Fig. 2(a)). The time complexity of the k-means algorithm is $O(kN)$, where N is the number of training feature vectors. Some applications (e.g. for object retrieval (Philbin et al. (2007))) need more discriminative vocabulary. One possible way how to reduce the time complexity is using *Hierarchical k-means* introduced by Nister and Stewenius (2006). Instead of solving one clustering with a large number of cluster centers, a tree organized hierarchy of smaller clustering problems is solved (Fig. 2(b)). This reduces the time complexity to $O(N. \log k)$. The problem with Hierarchical k-means is that it optimizes the problem only locally, per tree branch (see Eq. 1). Other approach reducing the time complexity is replacing the nearest neighbor search of k-means by *kd*-tree (Fig. 2(c)) or by random forest of *kd*-trees (Fig. 2(d)). The Philbin et al. (2007) called this approach as *Aproximate k-means*.

The quantization error after clustering procedure is expressed as a sum of distances of training samples to their nearest cluster as follows:

$$D = \frac{1}{N} \sum_{i=1}^{N} d(p_i, Q[p_i]) \qquad (1)$$

where N is the number of training samples Q is the nearest cluster center to the sample p_i and d is the distance function.

The preliminary experiments covered the influence of the vocabulary size and search type to quantization error, image retrieval precision and speed. The experiments for error and precision were evaluated for SURF local image features only and the speed-up factor for both, SURF and SIFT. The results in Table 2 were obtained using the Kentucky dataset (see Section 6 for more details). The speed-up factor of compared clustering strategies is expressed relatively to naive approach that is used as a baseline clustering strategy.

Table 2. Comparison of clustering strategies.

	clustering	vocabulary size			
	strategy	100	1k	10k	100k
error	naive	0.442	0.383	0.341	0.298
	kd-tree	0.443	0.392	0.357	0.321
precision	naive	52.4%	57.4%	60.0%	-
	kd-tree	52.5%	55.0%	55.1%	56.9%
speed-up	kd-tree	3.5	8.1	12.2	14.0

The visual vocabulary for the video synchronization system does not need to have high discriminative power; therefore the vocabulary of size 1k of visual words is utilized.

4. IMAGE SIGNATURE

In the presented framework, the *image signature* is a collection of weighted visual words representing the image content. This collection can be seen also as a vector of visual word frequencies. It degrades to a set-of-words when the weights represent only the word's presence. Otherwise, it is a bag-of-words. Figure 3 shows the process of describing the image content by an image signature.

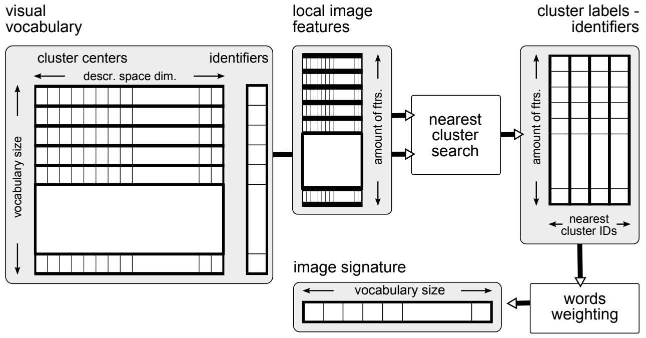

Fig. 3. Image signature extraction procedure.

Having visual vocabulary, for each descriptor of local features from the image, k words (nearest clusters) are found. The weight for each word is computed and used to increase the value of the image signature at the word's ID position. The image signature then can be defined as a histogram of occurred visual words. In recent works, several weighting schemes were introduced and evaluated.

In the pioneering work by Sivic and Zisserman (2006), standard weighting used in text retrieval is employed that is known as 'term frequency - inverse document frequency' - *tf-idf*. The *term frequency* reflects the entropy of a word with respect to each document unlike *inverse document frequency* down-weights words that appear often in the database. The resulting weight is then:

$$tf-idf(w) = tf(w).idf(w) \tag{2}$$
$$= \frac{|d(w)|}{|d|}.\log(\frac{|D|}{|D(w)})) \tag{3}$$

where d is a document (image signature), $|d|$ is a number of words in d and $|d(w)|$ is the number of occurrences of word w in d, D is a dataset of all documents and $D(w)$ is a set of documents containing the word w.

The later works (Philbin et al. (2008), Jiang et al. (2007)) reflected the fact that the quantization effect provides a very coarse approximation to the actual distance between two features - zero if assigned to the same visual word and infinite otherwise. The *soft-assignment* (soft-weighting) techniques assign a single descriptor to several visual words nearby in the descriptor space. Given the sorted list of *k nearest* visual words, an exponential function of the distance to the cluster center used in Philbin et al. (2008) is

$$w = \exp(-\frac{d^2}{2\sigma^2}) \tag{4}$$

where d is the distance from the cluster center to the descriptor point. In practice, σ is chosen so that substantial weight is only assigned to a small number of cells. The authors experimentally evaluated and suggest $k = 3; \sigma^2 = 6.250$. Another approach Jiang et al. (2007) is based on the rank of the assigned cluster:

$$w = \frac{1}{2^{i-1}} \tag{5}$$

where i is the ith nearest neighbor.

During the retrieval stage, documents are ranked by their *similarity*. One of the frequently used *similarity metrics* in text retrieval is normalized scalar product (cosine of angle) between the query vector $\mathbf{q}$ and all document vectors $\mathbf{d}$ in the database. The cosine similarity can be seen as a method of normalizing document length during comparison.

$$sim(\mathbf{q}, \mathbf{d}) = \frac{\mathbf{q}.\mathbf{d}}{|\mathbf{q}||\mathbf{d}|} \tag{6}$$

where . is dot product and || is the vector magnitude. The cosine similarity of two image signatures will range from 0 meaning independent images to 1 meaning exactly the same images, since the word weights (e.g. *tf-idf* weights) cannot be negative. For the experiments in this work, the standard *tf-idf* weighting scheme is used when visual words are weighted in bag-of-words.

5. TIME SHIFT DETECTION

The *time shift detection* task is formulated as follows. Having one reference and one (or more) query video stream, find the time shift between reference and query stream. Each incoming frame is translated to represent the frame by its image signature and then added into frame buffer. Frame buffer represents the video stream history. The query frame is then used to search for the most similar frame in reference buffer. The search step results in the most probable offset (time shift). Additional step in time shift detection procedure stores the detected offset for each time step into histogram of detected offset. The peak in histogram then represents the time shift. The scheme in Figure 4 shows the procedure steps.

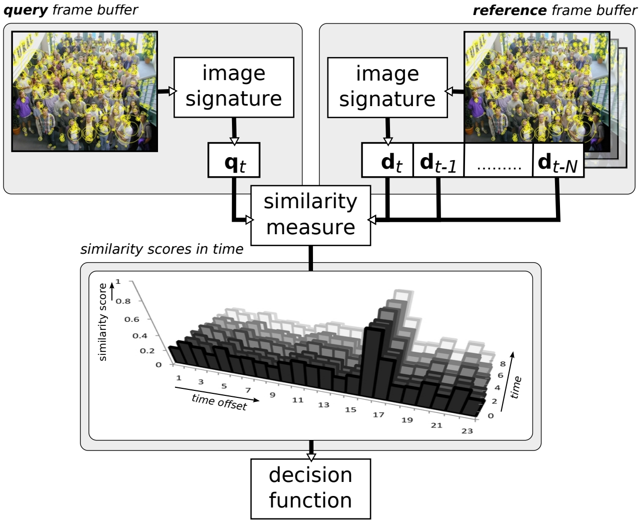

Fig. 4. Image signature extraction procedure.

The similarities between the query frame $\mathbf{q}$ and the frames from the buffer of the reference frames $\mathbf{d}$ in time t can be used to express the similarity score of the time-shift i as:

$$s(t, i) = sim(\mathbf{q}_t, \mathbf{d}_{t-i}), i \in \{0, \ldots, N\} \tag{7}$$

where N is the buffer size. The results of the function $s(t, i)$ in time t are similarity scores of time offsets i in interval 0 to N based on distances between the t^{th} query image and all images from reference buffer. Having the similarity score in time, is defined the decision function as:

$$D(t) = D(R(t), R(t-1), \ldots, R(t-\infty)) \tag{8}$$

where $R(t) = R(s(t, i))$ is the window decision function working on time interval $t, \ldots, t-N$. Both decision functions results in time offset value. The presented results in this work are based on window decision function $R(t)$ that makes an early fusion of similarity scores in time and then results with the offset with the highest fused similarity score. The function $O(t, i)$ represents the fusion as:

$$O(t, i) = w(t) \otimes s(t, i) \tag{9}$$

where $\otimes$ is the convolution operation and w is the weighting function. Two weighting functions were designed for the system - averaging (Eq. 10) and exponential function (Eq. 11):

$$w_{avg}(t) = \frac{1}{N} \tag{10}$$
$$w_{exp}(t) = 1 - \log_N t \tag{11}$$

The averaging function computes the similarity scores for the particular time offset i as the average over all previous similarity scores. The exponential function down-weights the older results. The decision result is then the time offset function shift(t) with the highest value of the function $O(t, i)$ as:

$$R(t) = \text{shift}(t) = \arg\max_i O(t, i) \tag{12}$$

For the purpose of the presented system, we formulated the final decision function $D(t)$ as the procedure computing the histogram $H(t, i)$ of $R(t)$. The location of the best histogram bin corresponds to the final decision of detected time shift.

$$D(t) = \text{rank}(H(t, i), 1). \tag{13}$$

where function $\text{rank}(L, r)$ ranks the list L and returns r^{th} element of the ranked list.

Along with the detected time shift, the *confidence level* $\kappa(t)$ of the decision is needed to approve or refuse the time shift decision. The confidence level computation is based on the ratio of the two best bins from the histogram of time shifts as:

$$\kappa(t) = \frac{\text{rank}(H(t, i), 1)}{\text{rank}(H(t, i), 2)}. \tag{14}$$

The confidence level represents the certainty of obtained statements about the detected time offset and is used as the main indicator of quality of the method.

6. EXPERIMENTS

Two different datasets were used for the presented experiments. The *Kentucky* dataset was used to analyze the characteristics of various parameters of feature extraction and visual vocabulary building. The *TRECVID 2009 data BBC subset* was used to evaluate the time shift detection method.

The *Kentucky* dataset was created as a recognition benchmark at Kentucky University by Nister and Stewenius (2006). The set consists of 2550 groups of 4 images each, that is 10200 images in total. The size of the images is 640x480 pixels. This dataset was used to compare the performance of different image feature extraction methods, search types of k-means algorithm and influence of the vocabulary size to its performace.

The *TRECVID 2009 data BBC subset* (Smeaton et al. (2006)) contains 77 video files. The size of video frames is 352x288 pixels. The video streams are coded according to MPEG1 with 25 fps. The TRECVID video data contains a variety of challenging disruption itself, especially for the time video synchronization task.

The TRECVID videos begin with an initialization shot (cca 40-60 seconds) with color initialization stripes where very few local features are detectable. Furthermore, many of the shots are static or almost static, so the frame comparison does not favor any frame and the detection does not decide for any particular time shift.

The query video streams were created artificially - the real reference video was distorted and delayed. In this work, we focused on fundamental types of video distortions:

- uniform scale transformation,
- partial occlusion of the video by static banner,
- white noise.

The examples of the simulated distortions are shown in Figure 5.

All experiments used the visual vocabulary built on a different dataset than the testing one. It simulates the real situation of on-line video processing, where the system might work on different data then was trained on. The visual vocabulary was built upon the Kentucky dataset and all 10200 images were used to train the vocabulary.

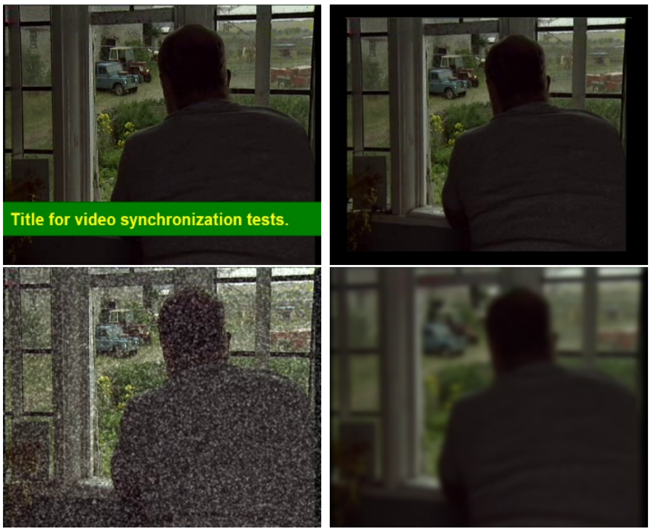

Fig. 5. Examples of video distortions (top - overlaid banner, 90% resize; bottom - 30% noise, blur with $\sigma = 3.0$).

7. RESULTS

When detecting the time shift between two similar video streams based on image content analysis, the most crucial is the content itself. When the image entropy is low or constant over the time, it is not possible to neither detect any features and their changes nor synchronize the video streams. Each experimental run was done on all 77 video files. The result of the processing of each video file is the detected time shift and confidence level $\kappa(t)$ for each video frame. Due to the different length of the video initialization shot at the beginning of each video, the results are aligned in time domain. The presented results are then averaged over all of the runs for each particular experiment.

Two ground-truth time delays were used in experiments: 25 and 250 frames (*d25* and *d250*). The experiment results are presented using graphs where y-axis is the confidence level $\kappa(t)$ of the detected time-shift and x-axis is the time from the beginning of the video sequence in frames. When the confidence level is not defined, the method wrongly detects the time shift or fails entirely. The positive detection of the time shift is represented by the positive confidence level value. In general, the graphs shows the required time to positively detect the time shift.

7.1 Window Decision Function

The first experiments were focused on comparison of two weighting functions: average and exponential (Eq. 10 resp. Eq. 11). The functions are used for the early fusion of similarity scores in the window decision function (Eq. 8. When undistorted data was used for the test, no significant difference appear (see Figure 6).

The tests with query video resized to 90% of the original size revealed that when query video is delayed by 25 frames and also by 250 frames, the exponential weighting function outperforms the averaging function.

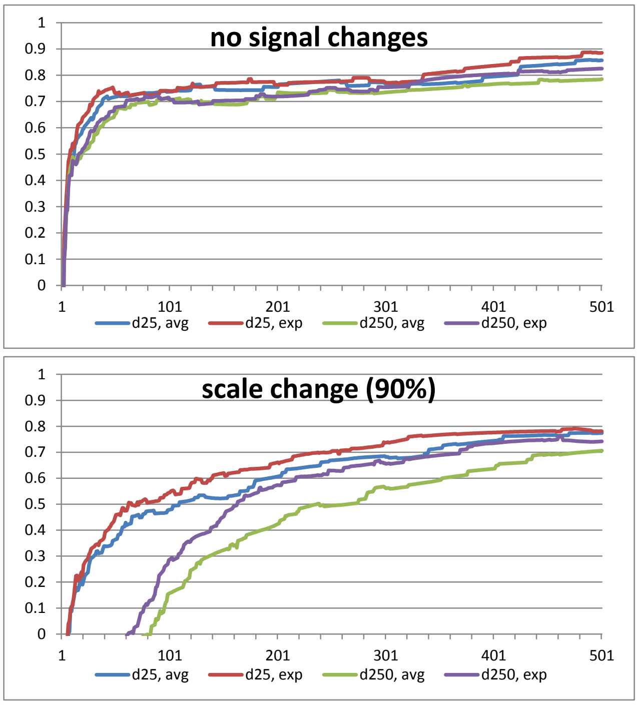

Fig. 6. Comparison of window decision functions (top - without distortion, bottom - resize to 90%).

7.2 Time Shift Detection

The robustness and precision of the time shift detection algorithm was evaluated on several video distortions and transformations that are likely to occur in real applications. Such cases are occlusion, noise, blur, and scale change.

The robustness to partial occlusion of the video frames was tested using overlaid banner. The graph in Figure 7 shows that the precision of the detector is affected by partial occlusion for longer delays. The robustness to the noise was tested with the noise level in between 0%-30%. Figure 8 shows how the detector precision decreases with increasing level of noise. The experiments with blurring have proven the expected properties about the used feature extractor. The SURF features are sensitive to blobs so their repeatability decreases very slowly with increasing amount of blur. Because the system's properties directly depend on the robustness of the feature detector, the algorithm performance is only slightly affected by smooth level as might be seen in Figure 9.

The computational costs of the algorithm depend on amount of extracted features. Data used in evaluation contain cca 200 features per frame. The algorithm was able to process 13 frames per second on the desktop PC Intel Core Duo 2.4GHz, 3.5GB.

8. CONCLUSIONS

The objective of the presented work was to design and evaluate the real-time on-line system for visual-content-based video analysis. The solution is based on the visual vocabulary and image signatures. Several types of video distortions were addressed in the experiments. The results of evaluated experiments show that the proposed solution is usable for the on-line real-time video analysis and video content comparison. The solution can be used as the base for the other methods; for example, for dealing with geometrical transformation between video streams. The knowledge of the geometrical transformation might be use-

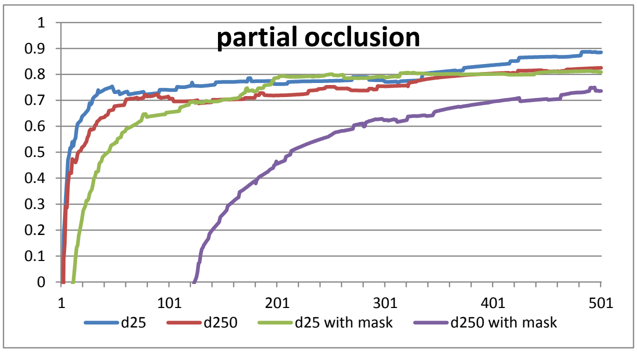

Fig. 7. Partial occlusion by a inserted banner.

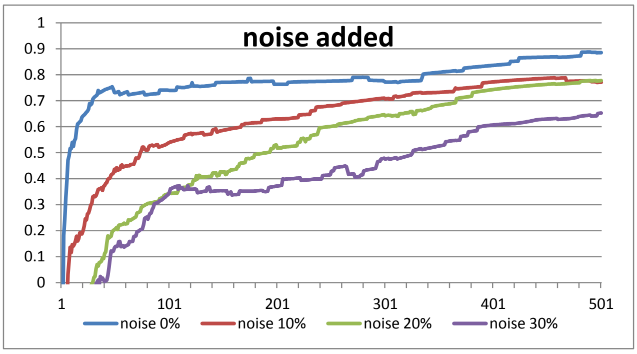

Fig. 8. Added noise of levels between 0% and 30%.

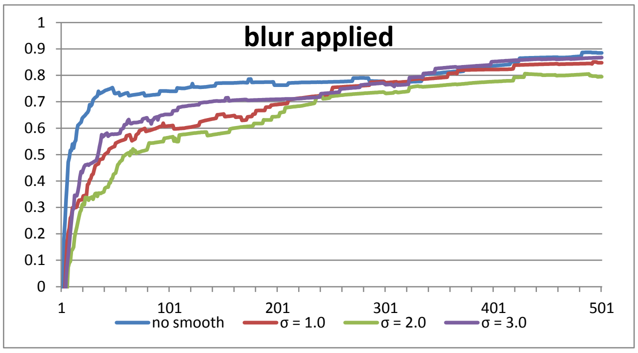

Fig. 9. Gaussian blur applied with various σ.

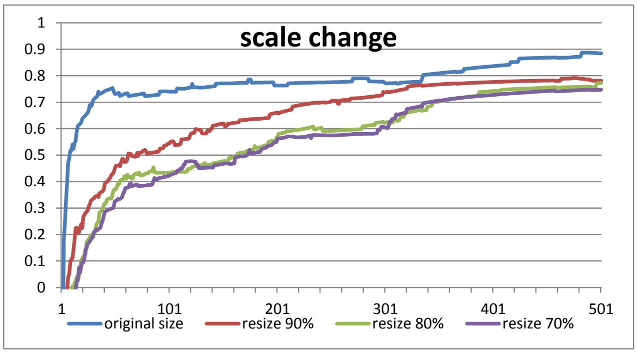

Fig. 10. Uniform rescale transformation.

ful when non-uniform geometrical distortion is expected and the automatic video quality control task analyzing noise ratio, contrast changes, color bias, etc. is required.

ACKNOWLEDGEMENTS

This work has been supported by the Ministry of Education, Youth and Sports of the Czech Republic under the research program LC-06008 (Center for Computer Graphics) and by the research project "Security-Oriented Research in Informational Technology" CEZMŠMT.

REFERENCES

Bay, H., Tuytelaars, T., and Gool, L.V. (2006). Surf: Speeded up robust features. In *In ECCV*, 404–417.

Chum, O., Perdoch, M., and Matas, J. (2009). Geometric min-hashing: Finding a (thick) needle in a haystack. In *Conf. on Computer Vision and Pattern Recognition.*

Chum, O., Philbin, J., Isard, M., and Zisserman, A. (2007). Scalable near identical image and shot detection. In *CIVR '07: Proceedings of the 6th ACM international conference on Image and video retrieval*, 549–556. ACM, New York, NY, USA.

Chum, O., Philbin, J., and Zisserman, A. (2008). Near duplicate image detection: min-hash and tf-idf weighting. In *Proceedings of the British Machine Vision Conference.*

Jiang, Y.G., Ngo, C.W., and Yang, J. (2007). Towards optimal bag-of-features for object categorization and semantic video retrieval. In *CIVR '07: Proceedings of the 6th ACM international conference on Image and video retrieval*, 494–501. ACM, New York, NY, USA.

Lindeberg, T. (1998). Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30, 79–116.

Lowe, D.G. (2004). Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2), 91–110.

Matas, J., Chum, O., Urban, M., and Pajdla, T. (2002). Robust wide baseline stereo from maximally stable extremal regions. In P.L. Rosin and D. Marshall (eds.), *Proceedings of the British Machine Vision Conference*, volume 1, 384–393. BMVA, London, UK.

Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., and Gool, L.V. (2005). A comparison of affine region detectors. *Int. J. Comput. Vision*, 65(1-2), 43–72.

Mikolajczyk, K. and Schmid, C. (2004). Scale & affine invariant interest point detectors. *Int. J. Comput. Vision*, 60(1), 63–86.

Mikolajczyk, K. and Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(10), 1615–1630.

Nister, D. and Stewenius, H. (2006). Scalable recognition with a vocabulary tree. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2161–2168. IEEE Computer Society, Washington, DC, USA.

Philbin, J., Chum, O., Isard, M., Sivic, J., and Zisserman, A. (2007). Object retrieval with large vocabularies and fast spatial matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.*

Philbin, J., Chum, O., Isard, M., Sivic, J., and Zisserman, A. (2008). Lost in quantization: Improving particular object retrieval in large scale image databases. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.*

Rosten, E. and Drummond, T. (2006). Machine learning for high-speed corner detection. In *In European Conference on Computer Vision*, 430–443.

Sivic, J. and Zisserman, A. (2006). Video Google: Efficient visual search of videos. In J. Ponce, M. Hebert, C. Schmid, and A. Zisserman (eds.), *Toward Category-Level Object Recognition*, volume 4170 of *LNCS*, 127–144. Springer.

Smeaton, A.F., Over, P., and Kraaij, W. (2006). Evaluation campaigns and trecvid. In *MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, 321–330. ACM Press, New York, NY, USA.