Dissimilarity Detection of Two Video Sequences

Lukáš Klicnar[*]
Faculty of Information Technology,
Brno University of Technology

Vítězslav Beran[†]
Faculty of Information Technology,
Brno University of Technology

Pavel Zemčík[‡]
Faculty of Information Technology,
Brno University of Technology

Abstract

This paper presents an approach for detection of differences between two visually identical video sequences. The video processing task for detection of short- and long-term changes between two video sequences is defined in detail. The algorithm comparing two video sequences (reference and query) is introduced together with definition of particular situations that the algorithm must be able to detect: re-written parts, removals or injected parts. The image processing methods are selected to be robust to several practical distortions that might appear in defined task. The appropriate computer-vision methods are presented and discussed, then proposed method and experiments are introduced and evaluated on manually generated dataset. Main focus of this work is on comparison of two different approaches for keyframe extraction: The first, more robust one is based on local features tracking, which we attempt to replace with computationally much less-expensive global descriptor approach with preservation of approximately the same video sequence dissimilarities detection success rate. Results of the different approaches are presented and discussed.

CR Categories: I.4.6 [Image Processing and Computer Vision]: Segmentation—Edge and feature detection; I.4.7 [Image Processing and Computer Vision]: Feature Measurement—Feature representation; I.4.9 [Image Processing and Computer Vision]: Applications; K.6.1 [Computing Methodologies]: Vision and Scene Understanding—Video analysis;

Keywords: Video comparison, Dissimilarity, Histogram, Motion segmentation, Similarity matrix, Temporal analysis, Keyframe detection, Video segmentation

1 Introduction

The main task of the video comparison systems is to detect and validate the differences between two visually *almost* identical video sequences. In some cases, even when two video sequences are declared as the identical, small differences might appear and manual detection and validation of such video parts may become extremely time consuming and unbearable. The example of video-pair disruption with dissimilarity types is showed on Figure 1.

When comparing the similarity between two video sequences (reference and query), we define three types of dissimilarity that might occur:

- Rewriting – part of the query video is rewriten by different visual content than in reference video and the length of query video part is the same as the reference video part
- Injection – part of the query video is new - added to original (reference) content, so the query video part is longer than the reference video part
- Removal – part of the query video is removed, so the query video part is shorter than the reference video part

Presented research is focused on visual content, so audio is omitted. The visual part of the video is sequence of consecutive images, video frames, and one way of evaluating similarity between two videos (or its parts) is to compare the similarities between video frames and compute statistical analysis. In our work, we represent the video sequence as the set of video-parts. Each video-part is represented by its temporal information (begin and end) and also by one or more key-frames. The video-part key-frames are in some sense interesting video frames and are usually represented by image descriptors. One of the research goals is to analyse the influence of the density of the video-part key-frames to stability and precision of the entire video-pair comparison approach.

2 Image and Temporal Analysis

The similarity between two images (video frames) can be in general evaluated using two types of visual content description: *global* and *local*. The *global* approach of image description extracts image features from the entire image and utilizes statistics for their representation. Global approach might be very computation cost effective but is usually not very robust to geometrical distortions as no spatial information is taken into account.

2.1 Global image features

We represents the image content by colour histograms combined with a spatial pyramid over the image to jointly encode global and local information [Chum et al. 2007]. We use several colour models (grey-scale, HSV, IO_1O_2). The IO_1O_2 colour model, known as the opponent colour model [J. Geusebroek and Geerts 2001], is partially colour normalized and simple to compute. The spatial pyramid is arranged so that low number of bytes of data is describing each pyramid level. These are appended to create the final feature vector. On descending to the next level in the pyramid, the number of segments the histograms are taken over increases four-fold.

Besides colour information, we compute also histograms of image gradients to represent the image intensity changes. We improved the stability of the existing approach by increasing of the overlap of subdivision regions and also by weighting the values withing each region [Sailer 2012].

[*] e-mail:iklicnar@fit.vutbr.cz
[†] e-mail:beranv@fit.vutbr.cz
[‡] e-mail:zemcik@fit.vutbr.cz

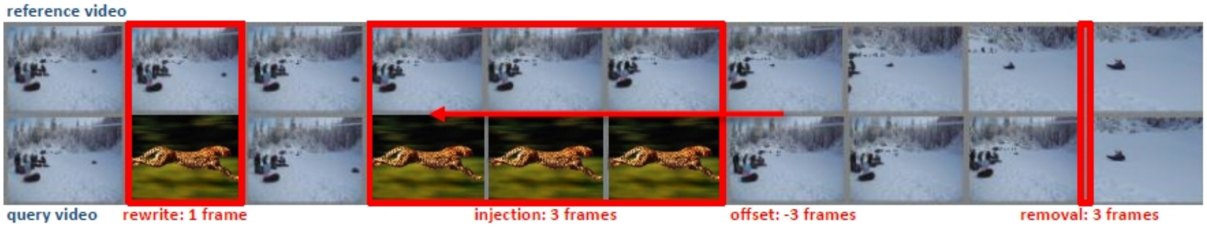

Figure 1: Examples of video-pair disruption.

2.2 Local image features

The *local* approach extracts local image features such as corners or blobs and represents the image content as the set of such local features and their descriptors. The local approach is more robust to geometrical distortions but might have poor results with noisy data and is computationally more expensive. From the numerous existing methods for local image analysis and description, we choose following approaches that differs in computational efficiency, stability and precision. One of the widely used approach is the SIFT detector proposed by Lowe [Lowe 2004] for its high spatial and scale precision of detected local features and also because it includes also very robust method for feature description. Next favourite method that accelerates the Hessian-based approach is known as SURF detector [Bay et al. 2006]. The approach introduced by Rosten and Drummond known as FAST corners [Rosten and Drummond 2006] employs machine learning to construct corner detector that outperforms all know approaches in the speed.

2.3 Temporal analysis

The temporal analysis is usually used in video processing to analyse the geometrical changes in consecutive video frames. According to application, the analysis serves e.g. to detect cuts in video sequences or find the visually most representative candidates of video-parts. We have selected two distinct approaches. First approach computes the differences between several adjacent video frames represented by global image features using Euclidean distance for metric features. The differences are evaluated over flowing window.

Other approach is based on tracking of local image features over the close video frames. The approach is motivated by work of Sivic et al. [Sivic et al. 2006] and further developed by Klicnar and Beran [Klicnar and Beran 2012] for computationally efficient video segmentation. The existing method was adapted to a higher computational speed and on-line processing. The proposed approach is based on sparse local image features and the KLT tracker for feature trajectory computation. A RANSAC-based method is used for initial motion segmentation, resulting motion groups are partitioned by a spatial-proximity constraints. The correspondence of motion groups across frames is solved by one-frame label propagation in forward and backward directions. The method results in stable trajectory bundles that represents distinctive image regions.

3 Video Sequence Comparison

This section describes basic principles of the proposed system for video dissimilarity detection. Its design consists of several independent consecutive steps (see also the block diagram on Figure 2):

1. Preprocessing of both, reference and query video sequences
2. Computation of similarity matrix
3. Detection of corresponding segments in both videos

3.1 Key-frame extraction

The goal of the keyframe extraction is to describe to whole video sequence by a set of keyframes, which represents the individual video parts. Every part is described by several keyframes – the start frame, the end frame and possibly several frames inside. Actually, it is possible not to detect segment boundaries, but create keyframes directly by using every Nth frame of the video sequence. The precision of detection of their boundaries is then dependent on the period N and very short segments cannot be detected – they may be simply passed unnoticed if they lie between two keyframes. This is the reason why every frame must be inspected and boundaries of the individual video parts must be found.

Frames are processed sequentially, a significant change in the video is considered as a boundary between two segments (keyframes are created). In addition, keyframes are also created periodically every N frames (approx. every 1 second), which improves the response and stability of long segments in their detection process. The last thing is that how segment boundaries are detected. In this work, we compare two approaches - by observing the global features or by utilizing the local features and their development in time.

3.2 Similarity matrix

Similarity of the extracted keyframes is described by a so called similarity matrix S. Its rows represent keyframes from a reference sequence V_R, while columns represent keyframes from a tested sequence V_T. Every value in this matrix represents the dissimilarity of keyframes $V_R(r)$ and $V_T(t)$, in our case it is the distance between descriptors of both keyframes $S(r,t) = d(V_R(r),V_T(t))$. Similar parts of both video sequences forms evident diagonal line segments in the matrix.

3.3 Dissimilarities detection

As the corresponding parts of both sequences appears as line segments (with high frame-to-frame similarity) on the similarity matrix, video matching problem can be reduced to searching for these lines. We suppose several assumptions: First, frame rates of both video sequences doesn't differ, so the lines are nearly diagonal. Also the order of the scenes is preserved, only some of them are removed, replaced, or there is some other content inserted

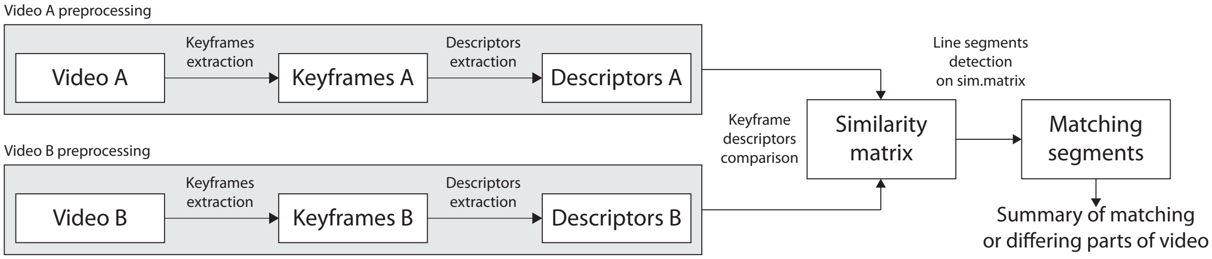

Figure 2: Basic block diagram of the proposed system for video sequence comparison.

– disconnected line segments on the similarity matrix can only be shifted to the right and/or downwards from the previous segment.

We developed an recursive algorithm for segments detection based on the *divide and conquer* technique. First, we need to define a continuous segment, which is a line of neighbouring points on thresholded similarity matrix S_T that goes from $A = (a_x, a_y)$ to $B = (b_x, b_y)$. From a given starting point $A_1 = (a^1_x, a^1_y)$, the segment can be gradually constructed by following the diagonal or by doing a vertical/horizontal step. The segment is constructed until $A_{n+1} = A_n$. The length of the segment is defined as Euclidean distance: $d(A,B) = \sqrt{(b_x - a_x)^2 + (b_y - a_y)^2}$.

The algorithm works as follows: At first, the whole matrix is searched for the point, from which the longest continuous segment can be constructed. If the length of this segment $d(A,B) > d_{min}$, it is accepted and the matrix is subsequently divided into these three areas (as illustrated in fig. 3):

1. Rectangle from upper left corner to start of the segment
2. Rectangle from end of the segment to lower right corner
3. Remaining areas

Regions 1 and 2 are then processed recursively. In each of them, the dominant continuous segment is detected and if fulfils the minimal length criterion, it is accepted, the region is subdivided and the recursion is repeated. We suppose that longer continuous segments are less probable to be formed by noise, so the extraction of the most dominant lines as first improves the robustness. Figure 3 shows how this algorithm searches for the new line segments in the rectangular areas between the already found ones. White colour represents areas of the matrix for the next segment detection, while the grey colour marks already inspected or rejected regions, where no further segments can be detected.

4 Results

The main specifics is that we need actually two sequences – one is the *reference*, which is the sequence without modifications. The modified is a *query* one. We created a small dataset that we used for algorithm design and its evaluation. It consists of a 50 reference sequences from TRECVID dataset, query sequences were made by randomly chosing 2 of them (base sequence and a one used for addition). 10 sequences were made containing each dissimilarity type only, that results a total of 30 sequences.

4.1 Detection performance

We involved standard metrics for evaluation, such as Miss rate (MR), False alarm rate (FAR) and F-measure (Fm), which indicates the accuracy of detected dissimilarity length and position. Great emphasis is put on necessary detection of all sequence changes, so the detector was set to a minimal miss rate at a costs of increased number of false alarms. Results are shown on Table 1, global and local approach performs nearly the same. The relatively low F-measure is caused partially by oversegmentation and also the method failure on sequences, where all frames are nearly similar.

	Global			**Local**		
	MR	**FAR**	**Fm**	**MR**	**FAR**	**Fm**
Injection	0.00	0.57	0.68	0.00	0.58	0.77
Removal	0.00	0.64	0.60	0.00	0.64	0.59
Rewriting	0.00	0.63	0.80	0.00	0.62	0.89

Table 1: Results of the dissimilarities extraction.

4.2 Computational speed

Performance was measured on system with Intel Core 2 Duo T7100@1.80Ghz processor and 4GB of RAM, used sequence contains a total of 28358 frames with resolution of 624x352px. Achieved results in Table 2 show that global approach for keyframes extraction is much faster than the local, tracking-based one. Considering that both of them give similar results on the test data, we claim the histogram-based approach sufficient.

	Global	**Local**
Comp. time (relative)	1.0x	13.0x
Framerate	172.4 frames/s	13.3 frames/s

Table 2: Performance of the whole video comparison process.

5 Conclusion

The presented work describes video processing task of detection of short- and long-term changes between two video sequences. Two video sequences might be declared as the identical, but small differences might appear. Results show that both local and global approach gives approximately the same results on the described dataset. The main difference is that computation of the global feature-based keyframe extraction is approximately 13 times faster. This means that faster but less robust global approach can be used for keyframe extraction in this application without insulting the results significantly.

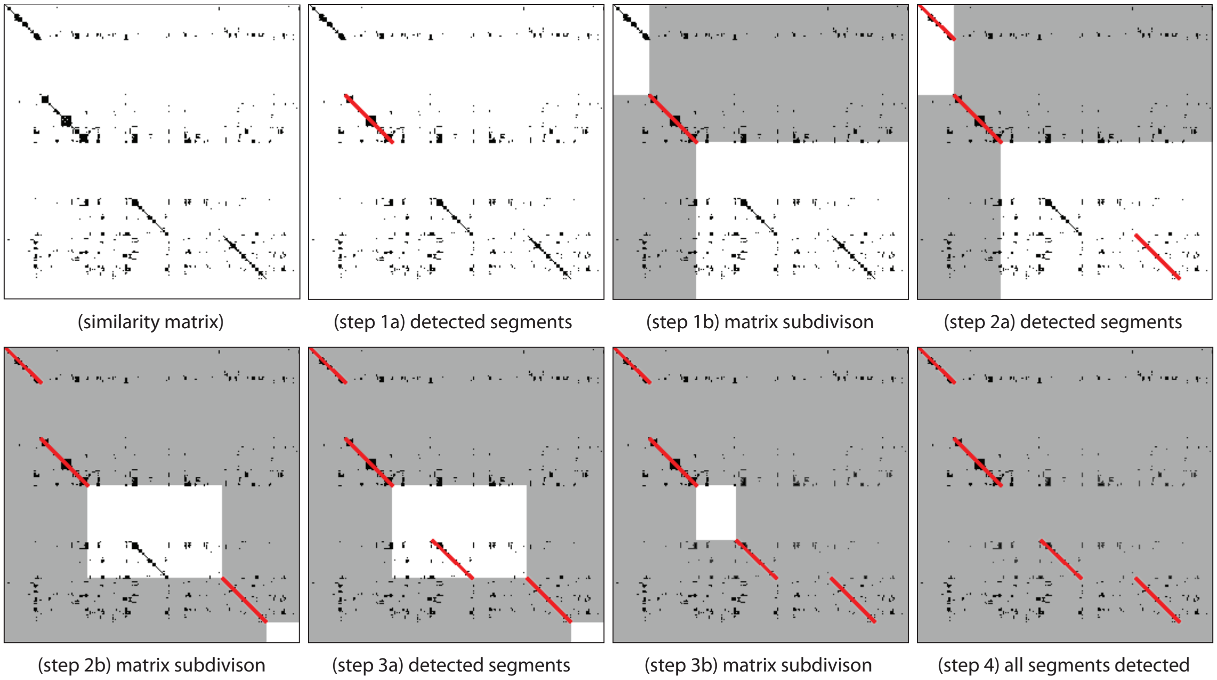
(similarity matrix)
(step 1a) detected segments
(step 1b) matrix subdivison
(step 2a) detected segments
(step 2b) matrix subdivison
(step 3a) detected segments
(step 3b) matrix subdivison
(step 4) all segments detected

Figure 3: Example of dominant segments detection.

Acknowledgment

This work has been supported by the research project "Security-Oriented Research in Information Technology" CEZMSMT, MSM0021630528, "Tools and Methods for Video and Image Processing for the Fight against Terrorism", VG20102015006 and European Union, 7th Framework Programme, Artemis JU grant 295371 - CRAFTERS.

References

BAY, H., TUYTELAARS, T., AND GOOL, L. V. 2006. Surf: Speeded up robust features. In *In ECCV*, 404–417.

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*, ACM, New York, NY, USA, 549–556.

J. GEUSEBROEK, R. VAN DEN BOOMGAARD, A. S., AND GEERTS, H. 2001. Color invariance. *PAMI 23*, 12, 1338–1350.

JURIE, F., AND SCHMID, C. 2004. Scale-invariant shape features for recognition of object categories. *Conference on Computer Vision and Pattern Recognition 2*, 90–96.

KADIR, T., AND BRADY, M. 2001. Scale, saliency and image description. *International Journal of Computer Vision 45*, 2, 83–105.

KLICNAR, L., AND BERAN, V. 2012. Robust motion segmentation for on-line application. In *Proceedings of WSCG'12*, University of West Bohemia in Pilsen, 20-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, 1–6.

LOWE, D. G. 2004. Distinctive image features from scale-invariant keypoints. *International Journal of Computer 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 *Proceedings of the British Machine Vision Conference*, BMVA, London, UK, P. L. Rosin and D. Marshall, Eds., vol. 1, 384–393.

MIKOLAJCZYK, K., AND SCHMID, C. 2004. Scale & affine invariant interest point detectors. *International Journal of Computer Vision 60*, 1, 63–86.

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. *International Journal of Computer Vision 65*, 1-2, 43–72.

ROSTEN, E., AND DRUMMOND, T. 2006. Machine learning for high-speed corner detection. In *In European Conference on Computer Vision*, 430–443.

SAILER, Z., 2012. Image retrieval based on color histograms.

SIVIC, J., AND ZISSERMAN, A. 2006. Video Google: Efficient visual search of videos. In *Toward Category-Level Object Recognition*, J. Ponce, M. Hebert, C. Schmid, and A. Zisserman, Eds., vol. 4170 of *LNCS*. Springer, 127–144.

SIVIC, J., SCHAFFALITZKY, F., AND ZISSERMAN, A. 2006. Object level grouping for video shots. *International Journal of Computer Vision 67*, 2, 189–210.

TUYTELAARS, T., AND GOOL, L. V. 2004. Matching widely separated views based on affine invariant regions. *Int. J. Comput. Vision 59*, 1, 61–85.