QUALITY ASSURANCE IN LARGE COLLECTIONS OF VIDEO SEQUENCES

Lukáš Polok, Lukáš Klicnar, Vítězslav Beran, Pavel Smrž, Pavel Zemčík

{ipolok,iklicnar,beran,smrz,zemcik}@fit.vutbr.cz
Brno University of Technology, Faculty of Information Technology.
Bozetechova 2, 612 66 Brno, Czech Republic

ABSTRACT

In the modern digital cinema production, extremely large volumes (in order of 10s of TB) of footage data are captured every day. The process of cataloging and reviewing such footage is nowadays largely manual and time consuming process. In our work, we aim at technical quality aspects, such as correct exposure, color compatibility of adjacent shots, and focusing. The main goal is to assist the reviewing process by providing shot quality meta-data and possibly ordering or even culling significant portion of the data from the review, as better quality shot of the same scene exists. However, in order to meaningfully compare technical quality, temporal shot synchronization needs to be performed first.

We propose a fast and robust method for time synchronization of video sequences, capturing similar scenes, which arise naturally in digital cinema production. The method is tested on an extensive library of sequences and its performance is evaluated. We further present a preview of application of the proposed method to detecting focusing errors.

Index Terms— large video databases, automatic quality assurance, time synchronization

1. INTRODUCTION AND RELATED WORK

Video synchronization is an important task that can benefit many contemporary video processing and quality assessment applications. In quality assessment of film footage, video synchronization may play an important role as well.

Results of functions used as quality metrics of video sequences and/or its individual frames are strongly dependent on the content of the frames. Therefore, comparison of quality metric results of two different pieces of footage with different content does not often really tell much about the quality but rather about the content. For the same reason, comparing quality metrics of two pieces of footage with the same content but with different timing of object and/or event occurrences often favors such piece in which some object occurs longer rather the piece of footage which is better. The way to overcome this problem is in synchronization of the pieces of footage so that quality metrics can reflect the footage timing and so that the comparison of quality metrics are applied on pieces of footage with comparable content.

The video synchronization methods have been widely researched in the past and quite many approaches have been attempted. For example, Caspi and Irani [1, 2] present a method for sequence alignment in both time and space, assuming static calibrated cameras related by a homography and also a constant time displacement. A similar method is also proposed by Tresadern and Reid [3]. Rao et al. [4] propose a more general method, which can cope with more general scenes and non-constant time delays. Tuytelaars and Van Gool [5] further extend to allow fully general scenes and independently moving cameras. These works employ homography or epipolar geometry as constraints in 3D space. Sivic and Zisserman [6] applied methods derived from text retrieval. Whitehead et al. [7] used approach based on consensus of feature motion in multiple videos. Further work brought variety of new approaches. For example, Ushizaki et. al. [8] introduced method that uses co-occurrence of matching parts of image, Chum et al. [9] introduced similar method even invariant to scale. Others, for example Yan and Pollefeys [10], Le et al. [11], or Reznicek et al. [12] attempted also feature based methods for detection of similarity of actions in video, based on space-time features with bag of words (BOW) and machine learning evaluation.

This work is partly based on the work of Beran et al. [13] and Klicnar et al. [14] who applied similarity metrics of the individual frames and processing of mutual similarities of large sets of frames. In their approach, each frame is processed by SURF feature extractor, the features are clustered using a visual vocabulary and BOWs are assembled. The BOWs obtained from different frames are then used as a frame signature and is it assumed that the closer the signatures the closer are also the frames.

The rest of this paper is organized as follows: the next chapter describes the proposed approach and its mathemati-

The research leading to these results has received funding from the European Union, 7^{th} Framework Programme grants 316564-IMPART and the IT4Innovations Centre of Excellence, grant n. CZ.1.05/1.1.00/02.0070, supported by Operational Programme Research and Development for Innovations funded by Structural Funds of the European Union and the state budget of the Czech Republic. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the GPUs used for this research.

cal formulation, chapter 3 describes the implementation used in the evaluations, chapter 4 describes the datasets used and presents the results achieved. Finally, 5 summarizes the paper and describes possible future directions.

2. PROPOSED APPROACH

The proposed approach starts by extracting frame descriptors for the two video sequences being matched. Note that this step can be highly parallelized, and can be done for arbitrary number of video sequences, which can be later synchronized arbitrarily between each other. There is a wide variety of suitable descriptors available in the literature, including color and gradient histograms, SIFT, SURF, and bag of words approaches, see [13] for a comprehensive survey. In our implementation, we use a simple tiled color histograms, in conjunction with IO_1O_2 color transform [15]. It was empirically determined, that it is sufficient to calculate a single frame descriptor only every eight frames, making a trade-off between processing speed and accuracy.

The subsequent stages of the algorithm take place on pairs of feature vectors, corresponding to a pair of video sequences x, y to be synchronized. For each pair of the feature vectors F_x, F_y we calculate their cross-covariance matrix:

$$\gamma_{xy} = \frac{(F_x - F_y)^T (F_x - F_y)}{||(F_x - F_y)^T (F_x - F_y)||} \tag{1}$$

We can directly use this matrix to calculate the synchronization of the two shots using dynamic programming. We will drop the sequence indices x, y in the following equation, in order to reduce the notation clutter. Let us define a cost function $q(i,j)$ which denotes the cost of frames i (in sequence x) and j (in sequence y) being associated with each other:

$$\Gamma_{i,j} = -log(\gamma_{i,j}) \tag{2}$$

$$q(i,j) = \begin{cases} 0 & \text{if } i = j = 1 \\ min(q(i-1,j-1) + \Gamma_{i-1,j-1}, & \\ \quad q(i,j-1) + \Gamma_{i,j-1}, & \\ \quad q(i-1,j) + \Gamma_{i,j}) & \text{otherwise} \end{cases} \tag{3}$$

We then declare the minimum-cost sequence of frame associations, which connects the first frames $q(1,1)$ with the last frames $q(\texttt{rows}(F_x), \texttt{rows}(F_y))$ of both respective shots, and is evaluated using dynamic programming. The result of this operation is depicted in Fig. 1 a). We can notice a slight discrepancy of the calculated frame association curve in the bottom right part of the image. This is likely due to self-similarity of one of the shots, which can be seen as prominent block-like patterns, spanning entire rows or columns of the γ_{xy} matrix.

This can be partially reduced by taking this self-similarity into account. This quantity is easily evaluated as auto-covariance γ_{xx} and γ_{yy} (calculated using (1)), as follows:

$$\hat{\gamma}_{xy} = \gamma_{xx}^{-1} \gamma_{xy} \gamma_{yy}^{-1} \tag{4}$$

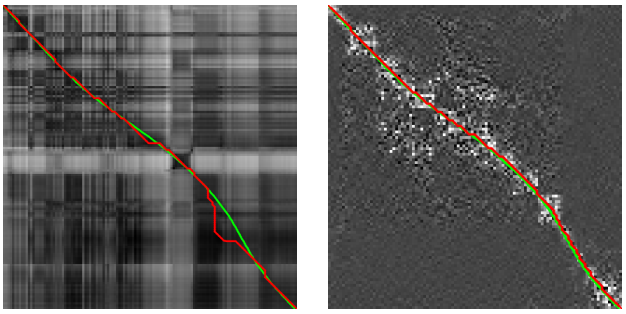

Fig. 1. The cross-covariance matrices and time synchronization curves for a sample from the Hollywood dataset [16], ground truth in green, the obtained results in red. a) results for simple γ (left), b) results with auto-correlations (right). Best viewed in color.

This can lead to some numerical issues if one of γ_{xx} or γ_{yy} is near to singular, which would happen for a sequence containing no motion. In practice, however, even the slight noise in the video is sufficient to keep the computation stable. In any case, the singularity is easily detected by calculating determinants of the two matrices and comparing it to a threshold. Note that the determinant calculation is essentially free, as we have already calculated LU decomposition of both matrices in order to perform the inverse. If a singularity is detected, the method falls back to using the original γ_{xy}. An example result using this method is depicted at Fig. 1 b).

3. IMPLEMENTATION

The implementation was written in C++. For the calculations of covariance matrices and their inverse, an OpenCL GPU implementation was used. For calculating (1), a similar method to the one described in [17] was used: the whole operation can be seen as many batched operations with small vectors (the individual descriptors). This is actually beneficial from the memory traffic point of view, as the individual vectors can be better cached in the shared memory, and most of the operations are performed in registers, rather than forming intermediate matrices, as would happen in the dense case. The matrix is copied back to the CPU side for the dynamic programming algorithm. All the times of memory transfers between CPU and GPU are included in the timing evaluations.

4. EXPERIMENTAL EVALUATION

The proposed methods were tested on a subset of TRECVID 2008 dataset [18], especially the BBC rushes. Additional evaluations were made using sequences from the Hollywood dataset, which were artificially modified to provide ground truth for the evaluation.

The Hollywood dataset [16] only contains one version of each shot (as opposed to the BBC rushes), and so several

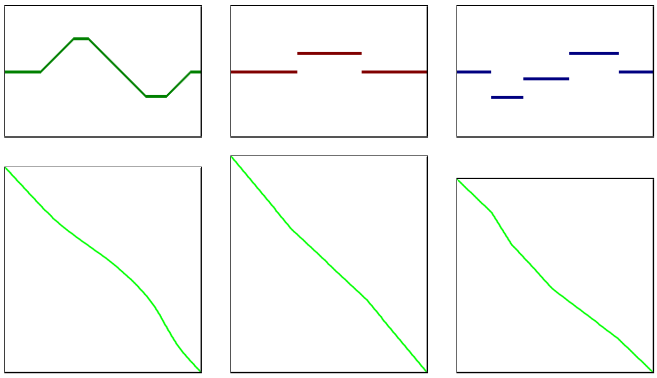

Fig. 2. Three different time skew profiles for the ground truth data, used in the evaluation. Top row: relative speed profiles; the horizontal axis is time, the vertical axis is playback speed. Bottom row: corresponding relative time profiles; the horizontal axis is time of the original shot, vertical axis is the time of the perturbed shot

different versions of each shot were created by artificially changing the playback speed of the video, to provide annotated ground truth data. Three different curves were used, as depicted in Fig. 2. These curves give the relative playback speed, and their normalized cumulative sum gives relative playback time, which is for all three cases a smooth curve.

All the tests were performed on a computer with NVIDIA Tesla K40 (12 GB RAM), a pair of AMD Opteron 2360 SE CPUs running at 2.5 GHz and 16 GB of RAM. The program was compiled as x64, and CUDA was set to use 64-bit pointers. The latest GPU drivers (version 344.48) were used. CUDA implementations were linked against CUDA 6.5 SDK libraries. During the tests, the computer was not running any time-consuming processes in the background. Each test was run at least ten times until cumulative time of at least 5 seconds was reached, and the average time was calculated in order to avoid measurement errors. Explicit CPU - GPU synchronization was always performed, using `cuCtxSynchronize()`. ECC memory protection was disabled on the Tesla GPU.

Precision of the calculated time synchronization was evaluated, using the mean square error of the ground truth and the calculated frame time value, summed over the frames which had a frame descriptor associated with them. In addition to the mean, a maximum error (the distance of the worst matched corresponding frames) is also reported. The results of this evaluation are showin in Fig. 3. There were no noticeable precision differences between the shots from the TRECVID and from the Hollywood dataset.

In case of the algorithm using the auto-covariances, all the failure cases were detected using a threshold of 0.01 relative to the norm of the matrix. An illustration of one such failure case can be seen in Fig. 6.

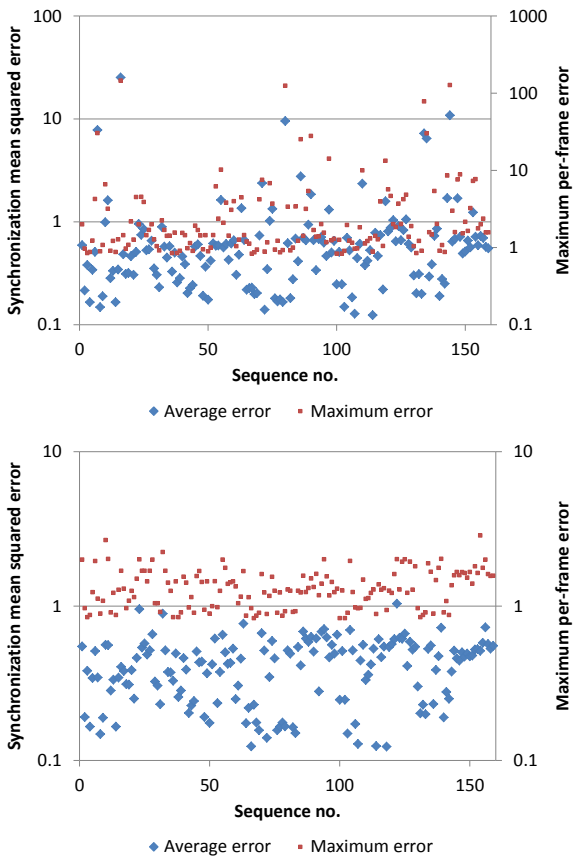

Fig. 3. Precision comparison for all the tested sequences. On top without auto-covariances, bottom the same sequences with auto-covariances.

The presented results are qualitatively different from the ones in previous works of Beran et. al. [13] and Klicnar et. al. [14], who aimed at detection of similar shots within a longer video programmes, such as TV news broadcast, with the aim of identifying scenes which were cut or added. In contrast, the sequences used in this paper are each a single shot, and we strive to match the individual frames, rather than longer blocks.

Time of the processing was also evaluated on several randomly chosen sequences, as reporting times for all the sequences would be rather chaotic. The processing time is dominated by the extraction of the frame description vectors, which is currently implemented using the OpenCV library [1], but could be also easily accelerated by the GPU. The relative times are reported in Fig. 4. The absolute times are reported in Fig. 5, and indicate superlinear scaling, which is due to efficient implementation. Please note that the time of video decoding, which was done using the FFmpeg library [2], was included in the feature extraction stage.

[1] http://opencv.org/
[2] http://ffmpeg.org/

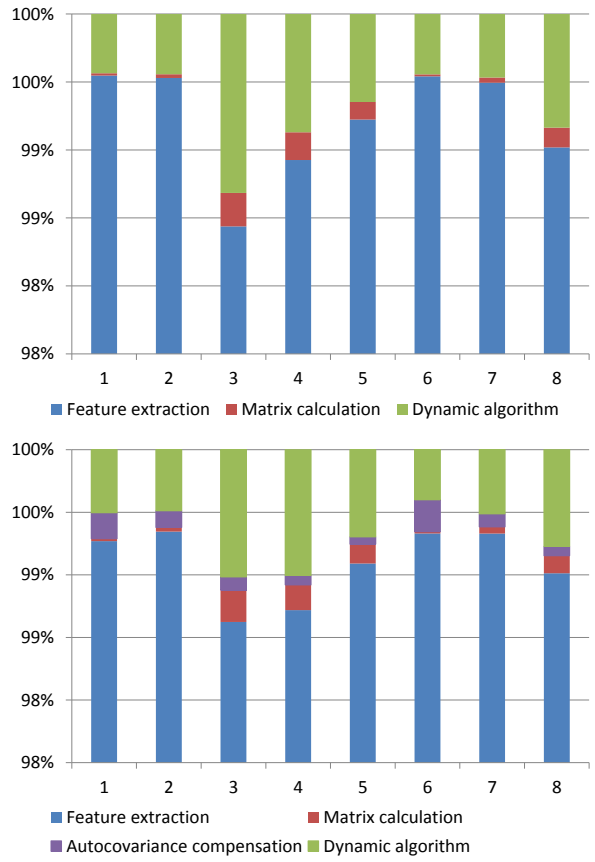

Fig. 4. Timing comparison on a few randomly chosen pairs of sequences. On top without auto-covariances, bottom the same sequences with auto-covariances. Please note that the feature extraction takes a vast majority of the time (see the vertical axis labels).

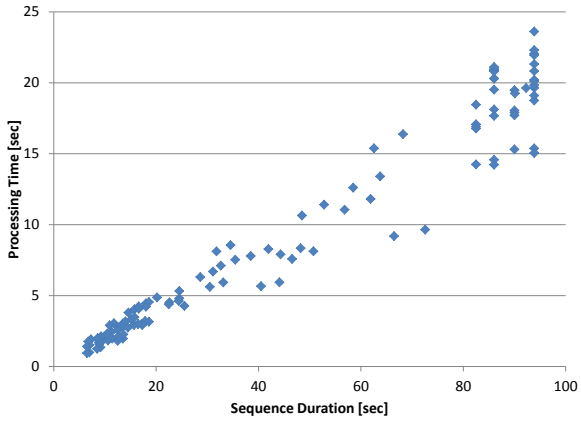

Fig. 5. Scaling of the runtime of the proposed algorithm, with respect to sequence duration (times with and without auto-covariances are practically identical).

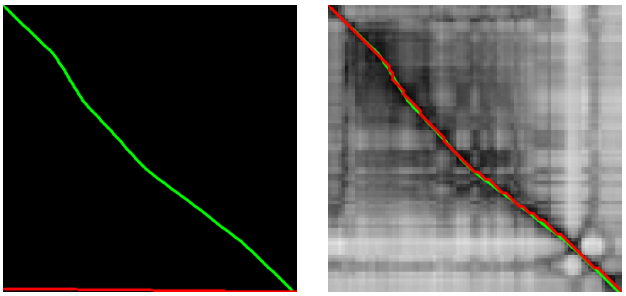

Fig. 6. The cross-covariance matrices and time synchronization curves, ground truth in green, the obtained results in red. a) results with auto-correlations (a failure case) on the left, b) results for simple γ (fallback) on the right. Best viewed in color.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we described a fast and robust method for low-level activity description and time synchronization of video sequences. This is highly applicable in digital cinema production, where many takes of the same scene are captured with slightly different timing of actor actions. The proposed method allows for time synchronization of such shots, so that technical quality indicators can be evaluated on a per-frame basis and compared, e.g. to rank the takes by quality. One such indicator is in-focus detection, which is meant to help the filmmakers to quickly discard out-of-focus footage. This method will be presented in a separate paper.

The proposed method was tested on two suitable datasets, where it performed admirably. As demonstrated by the synthetic data, even abrupt changes in speed of actions in the video lead to only gradual changes in the time synchronization curves. The dynamic algorithm could be thus regularized to recover smooth curves, to further improve the precision.

Although the algorithm is precise, its computational performance is currently limited by the extraction of the frame descriptors. This extraction would be suitable for GPU acceleration.

Additionally, with the very large data in mind, a slightly different approach could be devised, with a different performance scaling. The proposed method incurs one cost for the preprocessing step (feature extraction), which can be performed independently for each shot, and then the pairwise matching incurs a second cost, which is performed pairwise. This is not generally a problem in the movie industry, where only a limited number of takes of each particular shot exists. But when matching a large set, the cost of the pairwise processing will grow exponentially. For the means of the technical quality evaluation, the video synchronization is not entirely needed. It should be possible to evaluate an (approximate) function which gives quality sampling density over the shot, so the (scalar) qualities of any two shots would be comparable. This would yield completely independent processing of each shot, and thus perfect linear scaling.

6. REFERENCES

[1] Yaron Caspi and Michal Irani, "A step towards sequence-to-sequence alignment," in *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*. IEEE, 2000, vol. 2, pp. 682–689.

[2] Yaron Caspi, Denis Simakov, and Michal Irani, "Feature-based sequence-to-sequence matching," *International Journal of Computer Vision*, vol. 68, no. 1, pp. 53–64, 2006.

[3] Philip A Tresadern and Ian Reid, "Synchronizing image sequences of non-rigid objects.," in *BMVC*, 2003, pp. 1–10.

[4] Cen Rao, Alexei Gritai, Mubarak Shah, and Tanveer Syeda-Mahmood, "View-invariant alignment and matching of video sequences," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*. IEEE, 2003, pp. 939–945.

[5] Tinne Tuytelaars and Luc Van Gool, "Synchronizing video sequences," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*. IEEE, 2004, vol. 1, pp. I–762.

[6] Josef Sivic and Andrew Zisserman, "Video google: A text retrieval approach to object matching in videos," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*. IEEE, 2003, pp. 1470–1477.

[7] Anthony Whitehead, Robert Laganiere, and Prosenjit Bose, "Temporal synchronization of video sequences in theory and in practice," in *Application of Computer Vision, 2005. WACV/MOTIONS'05 Volume 1. Seventh IEEE Workshops on*. IEEE, 2005, vol. 2, pp. 132–137.

[8] Manabu Ushizaki, Takayuki Okatani, and Koichiro Deguchi, "Video synchronization based on co-occurrence of appearance changes in video sequences," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*. IEEE, 2006, vol. 3, pp. 71–74.

[9] Ondřej Chum, James Philbin, Michael Isard, and Andrew Zisserman, "Scalable near identical image and shot detection," in *Proceedings of the 6th ACM international conference on Image and video retrieval*. ACM, 2007, pp. 549–556.

[10] Jingyu Yan and Marc Pollefeys, "Video synchronization via space-time interest point distribution," in *Advanced Concepts for Intelligent Vision Systems*, 2004, pp. 501–504.

[11] Quoc V Le, Will Y Zou, Serena Y Yeung, and Andrew Y Ng, "Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 3361–3368.

[12] Ivo Řezníček and Pavel Zemčík, "Human action recognition for real-time applications," in *Proceedings of the 3rd International Conference on Pattern Recognition Applications and Methods*, 2014, pp. 646–653.

[13] Vítězslav Beran, Adam Herout, and Pavel Zemčík, "Online video synchronization based on visual vocabularies," 2010.

[14] Lukáš Klicnar, Vítězslav Beran, and Pavel Zemčík, "Dissimilarity detection of two video sequences," *Proceedings of SCCG 2013*, vol. 2013, pp. 28–31, 2013.

[15] J-M Geusebroek, Rein Van den Boomgaard, Arnold W. M. Smeulders, and Hugo Geerts, "Color invariance," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, no. 12, pp. 1338–1350, 2001.

[16] Marcin Marszałek, Ivan Laptev, and Cordelia Schmid, "Actions in context," in *IEEE Conference on Computer Vision & Pattern Recognition*, 2009.

[17] Lukáš Polok and Pavel Smrz, "Fast linear algebra on gpu," in *IEEE conference proceedings*. 2012, p. 6, IEEE Computer Society.

[18] George Awad, Paul Over, and Wessel Kraaij, "Content-based video copy detection benchmarking at trecvid," *ACM Trans. Inf. Syst.*, vol. 32, no. 3, pp. 14:1–14:40, July 2014.