On-line Video Synchronization Based on Visual Vocabularies

Vítězslav Beran, Adam Herout, Pavel Zemčík

Brno University of Technology
Faculty of Information Technology
Department of Computer Graphics and Multimedia
Božetěchova 2, 612 66 Brno, CZ

beranv, herout, zemcik@fit.vutbr.cz

ABSTRACT

This paper presents the procedure for on-line visual-content-based video synchronization. The motivation of our pioneering work is the existence of several off-line video processing systems employed in video classification or summarization applications, but no evidence of on-line solutions for video analysis. In some applications, the video streams go through the 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. Our solution employs visual vocabularies that allow signing the video frames by bag-of-words. The synchronization procedure is then 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 discuss 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 the video synchronization systems is to detect and validate the time offset between similar video streams. The motivation of our pioneering work is the existence of several off-line video processing systems employed in video classification or summarization applications, but no evidence of on-line solutions for video analysis. In some applications, the video streams go through the broadcast systems that delay the original video and also distort the original signal. The system that would be able to automatically detect such delays or distortions or detect the transmission errors is highly required.

Our research is focused on visual content, so audio is omitted. Our previous approaches [Ber08] based mostly on global features were poorly robust geometrical transformations that led our research to employ local features and visual vocabularies.

The visual vocabulary used for video content analysis is introduced in [Siv03]. Motivated by text retrieval techniques, visual 'words' are pre-computed using vector quantization and inverted file approach document ranking are used. It results in immediate returning of a ranked list of documents (key frames, shots, etc.) in the manner of search in text documents. Two types of sparse features are used: corner-like detector with full affine adaptation [Mik04] and detection of stable regions (MSERs [Mat02]). The local features are described by SIFT descriptor and quantized using naïve *k*-means algorithm. When introduced, the visual vocabularies were used in image and object retrieval applications. Needs for large, more discriminative vocabularies lead the research to find faster clustering approaches. Two significant methods were developed: hierarchical *k*-means [Nis06] and approximated *k*-means [Phi07]. The developed methods allow creation of vocabularies with size about 1M of visual words in reasonable time and computational cost.

The later research experimenting with different schemas when weighting the words into the bag-of-words showed that the discriminative power of the vocabulary could be improved not only by extending the vocabulary's size. Instead of standard *tf-idf* weighting [Siv03], a single feature descriptor could be assigned to several visual words nearby in the descriptor space. The soft-weighting results are

published in [Phi08][Jia07]. Further improvements in precision and number of retrieved candidates were done in [Chu08], where authors employed min_Hash algorithm and later also utilize the geometrical information from local features [Chu09].

The methods dealing with NDID (Near-Duplicate Image Detection) or NDSD (Near-Duplicate Shot Detection) is mainly focused on searching the most similar samples in the image database. The [Chu07] employed both, global (color histograms with locality sensitive hashing) and local features to search for near-duplicates in the large image database.

The real-time applications such as on-line video synchronization introduce specific demands to the commonly used techniques. The stability and robustness of the local features detection and discriminative power of feature description and quantization could be decreased at the expense of execution time increase. The computational cost demands also derive the size of the visual vocabulary.

The paper firstly introduces the overview of two stages of the video synchronization system. Next sections describe the particular steps of each stage in more detail. Section 3 makes an overview of state-of-the-art image feature extraction and description methods. The searching strategies are described in section 4. The possibilities of visual words weighting when bag-of-words are constructed are discussed in section 5. The sections 6 and 7 describe the experiments – used data and explored transformations and discuss the results.

2. SYSTEM STAGES

The general objective of designed system is on-line monitoring of two or more video streams (query video streams) and detection of possible distortions in space or time domain comparing to the reference video stream.

This work we focused on visual-based video synchronization in time domain. The objectives of the algorithm are:

- real-time image signatures computation,
- real-time image retrieval,
- on-line time shift detection robust to basic video distortions.

The way we utilize the visual vocabularies comprise two stages: off-line vocabulary construction and on-line image retrieval. The steps of both procedures are examined in more detail in following chapters.

The first step of the **visual vocabulary construction** is the extraction of the image features from training dataset. The next section overviews the existed feature detectors in more detail. Then the clustering step divides the features' descriptor space. The clusters' labels are used as visual words. Usually, the iterative *k*-means clustering algorithm is used. The vocabulary training stage is on Figure 1.

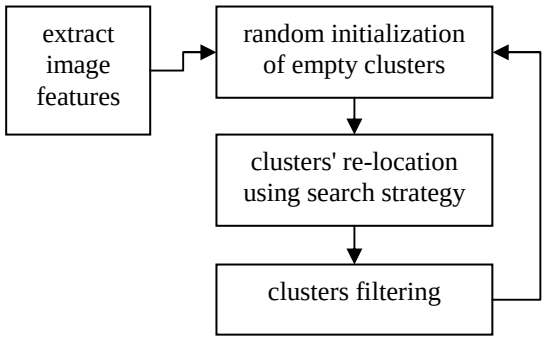

Figure 1. Visual vocabulary construction.

The procedure iteratively re-locates the cluster centers based on mean value of the closest samples to particular cluster. The filtering stage removes (empties) the clusters that do not follow predefined constraints (e.g. cluster size). The various searching strategies are described in section 4 for more details.

The on-line **image retrieval** stage firstly extract features from the image, translate the features' descriptors to visual words using visual vocabulary and compute image signature (bag-of-words). The weighting scheme usually used for image signature computing is described in details in section 5. The steps of on-line stage are on Figure 2.

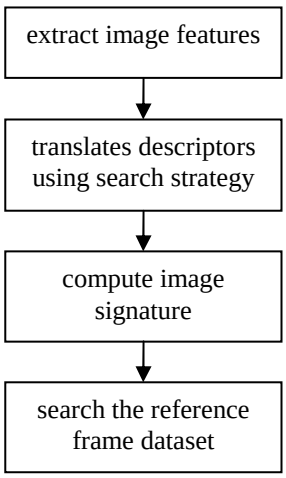

Figure 2. On-line image retrieval.

The image retrieval is usually based on inverted file approach. This is convenient to use when huge dataset needs to be searched. Our main objective is to design the on-line system, so we keep only limited amount of last video frames in temporary buffer. Until the size of the buffer is up to 10k frames, the naïve serial search for image retrieval can be used.

3. SPARSE FEATURES

In this work, the sparse features mean image key-regions. The key-regions can be constructed on two fundamental image structures: key-points or regions. The following section discusses the most popular approaches. The key-points 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.

One subset of detectors analyses the local changes in image intensity. The scale-adopted Harris function [Mik04] is sensitive to corner-like structures. The Hessian [Mik05] function, Difference of Gaussian (DoG) [Low04], Laplacian of Gaussian (LoG) [Lin98] detects the blob-like structures. The Harris and Hessian function are extended by characteristic scale detection which is where the LoG function 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 [Bay06] show that the Hessian function can be effectively approximated by block filters. The SURFs are based on effective platform based on computation of Haar-wavelets on integral images. The authors also introduced new descriptors utilizing the same platform.

The Harris corners and Canny curves are also adopted in edge-based approach [Tuy04] which analyzes the parallelograms defined by one anchor point (Harris corner) and two points running from the anchor point along two lines (Canny curves). The local extremas of intensity functions computed over the parallelogram is taken as the stable region. The authors developed another approach based strictly on image intensity values. Both approaches result in affine invariant regions.

The approach known as "FAST corners" [Ros06] employs machine learning to construct 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 can be applied. Unfortunately, they significantly slows down the process. In our application, only slight geometrical transformations are supposed so the fully affine invariance is not needed.

The region-based detectors are represented by MSER (Maximally Stable Extremal Regions) developed by [Mat02]. 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.

In this work, the speed factor is the most crucial. The SURF detector offers convenient balance between speed, robustness, precision and discriminative power. We used the SURF detector and descriptor in extended version that gives 128-dimensional feature descriptors.

4. VISUAL VOCABULARY

The idea of visual vocabulary, firstly used in "Video Google" [Siv03], 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 by a quantization process. The objective is to use vector quantization to descriptors to translate them into clusters' labels which represents the visual words.

Visual vocabulary is created during the training stage. A part of the data (training data) is used to find cluster centers in the descriptor space. When the size of the resulting vocabulary is small ($k < 10^5$), the naïve *k*-means algorithm can be used. 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) need more discriminative vocabulary. One possible way how to reduce the time complexity is using Hierarchical *k*-means [Nis06]. Instead of solving one clustering with a large number of cluster centers, a tree organized hierarchy of smaller clustering problems is solved. This reduces the time complexity to $O(N.\log k)$. The problem with HKM is that it optimizes the problem only locally, per tree branch. Other approach reducing the time complexity is replacing the nearest neighbor search of *k*-means by forest of kd-trees. The authors [Phi07] called this approach as Aproximate *k*-means.

This work utilizes the idea from AKM with one *kd*-tree for space search. A random selection of images from training datasets serves to generate feature descriptions. Because the discriminative power of the visual vocabulary video synchronization system does not need to be superior, we built the vocabulary of sizes 1k of visual words.

5. IMAGE SIGNATURE

In our framework, the *image signature* is a collection of weighted words – vector of words' frequencies. It degrades to a set-of-words when the weights represent only the word's presence. Otherwise, it is a

bag-of-words. In recent works, several weighting schemes were introduced and evaluated.

In the pioneering work [Siv03], 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:

$$tf-idf(w) = tf(w) * idf(w) = \frac{|d(w)|}{|d|} * \log\left(\frac{|D|}{|D(w)|}\right) \quad (1)$$

where d is a document (set of words), $|d|$ 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 work 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. Let us have a sorted list of k nearest visual words. An exponential function of the distance to the cluster center used in [Phi08] is

$$weight = \exp\left(-\frac{d^2}{2\sigma^2}\right) \quad (2)$$

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 [Jia07] is based on the rank of the assigned cluster:

$$weight = \frac{1}{2^{i-1}} \quad (3)$$

where i is the ith nearest neighbor.

At the retrieval stage, documents are ranked by their similarity. One of the mostly used similarity metric 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.

$$sim(\mathbf{q},\mathbf{d}) = \frac{\mathbf{q}\cdot\mathbf{d}}{|\mathbf{q}||\mathbf{d}|} \quad (4)$$

where $\cdot$ is a dot product and $|\ldots|$ is the vector magnitude.

For the experiments in this work, the standard *tf-idf* weighting scheme is used when visual words are weighted in bag-of-words.

6. TIME SHIFT DETECTION

The system contains a frame buffer for each video stream. Local features are extracted and described for each incoming frame (query and reference streams) and used to create frame signature. The signatures are stored to the buffer for each video stream and used to compare frames. The similarities between each new frame from query video and all reference frames in the reference buffer are computed.

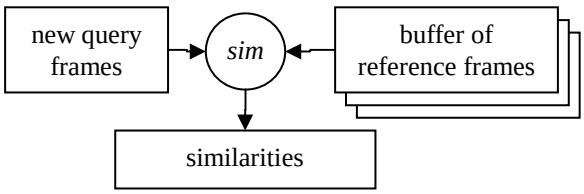

Figure 3. Time-shift detection.

The similarities between query frame and the frames from buffer of reference frames in time t can be expressed as a function:

$$s(t,i) = sim(\mathbf{q}_t, \mathbf{d}_{t,i}) \quad (5)$$

where $\mathbf{q}_t$ is a signature of the query video frame and $\mathbf{d}_{t,i}$ is the signature of the i-th frame in the reference buffer at time t. Using the function in equation (5), the probability of a time offset i can be expressed as the weighted sum of previous similarities:

$$O(t,i) = w(t) \otimes s(t,i) \quad (6)$$

where w is the weighting function. Two weighting functions were used in experiments – averaging (Eq. 7) and exponential function (Eq. 8):

$$w(t) = \frac{1}{N} \quad (7)$$

$$w(t) = 1 - \log_N t \quad (8)$$

where N is the window (buffer) size. The most probable time offset is then taken as the detection result:

$$offset(t) = \arg\max_i O(t,i) \quad (9)$$

The voting strategy is employed to sum up the most probable time offsets over the time which stabilizes the detection results.

Along with the detected time shift, the certainty level of the detection is computed. The certainty level is computed from the two bins from the voting that have the most votes. The precise function computing the certainty level is:

$$c\_level(t) = 1 - \frac{|votes(t)|_2}{|votes(t)|_1} \quad (10)$$

where $votes(t)$ is the ranked list of voting bins in time t and $|\ldots|_r$ is the amount of votes in the r-th bin.

7. EXPERIMENTS

We use "TRECVID 2009 data BBC subset" to evaluate our system. The subset contains 77 video files. The TRECVID video data contains a variety of challenging disruption itself, especially for time video synchronization task.

The most crucial when detecting the time shift between two similar video streams based on image content analysis 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.

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

The query video streams were created artificially. In this work, we focused on fundamental types of video distortions:

- a uniform scale transformation,
- partial occlusion of the visual content by static banner and
- white noise.

The examples of particular simulated distortions are on Figure 4.

Figure 4. Examples of video distortions (top – overlaid banner, 90% resize; bottom – 30% noise, smooth with σ 3.0).

The system sensitivity to the different time shift was also explored. The reference video was delayed by 1 and 10 seconds.

All experiments used the visually vocabulary trained on different dataset than the testing one. For training purposes, the Kentucky dataset [Nis06], containing 10200 images, was used. The dataset contains 4-tuples of object images taken from difference viewpoint.

8. RESULTS

Each experiment run was done on all 77 video files and certainty level from all files was aligned and averaged. The alignment is necessary because the length of the initialization shot with color stripes at the beginning of each video differs. The video frame size is 352x288 pixels.

The first experiments were focused on comparison of two weighting functions used in time shift detection. When undistorted data were used for the test, no significant difference appears.

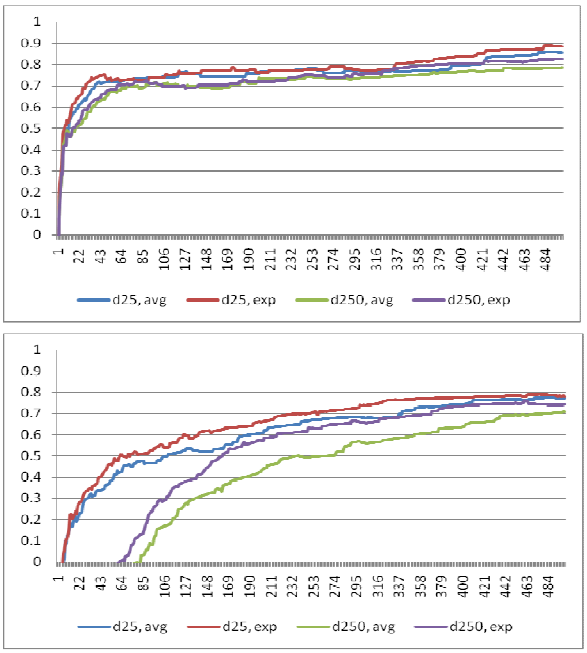

Figure 5. Weighting functions comparison for time-shift detection (top – without distortion, bottom – resize to 90%).

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

The robustness to partial occlusion of the video frames was tested using overlaid banner. The graph on Figure 6 shows that the precision of the detector is not affected by partial occlusion; at least up to 25 frames delay. The robustness to the noise was tested with the noise level in between 0%-30%. The Figure 7 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 on Figure 8.

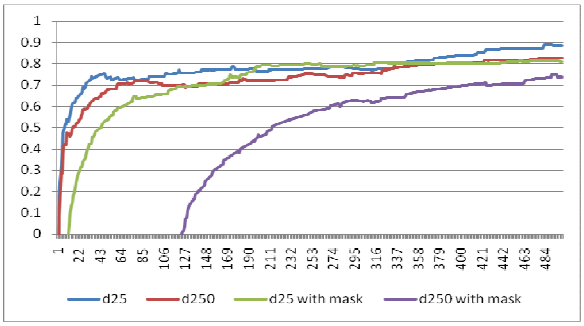

Figure 6. Partial occlusion by a banner.

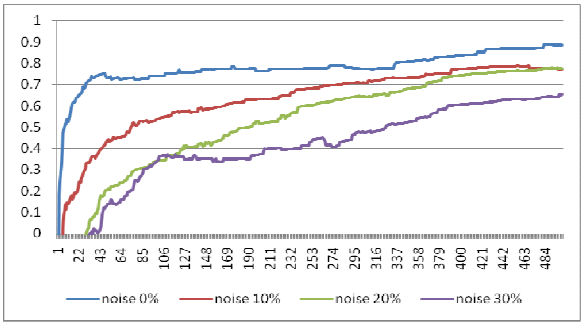

Figure 7. Added noise of different levels.

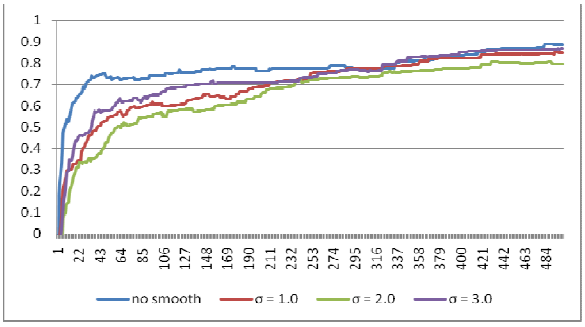

Figure 8. Gaussian blur applied with various σ.

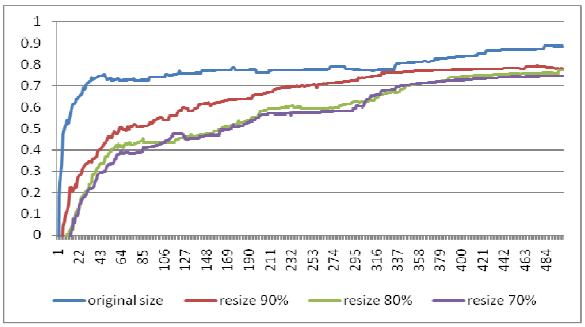

Figure 9. Uniform rescale transformation.

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.

9. CONCLUSION

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 bag-of-words. Several types of video distortions were addressed. 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 basement for the other methods; e.g. dealing with geometrical transformation between video streams. The knowledge of the geometrical transformation might be useful when non-uniform geometrical distortion is expected and the automatic video quality control task analyzing noise ratio, contrast changes, color bias, etc. is required.

10. 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), by the research project "Security-Oriented Research in Informational Technology" CEZMŠMT, MSM0021630528 and EU IST FP6 projects "AMIDA" EU-6FP-IST, IST-033812-AMIDA.

11. REFERENCES

[Bay06] Bay, H., Tuytelaars, T., Van Gool, L. *SURF: Speeded Up Robust Features*, Proceedings of the ninth European Conference on Computer Vision, 2006.

[Ber08] Beran, V., Herout, A., Hradiš, M., Řezníček, I., Zemčík, P. *Video Summarization at Brno University of Technology.* In: ACM Multimedia 2008, New York: ACM, 2008.

[Chu07] Chum, O., Philbin, J., Isard, M. and Zisserman, A. *Scalable Near Identical Image and Shot Detection.* CIVR, 2007.

[Chu08] Chum, O. , Philbin, J. and Zisserman, A. *Near Duplicate Image Detection: min-Hash and tf-idf Weighting*. BMVC, 2008.

[Chu09] Chum, O., Perdoch, M., and Matas, J. *Geometric min-Hashing: Finding a (Thick) Needle in a Haystack*. CVPR, 2009.

[Jia07] Jiang, Y., Ngo, C., and Yang, J. *Towards optimal bag-of-features for object categorization and semantic video retrieval.* In Proc CIVR, 2007.

[Lin98] Lindeberg, T. Feature detection with automatic scale selection. Int. J. of Computer Vision, 30:79-116, 1998.

[Low04] Lowe, D. Distinctive image features from scale-invariant keypoints. Int. Journal on Computer Vision, 60(2):91-110, 2004.

[Mat02] Matas, J., Chum, O., Urban, M. and Pajdla, T. *Robust wide-baseline stereo from maximally stable extremal regions.* In Proceedings of the British Machine Vision Conference, Cardiff, UK, pp. 384–393, 2002.

[Mik05] Mikolajczyk, K., et. al. A comparison of affine region detectors. Int. Journal on Computer Vision, 65(1/2):43-72, 2005.

[Mik04] Mikolajczyk, K., Schmid, C. Scale and affine invariant interest point detectors. Int. Journal of Computer Vision, 60:63-86, 2004.

[Nis06] Nister, D., Stewenius, H. *Scalable recognition with a vocabulary tree.* In Proc. CVPR, 2006.

[Phi07] Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A. *Object retrieval with large vocabularies and fast spatial matching.* In Proc. CVPR, 2007.

[Phi08] Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A. *Lost in quantization: Improving particular object retrieval in large scale image databases.* In Proc. CVPR, 2008.

[Ros06] Rosten, E., Drummond, T. *Machine learning for high-speed corner detection.* In European Conference on Computer Vision, 2006.

[Siv03] Sivic, J., Zisserman, A. *Video Google: A text retrieval approach to object matching in videos.* In Proc. ICCV, 2003.

[Tuy04] Tuytelaars, T., Van Gool, L. *Matching Widely Separated Views based on Affine Invariant Regions.* International Journal on Computer Vision, 59(1):61–85, 2004.