Brno University of Technology at TRECVid 2010 SIN, CCD

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

Semantic indexing

Michal Hradiš, Ivo Řezníček, David Bařina, Adam Vlček, Pavel Zemčík
ihradis, ireznice, ibarina, ivlcek, zemcik@fit.vutbr.cz

1. The runs differ in the types of visual features used. All runs use several bag-of-word representations fed to separate linear SVMs and the SVMs were fused by logistic regression.

*F_A_Brno_resource_4: Only single best visual features (on the training set) are used – dense image sampling with rgb-SIFT.

* F_A_Brno_basic_3: This run uses dense sampling and Harris-Laplace detector in combination with SIFT and rgb-sift descriptors.

*F_A_Brno_color_2: This run extends F_A_Brno_basic_3 by adding dense sampling with rg-SIFT, Opponent-SIFT, Hue-SIFT, HSV-SIFT, C-SIFT and opponent histogram descriptors.

* F_A_Brno_spacetime_1: This run extends F_A_Brno_color_2 by adding space-time visual features STIP and HESSTIP.

2. Combining multiple types of visual features improves results significantly. F_A_Brno_color_2 achieve more than twice better results than F_A_Brno_resource_4. The space-time visual features did not improve results.

3. Combining multiple types of visual features is important. Linear SVM is inferior to non-linear SVM in the context of semantic indexing.

Content-based Copy Detection

Vítězslav Beran, Adam Herout, Pavel Zemčík
beranv, herout, zemcik@fit.vutbr.cz

1. Two runs submitted, but with similar settings; the difference is only in amount of processed test data (40% and 60%)
   - brno.m.*.l3sl2: SURF, bag-of-words (visual codebook: 2k size, 4 nearest neighbors used in soft-assignment), inverted file index, geometry (homography) based image similarity metric
2. What if any significant differences (in terms of what measures) did you find among the runs?

   - only one setting used – no differences
3. Based on the results, can you estimate the relative contribution of each component of your system/approach to its effectiveness?
   - slow search in reference dataset due to unsuitable configuration of used visual codebook
4. Overall, what did you learn about runs/approaches and the research question(s) that motivated them?
   - change the way of describing the video content – frame based (or key-frame based) approach is not sufficient

Acknowledgements

This work has been supported by the project Center of Computer Graphics, center of basic research, Czech Ministry of Education, Youth and Sports MSMT LC-06008; and Security-Oriented Research in Informational Technology, Czech Ministry of Education, Youth and Sports, CEZMSMT MSM0021630528; and Recognition and presentation of multimedia data, Faculty of Information Technology, Brno University of Technology, Czech Republic, FIT-S-10-2; and the EU FP7 project TA2: Together Anywhere, Together Anytime ICT-2007-214793, grant no 214793; and by EU project FP7-ARTEMIS "R3-COP", grant no. 100233; and EU project FP7-ARTEMIS "SMECY", grant no. 100230.

Semantic indexing

Our approach to semantic indexing combines supervised machine learning with description of video shots in terms of frequencies of local visual primitives. Similar approaches were previously shown to be suitable for this type of tasks [Lazebnik et al., 2006; van de Sande et al., 2010; Snoek et al., 2009].

The construction of video-shot descriptors can be divided into three separate steps. First, a sampling was used to select parts of the video which are of interest. The appearance of the selected video parts was then expressed by a multidimensional feature vector which is resistant to small displacements and other local transformations while retaining most of the useful information. Based on the local descriptors, a bag-of-word representation describing the whole shot was created. A shot was represented as multiple bag-of-word vectors, each based on different combination of sampling and appearance description. Linear support vector machine (SVM) classifiers were trained separately on these bag-of-word representations and their predictions were fused by logistic regression. An overview of the whole processing pipeline is shown in Figure 1.

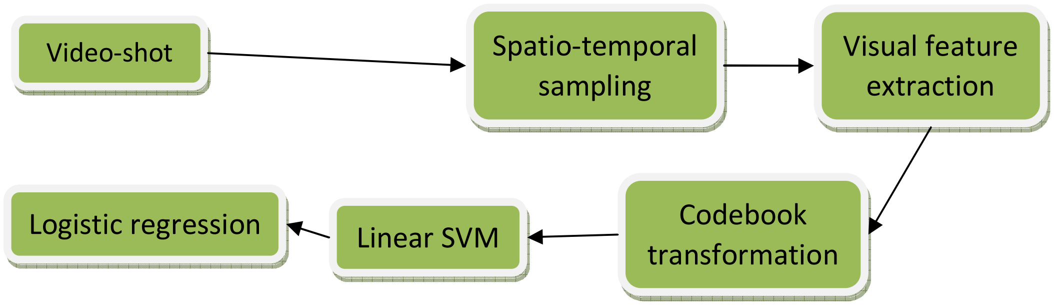

Figure 1 SIN - processing pipeline

The following text explains in detail each part of the processing pipeline and also the results achieved in TRECVID 2010 evaluations. First, the sampling is explained together with appearance description. Next, the transformation to bag-of-word representation is discussed. The following part then gives details on the machine learning. Finally, the achieved results are presented together with discussion of contributions of the individual parts of the pipeline.

Spatio-temporal sampling

Three different types of sampling were used. Dense sampling and Harris-Laplace interest region detector were used to select regions in key-images. Only single key-image was selected to represent each shot. Additionally, two spatio-temporal interest point detectors were used to sample salient spatio-temporal volumes from the whole shot.

The Harris-Laplace [Mikolajczyk and Schmid, 2004] detector selects stable areas with high intensity changes and it also provides characteristic scale of the local area. Sapling using Harris-Laplace detector is denoted as HARLAP in the further text.

The dense sampling samples images on regular spatial grid. We have used spacing between the sampled areas 8 pixels in both horizontal and vertical directions and the radius of the extracted areas was 8 (DENSE8) and 16 (DENSE16) pixels.

The Harris3D detector 0 is a space-time extension of the Harris detector. It is based on the spatio-temporal second-moment matrix

$$\mu(.;\sigma,\tau) = g(.;s\sigma,s\tau) * (\nabla L(.;\sigma,\tau)(\nabla L(.;\sigma,\tau))^{T})$$

using independent spatial and temporal scale values σ, τ, a separable Gaussian smoothing function *g*, and space-time gradients ∇L. The final locations of space time interest points are given by local maxima of H = det(μ) – *k* trace^{3}(μ), H > 0. This extraction method will be further denoted as the STIP detector.

The Hessian detector 0 is a spatio-temporal extension of the Hessian saliency measure for blob detection in images. This detector measures the saliency with the determinant of the 3D Hessian matrix. The position and the scale of the interest points are simultaneously localized without any iterative procedure. In order to speed up the detector, approximating box-filter operations on an integral video structure is used. Each image scale-space octave is divided into 5 scales, with a ratio between subsequent scales in the range 1.2 – 1.5 for the inner 3 scales. The determinant of the Hessian is computed over several octaves of both the spatial and temporal scales. A non-maximum suppression algorithm selects joint extrema over space, time and scales: (x, y, t, σ, τ). This extraction method will be further denoted as the HESSTIP detector.

Visual feature extraction

Appearance of the sampled regions was expressed in terms of multiple local descriptors. For Harris-Laplace detector and dense sampling, various types of SIFT-like descriptors and color histograms were used. In the case of spatio-temporal volumes two 3D descriptors were used.

Van de Sade et al. [van de Sande 2010] analyzed properties of various visual descriptors focusing on illumination invariance. We used a subset of these descriptors. The particular descriptors were SIFT, rgb-SIFT, rg-SIFT, Opponent-SIFT, Hue-SIFT, HSV-SIFT, C-SIFT and Opponent histogram.

The descriptor for the STIP detector is based on computation of spatial gradient and optic flow accumulated in space-time neighborhoods of detected interest points; these two histograms are finally concatenated. The HESSTIP descriptor is based on extension of SURF [Bay et al., 2006] image descriptors for videos. 3D patches are divided into cells, and each cell is represented by a vector of weighted sums of uniformly sampled responses to the Haar-Wavelets along all three axes.

Codebook transform

Codebook transformation creates compact, yet powerful representation. In the original form [Lazebnik et al., 2006], the visual features are assigned each to the most similar visual word based on distance in the visual descriptor space. The prototypes of the visual words together form a codebook – thus the name codebook transformation. The codebook transformation produces bag-of-word representation which captures occurrence frequencies of the visual words in a document while discarding any spatial information. Simple ways how to retain some of the spatial information exist [Lazebnik et al., 2006], but these were not used in our system. The further text explains the specifics of our approach.

Separate codebooks were created of each combination of sampling and descriptor. The codebooks were constructed by running 15 iterations of k-means algorithm on 600 MB of randomly selected local features from the training dataset. The size of all codebooks was 4098.

To minimize the amount of lost information, the local features were translated to visual words by soft-assignment instead of hard-assignment. Particularly, codeword uncertainty 0 was used. The kernel was Gaussian and its size represented by standard deviation was equal to average distance of the closest words in the particular codebook. The resulting histograms were not normalized

Classification

The schema of the classification is shown in Figure 2. The main issues for the machine learning part were how to merge information from multiple sources and how to manage relatively large dataset with 130 classes. Generally, SVM is the most common choice of learning algorithm for classification problems where the feature vectors are bags-of-visual-words [Lazebnik et al., 2006; van de Sande et al., 2010; Snoek et al., 2009] and information from different sources is usually merged in kernel [Snoek et al., 2009]. Another possibility is to perform late fusion of separate classifiers each based on the individual information source.

For computational reasons, we decided to use linear SVM to learn separate classifiers for each type of bag-of-word representation and to fuse the separate models linearly by adapting weights of individual models by logistic regression. This approach allowed us to utilize all annotated samples from the training set.

LIBLINEAR [Fan et al. 2008] implementation of SVM solver and logistic regression was used to learn all models. The library was slightly modified to allow terminating computation after a fixed number of iterations.

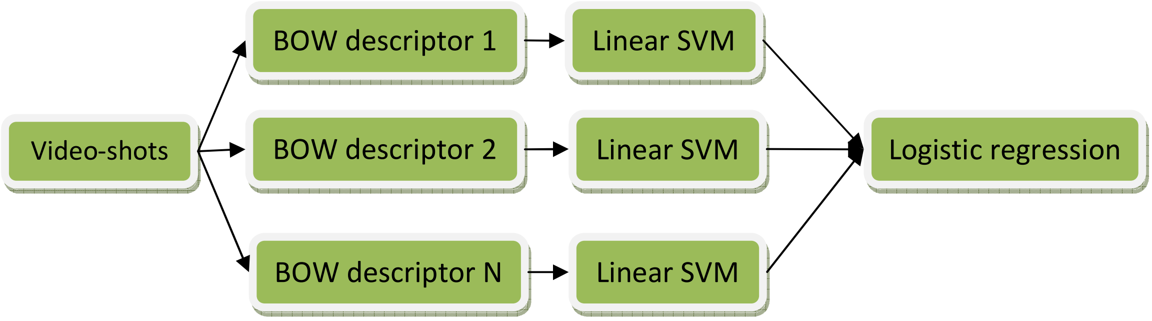

Figure 2 SIN - Classification schema

The soft margin parameter of SVM and the regularization parameter of the logistic regression were both selected separately by grid search with 5-fold cross-validation. The objective function for this parameter optimization was the average precision and the parameters were optimized for each class separately. To utilize all training data for both SVM learning and for the logistic regression, the five SVM classifiers created in cross-validation produced responses for the samples from the training set which were not used to train the particular classifier. Logistic regression was than trained on this whole dataset merged from responses of the five classifiers. The final SVM classifiers were trained on the whole training set and the final fusion was also learned on the full dataset.

The runs

We submitted four different runs which differ in types of visual features used. The complete overview of the visual features used in each run is summarized in Table 1.

Brno_resource used only single sampling and single descriptor. In this case, no fusion was involved. The sampling was DENSE16 which was combined with rgb-SIFT descriptor. This combination was the visual feature type best performing on the training set individually.

Brno_basic combined HARLAP, DENSE8 and DENSE16 sampling with SIFT and rgb-SIFT descriptors.

Brno_color extended Brno_basic with additional color descriptors. These were rg-SIFT, Opponent-SIFT, Hue-SIFT, HSV-SIFT, C-SIFT and Opponent histogram combined with DENSE16 sampling.

Brno_spacetime extended Brno_color by adding STIP and HESSTIP space-time visual features.

Results

The results achieved on training set in five-fold cross-validation by separate types of visual features are shown in Table 1. The results were measured as mean average precision across all 130 classes. It can be clearly seen that dense sampling provides generally better results than Harris-Laplace detector. However, these observations are no longer valid when looking at performance on separate classes. Opponent-histogram descriptor performs quite poorly as it captures only color information and not he patterns. The spatio-temporal features also do not provide competitive results which could be due to too sparse sampling or due to the fact that we did not normalize the feature vectors with respect to video-shot length.

Result of the individual runs in both the cross-validation on the training data and also the results on test data assessed by NIST are shown in Table 2. The results show that fusion of multiple visual features improves results significantly over single best visual feature type represented by Brno_resource. Brno_color gave more than twice better results on the test set than Brno_resource. The addition of space-time features in Brno_spacetime did not improve the result. In reality, the effect was opposite. This degradation of results can be explained by relatively worse results achieved by STIP and HESSTIP features individually (see Table 1).

The results on the test set are still significantly worse than the best results achieved in the evaluations. Most of this performance gap can be explained by the fact that we use only linear SVM. From our other experiments, we expect that switching to non-linear SVM would improve the results by 50–70 %. We also use only training data specific to TRECVID 2010 and utilizing past data would also improve the results. Another shortcoming of our approach is that we handle the classes independently, even thou they are certainly not. Modeling these dependencies would improve results especially for classes where there is not much training data available. Detectors of object classes such as faces, people and cars used as feature extractors would also improve results and we plant to follow this idea in the future.

Sampling	Descriptor	Rotation Invariant	Mean AP	resource	basic	color	Space-time
HARLAP	SIFT	yes	0.161		X	X	X
		no	0.173		X	X	X
	rgb-SIFT	yes	0.164		X	X	X
		no	0.173		X	X	X
DENSE8	SIFT	yes	0.179		X	X	X
		no	0.200		X	X	X
	rgb-SIFT	yes					
		no	0.211		X	X	X
DENSE16	SIFT	yes	0.194		X	X	X
		no	0.212		X	X	X
	rgb-SIFT	yes					
		no	0.223	X	X	X	X
DENSE16	rg-SIFT	no	0.215			X	X
	O-SIFT	no	0.216			X	X
	Hue-SIFT	no	0.177			X	X
	HSV-SIFT	no	0.201			X	X
	C-SIFT	no	0.220			X	X
	O-hist	no	0.137			X	X
STIP	STIP	no	0.080				X
HESSTIP	HESSTIP	no	0.075				X

Table 1 Visual features used by individual runs and results on the training set in cross-validation achieved by the separate types of features on all 130 classes.

RUN name	Training set mean AP	Test set inf. mAP
Brno_resource	0.223	0.021
Brno_basic	0.275	0.036
Brno_color	**0.291**	**0.045**
Brno_spacetime	0.280	0.041
TRECVID best		0.090
TRECVID median		0.039

Table 2 Results of the individual runs on training set in cross-validation on all 130 classes and on the testing set on the 20 classes selected by NIST for evaluation.

Content-based Copy Detection

System design

The CCD system is composed from three main parts: *video processing*, *reference database search* and *copy candidate verification*. The goal of the system is to find the possible existence of its parts in the reference dataset and detect the positions of the similar video-segments.

Having the reference database prepared, the query video is processed and query key-frames are detected, described and searched in database. The candidates (returned reference key-frames) of adjacent query key-frames are grouped into larger segments when possible. The candidate segments are then verified using more precise frame-content analysis with geometrical constraints and the location of the detected copy segment is refined.

Video processing

The presented system describes the visual content of video frames using SURF [Bay et al., 2006] and bag-of-words based on visual codebook [Sivic and Zisserman, 2003] (see Reference database search for more details). No other image features such as color histograms, texture analysis, gradient distribution, etc. are employed.

The SURF is the method for detection and description of significant local image structures. The method has very low computational cost and also high stability, robustness and for CCD task also tolerable precision. The frames are then represented as lists of local structures (position, scale), the characteristic orientation of the structure and feature vector (128 dimensions).

Two metrics are defined for content-based frame comparison: **Frame Similarity Error** and **Video Continuity Error**.

Frame Similarity Error - contains three scores: Mean Distance of Inliers, Inliers Ratio Score and Geometry Error.

- Score values represent errors: (0.0, 1.0), 0.0 – no error, 1.0 – high error value
- Having two frames and list of *matching points*
- Compute *homography* between frames using RANSAC
- Take inliers only - subset of matching points with *distance < threshold*
- **Mean Distance of Inliers** (normalized by *threshold*)
- **Inliers Ratio Score** – ratio between amount of inliers and number of matching points (normalized)
- **Geometry Error** – based on homography, distance from homography transformation parameters (scale, shear, perspective), the maximal distance from default parameter's values is taken as error

Video Continuity Error, based on analysis of adjacent frames, corresponds to amount of changed inliers of two adjacent frames.

During the video processing, the adjacent frames are compared and when the *Frame Similarity Error* is too high (*Inliers Ratio Error > 0.5* and *Geometry Error == 1.0*) the key-frame is detected. The result of the video processing step is the list of key-frames.

Reference database search

The approach for image content comparison using *Frame Similarity Error* cannot be used to search for the similar image in the dataset of hundreds of thousands images. The computational cost is unbearable.

The key-frame's list of descriptors is translated into one single vector called bag-of-words. Having the *visual codebook*, each descriptor can be assigned to one (hard assignment) or more (soft assignment) visual word. The bag-of-words then represents the distribution of visual words in the key-frame.

The **visual codebook** is trained in off-line stage using the descriptors from training data. The goal of the visual codebook is to represent the distribution of descriptors in the descriptor space. From the amount of existing approaches, the presented system uses visual vocabulary based on *k-mean* clustering with *kd-tree* search structure and *soft-assignment* schema [Beran et al., 2010a]. The visual codebook and translation procedure has the following setting:

- 128 dimensional descriptors space,
- 2k codebook size,
- 4 nearest neighbors in soft-assignment using exponential function [Philbin et al., 2008],
- standard TF-IDF weighting schema using logarithmic function [Manning and Hinrich, 2008].

The bag-of-words of the key-frames are efficiently stored in database (PostgreSQL) and Generalized Inverted (document) Index [PostgreSQL, 2008] is used to speed up the queries. The *cosine distance* [Manning and Hinrich, 2008] is used as the similarity metric for bag-of-word comparison.

Given the key-frame and its list of descriptors, the bag-of-word is computed for the key-frame and used to query the database. The result is the list of most similar key-frames based on bag-of-word comparison (no geometrical information employed).

Copy candidate verification

Having the list of candidate key-frames from the reference dataset for each key-frame from the query video, first, the block segments are constructed from adjacent query key-frames referencing to the similar video. Then each candidate reference segment (reference video segment) of the each query segment (query video segment) is analyzed based on *Frame Similarity Error* and *Video Continuity Error*.

Both segments are represented (besides other characteristics) by the major key-frame. The location alignment method is motivated by results from on-line video synchronization approach [Beran et al., 2010b]. The procedure of copy verification and copy location alignment is as follows:

- **Search for the nearest cuts** – based on *Frame Similarity Error* (*Inliers Ratio Error > 0.5* and *Geometry Error == 1.0*), find the cuts in the both videos,
- **Compare the cuts** – use adjacent frames to analyze the cut type (left, right or both) and compare their content, when the cuts (reference and query) are not similar, stop the procedure

- **Search for the segment margins** – while the video contents are similar (reference and query), enlarge the segments
- **Join adjacent or close segments** – when close query segments have the same reference segments, there are joined to one segment

The procedure is showed also on Figure 3.

When analyzing the video content, the **progressive sampling** of video frames is mostly used. The *progressive sampling* means, that the sampling rate is changed during the video processing according to measured value; e.g. when searching for the video-cut, the sampling rate is 50 while adjacent frames are similar, then the rate is recomputed (e.g. 5) and the last video-part is analyzed again.

Each query segment might reference to ~2000 candidates, but most of them are refused at the very beginning of the verification procedure. The reference candidates are then sorted according to accumulated characteristics (scores and errors) and reported.

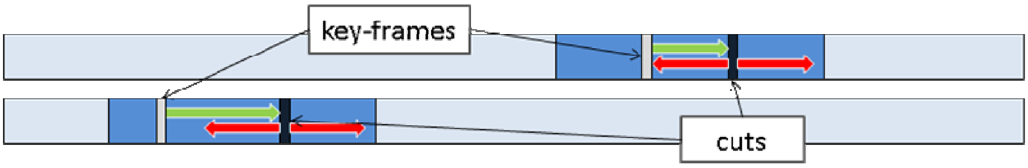

Figure 3 Visualization of the copy verification and copy location alignment procedure.

Results

The experiments with the system reveal two major parts of interest: *visual codebook setting* correlating to index performance and *video-content description* approach.

Visual codebook setting seems to be the crucial for index performance. The setting used in the presented system (especially codebook size and amount of assignment words) caused that each key-frame in the database contains almost 90% of visual words, so using inverted-file index has no effect.

Video-content description method was taken from image retrieval system with no adaptation to video (temporal) data. Also, the noted video transformations were not particularly studied, so the frame content analysis (feature extraction and description) did not reflect the possible transformations at all.

References

[Bay et al., 2006] Bay, H., Tuytelaars, T., and Gool, L. V. Surf: Speeded up robust features. In In ECCV, pp. 404-417, 2006.

[Beran et al., 2010a] Beran Vítězslav, Zemčík Pavel: Visual Codebooks Survey for Video On-line Processing, In: Computer Vision and Graphics: Proc. ICCVG 2010, Warsaw, PL, Springer, p. 10, 2010.

[Beran et al., 2010b] Beran Vítězslav, Zemčík Pavel, Herout Adam: On-line Video Synchronization Based on Visual Vocabularies, In: International Journal of Signal and Image Processing, Vol. 2010, No. 2, TN, p. 7, ISSN 1737-9253, 2010.

[Manning and Hinrich, 2008] Manning Christopher D., R. P., and Hinrich, S.: Introduction to Infor-mation Retrieval. Cambridge University Press, 2008.

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

[PostgreSQL, 2008] PostgreSQL Global Development Group: PostgreSQL 8.3 Documentation: GIN Indexes. 2008. http://www.postgresql.org/docs/8.3/static/gin.html.

[Sivic and Zisserman, 2003] Sivic, J., and Zisserman, A.: Video google: A text retrieval approach to object matching in videos. In ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision (Washington, DC, USA, 2003), vol. 2, IEEE Computer Society, pp. 1470-1477, 2003.

[Chmelar et al., 2009] Petr Chmelar et al.: Brno University of Technology at TRECVid 2009. TRECVID 2009: Participant Notebook Papers and Slides. National Institute of Standards and Technology, Gaithersburg, MD, US, 2009.

[Lowe, 2004] Lowe, David G.: Distinctive Image Features from Scale-Invariant Keypoints. In Int. J. Comput. Vision, vol 60, No. 2, pp. 91-110, 2004.

[Koen et al., 2010] Koen E. A. van de Sande, Theo Gevers and Cees G. M. Snoek: Evaluating Color Descriptors for Object and Scene Recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence (in press), 2010.

[Gemert et al., 2010] Jan C. van Gemert, Cor J. Veenman, Arnold W.M. Smeulders, Jan-Mark Geusebroek: Visual Word Ambiguity, PAMI, pp. 1271-1283, July, 2010.

[Lazebnik et al., 2006] Lazebnik, S., Schmid, C., Ponce, J.: Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories, Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on , vol.2, no., pp. 2169- 2178, 2006.

[Fan et al. 2008] R.-E. Fan et al.: LIBLINEAR: A library for large linear classification. Journal of Machine Learning Research 9(2008), pp. 1871-1874, 2008.

[Snoek et al., 2009] C.G.M. Snoek et al.: The MediaMill TRECVID 2009 Semantic Video Search Engine. TRECVID 2009: Participant Notebook Papers and Slides. National Institute of Standards and Technology, Gaithersburg, MD, US, 2009.

[Mikolajczyk and Schmid, 2004] Mikolajczyk, K. and Schmid, C.: Scale & affine invariant interest point detectors. International Journal on Computer Vision 60(1):63-86, 2004.

[Laptev and Lindeberg, 2003] I. Laptev and T. Lindeberg: Space-time interest points. In ICCV, 2003.

[Willems et al., 2008] G. Willems et al.: An efficient dense and scale-in variant spatio-temporal interest point detector. In ECCV, 2008

[van de Sande et al., 2010] Koen E. A. van de Sande, Theo Gevers and Cees G. M. Snoek: Evaluating Color Descriptors for Object and Scene Recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence, volume 32 (9), pages 1582-1596, 2010.