Research Collection School Of Computing and Information Systems

School of Computing and Information Systems

7-2006

Keyframe retrieval by keypoints: Can point-to-point matching help?

Wanlei ZHAO
City University of Hong Kong

Yu-Gang JIANG
City University of Hong Kong

Chong-Wah NGO
Singapore Management University, cwngo@smu.edu.sg

Citation

Keyframe Retrieval by Keypoints: Can Point-to-Point Matching Help?

Wanlei Zhao, Yu-Gang Jiang, and Chong-Wah Ngo

Department of Computer Science
City University of Hong Kong, Kowloon, Hong Kong
{wzhao2, yjiang, cwngo}@cs.cityu.edu.hk

Abstract. Bag-of-words representation with visual keypoints has recently emerged as an attractive approach for video search. In this paper, we study the degree of improvement when point-to-point (P2P) constraint is imposed on the bag-of-words. We conduct investigation on two tasks: near-duplicate keyframe (NDK) retrieval, and high-level concept classification, covering parts of TRECVID 2003 and 2005 datasets. In P2P matching, we propose a one-to-one symmetric keypoint matching strategy to diminish the noise effect during keyframe comparison. In addition, a new multi-dimensional index structure is proposed to speed up the matching process with keypoint filtering. Through experiments, we demonstrate that P2P constraint can significantly boost the performance of NDK retrieval, while showing competitive accuracy in concept classification of broadcast domain.

1 Introduction

Keyframe based retrieval is one of the earliest studied topics in video search. In large video corpus, keyframe retrieval can aid in the threading of stories with similar content [1], and the tracking of shots with similar concepts [2]. Two recent related efforts are the retrieval of near-duplicate keyframes [3] and the extraction of high-level features in TRECVID [4]. To date, retrieving keyframes with region or object-of-interest is still challenging since video frames are more easily affected by various factors. These factors include the variations in lighting and viewpoint, the artifacts due to motion-blur and compression, and the presence of background clutter.

Recently, retrieval with local keypoints emerges as a promising approach for the aforementioned problems [5]. Keypoints are salient regions detected over image scales and their descriptors are invariant to certain transformations exist in different images. In [6], Sivic & Zisserman show the effectiveness of keypoints for object matching and mining in movies. In [7], Ke & Sukthankar demonstrate the advantage of keypoints over global features such as color histogram in retrieving near-duplicate high-resolution art images. Nevertheless, the number of keypoints in a keyframe can range from a few up to several thousands. The matching of keypoints in two keyframes can consume a significant amount of time which makes efficient on-line retrieval intractable. To tackle this problem,

the offline quantization of keypoints is adopted in [6] where a visual dictionary is constructed. Each keyframe is indexed with a vector of keypoints. Comparison of two keyframes can be as simple as the dot product of two vectors. In contrast to [6], [7] speeds up the search of nearest keypoints with locality sensitive hashing (LSH). Intuitively, [7] is more effective than [6] due to the engagement of point-to-point matching during search. However, [7] is still slower despite the fact that the large amount of keypoints can be pruned with LSH.

This paper investigates the role of point-to-point (P2P) matching in keypoint-based retrieval. We study this topic for two problems: (i) near-duplicate keyframe (NDK) retrieval, and (ii) high-level concept classification, by contrasting the performances of with and without P2P matching. We examine *when* and *how* the P2P matching can boost the performance of these two tasks. In P2P, we propose a new index structure, namely LIP-IS, for fast matching and effective filtering. Under our investigation, LSH is not effective for filtering in noisy environment. As demonstrated in our experiments with TRECVID datasets, LSH indeed deteriorates the performance of matching because the chance of returning nearest neighbors is practically not high. In non-P2P matching, as in [6], we generate a dictionary of visual keypoints by clustering. Each keyframe is represented as a point in the vector space model of dictionary. Thus, the nearest neighbor search is not an issue in this strategy. The potential risk, however, is the difficulty in determining the number of clusters during quantization. In addition, the co-occurrence and distribution of keypoints are inherently neglected.

The usefulness of point-to-point matching is also studied in [8] for three datasets with scenes (from the sitcom *Friends*), objects (from ETH-80) and textures (from VisTex) respectively. Empirically, [8] shows that P2P matching is useful for retrieving keyframes with similar scenes, but not always so for objects and textures. The success of P2P matching also largely depends on the underlying matching strategy. In [8], the embedded Earth Mover's Distance (eEMD) with LSH filtering support is employed. Basically eEMD projects keypoints to a high dimensional feature space, and LSH utilizes this sparseness property to increase the chance of finding the nearest neighbors in high speed. The eEMD belongs to multi-point matching technique, and is susceptible to noise if no proper mechanism is used to constrain the flow of mass from one keypoint to the other [9]. Under the presence of background clutter, the matching can become random simply to meet the goal of minimizing the amount of efforts in transforming one signature to another. For robustness consideration, we use a relatively simple one-to-one symmetric (OOS) keypoints matching strategy to reduce as many false matches as possible. Empirically we find that OOS outperforms the nearest neighbor search with many-to-one matching strategy [7].

The remainder of this paper is organized as follows. Section 2 compares different keypoint detectors and their descriptors. Section 3 details the proposed one-to-one symmetric matching algorithm and its filtering support. Section 4 describes the construction of visual dictionary with keypoints. Section 5 and Section 6 present the experimental results, and Section 7 concludes our findings.

2   Keypoint Detectors and Descriptors

There are numerous keypoint detectors and descriptors in the literature. A good survey of these works can be found in [10], [11]. The detectors basically locate stable keypoints (and their support regions) which are invariant to certain variations introduced by geometric and photometric changes. Popular detectors include Harris-Affine [10], Hessian-Affine [10], Difference of Gaussian (DoG) [12], and Maximal Stable Extreme Region (MSER) [13]. Harris-Affine, which is derived from Harris-Laplace, estimates the affine neighborhood by the affine adaptation process based on the second moment matrix. Keypoints of Hessian-Laplace are points which reach the local maxima of Hessian determinant in space and fall into the local maxima of Laplacian-of-Gaussian in scale. DoG uses the similar method as Hessian-Laplace to localize the keypoint at local space-scale maxima of the difference of Gaussian. MSER is detected by a watershed like process and is invariant to affine transformations and robust to viewpoint changes.

In [14], SIFT (scale-invariant feature transform) has shown to be one of the best descriptors for keypoints. SIFT is a 128-dimensional feature vector that captures the spatial structure and the local orientation distribution of a patch surrounding keypoints. PCA-SIFT, proposed in [15], is a compact version of SIFT with principal component analysis. In this paper, we adopt the 36-dimensional PCA-SIFT as the descriptors of keypoints due to its compactness and retrieval effectiveness, as indicated in [15]. Based on PCA-SIFT, we use Cosine similarity to measure the closeness of two keypoints.

3   Keypoint Matching and Filtering

3.1   One-to-One Symmetric Keypoint Matching

Given two sets of keypoints respectively from two keyframes, the alignment between them can be solved with bipartite graph matching algorithms. Depending on the mapping constraint being imposed, we can categorize them as many-to-many (M2M), many-to-one (M2O), one-to-many (O2M) and one-to-one (O2O) matching. The factors that affect the choice of matching strategy include noise tolerance, similarity measure, matching effectiveness and efficiency. In videos, frames are always suffering from low-resolution, motion-blur and compression artifact. Noise becomes a crucial factor in selecting matching algorithm, particularly when the matching decision is made upon a small local patch surrounding keypoints. Noise can affect the performance of keypoint detectors [12]. The localization errors caused by detectors can deteriorate the distinctiveness of PCA-SIFT. It becomes very common that a keypoints fails to find its corresponding neighbor in another keyframe, and on the other extreme, a keypoint can simply match to many other keypoints due to mapping ambiguity. In principle, to suppress faulty matches, O2O matching appears to be noise tolerant although some correct matches may be missed.

For effective keyframe retrieval, in our opinion, the false matches should be filtered off as many as possible. To retain only the most reliable matches for

retrieval, we introduce a new scheme – namely one-to-one symmetric (OOS) matching. OOS ensures all the matches are the nearest neighbors. The symmetric property is also emphasized so that if keypoint P matches to Q, then P is the nearest neighbor of Q (i.e., $P \rightarrow Q$) and similarly $P \leftarrow Q$. This property indeed makes OOS stable and unique, i.e., the result of matching a keypoint set A to set B is exactly the same as B to A, unless there are keypoints that have more than one nearest neighbor. Generally speaking, O2O matching cannot guarantee each matched keypoints pair to be meaningful. Some false matches indeed could exist with high similarity value. But it becomes a rare case for these false matches to be symmetrically stable and paired to each other in both directions.

3.2 Fast Keypoint Filtering

Point-by-point matching between two sets is generally a time consuming task especially when the set cardinality is high. To allow fast retrieval of OOS, we perform approximate nearest neighbor search by indexing PCA-SIFT descriptors in a multi-dimensional structure called LIP-IS. The structure is a group of 36 histograms formed independently by every components of PCA-SIFT. LIP-IS is constructed by equally and independently quantizing each histogram into 8 bins, with a resolution of $\Delta = 0.25$ (the range of a PCA-SIFT components is [-1,1]). Given $P = [p_1, p_2, ..., p_i, ..., p_{36}]$, the index of P in dimension i is defined as

$$\mathcal{H}(p_i) = \lfloor \frac{p_i + 1}{\Delta} \rfloor \tag{1}$$

Totally, this index structure is composed of 8×36 bins. During indexing, a keypoint P is repeatedly indexed into the corresponding bins of 36 histograms, according to its quantized value in particular dimension. Thus, each keypoint is hashed and then distributed into 36 bins in this structure. In principle, the structure encodes the keypoints of a keyframe by decomposing the PCA-SIFT components and modeling them as 36 independent distributions. This structure is intuitive and reasonable since the PCA-SIFT components are orthogonal to each other. Based on this structure, we define the function that any two keypoints P and Q collide in dimension i if

$$\mathcal{C}(q_i, p_i) = \begin{cases} 1 \text{ if } |\mathcal{H}(q_i) - \mathcal{H}(p_i)| \leq 1 \\ 0 \text{ Otherwise} \end{cases} \tag{2}$$

When searching for the nearest neighbor of a query keypoint Q, the structure will return a candidate set A(Q) which includes the points collide with Q across all the 36 dimensions. Then we search for Q's nearest neighbor from the set A(Q) by OOS matching algorithm. This structure has the merit that it is efficient and easy to implement with simple bit operation.

With LIP-IS, basically two keypoints which are similar to each other are more likely to collide in every dimension. And in contrast, the dissimilar keypoints have a relatively lower chance of collision. Since each component of PCA-SIFT descriptors is theoretically Gaussian distributed, the probability that any two

keypoints collide in a dimension can be estimated. The probability that a keypoint Q will collide with P in i dimension, in its best ($\mathbf{P}_b$) and worst ($\mathbf{P}_w$) cases, can be estimated as follows

$$\mathbf{P}_b = 2 \int_0^{2\Delta} \frac{1}{\sqrt{2\pi}\sigma_i} \exp\{-\frac{q_i^2}{4\sigma_i^2}\} \, dq_i \tag{3}$$

$$\mathbf{P}_w = 2 \int_0^{\Delta} \frac{1}{\sqrt{2\pi}\sigma_i} \exp\{-\frac{q_i^2}{4\sigma_i^2}\} \, dq_i \tag{4}$$

Then, the probability that a point will collide with Q in 36 dimensions can be expressed as

$$\mathbf{P}_f = \mathbf{P}_b^{36} \tag{5}$$

Notice that $\mathbf{P}_f \ll 1$ in general. This also implies that the cardinality of $A(Q)$ can be very small, i.e., $\mathbf{P}_f \times n$, where n is the total number of keypoints to be searched. The probability of missing the nearest neighbor can also be estimated. Suppose M is the maximum number of dimensions that the nearest neighbor $\hat{P}$ and Q will not collide, the worst case probability is

$$\mathbf{P}_{miss} = \sum_{i=1}^{M} \binom{M}{i} \mathbf{P}_w^{M-i} (1 - \mathbf{P}_w)^i = 1 - \mathbf{P}_w^M \tag{6}$$

In theory, M can be estimated (and this value is much smaller than 36), if we set a threshold to exclude keypoints with low similarity from consideration. In our simulation, when searching for a nearest neighbor from a 1000 keypoint set, LIP-IS is often capable of filtering 99.5% of the points without missing the real candidate for OOS matching.

4   Visual Keywords Generation

We generate a visual dictionary based on [6]. We select approximately 900 keyframes from TRECVID-2005 development set, with about 70% of them containing the 39 high-level concepts specified in TRECVID [4]. In total, there are about 490,000 keypoints extracted. Empirically we quantize these keypoints into 5,000 clusters, and each cluster represents a visual keyword. Instead of employing K-means for clustering, we adopt a faster clustering algorithm based on the recent work in [16]. With this visual dictionary, the classical *tf-idf* is used to weight the importance of keywords. A keyframe is then represented as a vector of keywords, analogous to the traditional text-based vector space model.

5   Experiment-I: Near-Duplicate Keyframe Retrieval

We conduct experiments on the near-duplicate keyframe (NDK) dataset given by [3]. The dataset contains 150 NDK pairs and 300 non-NDKs selected from

TRECVID 2003 video corpus. We evaluate the retrieval performance with the probability of the successful top-k retrieval, defined as

$$\mathbf{P}(k) = \frac{Q_c}{Q_t}, \tag{7}$$

where Q_c is the number of queries that find its duplicate in the top k list, and Q_t is the total number of queries. In this experiment, we use all NDKs (300 keyframes) as queries. The ranking is based on the cardinality of keypoints being matched. In case the cardinality is the same, the average similarity of matched keypoints is further used. In the dataset, the number of keypoints per keyframe varies depending on the detectors and keyframe content. DoG, on average, detects 1000 keypoints per keyframe. In contrast, Harris-Affine, Hessian-Affine and MSER detectors only extract few hundreds of keypoints respectively.

5.1   Keypoint Comparison

We first compare the retrieval effectiveness of different keypoints under the one-to-one symmetric matching scheme. Figure 1(a) shows the performance of different detectors, and (b) shows the performance when re-ranking the top-k lists of two detectors with equal weight. Overall, DoG achieves the best performance, followed by Hessian Affine. Both detectors indeed win a large margin over Harris Affine as well as MSER. This is mainly because both DoG and Hessian Affine are capable of detecting more distinctive keypoints than Harris Affine and MSER. These two detectors are more reliable in the presence of partial occlusion and background clutter. Since there is no detector alone that is robust enough to against all kinds of transformations, we also attempt the fusion of two detectors by re-ranking their retrieved lists, as shown in Figure 1(b). We find that the combination of DoG with Hessian Affine detectors performs the best. However, this comes with the cost of time since the speed is basically double of the DoG and Hessian Affine alone. For this reason, we only use DoG for testing in the remaining experiments.
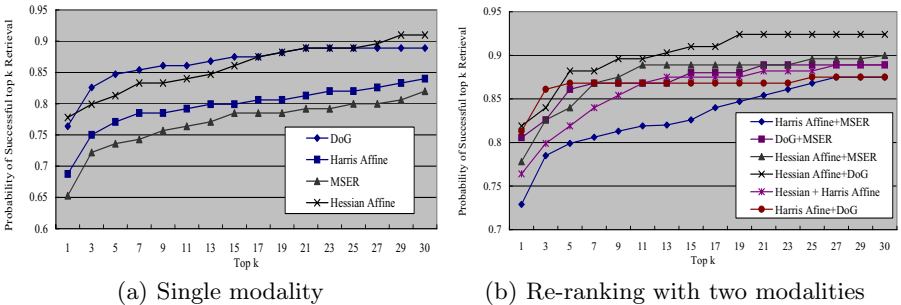

(a) Single modality   (b) Re-ranking with two modalities

Fig. 1. Comparison of different keypoint detectors

5.2   Effectiveness of Point-to-Point Matching

We mainly compare the effectiveness of OOS matching (OOS) and visual key-words (VK). In addition, we contrast their performances with many-to-one (M2O) matching based on the nearest neighbor search in [7], block-based color moment (CM), and global color histogram (CH). For CM, we use the first three color moments extracted in *Lab* color space over 5×5 grid partitions. In CH, we use HSV color space with 64 bins of H, 8 bins of S, and 4 bins of V.

Figure 2(a) shows the comparison of five different approaches. OOS outper-forms all other methods across all k (from 1 to 30) being tested. Moreover, the strategies based on point-to-point matching (OOS and M2O) significantly out-performs others. VK performs poorly and shows lower $P(k)$ than the baselines CM and CH. In VK, we find that although the small patches in a cluster tend to have high similarity value, they actually appear differently based on human perception. This may due to the problems of polysemy and synonymy as previ-ously mentioned by [17] when constructing the visual dictionary. Our dictionary is generated based on TRECVID 2005 corpus, it may have certain impact when applying to 2003 corpus. In addition, the selection of clustering parameters (e.g. number of clusters) can also affect the final results.
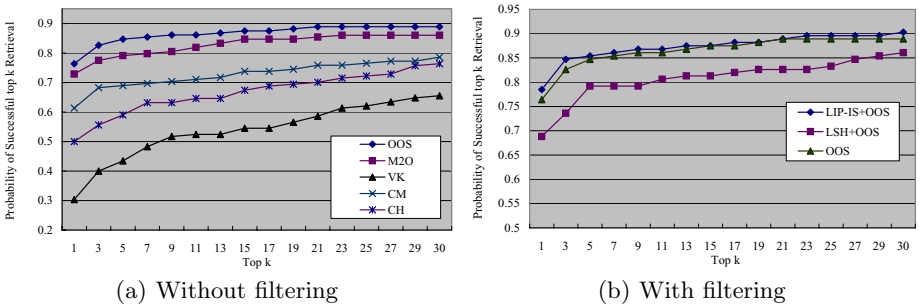

(a) Without filtering

(b) With filtering

Fig. 2. Performance Comparison of different retrieval techniques

Comparing the point-to-point matching strategies, the experiment shows that one-to-one symmetric (OOS) is constantly better than many-to-one (M2O). Based on the recent results presented in [3] where $P(k) \approx 0.6$ when $k = 1$ and $P(k) \approx 0.76$ when $k = 30$, both OOS and O2M demonstrate considerably better performance. For filtering support, we also compare the proposed LIP-IS and locality sensitive hashing (LSH), as shown in Figure 2(b). For LSH, we man-ually optimize the parameters by setting the K (number of random partition) and L (number of times to tessellate a set) with 108 and 2 respectively. Although filtering with LSH can be nearly 2.5 times faster than LIP-IS in our experiment, its performance is relatively poor since the nearest neighbors are not always re-turned with LSH. In contrast, our proposed scheme (LIP-IS+OOS) shows nearly the same retrieval performance with pure OOS.

5.3   Speed Efficiency

With large amount of keypoints in keyframes, speed becomes a critical concern for online retrieval. Table 1 shows the average time for comparing two keyframes with different features. All the approaches are tested on a Pentium-4 machine with 3G Hz CPU and 512M main memory in Windows-XP environment. LIP-IS is able to speed up OOS matching by 12.5 times. Apparently, the methods without point-to-point matching such as VK are faster since the comparison only requires the manipulation of two feature vectors.

Table 1. Speed efficiency for comparing two keyframes

Method	LIP-IS+OOS	OOS	VK	O2M	CM	CH
Time (s)	0.028	0.35	0.93×10^{-4}	0.34	10^{-5}	10^{-5}

6   Experiment-II: High-Level Concept Classification

In this experiment, we compare OOS and VK in classifying keyframes according to high-level concepts. We construct a dataset of four concepts containing 5589 keyframes extracted from the TRECVID 2005 common development feature annotation (by CMU and IBM tools) and test sets. The dataset is composed of 901 keyframes with US-flag, 2816 keyframes with maps, 910 keyframes with computer/TV screen, and 962 keyframes with waterscape/waterfront. We manually select 430 keyframes covering the four concepts for training, and leave the remaining keyframes as testing set. Figure 3 shows some samples with water and US-flag concepts. The targeted concepts in keyframes appear in varying forms in terms of lighting, viewpoint, color and scale changes. Some concepts are partially occluded and present in background clutter.

We train two types of classifiers for testing: k-NN and support vector machines (SVM). For OOS matching, we perform 1-NN classification. The similarity is based on the cardinality of matched keypoints found in a comparison. For visual keywords (VK), we construct 1, 3-NN classifiers, and a multi-class SVM. We use CCR (correct classification rate) to evaluate the accuracy of classification:

$$\text{CCR} = \frac{\text{number of correctly classified keyframes in class } i}{\text{number of keyframes in class } i}. \tag{8}$$

Fig. 3. Keyframes from two semantic concepts

Table 2. Classification rate of OOS and variant of VK classifiers

Methods	US-Flag	Maps	Screen	Water
OOS (1-NN)	0.645	0.810	0.723	0.790
VK (1-NN)	0.441	0.603	0.441	0.570
VK (3-NN)	0.347	0.567	0.244	0.475
VK (SVM)	0.445	0.527	0.126	0.723

Table 2 shows the classification performance of OOS and VK. Overall, point-to-point matching outperforms all runs based on VK. However, when we repeat 3-NN for OOS, surprisingly the results are less satisfactory. We investigate the results and find that indeed there are many duplicate keyframe pairs with common concepts in the broadcast videos. With 1-NN, OOS has an excellent success rate of classifying a keyframe if its near-duplicate version is also found in the training set. However, when the number of near-duplicate samples is less than k, k-NN does not perform well. This probably concludes that OOS is effective in finding near-duplicate keyframes, rather than the targeted concepts appeared in varying forms. Compared to OOS, VK is poorer partially because the co-occurrence of keypoints in a concept is not fully exploited under this representation. It is susceptible to noise due to the problems of polysemy and synonymy. Moreover, due to the limited amount of training samples, the CCR of 1-NN indeed works better than 3-NN and SVM classifiers.

7   Discussion and Conclusion

We have presented the proposed point-to-point (P2P) matching algorithm and compared its performance with visual keyword (VK) generation. Overall, P2P matching significantly outperforms VK in both tasks we investigate. VK, despite its simplicity and efficiency, has the deficiencies that it neglects the co-occurrence and distribution of keypoints, is vulnerable to the potential risks of polysemy and synonymy, and could be sensitive to the setting of parameters during clustering. P2P matching, in contrast, is relatively slower but much stable due to the fact that constraints can be easily imposed during matching. In addition, variant matching strategies exhibit different retrieval effectiveness in our experiments, however, overall they demonstrate the advantage of matching over VK. In NDK retrieval, our proposed OOS matching shows superior performance over all other methods. When LIP-IS is used for filtering support, OOS still performs consistently better with more than 10 times of speed up. In concept classification, OOS matching is still better than VK with k-NN and SVM classifiers. Nevertheless, we only conclude that OOS is effective in broadcast domain where the near-duplicate version of keyframes has a higher chance to be found in both testing and training samples. Other than that, we have no enough empirical evidence to support that P2P is a winner over VK. Basically, when the targeted concepts appear in quite different scales under background clutter, only few matches can be found with P2P. With these few matches, it becomes difficult to distinguish keyframes of different concepts.

Acknowledgements

References

1. Wu, X., Ngo, C.-W., Li, Q.: Threading and Autodocumenting News Videos. IEEE Signal Processing Magazine. **23** no.2 (2006) 59–68
2. Chang, S.-F. et. al: Columbia University TRECVID-2005 Video Search and High-Level Feature Extraction. TRECVID Online Proceedings. (2005)
3. Zhang, D.-Q., Chang, S.-F.: Detecting Image Near-Duplicate by Stochastic Attributed Relational Graph Matching with Learning. ACM International Conference on Multimedia. (2004) 877–884
4. TREC Video Retrieval Evaluation. In *http://www-nlpir.nist.gov/projects/trecvid/*.
5. Csurka, G., Dance, C., Fan, L. et. al: Visual Categorization with Bags of Keypoints. ECCV2004 Workshop on Statistical Learning in Computer Vision. (2004) 59–74
6. Sivic, J., Zisserman, A.: Video Google: A Text Retrieval Approach to Object Matching in Videos. International Conference on Computer Vision. (2003) 1470–1477
7. Ke, Y., Suthankar, R., Huston L.: Efficient Near-Duplicate Detection and Sub-image Retrieval. ACM International Conference on Multimedia. (2004) 869–876
8. Grauman, K., Darrell, T.: Efficient Image Matching with Distributions of Local Invariant Features. Computer Vision and Pattern Recognition. (2005) 627–634
9. Rubner, Y., Tomasi, C., Guibas, L.J.: The Earth Mover's Distance as a Metric for Image Retrieval. International Journal of Computer Vision. **40** (2000) 99–121
10. Mikolajczyk, K., Schmid, C.: Scale and Affine Invariant Interest Point Detectors. International Journal of Computer Vision. **60** (2004) 63–86
11. Mikolajczyk., K., Tuytelaars, T., Schmid, C. et. al: A Comparison of Affine Region Detectors. International Journal on Computer Vision **65** no.1-2 (2005) 43–72
12. Lowe, D.: Distinctive Image Features from Scale-Invariant Key Points. International Journal of Computer Vision. **60** (2004) 91–110
13. Matas, J., Chum O., Urban, M. et. al: Robust Wide Baseline Stereo from Maximally Stable Extremal Regions. British Machine Vision Conference. (2002) 384–393
14. Mikolajczyk, K., Schmid, C.: A Performance Evaluation of Local Descriptors. Computer Vision and Pattern Recognition. (2003) 257–263
15. Ke, Y., Sukthankar, R.: PCA-SIFT: A More Distinctive Representation for Local Image Descriptors. Computer Vision and Pattern Recognition. **2** (2004) 506–513
16. Zhao, Y., Karypis, G.: Empirical and Theoretical Comparisons of Selected Criterion Functions for Document Clustering. Machine Learning. **55** (2004) 311–331
17. Quelhas, P., Monay, F., et al.: Modeling Scenes with Local Descriptors and Latent Aspects. International Conference on Computer Vision. (2005) 883–890