Visual Codebooks Survey for Video On-line Processing

Vítězslav Beran and Pavel Zemčík

Department of Computer Graphics and Multimedia
Faculty of Information Technology
Brno University of Technology
Božetěchova 2, 612 66 Brno, CZ
{beranv, zemcik}@fit.vutbr.cz

Abstract. This paper explores techniques in the pipeline of image description based on visual codebooks suitable for video on-line processing. The pipeline components are *(i)* extraction and description of local image features, *(ii)* translation of each high-dimensional feature descriptor to several most appropriate visual words selected from the discrete codebook and *(iii)* combination of visual words into bag-of-words using hard or soft assignment weighting scheme. For each component, several state-of-the-art techniques are analyzed and discussed and their usability for video on-line processing is addressed. The experiments are evaluated on the standard Kentucky and Oxford building datasets using image retrieval framework. The results show the impact loosing the pipeline precision in the price of improving the time cost which is crucial for real-time video processing.

1 INTRODUCTION

The main interest of this work is the exploration of image retrieval techniques based on visual codebooks for their utilization for video on-line processing. The image retrieval frameworks based on visual codebooks are mainly composed from several components. The local visual features are extracted from the images and described using high-dimensional descriptors. Having pre-trained discrete visual codebook, the high-dimensional descriptors are translated to several most appropriate *visual words*. The images are then represented as the distribution of these visual words that is called bag-of-words.

Described approach is inspired by text retrieval systems. One of the first work introducing visual codebooks was *Video Google* [1]. The work utilized widely used SIFT transformation [2] as local features. The visual vocabulary training was based on naive k-means algorithm. When introduced, the visual codebooks were used in image and object retrieval applications. Similarly to text retrieval, the database of images is indexed by inverted file approach. Search for a query image in database results in immediate returning of a ranked list of documents (key frames, shots, etc.) in the manner of search in text documents.

Later works utilized more types of local image features such as corner-like detector with full affine adaptation or detection of stable regions. When the local features are finally described, the performance of the visual vocabulary approach correlates to performance of particular image feature extraction technique. Needs for large, more discriminative vocabularies lead the research to find faster clustering methods. Two significant methods were developed: hierarchical k-means [3] and approximated k-means [4]. The developed methods allow creation of vocabularies with size about 1M of visual words with reasonable time and computational cost.

The later research experimenting with different translation schemas when translating the local image features into bag-of-words showed that the discriminative power of the vocabulary could be improved not only by extending the vocabulary size. Instead of the standard approach, where one local feature is translated in just one visual word, a single image feature could be assigned to several visual words. The approach is known as *soft-weighting*.

The existing works mainly explore the solutions for their best *precision* performance. The work [5] compares several detectors of local image features and evaluate their performance on visual codebooks with different sizes. The explored detectors are time expensive and the codebook sizes are only up to 10k. Other solution [6] also offers the results of comparison of visual codebooks with different size (up to 1M), but also uses the time expensive image feature detector solution. The work [7] is focused on time effectiveness. Standard SIFT and SURF feature extractors are optimized for dense sampling and descriptors dimensionality is reduced by PCA.

This paper analyzes the state-of-the-art techniques used in each pipeline component in the utilization in video on-line processing point of view. Next Section contains an overview of local image feature extraction and description methods explaining their crucial attributes for real-time systems. The process of building the visual codebook is described in Section 3 together with codebook searching methods. The possibilities of visual words weighting when bag-of-words are constructed are discussed in Section 4. Sections 5 and 6 describe the experiments and discuss the results.

2 IMAGE LOCAL FEATURES

Real-time applications such as on-line video synchronization introduce specific demands to the commonly used techniques. The attributes of image local feature extraction methods are stability, repeatability and robustness to several types of transformations or distortions [8]. The characteristic of feature descriptor is its discriminative power. Usually, the more powerful the feature extraction and description methods are, the higher is their time cost. For the real-time applications dealing with consecutive video-frame processing, the methods performance could be decreased at the expense of execution time increase. The computational cost demands also deriving the size of the visual vocabulary. Detected local image features 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.

Widely used method is SIFT [2]. The method is sensitive to blob-like structure and is invariant to scale and orientation. Besides the local feature detection, SIFT also describes the local regions using histogram of gradients. The SIFT preserves the gradient location information by dividing the region into regular grid of 4x4 subregions. SIFT is often used only as a descriptor for different image local feature detectors. The blob-like structures can be also detected using Hessian matrix [8]. The approach based on Hessian matrix can be effectively approximated by block filters (SURF [9]). The SURF detector is based on effective platform computation of Haar-wavelets on integral images. The authors also introduced new descriptors utilizing the same platform. The approach known as *FAST corners* ([10]) employs machine learning to construct a 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 characteristic scale selection and affine adaptation [11] can be applied. Unfortunately, full affine transformation detection significantly slows down the process. Detected corners are described using a gradient distribution of the region around the detected point. Gradient distribution can be described by Histogram of Gradients (*HOG* [12]) or Gradient Location and Orientation Histogram (*GLOH* [13]).

The FAST detector combined with the GLOH or HOG descriptor is not rotational invariant. This paper present some modification of these descriptors to improve their robustness to rotation transformation.

3 VISUAL CODEBOOKS

The idea of visual vocabulary, firstly used in *Video Google* by [1], 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 built during the training stage. A part of the data (training data) is used to divide the descriptor space into clusters. Each cluster is labeled; has its own identification number. The vocabulary is then the list of cluster centers and identifiers. The clustering procedure based on k-mean algorithm contains the search step, when the sample should be assigned to the nearest. The later research introduced several solutions to avoid time consuming naive sequential search. Figure 1 schematically shows the different approaches described in detail below. When the size of the resulting vocabulary is small ($k < 10^3$), the *naive k-means* algorithm can be used (Fig. 1(a)). 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 [4]) need more dis-

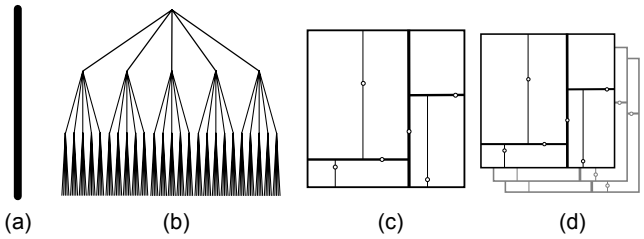

Fig. 1. Clustering strategies - (a) naive sequential, (b) hierarchical, (c) kd-tree and (d) random forest.

criminative vocabulary. One possible way how to reduce the time complexity is using *Hierarchical k-means* [3] (Fig. 1(b)). 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 Hierarchical k-means 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 kd-tree (Fig. 1(c)) or by random forest of kd-trees (Fig. 1(d)). This approach is called *Aproximate k-means* [4]. The quantization error after clustering procedure is expressed as a sum of distances of training samples to their nearest cluster as follows:

$$D = \frac{1}{N}\sum_{i=1}^{N} d(p_i, Q[p_i]) \tag{1}$$

where N is the number of training samples Q is the nearest cluster center to the sample p_i and d is the distance function.

Having visual codebook and the dataset, each visual word appears in different amount of images and also different times in each particular image. Some of the visual words are quite rare in contrary to visual words that appears very often. Usually, 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:

$$tf - idf(w) = tf(w).idf^{log}(w) \tag{2}$$

$$= \frac{|d(w)|}{|d|}.\log(\frac{|D|}{|D(w)|}) \tag{3}$$

where d is a document (image signature), $|d|$ 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 *idf* weighting function emphasizes the rare visual words and down-weights the frequent ones. The rare visual words does not necessarily mean that

they are best informative. We introduce another weighting function idf^{hat} that is defined as follows:

$$idf^{hat}(w) = \exp\left(-\frac{1}{2}\left(\frac{|D|}{|D(w)} - 1\right)^2 c\right) \tag{4}$$

where c is the steepness of the function experimentally evaluated and suggest $c = 9.0$. The hat-like function down-weights both frequent and rare visual words and emphasizes the common ones.

4 VISUAL WORDS ASSIGNMENT

The *bag-of-words* is a collection of weighted visual words representing the image content. The bag-of-words is also known as *image signature*. This collection can be seen also as a vector of visual word frequencies. It degrades to a set-of-words when the weights represent only the word's presence (binary vector). Otherwise, it is a bag-of-words. The Figure 2 shows the process of describing the image content by an image signature.

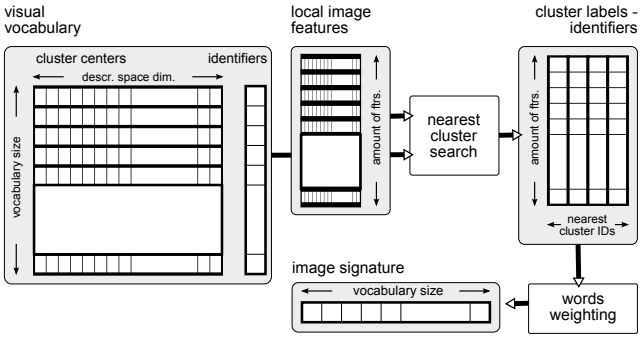

Fig. 2. Image signature extraction procedure.

Having visual codebook, for each descriptor of local features from the image, the k visual words (nearest clusters) are found. The weight for each word is computed and used to increase the value of the image signature at the word's ID position. The image signature then can be seen as a histogram of occurred visual words. The works ([5], [6]) 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. Such approach is called *hard assignment*. The *soft-assignment* (soft-weighting) techniques assign a single descriptor to several visual words nearby in the descriptor space. Given the sorted list of k *nearest* visual words, the weighting functions assign different

weight to the visual word according to its distance or its rank in the list. The weighting functions can be defined as follows:

$$w_i^{ratio} = \frac{d_0}{d_i} \tag{5}$$

$$w_i^{exp} = \exp(-\frac{d_i^2}{2\sigma^2}) \tag{6}$$

$$w_i^{rank} = \frac{1}{2^{i-1}} \tag{7}$$

where i is the rank and d_i is the distance of the ith descriptor point in the list to its closest visual word. The basic weighting function (Eq 5.) is the ratio between the distance of the descriptor point to the closest visual word and distance of the actual point to its closest visual word. The exponential function (Eq 6.) uses σ so that substantial weight is only assigned to a small number of cells. The authors [6] experimentally evaluated and suggest $k = 3; \sigma^2 = 6.250$. The similar idea is realized by rank function (Eq 7.) replacing the distance by the visual word's rank[5].

During the retrieval stage, documents are ranked by their *similarity*. One of the frequently 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. The cosine similarity can be seen as a method of normalizing document length during comparison and is defined as $sim(\mathbf{q}, \mathbf{d}) = \frac{\mathbf{q}.\mathbf{d}}{|\mathbf{q}||\mathbf{d}|}$ where . is dot product and || is the vector magnitude. The cosine similarity of two image signatures will range from 0 meaning independent images to 1 meaning exactly the same images, since the word weights cannot be negative.

5 EXPERIMENTS

The image datasets was used for the presented experiments analyzing the characteristics of various parameters of feature extraction and visual vocabulary building. *Kentucky Dataset* was created as a recognition benchmark at Kentucky University [3]. The set consists of 2550 groups of 4 images each, that is 10200 images in total. The objects images are taken from different angles and rotations. The size of the images is approximately 640x480 pixels. More details about the extracted local features is in Table 1.

Table 1. The number of images and features for each dataset.

dataset	images	SIFT	SURF	FAST HOG	FAST OGH
Kentucky	10200	13.161.824	6.541.246	2.664.175	5.013.715

The experiments evaluating the clustering process based on k-mean algorithm are designed to measure the clustering error (Eq. 1) in each iteration step. The measurement procedure is repeated several times to analyze the influence of the initialization error.

Next set of experiments are design to analyze the relation between the speed and precision performance. The speed is measured and compared for each pipeline component separately. The precision is measured for the entire retrieval pipeline as a *Mean Average Precision* (mAP). Average Precision represents the area under Precision-Recall curve for a query and can be directly computed from the ranked list of retrieved images as:

$$AP = \frac{1}{m} \sum_{i=1}^{n} \frac{relevant(x_i)}{i} \tag{8}$$

where n is the number of retrieved images, m is the number of relevant images, x_i is the i-th image in the ranked list of retrieved images $X = x_1, \ldots, x_n$ and $relevant(x_i)$ returns the number of relevant images in the first i images, only if the x_i is relevant image itself, and 0 otherwise. This measure gives a number in range (0, 1] where a higher number corresponds to a better performance.

Evaluation of different methods for their usability for video on-line processing is done by experimenting with different sizes of codebook, different clustering strategies, various weighting schemes in bag-of-words and different types of image local feature extractors.

6 RESULTS

Clustering error. The k-mean clustering used to create the visual codebook is an iterative process. The method minimizes the error in each step. The clustering error progress was evaluated in SIFT and SURF descriptor space. The resutls in Fig. 3 shows the considerable clustering improvements up to 4 iterations in average. The k-means clustering is known to be sensitive to initialization.

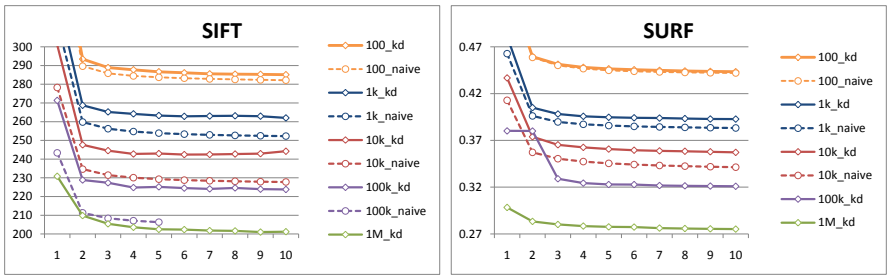

Fig. 3. Clustering error for SIFT and SURF descriptor space.

The experiments with different clustering initialization result in nearly same clustering error. It approves the hypotheses that the samples in the descriptor space do not gather in significant groups but are rather regularly distributed.

Vocabulary size. The previous works approved the assumption, that bigger codebook gives better results. The designed experiments compares the codebooks with size 100, 1k, 10k, 100k and 1M using two baseline image features SIFT and SURF focusing also on the speed of each run. The results in Fig. 4 show that the naive search strategy outperforms the kd-tree search but the time cost is exponentially higher. Depending on the image data type, the precision of bigger codebooks with naive search reach the precision of smaller ones with kd-tree search. The results are evaluated using idf^{log} and hard-assignment.

size	kd-tree		naive		log(speed)	
	SIFT	SURF	SIFT	SURF	kd-tree	naive
100	0.583	0.525	0.591	0.524	1.21	1.52
1k	0.594	0.550	0.637	0.574	1.21	2.54
10k	0.600	0.551	0.659	0.600	1.47	3.69
100k	0.669	0.569	0.756	0.674	1.76	4.64
1M	0.728	0.614			1.78	

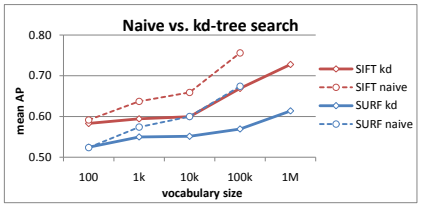

Fig. 4. Performance comparison of the codebooks with different sizes.

Codebook weighting. The proposed weighting function for computing inverse document frequency idf^{hat} was compared to standard approach based on logarithm function idf^{log}. The assumption that the enhancing of the rare visual words negatively influence the codebook precision is wrong. The results in Fig. 5 show that idf^{log} outperforms the idf^{hat} function. The performance for codebook with size 1k is caused clearly by soft-assignment approach. The results are evaluated using SURF descriptor and kd-tree search strategy.

size	knn=1	knn=10			
	Log	Hat			
	Ratio0	Ratio0	Ratio0	Exp0	None0
100	0.525	0.472	0.517	0.509	0.498
1k	0.550	0.481	0.572	0.573	0.554
10k	0.551	0.464	0.506	0.510	0.488
100k	0.569	0.510	0.549	0.550	0.518
1M	0.614	0.556	0.607	0.608	0.590

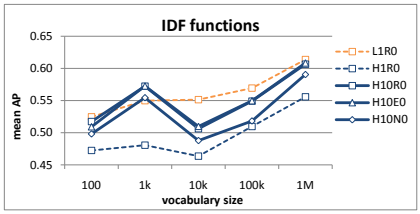

Fig. 5. The number of images and features for each dataset.

Soft assignment. The soft-assignment techniques significantly improves the pipeline performance. Several combinations of weighting functions are explored. The abbreviations used in result tables and graphs means - *None* for no weight-

ing, *Exp* for exponential function and *Ratio* for ratio function. These functions are combined with *Rank* function which is marked by number *2* in the name of the experiment run (when *0*, no ranking is used). The number of used visual words for soft-assignment is in the *knn* (k nearest neighbor) column. The experiments in Fig. 6 revealed that no more than 4 closest visual words to the descriptor significantly improved the overall performance.

knn	None0	Exp0	Exp2	Ratio0	Ratio2
1	0.550	0.550	0.550	0.550	0.550
4	0.584	0.606	0.612	0.608	0.609
10	0.588	0.606	0.615	0.609	0.611

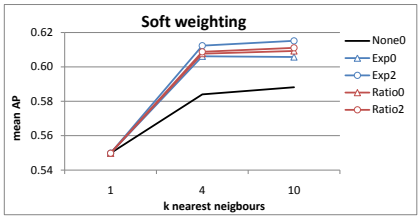

Fig. 6. Soft-assignment functions and their combinations.

Image local features. Four image local feature extracts and descriptors were evaluated in experiments - SIFT, SURF, FAST detector combined with HOG descriptor (FHOG) and FAST detector combined with GLOH descriptor (FGLOH). The GLOH descriptor contain modification for descriptor rotation invariance. The results in Fig. 7 show that the SURF has comparable precision as SIFT but is 3 times faster. The speed results are measured as average time of one feature extraction in 10^{-6} seconds.

size	SIFT	SURF	FHOG	FGLOH
1k	0.587	0.609	0.472	0.457
100k	0.649	0.593	0.465	0.445
speed	1.00	3.28	3.65	8.34

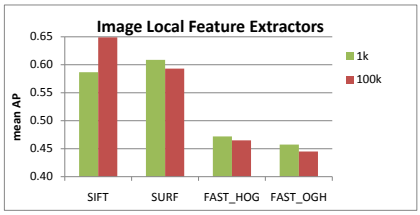

Fig. 7. Speed and performace comparison of different image local feature extraction methods.

Balancing the speed and precision performance, the codebook for SURF or FAST+GLOH with 100k visual words using the kd-tree search strategy and soft-assignment yield the usable results for video processing solutions working on-line.

7 CONCLUSION

The presented work explores the image retrieval pipeline components. The state-of-the-art techniques based on visual codebooks were discussed for their utilization for video on-line processing. The experiments cover the influence of method choice and their parameters configuration for each component. The results show the precisions and speeds of codebooks with various sizes, with different weighting functions computing the visual word entropy, couple of soft-assignment strategies and two different search strategies for clustering method.

References

[1] Sivic, J., Zisserman, A.: Video Google: Efficient visual search of videos. In Ponce, J., Hebert, M., Schmid, C., Zisserman, A., eds.: Toward Category-Level Object Recognition. Volume 4170 of LNCS. Springer (2006) 127–144

[2] Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision **60**(2) (2004) 91–110

[3] Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, IEEE Computer Society (2006) 2161–2168

[4] Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2007)

[5] Jiang, Y.G., Ngo, C.W., Yang, J.: Towards optimal bag-of-features for object categorization and semantic video retrieval. In: CIVR '07: Proceedings of the 6th ACM international conference on Image and video retrieval, New York, NY, USA, ACM (2007) 494–501

[6] Philbin, J., Chum, O., Isard, M., Sivic, J., 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)

[7] Uijlings, J.R.R., Smeulders, A.W.M., Scha, R.J.H.: Real-time bag of words, approximately. In: CIVR '09: Proceeding of the ACM International Conference on Image and Video Retrieval, New York, NY, USA, ACM (2009) 1–8

[8] Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Gool, L.V.: A comparison of affine region detectors. Int. J. Comput. Vision **65**(1-2) (2005) 43–72

[9] Bay, H., Tuytelaars, T., Gool, L.V.: Surf: Speeded up robust features. In: In ECCV. (2006) 404–417

[10] Rosten, E., Drummond, T.: Machine learning for high-speed corner detection. In: In European Conference on Computer Vision. (2006) 430–443

[11] Mikolajczyk, K., Schmid, C.: Scale & affine invariant interest point detectors. Int. J. Comput. Vision **60**(1) (2004) 63–86

[12] Dalal, N., Triggs, B., Schmid, C.: Human detection using oriented histograms of flow and appearance. In: In European Conference on Computer Vision, Springer (2006)

[13] Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Trans. Pattern Anal. Mach. Intell. **27**(10) (2005) 1615–1630