Automatic Image Annotation Using Global and Local Features

Mária Bieliková
Faculty of Informatics and Information Technologies
Slovak University of Technology in Bratislava
Ilkovičova 3, 842 16 Bratislava, Slovakia
bielik@fiit.stuba.sk

Eduard Kuric
Faculty of Informatics and Information Technologies
Slovak University of Technology in Bratislava
Ilkovičova 3, 842 16 Bratislava, Slovakia
kuric@fiit.stuba.sk

Abstract **—Automatic image annotation methods require a quality training image dataset, from which annotations for target images are obtained. At present, the main problem with these methods is their low effectiveness and scalability if a large-scale training dataset is used. Current methods use only global image features for search. We proposed a method to obtain annotations for target images, which is based on a novel combination of local and global features during search stage. We are able to ensure the robustness and generalization needed by complex queries and significantly eliminate irrelevant results. In our method, in analogy with text documents, the global features represent words extracted from paragraphs of a document with the highest frequency of occurrence and the local features represent key words extracted from the entire document. We are able to identify objects directly in target images and for each obtained annotation we estimate the probability of its relevance. During search, we retrieve similar images containing the correct keywords for a given target image. For example, we prioritize images where extracted objects of interest from the target images are dominant as it is more likely that words associated with the images describe the objects. We tailored our method to use large-scale image training datasets and evaluated it with the Corel5K corpus which consists of 5000 images from 50 Corel Stock Photo CDs.**

Keywords-automatic image annotation; global features, local features; large-scale image datasets; image analysis

I. INTRODUCTION

Each of us likely has many photos and each of us has probably once thought "I would like to show you the photo, but if I knew where it is, I am unable to find it". With the expansion and increasing popularity of digital and mobile phone cameras, we need to search images effectively and exactly more than ever before. Focusing on visual query forms, many content-based image retrieval (CBIR) methods and techniques have been proposed in recent years, but they have several drawbacks. On the one hand, for methods based on query by example, a query image is often absent. On the other hand, query by sketch approaches are too complex for common users and a visual content interpretation of a user image concept is difficult. Therefore, image search using keywords is presently the most widely used approach.

Content based indexing of images is more difficult than for textual documents because they do not contain units like words. Image search is based on using annotations and semantic tags that are associated with images. However, annotations are entered by users and their manual creation for a large quantity of images is very time-consuming with often subjective results. Therefore, for more than a decade, automatic image annotation has been a most challenging task. Automatic image annotation methods are usually categorized into two categories, namely probabilistic modeling-based methods and classification-based methods.

Probabilistic-based methods estimate correlations or joint probabilities between images and annotation keywords over a training image dataset (corpus). Mori et al. [14] proposed the Co-occurrence model to capture correlations between images and keywords. The designed model is considered the main pioneer and consists of two stages. First, a grid segmentation algorithm is used to uniformly divide each image into a set of sub-images (segments) and for each the segment, a global descriptor is calculated.

Second, for the set of segments, the probability of each keyword is estimated by using a vector quantization of the features of the segment. The drawback of the model is a relatively low annotation performance. In [6] Duygulu et al. proposed a model of object recognition as a machine translation. A statistical translation model was used to translate keywords of an image to visual terms (blobs). A vocabulary of blobs was generated by clustering image regions segmented using the N-cut algorithm. Mapping between blobs and keywords was learned using the Expectation-Maximization algorithm.

One of the key problems of the model is high computational complexity of the Expectation-Maximization algorithm and therefore it is not suitable for large-scale datasets. Inspired by the relevance language models for text retrieval and cross-lingual retrieval, several relevance models were proposed, such as Continuous Relevance Model [10] and Cross-Media Relevance Model [9]. Feng et al. proposed the Multiple Bernoulli Relevance Model [7] that takes into account image context, i.e. from training images it learns that a tiger is more often associated with grass and sky and less often with objects, such as buildings or car. In comparison with the translation model, it seems to be more effective for image annotation. However, its drawback is that only images consistent with the training images can be annotated with keywords in a limited vocabulary.

The task of classification-based methods is to construct image classifiers for annotation keywords that are trained to separate training images with the keywords from other keywords with some level of accuracy. After a classifier is trained, it is able to classify a target image into a class where the keywords in the training dataset and retrieved outputs (keywords) are used to annotate the target image. Typical representative classifiers are Support Vector Machine (SVM) [4], Hidden Markov models [11] or the Bayes point machine [2]. However, the drawback of most classifiers is that they are designed for small-scale image datasets, i.e. classification into a small numbers of classes (categories). It is still an open research problem to construct large-scale learning classifiers and therefore, these methods are usually used for annotation of specific objects, such as car brands or company logos.

For all presented methods, a high quality annotated training image dataset is crucial. There are some web-based methods, which use crawled data (images, annotations) as the training dataset such as AnnoSearch [16]. With a target photo, an initial keyword (caption) is provided to conduct a text-based search on a crawled web database. Then a CBIR method is used to search visually similar images and annotations are extracted from obtained descriptions. The notable advantage is the availability of a large-scale web image database. The main drawback is the use of only global features for similar image search. One related approach [17] modifies the basic idea and extends the proposed method. Its main contribution is the absence of an initial caption in the search process, but for the entire image, only a global descriptor is still calculated.

The significant drawbacks of the presented "art" models are their performance and scalability if a large-scale image dataset is used; and/or use of only global features during search or image classification, respectively. Therefore, in our method we have focused on addressing these drawbacks.

Global descriptors capture the entire information of an image in a single feature vector (e.g., color, texture and shape). Their advantages are relatively low computational complexity, compact dimensions of the feature vector (descriptor) and the ability to capture complex information. Therefore, they are often used in automatic image annotation approaches. Local descriptors are calculated over local features of an image, such as edges, corners, small patches around points of interest. Interest points are very popular features due to their invariance to illumination and geometric transformations.

They were initially proposed to solve problems in computer vision, such as object detection and recognition. In recent years, they are increasingly used to solve the near-duplicate image detection problem. However, the robustness of interest point based methods imposes a performance penalty.

A huge number of descriptors per image can be extracted, typically hundreds to thousands per image, depending on the complexity of the image content. In order to process a single query, hundreds, even thousands of matches must be found and therefore, they are not used in content-based image retrieval methods to search images in large-scale image datasets. The local descriptors are much more precise and discriminating than global descriptors. When searching for specific objects, this feature is welcome, but when searching complex categories it can be an obstacle.

Therefore, we combine global and local features to retrieve the best results. Compared to existing methods, we are able to ensure the robustness and generalization needed by complex queries. In our method, in analogy with text documents, the global features represent words extracted from paragraphs of a document with the highest frequency of occurrence and the local features represent key words extracted from the entire document.

We are able to identify objects directly in target images. Our method estimates the probability that the retrieved similar images contain the right keywords for a given target image. We prioritize images where extracted objects of interest from target images are dominant in retrieved images or their frequency of occurrence is greater. It is more likely, that words associated with the images describe the objects. Consequently, for each obtained word, we estimate the probability of its relevance.

We place great emphasis on performance and have thus tailored our method to use large-scale image training datasets. To cope with the huge number of extracted features, we have designed disk-based sensitive hashing for indexing and clustering descriptors.

II. OUR IMAGE ANNOTATION METHOD

Our method (see Figure 1) consists of two main stages, namely training dataset pre-processing and processing of target image (query).

Dataset pre-processing consists of image processing (A), local and global features calculation (B) and their indexing and clustering (C).

Processing of target image consists of image processing (1), local and global features calculation (2), querying the keypoint store and global features index (3). After queries are executed, similar images (candidates) to the target image are retrieved as result sets (4). Subsequently, the result sets are refined (5). A final stage of assigning annotation is performed and relevance of assigned tags is estimated (6).

A. Local features calculation

For detection of interest points and calculation of descriptors, we use Scale Invariant Feature Transform (SIFT) [12]. Despite the fact, that there are some alternative methods, such as Speeded Up Robust Features (SURF) [1], we have chosen SIFT, because the descriptor is considered to be one of the most robust descriptor representations [13].

Extracted descriptors are invariant to image scaling, translation, partially invariant to illumination changes and affine for 3D projection. They are well adapted for characterizing small details. Features are detected through local extremes in a Difference-of-Gaussians function and described using histograms of gradients. Each SIFT keypoint consists of a descriptor (128-dimensional vector of floats), scale, orientation and location (Cartesian coordinates x, y). Up to hundreds to thousands keypoints can be extracted per image, which all together describe the image. The total number of extracted keypoints depends on the complexity of image content. For example, far fewer keypoints will be extracted from an image with a dominating clear sky than from an image showing a colorful garden.

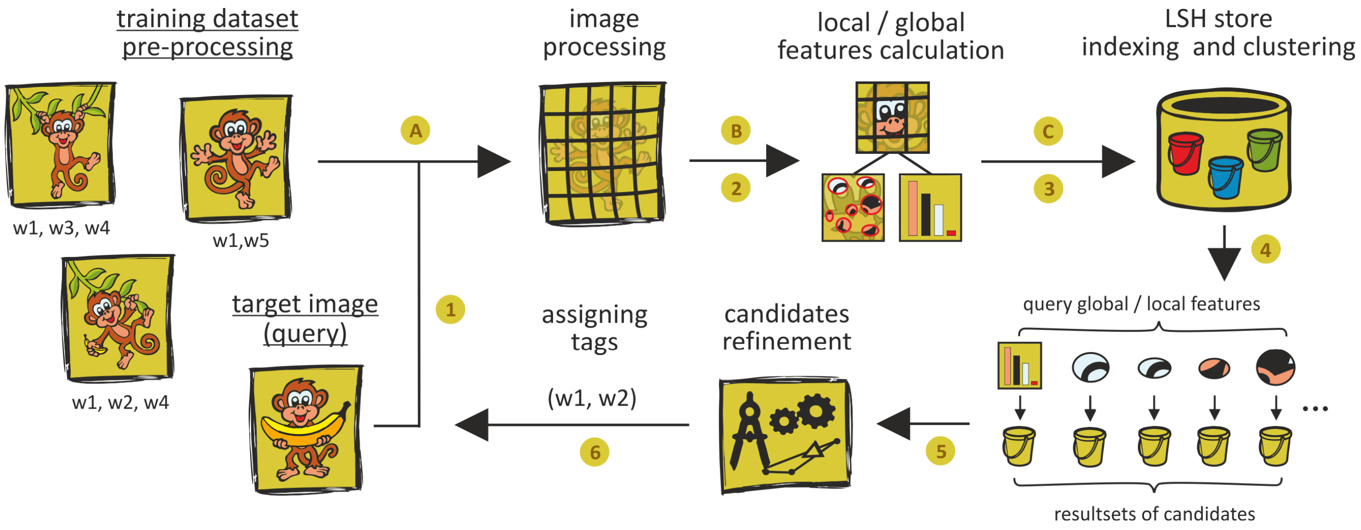


Figure 1. Scheme of our method for automatic image annotation.

In the case that an image has greater horizontal/vertical resolution than 768 pixels, it is scaled down with maintaining aspect ratio. Otherwise, the image is without change. We have set parameters of the SIFT method so that a maximum of 800 descriptors will be extracted from an image with a resolution of 768x512 pixels.

B. Indexing and clustering local features

For indexing extracted keypoints, we employ a disk-based locality-sensitive hashing (LSH) approach which solves the nearest neighbor search in high dimensional spaces. The basic idea is to hash descriptors so that similar descriptors are mapped to the same buckets with high probability.

More formally, if for a query-descriptor v_q, there exists an indexed-descriptor v_i such that $dist(v_i, v_q) \le r$, then an indexed-descriptor v'_i, such that $dist(v'_i, v_q) \le (1+\varepsilon)r$, will be returned with high probability. If no indexed-descriptor lies within $(1+\varepsilon)r$ of v_q, then nothing will be returned with high probability. We employ the LSH scheme [5] based on *p*-stable distributions as follows:

$$h(v)_{(a,b)} = \left\lfloor \frac{a.v+b}{w} \right\rfloor \qquad (1)$$

Each hash function $h(v)_{(a,b)} : R^d \to Z$ maps a *d*-dimensional descriptor *v* onto the set of integers. The parameter *a* is a *d*-dimensional vector with entries chosen from a *p*-stable distribution (Gaussian distribution), *b* is a real number chosen uniformly from the range [0, *w*]. The optimal value for *w* depends on the dataset and the query descriptor. In [5] it was suggested that *w* = 4.0 provides good results, therefore we chose this value. An LSH family *F* is a family of functions *h*. Each function g_i $(i = 1, ..., L)$ is obtained by concatenating *k* randomly chosen hash functions $h \in F$. Consequently, LSH constructs *L* hash tables, each corresponding to a given function g_i. Furthermore, the set of computed integers is mapped to a single natural number (unsigned integer) for bucket identification $g_i(h_{i_1}(v),...,h_{i_k}(v)) \to N$. The two parameters, *L* and *k* allow us to select a suitable compromise between accuracy and running time. In our method, we use *L = 15* and *k = 72*, based on performance with our experimental dataset.

For each extracted keypoint:

1) For each of the L LSH table, calculate a LSH hash (BucketID) using a descriptor of the keypoint.

2) Create a keypoint identifier by concatenating an ImageID and a keypoint location (ImageID_x_y).

3) Insert the keypoint identifier into all LSH tables according to the BucketIDs (see Table I).

4) Insert keypoint data into a keypoint table (see Table II).

The maximum size of each *BucketID* is 19 bytes. The *ImageID* is an identifier of the image, from which the keypoint was extracted. The *keypoint location* is given in Cartesian coordinates (x, y). The maximum size of each keypoint identifier is 27 bytes. All keypoint data are grouped in the keypoint table based on images, from which they were extracted. Before storing the descriptor, its elements are normalized into the interval $\langle 0, 255 \rangle$ of natural numbers.

After normalizing, the size of each descriptor is 128 bytes (1024 bits). Information about images is stored in an image dataset table (see Table III).

For storing the huge number of extracted local descriptors, we have adopted the distributed database management system Apache Cassandra. It is a highly scalable, distributed and structured key-value store with efficient disk access. It is a hybrid between column-oriented DBMS and row-oriented store. Cassandra was especially designed to handle very large amounts of data and is in use at Cisco, Facebook and Twitter. Using the Cassandra store and its cluster support, each LSH table can be stored on a single machine. The designed layout of the LSH table allows us even to split one LSH table onto multiple machines.

C. *Querying the keypoint store*

For a target image, we issue queries using a parallel set of steps:

1) *Extract keypoints from the target image.*

2) *For each target keypoint:*

a) *calculate the L bucket identifiers (BucketID's) for its descriptor,*

b) *select all keypoint identifiers, which are in buckets "labeled" by the BucketIDs,*

c) *associate the keypoint identifiers distinctly with the keypoint.*

3) *Group the returned keypoint identifiers according to ImageIDs.*

To maximize performance and efficiency for queries, we store only keypoint identifiers in each bucket. Therefore, for a target image, we can quickly estimate the best candidates from the retrieved keypoint identifiers.

After the query is executed, similar images (candidates) to the target image are retrieved as a result set and each of them is assigned its list of keypoint identifiers. Subsequently, image candidates are sorted in descending order according to cardinalities of the lists. Each keypoint of the target image is also assigned its own list of corresponding keypoints (see Table IV).

Because LSH returns approximate matches, we need to check for keypoints outside a threshold distance:

1) *For all keypoint candidates for correspondence, select their descriptors in binary representation from the keypoint table.*

2) *Calculate the Hamming distance between descriptors of each target keypoint and its candidates. The Hamming distance between two descriptors is the number of coefficients in which they differ.*

3) *Discard false matches by checking that the number is over the threshold.*

4) *Reorder the result set of the candidates.*

In our experiments, we chose the Hamming distance threshold of 170 bits. The calculation of the Hamming distances and comparison is very fast, because only a XOR logic operation is used.

Although, the candidates are already within the threshold distance, descriptors may be matched incorrectly, for example because of invariance failures of the used method or approximation errors. We need to verify geometric consistency between keypoints of a target image and their correspondence to keypoints of image candidates to eliminate outliers. For affine geometric verification, we use the RANSAC (RANdom SAmple Consensus) estimator [8].

After all stages, the final result set of local features is created and the best image candidates are returned in ascending order (see Table IV) and prepared for word extraction.

TABLE I. LAYOUT OF ONE LSH TABLE

BucketID	ImageID_x_y	ImageID_x_y	…
1	1_135_11	5_41_31	…
2	2_56_201	5_185_39	…
…	...	...	…

TABLE II. LAYOUT OF A KEYPOINT TABLE

ImageID	Keypoint Location (x_y)			…		
	Descriptor	*Orientation*	*Size*	…	…	…
1	135_11			…		
	[A_1, …, A_{128}]	B	C	…	…	…
2	56_201			…		
	[X_1, …, X_{128}]	Y	Z	…	…	…
…	…			…		
	…	…	…	…	…	…

TABLE III. LAYOUT OF A IMAGE DATASET TABLE

ImageID	File name	Keywords
1	Image file 1	w1_w2_w3
2	Image file 2	w1_w2_w4_w5
…	…	…

TABLE IV. RESULT SET OBTAINED VIA KEYPOINTS QUERIES

Target Image (File name)	Candidate (ImageID)	Candidate (ImageID)	…
sunset.jpg	5	1	
Target Keypoints	**Corresponding Keypoints**	**Corresponding Keypoints**	**…**
33_28	41_31; …	135_11; …	…
39_41	185_39; …	-	…
…	…	…	…

D. *Global features calculation*

Our calculated local descriptors do not contain important visual information regarding color because the SIFT method operates on grayscale images. Therefore, to capture complex information, we employ the Color and Edge Directivity Descriptor (CEDD) [3], where global descriptors ensure generalization. For example, they are able to describe relatively homogeneous regions in the image, such as clear sky and sand, which are regions that are usually ignored during detection of interest points. The CEDD belongs to the group of Compact Composite Descriptors (CCD), which combine information about color and texture in a single histogram. It was designed with regard to dimension, but without compromising their discriminating ability. The descriptor is partially robust against image deformation, noise and smoothing. Its size is limited to 54 bytes per image. The important attribute is the low computational complexity needed for extraction.

For the calculation of global descriptors, an image is scaled to the 3:2 (2:3) aspect ratio using bicubic interpolation. The original image size is changed to one of the nearest resolutions: 768x512, 384x256, 192x128 and 96x64 pixels. Thus, the image is scaled up (interpolated) if a difference between the nearest resolution and the original image resolution is less than one quarter of the nearest resolution. For example, if the original image resolution is 672x504 pixels, than the image is interpolated to the resolution 768x512 pixels.

Subsequently, the image is divided into 8x8, 4x4 or 2x2 sub-images (segments) using grid segmentation. The number of segments depends on image resolution. For example, an image with resolution 384x256 pixels is divided to 4x4 segments. The image resolution 96x64 pixels is canonical. After image segmentation, a global descriptor (CEDD) is calculated for each segment.

E. *Indexing and clustering of global features*

Indexing and clustering of global features is very similar to the introduced indexing and clustering of local features, because we use the same approach based on LSH hashing. All calculated global descriptors consist of 144 bins. Each bin contains a 3-bit number (0-7). Consequently, all the bins take together 54 bytes or 432 bits, respectively. The main difference is the LSH hash function which is now based on bit sampling. The LSH parameters for indexing of global features are $L = 10$ and $k = 320$. The maximum size of each *BucketID* is 40 bytes. The scheme of table layouts is similar, but the identifier of global descriptors (GD identifier) is in the form ImageID_SegmentIndex. The maximum size of each GD identifier is 22 bytes.

F. *Query to global features index*

The goal of this stage is to retrieve segments similar to segments of a target image similarly to querying for keypoints. The result is a result set of global features which are similar to the global target features (see Table V).

TABLE V. RESULT SET OBTAINED VIA SEGMENTS QUERIES

Target Image (File name)	**Candidate (ImageID)**	**Candidate (ImageID)**	**…**
sunset.jpg	5	1	
Target Segments	**Similar segments**	**Similar segments**	**…**
1	2; 3	1; 2	…
2	11	10	…
…	…	…	…

G. *Assigning annotation and relevance estimation*

We illustrate the principle of annotation on the result set obtained by keypoints queries. For the result set obtained by segments queries, the principle is similar.

We create objects from target keypoints on the basis of obtained correspondence. The idea is illustrated in Figure 1. From the keypoints of the target image *T*, we create 3 objects *A*, *B*, *C* and clusters *X, Y, Z*. The objects/clusters are created based on grouping the image candidates according to correspondence. For example, keypoints in the images *Z1* and *Z2* are "connected" with the same keypoints of the target image. Objects in clusters can have different sizes or different frequency of occurrence. Therefore, from coordinates of the keypoints and the connections, size and frequency of the objects are calculated. Subsequently, the clusters are reordered by the size (in descending order) and the frequency, respectively.

Each image in the clusters is associated with words (tags). For each keypoint of the target image, its corresponding cluster is iterated over images of and words associated with each image are assigned to the keypoint. If the frequency of an assigned word equals the threshold *M* or the frequency of all assigned words is lower than the threshold *M* and no more images are available, then iteration is stopped and the next keypoint is evaluated.

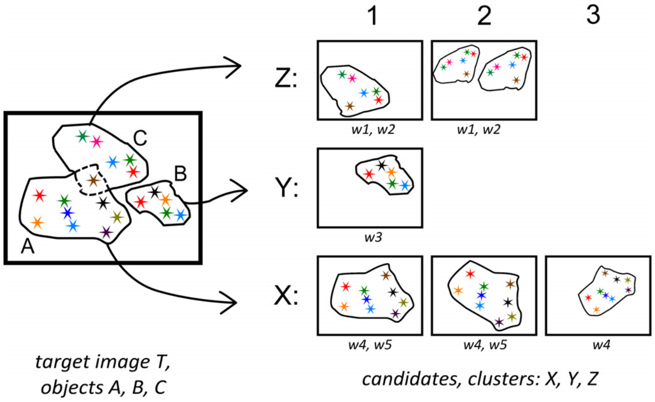


Figure 2. Creating of objects from target keypoints.

Subsequently, word relevance is estimated for all the keypoints:

1) *Init a set W = {}. Each element is a three element list of word, occurrence and relevance.*

2) *For each target keypoint,*

  a) *select an assigned word with the highest frequency,*

  b) *divide the frequency by a threshold M,*

  c) *if the word is not in W, set its occurrence value to 1, relevance to the calculated value and add the word to W else increment its co-occurrence value and add the calculated value to the current relevance value.*

  a) *if there are words with the same frequency,*

  b) *if the words are not in W, add the words to W, set their occurrence values and relevance values to 0,*

  c) *increment occurrence value of all the words (relevance value is without change)*

3) *Assign words from W to a target image and for each word calculate its local relevance value as a proportion of its relevance value and occurrence value.*

III. EVALUATION AND CONCLUSIONS

A. *Corel5K Dataset*

Our evaluation was conducted over the Corel5K corpus. It consists of 5,000 images from 50 Corel Stock Photo CDs and each CD includes 100 images with the same theme. The corpus is used widely in the automatic image annotation area and includes a variety of subjects, ranging from urban to nature scenes and from artificial objects to animals. It is divided into 2 sets: a training set of 4,500 photos and a test set of 500 photos. Each photo is associated with 1-5 keywords and all photos are in the resolution 384x256 pixels and 256x384 pixels, respectively.

B. *Annotation performance*

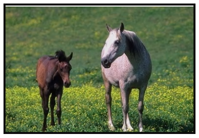
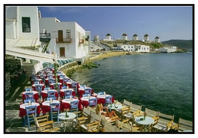
We compared our method with the Translation Model. We report the results on selected subset of the best 49 words which was used by Daygulu et al. [6]. To evaluate the annotation performance, we used the *precision* (P) and *recall* (R) metrics. Let A be the number of images automatically annotated with a given word, B the number of images correctly annotated with that word. C is the number of images having that word in ground-truth annotation. Then $R = \frac{B}{C}$ and $P = \frac{B}{A}$.

The result comparison of our method and the translation model is shown in Table VI. Examples of annotation results are shown in Table VII. Figure 3 shows precision and recall for a subset of 13 selected words.

TABLE VI. COMPARING RESULTS OF OUR METHOD AND THE TRANSLATION MODEL.

	Mean Precision	**Mean Recall**
Our method	0.23	0.31
Translation Model	0.20	0.34

TABLE VII. AUTOMATIC ANNOTATIONS COMPARED WITH THE HUMAN ANNOTATIONS.

	*Human annotation:* field, foals, horses, mare *Automatic annotation:* foals, horses, field, fence, mare
	*Human annotation:* buildings, cafe, shore, water *Automatic annotation:* water, sky, buildings, stone

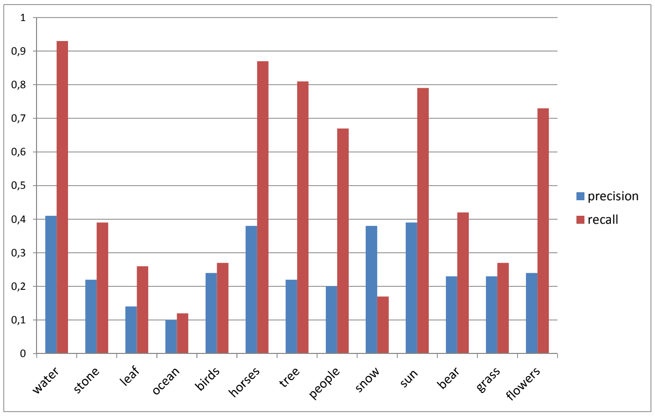


Figure 3. Performance of words for our method.

Our method for automatic image annotation is based on combining local and global features. It can be used for natural extension of image retrieval or navigation in large image sets, such as our faceted browser [15]. Even if the SIFT descriptor is successful in recognizing objects, its potential has not been fully exploited with the Corel5K dataset. In a more general object image database, global features are more important than local features. Therefore, via the combination of global and local features, we achieved the required robustness for effective automatic annotation.

ACKNOWLEDGEMENT

This work was partially supported by the grants VG1/0675/11/2011-2014, KEGA 028-025STU-4/2010, APVV-0208-10 and it is the partial result of the Research & Development Operational Programme for the project Research of methods for acquisition, analysis and personalized conveying of information and knowledge, ITMS 26240220039, co-funded by the ERDF.

REFERENCES

[1] Bay H., et al.: Speeded-Up Robust Features (SURF). *J. of Computer Vision and Image Understanding*. vol. 110, no. 3, 2008, pp. 346-359.

[2] Chang, E., et al.: CBSA: Content-Based Soft Annotation for Multimodal Image Retrieval Using Bayes Point Machines. *IEEE Trans. on Circuits and Systems for Video Technology*, 2003, vol. 13, no. 1, pp. 26–38.

[3] Chatzichristofis, S., A., Boutalis, Y., S.: CEDD: Color and edge directivity descriptor - A compact descriptor for image indexing and retrieval, *6th Int. Conf. in advanced research on Computer Vision Systems* ICVS 2008, Santorini, Greece, 2008.

[4] Cusano, C., Ciocca, G., Schettini, R.: Image annotation using SVM. In: *Proc. of Internet Imaging IV*, 2004, vol. SPIE 5304, pp. 330-338.

[5] Datar, M., et al.: Locality-sensitive hashing scheme based on p-stable distributions. In: *Proc. of the 20th Symposium on Computational geometry* (SCG '04). ACM, New York, 2004, pp. 253-262.

[6] Duygulu, P., Barnard, K.: Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary, *7th European Conf. on Computer Vision*, 2002, pp. IV:97-112.

[7] Feng, S. L., Manmatha, R., and Lavrenko, V.: Multiple Bernoulli relevance models for image and video annotation. In *Proc. of the Int. Conf. on Computer vision and pattern recognition* (CVPR'04). IEEE Computer Society, Washington, DC, USA, 2004, 1002-1009.

[8] Fischler, A., M., Bolles, C., R.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In *Readings in computer vision*, Morgan Kaufmann Publ. Inc., San Francisco, CA, USA, 1987, pp. 726-740.

[9] Jeon, J., Lavrenko, V., Manmatha, R.: Automatic image annotation and retrieval using cross-media relevance models. In: *Proc. of the 26th Int. Conf. on Research and Development in Information Retrieval* (SIGIR '03). ACM, New York, 2003, pp. 119-126.

[10] Lavrenko, V., Manmatha, R., Jeon, J.: A model for learning the semantics of pictures. In: *Proc. of the 16th Conf. on Advances in Neural Inf. Processing Systems* (NIPS '03). 2003.

[11] Li, J., Wang, J.: Automatic Linguistic Indexing of Pictures by a Statistical Modeling Approach. *IEEE Trans. Pattern Anal. Mach. Intell. 25, 9*, 2003 1075-1088.

[12] Lowe, D.: Distinctive Image Features from Scale-Invariant Keypoints, *Int.J. of Computer Vision*, 2004, vol. 2, no. 60, pp. 91-110.

[13] Mikolajczyk, K., Schmid, C.: A Performance Evaluation of Local Descriptors. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, IEEE, 2005, pp. 1615–1630.

[14] Mori, Y., Takahashi, H., Oka, R.: Image-to-word transformation based on dividing and vector quantizing images with words. In: *Proc. of the Int. Workshop on Multimedia Intelligent Storage and Retrieval Management*. 1999.

[15] Tvarožek, M.: Exploratory Search in the Adaptive Social SemanticWeb. Information Sciences and Technologies Bulletin of the ACM Slovakia,vol. 3, no. 1, pp.42-51, 2011.

[16] Wang, X., et al.: AnnoSearch: Image Auto-Annotation by Search. In: *Proc. of the Conf. on Computer Vision and Pattern Recognition* (CVPR '06), vol. 2. IEEE CS, Washington, 2006, pp. 1483-1490

[17] Wang, C., et al.: Scalable search-based image annotation of personal images. In: *Proc. of the 8th Int. Workshop on Multimedia Information Retrieval* (MIR '06). ACM, New York, 2006, pp. 269-278.