ANNOR: Efficient Image Annotation Based on Combining Local and Global Features

Eduard Kuric and Mária Bieliková

Faculty of Informatics and Information Technologies
Slovak University of Technology
Ilkovičova 2, 842 16 Bratislava 4, Slovakia
E-mail: {eduard.kuric,maria.bielikova}@stuba.sk

Abstract

Automatic image annotation methods based on searching for correlations require a quality training image dataset. For a target image, its annotation is predicted based on a mutual similarity of the target image to the training images. The one of the main problem of current methods is their low effectiveness and scalability if a relatively large-scale training dataset is used. In this paper we describe our approach "Automatic image aNNOtation Retriever" (ANNOR) for acquiring annotations for target images, which is based on a combination of local and global features. ANNOR is resistant to common transforms (cropping, scaling), which traditional approaches based on global features cannot cope with. We are able to ensure the robustness and generalization needed by complex queries and significantly eliminate irrelevant results. We identify objects directly in the target images and for each obtained annotation we estimate the probability of its relevance. We focus on the way, how people manually annotate images (human aspects of image perception). We have designed ANNOR to use large-scale image training datasets. We present experimental results for three challenging (baseline) datasets. ANNOR makes an improvement as compared to the current state-of-the-art.

Keywords: image retrieval, automatic annotation, object recognition, local features, global features, locality sensitive hashing

1. Introduction

Automatic image annotation has been studied extensively for several years. Many of us likely has hundreds to thousands photos and apparently each of us has probably at least once thought *"I would like to show her the photo, but 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 methods and techniques have been proposed, but they have several limitations. On the one hand, in query-by-example-based methods a query image is often absent. On the other hand, query-by-sketch approaches [1, 2] are too complex for common users and a visual content interpretation of a user image concept is difficult.

A text retrieval system often helps finding rapidly related documents from a vast amount of documents containing keywords. Image search using keywords is presently the most widely used approach. Content based indexing of images is more difficult than indexing of textual documents because they do not contain units like words. Image search is based on 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.

The goal of automatic image annotation is to assign a collection of keywords (annotation) from a given dictionary to a target (previously unseen) image. i.e., the input is the target (uncaptioned) image and the output is a collection of keywords that describes the target image in a best possible way.

Why automatic image annotation is a challenge? Automatic image annotation is on the frontier of different fields such as image analysis, machine learning and information retrieval. In present, to create a general system for automatic image annotation based on object recognition is practically impossible (it is doubtful if ever at all). The Imagenet Large Scale Visual Recognition Challenge (ILSVRC)[1] is the venue for evaluating the current state-of-the-art for image classification and recognition.

To extract the semantics from data, general object recognition and scene understanding is required. This is an extremely hard task. The same object can be captured from different angles, distances or under different lightning conditions. The manual annotation is subjective and sometimes it is difficult to describe image contents by keywords. In general, an object of the real world with the same "name" may have different visual form (e.g. shape, color). Good illustrative examples are methods for face recognition. There are several approaches for face recognition, e.g. methods based on comparing templates, which require a robust database of faces. The faces are searched based on correlating between an input (a target face) and the templates. Complex knowledge-based methods focus on analyzing morphological features such as eyes, mouth, skin and color. They

[1] ILSVRC: http://image-net.org/

are based on rules defined by the real features of human faces.

Here are some crucial questions that current automatic image annotation systems have to deal with:

- *Which image representation is appropriate to describe image?* The objects in images are often occluded and appear in poor lighting and exposure.
- *Which image features can be extracted to describe or characterize the visual content?* A feature is represented by a numerical feature vector (descriptor), by which we are able to describe a part of image content. In general, there are three essential requirements for the descriptors, their degree of robustness, discrimination ability and efficiency. The robustness represents invariance to the geometrical changes (e.g. viewpoint, zoom, object orientation) and noise-like signal distortions. The discrimination maximizes difference among non-duplicates and minimizes difference among duplicates. The feature extraction and matching requires fast computation.

Another question is the spatial and time complexity (computational cost). A huge number of features per image can be extracted and the dimension of the feature vector is crucial aspect, too. There is a problem how to index, store and compare the descriptors in real-time. Often in many cases, faster access to information means the need for more space allocation.

In this paper we propose a method for automatic image annotation using relatively large-scale image "training" dataset. We combine local and global features to ensure robustness and generalization needed by complex queries and therefore we focus on performance and scalability. For indexing and clustering features, we use disk-based locality sensitive hashing. To obtain annotation for a given target image, our approach is based on the way how people manually annotate images.

Compared with our previous work [3] we present completely new process of obtaining annotation called ANNOR (Automatic image aNNOtation Retriever). The evaluation part is also completely new. We have performed new experiments focused on evaluation of efficiency and quality of obtaining annotation. We have evaluated our approach on three datasets and we have compared the results of our approach with the state-of-the-art approaches.

This paper is structured as follows: Section 2 provides an overview of existing methods for automatic image annotation; Section 3 introduces our approach; Section 4 and 5 describe in details extracting, indexing, clustering and retrieving local features and global features, respectively. Section 6 describes in details obtaining annotation for the target image and estimation its relevance; Section 7 presents the evaluation results of our approach; and Section 8 contains discussion and conclusion.

2. Previous work on automatic image annotation

2.1. State-of-the-art

Automatic image annotation methods are usually divided 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. [4] 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.

Duygulu et al. [5] 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 [6] and Cross-Media Relevance Model [7], Dual cross-media relevance model [8], Multimodal Latent Binary Embedding [9]. Feng et al. proposed the Multiple Bernoulli Relevance Model [10] 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.

Metzler et al. [11] segment training images, connecting them and their annotations in an inference network. The inference network is based on Bayesian Network. It uses non-parametric methods to estimate probabilities within the inference network.

Yavlinsky et. al. [12] proposed a framework based on non-parametric density estimation and the technique of kernel smoothing. Their results are comparable with the inference network [11] and CRM [8].

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) [13, 14, 15, 16, 17], Hidden Markov models [18], Markov Random Fields [19], Supervised multi-class labeling [20] or the Bayes Point Machine (BPM) [21, 22].

The overall disadvantage 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 (corpus) is crucial. There are some web-based methods, which use crawled data (images, annotations) as the training dataset such as AnnoSearch [23]. With a target photo, an initial keyword (caption) is provided to conduct a text-based search on a crawled web database. Then a content-based image retrieval 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 the similar image search. One related approach [24] modifies the basic idea of AnnoSearch. 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 limitations of the presented "art" models are their performance and scalability if a large image dataset (corpus) is used; and/or use of only global or local features during searching or image classification, respectively.

2.2. Image representation: global and local feature-based approaches

The commonly used feature representation is based on a global feature set extracted from images. Global features capture the entire information of an image in a single feature vector (e.g., color distribution, 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.

Vailaya et al. [22] use Bayesian classifiers on the color and edge direction histograms to classify vacation photographs into a hierarchy of high-level classes. At first, images are classified as indoor or outdoor. The outdoor images are then classified as city or landscape. Finally, a subset of landscape images is further classified into classes such as sunset, forest, and mountain.

Yavlinsky et al. [12] use non-parametric models of distributions of image features. Authors present a framework for automatic image annotation based on non-parametric density estimation and employ global color and texture distributions. They use the Earth Mover's Distance (EMD) kernel which uses global color information. Results are reported on subsets of two photographic libraries, namely, the Corel Photo Archive and the Getty Image Archive.

Makadia et al. [25], Babenko et al. [26] and Guillaumin et al. [27] directly transfer annotations from training images to test images with global image similarities using a weighted nearest neighbor approach. For example, Makadia et al. [25] extract global color and texture as features; calculate image similarity as the average distance using these features; and the keywords are obtained from the nearest neighbors with the least distance.

Unlike global descriptors, local descriptors are calculated over local features of an image, such as edges, corners, small patches around points of interest. Repeatability is the most important quality for a local feature technique: even if the image suffers geometric deformations, or if the scene is captured from another viewpoint, the "near-duplicate" features must be found. In other words, this means that the extracted patches, edges and points must have suffered the same geometric transformation than the image, in order to fall over the same objects.

The interest points are very popular local 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, depending on the complexity of the image content. Often in order to process a single query, hundreds, even thousands of matches must be found and therefore, they are not usually used in content-based image retrieval approaches to search images in large-scale image datasets.

Local feature-based approaches can be divided into block-based and region-based approaches. The simplest way to extract block-based features is to roughly segment images into a fixed number of sub-blocks. Visual features are then extracted from these blocks.

Szummer and Picard [28] first segment each image into a fixed number of blocks; color and texture features of each block are extracted. Then, a k-NN (K-Nearest Neighbor) classifier is designed to classify the color and texture features of each block into indoor and outdoor categories individually. The final output is based on the blocks of an image which have the highest vote for one of the indoor and outdoor.

Serrano et al. [15] used SVMs to classify color and texture features of 16 blocks per image into indoor and outdoor classes individually. Zhang and Ma [16] proposed a blockfeature-based multi-class SVM. For image annotation, each image is segmented into five fixed-size blocks.

Yi and Tang [29] first divided the whole image into different sizes of blocks and generate suitable visual words. Learning is based on the Probabilistic Latent Semantic Analysis (PLSA) by given a set of image blocks for each semantic concept as training data. Finally, the classification of the images is carried out by combining all the image blocks in every block size.

The second approach for local feature representation is to divide the image into homogenous regions (objects) or edges (boundaries) using segmentation algorithms.

Blei and Jordan [30] described three hierarchical probabilistic mixture models for a database of annotated images, culminating in Correspondence Latent Dirichlet Allocation (Corr-LDA), a model that finds conditional relationships between latent variable representations of sets of image regions and sets of words, Authors demonstrated its use in automatic image annotation, automatic region annotation, and text-based image retrieval.

Yang et al. [31] used Multiple Instance Learning (MIL) to learn the correspondence between image regions and words. Tang and Lewis [32] proposed to realize automatic region-based

image annotation through a training image feature space.

The local features are much more precise and discriminating than global features. When searching for specific objects, this feature is welcome, but when searching complex categories it can be an obstacle. Global feature-based approaches have low computational cost for the feature extraction, they are more distinctive because they have the ability to capture complex and contextual layout information, i.e., they are advantageous in classifying simple scene categories. However, they do not capture spatial information and they are weak in characterizing the internal content of image especially when the image has multiple complex objects. In local feature-based approaches, images are divided into regions or blocks and a set of features is computed for each of the "segment", which means that an image is represented as a bag of features. A bag of features can represent images at object level and provides spatial information which makes them more precise and discriminating than the global feature-based approaches. Local features may not be accurate due to the usually unsupervised segmentation, and the appearance features extracted from segments are less distinctive and even with perfect segment labels; their union does not always match well. In addition, an image is represented by many visual feature vectors (descriptors), resulting in high computational cost.

2.3. Duplicate image search approaches used in automatic image annotation

Automatic image annotation systems are usually based on duplicate image search approaches. Duplicate images are close-enough similar images to a given (target) image which are exactly the same, or allow variances in scale, color, luminance change, and a small number of pixels, or have large variances in visual content (so-called near duplicate images).

The motivation is that visually close images (image regions) retrieved from an image corpus possess certain semantic similarity to a target image so that keywords (annotations) can be propagated among them. In other words, for target (uncaptioned) images we can find their duplicates in well-annotated and "unlimited" image corpus (visual vocabulary). Then, a target image can be annotated simply by propagating the words (annotation) from its duplicates, i.e., we are able to extract some words from the textual descriptions (annotations) of the image search results, and we can use the most salient words to annotate the target image.

In [33], authors use the K-Means algorithm for a near-duplicate detection of images. A vocabulary is built using the algorithm on local image descriptors. This is a costly step and image retrieval efficiency depends on a learning database. High recalls are obtained if the vocabulary is learned on the searched corpus. However, if the vocabulary is computed on a different set of images, efficiency decreases. This is a crucial problem for searching images in a corpus where images are regularly added or removed (re-calculation is needed). In other words, to ensure the best retrieval results, it is needed to update regularly the K-Means-based visual vocabulary, i.e., the main drawback of K-Means-based algorithm is that it is not adapted for regularly updated databases.

Lowe [34] used a non-exhaustive visiting algorithm (Best-Bin-First) on a KD-Tree. To be more effective, randomized KD-Tree forests were used in [35]. The problem is that there is no guarantee that parameters (number and depth of trees) will guarantee good performance if the number of images within the database evolves.

3. Concept of the proposed method for automatic image annotation

In our approach, we combine global and local features to retrieve the best results. The combination is more suitable to represent complex scenes and events categories. Global and local features have limitations describing images and none of them appears to be powerful enough to represent the large amount and variety of images. Global and local features provide different kinds of information. They have their own advantages in classifying certain categories. However, they have several complementary strengths and there are many situations where the automatic image annotation should be judged based on the combination of global and local features.

We use grid segmentation for extracting the global features and efficient graph-based segmentation for extracting local features. Compared to existing methods, we are able to ensure the *robustness* (invariance to geometrical changes) and *generalization* (description of homogeneous regions) needed by complex queries. In approach 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 keywords extracted from the entire document.

We are able to identify objects directly in target images. Our approach estimates the probability that the retrieved similar images (training images) contain the right words for a given target image. Our estimation of annotation is based on human aspects of image perception. We focus on the way, how people manually annotate images. We prioritize dominant objects and estimate relative importance of words in the training annotations. The estimated probability of words determines degree of accuracy with which the words describe the visual content of the target image.

Our approach (see Figure 1) consists of two main stages:

- training dataset pre-processing,
- processing of the target image (query).

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

Processing of the 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 (visual terms) to the target image are retrieved as result sets (4). Subsequently, the result sets are refined (5). A final stage of obtaining annotation is performed and relevance of assigned words is estimated (6).

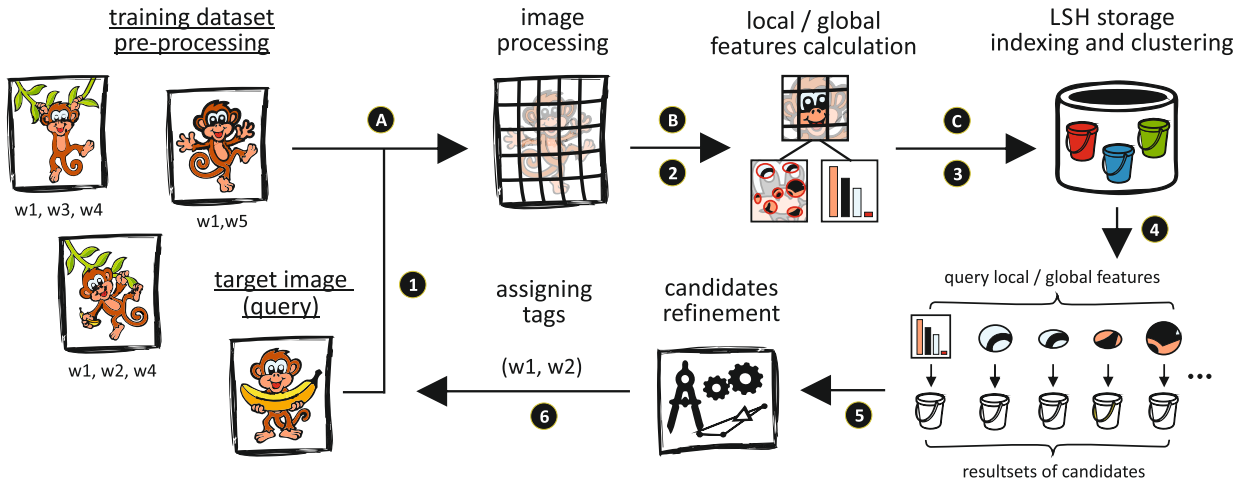

Figure 1: Scheme of our approach for automatic image annotation.

In our work we are looking for a good compromise among *high precision*, *high recall* and *time* needed to query an image. We argue that practical annotation system should satisfy all those aspects. Precision and recall are of course paramount, however, efficiency is just as important. We place great emphasis on performance. Thus we have designed our approach to use large-scale image training datasets. To cope with the huge number of extracted features, we have designed disk-based locality sensitive hashing for indexing and clustering descriptors. We have chosen locality sensitive hashing for several reasons. First, it is not related to any learning corpus, it may be fast, and retrieval performance does not evolve when modifying the database while this is not true for tree-based methods. Using a K-Means approach would require updating the visual vocabulary regularly to avoid degraded performance (and to define when to do these updates). Our approach is particularly suitable solution for applications where the image corpus evolves, i.e., our solution provides a good compromise between precision and speed.

4. Local features

4.1. Local features calculation

For detection of interest points and calculation of descriptors, we use Scale Invariant Feature Transform (SIFT) [34]. Despite the fact, that there are some alternative methods, such as Speeded Up Robust Features (SURF) [36], we have chosen SIFT, because the descriptor is considered to be one of the most robust descriptor representations [37].

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*.

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.

The training/target image is divided into regions (visterms visual terms, see Figure 2) using efficient graph-based image segmentation algorithm [38]. The algorithm is based on defining a predicate for measuring the evidence for a boundary between two regions using a graph-based representation of the image. Although, this algorithm makes greedy decisions, it produces segmentations that satisfy global properties. The algorithm runs in time nearly linear in the number of graph edges and is also fast in practice. An important characteristic of the algorithm is its ability to preserve detail in low-variability image regions while ignoring detail in high-variability regions.

After creating visterms, each one is labelled by a unique identifier (VisID). Subsequently, the SIFT keypoints are extracted from each visterm (see Figure 3) and indexed using locality-sensitive hashing algorithm. The detected keypoints, which are located on the edges in the visterms, are ignored when indexing. For each image, up to 256 keypoints are extracted. We limit the number of keypoints as it has been noticed in [39] that the recall almost does not decrease when passing from 1,000 features per image to 256.

4.2. Indexing and clustering local features

For indexing extracted keypoints, we employ a disk-based locality-sensitive hashing (LSH) approach which solves the problem of nearest-neighbor search in high dimensional spaces [40]. The basic idea of LSH 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 $dist(v_i, v_q) \leq r$, then an indexed-descriptor $v'i$, such that $dist(v'_i, v_q) \leq (1 + \epsilon)r$, will be returned

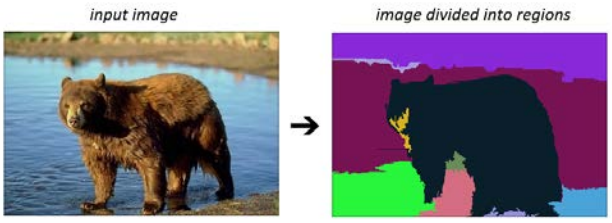

Figure 2: An illustration of segmentation results produced by efficient graph-based image segmentation algorithm. The input image is divided into 10 regions (visual terms).

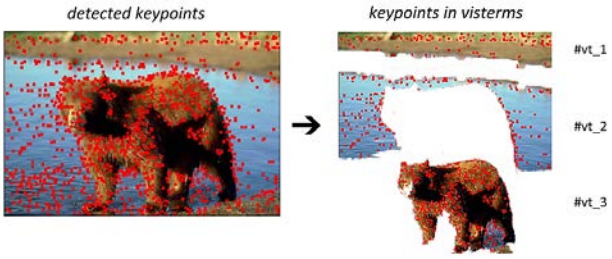

Figure 3: An illustration of detected keypoints in the input image and in the three examples of visual terms.

with high probability. If no indexed-descriptor lies within $(1 + \epsilon)r$ of v_q, then nothing will be returned with high probability. We employ the LSH scheme [41] based on p-stable distributions as follows:

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

Each hash function $h(v)_{(a,b)} : R^d \rightarrow Z$ 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 [41] 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 $g_i(h_{i_1}(v), ..., h_{i_k}(v)) \rightarrow N$. The two parameters L and k allow us to select a suitable compromise between accuracy and running time. In our approach, we use $L = 20$ and $k = 112$, based on performance with our experimental dataset.

For each extracted keypoint:

1. For each of the LSH table L, calculate a LSH hash (BucketID) using a descriptor of the keypoint.
2. Create a keypoint identifier by concatenating VisID and keypoint location (ImgID_x_y).
3. Insert the keypoint identifier into all L LSH tables according to the calculated BucketIDs (see Table 1).
4. Insert keypoint data into a keypoint table (see Table 2).

BucketID	VisID_x_y	VisID_x_y	...
1	1_135_11	5_41_31	...
2	2_56_201	5_185_39	...
...	...	...	...

Table 1: Layout of one LSH table for indexing keypoints.

VisID	ImgID	Keypoint Location (x_y)			...		
		Descr.	Orient.	Size	.	.	.
1	1	135_11			...		
		$[A_1, ..., A_{128}]$	B	C	.	.	.
2	1	56_201			...		
		$[X_1, ..., X_{128}]$	Y	Z	.	.	.
...	...	...			...		
		...	...	...	.	.	.

Table 2: Layout of a keypoint table.

The maximum size of each BucketID is 19 bytes. ImgID 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 49 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 3).

ImgID	File name	Keywords
1	Image file 1	w1_w2_w3
2	Image file 2	w1_w2_w4_w5
...	...	...

Table 3: Layout of an image dataset table.

For storing the huge number of extracted local descriptors, we need a data storage that allows us to store the descriptors not only in depth (rows) but also in breadth (columns) in real time. Classical relational databases are unable to provide that. There are several solutions that can cope with the problem, e.g. BigTable, Cassandra, HyperTable. We have chosen the database management system Cassandra[2]. It is a highly scalable, open-source, distributed and structured key-value store with efficient disk access (access complexity of $O(1)$). It is a hybrid between column-oriented DBMS and row-oriented store. Cassandra was especially designed to handle very large amounts of data. Using the Cassandra store and its cluster support, each LSH table (Column Family) 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.

4.3. Querying the LSH keypoint store

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

[2]Apache Cassandra: http://cassandra.apache.org/

Target Visterm Keypoint		Corresponding Keypoint			Corresponding Keypoint			...
x_y	VisID	x_y	Similarity	VisID	x_y	Similarity	VisID	
33_28	1	41_31	0.92	13	135_11	0.94	21	...
...		...	...		...	...		...
			$\sum$			$\sum$		
Target Visterm Keypoint		Corresponding Keypoint			Corresponding Keypoint			...
x_y	VisID	x_y	Similarity	VisID	x_y	Similarity	VisID	
...	2	...	...	...	...	...	...	...
...		...	...		...	...		...
			$\sum$			$\sum$		

Table 4: An illustration of a result set obtained via keypoints queries for a target image.

1. Divide the target image into visterms (using efficient graph-based segmentation algorithm) and extract target keypoints from each one.
2. For each target keypoint:
   (a) calculate the L bucket identifiers (BucketIDs) for its descriptor using the corresponding hash functions g_i,
   (b) select all keypoint identifiers which are in the buckets "labelled" by the BucketIDs,
   (c) associate the corresponding keypoint identifiers distinctly with the target keypoint.
3. Group the returned keypoint identifiers according to VisIDs.

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

After the query is executed, similar visterms (candidates) to the target visterms are retrieved as a result set. All target visterms have assigned a list of corresponding keypoints to their target keypoints. In other words, each keypoint of the target visterm is also assigned its own list of corresponding keypoints and they are grouped into visterms (see Table 4).

Because LSH returns approximate matches, we need to check for keypoints outside a threshold distance. All target keypoints are normalized into the interval $\langle 0, 255\rangle$ of natural numbers and for each corresponding keypoint, its descriptor is selected according to VistermID (see Table 2). Subsequently, each target keypoint is compared with the corresponding keypoint using *Tanimoto Distance*:

$$T(x_1, x_2) = \frac{{x_1}^T x_2}{{x_1}^T x_1 + {x_2}^T x_2 - {x_1}^T x_2}, \quad (2)$$

where x_1, x_2 are descriptors (vectors) of the keypoints, which are compared with each other, and x^T is the transposition of the vectors. The resulting distance has a value of 1 for identical vectors and 0 for extremely dissimilar vectors. After computing similarities, false matches are discarded by checking that the distance is over the threshold (experimentally set to 0.9). Subsequently, corresponding visterms to the target visterms are sorted in descending order according to cardinalities of their keypoint lists, in other words, the first is the corresponding visterm with the largest number of similar keypoints to the target visterm.

After this stage, the final result set of similar visterms is created (see Table 4) and prepared for retrieving annotation.

5. Global features

5.1. 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) [42]. Global descriptors ensure generalization, for example, they are able to describe homogeneous regions in the image, such as *clear sky* and *sand*, which are regions that are usually ignored during detection of interest points. This problem is well illustrated in Figure 3, where we can see relatively homogenous regions in which no keypoints were detected.

The CEDD belongs to the group of Compact Composite Descriptors [43], 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: 768×512, 384×256, 192×128 and 96×64 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. The image is not changed, if its resolution is less than 96×64 pixels. For example, if the original image resolution is 672×504 pixels, than the image is interpolated to the resolution 768×512 pixels.

Subsequently, the image is divided into 8×8, 4×4 or 2×2 sub-images (segments) using the grid segmentation. The number of segments depends on image resolution. For example, an image with resolution 384×256 pixels is divided to 4×4 segments. The image resolution less than or equal to 96×64 pixels is canonical. After image segmentation, a global descriptor (CEDD) is calculated for each segment.

5.2. Indexing and clustering global features

Indexing and clustering of global features is very similar to the introduced indexing and clustering of local features. 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 family F of hash functions h is defined as follows:

$$F = h : 0, 1^d \rightarrow 0, 1 | h(x) = x_i, i = 1, ..., d, \quad (3)$$

where d is the dimension (432) of a descriptor x and x_i is the i-th element of x. A random function h from F simply selects a random bit from the descriptor.

The LSH parameters for indexing of global features are $L = 10$ and $k = 320$ (experimentally set). The maximum size of each BucketID is 40 bytes. The global descriptor identifier (GD identifier) is in the form ImageID_SegmentIndex (see Table 5). As with keypoints, descriptors are stored in separated table (see Table 6). The maximum size of each GD identifier is 23 bytes.

BucketID	ImgID_SegmIndex	ImgID_SegmIndex	...
1	1_2	1_3	...
2	1_11	1_10	...
...	...	...	...

Table 5: A layout of one LSH table for global features.

ImgID	SegmIndex	Descriptor	...
1	2	$[A_1, ..., A_{144}]$	...
	3	$[B_1, ..., B_{144}]$	...
	...	...	...
...	...	...	...

Table 6: A layout of a table for storing global features.

5.3. Querying the LSH global features index

The goal of this stage is to retrieve segments similar to segments of the target image similarly to querying for keypoints. For a target image:

1. Divide the target image into segments (using grid segmentation) and calculate descriptors for each one.
2. For each segment (descriptor):
   (a) calculate the L bucket identifiers (BucketIDs) using the corresponding hash functions g_i,
   (b) select all GD identifiers stored in the buckets "labelled" by the BucketIDs,
   (c) associate the corresponding GD identifiers distinctly with the target segment.
3. Group the returned GD identifiers according to ImgIDs.

After the query is executed, similar segments (candidates) to the target segments are retrieved as a result set (see Table 7). All target segments have assigned a list of corresponding (similar) segments. The similar segments are grouped according to ImgID.

Because LSH returns approximate matches, we need to check for segments (their descriptors) outside a threshold distance. For each similar segment, its descriptor is selected according to ImgID (see Table 6). Subsequently, all target segments are compared with the corresponding (similar) segments using *Tanimoto Distance* (see Eq. 2). After computing similarities, false matches are discarded by checking that the distance is over the threshold (experimentally set to 0.8). Subsequently, image candidates to the target image are sorted in descending order according to cardinalities of their lists of the similar segments, in other words, the first is the image candidate with the largest number of similar segments to the target segments.

After sorting, the final result set of similar segments is created (see Table 7) and prepared for retrieving annotation.

6. Obtaining annotation and relevance estimation

Our goal is to obtain annotation (tags) for extracted visterms and segments of the target image and to estimate probability for each tag. The estimated probability determines degree of accuracy with which the tag describes the visterm or segment. The annotation is obtained using training dataset, which contains well annotated images.

When people (users) manually annotate images, they are often influenced by scales of objects. They tend to focus on dominant and central objects and the sequence of the entered tags (keywords) may be influenced by these factors. Therefore, we prioritize annotations, where similar objects (visterms) to the target object (visterm) are dominant. We estimate relative importance of the tags based on their positions in the annotation. The first words are more important, because they may describe the dominant objects, i.e., the first tags often describe the central objects.

The training dataset is processed using methods described in previous sections. The obtaining annotation and estimation its relevance is performed in two steps, separately for local features (visterms) and global features (segments).

6.1. Obtaining annotation and estimation its relevance for visterms

First, from the target image I_T, target visterms TV_j are extracted. After extracting keypoits from each TV_j, similar visterms SV_{jk} to each one are searched for.

Let $T = \{TV_j : j = 1, ..., |TV|\}$ be the set of all the target visterms of I_T. Let TVK_j be the set of all the keypoints of TV_j.

Let $S_{TV_j} = \{SV_{jk} : k = 1, ..., |SV_j|\}$ bet the set of all the similar visterms to the target visterm TV_j. Let SVK_{jk} be the set of all the similar keypoints of SV_{jk} to TVK_j.

Each similar visterm $SV_{jk} \in S_{TV_j}$ inherits annotation from the original image, from which was extracted.

Let ASV_{jk} be the set of all tags of the inherited annotation of SV_{jk}. The probability $P_{ASV_{jk}}$, that the inherited annotation

Target Image	Image Candidates		Image Candidates		...
Target Segments (Indexes)	Similar segments (Indexes):Similarity	ImgID	Similar segments (Indexes):Similarity	ImgID	...
1	2:0.86; 3:0.82	1	1:0.83; 2:0.94	5	...
2	11:0.92		10:0.8		...
...	...	...	...	...	...

Table 7: A result set obtained via target segments queries.

ASV_{jk} of the similar visterm SV_{jk} describes the target visterm TV_j, is estimated as follows:

$$P_{ASV_{jk}}(TV_j, SV_{jk}) = \frac{\sum_i sim_T(TVK_{j,i}, SVK_{jk,i})}{|TVK_j|} * \frac{RA_{SV}}{|ASV_{jk}|}, \quad (4)$$

where $\sum_i sim_T(TVK_{j,i}, SVK_{jk,i})$ is a sum of the calculated Tanimoto distances (similarities) between all the keypoints of the target visterm TV_j and the corresponding keypoints of the similar visterm SV_{jk} (see Table 4); $|ASV_{jk}|$ is the cardinality of ASV_{jk}; $|TVK_j|$ is the cardinality of TVK_j; and $RA \in \langle 0, 1\rangle$ represents the percentage of the rectangular area which is bounded by the keypoints of the similar visterm SV_{jk} to the total area of the original image.

The calculated probability $P_{ASV_{jk}}(TV_j, SV_{jk})$ is assigned to each tag $t_{l_{jk}} \in ASV_{jk}$. Subsequently, for each tag $t_{l_{jk}}$, a factor of its relative importance is calculated as follows:

$$IFt_{l_{jk}} = \frac{1}{log_2(1 + {t_{l_{jk}}}^{pos})}, \quad (5)$$

where ${t_{l_{jk}}}^{pos}$ is the position of the tag $t_{l_{jk}}$ in the inherited annotation of SV_{jk}. Finally, $A_P SV_{jk} = \{t_{l_{jk}}[P_{ASV_{jk}}(TV_j, SV_{jk}) * IFt_{l_{jk}}] : l = 1, ..., |ASVjk|\}$ is the set of all the original tags of $ASVjk$ with the calculated probability multiplied by the corresponding factor of relative importance. The probability estimations are calculated for all annotations of similar visterms to the target visterm.

Let $A_pTV_j = \bigcup_k A_pSV_{jk}$ be the set of the tags obtained from all SV_{jk} for TV_j with the calculated probabilities. Let $t_i[p_{ti}] \in A_PTV_j$, where t_i is a tag (term) r. Then, $X_rTV_j = \{p_{t_i} : (t_i = r)[p_{t_i}]\}$ be the set of all the probabilities of the tag r. When comparing $(t_i = r)$, the stemming technique (natural language processing) is applied. For example, using a stemming algorithm, the words {*fishing*, *fished*, *fish*} are reduced to the root word, i.e., *fish*. The probability of the tag r is evaluated as follows:

$$P_r(X_rTV_j) = \frac{\sum_{p \in X_rTV_j} p}{|X_rTV_j|}, \quad (6)$$

where $\sum_{p \in X_rTV_j} p$ is a sum of the probabilities and $|X_rTV_j|$ is the cardinality of X_rTV_j. As we illustrate in Figure 4, for the target visterm α, at least two similar visterms $\alpha 1$ and $\alpha 2$ were found. The visterm α has 5 target keypoints.

The visterm $\alpha 1$ has 5 keypoints similar to the target keypoints. The probability $P_{A\alpha 1}(\alpha, \alpha 1)$, that the annotation $A\alpha 1 = \{t1, t2, t3\}$ of $\alpha 1$ describes α, equals $\frac{4.6}{5} * \frac{0.36}{3} \approx 0.11$ (see Eq. 4). This probability is assigned to all tags $t_i \in A\alpha 1$, i.e., $A_P\, alpha1 = \{t1[0.11], t2[0.11], t3[0.11]\}$. Subsequently, for each t_i, the factor of its relative importance is calculated (see Eq. 4), i.e., $t1[1], t2[0.63]$ and $t3[0.5]$. The probability of t_i is multiplied by the corresponding factor of relative importance of t_i, i.e., $A_P\alpha 1^* = \{t1[0.11], t2[0.07], t3[0.06]\}$.

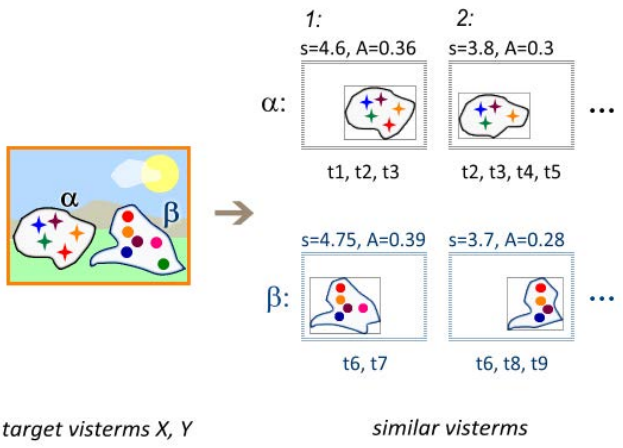

Figure 4: An illustration of the similar visterms $\alpha 1$ (similarity=4.6, area=0.36), $\alpha 2$ (similarity=3.8, area=0.3) to the target visterm α and the similar visterms $\beta 1$ (similarity=4.75, area=0.39), $\beta 2$ (similarity=3.7, area=0.28) to the target visterm β. All the similar visterms have assigned tags.

In a similar way, the visterm $\alpha 2$ has 4 keypoints similar to the target keypoints. The probability $P_{A\alpha 2}(\alpha, \alpha 2)$, that the annotation $A\alpha 2 = \{t2, t3, t4, t5\}$ of $\alpha 2$ describes α equals $\frac{3.8}{5} * \frac{0.3}{4} \approx 0.06$. This probability is assigned to all tags $t_j \in A\alpha 2$. For each t_j, the factor of its relative importance is calculated, i.e., $t2[1], t3[0.63], t4[0.5]$ and $t5[0.43]$. The probability of t_j is multiplied by the corresponding factor of relative importance of t_j, i.e., $A_P\alpha 2^* = \{t2[0.06], t3[0.04], t4[0.03], t5[0.03]\}$.

Finally, $A_P\alpha = \{t1[0.11], t2[0.07], t3[0.06], t2[0.06], t3[0.04], t4[0.03], t5[0.03]\}$ is the set of the tags obtained from $\alpha 1$ and $\alpha 2$ with the calculated probabilities; and $X_{t2}\alpha = \{0.07, 0.06\}$ is the set of the probabilities of $t2$. The tag $t2$ with the highest probability is selected as the top-estimated tag which describes α with the probability $P_{t2}(X_{t2}\alpha) = \frac{0.07+0.06}{2} \approx 0.07$ (see Eq. 6).

6.2. *Obtaining annotation and estimation its relevance for segments*

The target image I_T is divided into segments TS_j using the grid segmentation and from each one a descriptor is calculated (see Section 5.1). Let $T = \{TS_j : j = 1, ..., |TS|\}$ be the set of all the target segments of I_T. After calculating descriptors, from each target segment TS_j, similar segments $SS_{j,l}$ to each

one are searched for. Let $I_S = \{IS_i : i = 1, ..., |IS|\}$ be the set of images from which the similar segments were extracted. Then, the similar segments are grouped according to those images.

Let $SG_{j,ISi} = \{SS_{j,ISi,k} : k = 1, ..., |SS_{j,ISi}|$ be the set of the similar segments of IS_i to TS_j. The similar segments $SG_{j,ISi}$ inherit annotation from the original image. Let $ASG_{j,ISi}$ be the set of all tags of the inherited annotation of $SG_{j,ISi}$. The probability $P_{ASG_{j,ISi}}$, that the inherited annotation $ASG_{j,ISi}$ of the similar segments $SG_{j,ISi}$ describes the target segment TS_j, is estimated as follows:

$$P_{ASG_{j,ISi}}(TS_j, SG_{j,ISi}) = \frac{\sum_k sim_T(TS_j, SS_{j,ISi,k})}{CS} * \frac{1}{ASG_{j,ISi}}, \quad (7)$$

where $\sum_k sim_T(TS_j, SS_{j,ISi,k})$ is a sum of calculated Tanimoto distances (similarities) between the target segment TS_j and the similar segments $SS_{j,ISi,k} \in SG_{j,ISi}$ (see Table 7); CS is a constant of the number of segments in which the image IS was divided; and $|ASG_{j,ISi}|$ is the cardinality of $ASG_{j,ISi}$.

The calculated probability $P_{ASG_{j,ISi}}(TS_j, SG_{j,ISi})$ is assigned to each tag $t_{l_{j,ISi}} \in ASG_{j,ISi}$. Subsequently, for each tag $t_{l_{j,ISi}}$, a factor of its relative importance is calculated (see Eq. 5). Finally, $A_PSG_{j,ISi} = \{t_{l_{j,ISi}}[P_{ASG_{j,ISi}}(TS_j, SG_{j,ISi}) * IFt_{l_{j,ISi}}] : l = 1, ..., |ASG_{j,ISi}|\}$ is the set of all the original tags of $ASG_{j,ISi}$ with the calculated probability multiplied by the corresponding factor of relative importance. The probability estimations are calculated for all annotations of $ASG_{j,IS}$ to the target segment.

Let $A_PTS_j = \bigcup_{ISi} A_PSG_{j,ISi}$ be the set of the tags obtained from all $SG_{j,ISi}$ for TS_j with the calculated probabilities.Let $t_i[p_{t_i}] \in A_PTS_j$, where t_i is a tag (term) r. Then, $Y_rTS_j = \{p_{t_i} : (t_i = r)[p_{t_i}]\}$ be the set of all the probabilities of the tag r. When comparing $(t_i = r)$, the stemming technique (natural language processing) is applied. The probability of the tag r is evaluated as follows:

$$P_r(Y_rTS_j) = \frac{\sum_{p \in Y_rTS_j} p}{|Y_rTS_j|}, \quad (8)$$

where $\sum_{p \in Y_rTS_j} p$ is a sum of the probabilities and $|Y_rTS_j|$ is the cardinality of Y_rTS_j. As we illustrate in Figure 5, for the target segment $T[2, 1]$, at least four similar segments $\{J1[2, 1]; J1[2, 2]; J2[2, 1]; J2[2, 2]\}$, were found. The set of the similar segments of $J1$ to T is $SG_{T,J1} = \{J1[2, 1]; J1[2, 2]\}$ and the probability $P_{ASG_{T,J1}}(T, SG_{T,J1})$, that the annotation $ASG_{T,J1} = \{t6, t7, t8\}$ of $J1$ describes T, equals $\frac{(0.86+0.91)}{4} * \frac{1}{3} = 0.1475$ (see Eq. 7). This probability is assigned to all tags $t_i \in ASG_{T,J1}$. For each t_i, the factor of its relative importance is calculated (see Eq. 5), i.e., $t6[1]$, $t7[0.63]$ and $t8[0.5]$. The probability of t_i is multiplied by the factor of relative importance of t_i, i.e., $A_PSG_{T,J1}{}^* = \{t6[0.15], t7[0.09], t8[0.08]\}$.

Subsequently, the set of the similar segments of $J2$ to T is $SG_{T,J2} = \{J2[2, 1]; J2[2, 2]\}$ and the probability $P_{ASG_{T,J2}}(T, SG_{T,J2})$, that the annotation $ASG_{T,J2} = \{t7, t8\}$ of $J2$ describes T, equals $\frac{(0.82+0.84)}{4} * \frac{1}{2} \approx 0.21$. This probability is assigned to all tags $t_j \in ASG_{T,J1}$. For each t_j, the factor of its relative importance is calculated, i.e., $t7[1]$ and $t8[0.63]$. The probability of t_j is multiplied by the factor of relative importance of t_j, i.e., $A_PSG_{T,J2}{}^* = \{t7[0.21], t8[0.13]\}$.

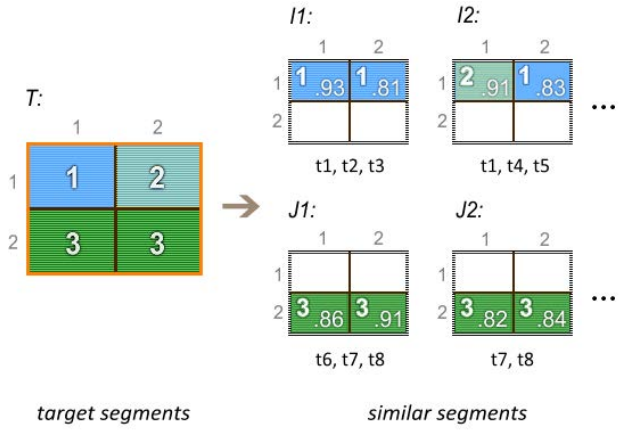

Figure 5: An illustration of the similar segments $I1[1, 1]$ (similarity=0.93), $I1[1, 2]$ (similarity=0.81), $I2[1, 2]$ (similarity=0.83) to the target segment $T[1, 1]$; the similar segment $I2[1, 1]$ (similarity=0.91) to the target segment $T[1, 2]$; and the similar segments $J1[2, 1]$ (similarity=0.86), $J1[2, 2]$ (similarity=0.91), $J2[2, 1]$ (similarity=0.82), $J2[2, 2]$ (similarity=0.84) to the target segments $T[2, 1]$, $T[2, 2]$. All the similar segments have assigned tags.

Finally, A_PT={$t6$[0.15],$t7$[0.09],$t8$[0.08],$t7$[0.21],$t8$[0.13]} is the set of the tags obtained from $J1$ and $J2$ with the calculated probabilities; and $Y_{t7}T$ ={0.09,0.21} is the set of the probabilities of $t7$. The tag ($t7$) with the highest probability is selected as the top-estimated tag which describe T with the probability $P_{t7}(Y_{t7}T) = \frac{(0.09+0.21)}{2} \approx 0.15$ (see Eq. 8).

6.3. Assigning annotation to the whole target image

For the target image, we obtain two sets of words (tags) with estimated probabilities, i.e., to each word a probability is assigned with which the word describes a part of the image (or the whole image). We sum the probabilities of the same words and we calculate its arithmetic mean. Subsequently we need to select the top n words. There is a problem, how to determine the number of the selected words, i.e., how to establish a threshold.

The number of selected words can depend on level of homogeneity of the target image. In other words, the number of selected words should reflect the number of extracted visterms from the target image. However, there are some problems, such as very heterogeneous images and several words may describe the same visterm (e.g. words describing properties such as color, shape).

Consider the case, where an image with *colorful flower* is automatically divided into many regions (visterms) due to its complexity of visual structure. If the automatic annotation contains fewer words than the number of the extracted visterms, we could select all of the words and consider that the annotation is not complete for some visterms (objects of interest). However, some words of the annotation may be incorrectly predicted. If the annotation contains more words than the number of visterms, we could automatically select the number of words equal to the number of visterms. However, several visterms may represent the same object due to preservation of high details during the image segmentation; on the contrary, the small number of visterms may be due to ignoring high details during the image segmentation; some words may describe properties;

and some words may be incorrectly predicted. Therefore, even if the words are sorted according to probabilities, there may occur a situation that some correct words will be excluded, or vice-versa, incorrect words will be selected.

Therefore, for each word w_i^Λ of the obtained annotation Λ, we compute so-called *co-occurrence rank* $\rho_{w_i^\Lambda}$ which reflects whether the word w_i^Λ occurs together with other words of Λ in training annotations and how often. The training annotations consist of word-lists, where each one is associated with a training image (for more information about the training dataset, see the next section Evaluation). Based on the word-lists, we create so-called term-by-term (co-occurrence) matrix.

First, a vocabulary V is created. It consists of distinct words w_i selected from the word-lists. Second, a co-occurrence matrix C is constructed, where rows and columns correspond to the words w_i of the vocabulary V, i.e., the matrix is squared ($m \times m$). Finally, for each $w_i \in V$, words w_j occurred with w_i in the word-lists are searched for.

Let $C_{m\times m}$ be an m-by-m matrix, where $1 \leq i, j \leq |V|$ and $|V|$ is the cardinality of V, then each element $[a_{i,j}]$ of $C_{m\times m}$ represents the number of occurrences of the word i with the word j in the word-lists. Once the co-occurrence matrix is constructed, for each $w_i^\Lambda \in \Lambda$, $\rho_{w_i^\Lambda}$ is calculated as follows:

$$\rho_{w_i^\Lambda} = \frac{|\Lambda|}{|W_{i,J}|} * \sum_{j \in W_{i,J}} a_{i,j}, \tag{9}$$

where $|\Lambda|$ is the number of the words of Λ, $W_{i,J}$ is the set of all the words w_j^C such, that $[a_{i,j}]_{j\in\Lambda} \neq 0$, $|W_{i,J}|$ is the cardinality of $W_{i,J}$, and $a_{i,j}$ is the number of the occurrence of w_i^Λ with w_j^C. Subsequently, all the calculated $\rho_{w_i^\Lambda}$ are normalized into the interval $\langle 0, 1\rangle$. The final probability of w_i^Λ is calculated as the product of its original probability and the *co-occurrence rank* $\rho_{w_i^\Lambda}$. After calculating the final probabilities, all the words of Λ with the zero probability are discarded.

7. Evaluation

The annotation problem can be understood as the problem of retrieving an image from the test set using words from the test vocabulary. To evaluate annotation performance, we retrieve images using the test keywords. Subsequently, we compare the automatic obtained keywords with the ground-truth (manual) annotations provided along with the test images.

To evaluate the annotation performance, we use the *precision* and *recall* 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 the manual annotation. Then *precision* is defined as follows:

$$P = \frac{B}{A}, \tag{10}$$

and *recall* is defined as follows:

$$R = \frac{B}{C} \tag{11}$$

We report mean precision (P) and mean recall (R), as well as the number of total keywords recalled (N^+), the number of keywords with non-zero recall value).

We argue that practical annotation system that address the applications discussed above should satisfy the following requirements:

- *High recall.* All images in the image database that contain subimages (visterms) that are present in the query image should be found, even if the visterms only occupy a small portion of the query image.
- *High precision.* If the image database and the query image do not have visterms in common, then they should not be matched.
- *Efficiency.* The time needed to query an image should be small, enabling the system to scale to large databases.

Our aim is to evaluate how well our approach satisfies these requirements compared with the state-of-the-art approaches.

Our experiments use a 2.8GHz Intel® Core™ i7 machine with 12GB of memory running Windows 7. The presented methods are implemented in C++.

7.1. Datasets

7.1.1. Corel5K dataset

The Corel5K corpus [5] consists of 5,000 images from 50 Corel Stock Photo CDs and each CD includes 100 images with the same theme. It includes a variety of subjects, ranging from urban to nature scenes and from artificial objects to animals. It is divided into two sets: a training set of 4,500 images and a test set of 500 images. Each image is associated with $1-5$ keywords (an average of 3.5 keywords per image). Overall there are 260 keywords that appear in both the train and the test set. All images have the resolution of 384×256 pixels (landscape) or 256×384 pixels (portrait).

7.1.2. Corel5K-Visterms dataset

As we have mentioned in Section 1, some approaches to obtain an annotation to the target image are based on searching correlations between the target image and labelled templates (training visterms). We have constructed a dataset of visterms from the original Corel5K dataset[3]. Our aim is to compare the annotation performance of our approach (ANNOR) on the original Corel5K dataset with ANNOR's annotation performance on the Corel5k-Visterms dataset.

For the construction of Corel5k-Visterms dataset we implemented a web application called *ANNO*^{R} (ANNOtation instRument[4], see Figure 6). It allows manual creating labelled visterms (from the training Corel5K images). For our experiment, we asked 30 users to create the labelled visterms from the original training Corel5K set using *ANNO*^{R}. For a user, there was randomly selected an image (1) with keywords (2), which

[3] Corel5K-Visterms dataset: http://annor.laude.sk/dataset
[4] *ANNO*^{R}: http://annor.laude.sk

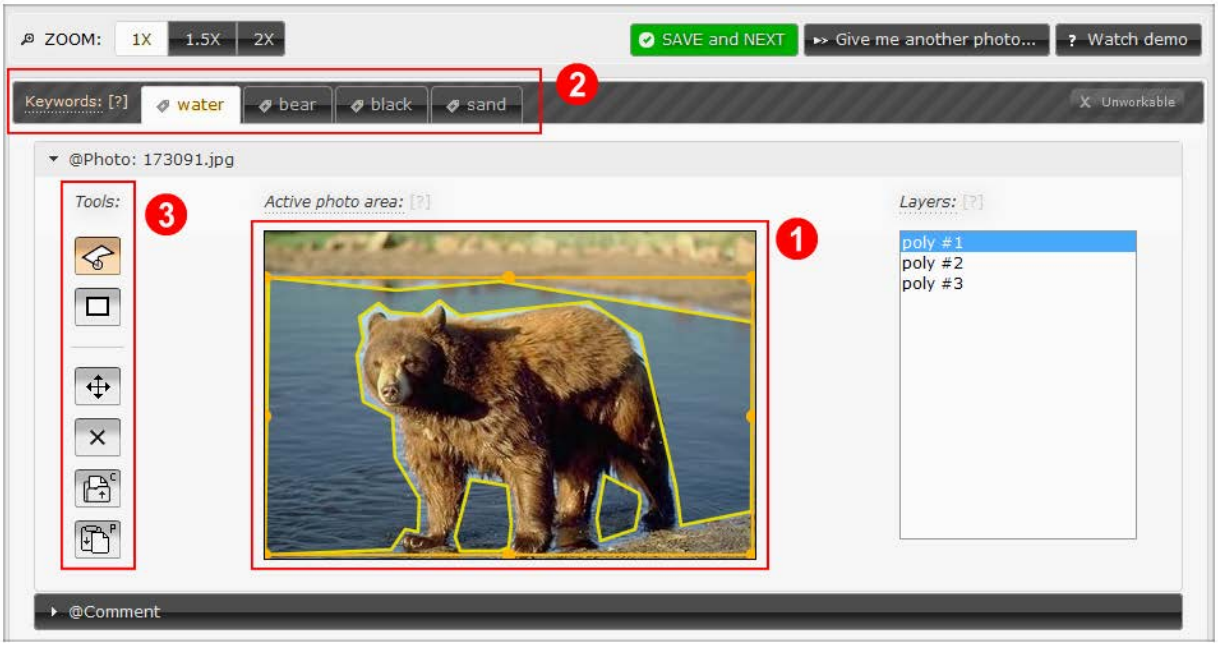

Figure 6: The user interface of *ANNO*^{R} for creating labelled visterms. (1) a selected image, (2) descriptive words associated with the image, (3) tools for creating bounding boxes for the keywords.

Figure 7: Examples of labelled visterms obtained using *ANNO*^{R}(*ANNOtationinstRument*).

describe its visual content. Using tools (3) such as irregular polygon, rectangle, move, remove, copy, paste, the user created bounding boxes (polygons) for each word separately. For example, for the word water, bounding polygons around all areas with the water were created. For each word, the tool provides a new instance of an active drawing area. Thus, for each word, the user has available a clear area to create the polygons. When switching between keywords, the created content is preserved. After saving, labelled visterms are obtained using created polygons (see Figure 6). This resulted in a dictionary containing 19, 739 polygons (visterms).

7.1.3. IAPR-TC12 dataset

The IAPR-TC12 dataset[5] is a collection of 19,805 images of natural scenes that include different sports, landscapes, people, cities, animals and so on. The images in IAPR-TC12 are associated with free-typing text captions. It was initially published for cross-lingual retrieval. We use the same resulting annotation as in [25]. It is divided into two sets: a training set of 17, 825 images and a test set of 1, 980 images. Overall there are 291 keywords (an average of 4.7 keywords per image) that appear in both the train and the test set.

7.1.4. ESP dataset

The ESP dataset[6] consists of a set of 21,844 images collected in ESP game[7]. ESP game is an online game where two players assign labels to the same image without communicating. They gain points by agreeing on labels describing the target image. As an image is shown to more teams, it has a list of so called *taboo* words, that is, words that cannot be entered as possible labels. In other words, once an image has been labelled enough times with the same word, that word becomes taboo. This dataset contains a wide variety of images such as logos, drawings, and personal photos. It is divided into two sets: a training set of 19, 659 images and a test set of 2, 185 images. Overall there are 269 keywords that appear in both the train and the test set. Each image is associated with up to 15 keywords (an average of 4.6 keywords per image).

7.2. Annotation performance

The results of experiments on the Corel5k dataset are summarized in Table 8. The table provides an overview of annotation performance in terms of P, R, and N^+ between our approach (ANNOR) and a selection of work. The table shows the published results of the current state-of-the-art methods that approach the annotation problem from different perspectives, using different image representations: CRM [6], InfNet [11], NPDE [12], MBRM [10], SML [20], JEC [25], and TagProb

[5]IAPR-TC12 dataset: http://imageclef.org/photodata

[6]ESP dataset: http://hunch.net/ jl/

[7]ESP game: http://www.cs.cmu.edu/ biglou/

Test image				
Automatic annotation	**iguana** (1.00) **marine** (0.94) **lizard** (0.91) **rocks** (0.82) water (0.61)	**kauai** (1.00) sand (0.94) tree (0.91) **people** (0.80) hawaii (0.72)	**formula** (1.00) **tracks** (1.00) **cars** (1.00) **wall** (0.89) arch (0.77)	**foals** (1.0) **horses** (0.94) **field** (0.91) meadow (0.82) **mare** (0.81)
Human annotation	**iguana, lizard, marine, rocks**	**kauai, people**	**cars, formula, tracks, wall**	**foals, horses, field,** fence, **mare**

Table 10: Examples of automatic annotation (the best five keywords with probabilities) compared with the human annotation.

	Corel5K-Original			Corel5K-Visterms		
Method	P	R	N^+	P	R	N^+
CRM [6]	16	19	107	-		
IfNet [11]	17	24	112			
NPDE [12]	18	21	114			
MBRM [10]	24	25	122			
SML [20]	23	29	137			
JEC [25]	27	32	139			
TagProp [27]	**33**	**42**	**160**			
ANNOR-L	19	27	68	23	30	74
ANNOR-G	22	29	129	27	33	143
ANNOR-LG	***31***	***38***	***154***	**37**	**48**	**172**

Table 8: An overview of annotation performance in terms of P, R, and N^+ between our approach (ANNOR) and a selection of earlier work on the Corel5K dataset. ANNOR-L is our approach using local features only. ANNOR-G is our approach using global features only. ANNOR-LG is our approach using both local and global features. We also show results concerning the annotation performance of ANNOR on the Corel5K-Visterms dataset.

	IAPR-TC12			ESP		
Method	P	R	N^+	P	R	N^+
MBRM [10]	24	23	223	18	19	209
JEC [25]	28	29	250	22	25	224
TagProp [27]	*46*	*35*	*266*	***39***	27	239
ANNOR-L	22	19	98	19	21	86
ANNOR-G	38	31	242	36	29	231
ANNOR-LG	***48***	***39***	***272***	**39**	**28**	**241**

Table 9: An overview of annotation performance in terms of P, R, and N^+ between our approach (ANNOR) and a selection of earlier work on the IAPR-TC12 and ESP datasets. ANNOR-L is our approach using local features only. ANNOR-G is our approach using global features only. ANNOR-LG is our approach using both local and global features.

[27]. ANNOR-L is our approach using local features only. ANNOR-G is our approach using global features only. ANNOR-LG is our approach using both local and global features. Table 9 shows the results of experiments on the IAPR-TC12 and ESP datasets. We compare ANNOR with MBRM [10], JEC [25], and TagProp [27]. The results of experiments on the Corel5k dataset are summarized in Table 8. The table provides an overview of annotation performance in terms of P, R, and N^+ between our approach (ANNOR) and a selection of work. The table shows the published results of the current state-of-the-art methods that approach the annotation problem from different perspectives, using different image representations: CRM [6], InfNet [11], NPDE [12], MBRM [10], SML [20], JEC [25], and TagProb [27]. ANNOR-L is our approach using local features only. ANNOR-G is our approach using global features only. ANNOR-LG is our approach using both local and global features. Table 9 shows the results of experiments on the IAPR-TC12 and ESP datasets. We compare ANNOR with MBRM [10], JEC [25], and TagProp [27].

As we can see ANNOR-LG outperforms the state-of-the-art methods on both IAPR-TC12 and ESP datasets. In the case of Corel5K-Original dataset, only TagProp slightly outperforms ANNOR-LG.

We can also see that our combination of the local and global features is very important (ANNOR-LG outperforms both ANNOR-L and ANNOR-G). As we have mentioned in Section 1 local features are suitable for search of specific objects while global features capture complex information. It ensures *robustness* and *generalization*, i.e., based on the combination ANNOR-LG is resistant to common transforms and it is able to describe relatively homogeneous regions. The result is that the annotation performance is higher.

The bottom rows of Table 9 contain the comparison of the ANNOR's annotation performance on the Corel5K-Original dataset with the ANNOR's annotation performance on the Corel5K-Visterms dataset. As we can see ANNOR's performance is better on the Corel5K-Visterms dataset. It has two main reasons. First, each manually created visterm has assigned clear label, therefore, the process of obtaining annotation is more accurate. Second, comparing similar visterms based on its visual content is also more accurate due to the "consistency" of the content.

In Table 10, there are shown some illustrative examples of the automatic annotation obtained by our approach compared with the human (manual) annotation (Corel5K corpus). As we can see, in the images 1 and 4, there are some keywords wrongly predicted. In the image 1, there is the keyword *water* wrongly predicted and in the image 4, there is the keyword *meadow* wrongly predicted. However, in both cases the keywords are not completely wrong, because their meaning is relatively close to their visual content. As we have mentioned in

	Corel5K-Original	Corel5K-Visterms	IAPR-TC12	ESP
#test set	500	500	1,980	2,185
#training set	4500	4500	17,825	19,659
#local features	903,056	990,250	3,814,550	4,855,773
#global features	72,002	19,805	285,200	314,544
image/annotation (ms)	1,750	1,140	2,950	3,430

Table 11: Statistics of the image datasets: #test set - a number of images in the test set, #training set - a number of images in the training set, #local features - a number of local features extracted from the training images, #global features - a number of global features extracted from training images, image/annotation (ms) - average time needed to obtain an annotation for the target image from the test set.

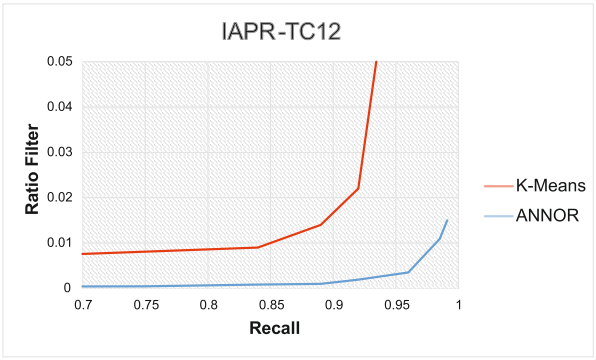

Figure 8: Comparison of our method and K-Means algorithm for range-neighbors search on the IAPR-TC12 dataset.

Table 12: Examples of transformations.

Section 1, the manual annotation task is subjective and therefore our aim is to propose a method that can find the best deal between annotating "nothing" and "everything". In Table 11 we summarize statistics of the image datasets.

7.3. Evaluation as a nearest neighbors method

To evaluate our LSH-based method we can consider the method as a filtering algorithm. In general, for a given target feature, we retrieve a subset of features that are potential neighbors, i.e., for a feature q, we retrieve all the database features x_i such that $d(q, x_i) < R$ (see Sections 4.3 and 5.3).

We evaluate our approach with respect to these two criteria: (1) the size of the retrieved subset of features and (2) the number of true neighbors found in this subset. The first criterion we evaluate as a ratio of the size of the retrieved subset and the total number of features in the database (Ratio Filter). For example, a ratio Filter of 0.01 means that only 1% of all features in the database is returned and thus the corresponding method can be at best 100 times faster than a linear search. The second criterion is measured as a classical recall. We compare our method with K-Means algorithm on the IAPR-TC12 dataset. Results are shown in Figure 8. We can see that our method outperforms K-Means.

7.4. Evaluation as an image retrieval method

The approach we use to evaluate our method is to retrieve images that are synthetic deformations of a set of images. The goal is to detect near-duplicates, i.e., if a given (target) image is a modification of a database image. We have established a list of transformations that our image retrieval mechanism should handle. The 11 transformations we use are changing contrast, colorizing, cropping, rotating, scaling, applying emboss filter, sharpness filter, median filter, and Gaussian blur, changing saturation, and intensity (see examples in Table 12). Our test framework is to pick 100 random images from the image datasets Corel5K and IAPR-TC12 (individually) and to apply the deformations to these images. The 100 original images will be used as queries. For each one, the method should only return the 11 deformations of the query image. The other images in the image datasets are so called "perturbators". The aim is to evaluate the robustness of our approach (invariance to geometrical changes). We compare our method with K-Means algorithm which uses kmeans clustering of the extracted features, learned on the image datasets.

The results are shown in Table 13 (Corel5K) and Table 14 (IAPR-TC12) respectively. The conclusions of this experiment are as follows. Our method outperforms K-means in recall and time. As we have mentioned, the main drawback of K-Means-based approach is that it is not adapted for regularly updated databases. If the database is created incrementally, one has to update the visterms frequently to guarantee optimal performance. This is not required by our approach. However, KMeans-based approaches are much more memory effective. Our approach requires several times more memory than the K-Means method. For example, ANNOR requires approximately 110 MiB for 1,000 images while the K-Means-based method requires approximately 9 MiB for 1,000 images. The reason is that features are indexed multiple times by LSH (it depends on the L parameter).

Method		Recall	Time (ms)
ANNOR-LG	k=320,l=10	0.939	623
K-Means	k = 4000	0.895	2180
	k = 8000	0.914	2485
	k = 16000	0.935	2803

Table 13: Image retrieval: ANNOR-LG compared with K-Means on the Corel5K dataset.

Method		Recall	Time (ms)
ANNOR-LG	k=320,l=10	0.952	1433
K-Means	k = 4000	0.875	7420
	k = 8000	0.884	8150
	k = 16000	0.922	8930

Table 14: Image retrieval: ANNOR-LG compared with K-Means on the IAPR-TC12 dataset.

8. Discussion and conclusion

In our approach, we combine global and local features. The local features are very successful for problems involving retrieval of target objects (objects of interest). They exhibit very good robustness to moderate scaling, brightness changes and "in-plane" rotation. The global features capture the entire information of an image (e.g. texture, color). Both have advantages and drawbacks, the local features are much more precise than global features and their discrimination ability is relatively high. When looking for a target object, this ability is welcome, however, when looking for a general category (e.g. find all yellow Ferrari), it may cause restrictions. A method for automatic image annotation should be able to ensure both the requirements, namely, robustness and generalization and it was one of our goals. Because we use a combination of local and global features, our approach is resistant to common transforms. Traditional approaches based on global features cannot cope with.

A potential drawback of using local features is that we need to store and index a huge number of extracted features and there is a need to query hundreds to thousands of features which could be slow. This "side-effect" often causes performance issues (e.g. approaches based on k Nearest Neighbors search) and it limits using large-scale image (training) datasets. For this problem, we employed efficient solution through locality sensitive hashing which is based on the idea that similar objects are stored to the same bucket. We have also adopted the distributed database management system Cassandra that was specially designed for storing the huge number of data. For efficient access to extracted data, we have designed data layouts for using with LSH. On the one hand, our solution provides a good compromise between precision and speed; it allows random access to stored data (in sub-linear time); and index is generated dynamically. On the other hand, KMeans-based algorithms (either flat or hierarchical) much more memory effective compared with LSH.

The important part of automatic image annotation is obtaining an annotation for the target image and estimation probabilities of particular words in the annotation. In this process, we have focused on the way, how people manually annotate images. When people form an annotation, they may be influenced by scales of objects, they focus primarily on dominant, significant (glaring) and central objects. The sequence of the words is influenced by these factors. Therefore, we prioritize annotations of training images, where similar objects (visterms) to the target object (visterm) are dominant. It is more likely that a word of an annotation describes the target object. For the words, we estimate their relative importance based on their positions in the annotation, i.e., their importance decreases exponentially. The first words are more important, because they may describe the dominant objects. Finally, for the obtained words, we estimate, whether these words occur together in the training annotations and how often. Based on this, we discard "isolated" (incorrectly predicted) words. The result is, that each word of the obtained annotation has assigned a probability of its relevance and they are sorted based on their relative importance.

As we have mentioned, combining of local and global features is able to ensure robustness and generalization needed by complex queries. However, we observed in the evaluation that the process of obtaining annotations using local features predicts often noisy results. Even if the used SIFT method is very successful for recognizing rigid (specific) objects, its performance on more general object classes is not satisfactory. For example, the word *sky* obtained the high precision but the recall is very low. On the contrary, words such as *ocean*, *desert* have the high recall but the low precision. It means that, these words can be easily predict but with the very low accuracy.

Thus, in a more general object database, extraction of global information is more important than the local information, because the global features capture more important information of images (e.g. color, texture) than the local features.

In the process of obtaining annotations using global features, we observed that some words such as *sun*, *sunset*, *sky* have high precision and high recall. It means that the words can be easily predicted with a high probability that it is a true annotation. Such visterms are more easily recognizable because of their relative homogeneity (characteristic color and texture) than other visterms (*flowers*, *tiger*).

The correct recognition of invariant objects depends on a quality of the used training dataset. In a more specific object database, the combination of the local and global features can better show its potential.

Acknowledgements

This work was partially supported by grants No. VG 1/0752/14 and it is a partial result of the Research and Development Operational Programme for the project International centre of excellence for research of intelligent and secure information-communication technologies and systems, ITMS 26240120039, co-funded by the ERDF.

References

[1] Eitz M, Hildebrand K, Boubekeur T, Alexa M. An evaluation of descriptors for large-scale image retrieval from sketched feature lines. Computers & Graphics 2010;34(5):482 –98. CAD/GRAPHICS 2009 Extended

papers from the 2009 Sketch-Based Interfaces and Modeling Conference Vision, Modeling & Visualization.

[2] Eitz M, Hays J, Alexa M. How do humans sketch objects? ACM Trans Graph 2012;31(4):44:1–44:10.

[3] Authors preliminary paper published in proceedings of international workshop. ????

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

[5] Duygulu P, Barnard K, Freitas JFGd, Forsyth DA. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In: Proceedings of the 7th European Conference on Computer Vision-Part IV. ECCV '02; London, UK, UK: Springer-Verlag. ISBN 3-540-43748-7; 2002, p. 97–112.

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

[7] Jeon J, Lavrenko V, Manmatha R. Automatic image annotation and retrieval using cross-media relevance models. In: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval. SIGIR '03; New York, NY, USA: ACM. ISBN 1-58113-646-3; 2003, p. 119–26.

[8] Liu J, Wang B, Li M, Li Z, Ma W, Lu H, et al. Dual cross-media relevance model for image annotation. In: Proceedings of the 15th International Conference on Multimedia. MULTIMEDIA '07; New York, NY, USA: ACM. ISBN 978-1-59593-702-5; 2007, p. 605–14.

[9] Zhen Y, Yeung DY. A probabilistic model for multimodal hash function learning. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '12; New York, NY, USA: ACM. ISBN 978-1-4503-1462-6; 2012, p. 940–8.

[10] Feng SL, Manmatha R, Lavrenko V. Multiple bernoulli relevance models for image and video annotation. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR'04; Washington, DC, USA: IEEE Computer Society; 2004, p. 1002–9.

[11] Metzler D, Manmatha R. An inference network approach to image retrieval. In: Enser P, Kompatsiaris Y, OConnor N, Smeaton A, Smeulders A, editors. Image and Video Retrieval; vol. 3115 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg. ISBN 978-3-540-22539-3; 2005, p. 42–50.

[12] Yavlinsky A, Schofield E, Rüger S. Automated image annotation using global features and robust nonparametric density estimation. In: Proceedings of the 4th International Conference on Image and Video Retrieval. CIVR'05; Berlin, Heidelberg: Springer-Verlag. ISBN 3-540-27858-3, 978-3-540-27858-0; 2005, p. 507–17.

[13] Andrews S, Tsochantaridis I, Hofmann T. Support vector machines for multiple-instance learning. Advances in neural information processing systems 2003;15.

[14] Cusano C, Ciocca G, Schettini R. Image annotation using svm. Proceedings of SPIE Conference on Internet Imaging V 2004;5304:330–8.

[15] Savakis A. A computationally efficient approach to indoor/outdoor scene classification. In: Proceedings of the 16 th International Conference on Pattern Recognition (ICPR'02) Volume 4 - Volume 4. ICPR '02; Washington, DC, USA: IEEE Computer Society. ISBN 0-7695-1695-X; 2002, p. 40146–.

[16] Zhang L, Ma J. Image annotation by incorporating word correlations into multi-class svm. Soft Comput 2011;15(5):917–27.

[17] Patterson G. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In: Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). CVPR '12; Washington, DC, USA: IEEE Computer Society. ISBN 978-1-4673-1226-4; 2012, p. 2751–8.

[18] Li J, Wang JZ. Automatic linguistic indexing of pictures by a statistical modeling approach. IEEE Trans Pattern Anal Mach Intell 2003;25(9):1075–88.

[19] Llorente A, Manmatha R, Rüger S. Image retrieval using markov random fields and global image features. In: Proceedings of the ACM International Conference on Image and Video Retrieval. CIVR '10; New York, NY, USA: ACM. ISBN 978-1-4503-0117-6; 2010, p. 243–50.

[20] Carneiro G, Chan AB, Moreno PJ, Vasconcelos N. Supervised learning of semantic classes for image annotation and retrieval. IEEE Trans Pattern Anal Mach Intell 2007;29(3):394–410.

[21] Chang E, Goh K, Sychay G, Wu G. Cbsa: content-based soft annotation for multimodal image retrieval using bayes point machines. Circuits and Systems for Video Technology, IEEE Transactions on 2003;13(1):26–38.

[22] Vailaya A, Figueiredo MA, Jain AK, Zhang HJ. Image classification for content-based indexing. Trans Img Proc 2001;10(1):117–30.

[23] Wang XJ, Zhang L, Jing F, Ma WY. Annosearch: Image auto-annotation by search. In: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2. CVPR '06; Washington, DC, USA: IEEE Computer Society. ISBN 0-7695-2597-0; 2006, p. 1483–90.

[24] Wang C, Jing F, Zhang L, Zhang HJ. Scalable search-based image annotation of personal images. In: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval. MIR '06; New York, NY, USA: ACM. ISBN 1-59593-495-2; 2006, p. 269–78.

[25] Makadia A, Pavlovic V, Kumar S. A new baseline for image annotation. In: Proceedings of the 10th European Conference on Computer Vision: Part III. ECCV '08; Berlin, Heidelberg: Springer-Verlag. ISBN 978-3-540-88689-1; 2008, p. 316–29.

[26] Babenko B, Branson S, Belongie S. Similarity metrics for categorization: from monolithic to category specific. Int Conf on Computer Vision (ICCV) 2009;.

[27] Guillaumin M, Mensink T, Verbeek J, Schmid C. Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation. In: Computer Vision, 2009 IEEE 12th International Conference on. 2009, p. 309–16.

[28] Szummer M, Picard RW. Indoor-outdoor image classification. In: Proceedings of the 1998 International Workshop on Content-Based Access of Image and Video Databases (CAIVD '98). CAIVD '98; Washington, DC, USA: IEEE Computer Society. ISBN 0-8186-8329-5; 1998, p. 42–.

[29] Yi W, Tang H. Experimental analysis on classification of unmanned aerial vehicles images using the probabilistic latent semantic analysis. Int Symposium on Spatial Analysis, Spatial-Temporal Data Modeling, and Data Mining 2009;7492(3).

[30] Blei DM, Jordan MI. Modeling annotated data. In: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval. SIGIR '03; New York, NY, USA: ACM. ISBN 1-58113-646-3; 2003, p. 127–34.

[31] Yang C, Dong M, Fotouhi F. Region based image annotation through multiple-instance learning. In: Proceedings of the 13th Annual ACM International Conference on Multimedia. MULTIMEDIA '05; New York, NY, USA: ACM. ISBN 1-59593-044-2; 2005, p. 435–8.

[32] Tang J, Lewis PH. Using multiple segmentations for image auto-annotation. In: Proceedings of the 6th ACM International Conference on Image and Video Retrieval. CIVR '07; New York, NY, USA: ACM. ISBN 978-1-59593-733-9; 2007, p. 581–6.

[33] Sivic J, Zisserman A. Video google: A text retrieval approach to object matching in videos. In: Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2. ICCV '03; Washington, DC, USA: IEEE Computer Society. ISBN 0-7695-1950-4; 2003, p. 1470–.

[34] Lowe DG. Distinctive image features from scale-invariant keypoints. Int J Comput Vision 2004;60(2):91–110.

[35] Philbin J, Chum O, Isard M, Sivic J, Zisserman A. Object retrieval with large vocabularies and fast spatial matching. In: Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on. 2007, p. 1–8.

[36] Bay H, Ess A, Tuytelaars T, Van Gool L. Speeded-up robust features (surf). Comput Vis Image Underst 2008;110(3):346–59.

[37] Mikolajczyk K, Schmid C. A performance evaluation of local descriptors. IEEE Trans Pattern Anal Mach Intell 2005;27(10):1615–30.

[38] Felzenszwalb PF, Huttenlocher DP. Efficient graph-based image segmentation. Int J Comput Vision 2004;59(2):167–81.

[39] Foo JJ, Sinha R. Pruning sift for scalable near-duplicate image matching. In: Proceedings of the Eighteenth Conference on Australasian Database - Volume 63. ADC '07; Darlinghurst, Australia, Australia: Australian Computer Society, Inc. ISBN 1-920-68244-9; 2007, p. 63–71.

[40] Fonseca MJ, Jorge JA. Towards content-based retrieval of technical drawings through high-dimensional indexing. Computers & Graphics 2003;27(1):61 –9.

[41] Datar M, Immorlica N, Indyk P, Mirrokni VS. Locality-sensitive hashing scheme based on p-stable distributions. In: Proceedings of the Twentieth

Annual Symposium on Computational Geometry. SCG '04; New York, NY, USA: ACM. ISBN 1-58113-885-7; 2004, p. 253–62.

[42] Chatzichristofis SA, Boutalis YS. Cedd: Color and edge directivity descriptor: A compact descriptor for image indexing and retrieval. In: Proceedings of the 6th International Conference on Computer Vision Systems. ICVS'08; Berlin, Heidelberg: Springer-Verlag. ISBN 3-540-79546-4, 978-3-540-79546-9; 2008, p. 312–22.

[43] Chatzichristofis S, Zagoris K, Boutalis Y, Papamarkos N. Accurate image retrieval based on compact composite descriptors and relevance feedback information. International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI) 2010;24(2):207–44.