Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 6, June 2015, pg.1071 – 1074

RESEARCH ARTICLE

Hash Code Based Image Indexing and Retrieval

Ms. Snehal S. Bhamre
Department of Computer Engineering, KKWIEER, Pune University, India
snehalbhamre51@gmail.com

Abstract— the rapid evolution of multimedia and application has brought about an explosive growth of digital images in computer vision. Increasing use of image acquisition and data storage technologies have enabled the creation of large database. This development has actually increase need for image retrieval system. So, it is necessary to develop appropriate information management system to efficiently manage these collections and needed a system to retrieve required images from these collections. This paper proposed a Hash code based image indexing and retrieval system to retrieve images similar to the query image specified by user from database. Hashing methods embed high-dimensional image features into hamming space performing real time search based on hamming distance. Depending upon minimum hamming distance it returns the similar image to query image.

Keywords— Hash code, image retrieval, High-dimensional image feature, Hamming space, Hamming distance.

I. INTRODUCTION

Image databases and collections can be enormous in size, containing hundreds, thousands or even millions of images. Traditional image search mechanisms highly rely on textual words associated to the images, scalable content based image search is becoming popular. Apart from providing better image search experience for ordinary Web users, large-scale similar image search has also been demonstrated to be very helpful for solving a number of very hard problems in computer vision and multimedia such as image categorization.

The text-based approach requires a previous annotation of the database images, which is very lengthy and time consuming. Furthermore, the annotation process is inefficient because, different users tend to use different keywords to describe the same image characteristics. The lack of systemization in annotation process decreases the performance of text-based image retrieval. The alternative content-based method indexes images in database by identifying similarities between them based on low-level visual features as colour, texture, shape and spatial information.

Feature extraction is the basis of content-based image retrieval. These features may include both text-based features (key words, annotations) and visual features (colour, texture, shape, faces). In this work, images are represented using the popular bag-of-visual-words (BoW) framework, where local invariant image descriptors are extracted and quantized based on a set of visual words. The BoW features are then embedded into compact hash codes for efficient search. For this, a hashing technique including semi-supervised hashing and semantic hashing with deep belief networks is considered. Hashing is preferable over tree-based indexing structures as it generally requires greatly reduced memory and also works better for high-dimensional samples. With the hash codes, image similarities can be efficiently measured.

II. LITERATURE SURVEY

There are many surveys on general image retrieval task. Many people adopted simple features such as colour and texture in systems developed in the early years, while more effective features such as GIST [1] and SIFT [2] have been popular recently. Traditional image search engines heavily rely on textual words associated to the images, scalable content-based search is receiving increasing attention. Apart from providing better image search experience for ordinary Web users, large-scale similar image search has also been demonstrated to be very helpful for solving a number of very hard problems in computer vision and multimedia such as image categorization.

Inverted index was initially proposed and is still very popular for document retrieval in the informational retrieval community [3].A key difference of document retrieval from visual search, however, is that the textual queries usually contain very few words.

Indexing with tree-like structure has been frequently applied to fast visual search. Nister and Stewenius [5] used a visual vocabulary tree to achieve real-time object retrieval in 40000 images. Muja and Lowe [6] adopted multiple randomized d-trees [7] for SIFT feature matching in image applications. One drawback of the classical tree-based methods is that they normally do not work well with high-dimensional feature.

In view of the limitations of both inverted file and tree-based indexing, embedding high dimensional image features into hash codes has become very popular recently. Hashing satisfies both query time and memory requirements as the binary hash codes are compact in memory and efficient in search via hash table lookup or bitwise operations. For this, supervised and semi-supervised hashing is used. All these hashing methods have one limitation when applied to image search. The Hamming distance of hash codes cannot offer fine-grained ranking of search results, which is very important in practice.

III. SYSTEM ARCHITECTURE

The block diagram of system is as shown below:

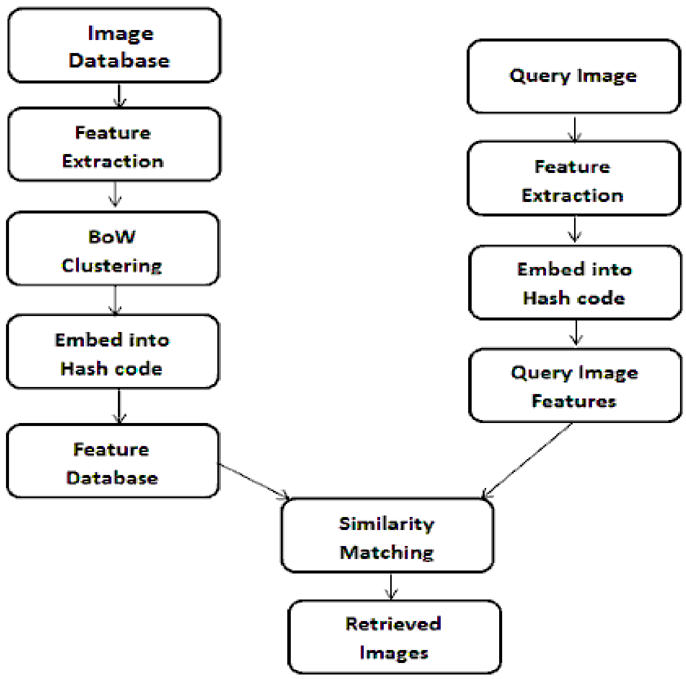

Fig. 1 System Architecture

Function of each block is discussed below:

- **Image database**: It is the original dataset of images.
- **Feature extraction**: It is used for extracting features of the images stored in dataset as well as that of query image fired by the user. It uses SIFT (Scale Invariant Feature Transform) algorithm for feature extraction. SIFT is an algorithm for extracting stable feature description of objects call keypoints that are robust to changes in scale, orientation, shear, position, and illumination.
- **BoW Clustering:** In the bag of words model, the feature space is divided into clusters (or words). The feature space is divided by applying the k-means clustering algorithm to the SIFT features descriptor. Then each descriptor is assigned to one or more clusters with closest centers (in metrics like Euclidean).

Instead of storing a whole descriptor cluster number is stored. A bag of words is a sparse vector of occurrence counts of words (or clusters).

- **Embed into hash code:** BoW features are quantized and embedded into hash codes. The quantization strategy is that if a feature component is larger than the mean of this vector, it is quantized to 1, otherwise to 0. A 32-bit hash code is generated.
- **Feature database**: it is used to store the hash code.
- **Query image feature**: For a given query image, similarly extract its features and form a feature vector.
- **Similarity Matching**: It is the process which compares query image feature vector with already store feature vectors in image database. It is based on some similarity measure to calculate distance between the query image feature vector and feature database.
- **Retrieved images**: Depending on similarity measure it generates a list retrieves images.

IV. ALGORITHM

For implementation of this system following algorithms have been used:

1. **SIFT algorithm:**
   SIFT algorithm consist of following steps:
   Step 1: Scale-space extrema detection: this step will search overall scales and image locations. It is implemented efficiently by using a difference-of-Gaussian function.
   Step 2: Keypoint localization: At each location, a detailed model is fit to determine location and scale and accordingly Keypoints are selected.
   Step 3: Orientation assignment: One or more orientations are assigned to each keypoint location based on local image gradient directions.
   Step 4: Keypoint descriptor: The local image gradients are measured and are transformed into a representation that allows for significant levels of local shape distortion and change in illumination.
2. **BoW Algorithm:**
   Step 1: Define the number of bags (or we can say classes).
   Step 2: Cluster the set of feature descriptors for the amount of bags that are defined.
   Step 3: Train the bags with clustered feature descriptors.
   Step 4: Obtain the visual vocabulary.

After this, the feature vector is converted into hash codes by using inbuilt Hash functions of OpenCv.

V. RESULTS AND DISCUSSION

A. Data Set

The system is evaluated on the Corel dataset. This Database is composed of 1000 colour images. The size of all images is either 256 × 384 or 384 × 256 pixels. The dataset consist of 10 different categories of image and each category has 100 images of similar type. This dataset is widely used as a benchmark for many image indexing and image retrieval algorithms. Sample images from the Corel dataset in 10 categories vis African, Beach, Buildings, Dinosaur, Bus, Elephant, Rose, Horse, Mountain and Dish are shown in Fig.

B. Result Set

In Image retrieval system, the most commonly used measures for the performance evaluation are Precision and Recall. Precision is defined as the ratio of the number of relevant images retrieved to the total number of images retrieved.

P = (number of relevant images retrieved) / (total number of images retrieved)

Recall is defined as the ratio of the number of relevant images retrieved to the total number of relevant images in the database.

R = (number of relevant images retrieved) / (total number of relevant images in the database.)

The work done was tested for images from the Corel dataset. Here we computed the percentage precision for images of 10 different categories at each level of recall. Table shows the result for images from the Corel Dataset.

TABLE I
AVERAGE PRECISION FOR IMAGES FROM COREL DATASET

% Recall	African	Beach	Building	Bus	Dinosaurs	Elephant	Rose	Horse	Mountain	Food
10	71.21	85.15	84.09	87.09	100.0	76.94	84.86	81.42	76.75	78.13
20	68.34	75.52	77.76	78.76	100.0	73.04	82.85	76.50	74.85	77.79
30	67.52	70.14	76.26	73.45	100.0	68.00	78.83	76.15	69.84	70.51
40	64.46	63.07	72.50	71.56	100.0	66.34	72.66	72.50	65.99	65.24
50	59.51	58.91	63.76	66.54	97.73	61.46	68.70	65.21	59.23	59.70
60	54.68	54.39	60.35	62.45	92.71	51.95	66.62	63.18	55.62	53.30
70	47.70	49.53	53.52	56.88	89.74	46.77	60.64	56.78	50.60	44.56
80	42.55	44.32	44.64	51.41	83.86	41.38	53.16	47.52	45.36	38.08
90	34.14	35.69	32.86	40.47	75.11	34.39	42.32	40.52	36.54	31.44
100	20.06	21.46	20.08	24.77	65.87	20.54	23.54	20.94	21.93	19.74

VI. CONCLUSIONS

A novel framework for query-adaptive image search with hash codes is presented. By harnessing a large set of predefined semantic concept classes, the approach is able to predict query-adaptive bitwise weights of hash codes in real-time, with which search results can be rapidly ranked by at finer-grained hash code level. This capability largely alleviates the effect of a coarse ranking problem that is common in hashing-based image search.

One can further extend framework for query-adaptive hash code selection. Instead of image specific codes, the class specific codes can further improve search performance significantly. One drawback is that nontrivial extra memory is required by the use of additional class-specific codes, and therefore a careful examination of the actual application is needed and hardware environment in order to decide whether this extension could be adopted.

REFERENCES

[1] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *Int. J. Comput. Vision*, vol. 42, pp. 145–175, 2001.

[2] B. Kulis and K. Grauman, "Kernelized locality-sensitive hashing for scalable image search," in *Proc. IEEE Int. Conf. Computer Vision*, 2009.

[3] B. Kulis and T. Darrell, "Learning to hash with binary reconstructive embeddings," in *Adv. Neural Inf. Process. Syst.*, 2009.

[4] D. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[5] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2006.

[6] H. Jegou, M. Douze, and C. Schmid, "Packing bag-of-features," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2009.

[7] *J. L. Bentley, "Multidimensional binary search trees used for associative searching," Commun. ACM, vol. 18, no. 9, pp. 509–517, 1975.*