Journal of Engineering and Applied Sciences 13 (Special Issue 9): 7068-7073, 2018
ISSN: 1816-949X

A Tree-Like Image Representation for Efficient Image Storage and Retrieval

Muhammad Suzuri Bin Hitam, Pong Kuan Peng, Wan Nural Jawahir Hj Wan Yussof, Abdul Aziz K Abdul and Ghazali Sulong
School of Informatics and Applied Mathematics, Universiti Malaysia Terengganu, 21030 Kuala Terengganu, Malaysia

Abstract: With the rapid advancement of information and communication technology, the number of images and multimedia contents have increased drastically. Therefore, there is an urgent need for efficient storage and retrieval technique for digital images as well as multimedia contents. This study presents a new approach for storing and retrieval of binary images in a tree-like image data structure. The proposed approach works in three main stages, determining a unique primary low level binary image pattern, storing them into a tree-like image data structure and finally retrieving similar look images from the tree-like image data structure. By determining a set of unique primary low level binary image patterns there will be no redundant patterns are stored in the tree-like image data structure thus leading to tremendously save storage space while the retrieval of similar look image is simply accomplished by finding closest similar patterns inside the tree-like image data structure. The proposed approach provides tremendous saving in storage space and promising results to be used for fast image retrieval.

Key words: Binary image, storage, retrieval and data structure, increased drastically, multimedia contents, efficient storage

INTRODUCTION

The rapid development in digital image technology has led to the availability of extremely huge amount of digital images. Therefore, there is an urgent need for an effective method for storing as well as retrieving similar look images from image database. Thus, reseachers from diverse field of researches have done much efforts in proposing many methods for efficient and fast retrieval of digital images. However, the research community in image retrieval have yet, to find a truely effective storage and retrieval technique despite over a decade of research in this area. Basically an effective storage and image retrieval system needs to operate on the large collection of images to store and retrieve the relevant images in short time period based on the query given by the user which must conforms as closely as possible to user perception and requirement. Due to the complexity and subjectivity of the different user's perception and requirement, the process of digital image retrieval becomes very challenging. To date, the most commonly used method to store images in image database is by using a saparate image file although, they are exactly similar image, thus, leads to inefficient management of storage space.

Due to the nature of images that contain large amount of information, more storage capacity and transmission bandwidth were needed to store and transmit images across the internet (Dhawan, 2011). As a matter of fact, some of the images may contain redundant information. To avoid this problem, this study introduced a new method to store and retrieve binary images for more efficient storage and retrieval. The method works on storing only unique low level binary pattern in a new image data structure. In this proposed approach, a set of basic unique low level binary patterns will be determined from a given image. Combination of these patterns will be stored in a new tree-like image data structure. Image retrieval is achieved by simply searching through the tree-like image data structure. In this way, it saves tremendous storage space due to no redundant information were stored and searching for similar look images is just by finding closest unique image patterns in the tree-like image data structure to the query image.

Literature review: During the past decade, image retrieval has been an active and fast advancing research area. In the late 1970's, image retrieval generally based on textual annotation of images (Tamura and Yokoya, 1984). Images were annotated with text and searched through traditional database management systems. However, text-based image retrieval technique requires manual annotation of images thus it becomes subjective and expensive task for large image database (Rui *et al*., 1999). In the early 1990's,

Corresponding Author: Muhammad Suzuri Bin Hitam, School of Informatics and Applied Mathematics, Universiti Malaysia Terengganu, 21030 Kuala Terengganu, Terengganu, Malaysia

advances in the internet and digital imaging made text-based image retrieval become more difficult and insufficient. To overcome this problem, Content-Based Image Retrieval (CBIR) had emerged as an alternative in image retrieval.

Content-based image retrieval extracted image features such as color texture and shapes to index image. These image features were stored in the image database. To retrieve images, a query image is submitted to the system and the query image is matched with similar images in the database using some similarity measurements. The similarity measurement will determine the distance between query images and indexed images in database and then rank them according to their closest distance.

Although, many CBIR techniques have been explored these low level-features still provide gap with high-level human perception. Humans use high level features such as keyword and text descriptors to interpret images (Liu *et al*., 2007). This high-level perception is not easily expressible in term of low-level features. Studies had shown that low-level image descriptors predominantly fail to describe high-level human perception (Alzu *et al*., 2015). Recently, there is an effort made to combine low-level features with high-level features to represent images semantically in order to improve retrieval accuracy. However, bridging the semantic gap between low-level features and high level features is a very challenging task to be solved. (Pandey *et al*., 2016) presented a content based semantic and image retrieval system for semantically hierarchical clustering large image databases. By Fu *et al*. (2000) employed image visual features categorized as nodes in image tree where each node has a set of visual signatures formed with color texture and shape. To reduce search time, visual signatures of a node are arranged using three vantage point trees that build index based on distance to vantage point. Rank of nodes are computed based on visual signatures and it is obtained by arranging nodes in ascending order of distances. Nodes that match specific semantic is retrieved.

High dimensional indexing mechanism is one of the main factors of archiving fast and accurate image retrieval (Ai *et al*., 2013). To retrieve images efficiently, huge amount of visual images stored in computer memory need to be organized. High dimensional indexing mechanism can be divided into three categories which are tree-based index, hashing-based index and visual words based inverted index. Tree-based index represents the data space of images to form a hierarchical tree structure. The non-leaf nodes act as directory nodes that storing the information of data space and the leaf nodes storing the information that need to be indexed. The existing popular tree-based index are KD-tree (Silpa-Anan and Hartley, 2008) R-tree (Rallabandi and Sett, 2007), M-tree (Skopal and Lakoc, 2009) and EHD-tree (Zhang *et al*., 2009).

Researchers also, used vocabulary tree to index features vector. Subrahmanyam *et al*. (2012) proposed a CBIR combining color histogram and spatial orientation tree wavelet coefficients. Vector points of each image are indexed using vocabulary tree. Gonde *et al*. (2013) used a modified curvelet transform using the ridgelet and curvelet transform. Descriptor vectors of each image are indexed using vocabulary tree. Even though indexing tree speeding up retrieval, as features space dimension increase, the indexing efficiency is decreased. Qian and Tagare (2010) employed two tree adaption which are metric-based and coordinate-based indexing trees to improve indexing efficiency.

The Dynamically Rearranged High Dimensional Data-tree (Priya and Vadivel, 2012) (DRHDD-tree) is constructed with the capability to reduce dimensionality based on the occurrences of vectors. In DRHDD-tree, the features vectors are clustered by patterns to avoid features vectors overlapping in high-dimensional space. The extracted feature vectors are represented as a branch using dynamically rearranged prefix tree (Priya *et al*., 2012). To achieve compact form the tree structure can be dynamically rearranged by merging the common vectors thus consume less memory space. A linked-node m-ary tree (Pham *et al*., 2015) (LM-tree) is constructed to produce the queries for improvement in the search performance. In this study, features vector space is indexed by using the LM-tree algorithm. The LM-tree is constructed by recursively partitioning the image dataset into roughly equal-sized subsets. In this algorithm, two axes are employed for dividing the image data where as many existing tree-based indexing algorithms only employs one axis to partition the data.

Tree-based index can maintain order while hash-based indexing could not maintain order and just based on mapping the search-key values on a collection of buckets. The bucket to which a value is mapped is determined by hash function (Moro *et al*., 2009). Hash-based indexing projects feature data from high to low dimensional using hash function that can be categorized into main two groups, sensitive hashing and spectral hashing. Sensitive hashing[12] projects feature Euclidean data into low dimensional Euclidean space while spectral hashing (Pandey *et al*., 2016) maps close features data point in Euclidean space to similar binary codes in a low dimensional Hamming space.

Visual words based inverted index was introduced from text retrieval system (Indyk and Motwani, 1998) . In text retrieval approach called Bag-of-Words (BoW), each documents is represented by a vector with the frequency of occurrence of the words contains in the document. All of the vector representing document will be organized as inverted files (Shekhar and Jawahar, 2012). Sivic and Zisserma (2003) used Scale-Invariant Features Transform (SIFT) technique (Lowe, 1999) to extract descriptor and vector quantize the descriptors into visual words for retrieval. The descriptor is indexed by an inverted file. Instead of using vector with the frequency of occurrence directly for indexing, a standard weighting called Term Frequency Inverse Document Frequency (TF-IDF) is used to determine relevance of query vector and all document vector in the database. Nister and Stewenius (Nister and Stewenius, 2006) proposed a hierarchical TF-IDF scoring to form a hierarchical vocabulary tree. Both researchers are inspired by Sivic and Zisserman's text retrieval concept in retrieving key frame and shots from a video. Both methods are evaluated through retrieval on a database with group of images of the same object or location but under different view point, rotation, scale and lighting conditions. The results show that Nister and Stewenius's method of hierarchically quantizes descriptor from image allow more efficient lookup of visual word in large vocabulary compared with the previous works.

Most of these works still used original image stored in the image database, thus, making storage size requirement becomes vital when huge image collection is involved. One of the challenges in image retrieval is to scale down the size of image database and retrieve image data out of large database in acceptable time. This paper addresses this particular problem in storing and retrieving binary image files.

MATERIALS AND METHODS

Tree-like image representation: This study presents the proposed image representation structure. The overall structure of the approaches is performed in three steps. Firstly, a set of unique primary low level binary image is determined. Next, the unique primary low level binary images are organized and stored into a tree-like data structures for image representation. Finally, the tree-like image data structure will act as dictionary for encoding and decoding image. This means that all images will be stored in a single tree-like data structure during encoding process and retrieval is performed by decoding process. The encoder and decoder will be using the same dictionary and the decoder can recover the image information from the same execution path.

A tree structure is a powerful tool for storing and manipulating all kind of data. It has been used across many disciplines for storing and analyzing scientific or business data. In this study, it will be used to store basic unique image pattern, i.e., an image pattern will no longer be stored in a separate manner but all image patterns will be systematically stored in a complex tree-like data structure. The new approach of representing the image in a tree-like data structure uses no redundant information of image pattern, thus, there will be no repeated pattern stored in this newly proposed image representation. This approach not only will lead to fast image retrieval but it also, will results in saving the memory storage of the image pattern.

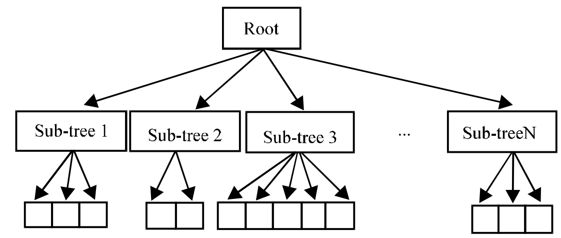

Fig. 1: Example of tree-like structure of image pattern representation

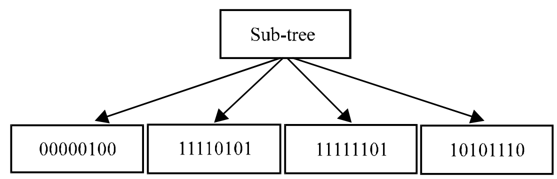

Fig. 2: Example of 2×4 binary image pattern in sub-tree

Table 1: Example of node path for sub-tree

Node path	Binary image pattern
a	0000 0100
b	1111 0101
c	1111 1101
d	1010 1110

The tree-like data structure consists of many unique image pattern sub-trees; Refer to graphical illustration in Fig. 1. Each sub-tree will represent unique 16×16 basic binary image patterns. As can be observed in Fig. 1, each sub-tree will hold different number of unique primary low level binary image patterns. This is the result of redundant low level binary image patterns that had been removed from the respective sub-trees. The combination of unique low level binary image pattern will create a new 16×16 image pattern.

To ease the prototype implementation, a 2×4 image resolution has been selected as the unique primary low level binary image pattern. As can be seen in Fig. 2, each node in the sub-tree represents 2×4 unique primary low level binary image pattern. Each of the 2×4 low level binary image patterns will be given a unique node path labeled as indicated in Table 1.

Table 2: Example of image pattern from a sub-tree

Pattern names	Basic image pattern
Tr1P1	abcdaaaaaaaaaaabdaaaaaaaaaaaaaa
Tr1P2	abbbadaaaaaaaaacaaaaabbbbbbbbbb
Tr1P3	bbbbbbbbbbaaaacaadddaaaaaaaaaaa

A list of combination of 2×4 binary image patterns in the sub-tree will act as 16×16 basic primary image patterns. A unique code name or pattern name will represent each basic image patterns. This code name will be used in the encode words during the encoding process. As can be seen from Fig. 2, each sub-tree has four unique 2×4 binary image patterns. This four unique 2×4 binary image patterns can create a few combinations to become new basic image pattern as shown in Table 2.

Before image encoding process is taken place, the respective image is partitioned into a square a 16×16 image blocks. The image block consists of redundant binary image pattern, thus a unique 2×4 binary image patterns will be extracted to create a new sub-tree. This is the reason why sub-tree has different number of nodes, as each image block have different number of unique 2×4 binary image patterns. The image blocks will be going through matching process with the image dictionary. Each image block will be assigned a code word matched with the basic image pattern. Each code word will have its pattern name. The combination of pattern name will represent the entire image. Example of combination of pattern name is 'Tr2P1, Tr3P2, Tr1P1, ... Decoding process will use the list of pattern name to decode the image. In the decode process, empty blocks will be created to fill in the list of pattern name. Each patternname will go through the image pattern dictionary looking for a match and the empty blocks will be filled. Algorithm 1and 2 show the encoding and decoding algorithms, respectively. The final tree will be composed of many of the sub-tree-like image representation data structures.

Algorithm 1; The encoding algorithm:

Input	Image M
Step 1	Partition image M into N blocks of 16×16 images called image blocks
Step 2	A unique 2×4 binary image patterns will be extracted from image blocks Match each image pattern to the tree-like data structure image dictionar The search return results of combination nodes from the image dictionary
Step 3	Go to step 2 until N blocks of image block matched A list of pattern name is created
Step 4	Check the list of pattern name for red redundancy. If pattern name redundant, Remove the redundant pattern nam Only keep the location index of pattern name
Output	Encoded Image E

Algorithm 2; The decoding algorithm:

Input	Encoded Image E
Step 1	Create N empty block of images M
Step 2	Match each code word to the tree like data structure image dictionary. The search return results of 2×4 binary image from the image dictionary the empty block with the binary Images
Step 3	Go to STEP 2 until N blocks of image block filled
Output	Image M

RESULTS AND DISCUSSION

In this study, the performance of the proposed method is tested by using MPEG-7 core experiment CE shape-1 test set. The dataset consists of 1400 binary images with 70 categories of similar look objects. To ease the implementation, all the images were resize into 256×256 resolution. Figure 3 shows samples of images from the dataset.

Figure 4 shows the comparison of total image size conventionally stored in the image database and the new encoded image size in a tree-like data structure. It can be clearly seen from this figure that the proposed algorithm is capable of reducing the use of storage space very effectively. Most of the images are able to be reduced to <50% of its original size. The reason of proposed approach shows tremendous reduction in image storage size is due to the way patterns were stored. Only the unique non-redundant image patterns were stored in tree-like image data structure representation. In this way, the total of image size is tremendously reduced when the image number is increased. It can also be seen that an increment in the total image size of the encoded images is less consistent compared to total size of the original image. This results is due to certain images contains less redundant image patterns as compared to others, thus, leading the image size didn't reduce proportionately.

To retrieve similar images from the tree-like image data structure or simply called as image tree, a Euclidean distance is defined to measure the difference between the query image and image stored in image tree as in Eq. 1:

$$d_{x,y} = \sqrt{\sum_{n=1}^{N} (x_n - y_n)^2} \quad (1)$$

Where:

x = Query image
y = Image tree
N = Total number of image patterns

To simplify the computation process, each image pattern's frequency of occurrence have been normalized

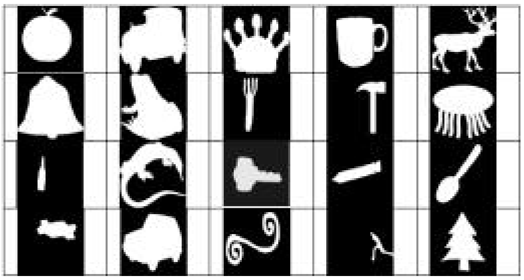

Fig. 3: Sample images from 20 categories of the dataset

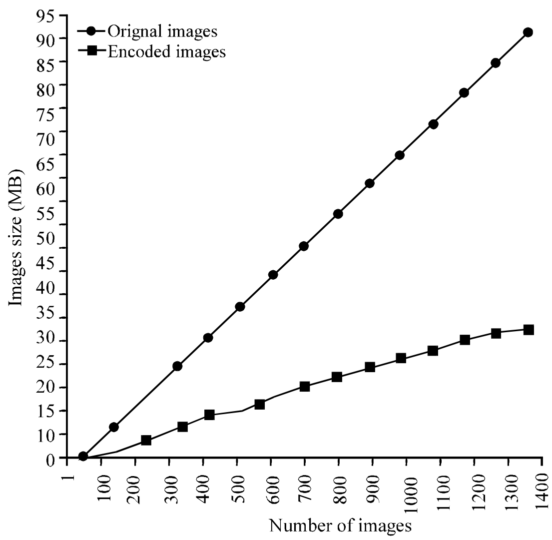

Fig. 4: Comparison of original image size with encoded image size

into the range of [0, 1]. Figure 5 shows the top 10 ranking of similar images retrieved based on the given query image. Most of the query images are able to retrieve similar images category, especially, retrieving images with similar object size and object orientation. However, when the objects shape is almost similar it will be more difficult to retrieved similar look objects due to the fact that after splitting the image, the search space consists of almost similar image patterns. This phenomena have resulted in retrieving other images category as similar category objects to the query image. Thus, it somehow reduces the accuracy of the image retrieval. From query image number 5, 6, 8, 10, 12, 19 and 20 it can be observed that some of the retrieved images did not fall in the same category. For example, in the case of image query number 5 it can only able to retrieved 3 out of 10 images. This retrieval results is perhaps due to other retrieved image have closely similar patterns with those patterns in the query image and thus caused not relevant object category have been retrieved. In the future, we proposed further experimental

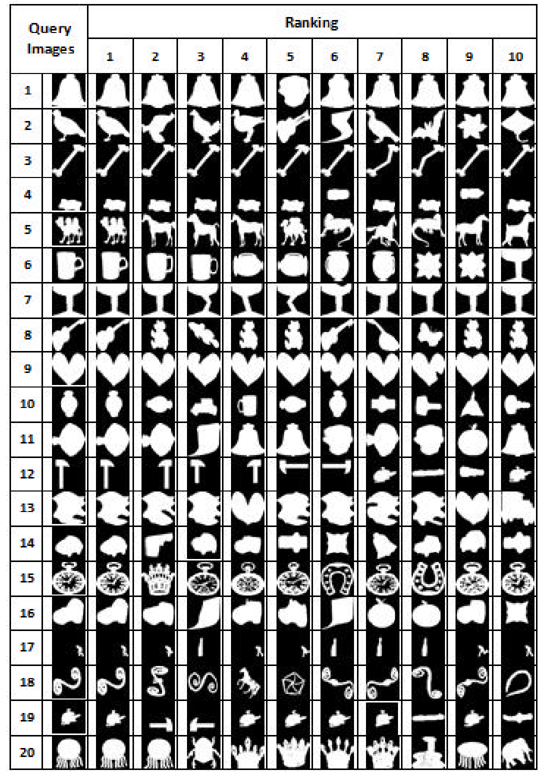

Fig. 5: Top 10 ranked of image similarity retrieval results for 20 query images from different image categories

investigation have to be carried out on the partitioning size of low level image patterns as well as the technique for systematic storage of similar look patterns in the image tree.

CONCLUSION

In this studies, a new approach is introduced to represent an image in a tree-like data structure, so that, storage, indexing and retrieval are facilitated. Experimental results using benchmark binary image database showed that the proposed method could significantly reduce image storage size while providing a promising approach for retrieving similar look objects from the newly introduced image data structure representation. The newly proposed method only store the unique image patterns into the tree-like image data structure. This approach results in saving the memory storage of the image pattern while at the same time retrieving similar look image from the proposed image data structure is efficient by just searching for closest frequency of similar pattern stored in the image tree.

RECOMMENDATIONS

However, there are still many other challenges that need to be addressed in the future, i.e., scale invariant, translation invariant, robustness to noise as well as affine transformation.

ACKNOWLEDGEMENTS

The researchers acknowledge the financial support from Ministry of Higher Education Malaysia under Fundamental Research Grant Scheme (FRGS) (vot no. 59288) to carry out this research.

REFERENCES

Ai, L.F., J.Q. Yu, Y.F. He and T. Guan, 2013. High-dimensional indexing technologies for large scale content-based image retrieval: A review. J. Zhejiang Univ. Sci. C., 14: 505-520.

Alzu, B.A., A. Amira and N. Ramzan, 2015. Semantic content-based image retrieval: A comprehensive study. J. Visual Commun. Image Represent., 32: 20-54.

Dhawan, S., 2011. A review of image compression and comparison of its algorithms. Int. J. Electron. Commun. Technol., 2: 22-26.

Fu, A.W.C., P.M.S. Chan, Y.L. Cheung and Y.S. Moon, 2000. Dynamic vp-tree indexing for n-nearest neighbor search given pair-wise distances. VLDB. J. Int. J. Very Large Data Bases, 9: 154-173.

Gonde, A.B., R.P. Maheshwari and R. Balasubramanian, 2013. Modified curvelet transform with vocabulary tree for content based image retrieval. Digital Signal Processing, 23: 142-150.

Indyk, P. and R. Motwani, 1998. Approximate nearest neighbors: Towards removing the curse of dimensionality. Proceedings of the 30th Annual ACM Symposium on Theory of Computing, May 24-26, 1998, Dallas, TX., USA., pp: 604-613.

Liu, Y., D. Zhang, G. Lu and W.Y. Ma, 2007. A survey of content-based image retrieval with high-level semantics. Pattern Recogn., 40: 262-282.

Lowe, D.G., 1999. Object recognition from local scale-invariant features. Proc. 7th IEEE Int. Conf. Comput. Vision, 2: 1150-1157.

Moro, M.M., D. Zhang and V.J. Tsotras, 2009. Hash-Based Indexing, in Encyclopedia of Database Systems. Springer, Boston, Massachusetts,.

Nister, D. and H. Stewenius, 2006. Scalable recognition with a vocabulary tree. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Volume 2, June 17-22, 2006, New York, USA., pp: 2169-2178.

Pandey, S., P. Khanna and H. Yokota, 2016. A semantics and image retrieval system for hierarchical image databases. Inf. Process. Manage., 52: 571-591.

Pham, T.A., S. Barrat, M. Delalandre and J.Y. Ramel, 2015. An efficient tree structure for indexing feature vectors. Pattern Recognit. Lett., 55: 42-50.

Priya, R.V. and A. Vadivel, 2012. Incremental indexing for high-dimensional data using tree structure. Procedia Technol., 6: 540-547.

Priya, R.V., A. Vadivel and R.S. Thakur, 2012. Maximal pattern mining using fast CP-tree for knowledge discovery. Int. J. Inf. Syst. Soc. Change, 3: 56-74.

Qian, X. and H.D. Tagare, 2010. Adapting indexing trees to data distribution in feature spaces. Comput. Vision Image Understanding, 114: 111-124.

Rallabandi, V.S. and S.K. Sett, 2007. Image retrieval system using R-tree self-organizing map. Data Knowl. Eng., 61: 524-539.

Rui, Y., T.S. Huang and S.F. Chang, 1999. Image retrieval: Current techniques, promising directions and open issues. J. Vis. Commun. Image Represent., 10: 39-62.

Shekhar, R. and C.V. Jawahar, 2012. Word image retrieval using bag of visual words. Proceeding of the10th International Workshop on Document Analysis Systems (DAS), March 27-29, 2012, IEEE, Hyderabad, India, ISBN:978-0-7695-4661-2, pp: 297-301.

Silpa-Anan, C. and R. Hartley, 2008. Optimised KD-trees for fast image descriptor matching. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 23-28, 2008, Anchorage, AK., USA., pp: 1-8.

Sivic, J. and A. Zisserman, 2003. Video Google: A text retrieval approach to object matching in videos. Proceedings of the 9th International Conference on Computer Vision, October 13-16, 2003, Nice, France, pp: 1470-1477.

Skopal, T. and J. Lokoc, 2009. New dynamic construction techniques for M-tree. J. Discrete Algorithms, 7: 62-77.

Subrahmanyam, M., R.P. Maheshwari and R. Balasubramanian, 2012. Expert system design using wavelet and color vocabulary trees for image retrieval. Expert Syst. Applic., 39: 5104-5114.

Tamura, H. and N. Yokoya, 1984. Image database systems: A survey. Pattern Recogn., 17: 29-43.

Zhang, X., Z. Li, L. Zhang, W.Y. Ma and H.Y. Shum, 2009. Efficient indexing for large scale visual search. Proceeding of the IEEE 12th International Conference on Computer Vision, September 29-October 2, 2009, IEEE, New York, USA., ISBN:978-1-4244-4420-5, pp: 1103-1110.