Tripartite Graph Models for Multi Modal Image Retrieval

Chandrika Pulla
chandrika@research.iiit.ac.in
C. V. Jawahar
jawahar@iiit.ac.in

Center for Visual Information Technology
IIIT-Hyderabad
Hyderabad, INDIA
http://cvit.iiit.ac.in

Abstract

Most of the traditional image retrieval methods use either low level visual features or embedded text for representation and indexing. In recent years, there has been significant interest in combining these two different modalities for effective retrieval. In this paper, we propose a tri-partite graph based representation of the multi model data for image retrieval tasks. Our representation is ideally suited for dynamically changing or evolving datasets, where repeated semantic indexing is practically impossible. We employ a graph partitioning algorithm for retrieving semantically relevant images from the database of images represented using the tripartite graph. Being "just in time semantic indexing", our method is computationally light and less resource intensive. Experimental results show that the data structure used is scalable. We also show that the performance of our method is comparable with other multi model approaches, with significantly lower computational and resources requirements.

1 Introduction

With the development of Internet, the size of online digital image collections is increasing rapidly. In many of these repositories, images get tagged or annotated by users. Such textual tags remain the primary method for accessing/searching such image collections. Therefore the necessity for efficient as well as effective retrieval methods for large scale dynamic image collections, is on the rise. The current image retrieval systems are of two types: text-based and content-based. In text based approaches, images are manually annotated by text descriptors which are then used by search engines to perform real-time retrieval [1, 17]. However, the cost of accurate annotation is very high. Also the whole process suffers from subjectivity of the annotations. To address these fundamental limitations of text based methods, content based image retrieval was introduced in late 80s. In traditional content based retrieval, images are indexed by their visual content such as color, textures, shape, spatial relationships etc. The research in this area is well established. However the retrieval effectiveness is still bottlenecked by the semantic gap [20]. In general, there is no direct relationship between the high-level visual concepts and the low-level image features. Semantic Indexing techniques like Latent Semantic Indexing (LSI),Probabilistic Latent Semantic Analysis (pLSA) and Latent Dirichlet Allocation (LDA) were proposed to improve the retrieval performance by reducing or bridging the semantic gap. These are unsupervised methods where a document is viewed as collection (bag) of words. A set of latent concepts or hidden topics is then introducted between words and documents. A generative model is first learnt. The learnt model is then used for mapping the problem from an input space to a novel feature space. It is believed that this new representation is closer to the semantic description.

Image retrieval has matured a lot in the recent years. On one end of the spectrum, we see successful laboratory prototypes to retrieve similar images from large image collections based on visual bag of words (BoW) model [13, 19]. On the other end of the spectrum, we see commercial systems with very rich user base sharing photographs, and enabling browsing based on manually attached textual tags [1]. Most of the current retrieval systems use either text or visual features in isolation. However, in many practical cases, information available is richer and consists of both these modalities. For example web pages contains text, imagery and other forms of information. Thus, image retrieval systems need to focus on exploiting the synergy between different modes in improving the retrieval efficiency. There is now active interest in integrating these two descriptions for building effective image retrieval systems. Ruofei *et al.* [25] proposed a probabilistic semantic model, which generates an offline image to concept word model, on which an online image-to-text and text-to-image retrieval are performed in a Bayesian framework. Wang *et al.* [22] proposed a multi model web image retrieval techniques based on multi-graph enabled active learning. Here, three graphs are constructed on images content features, textual annotation and hyper links respectively. Then a maximal margin classifier is applied for retrieval. Romberg *et al.* [11] proposed a mm-pLSA, with two separate leaf-pLSAs, and a single top level pLSA node merging the two leaf-pLSAs. Here, they apply pLSA to each mode, i.e., visual features and textual words separately, and then concatenate the derived topic vectors of each mode to learn another pLSA on top of that.

Semantic indexing schemes proposed in the literature are primarily for a single mode data [2]. There are also attempts for extending them to multimodal data [3, 11, 14]. However, all of them require complex mathematical computations involving large matrices. This makes it difficult to use it for continuously evolving data, where repeated semantic indexing (after addition of every new image) is prohibitive. In this paper, we propose a tri-partite graph based approach for multi model image retrieval for dynamically changing datasets. We represent the data as a graph, with simple procedures for insertion. Given a query image, we employ a graph partitioning scheme for separating relevant images from the irrelevant ones and thereby retrieving images from the database. The experimental results show that the data structure used is scalable, and ideally suited for incremental computation. With a computationally efficient technique, we report results on standard data set, where we show that our retrieval is as effective as that of the best reported multimodal semantic indexing schemes.

2 Direct Multimodal Semantic Indexing

In this section, we discuss multimodal semantic indexing schemes. Specifically, we describe two direct multi-modal approaches, Multi-modal Latent Semantic Indexing (*MMLSI*), and Multi-modal Probabilistic Latent Semantic Analysis (*MMpLSA*). These recent methods [3] extend the traditional semantic analysis schemes with the help of a tensorial representation. In MMLSI the data is represented by a 3-order tensor where the first dimension is text words, second is visual words and the third is the images. Three-mode analysis using Higher Order Singular Value Decomposition (HOSVD) [8] is performed on the 3-order tensor which captures the latent semantics between multiple objects like images, low-level features and surrounding text. HOSVD technique find some underlying and latent structure of images and is direct to implement. Thus it helps to find correlated dimensions within the same mode and across different modes.

Algorithm 1 Direct MultiModel Indexing

```
INPUT:
Textwords-document Matrix $N \in R^{I_1 X I_3}$.
Visualwords-document Matrix $M \in R^{I_1 X I_2}$,
Where $I_1$,$I_2$,$I_3$ are the numbers of image, visual words and text words respectively.
Procedure:
1: Construct tensor $\mathscr{A} \in R^{I_1 X I_2 X I_3}$ data.
2: Apply MMLSI or MMpLSA to reduce the dimensionality of the tensor into a semantic
   space.
   Retrieval:
3: Given a query, it is mapped to the semantic space.
4: The Euclidean distance norm D between the projected image and the query is calculated
   to get similar images.
```

In MMpLSA, an EM based algorithm is used to learn the latent concepts between images, text words and visual words. The unobservable latent variable $z \in Z = \{z_1, \ldots, z_k\}$ is associated with each occurrence of the text word $t \in T = \{t_1, \ldots, t_n\}$ and visual word $v \in V = \{v_1, \ldots, v_m\}$ in a document $d \in D = \{d_1, \ldots, d_i\}$. Here, n and m are the vocabulary sizes of text words and visual words respectively, i is the number of images in the dataset and k is the number of concepts. To simplify the model, we assume that the pairs of random variables (v_m, t_n) are conditionally independent given the respective image or document d_i. Thus the generative model is expressed in terms of the following :

$$P(d_i, t_n, v_m) = P(d_i)P(t_n|d_i)P(v_m|d_i). \tag{1}$$

The joint probabilistic model for the above generative model is given as $P(d_i, t_n, v_m) =$

$$P(d_i)\sum_k P(t_n|z_k)P(z_k|d_i)P(v_m|z_k)P(z_k|d_i) = \frac{P(d_i)^2 \sum_k P(t_n|z_k)P(v_m|z_k)P(z_k|d_i)^2}{P(z_k)} \tag{2}$$

Where, $P(t_n|z_k)$, $P(v_m|z_k)$ and $P(z_k|d_i)$ are the probability distribution of text words, visual words and images respectively over the concepts which are learned using Expectation-Maximization Algorithm (EM-Algorithm).

Though these methods are direct extensions of the traditional single mode counter parts, they provide superior results to other existing multimodal semantic indexing schemes. Comparative studies are carried out on popular databases, and results are shown in Table 1. We compare our method with single mode methods (visual-based, tag-based) first. It can be seen that tag based is superior to visual features except for the UW dataset. We also built a Multi modal document indexing and retrieval system by concatenating the vocabulary of text words and visual words into one column to form a single matrix on which LSI and pLSA are applied separately. This is similar to the methods in [15]. We also compare our multimodal pLSA with multilayer multi modal pLSA(mm-pLSA) proposed by [11]. From the comparative study on a variety of popular data sets, we find that the results obtained by direct method are superior to other methods. However, semantic indexing schemes also have certain disadvantages. The disadvantages of the existing Multimodal semantic indexing include the high computational and memory requirement. In this paper, we address this problem with the help of a tripartite graph model as we describe in the next section.

	LSI				PLSA				
	visual	tag	concat[14]	[3]	visual	tag	concat	mm-pLSA	[3]
UW	0.55	0.46	0.55	0.63	0.60	0.57	0.59	0.68	0.70
MultiLable	0.33	0.42	0.39	0.49	0.36	0.41	0.36	0.50	0.51
IAPR	0.42	0.46	0.43	0.55	0.43	0.47	0.44	0.56	0.59
Corel	0.25	0.46	0.47	0.53	0.33	0.47	0.46	0.59	0.59

Table 1: Comparison of mean average precision (mAP) for various multimodal semantic indexing schemes. Single mode ('visual' as well as 'tag') methods are compared against multimodal semantic indexing scheme(concat[13] as proposed in [14]). The tensorial methods proposed in [3] is superior to the single mode counterparts as well as other possible multimodal semantic indexing schemes.

3 Tripartite Graph Representation and Retrieval

The basic idea, here, is to encode the tensorial representation as a Tripartite graph of text words, visual words and images. An undirected tripartite graph $G = (T, V, D, E)$ has three sets of vertices where, $T = \{t_1, t_2 \ldots, t_n\}$ are text words, $V = \{v_1, v_2 \ldots, v_m\}$ are visual words and $D = \{d_1, d_2 \ldots, d_i\}$ are images with $E = \{e^{d_1}_{t_1}, .., e^{d_i}_{t_n}, e^{d_1}_{v_1}, .., e^{d_i}_{v_m}, e^{t_1}_{v_1}, .., e^{t_n}_{v_m}\}$ as set of edges. Figure 1 pictorially represent the tripartite graph model (TGM) we use. Thus this model has three sets of vertices (images, text words and visual words) and edges going from one set to other. The nodes correspond to visual words as well as text words store the inverse document frequency (IDF) corresponding to the document(image) collection. The edges from text words to images as well as those from visual words to images, encode the term frequency (TF) corresponding to the word-image pair. However, the weights of edges which relate the text words with visual words can not be directly assigned. These edges are weighed as:

$$W_{pq} = \frac{\sum_i C_{t_p,v_p}(\alpha e^{d_i}_{t_p} + (1-\alpha)e^{d_i}_{v_q})}{\sum_i \alpha e^{d_i}_{t_p} + (1-\alpha)e^{d_i}_{v_q}}$$

Where $C_{t_p,v_q} = 1$, if t_p and v_q are there in document d_i. Since the documents (images) are the entity which connects text words and visual words, summations are carried out over the images/documents. For indexing, a tripartite graph G is constructed with the nodes and edges as mentioned above. Given a collection of images and textual tags, building a TGM is possible. However, when additional images come, TGM shows its advantage in insertion. To insert an additional image, the TF and IDFs are computed with the new document. We assume the vocabularies to be static. This is computationally light. For retrieval, we partition the vertex set D of G into two vertex sets $(V1, V2)$, such that vertex set $V1$ contains documents which are relevant to the query, and $V2$ contains all other nodes. This is done as explained below.

When a query image is given, it is inserted into the TGM with the same (visual as well as text) vocabulary. Our objective now is to identify similar images to the query, which are already indexed. For computing this, we start from the query image, and traverse to the

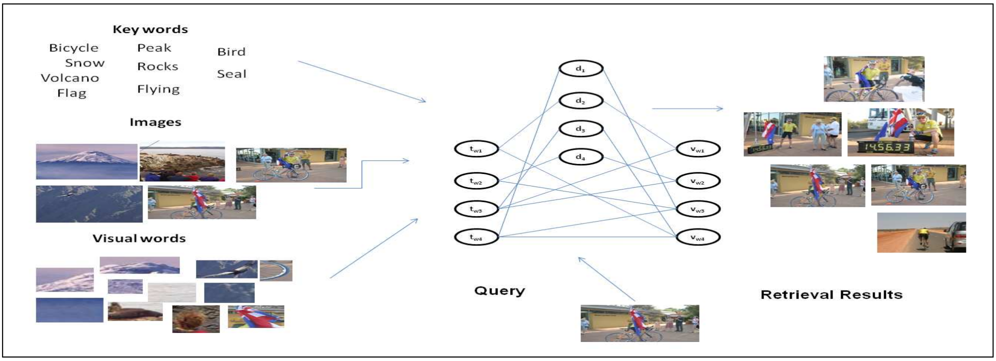

Figure 1: Tri-partite Graph Representation of dataset, t_{w_i} are text words, v_{w_i} are visual words and d_i are the images

neighboring nodes (visual as well as text words). An initial relevance score (R) which is 100 at the query node gets distributed to all the neighboring nodes, according to the TF values. Then these word-nodes, propagate this relevance score back to the connected documents. This is done based on IDF values. The relevance score is propogated between the text and visual words based on the edge weights connecting them. Thus in one iteration, the relevance score gets distributed over multiple documents. The entire process is repeated multiple times. Finally all documents, which contain at least a specific relevance score, are grouped together as a set of relevant images $V1$.

A special case of TGM is a Bipartite Graph model(BGM). An undirected bipartite graph $G = (D,W,E)$ has set of edges $E = \{e^{d_1}_{w_1}, e^{d_2}_{w_7} \ldots, e^{d_i}_{w_m}\}$ and two sets of vertices $D = \{d_1, d_2, \ldots, d_i\}$ which represents images and $W = \{w_1, w_2 \ldots, w_n\}$ which represents visual words of the images. Here the weight associated with $w_1 = IDF(w_1)$ and $e^{d_1}_{w_1} = TF(w_1, d_1)$. This can index and retrieve images with a visual bag of words model. Note that this is not a multi-modal case. We discuss about BGM, because it helps in directly comparing the utility of the proposed graph-based semantic indexing, in comparison with pLSA model popular in computer vision literature. We show experiments with both BGM and TGM. In the case, of BGM, it is quantitatively shown that the Graph-based indexing models are lighter and efficient in practice. The mAP obtained are quite comparable. An important advantages of the graph-based models is that, this does not require the prior knownledge about the number of hidden concepts as in the case of methods like pLSA.

3.1 Learning Edge Weights

Here we present a method for learning the edge weights for the Tripartite graph. In the above section the edge weight in the tripartite graph was determined by the widely used Term Frequency. Though it is simple, the quality of the similarity measure is not domain dependent and cannot be easily adjusted to better fit the final objective. Therefore we used the method proposed by [24] to learn the edge weights of the tripartite graph to improve the retrieval performance. A term-weighting learning framework is constructed using a parametric function of features for each text word and visual word, where the model paremeters are learnt from labeled data. Each document is represented with text word vector of length n, $v_t = (s_t^1, s_t^2, \ldots, s_t^n)$ and a visual word vector of length m, $v_v = (s_v^1, s_v^2, \ldots, s_v^k)$. Where s_t^i is the weight of the text word t_n which is determined by the term weighting function $f_t(t_n, d_i)$ and s_v^i is the weight of the visual word v_m which is determined by the term weighting function $f_v(v_m, d_i)$.

For every image in the training set we assign two labels. The first label between image and each visual word is denoted as $\{(y_1,(v_1,d_1)),(y_2,(v_2,d_1)),(y_3,(v_1,d_2))\ldots,(y_{mxi},(v_m,d_i))\}$. The label y_{mxi} is the visual term frequency of v_m in the image d_i. and second label between image and each text word, is denoted as $\{(h_1,(t_1,d_1)),(h_2,(t_2,d_1)),(h_2,(t_2,d_1)),\ldots,(h_{nxi},(t_n,d_i))\}$. The label h_{nxi} is the text term frequency of the t_n in the images d_i. A parametric function of features for each visual word and text word are calculated separately.

Then we use general loss functions sum-of-squares error and log loss to learn the model parameter by using L-BFGS for fast convergence and local minima as described in [24]. The final value of y_{kxm} and h_{nxm} gives the relevance between the image and the corresponding visual words and text words respectively, which can be considered as the weights of the Tripartite graph. Then we apply graph partitioning algorithm as mentioned in the above section.

3.2 Offline Indexing

Here we discuss Bipartite graph model as a special case of TGM. An offline indexing technique for BGM is presented to reduce the computational time for retrieval. In BGM, the edges are weighted with term frequencies of words in the documents and each term is also associated with an inverse document frequency value. These values determine the relevance of a word in a particular image. We use graph comparison method in [21] to obtain the similarity between images. Here, first we present some basic definitions and then explain how graph comparison method is used for computing similarity between images.

A similarity matrix **S** is computed between two graphs G_A and G_B as a limit of the normalized even iterates of $S_{p+1} = BS_pA^T + B^TS_pA$, where A and B are the adjacency matrix of G_A and G_B respectively. The entry s_{xy} in similarity matrix **S** gives the similarity score between a vertex x in G_A to a vertex y in G_B. A special case is G_A =G_B=G', where G' is a graph. The similarity matrix **S** gives similarity scores between vertices of G', which is self similarity matrix of G'. Truong *et al.* [21] shows the application of this for document retrieval. Here we demonstrate this for image retrieval. The values for the similarity matrix can be either initialized to a known prior knowledge between the vertices's of the graphs or same similarity values (for example 1). Let M be the adjacency matrix of a bipartite graph G where the vertices's have been ordered such that the first i rows are the number of images in D and last m rows are the visual words in W. The initial values of the similarity matrix is computed as follows:

$$S_0(x,y) = \frac{\sum\limits_{p=1\to i+m} M(x,p)*M(y,p)}{\sqrt{\sum\limits_{p=1\to i+m} M(x,p)*M(y,p)} * \sqrt{\sum\limits_{p=1\to i+m} M(x,p)*M(y,p)}} \quad (3)$$

The S_0 can be written as $\begin{bmatrix} S_W & 0 \\ 0 & S_D \end{bmatrix}$ where S_W is the $m \times m$ visual word similarity matrix and S_D is the $i \times i$ image similarity matrix.

$$S_{p+1} = \frac{\begin{bmatrix} L^t L S_{W_p} L^t L & 0 \\ 0 & L^t L S_{D_p} L^t L \end{bmatrix}}{\sqrt{\|L^t L S_{W_p} L^t L\|^2 + \|L^t L S_{D_p} L^t L\|^2}} \tag{4}$$

Where L is the term document matrix. Iterating the equation 4 until convergence is achieved will result in a similarity matrix S_p which gives the similarity measure between the images in the graph G.

4 Results and Discussions

4.1 BGM, PLSA and Image Retrieval

First we demonstrate the application of a simplified TGM(i.e, BGM) model for image retrieval. Bipartite Graph Model(BGM) represents vector space model as bipartite graph of documents and words. The edges of the graph are weighted with term frequencies of words in the documents and each term is also associated with an inverse document frequency value. For retrieval, a graph partitioning algorithm is applied which is a variation of [9, 16]. To study the retrieval performance of BGM and to compare it with pLSA and IpLSA [23], we use holiday dataset [7] which contains 500 image groups, each representing a different scene or object. The dataset contains 1491 images. In each group the first image is the query image and the remaining images are the relevant images pertaining to the query image. We made extensive use of local detectors like Laplacian of Gaussian(log) and the SIFT descriptors[5]. Initially all the images from the dataset were downsampled to reduce number of interest points, and afterwards feature detection and SIFT feature extraction was done. Once the features were extracted the cumulative feature space was vector quantized using K-means. With the aid of this quantization the images were converted into documents or collection of visual words. For pLSA, we first construct a term document matrix A of the order $M \times N$ where M is the vocabulary size and N is the number of documents. Here, each image is represented as a histogram of visual words. An unobservable latent topic Z_k is introduced between the documents and the words. Thus $P(w_m, d_i) = P(d_i)\sum_k P(z_k|d_i)P(w_m|z_k)$. Here we learn the unobservable probability distribution using EM algorithm. Wu *et al.* [23] proposed an Incremental pLSA wherein the probability of a latent topic given the document $P(z|d)$ and the probability of words given topic $P(w|z)$ are updated based on Generalized Expectation Maximization [12, 23] whenever a new image is added. The performance of theses methods both in terms of computation efficiency and retrieval performance are quite good. Yet they don't effectively address the issue of updating the number of global latent concepts as the

Model	mAP	time	space
Probabilistic LSA	0.602	547s	3267Mb
Incremental PLSA	0.567	56s	3356Mb
BGM	0.594	42s	57Mb

Table 2: Mean Average Precision for both BGM, pLSA and IpLSA for the holiday dataset, along with time taken to perform semantic indexing and memory space used during indexing.

database grows. For retrieval the Euclidean distance of the documents over topic probabilities was used to retrieve the top 10 similar images. Retrieval results for the both BGM and pLSA were aggregated and the evaluation code provided for the holiday dataset was used to calculate the Mean Average Precision(mAP). The results are shown in Table 2.

Now we demonstrate the performance of the matrix based offline indexing technique for BGM. The Table 3 shows the comparison of the online BGM and offline BGM as we can see there is only a neglegible difference in the performance.

Model	mAP	time	space
BGM online	0.594	42s	57Mb
BGM offline	0.57	120s	86Mb

Table 3: Mean Average Precision for both BGM online and offline for the holiday dataset, along with time taken to perform semantic indexing and memory space used during indexing.

4.2 Multimodal Retrieval

In this section, we present the experimental results for the proposed TGM and compare with the other Multimodal retrieval systems. We used four datasets for the evaluation of the methods proposed. *University of Washington(UW) Dataset*: This dataset is used in [10] and consists of 1109 images with a ground truth of manually annotated key words. For evaluation the retrieved image is considered relevent if it belongs to the same class as the query image. *Multi-label Image Dataset*: This dataset is used in [18] and consists of 139 urban scene images and four overlapping labels: *Buildings*, $Flora$, *People* and *Sky*. For visual evaluation we manually created a ground truth data for 50 images. *IAPR TC12 Dataset*: This data set consists of 20,000 images of natural scenes. Here the images are accompanied with description in several languages and typically used for cross-language retrieval[6], we have concentrated on English captions and extracted keywords using natural language processing techniques. The vocabulary size is 291 and 17,825 images were used for training, and 1,980 for testing. *NUS-WIDE*[4]: It consist of 269,648 images and the associated tags from Flickr, with a total number of 5,018 unique tags. Initially all the images from the datasets were down sampled to reduce number of interest points, after which feature detection and SIFT feature extraction [5] is applied. Now the features are vector quantized using k-means. For our experiments we created a visual vocabulary size of 500 for all the datasets, except for IAPR for which the vocabulary size is 1000.

Table 4: Comparing TGM with Multi Modal LSI and Multi Modal pLSA for different the datasets

	MMLSI	MMpLSA	mm-pLSA	TGM-TFIDF	TGM-learning
UW	0.63	0.70	0.68	0.64	0.67
MultiLable	0.49	0.51	0.50	0.49	0.50
IAPR	0.55	0.59	0.56	0.56	0.59
NUS-WIDE	0.33	0.39	0.37	0.35	0.38

We implemented Multi modal LSI(MMLSI) and Multi Modal pLSA(MMpLSA) as explained in section 2. The latent concept k is set to the value specified in the dataset. We also implemented a multi-layer multi modal pLSA as explained in [11]. An improvement in performance is expected over naive merging of dictionaries, as the effect of difference in distribution patterns of each mode is normalized in this method. However, it has an intrinsic problem of having to merge dictionaries of the different modes. This method does not place importance to interactions between the different modes. We argue that such interactions have the ability to find useful information in the dataset.

A TGM with edge weights as TF and as weighted-learning is constructed for all the datasets as explained in sections 3 and 3.1 respectively. Table 4 shows the comparison of these methods in mean Average Precision(mAP) values. For all our experiments the number of concepts is determined by the concepts present in the respective databases that are known. The mAP results show that performance of TGM is comparable to other methods. The performance of TGM with weighted-learning is slightly better that that with the TF. The advantage of TGM is noticeable when new images are added to database. TGM takes only few milliseconds for semantic indexing whereas for variants of pLSA the entire semantic indexing needs to be done again, incurring high time and memory costs.

Figure 2 shows that multi mode TGM performs well compared to single mode TGM. This is manly because graph partition ranks images based on both visual words and text words. Also, consideration of both visual words and text words eliminates the irrelevant images from appearing in the results.

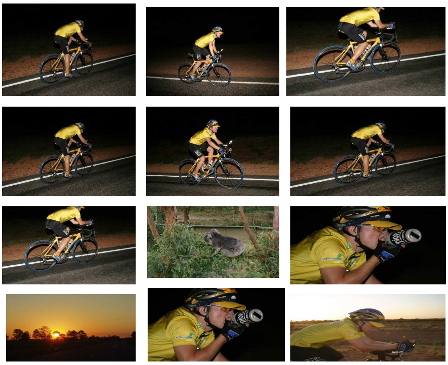

Figure 2: The first image is the query, the rest of the images in the first column are the visual results, the images in the second column were obtained when text query “Cyclist in Australia”was given. Last column comprises of multimodal results of TGM-learning.

5 Conclusion

In this paper, a tripartite graph based multi modal semantic indexing applicable to image retrieval for dynamically changing or evolving datasets is proposed. We also propose a graph partitioning algorithm for retrieving semantically relevant images from the database. We show that the proposed algorithm is comparable with other multi model methods. Our experimental results show that the data structure used is scalable, computationally light and less resource intensive.

References

[1] www.flickr.com.

[2] A. Bosch, A. Zisserman, and X. Munoz. Scene classification via pLSA. In *ECCV*, 2006.

[3] Pulla Chandrika and C. V. Jawahar. Multi modal semantic indexing for image retrieval. In *CIVR*, 2010.

[4] Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yan-Tao.Zheng. Nus-wide: A real-world web image database from national university of singapore. In *CIVR*, 2009.

[5] D.Lowe. Distinctive image feature scale-invariant keypoints. In *IJCV*, 2004.

[6] M. Grubinger. Analysis and evaluation of visual information systems performance. In *PhD thesis, Victoria University Melbourne, Australia*, 2007.

[7] Herve Jegou, Matthijs Douze, and Cordelia Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *ECCV*, 2008.

[8] J. Vandewalle. L. De Lathauwer, B. DeMoor. A multilinear singular value decomposition. In *SIAM J. Matrix Anal. Appl*, 2000.

[9] Amy N. Langville and Carl D. Meyer. *Google's PageRank and Beyond: The Science of Search Engine Rankings*. 2006.

[10] Changhu Wang. Lei Zhang, Hong-Jiang Zhang. Scalable markov model-based image annotation. In *IJCV*, 2004.

[11] Rainer Lienhart, Stefan Romberg, and Eva Hörster. Multilayer plsa for multimodal image retrieval. In *CIVR*, 2009.

[12] Radford M. Neal and Geoffrey E. Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. 1999.

[13] David Nister and Henrik Stewenius. Scalable recognition with a vocabulary tree. In *CVPR*, 2006.

[14] Trong-Ton Pham, Nicolas Eric Maillot, Joo-Hwee Lim, and Jean-Pierre Chevallet. Latent semantic fusion model for image retrieval and annotation. In *CIKM*, 2007.

[15] Trong-Ton Pham, Nicolas Eric Maillot, Joo-Hwee Lim, and Jean-Pierre Chevallet. Latent semantic fusion model for image retrieval and annotation. In *CIKM*, 2007.

[16] U.N. Raghavan, R. Albert, and S. Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, 2007.

[17] Yong Rui, Thomas S. Huang, and Shih fu Chang. Image retrieval: Past, present, and future. In *Journal of Visual Communication and Image Representation*, 1997.

[18] Singh, P. M., Cunningham, and E. Curran. Active learning for multi-label image annotation. In *AICS*, 2008.

[19] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, 2003.

[20] Arnold W. M. Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. Content-based image retrieval at the end of the early years. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2000.

[21] Quoc-Dinh Truong, Taoufiq Dkaki, josiane Mothe, and Pierre-Jean Charrel. Information retrieval model based on graph comparison. In *JADT*, 2008.

[22] Xin-Jing Wang, Wei-Ying Ma, Lei Zhang, and Xing Li. Multi-graph enabled active learning for multimodal web image retrieval. In *MIR*, 2005.

[23] Hu Wu, Yongji Wang, and Xiang Cheng. Incremental probabilistic latent semantic analysis for automatic question recommendation. In *RecSys*, 2008.

[24] Wen-tau Yih. Learning term-weighting functions for similarity measures. In *EMNLP*, 2009.

[25] Ruofei Zhang, Zhongfei (Mark) Zhang, Mingjing Li, Wei-Ying Ma, and Hong-Jiang Zhang. A probabilistic semantic model for image annotation and multi-modal image retrieva. In *ICCV*, 2005.