Graph mining-based Image Indexing

Gábor Iváncsy, Renáta Iváncsy and István Vajk

Department of Automation and Applied Informatics,
Budapest University of Technology and Economics,
1111, Goldmann Gy. ter 3.
Budapest, Hungary
e-mail: {gabor.ivancsy, renata.ivancsy, vajk}@aut.bme.hu

Abstract: With the increasing importance on multimedia applications, the production of multimedia information resulted in a large amount of image and video data which are stored in multimedia databases. Thus image indexing has become important since ever huger databases exist to store this kind of data. The effectiveness of the image retrieval can be enhanced by efficiently indexing the images. Several techniques have been developed to query image databases by their image content which is based on different features of the image, such as color-histogram, spatial similarity, signature etc. The usage of these features of the image for indexing is limited, thus a new approach is needed to efficiently handle the large amount of image data. In this paper a novel approach is suggested in order to efficiently indexing the images. The main contribution of the paper is to represent the images as graphs, and indexing them using graph mining technique. In this case the indexing is based on the frequent substructures of the images which are discovered using an efficient graph mining method. The index structure for an image database consists of frequent substructures of the images.

Keywords: image retrieval, image indexing, graph mining, graph indexing

1. Introduction

The problem of image indexing is a heavily researched area in the field of image retrieval. Several applications exist which serves a lot of multimedia data such as video streams and digital images. For example most of the institutes (banks, supermarkets, universities etc.) have security cameras to supervise the area of the building. These systems produce a lot of image information which have to be stored in databases for future use. On another hand, since the price of digital cameras is sinking, the usage of these instruments is even more popular. This results in a lot of images, which are stored on the file system of the user's computer, or in some cases in huge digital multimedia libraries. Other application

areas are for example medical imaging, remote sensing, cartographic systems, robotics, CAD and CAM systems, document image processing and so on.

The approach of image indexing aims to retrieve images efficiently from the image database. The main task of image retrieval is to discover a set of images from the image database so that the user can easily find the picture searched for. In case of content-based image retrieval the query is an image, and the system is discovering similarly pictures in the database. Usually the similarity measure is predefined and it is dependent on the particular choice of the features used to represent the image.

The image retrieval systems can be classified into three main classes [1]. The most basic systems search for images by using primitive features such as color, shape and texture. Other systems discover the pictures by derived attributes involving logical inference about the objects depicted in the image. The most advanced image retrieval systems use abstract attributes and logical reasoning to determine what the query picture shows and produces pictures of similar objects. Thus far such systems seem to be impossible to develop.

A novel research trend is to use graph representation when querying the image database. For example when using Region Adjacency Graph [2] the model contains both structural and perceptual features of the images. Using graphs the indexing of the data can be solved by discovering the frequent substructures of the graphs. This produces an index structure for the querying process. When querying the database by a certain image, the input image is converted into a graph, and it is matched to the indexed graphs and subgraphs in the database. If there exists a mapping between the input graph and one of the indexed graph the image is provided as the result. In other cases subgraphs of the input graph is discovered and mapped using the index structure. In this way the most similar images to the input picture can be found.

The organization of the paper is as following. In Section 2 the main aspects of image indexing is described. Because the suggested method uses a graph mining technique, the concepts of graph mining is outlined in Section 3. Section 4 describes how an image can be represented using a graph based on Region Adjacency Graphs. Section 5 describes the image indexing method. Conclusion and future work can be found in Section 6.

2. Problem Statement and Related Work

When querying a large database, indexing is one of the techniques with which the process of obtaining the result data can be enhanced significantly. This stands for image databases as well. However, while indexing a table in a relational database

based on the primary keys of the table is a well known method; indexing a multimedia database based on its content is so far a problematic task.

The requirements to an image index structure are the following. The index structure should support both general and specific queries with some constraints. In a general query the input is a set of objects, and the expected answer is a set of images containing the given object. When a specific query is defined some constraints are added to a general query. An index structure should be dynamic so that new images can be indexed, and the index for the deleted images is deleted as well. One of the most important features of index structures is its efficiency. Using the index structure the searching mechanism should filter out the non relevant images while it should not discard any relevant ones. Another important point is the storage requirement of the index structure.

Image indexing is an extensively studied area, thus several image indexing methods were proposed so far. In [3, 4] the content-based image retrieval system uses spatial color histograms to obtain the required information. In this case the color histograms are extended so that they contain information also about the spatial features of the image. In [5] the indexing problem is formulated as a multi-dimensional nearest neighbor search problem. The system uses an optimistic vantage-point tree algorithm. [6] suggests a Gabor-filter-based feature extraction for medical image indexing and classification.

3. Graph mining

Graph is an appropriate tool for modeling several real-world structures, like Web links, chemical compounds, XML documents, academically citations, images and so on. Graph mining is a method to discover frequent substructures in large graph databases. First of all the introduction of some definition related to graphs is needed in order to explain the main steps of graph data mining.

A **graph** G = (V,E) is a collection of vertices V and edges E. Each edge is a pair of vertices, formally $E \subseteq V \times V$. A **labeled graph** G is a five element tuple G = (V, E, L_V, L_E, l) where V is the set of vertices, E is the set of edges; L_V and L_E are the vertex labels and the edge labels, respectively. The function l defines an edge label and a vertex label to each edge and to each vertex. Each vertex and edge of the graph are not required to have a unique label and the same label can be assigned to many vertices or edges in the graph i.e. l is not a bijective function.

Given two labeled graphs G = G = (V, E, L_V, L_E, l) and G' = (V',E',L'_V, L'_E, l'), G' is a **subgraph** of G iff $V' \subseteq V$ and $E' \subseteq E$ and $\forall$ v' $\in$ V', (l(v' = l(v)) and $\forall$ (v'_i ,v'_j) $\in$ E', (l'(v'_i , v'_j) = l(v'_i , v'_j)). G' is an **induced subgraph** of G iff $V' \subset V$, $E' \subset E$ and $\forall$ v_i, $v_j \in V'$, $e_{ij} = (v_i, v_j) \in E'$ if and only if $e_{ij} = (v_i, v_j) \in E$. A

graph is a **connected graph** if all its vertices are mutually reachable through some edges of it.

The problem of frequent pattern discovery in graph databases can be defined in two ways, namely, as graph-transaction setting and as single-graph setting. In single-graph setting the input of the mining system is one single graph with large number of nodes, and the task is to find frequent recurring subgraphs of the single input graph. In graph-transaction setting the database to be mined is a set of graphs which are relatively small, and the task of the mining process is to find frequently recurring graphs in this graph dataset. Because the problem of image indexing fits to the concept of graph-transaction setting henceforth only the concepts of graph-transaction setting is described.

The method of pattern mining in graph data means that the number of occurrences of a certain substructure is to be counted. For this reason one should determine to how many graphs in the dataset is the candidate graph subgraph isomorphic. A labeled graph $G = (V, E, L_V , L_E, l)$ is **isomorphic** to another graph $G' = (V' ,E';L'_V ;L'_E; l')$ if they are topologically identical to each other, that is, there is a mapping from V to V' such that each edge in E is mapped to a single edge in E' and vice versa, so that the mapping preserves the labels. A labeled graph G' is **subgraph isomorphic** to a labeled graph G, denoted by $G' \subseteq G$ iff there exists a subgraph G'' of G such that G' is isomorphic to G''. The support of a graph G in the database D is the number of graphs G', so that $G \subseteq G'$. In most cases the support is defined as the fraction of such graphs in the database which has G as a subgraph (in this case the number of the found graphs should be divided with the number of the graphs in the database).

There are several algorithms which deal with the problem of discovering frequent graphs in a graph database. All of them use the a-priori hypothesis, namely, a graph can be only frequent if all its subgraphs are frequent as well. In other words, if a graph is not frequent no superset of it will be frequent. Using this knowledge the search space of the mining process can be reduced significantly. If a graph is proven not to be frequent, this graphs need not to be considered as a basis of further supergraphs.

The different algorithms use different approaches to efficiently discover the frequent subgraphs. Several of them use a level-wise “candidate generate and test” approach which was first introduced by Agrawal et al. [7] as a solution to the frequent itemset mining problem. The basic idea of these algorithms is to process the database level-wise. On each level candidates are generated from the frequent graphs discovered before, which are one vertex (or one edge) greater than the graphs discovered in the previous level. The AGM [8], AcGM [9], and FSG[10] uses a breadth-first search approach when generating the candidates, while the FFSM [11] and gSpan [12] uses depth-first search traversal of the search space.

One of the basic algorithms is the AGM algorithm. Its basic principle is similar to that of the itemset mining Apriori algorithm. It begins the process by discovering

the frequent subgraphs having only a single vertex, and the larger subgraphs are found by generating candidates by adding an extra vertex to the graph discovered so far. Each graph is represented with its adjacency matrix. The cell belonging to the i^{th} row and j^{th} column contains the label of the edge between the vertex i and j if it exists. If there is no edge between the two vertices the value of the cell is set to zero. Because there exist $n!$ adjacency matrices for the same graph having n vertices an ordering of the vertices is needed to decrease the number of the vertex combination. The AGM algorithm orders the vertices regarding the vertices labels and creates this way the vertex-sorted adjacency matrix. In itemset mining the itemsets have a lexicographic order, thus the candidate generation step can be achieved without generating any redundant candidates. However the vertex-sorted adjacency matrices do not have such ordering thus a coding method need to be introduced. Given a vertex-sorted adjacency matrix:

$$X_k = \begin{pmatrix} x_{1,1} & x_{1,2} & x_{1,3} & \dots & x_{1,k} \\ x_{2,1} & x_{2,2} & x_{2,3} & \dots & x_{2,k} \\ x_{3,1} & x_{3,2} & x_{3,3} & \dots & x_{3,k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{k,1} & x_{k,2} & x_{k,3} & \dots & x_{k,k} \end{pmatrix}, \quad (1)$$

the code of the matrix X_k is defined as follows:

$$code(X_k) = x_{1,1}x_{1,2}\, x_{2,2}x_{1,3}x_{2,3}x_{3,3}x_{1,4}\, \dots x_{k-1,k}x_{k,k}, \quad (2)$$

where only the upper triangular matrix is used because of the diagonal symmetry of the adjacency matrix for an undirected graph. The introduction of this code significantly reduces the graph representation and the search space. The $(k + 1)$-sized candidate generation of the subgraphs is done by joining two k-sized graphs. This is done only when the following conditions are satisfied. Given the adjacency matrices of two k-sized graphs X_k and Y_k. If both X_k and Y_k have equal elements except for the elements in the k^{th} row and the k^{th} column and $code(X_k) \leq code(Y_k)$, then they are joined to generate Z_{k+1}.

$$X_k = \begin{pmatrix} X_{k-1} & x_1 \\ x_1^T & x_{k,k} \end{pmatrix}, Y_k = \begin{pmatrix} X_{k-1} & y_1 \\ y_1^T & y_{k,k} \end{pmatrix}, Z_{k+1} = \begin{pmatrix} X_{k-1} & x_1 & y_1 \\ x_1^T & x_{k,k} & z_{k,k+1} \\ y_1^T & z_{k+1,k} & y_{k,k} \end{pmatrix}, \quad (3)$$

where X_{k-1} is the adjacency matrix of the $(k-1)$-sized graph, x_i and y_i are column vectors. The elements $z_{k,k+1}$ and $z_{k+1,k}$ represent an edge label between the k^{th} vertices of X_k and Y_k. The adjacency matrix generated in this way is called a normal form. In the standard itemset analysis a $(k + 1)$-itemset becomes a candidate only when all its k-subsets are frequent. This is similarly in the graph mining. A graph G is only a candidate when all adjacency matrices generated by

removing the i^{th} vertex and all its links are to be proved frequent. This can be done without reading the dataset. After the candidates are generated they support is counted during a database scan. In order to enhance the computation of counting the graphs are stored with their normal forms, and the subgraph matching is made between the normal forms.

4. Representing Images as Graphs

The representation of an image is a structure, which contains all relevant information, such as the structure and color, while discarding unused information. The extraction of the features allows for compact representations and fast processing. The main idea of the graph-based representation is, that regions of the image, witch possess similar characteristics can be interpreted as a node of a graph. The nodes attributes contain the information to this similarity as well as information about the region itself, such as area or shape. To receive the aforementioned regions, the image has to be segmented. This segmentation is achieved by a region growing algorithm (RGA). The algorithm is as following:

1. Each pixel in the image is assigned to a region, thus, each pixel represents a region by itself.
2. At each step adjacent regions are merged, if the regions features satisfy a predetermined criterion. The criterion is a comparison is a homogeneity condition.
3. The process is stopped, when no adjacent regions can satisfy the criterion.

As one can easily see, the results depend heavily on the criterion. For further processing, the criterion is the following [13]:

$$min(A(R1),A(R2))^{\alpha} * CD\ (R1,R2) < \Theta \quad (4)$$

where *R1* and *R2* are the regions to be merged, *A(R)* is the normalized area of region *R* , *CD* is the average color distance of the regions, and α and Θ are predetermined parameters. It seems obvious, that the resulting regions are connected.

The connections of the graph nodes correspond to the structural information of the image. Two regions are adjacent, if they have adjacent pixels. For each adjacent region the respective graph nodes are connected. The resulting graph is the Region Adjacency Graph (RAG). In Figure 1 the graph representation of a sample image is shown. Figure 1.a) depicts the original image, in Figure 1.b) the regions and the nodes can be seen and Figure 1.c) shows the resulting RAG. If information about the relationship of the regions is to be saved into the graph, these should appear as

edge labels. Such information can be distance of average midpoints of the region, angle of the edges, etc. Scale invariance can be preserved by normalizing these values according to the area of the regions.

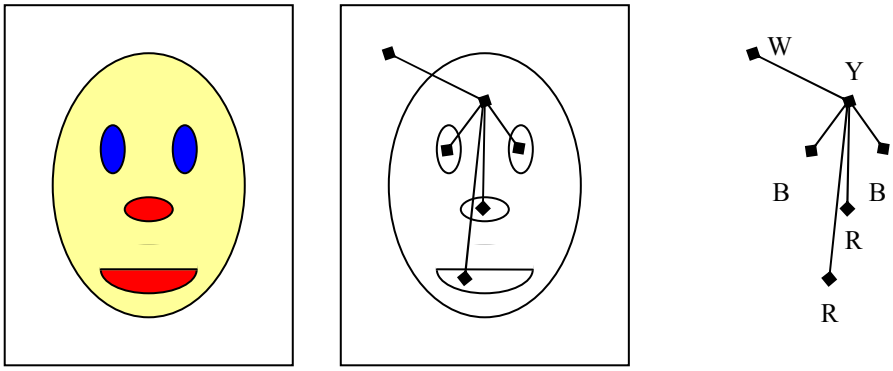

Figure 1. Region Adjacency Graph of an image, a) original image, b) regions and graph nodes c) labeled graph

5. Indexing the Image Database

As described in the previous section, the images are represented as graphs. The approach proposed in this paper is based on this concept and uses a graph mining algorithm to receive the resulting images. One of the input parameters of the mining algorithm is the minimum support threshold. If a support of a subgraph exceeds the user given minimum support threshold, it is denoted as frequent subgraph. Frequent subgraphs expose the intrinsic characteristic of a graph database.

The indexing mechanism is similarly to the approach described in [14] and works as follows. After creating the graph representation of each image the frequent subgraphs are discovered. The key issue is the appropriate choice of the minimum support threshold because it determined which subgraphs are frequent. If the threshold is set to too large, only a few images will exceed the threshold, thus they will not represent the database correctly. Setting the minimum support threshold to too low results in too much graphs, which can be handled very difficulty. The graph mining can be done using any of the known graph mining algorithms. After discovering the frequent subgraphs in the database the images are indexed using the frequent subgraphs as the indexing key.

Given a query graph G_q, if G_q is a frequent subgraph, then it is indexed and the images containing G_q can be retrieved quickly. If G_q is not a frequent graph, it has probably a subgraph, which is frequent. In this case those images constitute the result image set which contain the frequent subgraph of G_q. This is a candidate set which is to be processed to discover whether they contain the graph G_q.

The subgraph isomorphism problem is solved like in the graph mining algorithm used for creating the index structure, namely, using canonical labels. However an important question is how to define the isomorphism between the graphs. In real world images the regions of an image depicting the same object could differ much depending on the objects position, light conditions, occlusion, etc. This results in graphs that are similar but have somewhat different labels. Nevertheless, they are generated from images of the same object. Thus a distance function of the features needs to be created witch can create a score for the retrieved image. This distance function can be used later to determine the score of the retrieved images. The same distance function should be used when discovering the frequent subgraphs in the data mining process, to avoid similar subgraphs being reported only once.

This approach enables the system to search for objects without actually defining objects, since it is likely that object describing graph segments are frequent fragments in the database. By querying for an image, the most frequent features get chosen automatically by the mining algorithm.

6. Conclusion and Future Work

Image indexing plays an important role by efficient querying large image databases. In this paper a novel approach to the problem of image indexing is discussed. The main contribution of the paper is to represent the images as graphs and to use graph mining technique to efficient indexing the image database. The benefit of using graphs instead of using perceprional features is that by using graphs the spatial behaviour of the image can be modeled as well.

The given solution works well for images that consist of well defined regions, or depict objects that are such, but the region adjacency graph method fails for most real life images. In the future, a better describing graph representation will be seeked, that handles inconsistencies in real life pictures better. Occlusions can disturb the structure of the region adjacency graph. A promising direction to solve this problem is to connect not only adjacent regions, but also regions a bit farther, with edge labels that mirror the adjacency of the nodes. An other idea is to be able to compare multiple nodes to a single node.

References

[1] E. A. El-Kwae and M. R. Kabuka, Efficient Content-Based Indexing of Large Image Databases, ACM Transactions on Information Systems, Volume 18, Issue 2, April 2000, pp. 171-210.

[2] K. Saarinen, Color image segmentation by a watershed algorithm region adjacency graph processing, In Proc. Of ICIP-94, Austin, Texas, USA, 1994, pp. 1021-1025.

[3] A. Rao, R. K. Srihari and Z. Zhang, Spatial Color Histograms for Content-Based Image Retrieval, In. Proc. Of 11th IEEE International Conference on Tools with Artificial Intelligence, Chicago, Illinois, Nov. 8-10, 1999. pp. 183-187.

[4] Wynne Hsu, T. S. Chua, and H. K. Pung. An Integrated Color-Spatial Approach to Content-based Image Retrieval, In Proc. Of ACM Multimedia Conference, San Francisco, CA, November, 1995, pp. 305-313.

[5] Content-based Image Indexing, In. Proc. of the 20th International Conference on Very Large Data Bases, Santiago de Chile, Chile, September 12-15, 1994, pp.582-593,

[6] T. Glatard, J. Montagnat and I. E. Magnin, Texture Based Medical Image Indexing and Retrieval: Application to Cardiac Imaging, ACM SIGMM international workshop on Multimedia Information Retrieval (MIR'04), Proceedings of ACM Multimedia 2004, New-York, USA, October 15-16, 2004.

[7] Agrawal R, Imielinski T and Swami A, Mining associations between sets of items in large databases, In Proc. of the 1993 ACM SIGMOD International Conference on Management of Data, Washington DC, 1993, pp. 207-216.

[8] A. Inokuchi, T. Washio and H. Motoda, An aprioribased algorithm for mining frequent substructures from graph data, In Proc. of the 4th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'00), Lyon, France, 2003, pp. 13-23.

[9] A. Inokuchi, T. Washio, K. Nishimura and H. Motoda, A Fast Algorithm for Mining Frequent Connected Subgraphs, Technical Report, RT0448 in 2002

[10] M. Kuramochi and G. Karypis, An efficient algorithm for discovering frequent subgraph, IEEE Transactions on Knowledge and Data Engineering, in press

[11] J. Huan, W. Wang and J. Prins, Efficient mining of frequent subgraph in the presence of isomorphism, In Proc. of 2003 IEEE International Conference on Data Mining (ICDM'03), Melbourne, Florida, USA, 2003.

[12] X. Yan and J. Han, gSpan: graph-based substructure pattern mining, In Proc. of 2002 IEEE International Conference on Data Mining (ICDM'02), Maebashi City, Japan, 2002. pp. 721-724.

[13] C. Mauro, M. Diligenti, M. Gori and M. Maggini, Similarity Learning for Graph-based Image Representations, Pattern Recognition Letters 24, 2003, pp. 1115-1122.

[14] X. Yan, P. S. Yu and J. Han, Graph Indexing: A frequent structure-based approach, In. Proc. Of ACM SIGMOD/PODS 2004 Conference, Paris, France, June 13-18, 2004.