INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH & DEVELOPMENT

ISSN: 2278 – 0211 (Online)

Pattern Recognition Using Graph Theory

Aditya Doshi
Department Of Computer Science And Engineering, Vellore Institute Of Technology
Vellore, India
Manmohan Jangid
Department Of Computer Science And Engineering, Vellore Institute Of Technology
Vellore, India

Abstract:
Graphs and graph matching algorithms serve as a powerful tool in the process of search and comparison due to their efficiency and easy utility in form of representation. The search carried out in devices using the conventional textual form of search now seems tedious, considering the advances made in human software interaction since the emergence and development in touch-screen interfaces. In the past decade finger-touch or multi-touch interfaces have completely altered the way we interact with the devices. The input method implemented in search mechanisms, use the textual form of string input method that can be extended to graphical search which would enhance the interaction of human to software and also is efficient. A number of graph comparison approaches are required to answer whether a known symbol appears in a document and under which degree of confidence. This paper explores the idea of proposing efficient search algorithm for pattern matching that would take input in form of graphical drawings. Here we propose two strategies to recognize symbols depending on the type of their substructures. For those symbols that can be defined by a prototype pattern, we propose a graph isomorphism approach. On the other hand, for those structures consisting of repetitive patterns, we propose a syntactic approach based on graph grammar.

1.Introduction

Graphs are a general and powerful data structure for the representation of objects and concepts. Due to invariance property of graphs, that is a graph drawn on paper, if is rotated, translated or transformed into its mirror image, it still remains the same from point of view if mathematics, which makes graphs best suited for applications such as pattern recognition and computer vision where graph matching is an vital aspect. Pattern recognition field may have its input entities in various forms textual, speech, graphical. Of which this paper emphasises on pattern searching in graphical recognition field which finds its applications in various electronic devices in various fields of engineering, electronics etc. ,where documents containing graphical entities are to be recognized or entities that could be searched for using a graphical tool to sketch drawings or patterns. These systems use domain-dependent graphic notations, which have alphabets, assigned to notations that later serve as basis. Also the graph theory and isomorphism techniques serve as base to search or compare template pattern to target pattern assisted by knowledge based systems. Although no efficient algorithm is known yet that can determine isomorphism between two finite graphs, one which returns the solution in time T, where T would be proportional to order n of graph. Applications like hand-drawn based user interfaces for design systems, mainly concerning user cantered design and usability engineering focusing on interface design and usability to user, retrieving by content in graphical document databases are some applications in the field of graphics recognition that involve symbol recognition processes. Given a database and a query we may find nearest search results using basic graph concepts of graph isomorphism, maximum subgraph, and subgraph isomorphism. Since we are dealing with finite graphs, and real time systems, the deterministic isomorphism algorithm can only provide nearest solutions and not the exact ones, by reordering the template graph, which results into all possible graphs and then apply search algorithms to it.
This paper explores an efficient technique to recognize icons or patterns using graph algorithms. Pattern recognition concerns two main issues: first, to have definition of pattern to input in any learning network and second recognition approach based on the template pattern.
Here we propose a graph model to represent line drawing images and recognize constituent symbols. Our graph model is an attributed graph whose core representational unit is formulated in terms of the regions of the input image. Thus, default patterns stored in database and the template documents are represented in terms of Region Adjacency Graphs (RAG). Using RAG is advantageous due to: first, it can tolerate if there is an error in model for recognition in noisy or distorted template pattern, as we deal with real time scenarios. Further it helps to generate formula for symbol recognition for the technique implemented to search for target pattern.

2.Theoretical And Real Time Grounds

As mentioned in introduction, graphs are powerful data structures for representation of complex entities. Graph representation nodes describe objects while edges represent interrelationships between objects. If graphs possess labels and are directed then are known as relational graphs or relational structures. Isomorphism concepts may then be applied on say template graph g, to

compare it with target graphs. If we delete some of the nodes from the graph, along with its incident edges, we result in subgraph g'. A graph isomorphism is a bijective mapping between g and g' if the structure of edges and labels in common to g' are similar. Another important concept useful in this paper is maximum common subgraph. A maximum common subgraph refers to graph g'' that is a subgraph of g and g' and has its all possible subgraphs the maximum number if nodes. Hence graph isomorphism is a useful concept to find out if two objects and their underlying invariance properties are similar in their representation. Also it can be used to analyse whether or not a subgraph object is a part of another, or a group of objects which is primary concern in design and knowledge based systems that support decision making and unsupervised learning. Whereas maximum common subgraph may be utilised in measuring the similarity between objects that may not even have subgraph isomorphism among them, which is helpful in retrieving results that are nearest to search query which is the primary concern for this paper.

Another important issue of concern is, real world objects are usually affected by noise such that the graph representation of identical objects may not exactly match. Therefore to avoid such errors, it is necessary to integrate some degree of error tolerance into the graph matching process. A powerful alternative to maximum common subgraph computation is error-tolerant graph matching using graph edit distance. By the comparative study of error tolerant graph matching using graph edit distance and the well-known technique of maximum common subgraph, it is known that they are equivalent to each other for a particular class of cost functions. In particular the maximum common subgraph g'' of 2 graphs, g and g' and their edit distance d(g, g') are related with each other using following equation. So any algorithm for maximum common subgraph can be used to compute graph edit distance and vice versa until cost function condition satisfies.

3.Graph Matching Algorithms

A wide spectrum of graph matching algorithms with different characteristics is available. The standard algorithm for graph and subgraph isomorphism detection is the one given by Ullman. Maximum common subgraph detection, Classical methods for error-tolerant graph matching particular versions of the A* search procedure, i.e., they rely on some kind of tree search incorporating various heuristic look ahead-techniques in order to prune the search space.

These methods are guaranteed to find the optimal solution but require exponential time and space due to the NP completeness of the problem. Suboptimal, or approximated methods, on the other hand, are polynomial bounded in the number of computation steps but may fail to find the optimal solution. Other approaches based on neural networks such as the Hopfield network or the kohenon map. Also some genetic algorithms have been proposed recently. However, all of these approximate methods may get tracked in local minima and miss the optimal solution.

4.Graph Based Matching Technique

The initial step combines attributed graph and graph grammar. A graph represented in the form G= (V, E, LV, LE) is known as attributed graph, where V, E, LV, LE are respectively set of vertices, set of edges, label of vertices and label of edges. We use a hierarchical structure to for the attributed graph to categorize it into levels consisting of three abstraction levels. The lower level contains vectorial information of line drawing image, which builds an attributed graph after vectorization. The second level generates a Regional Adjacency Graph(RAG) with respect to the attributed graph generated in first level. A parallel structure to that of RAG is constructed by the help of string edit distance algorithm which consists of an index table of RAG nodes. These levels are utilized in computing the template documents that need to be recognized. The third level attributed graph defines model symbols. Then in second stage of the technique, Symbols are recognized in documents by using graph matching and the results returned by the parser of all the target graphs or similar graphs extracted from database. In addition to cope with symbols that combine both substructures, two important properties of our recognition method are: first, symbols to recognize are not previously segmented, i.e. the recognition step operates directly on the graph representing the template pattern, and second; the proposed methods allow recognition under distortion. Later prototype patterns are recognized by an error-tolerant graph matching approach. Given a model RAG GM an input RAG GT, the algorithm finds an error-tolerant subgraph isomorphism from GM, to GT. Most of the results produce a distortion and to avoid such, a well-known approach is adapted that assigns cost to each individual edit operation. Thus, the algorithm operates in order to find the sequence of edit operations of minimum cost that transforms the model graph in a subgraph of the input graph. Hence the algorithm: uses the model vertices of GM as indices in the index table T, of the input graph GI. Each model vertex is assigned to the most similar element in the index table. Let TG[i] be the most similar element to a model vertex vj . GM. Then, the set of input vertices represented by Tc, [i] are considered the set of compatible vertices with uj. The matching process is based on a state space search using a branch and bound strategy. Each state in the search tree represents a partial matching from a subset of GM vertices to a subset of GT vertices. The generation of successor states is guided by the cost of edit operations, i.e. the state with the minimum cost is expanded at each step by mapping a new model vertex to every input vertex not yet used in this partial matching and that belongs to the set of its compatible vertices. A parser is implemented for those components described by a graph grammar. Given a model grammar Hh and an input RAG GT, the parser algorithm finds if a subgraph of GI is accepted by HM. The first step is similar as in the graph matching method, an intensive pre-processing step which converts database of model graphs into a decision tree. Now at run time, the template graph is classified by the decision tree and all model graphs for which there exists a subgraph isomorphism from the template are detected. If we neglect the time needed for pre-processing, the computational complexity of the new subgraph isomorphism, this algorithm is only quadratic in the number of template graph vertices. In particular, it is independent of the number of model graphs and the number of edges in any of the graphs. However, the decision tree constructed in the pre-processing step is of exponential size in terms of the number of vertices of the model graphs. Now as far as grammar is concerned it describes regular repetitions of tokens, the class of compatible vertices with are candidates to belong to a texture pattern whenever the placement tuples described by the grammar productions are satisfied. The process iteratively continues starting, at each iteration. The presence of inserted vertices isomorphism when a production is applied

can be seen as an edit operation, i.e. a way to modulate errors and distortion in the grammar. Finally the query results are displayed along with all the possible other results found similar to query.

5.Conclusion

In this paper we have reviewed various utilities of graph and graph matching algorithms, as to how they are applied in real time framework along with various techniques applied in field of graph matching. It can be concluded that graphs are versatile and flexible representation formalism suitable for a wide range of problems in intelligent information processing, including the areas of pattern recognition and computer vision. A wide spectrum of graph matching algorithms has become available meanwhile. They range from deterministic approaches, suitable for finding optimal solutions to problems involving graphs with a limited number of nodes and edges, to approximate methods that are applicable to large-scale problems. The graph matching algorithms reviewed in this paper may also serve as a viable approach to icon/pattern recognition as the algorithm proposed uses dynamic knowledge based systems as base databases through which parser can search and compare efficiently in less time period. Although the pattern recognition problem falls under NP complete problems category an deterministic algorithm resulting in finite time completion and returning approximate results may be achieved as one in this paper.

6.References

1. Bunke, H. (1997). "On a relation between graph edit distance and maximum common subgraph", Pattern Recognition Letters.
2. Bunke, H. (1998). "Error-tolerant graph matching: a formal framework and algorithms", in A.Amin, D.Dori, P.Pudil, and H.Freeman (eds.): Advances in Pattern Recognition, LNCS 1451, Springer Verlag, pp. 1–14.
3. Bunke, H. and Shearer, K. (1998). "A graph distance metric based on maximal common subgraph", Pattern Recognition.
4. Messmer, B.T. and Bunke, H. (2000). "Efficient subgraph isomorphism detection - a decompostion approach", to appear in IEEE Trans. on Data and Knowledge Engineering.
5. C. Ah-Soon and K. Tombre, "Architectural symbol recognition using a network of constraints," PatternRecognition Letters".
6. H. Bunke, "Attributed programmed graph grammars and their application to schematic diagram interpretation,'' IEEE Trans. on PAMI,Nov 1982.
7. J.T.L. Wang, K. Zhang, and Chim G.W., "The approximate graph matching problem," in Proceedings of l2th. International Conference on Pattern Recognition (b), October 1994, Jerusalem, Israel.
8. G. SQnchez, J. Llados, and K. Tomhre, "An errorcorrection graph grammar to recognize texture symbols," in Graphics Recognition: Algorithms and Applications, D. Blostein and Y.B. Kwon, Eds. Springer, Berlin, 2002, Vol. 2390 of LNCS.