Attributed relational graphs based description and retrieval of shoeprints in the image database system SHARS

Ch.Feist[1], *V.Kiiko*[2], *R.Knöfel*[1], *G. Stanke*[1]

[1]Gesellschaft zur Förderung angewandter Informatik e. V. (GFaI),
Albert-Einstein-Str. 16, D-12489 Berlin
Tel.:+49 (0)30 6392-1610, Fax: +49 (0)30 6392-1661, e-mail: stanke@gfai.de
[2]Int. Research and Training Centre of Information Technologies and Systems of National Academy of Sciences of Ukraine, Prospect Academica Glushkova 40, 252022 Kiev
Tel.:+380 (0)44 266-3018, Fax: +380 (0)44 266-1570, e-mail: kiyko@image.kiev.ua

Abstract

In this paper attributed relational graphs based approach implemented in the image database system SHARS for archival and retrieval of shoeprints from a database is described. The initial description of the shoeprints is a set of drawn by user figures, which then is transformed to an attributed relational graph (ARG) describing shoeprints in terms of some primitives and their relations. To retrieve shoeprints from the database this ARG is matched with all ARGs in the database using proposed exact matching algorithms.

1. Introduction

So called "geometric" approach for retrieval of images from the database is to get a structural description of the image as a set of extracted geometric features (lines, arcs, figures, etc.) and their relations and to match it to model descriptions or to descriptions of images in the database. The automatic detection of such features is a subject of many researches but still remains on in general unsolved problem. One of the recent works in this field [1] represents an algorithm for search or adjusting figures consisting of line segments and arcs in grey scale images. This algorithm is based on a computationally efficient procedure for defining similarity between two images.

One of the powerful tools for describing structured objects are attributed relational graphs (ARGs). The nodes of ARG represent primitives or subpatterns of structured objects, and branches between nodes represent mutual relations between these primitives [2]. To match two graphs we need to find a one-to-one correspondence between their vertices and between their edges such that an incidence relationship is preserved [3]. This matching of two graphs is called graph isomorphism. If the matching is fulfilled between a graph and a subgraph of another graph, then it is called subgraph isomorphism or graph monomorphism. Terms of exact and best matching of ARGs are also important. Under exact matching there is subgraph isomorphism between two graphs with the same attributes on correspondent vertices and edges. The best matching means an optimal solution of the subgraph isomorphism problem when the distance between two matched graphs is minimal.

A number of algorithms are proposed for matching attributed relational graphs [2,4-10]. It is known that the graph matching problem belongs to the class of *NP* complete problems and computational complexity of any conventional search algorithm may grow exponentially when increasing the size of matched graphs. Some proposed algorithms have polynomial complexity for special types of graphs but still have exponential complexity at the worst case.

In [10] the algorithm for detection subgraph optimal isomorphism between two ARGs is proposed which is known to have an $O(N^4)$ average case complexity and an $O(N^5)$ in the worst-case complexity. In [11] the algorithm for detection subgraph isomorphism between two planar graphs in time $O(c^{w \log w} n)$ is described where *w* and *n* are numbers of nodes in input and reference graphs and *c* is some constant value.

Widely used way to retrieve images from database that have common features with a current one is to get structural description of this image, transform it into an ARG, then to match it with ARGs representing structures of the other images in the database. In this paper some algorithms for description and retrieval of shoeprints along this way are considered.

The definition of the attributed relational graph and also steps of the description of the shoeprints in SHARS are shown in Section 2. Section 3 introduces some subgraph isomorphism based retrieval algorithms. Experimental results are presented in Section 4, and finally, the conclusions are given in Section 5.

2. Description of shoeprints

An attributed relational graph is defined as [5]

$$G=(N, B, A, E, G_N, G_B),$$

where
$N=\{n_1, n_2,\dots, n_{|N|}\}$ is a finite set of nodes; $|N|$ is the number of nodes in N;
$B=\{b_1, b_2,\dots, b_{|B|}\}$ is a set of ordered node pairs (or directed branches) i.e., $b=(n_i, n_j)$ for some $1\le i,j \le |N|$ denotes the branch emanating from node n_i to node n_j, and $|B|$ is the number of branches in B;
A is an alphabet of node attributes;
E is an alphabet of branch attributes;
$G_N: N\to 2^{|A|}$ is a function that defines the node attributes;

G_B:B→$2^{|E|}$ is a function that defines the branch attributes.

Let us consider now separate steps of description of the shoeprints and representing them by ARGs in image database system SHARS. The initial description of the shoeprint is a set of geometrical figures drawn by user in the image of this shoeprint. User can also define one or more characteristic subsets of this set, which must be on retrieved shoeprints.

Each of currently used figures (line, half of the circle, circle, polyline, polygon, square, triangle, rectangle, parallelogram, rhombus, rings, wave, tooth and position) contains so called control points and to draw or modify any figure in the image only these points must be defined.

On the next step every figure is represented by ARG with some attributes on its branches and nodes. Every node or branch of this graph has one context mark or attribute (centre of the circle, centre of half of the circle, end of line segment, line segment) and none or some of the following metric attributes: coordinates, radius, length and direction of the line segment, cross angle and relation of lengths of two neighbour line segments. The number of used metric attributes depends on some options of the program and also of the possibility for the user to define the position and orientation of the shoeprint in the image as well as the resolution of this image.

As an example let us consider attributed relational graph G_p of the figure „polyline" that consists of two line segments. This graph consists of three nodes and two branches. The nodes of the graph G_p have the context mark „end of line segment" and the branches have the context mark „line segment". The nodes of the graph have metric attributes "coordinates" if: 1) the resolution of the image is known, 2) the user has estimated the position and direction of the shoeprint in the image by drawing the figure „position" and 3) the option „Coordinates" is switched on. The middle node of the graph in any case has also metric attributes „lengths relation of neighbour line segments" and „angle between neighbour line segments". The branches of the graph have the metric attribute „length", if the resolution of the image is known, and the metric attribute „direction", if the user has defined the orientation of the shoeprint in the image and the option „Angle" is switched on.

It follows that

$$G_p=(N, B, A, E, G_N, G_B),$$

where

$N=\{ n_1, n_2, n_3 \}$;

$B=\{ b_1, b_2 \}$;

$A=\{$ end of the line segment, coordinates, lengths relation of neighbour line segments, angle between neighbour line segments $\}$.

$E=\{$length, direction$\}$.

Attributed relational graphs that define characteristic groups of figures in the shoeprint are similar to ARGs of the figure "polyline" but instead of the node attribute "end of line segment" they have the node attribute "label of figure".

3. Retrieval algorithms

Retrieval algorithms are based on matching (subgraph isomorphism) of the input ARG with ARGs of shoeprints in the database and also on the distance measure between these graphs. For calculating the distance between two ARGs the concept of error-correcting transformation [5,7,10] is adopted. The cost of matching two ARGs (or distance between these graphs) is defined as the cost of the sequence of transformations with minimal total cost that must be performed on one of the two ARGs in order to produce another ARG. These transformation operations are: *node insertion, node deletion, branch deletion, branch insertion, node label substitution and branch label substitution.* The costs corresponding to each operation are: w_{ni}, w_{nd}, w_{bi}, w_{bd}, w_{ns} and w_{bs} respectively. It is reasonable in our case to assume that $w_{nd}=w_{bd}=0$, $w_{ni}>0$, $w_{bi}>0$, $w_{ns}=w_{bs}=1$ and the distance $dist(x,y)$ between two numeric attributes x and y is equal to v=abs(x-y) if v exceeds some threshold value and to 0 otherwise. It can be defined that we have the exact matching if the distance between two graphs is equal to 0 and the best matching otherwise. In both cases the part of one graph must be defined that has the minimal distance with another graph but for exact matching this distance is equal to 0 and less computational time is usually required.

ARGs, corresponding to currently use in SHARS figures, are either chains or cycles. That is why the way of checking exact matching between two ARGs during moving lesser graph along greater one can be used. The computational complexity of this algorithm is $O(M*N)$.

There are some criteria for matching of input and reference descriptions of shoeprints implemented in the system SHARS. By default two descriptions are matched if at least one figure in input description is exact matched with one or more figures in the reference shoeprint. The user may also define the number of pairs of figures or group of figures in the input shoeprint that must be matched with ARGs of the reference shoeprints.

To match the group of figures we need to find subgraph isomorphism between two graphs. One of them is the complete (fully connected) graph such that nodes of this graph represent different figures in the shoeprint from the database and each of these nodes is connected to all other nodes of the graph. The other graph G represents the chain of figures in the input shoeprint. Each node of this graph has a label of the corresponding figure in the group and number of nodes in G is equal to number of figures in the group. The following algorithm checks some necessary conditions of subgraph isomorphism between these two graphs. During the first part of the algorithm for each node of the graph G the list of exact matched with this node figures from the reference description is constructed. During the next step all possible excludings of those figures from these lists that do not satisfy to local relations between the neighbour figures in the group are carried out. It means that each figure F must be excluded from the list if there are no in neighbour lists two other figures that satisfy with F to local relations of neighbour relations of figures in the group. The process of excluding figures is stopped when additionally none of the figures can be excluded. Therefore the input and reference graphs are considered as matched if the final lists of these figures are not empty in spite of the fact that there is exactly subgraph isomorphism between these graphs only when each of lists has only one label or figure and all these labels are different.

For matching ARGs thar represent figures of other types than currently used in SHARS one of the algorithms [2,4-10] can be implemented. For this aim new algorithm for exact matching ARGs was also developed which is described below.

The main problem using the error-correction distance measure is the computational complexity of the matching that may grow up exponentially when the graph size increases. To reduce the complexity original graphs can be decomposed into smaller subgraphs (basic attributed relational graphs or BARGs) and the matching of the original graphs to fulfil through the matching of these decomposed subgraphs [10]. The BARG represents a one-level tree, which consists of a node (root), and every node (descendant) connected to that node by one emanating or incoming branch. The cost of matching two BARGs G_k and G_p can be defined as

$$\text{Distance}(G_k, G_p) = w_{ns} * \text{dist}(r_k, r_p) + \text{dist}(b's, e's),$$

where

$\text{dist}(r_k, r_p)$ is the distance between root node labels and attributes of two BARGs that is computed depending on their data types,

k and p are the number of branches connected to the root node BARGs being matched, and

dist(b's, e's) is calculated as the minimum of a weighted bipartite graph constructed from branches of BARGs as their nodes and its structure is as shown in Fig. 1.

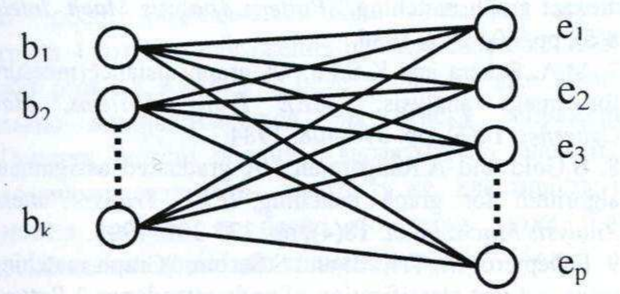

Fig. 1. Weighted bipartite graph corresponding to branches distance.

The minimum of a weighted bipartite graph is defined as the complete matching with the minimum sum of the matching edges weights. There are some algorithms for such matching with an $O(N^2)$ average case complexity and an $O(N^3)$ worst-case complexity [12].

Exact matching between branches of two BARGs means that dist(b's, e's) = 0, i.e., there is one-to-one correspondence between branches of these BARGs and correspondent branches have the same attributes. To define such exact matching it is necessary to exclude from bipartite graph in Fig. 1 those branches that have non-zero weights and then to use one of the algorithms f.e., [13] to find assignment for remain branches.

Let G_i and G_r be input and a reference ARGs. The first step of the algorithm for checking exact matching between these two graphs consists in their decomposition into BARGs. Then the distances between decomposed BARGs are defined, it means that a distance matrix D between BARGs of reference and input graphs is constructed. The $D_{i,j}$ element of this matrix represents the distance between the ith BARG in the input graph and the jth BARG in the reference graph, and $D_{i,j}=0$ if there is exact matching between these two BARGs and $D_{i,j}>0$ otherwise. On the base of distance matrix for every node of lesser graph (having less nodes than another one) a list of BARGs of greater graph that have exact matching with this node and not lesser number of connected branches is made out.

During the next step of the algorithm we scan constructed lists of BARGs and exclude every BARG B in the current list if the list of BARGs in the neighbour node is either empty or contains only BARGs that are not neighbours to B in the reference graph. During this step can be also helpful some additional tests, f.e. : a) if some list L contains only one BARG this BARG can be excluded from all other lists or b) if the number of different BARGs in all lists less than the number of nodes in input graph the input and reference graphs can not be matched.

This process is stopped when additionally none of the BARGs can be excluded. Therefore the input and reference graphs are not matched if the final lists of BARGs are empty and are exact matched if each of these lists contains only one BARG and all these final BARGs are different. In the case when final lists contain more than one BARG additional testing is needed.

Let us estimate computational complexity of the algorithm. Suppose that the input and reference graphs have M and N nodes relatively. To decompose both G_i and G_r into BARGs we need quadratic time for each graph, i.e., this step has a time complexity of $O(M^2+N^2)$. The computational complexity of exact matching between two BARGs is $O(M*N)$, assuming that any node is connected to all other nodes in the graph, and the computational complexity of calculating the distances between all BARGs of both input and reference graphs is $O(M^2*N^2)$. The computational complexity of the second (BARGs excluding) step of the algorithm is $O(M^2*N^3)$, $M<N$ in the worst case. In summary, the computational complexity of the proposed algorithm for checking the exact matching between two ARGs is also $O(M^2*N^3)$ in the worst case.

4. Experiments

During experiments 150 shoeprints were encoded and saved in the database. After that the testing group of the 25 different shoeprints was formed and shoeprints of this group were then encoded and retrieved by some users.

Each shoeprint from this group has one or more shoeprints of the same type in the database, which differ from each other by size, position and many other features, for example some of these shoeprints are only parts of other ones (Fig. 2).

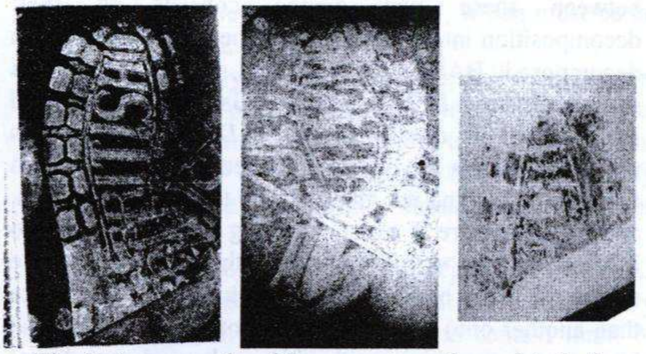

Fig. 2. An example of the same type shoeprints in the database.

Many shoeprints in the database are not supplied by metric information because of the absence of data about resolution of the image or the position of the shoeprint in this image. The number of retrieved shoeprints by one figure or by one pair of figures for such shoeprints is not so small. As follows from experimental data the most effective possibility for the retrieval of most shoeprints from a database consists in defining of a group of figures in the input shoeprint. This possibility provides a sharp reduction of number of retrieved shoeprints and preserves the majority of the same type retrieved shoeprints from the database.

It is important that retrieval of the shoeprints from database is fulfilled using not the names and parameters of figures but using attributed graphs that represent these figures and also on the base of subgraph isomorphism of these graphs. This fact reduces dependence of the retrieval results on subjective actions of the user during describing or coding input shoeprints because in many cases the same object in the shoeprint can be described by figures that have different names and/or different numbers of its components but attributed graphs that represent these figures can be matched. Introduced options and some possibilities for the definition of an inquiry to a database (retrieval by figure, by pairs or by groups of figures) can be also useful in this direction.

5. Conclusions

In this paper ARG based approach and algorithms for the retrieval of shoeprints from a database are proposed. The proposed algorithms are based on the exact matching of ARGs that describe input and reference shoeprints. The distance is calculated based on the concept of error-correcting transformations with taking in account some peculiarities of the shoeprints description and retrieval problem. From experimental results the following remarks can be concluded.

The developed tools can be used for the description and retrieval of shoeprints from a database. The proposed algorithms for detection subgraph isomorphism between input and reference ARGs and also the introduced different possibilities for the definition of an inquiry to a database reduce dependence of the retrieval results on subjective actions of the user during describing or coding input shoeprints. On the next step the algorithms for optimal subgraph isomorphism will be tested and added to the system for the further reduction of this dependency.

The implemented ARGs based tools and algorithms are an addition but not a substitution for the other method implemented in SHARS that is based on coding shoeprints by rather global geometrical features, that describe shoeprint in its different parts.

References

1. I.Aksak, Ch. Feist and others, "Detection of the Objects with the Given Shape on the Grey-Valued Pictures," CAIP'97 "Computer Analysis of Images and Patterns", 1997, Kiel, p.551-558
2. A. Sanfelio and K.S. Fu, "A distance between attributed relational graphs for pattern recognition," *IEEE Trans. Systems, Man Cybernet.* **13(3)**, pp. 353-362, 1983.
3. Narsingh Deo, "Graph Theory with Applications to Engineering and Computer Science," Printice-Hall, Englewood Cliffs, New Jersey, 1995.
4. A.K.C. Wong, M. You and S.C. Chan, "An algorithm for graph optimal monomorphism, *IEEE Trans. Systems, Man Cybernet.* **20(3)**, pp. 628-636, 1990.
5. W.Tsai and K.S.Fu, "Subgraph error-correcting isomorphism for syntactic pattern recognition," *IEEE Trans. Systems, Man Cybernet.* **13(1)**, pp. 48-62, 1983.
6. M.A. Eshera and K.S.Fu, "An image understanding system using attributed symbolic representation and inexact graph matching," *Pattern Analysis Mach. Intell.* **8(5)**, pp. 604-617, 1986.
7. M.A. Eshera and K.S.Fu,"A graph distance measure for image analysis," *IEEE Trans. Systems, Man Cybernet.* **14(3)**, pp. 398-408, 1984.
8. S.Gold and A.Rangarajan, "A graduated assignment algorithm for graph matching,"*IEEE Trans. Pattern Analysis Mach. Intell.* **18(4)**, pp. 377-388, 1996.
9. F.Depiero, M. Trivedi and S.Serbin, "Graph matching using a direct classification of node attendance," *Pattern Recognition,* **29(6)**, pp. 1031-1048, 1996.
10. Yasser El-Sonbaty and M.A.Ismail, "A new algorithm for subgraph optimal isomorphism," *Pattern Recognition,* **31(2)**, pp. 205-218, 1998.
11. D.Eppstein, "Subgraph isomorpism in planar graphs and related problems," Department of information and computer science, University of California, Irvine, CA 92717, Tech. Report 94-25, p. 27, 1994.
12. S.Yamada and T.Hasai, "An efficient algorithm for the linear assignment problem," *Elect. Comm. Japan,* Part 3, **73(12)**, pp. 28-36, 1990.
13. O.Ope. Теория графов. Москва, 352 с, 1968.