MATCHING GRAPHS WITH FUZZY ATTRIBUTES IN MACHINE VISION

G.A. Bilodeau and R. Bergevin
Laboratoire de vision et systèmes numériques, Pavillon Adrien-Pouliot
Université Laval, Sainte-Foy (QC), Canada, G1K 7P4
bilodeau@gel.ulaval.ca, bergevin@gel.ulaval.ca

Abstract

In object recognition and image querying applications, complex graphs often have to be compared to verify the similarity between two models. Since there is always uncertainty while models are constructed, the nodes and the edges require fuzzy attributes to properly describe the scene or the object. This paper addresses the problem of matching graphs with fuzzy attributes (GFAs) obtained by hypothesizing volumetric primitives from 2D parts. The GFAs of interests have nodes with many fuzzy attributes that correspond to volumetric hypotheses, and edges that describe the spatial relationship between the hypothesized volumetric primitives. A model for representing 2D parts by volumetric primitives is presented. Then, a method using structural indexing adapted to GFAs is proposed. This inexact matching method has been designed for matching GFAs in large databases.

Key Words

Part graph, structural indexing, fuzzy attributes, graph matching, object model.

1. Introduction

In object recognition and image querying applications, graphs are often used to model scenes or objects in the image. Since creating a model involves interpretation phases, there is always uncertainty. Hence realistically, graphs with fuzzy attributes (GFAs) must

be used. Taking our application as an example [1], the fuzzy attributes may be all the possible volumetric primitives that can be inferred from a part. Hence, one node (a part) of a graph can have many fuzzy attributes (each possible volumetric primitive and a fuzzy ranking value). The edges of the graph can also have many fuzzy attributes (e.g. the different ways parts may be connected). How can such graphs be compared?

A significant research effort ([2],[3],[4]) has been devoted to inexact matching of attributed graphs. However, in general, these studies do not address the problem of matching inexact graphs, i.e. GFAs. An exception is the research of Christmas et al. [4]. Their method can match graphs where the attributes are fuzzy. However, because they are using probabilistic relaxation, their method needs a convergence criterion on the probability. A method with no convergence criterion would be more appropriate to image retrieval applications as the level of similarity attainable is not known a priori. In addition, their method is more adapted to find a similar graph, than to rank graphs based on a query graph. Chan and Cheung [5] have studied matching of a GFA and an attributed graph. Their method does not apply to matching pairs of GFAs. Perchant et al. [6] and Medasani et al. [7] have addressed this specific topic. Their approaches imply sequential matching of pairs of graphs using relaxation algorithms or genetic algorithms.

Because only few works have been devoted to graphs with fuzzy attributes, matching of attributed graphs has also been studied to assess if the method used could be adapted to graphs with fuzzy attributes. To avoid matching numerous graphs one by one, two main types of approaches exist. Matching by decision trees ([8],[9]) and matching by indexing ([10],[11]). Decision trees require a model graph without uncertainties. Since our application of image querying requires fast matching, similarity rankings of many

graphs, and do not use reference models of objects, an approach using indexing is more appropriate. Besides, it is less computationally expensive [10] and well adapted to rank graphs. To compare graphs quickly, structural indexing is a good choice as it compares graphs by matching their subgraphs independently. However, structural indexing as defined and used in the literature cannot be directly applied to GFAs.

In this paper, a model for representing 2D parts by volumetric primitives is presented and structural indexing is extended to compare GFAs. The extension of structural indexing to GFAs is the main contribution of this paper. This extension is not trivial as it is exemplified in Section 3.2, yet essential in computer vision applications.

The paper is organized as follows. Our interest in matching GFAs is explained by presenting our application (named PLASTIQUE) in Section 2. Section 3 details the fuzzy matching problem and describes the matching method. Section 4 gives results of experiments aimed at validating our structural indexing method and its applicability in matching graphs of volumetric primitives. Section 5 concludes the paper.

2. GFAs matching context

The application under development, named PLASTIQUE (Parts, Links, and ASsociated Templates Image QUery Engine), aims at querying an image database of manufactured objects, which are interpretable as arrangements of simple volumetric primitives. Images in the database are from real scenes with one main object in the foreground that must be detectable in the image (see [12]).

To query or add an image in the database, the user gives as input an example 2D image or a sketch of the 3D object. The image is first processed to obtain contours of

linked local intensity edges that are segmented to produce a map of constant curvature primitives (CCPs) [13]. An initial grouping of the CCPs produces the outline of the object [12]. The CCP map is then processed further to obtain parts using the extracted outline [14] [15]. These parts are labelled based on the possible volumetric primitives that may project onto them [15]. Spatial relationships between parts are also computed at this step. Finally, the constructed model is compared with models in the database.

Parts found in an image are not matched directly. A direct matching of the parts would not account for viewpoint variations. For this reason, volumetric primitive hypotheses are inferred from the parts. Parts for a lamp are shown in Fig. 1A. The goal of the 3D inference is to establish which volumetric primitives, when projected in an image, may look like the parts found.

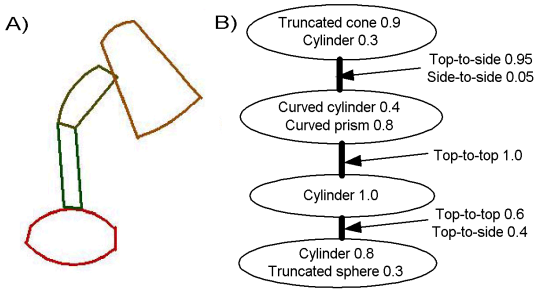


Figure 1. Example of parts for a lamp and its associated graph.

Graphs are constructed from the spatial relationships of the parts and from hypothesized volumetric primitives. Volumetric primitive hypothesizing is performed using a rule-based classifier. The rules are based on measures on the boundary and the shape of each part. These measures verify the convexity and the symmetry of the part, the type and arrangements of the constant curvature primitives (circular arc or straight line segment) making the boundary. When a part verifies a rule, one or more ranked volumetric primitive hypotheses are generated. Since parts in an image each corresponds

to a node of the graph under construction, each hypothesized volumetric primitive corresponds to a node attribute. More specifically, nodes attributes are the labels of each hypothesis associated with a fuzzy value (see Fig. 1B). The constructed graph will have nodes with one or more attributes that have values between 0 and 1 reflecting the ranking between the hypotheses. Edges of the graph correspond to the spatial proximity relationships between the parts. Edges have attributes with values between 0 and 1 that describe the connections between the hypothesized volumetric primitives. The edges attributes consist of a label that describes a connection type and a fuzzy value (see Fig. 1B). Edges are undirected.

3. Our approach to matching

Our approach to match these graphs is to use structural indexing. Instead of attempting to compare graphs pairwise, their level of similarity is established by a voting mechanism and by comparing subgraph structures.

3.1 Details on the matching problem

The constructed graphs have nodes with 18 fuzzy attributes and edges with 4 fuzzy attributes. Each attribute has a fuzzy value between 0 and 1. The attributes for the nodes are labels identifying the 18 volumetric primitives (see [15]) that can be hypothesized from each part. The attributes for the edges are labels that identify how the volumetric primitives are connected. Since objects have parts that interconnect in complex fashions, nodes can form cycles, and because of processing errors, some nodes may not be connected.

These graphs must be matched in the context of an image database query engine. Thus, it has to be done quickly, and matching cannot rely on perfect models of objects, because the same processing modules model both query and database images.

3.2 Our method introduced by an example

The following matching example has been designed to show how the data is structured for matching, and the difficulties specific to indexing graphs with fuzzy attributes. Furthermore and more importantly, it gives an overview of our method. Specific details of the method appear in next section.

In this example, a query graph (Graph Q) is to be matched with two graphs (Graph DB1 and DB2) in a database. These three graphs are shown in Fig. 2. Attributes of each node and edge are represented by a letter. Attributes are volumetric primitives or a type of connexion, associated to a fuzzy value.

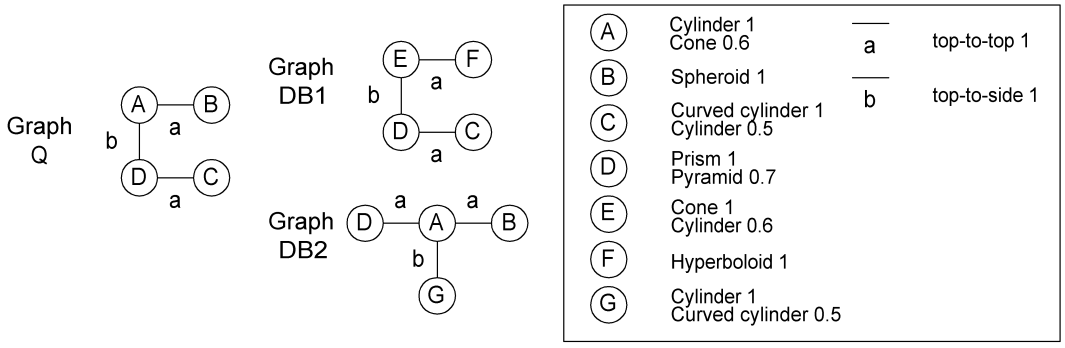


Figure 2. Graphs used for the example. The legend at the right identifies the attributes of the nodes and of the edges.

To match graph Q with graphs DB1 and DB2, graphs DB1 and DB2 are first decomposed into subgraphs, named SGEs (SubGraph Entities). These SGEs are composed of a node, two nodes and an edge, or three nodes and two edges (Fig. 3). For

SGEs with more than one node, composed attributes are constructed by concatenating the attributes of the nodes and the edges.

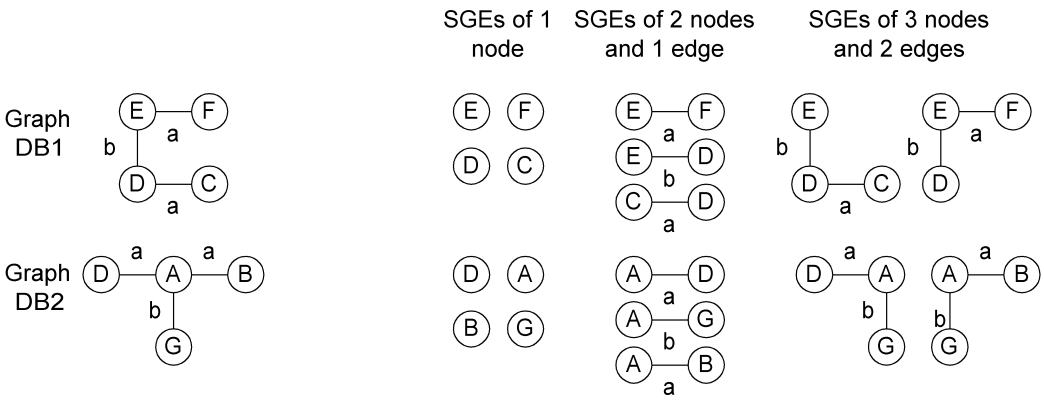


Figure 3. Graphs DB1 and DB2 are decomposed into SGEs.

These SGEs are then inserted into an index (Fig. 4). The index has entries for each possible volumetric primitive and all possible combination of two volumetric primitives and a connexion type, and of three volumetric primitives and two connexion types. A SGE can appear in many index entries since, a SGE usually has more than one attribute. For example, this is the case for the node identified by E in graph DB1. It appears at the *Cone* and *Cylinder* entries. Hence, the same SGE can be indexed from different entries. Since nodes of graph are to be matched one-to-one, a mechanism is needed to ensure that a SGE is used only once for the graphs in the index. This is a difficulty that we resolve with our data structure.

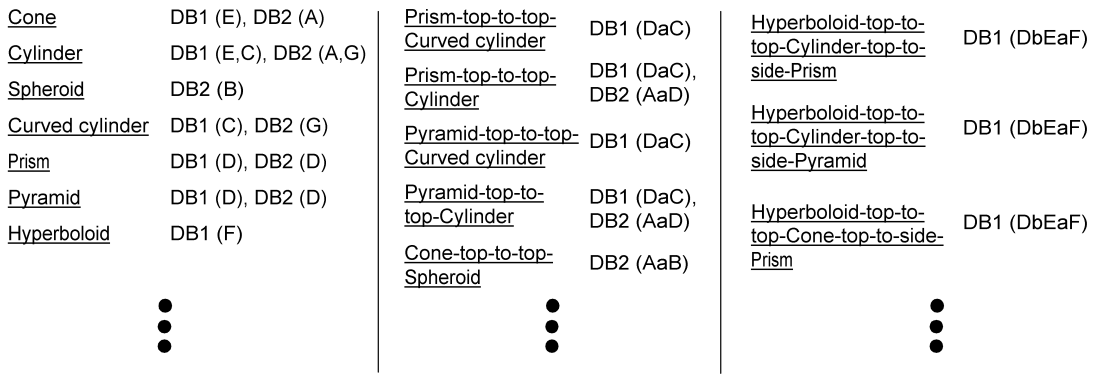


Figure 4. Graphs DB1 and DB2 are inserted into an index. There are index entries for all possible attributes.

When the graphs have been inserted in the database, the matching process begins. First of all, the query graph (Graph Q) is decomposed into SGEs (Fig. 5).

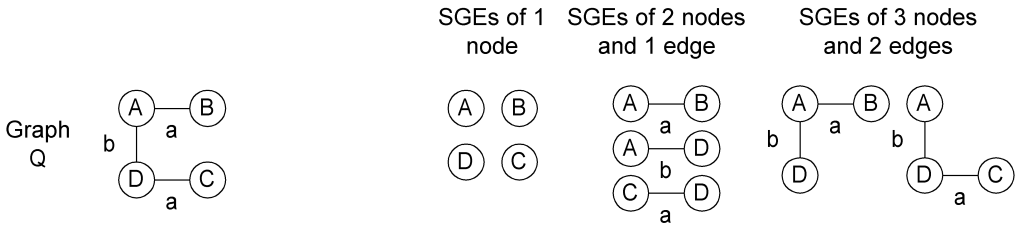


Figure 5. Graph Q is decomposed into SGEs.

The SGEs from Q of the three types are then sorted by increasing uncertainty. This is done to first match SGEs with fewer attributes and spread out values of attributes. These SGEs are considered less ambiguous. Matching is then done as follows. Starting with the SGE that have the less uncertainty, its best attribute (or composed attribute) is identified. Then, the index is searched to find graphs having an SGE with a matching attribute. For the SGE composed of node A (Fig. 6), the attribute with the best value is the cylinder. Both the graphs DB1 and DB2 have a SGE with a cylinder (see Fig. 4). Hence, these two graphs will receive a weighted vote proportional to the similarity of all the attributes of the matched SGEs. A method for the calculation of the weighted votes has been defined. Note, that graph DB1 has two SGEs (composed of single nodes identified by E and C)

with a none-zero cylinder attribute. The match is done with the unused SGE that have the highest value for the attribute (Node E). Hence a mechanism is necessary to track which SGEs in the index have been matched, and which one has the highest value for a given attribute.

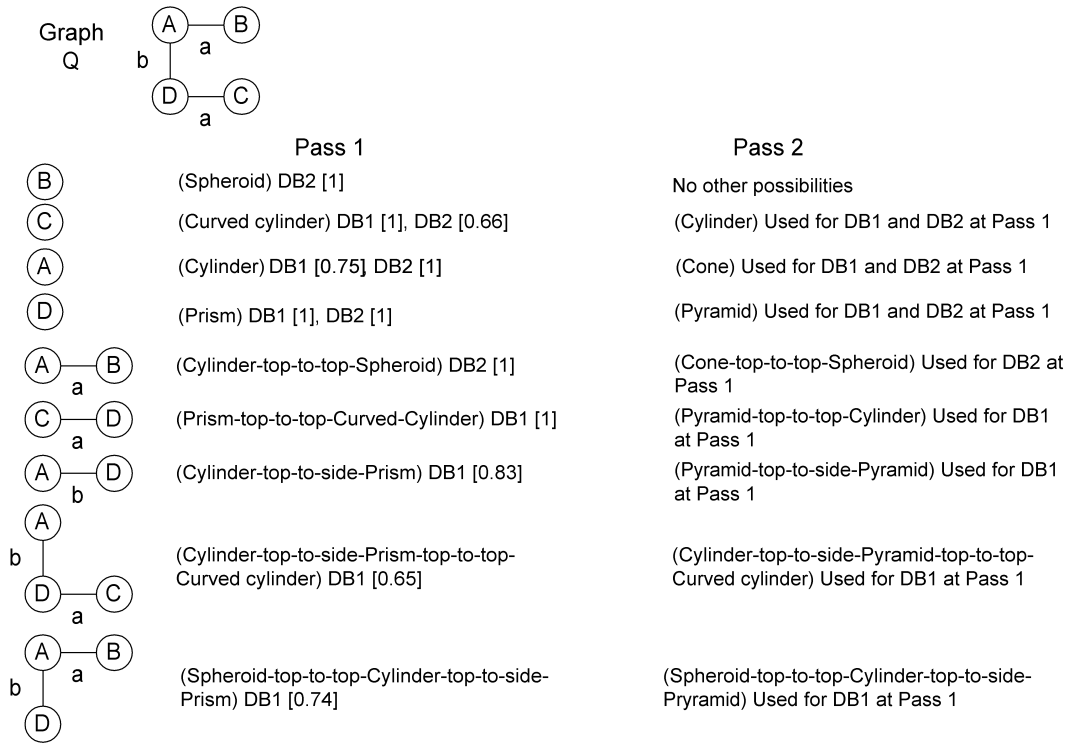


Figure 6. The SGEs of Q are used to search the index in consecutives passes, each with a different attribute. The graphs matched are identified along with the matching score (in bracket) obtained. In parenthesis, the attribute used for matching is given.

Lastly, the matches are done in consecutives passes. First, with the attribute of the SGEs of Q that has the highest value (Pass 1 in Fig. 6), and then with the attribute that has the second highest value (Pass 2 in Fig. 6), and so on. Hence, it is necessary to know which SGEs of graph Q have been matched with the graphs in the index to ensure a one-to-one correspondence. When all the passes have been completed, the total score for each graph in the database is computed.

3.3 Details of our GFAs matching method

Specific details of the method left out in the previous overview are now presented.

3.3.1 Decomposing to SGEs

Graphs are decomposed into subgraph entities (SGEs). The SGEs are single nodes, groups of two nodes and an edge, and groups of three nodes and two edges. Recall that nodes have 18 attributes and edges have 4 attributes. Therefore 94625 (18 + 18*4*18 + 18*4*18*4*18) index entries are required to cover all combinations of volumetric primitives and connection types. Combined attributes are used for SGEs with more than one node. The combined attributes are obtained by concatenating the attributes of the nodes and the edges for each possible combination of the hypotheses associated to nodes and edges for a given subgraph. For example, for the SGE AbD (Node A, edge b, and Node D) for graph Q in Fig. 5, four different combined attributes (Cylinder-top-to-side-Prism 1, Cone-top-to-side-Prism 0.6, and so on) are obtained.

3.3.2 Sorting by uncertainty

For the query graph, SGEs are sorted by increasing uncertainty. In our case, the uncertainty is calculated by:

$$Uncertainty = 1 - \frac{FVal_{\max}}{\sum FVal} \qquad (1)$$

where $FVal_{max}$ is the maximum fuzzy value for any attribute of the SGE and $\sum FVal$ is the sum of all the fuzzy values of the attributes of the SGE. Our measure estimates how ambiguous the volumetric primitive hypotheses are. This definition of uncertainty is different from the entropy definition used in fuzzy logic, because the fuzzy values of all

attributes in a SGE do not sum to 1. Our uncertainty measure is based on the fact that a SGE is less ambiguous when its best attribute has a fuzzy value much larger than the other attributes. The sorting is done to ensure that the matching order does not influence results. Parts of objects that have been hypothesized with less ambiguity are matched first.

3.3.3 Adding SGEs to the index

Before being added to the index (database), the SGEs can be truncated to the first *n* best attributes. The number of attributes of a SGE to include in the database depends on how much memory and disk space is available for the index. In our current implementation we have chosen to truncate to the first forty best attributes. The chosen truncation does not significantly affect matching results because the fuzzy values of the truncated attributes are usually very low. For 312 graphs, the index size is about seventeen megabytes.

In each entry of the index, the graphs that have none-zero values for the attribute corresponding to the entry are listed (Fig. 4). A graph can be in an entry multiple times if it has the attribute corresponding to the entry for more than one SGE. To allow access to a SGE in the database not only from its best attributes, the SGE is inserted in the entries for its *n* best attributes. Hence, SGEs with different best attributes can be matched (see Fig. 6 where the SGE composed of the attribute set A is matched with a SGE having attribute set E). However, this requires more monitoring to ensure one-to-one correspondence.

3.3.4 Computing the matching scores

When two SGEs are matched, the matching score has to be computed. This is done by adding the minimum values of each none-zero attributes. Formally, the matching score for two matched SGEs is:

$$MS = \frac{\sum_{i \approx j} \min(RV_Q(i), RV_D(j))}{\sum_{i \in SGE_Q} RV_Q(i)} \qquad (2)$$

where SGE_Q is the query SGE, and $RV_Q(i)$ and $RV_D(j)$ are the fuzzy values for the ith or jth attributes of the query and of the database SGE respectively. Symbol $i \approx j$ means that the sum is computed only for matching hypotheses in the two SGEs. The total score for two graphs (for all SGEs of both graphs) is given by:

$$TS = a \sum_{i=1}^{3} \left(\frac{SGEQ_i}{SGEDB_i} \right) + \sum_{i=1}^{3} \left(\frac{b_i \sum MS_i}{\min(SGEQ_i, SGEDB_i)} \right) \qquad (3)$$

where $SGEQ_i$ and $SGEDB_i$ are the number of SGEs with i nodes in the query and in the index graph, respectively. MS_i is a matching score for a SGE of i nodes, and a and b_i are weighting coefficients. The four weighting coefficients are adjusted dynamically: the smaller the term a coefficient multiplies, the larger its adjusted value. This allows to emphasize the differences between the two graphs. The sum of the weighting coefficients is 1 and, in the current implementation, the coefficients are selected from sets of fixed values based on the value of each term of *TS*. The value for *TS* varies between 0 and 1, where 1 is a perfect match between two graphs.

3.3.5 Ensuring one-to-one correspondences

When matching, the one-to-one correspondence between the SGEs of two graphs is ensured by using a proper structure for the index, and by using a special vote accumulator. In the index, fields are used to record which SGEs have already been matched (Fig. 7A). Since a SGE can be in many different entries, when it is matched, it has to be set as being used in all the entries. This is done by indexing and scanning the entries for the other attributes of the SGE. SGEs have unique identifiers.

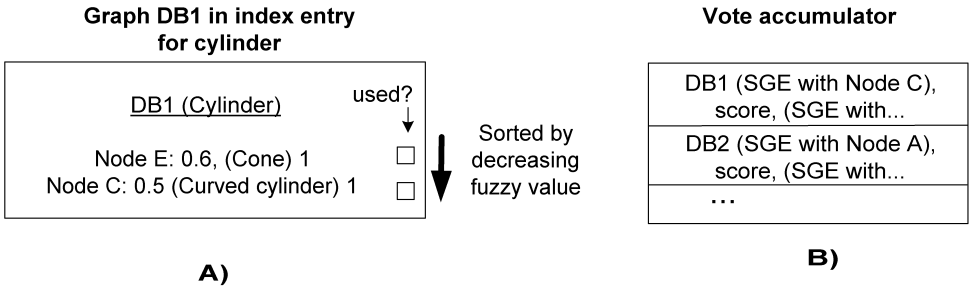


Figure 7. A) The structure used to insert a graph in an index entry, B) The structure used to record

The vote accumulator, is a simple structure that records the number of weighted votes each graph obtains. It also records how many SGEs have been used for matching, and their identifier to avoid multiple matches with the same SGEs.

3.3.6 Ensuring results consistency

To ensure results consistency, matching scores are optimized. This is done by performing matches in consecutive passes. The first pass indexes the best attributes in each SGE of the query. The second pass indexes the second best attributes, and so on. This is done to consider cases where an object or part of an object is seen from another viewpoint, and where noise or errors slightly deform parts. Therefore, a partial SGE match must be considered. Indexing of the SGEs of the query must be done in a constant and ordered

fashion to ensure consistency in the values of *TS*. A change in the order of indexing change the value of *TS* for two matched graphs. This is why a fixed multi-pass procedure is used to index the database.

For the index, SGEs (composed of node E and node C in Fig. 7) are sorted on the fuzzy values of the attribute to optimize matching, as this approximates sorting by uncertainties. Hence, the most unambiguous SGEs of both the query graph and the graph in the database are matched first.

4. Experiments

In this section, the matching performance of our method is analysed by two experiments. The first experiment investigates how the total matching score (*TS*) varies when a graph is matched with a deformed version of itself. The second experiment studies the ability of the combination of the volumetric primitive model and the matching method to group images of parts from six objects. It is compared to human abilities.

4.1 Behaviour of the matching method

This experiment has been designed to study how the total matching score for two graphs varies when nodes are removed, added, modified, or rearranged. A synthetic six-node graph has been added to a database of 312 graphs of the same average size that were extracted from a variety of images. Modified versions of this graph have been used as queries. Fig. 8 shows the results obtained for various modifications. Because of the formulation of *TS*, it varies about similarly for the different modifications, particularly for node addition and removal. Hence, graphs with a different node, or with one node less or one node more than the query are considered as similar or dissimilar to the query. This is

a behaviour we are looking for since *TS* must have a similar value if we compared graph A based on graph B and graph B based on graph A.

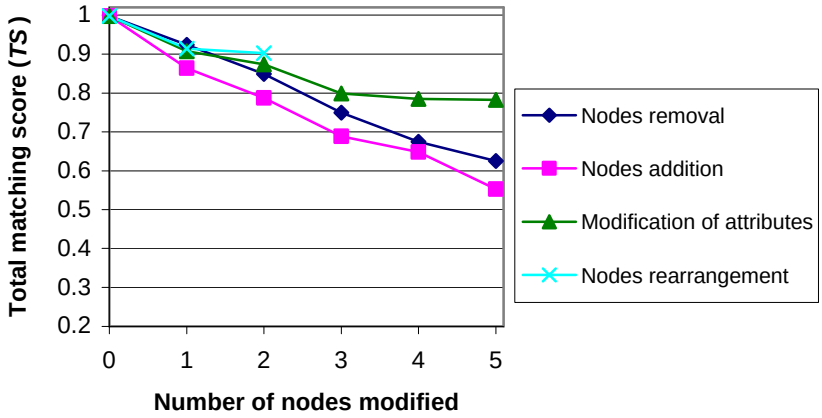


Figure 8. Behaviour of the total matching score when a graph is compared with modified versions of itself.

Experiments have shown that when each of the 312 graphs in a database is used as a query, the best graph match is the graph corresponding to the query graph [16]. Hence, although our method is an inexact matching method, it is exact enough to differentiate a graph from another graph, while it is flexible enough to account for some differences. This is shown by Fig. 9 which is from the same experiment as for Fig. 8. It shows how the original graph ranks for the total score (*TS*) for each type of modification. Fig. 9 shows that for all types of modification, the original graph ranks first if two nodes or less have been modified.

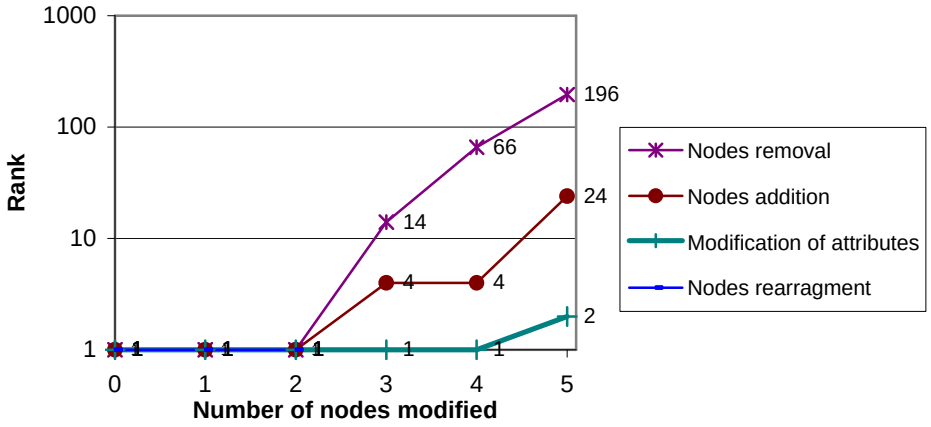


Figure 9. Behaviour of the rank when a graph is compared with modified versions of itself.

4.2 Grouping performance compared to human cognitive abilities

This experiment has been designed to assess the abilities of our volumetric primitive model and matching method to group images of parts obtained for six objects viewed in 195 images (about the same number of images for each object). Fig. 10 gives, for each object, two example images and associated parts used for this experiments. This experiment verifies whether the proposed 3D model and matching method allow PLASTIQUE to abstract viewpoint, variation of shape within an object family, and processing imperfections. To do so, PLASTIQUE is compared with six humans that have performed a similar clustering task. Among the six humans, four are researchers in the field.

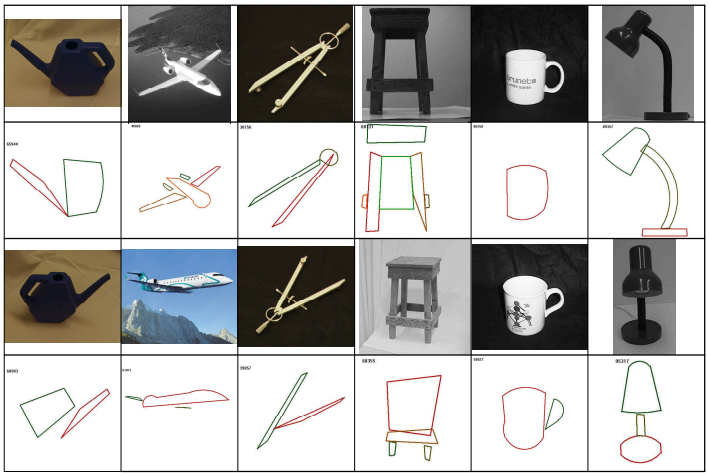
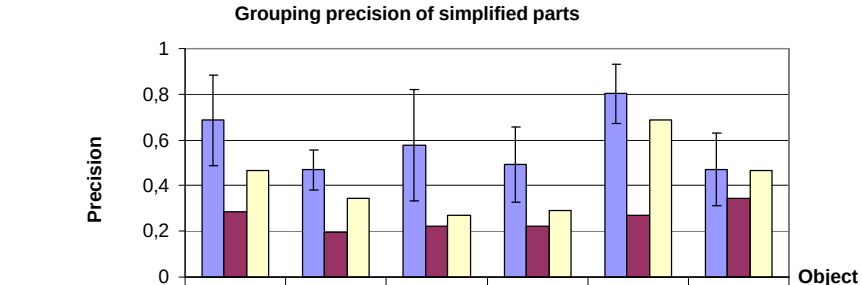


Figure 10. Example of images used for the grouping performance experiment.

The experiment with human subjects has been conducted as follows. Printouts of the parts of the 195 images have been given to the human subject. They have been asked to classify the printouts in groups they thought corresponded to the same object. It has been mentioned to the human subjects, that the objects could be seen from different viewpoints and deformed. The number of groups to form was not specified.

The same clustering task has been performed with PLASTIQUE. Images of parts were grouped together if the total matching score, with an image of parts given as a query, was within a threshold. Then, the grouped images of parts were removed from the index, and another query (an image of parts still in the index) was used. This process has been done until the index was empty. This task has been performed with two thresholds: 60% and 70%. That is, the matching score had to be higher than 0.6 and 0.7 (a perfect matching score being equal to 1). Threshold values larger than 70% give very small groups. Hence, these values are not of interest. Furthermore, threshold values smaller than 60% give poor grouping performance, as a lot of graphs are considered alike.

The graph of Fig. 11 shows the precision obtained for each object. The precisions have been computed as follows. The number of images or printouts in each group where an instance of a particular object was found has been summed. The number of images of the particular object in the database was divided by this value. This precision value reflects how many images or printouts of parts had to be considered so the human subjects or PLASTIQUE could find all the images of the particular object. The smaller the value, the larger the number of images considered before finding all the images of an object.



	VR Room Lamp	Coffee cup	Airplane	Stool	Compass	Watering can
Human average	0,6854469	0,4683908	0,5770591	0,4943092	0,8024038	0,4706508
PLASTIQUE 60%	0,2866242	0,1979167	0,2222222	0,2231405	0,2721088	0,3434343
PLASTIQUE 70%	0,4639175	0,3454545	0,2678571	0,2934783	0,6896552	0,4657534

Figure 11. Grouping precision of parts.

Except for the watering can, the human subjects outperform PLASTIQUE on average. Considering that human subjects have a priori knowledge about the objects grouped in the experiment, PLASTIQUE suffers from a major handicap. PLASTIQUE does not have a priori models of objects. It simply compares the parts by hypothesizing and comparing 3D shapes and verifying their relationships. The human subjects on the other hand can guess the identity of objects by the area or the shape the parts are forming altogether. In any case, PLASTIQUE using a 70% threshold approaches the lowest

precision level achieved by human subjects. We believe, this part of our system can be improved to reduce the gap between PLASTIQUE and human subjects. For example, the use of global shape criteria on the area formed by all the parts may significantly increase the precision of PLASTIQUE. Note that human subjects had difficulty recognizing objects such as the stool, the coffee cup and the watering can. This is because the parts extracted do not always adequately represent the object in the images.

It can be noted from Fig. 11 that some complex objects are grouped with more precision then simple objects, even by humans. It is the case for airplanes and coffee cups. This is explained by the parts obtained for these objects. For instance, the parts obtained for the airplanes capture more their shapes than those for the coffee cups, because our part segmentation algorithm is not aware of holes in objects [14] (see parts of stools and coffee cups in Fig. 10). Therefore, the handle is not well segmented from the body of the coffee cup.

5. Conclusion

This paper has presented a novel method for inexact matching of graphs with fuzzy attributes (GFAs) and a 3D model of representation for 2D parts. We first stated that matching GFAs is essential to many computer vision applications since interpretation phases are always uncertain, as it is the case for our volumetric primitive inference method. These uncertainties can be accounted for by using GFAs.

Our matching method adapts structural indexing to GFAs. The use of an uncertainty measure and adapted data structures allows our method to handle the fuzzy nature of the graphs. Our method ranks graphs by similarity based on a query graph. Matching is done

quickly as the models are not compared sequentially. For 312 graphs, matching and displaying the results take less then one second on a Athlon 1.2Ghz. The complexity of our matching method is $O(nm)$, where n is the number of SGEs of the query and m is the number of graphs in the database. This is the worst-case complexity where each graph in the database has at least an SGE at each row of the index. Furthermore, the size of the index can be adjusted to fit one needs of matching accuracy and computer memory utilization. For 312 objects and SGEs truncated to 40 hypotheses, the memory requirement is about seventeen megabytes.

The results obtained thus far show that our method can match GFAs adequately. When an image in the index is queried, it ranks first for the matching score. Also, slightly different graphs obtain matching values that are close to the query. Grouping images of parts using our method gives results that are still inferior to human subjects. However, this experiment allowed us to gain valuable knowledge and discover new ways to improve our method. These improvements will be the subject of future work.

6. Acknowledgements

This work is supported by a postgraduate scholarship from the Natural Sciences and Engineering Research Council of Canada (NSERC) and from le Fonds pour la Formation de Chercheurs et l'Aide à la Recherche (FCAR).

References

[1] Bilodeau, G.A., Bergevin, R., PLASTIQUE: Image Retrieval Based on Cognitive Theories, *Proc. Vision Interface 2003*, Halifax, Canada, 2003, 292-298.

[2] Messmer, B.T., Bunke, H., A New Algorithm for Error-Tolerant Subgraph Isomorphism Detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(5), 1998, 493-504.

[3] Tirthapura, S., et al, Indexing Based on Edit-Distance Matching of Shape Graphs, *Proc. SPIE Multimedia Storage and Archiving Systems II*, Dallas, TX, 1998, 25-36.
[4] Christmas, W.J., Kittler, J., Petrou, M., Structural Matching in Computer Vision Using Probabilistic Relaxation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *17*(8), 1995, 749-764.
[5] Chan K.P., Cheung, Y.S., Correction to "Fuzzy-Attribute Graph with application to Chinese Character Recognition", *IEEE Trans. Syst., Mans, Cybern*., *22*(2), 1992, 402-410.
[6] Perchant, A., et al. Model-based Scene Recognition using Graph Fuzzy Homomorphism Solved by Genetic Algorithms, *Proc. Graph-Based Representation*, Haindorf, Austria, 1999, 61-70
[7] Medasani, S., Krishnapuram, R., Choi Y.S., Graph Matching by Relaxation of Fuzzy Assignments, *IEEE Trans. on Fuzzy Systems*, *9*(1), 2001, 173-182.
[8] Shearer, K., Bunke, H., Venkatesh, S., Video Indexing and Similarity Retrieval by Largest Common Subgraph Detection using Decision Trees, *Pattern Recognition*, *34*, 2001, 1075-1091.
[9] Sengupta, K., Boyer, K.L., Organizing Large Structural Modelbases, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *17*(4), 1995, 321-332.
[10] Stein, F., Medioni, G., Structural Indexing: Efficient 2-D Object Recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *14*(12), 1992, 1198-1204.
[11] Nishida, H. Shape Retrieval from Image Database through Structural Feature Indexing, *Proc. Vision Interface'99*, Trois-Rivières, Canada, 1999, 328-335.
[12] Bilodeau, G.A., Bergevin, R., Generic Modeling of 3D Objects from Single 2D Images, *Proc. International Conference on Pattern Recognition*, Barcelona, Spain, 2000, 770-773
[13] M. Mokhtari and R. Bergevin, Generic Multi-scale Segmentation and Curve Approximation Method, *Proc. LNCS 2106: Scale-Space and Morphology in Computer Vision, Third Int. Conf.*, Vancouver, Canada, 2001, 227-235
[14] Bilodeau, G.A., Bergevin, R., Part segmentation of objects in real images, *Pattern Recognition, vol. 35*, November 2002, 387-400.
[15] Bilodeau, G.A., Bergevin, R., Modeling of 2D Parts Applied to Database Query, *Proc. Vision interface 2001*, Ottawa, Canada, 2001, 228-235.
[16] Bilodeau, G.A., Bergevin, R., Constructing and matching fuzzy graphs of volumetric primitives hypotheses, *Proc. Vision Interface 2003*, Halifax, Canada, 2003, 278-285.

Bibliographies

Guillaume-Alexandre Bilodeau received the B.Sc.A. Degree in Computer Engineering from Université Laval, Canada, in 1997, and the M.Sc.A. Degree in Electrical Engineering from the same University in 1999. He is currently a Ph.D. candidate at the

Computer Vision and Systems Laboratory of Université Laval. His research interests include object recognition, image retrieval and graph matching. He is a member of Quebec's association of professional engineers (OIQ), and a member of CIPPRS.

Robert Bergevin received the B.Eng. degree in Electrical Engineering and the M.A.Sc. degree in Biomedical Engineering from École Polytechnique in Montréal, in 1982 and 1984, respectively, and the Ph.D. degree in Electrical Engineering from McGill University in 1990. From May 1984 to September 1985, he was a practicing clinical engineer at Sacré-Coeur Hospital in Montréal. From March to August 1990, he worked as a researcher in the field of automated inspection at the Production Engineering Research Laboratory of Hitachi Ltd. in Yokohama, Japan. From October 2002 to April 2003, he was Visiting Scholar at the IRIS Laboratory, University of Southern California, Los Angeles. His research interests are in image analysis and cognitive vision. His contributions have been in generic modelling and recognition of objects in static images and people in image sequences. He is also actively interested in research methodology, specifically in explicating the respective roles of formalism and subjectivity. Since November 1990, he has been with the Computer Vision and Systems Laboratory at Laval University, Quebec City, where he is currently a Professor in the Department of Electrical and Computer Engineering and the Chairman of the undergraduate computer engineering program. Dr Bergevin is a member of the Province of Quebec's Association of Professional Engineers (OIQ), the IEEE Computer Society, and CIPPRS. He is Associate Editor for the Pattern Recognition journal and Area Editor for the Computer Vision and Image Understanding journal.