Fuzzy Conceptual Graphs for Matching Images of Natural Scenes

Philippe Mulhem†, Wee Kheng Leow‡, and Yoong Keok Lee‡
† IPAL-CNRS, National University of Singapore
‡ School of Computing, National University of Singapore
3 Science Drive 2, Singapore 117543
mulhem, leowwk, leeyoong@comp.nus.edu.sg *

Abstract

Conceptual graphs are very useful for representing structured knowledge. However, existing formulations of fuzzy conceptual graphs are not suitable for matching images of natural scenes. This paper presents a new variation of fuzzy conceptual graphs that is more suited to image matching. This variant differentiates between a model graph that describes a known scene and an image graph which describes an input image. A new measurement is defined to measure how well a model graph matches an image graph. A fuzzy graph matching algorithm is developed based on error-tolerant subgraph isomorphism. Test results show that the matching algorithm gives very good results for matching images to predefined scene models

1 Introduction

Conceptual graphs [Sowa, 1984; 1990; 1993; 2000] are expressive for structured knowledge representation and they can be converted to first-order logic expressions. Hence, they are used with theoretical logic-based approaches in applications like hypertext modeling [Kherbeik and Chiaramella, 1995] and image representation and retrieval [Mechkour, 1995; Ounis and Pasça, 1998]. However, in these applications, uncertainty is not represented in the conceptual graphs.

To incorporate uncertainty, Morton [Morton, 1987] extended Sowa's conceptual graph theory to fuzzy conceptual graphs, which include fuzzy referents, fuzzy operations, and fuzzy open and closed worlds. Wuwongse and colleagues [Wuwongse and Manzano, 1993; Wuwongse and Tru, 1996] extended Morton's fuzzy conceptual graphs to incorporate fuzzy conceptual relations. In addition, Maher [Maher, 1991] proposed a similarity measurement for matching simple fuzzy conceptual graphs.

Existing formulations of conceptual graphs are important developments of the theory of fuzzy conceptual graphs. However, they are not suitable for matching uncertain information in images with model scenes. For example, in existing formulations, a concept has only one type. In practice, however, it is very difficult to accurately determine the concept types of the regions in an image. In [Petrakis and Faloutsos, 1997], Attributed Relational Graphs are used for image content representation, and [Medasani and Krishnapuram, 1999] proposed fuzzy matching of these graphs. But, both methods do not handle hierarchies of concept types and of relations.

This paper presents a new variation of fuzzy conceptual graphs that is more suited to matching images of natural scenes. This variant has several distinct characteristics that distinguish it from existing formulations:

- It associates a concept with multiple concept types so as to handle the uncertainty of classification.
- It introduces a new component called *relation attribute* for capturing concepts such as the ratio between the sizes of two image regions. Separating relation attributes from relations allows two or more correlated relations to share the same attribute explicitly. Relationships between attributes can be represented as well. For example, we can explicitly represent the fact that the ratio between the sizes of two objects is larger than the size ratio of two other objects.
- It differentiates between *model graphs* and *image graphs*. A model graph describes a model of a scene whereas an image graph describes the components and their relationships in an image. As will be described in more details in Section 2, these two types of fuzzy graphs have different characteristics.
- It defines a similarity measurement between a model graph and an image graph in terms of graph projection. In addition, an algorithm is implemented to compute the similarity based on subgraph isomorphism.

2 Fuzzy Conceptual Graphs

This section first introduces and defines the basic elements of a fuzzy conceptual graph. Next, the degree of match between conceptual graphs is defined and, finally, the graph matching algorithm is described.

2.1 Basic Elements

A *fuzzy conceptual graph* $G(C, R, A)$ is a directed graph formed from three kinds of components: *fuzzy concepts*, *fuzzy relations*, and *fuzzy relation attributes*. The set C is the set of fuzzy concepts, R the set of fuzzy relations, and A the set of fuzzy relation attributes.

*This research is supported by NUS ARF R-252-000-049-112 and joint NSTB and NUS ARF RP3999903.

A *fuzzy concept* c is a 3-tuple $[T, e, f]$ that consists of a set T of concept types t_i, a referent e, and a fuzzy membership function $f(t_i)$, $0 \leq f(t_i) \leq 1$ for each t_i. In [Wuwongse and Manzano, 1993; Wuwongse and Tru, 1996], $f(t_i)$ is called the *measure* assigned to the type t_i. In practice, $f(t_i)$ can be computed as the confidence of classifying a region as type t_i. A *crisp concept* c can be regarded as a special kind of fuzzy concept whose set T consists of only one concept type t with $f(t) = 1$. For notational convenience, we denote a crisp concept as a 3-tuple of the form $[t, e, 1]$.

A *fuzzy relation* r is a 2-tuple (t, v) that consists of a crisp relation type t and a fuzziness value v which represents the probability of occurrence of the relation r in the images. A *crisp relation* is a special kind of fuzzy relation of the form $(t, 1)$, i.e., $v = 1$ which means that the relation is not fuzzy.

A *fuzzy relation attribute* a is a 3-tuple $[t, e, f]$ that consists of a crisp relation attribute type t, a referent e, and a fuzzy membership function $f(e)$ such that $0 \leq f(e) \leq 1$. A *crisp relation attribute* is a special kind of fuzzy relation attribute of the form $[t, e, 1]$, where $f(e) = 1$.

The *denotation* $\mathcal{D}(t)$ of type t (concept, relation, or relation attribute type) is the set of all possible individual referents of t. A type t' is a *specialization* or *subtype* of type t, denoted $t' \leq t$, if $\mathcal{D}(t') \subseteq \mathcal{D}(t)$ [Sowa, 1984]. The reciprocity holds also. Note that the specialization relationship is reflexive and transitive but not symmetric. The set of concept types together with the specialization relationship forms a finite *lattice of concept types*. Relation attributes are special kinds of concepts. So, the set of relation attribute types form a sub-lattice in the lattice of concept types. In the same way, the set of relation types forms a finite lattice of relation types.

A *model graph* represents a model of a known scene. In a model, the concept types are known. For example, a mountain-and-lake scene contains sky, mountain, water, tree, etc. (Fig. 1) Therefore, the concepts in a model graph are crisp concepts. However, the relations between the concepts may vary from one image to the next. Moreover, to capture fuzzy linguistics such as "much smaller" and "slightly smaller", the relations and relation attributes have to be fuzzy. For example, the relation $(\text{comp}, *, 0.9)$ (represented as $(\text{comp} \,|\, 0.9)$ in Fig. 1) from [tree] to [image] indicates that trees appear in 90% of the images of mountain-and-lake scene. The relation $(\text{smaller}, *, 0.7)$ (represented as $(\text{smaller} \,|\, 0.7)$ in Fig. 1), with attribute $[\text{ratio}, *, \text{f0.5}]$ (represented as $[\text{ratio} \,|\, \text{f0.5}]$ in Fig. 1), from [tree] to [mountain] indicates that in 70% of the images of mountain-and-lake scene, the area occupied by trees is 50% that of the mountains. The symbol f0.5 denotes a fuzzy membership function that peaks at 0.5. If the ratio v in an input image is not exactly 0.5, then f5.0(v) will be smaller than the maximum value of 1.0.

Several relations can connect the same two concepts in a model graph. Among these relations, those that have the same relation type are regarded as forming a disjunctive group (i.e., OR) of relations (e.g., the "smaller" relation between tree and mountain in Fig. 1). The various disjunctive groups then form a conjunction (i.e., AND) of relations.

An *image graph* represents the structure of a particular input image. Due to the inherent uncertainty in the algorithms that classify the image regions, the concepts in an image

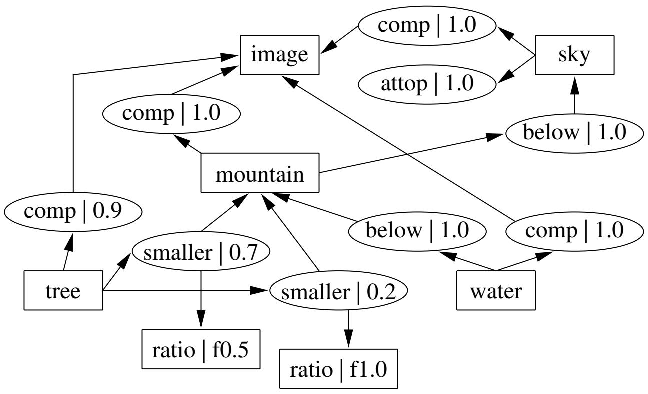

Figure 1: A (partial) model graph representing a mountain-and-lake scene. Relations are represented as ellipses. Concepts and relation attributes are represented as rectangles. Due to space limitation, only some of the concepts, relations, and relation attributes are shown.

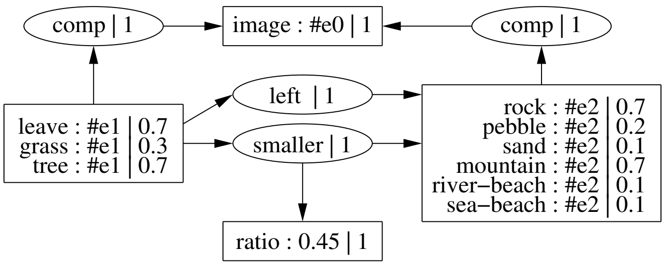

Figure 2: An example (partial) image graph of an input image. Relations are represented as ellipses. Concepts and relation attributes are represented as rectangles.

graph are fuzzy. However, the relations between the regions in an image can be computed exactly. Therefore, the relations and their attributes are not fuzzy. For example, in Fig. 2, due to the ambiguity between leaves texture and grass texture, concept #e1 has a *confidence level* of 0.7 of being a leave texture or a tree texture, and a lower confidence level of 0.3 of being a grass texture. The rectangular box for #e1 represents the fuzzy concept $[T, \#\text{e1}, f]$ where $T = \{\text{leave, grass, tree}\}$ and f(leave) = 0.7, f(grass) = 0.3, and f(tree) = 0.7. The relation $(\text{comp} \,|\, 1)$ from concept #e1 to #e0 is crisp because the region #e1 is a component of the image #e0. Similarly, the relation $(\text{smaller} \,|\, 1)$ and the relation attribute $[\text{ratio} : 0.45 \,|\, 1]$ are crisp because the ratio of 0.45 is measured according to the areas of the regions in the image. In summary, a model graph contains crisp concepts, fuzzy relations, and fuzzy relation attributes, while an image graph contains fuzzy concepts, crisp relations, and crisp relation attributes (Table 1).

2.2 Fuzzy Graph Matching

The degree of match between a model graph and an image graph is defined in terms of *graph projection* [Sowa, 1984] and the degrees of match between their concepts, relations, and relation attributes. The degree of match M between a model concept $c = [t, *, 1]$ and an image concept

Table 1: Summary of the elements of fuzzy conceptual graphs for scene models and input images.

	element	notation	notation in figure
model graph	crisp concept	$[t, *, 1]$	$[t]$
	fuzzy relation	(t, v)	$(t \mid v)$
	fuzzy relation attribute	$[t, *, f]$	$[t \mid f]$
image graph	fuzzy concept	$[T, e, f]$	$[t_i : e \mid f(t_i)]$
	crisp relation	$(t, 1)$	$(t \mid 1)$
	crisp relation attribute	$[t, e, 1]$	$[t : e \mid 1]$

$c' = [T, e, f]$ is defined as

$$M(c, c') = \begin{cases} \max_{t_i \leq t} f(t_i) & \text{if } \exists\, t_i \in T \text{ such that } t_i \leq t \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

The degree of match M between a model relation $r = (t, v)$ and an image relation $r' = (t', 1)$ is defined as

$$M(r, r') = \begin{cases} v & \text{if } t' \leq t \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

The degree of match M between a model relation attribute $a = [t, *, f]$ and an image relation attribute $a' = [t', e', 1]$ is defined as

$$M(a, a') = \begin{cases} f(e') & \text{if } t' \leq t \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

A *projection* π from a fuzzy graph $G(C, R, A)$ to another fuzzy graph $G'(C', R', A')$ is a mapping such that

- for each concept c in C, $\pi(c)$ is a concept in C' and $M(c, \pi(c)) > 0$,
- for each relation r in R, $\pi(r)$ is a relation in R' and $M(r, \pi(r)) > 0$,
- for each relation attribute a in A, $\pi(a)$ is a relation attribute in A' and $M(a, \pi(a)) > 0$,
- for any two concepts or relation attributes s and t related by a relation r in G, $\pi(s)$ and $\pi(t)$ are related by $\pi(r)$ in G'.

In general, the projection $\pi(G)$ may not exist but the projection $\pi(g)$ from a subgraph g of G may exist. Therefore, the matching problem becomes one of finding the best matching subgraph g of G that maximizes $M_G(g, \pi(g))$.

Given a projection π from a subgraph $g(C, R, A)$ of model graph G to an image graph G', the degree of match M_G between g and $\pi(g)$ is defined as

$$M_G(g, \pi(g)) = \frac{1}{|G|} \left[\alpha_c \sum_{c_i \in C} M(c_i, \pi(c_i)) + \alpha_r \sum_{r_i \in R} M(r_i, \pi(r_i)) + \alpha_a \sum_{a_i \in A} M(a_i, \pi(a_i)) \right] \quad (4)$$

where $|G|$ is the sum of the number of concepts, the number of disjunctive groups of relations, and the sum of the maximum number of relation attributes in each disjunctive group of relation in the model graph G. The second summation in Eq. 4 sums over the conjunctive groups of relations. Note that if $g = G$ and $\alpha_c = \alpha_r = \alpha_a = 1$, then $M_G(g, \pi(g)) = M_G(G, \pi(G)) = |G|/|G| = 1$. That is, a perfect match yields a matching value of 1. We set $\alpha_c = \alpha_r = \alpha_a = 1$ in our test.

2.3 Graph Matching Algorithm

To facilitate the matching process, a graph is first decomposed into a set of *arches*, each consisting of a source concept s, a target concept t, a relation r from s to t, and zero or more relation attributes a_i. Thus, arches are actually subgraphs of the model graph or image graph. The match value between a model arch h and an image arch h' is simply the normalized match values of the concepts, relation, and relation attributes of the arches (i.e., Eq. 4).

Our fuzzy graph matching algorithm is a variant of Messmer and Bunke's error-tolerant subgraph isomorphism algorithm [Messmer and Bunke, 1998]. The main differences between these algorithms include the following:

- In [Messmer and Bunke, 1998], the smallest unit of operation is a vertex. In our algorithm, the smallest unit is an arch.
- For error-tolerant graph matching, our algorithm needs to consider only deletion of graph vertices. It does not require addition and replacement of vertices and edges, which are supported in [Messmer and Bunke, 1998].
- Two queues are maintained, one for normal search, and the other one for backtracking to the first-level subgraphs (just below the root). This method improves the quality of the solution found by the algorithm.

The fuzzy graph matching algorithm is as follows:

```
Convert model graph G to arch set H, image graph G' to
arch set H'.
For each pair (g, g') ∈ H×H'
  Compute M_G(g, g').
  If M_G(g, g') > threshold, then insert (g, g') into 1st-level
  queue Q_1 in decreasing order of M_G(g, g').
Repeat n times
  Empty normal search queue Q_n.
  Remove first pair (g, g') from Q_1 and insert into Q_n.
  Repeat
    Remove first pair (g, g') from Q_n.
    If all arches in H have been considered for connecting
    to g, then break out of the inner repeat loop.
    For each pair (h, h') ∈ H×H' such that M_G(h, h') >
    0, h can connect to g, and h' can connect to g'
      Connect h to g, h' to g'.
      Compute M_G(g, g').
      If M_G(g, g') > threshold, then insert (g, g') into Q_n
      in decreasing order of M_G(g, g').
  Save (g, g') and the match value as a candidate solution.
Return the candidate solution with the highest match value.
```

The basic idea is to start off with a single arch in the model graph that best matches an arch in the image graph. This pair of arches forms the initial matching model and image subgraphs. Subsequently, other model arches that have matching image arches are added to the matching subgraphs. For model arches that form a disjunctive group, only the best matching model arch is selected. This matching process repeats until all the arches in the model graph have been considered for addition to the matching subgraphs. Then, the matching model and image subgraphs are saved as a candidate solution.

Due to possible ambiguity of the concepts, initial erroneous matches that happen to have large match values may lead the search algorithm down the wrong search paths. So, another queue Q_1 is used to keep the first-level subgraphs. Once a candidate solution is found and saved, the normal search queue Q_n is emptied and a new search from another first-level subgraph is initiated. This method allows the algorithm to escape from local maxima of the solution space. This procedure is repeated n times, and the best solution among them is returned as the final matching result. In the tests reported in Section 3, n is set to 5.

The threshold in the algorithm is used to prune away subgraphs that have poor matching values. This method reduces the amount of searches that need to be considered and helps to improve the efficiency of the algorithm. In the tests, the threshold is set to 0.2.

During the execution of the algorithm, the value $M_G(g, g')$ can be computed incrementally. Let $p = g + h$ at c, i.e., connecting arch h to model graph g at the concept c results in the new model graph p. Note that $c \in g, h$. Similarly, let $p' = g' + h'$ at c'. Then, $M_G(p, p')$ can be computed incrementally as follows:

$$M_G(p, p') = M_G(g, g') + M_G(h, h') - \frac{1}{|G|} M(c, c') \ . \quad (5)$$

The time complexity of the matching algorithm is $O(n_a N_m^2 N_i (\log N_c + \log N_r))$ where N_m and N_i are the number of model and image arches respectively, n_a is the average number of relation attributes per relation, N_c is the number of concept types in the concept lattice, N_r is the number of relation types in the relation lattice. If each relation has at most one attribute (i.e. $n_a = 1$), then the time complexity reduces to $O(N_m^2 N_i (\log N_c + \log N_r))$.

3 Test Results

In our current implementation, an input image is segmented using a color and texture segmentation algorithm. The color histogram and Gabor texture features of each region are analyzed to classify the region into several possible classes, and the confidence level for each classification is measured. These measurements are used to generate the image graphs of images by a graph generation program. The model graphs of scene models are generated manually because, at present, we are still in the process of developing a learning algorithm for learning model graphs from example images.

Tests were conducted to match four scene models of different complexity (Fig. 3) with 200 images. A sample of the images are shown in Fig. 4. Table 2 summarizes the test results. Images that match the models well have higher match values than those that match the models poorly. Model a is

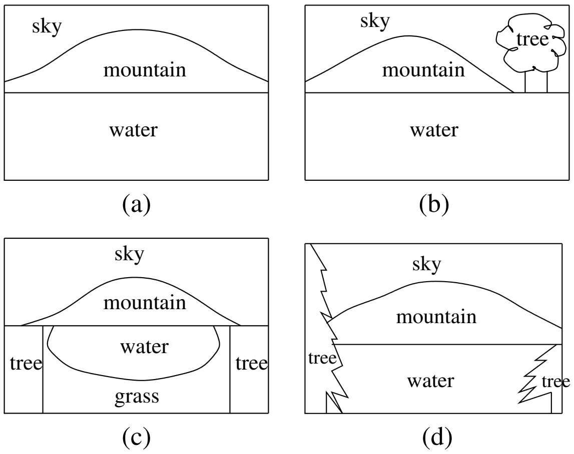

Figure 3: Schematic drawings of four scene models with different complexity.

Table 2: Image matching results. Numbers in brackets are the number of concepts, relations, and relation attributes of the respective conceptual graphs. The other numbers are the match values.

image	model			
	a (4,12,0)	b (5,23,6)	c (7,41,12)	d (6,30,9)
1 (4,16,3)	0.838	0.482	0.191	0.321
2 (6,29,7)	0.814	0.671	0.343	0.471
3 (4,16,3)	0.814	0.444	0.177	0.367
4 (7,31,8)	0.795	0.482	0.290	0.421
5 (4,18,4)	0.767	0.444	0.181	0.387
6 (7,34,10)	0.686	0.666	0.354	0.485
7 (8,37,10)	0.623	0.621	0.432	0.541
8 (7,35,9)	0.471	0.465	0.403	0.895
9 (6,31,9)	0.519	0.371	0.313	0.638
10 (8,40,12)	0.448	0.409	0.337	0.405

the simplest among the four models. Most images can fully match model a successfully and thus have high match values. In particular, the first 5 images best match model a and have the highest match values. Model b differs from model a by having a small tree above the water. Therefore, it best matches images 2, 6, and 7.

Model c is the most complex model which has the most number of components. It is used to show that the algorithm can perform partial matching because none of the images can match model c fully. As a result, the images have smaller match values than those for matching the other models. Nevertheless, the ranking of the images is still consistent: images that match model c better have higher match values. In particular, images 7 and 8 match model c best.

Model d is used to illustrate an extreme case in which a very specific model is specified. In this test, model d is created according to image 8. The test results show that image 8 indeed matches model d very well. Image 9 differs a bit

(1)

(2)

(3)

(4)

(5)

(6)

(7)

(8)

Figure 4: Sample images used in the tests.

(9)

(10)

Figure 4: (Cont.) Sample images used in the tests.

from model d in that image 9 has tall trees on both sides of the image whereas model d has tall trees on only one side. Thus, image 9 has a lower match value than image 8. All other images cannot match model d very well and have very small match values.

4 Conclusion

This paper presented a new variant of fuzzy conceptual graphs that is more appropriate than existing formulations for representing and matching input images and model scenes. A new component called relation attribute is introduced into fuzzy conceptual graphs. Moreover, model graphs that describe known scenes are distinguished from image graphs that represent input images. A similarity measure is defined to assess the degree of match between model and image graphs. A graph matching algorithm is developed based on error-tolerant subgraph isomorphism algorithm. Experimental tests show that the matching algorithm gives very good results for matching images to scene models.

The fuzzy graphs can be input and modified manually using a graphical user interface. The image graphs of the images that match a model can be displayed by an image retrieval system for explaining how well the image matches the model. We are currently developing a method of learning model graphs given example images of the scenes.

References

[Kherbeik and Chiaramella, 1995] A. Kherbeik and Y. Chiaramella. Integrating hypermedia and information retrieval with conceptual graphs. In *Proc. Int. Conf. on Hypermedia, Information Retrieval and Multimedia*, pages 47–60, 1995.

[Maher, 1991] P. E. Maher. Conceptual graphs - a framework for uncertainty management. In *Proc. 10th Annual NAFIPS Conference*, pages 106–110, 1991.

[Mechkour, 1995] M. Mechkour. EMIR2: An extended model for image representation and retrieval. In *Proc. Int. Conf. on Databases and Expert Systems Applications (DEXA '95)*, pages 395–404, 1995.

[Medasani and Krishnapuram, 1999] S. Medasani and R. Krishnapuram. A fuzzy approach to content-based image retrieval. In *Proceedings of the IEEE Conference on Fuzzy Systems*, pages 1251–1257, Seoul, 1999.

[Messmer and Bunke, 1998] B. T. Messmer and H. Bunke. A new algorithm for error-tolerant subgraph isomorphism detection. *IEEE Trans. PAMI*, 20(5):493–504, 1998.

[Morton, 1987] S. Morton. *Conceptual Graphs and Fuzziness in Artificial Intelligence*. PhD thesis, University of Bristol, 1987.

[Ounis and Paşca, 1998] I. Ounis and M. Paşca. RELIEF: Combinig evidences and rapidity into a signle system. In *Proc. ACM SIGIR '98*, pages 266–274, 1998.

[Petrakis and Faloutsos, 1997] G. M. Petrakis and C. Faloutsos. Similarity searching in large image databases. *IEEE Transactions on Knowledge and Data Engineering*, 9(3):435–447, 1997.

[Sowa, 1984] J. F. Sowa. *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley, 1984.

[Sowa, 1990] J. F. Sowa. Towards the expressive power of natural language. In *Proc. 5th Annual Workshop on Conceptual Graphs*, 1990.

[Sowa, 1993] J. Sowa. Relating diagrams to logic. In *Proc. Int. Conf. on Conceptual Structures (ICCS '93), (LNAI 699)*, pages 1–35, 1993.

[Sowa, 2000] J. Sowa. *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Brooks/Cole Publisher, 2000.

[Wuwongse and Manzano, 1993] V. Wuwongse and M. Manzano. Fuzzy conceptual graph. In *Proc. Int. Conf. on Conceptual Structures (ICCS '93) (LNAI 699)*, pages 430–449, 1993.

[Wuwongse and Tru, 1996] V. Wuwongse and C. H. Tru. Towards fuzzy conceptual graph programs. In *Proc. Int. Conf. on Conceptual Structures (ICCS '96) (LNAI 1115)*, pages 263–276, 1996.