USING DISCRIMINATION GRAPHS TO REPRESENT VISUAL INTERPRETATIONS THAT ARE HYPOTHETICAL AND AMBIGUOUS

by
Jan A. Mulder

Department of Mathematics, Statistics, and Computing Science
Dalhousie University
Halifax, Nova Scotia
Canada B3H 4H8[1]

Abstract

The use of specialization hierarchies in model-based vision systems may cause problems with uniformity in representation and efficiency. The concept of a discrimination graph is introduced. Such a graph facilitates the representation of hypothetical and ambiguous interpretations in a uniform and efficient way. We describe the implementation of Mapsee-3, a sketch map interpretation program that uses discrimination graphs in combination with a hierarchical constraint propagation algorithm.

1. Introduction

A key issue in Computational Vision is the proper mapping from image features to interpretations. Image features are often highly ambiguous with respect to interpretation. As a result, alternative interpretations for a single feature have to be represented as hypotheses. These hypotheses are instantiations of scene objects and they can be represented by means of an interpretation graph in which each variable (node) represents a hypothesis and the arcs represent constraints between different hypotheses. In a computational vision system the number of hypotheses can be quite large. Hence, the propagation of constraints over the interpretation graph can be a complex and cumbersome operation, because the addition of new hypotheses and invalidation of existing ones requires a continuous restructuring of the graph.

Most model-based vision systems use specialization hierarchies[2] in an attempt to alleviate this problem. These hierarchies can be used to replace sets of elementary interpretations with a similar appearance in the image by a smaller set of more abstract interpretations. These interpretations are not only hypothetical, they are also ambiguous. They reduce the number of hypotheses that the system has to deal with. Unfortunately, specialization hierarchies only offer a partial solution to the interpretation explosion problem. They are natural categorization schemes and many image features allow for interpretations that do not fit into such schemes. For example, a grassy area seen from low altitude could be farm land as well as a golf course and there is no specialization hierarchy that joins both concepts into one.

The use of specialization hierarchies may therefore cause some interpretations to be hypothetical, whereas others are both hypothetical and ambiguous. Hypothetical interpretations are *explicitly* represented in an interpretation graph by means of different variables, whereas ambiguous interpretations are *implicitly* represented by means of a single variable. Such a representation is not uniform.

The efficiency of the system is also affected, in particular, with respect to constraint propagation. Most images allow only one globally consistent interpretation for each image primitive. At the start of the interpretation process, however, each image feature has many possible interpretations, most of which are explicitly represented in the interpretation graph. Once, more global constraints are found, most hypotheses have to be deleted from the interpretation graph.

Both the problems of uniformity and efficiency can be alleviated, if discrimination graphs are used. Such graphs permit the representation of all interpretations, hypothetical or ambiguous, by means of a single variable which is never deleted from the interpretation graph, once it is constructed. The use of a hierarchical constraint propagation algorithm in combination with discrimination graphs further increases the efficiency of the system. We have designed a model-based vision system that uses discrimination graphs in combination with such an algorithm. This system has been implemented as a schema-based sketch map interpretation program.

2. Discrimination graphs

The idea of discrimination graphs is based on the assumption that we can classify image features with respect to a particular characteristic (e.g. shape, texture) the result of which is a finite number of categories. As well, we assume that there is only a finite number of scene objects whose image appearance falls in a particular category. Discrimination graphs are based on a categorization of object classes that belong to a particular image feature category. The source node of the graph is an abstract object class that intensionally represents all the elementary object classes described by a particular image feature category. The leaves of this graph are the elementary object classes. Elementary object classes can belong to more than one image feature category. As a result, discrimination graphs can become tangled hierarchies with multiple source nodes.

Discrimination graphs differ from specialization

[1] The research reported in this paper was carried out in the Laboratory for Computational Vision at the University of British Columbia.

[2] For a review of different uses of specializations hierarchies, see (1)

hierarchies in at least two respects. An abstract class often represents elementary classes that cannot be joined in a natural specialization hierarchy. As well, the tangled structure of the graph means that not all members of a subclass are automatically a member of a superclass as well (i.e. no universal implication). At the source nodes, discrimination graphs represent object classes which are unique with respect to a particular image appearance, but highly ambiguous with respect to interpretation. At the leaves, on the other hand, we find elementary classes which are unique with respect to interpretation, but ambiguous with respect to their physical appearance.

Discrimination graphs permit the construction an interpretation graph in which each image feature is represented by means of a single variable. This variable is an instance of an abstract object class which intensionally represents the whole range of possible (elementary) interpretations for the feature concerned. The elementary interpretations can be represented *explicitly* as a set of labels in the domain of the variable, or, if we use discrimination graphs, *implicitly* by an abstract label. As the interpretation progresses, we can expect an invalidation of some of these labels. The variables, however, only represent information that is true for all interpretations. Thus, invalidation of one label only requires its deletion or replacement. This can be done without changing the structure of the interpretation graph. As well, all interpretations, hypothetical or ambiguous, are now represented as labels in the domain of a *single* variable.

3. Mapsee-3

Mapsee-3, the sequel to Mapsee-2 [2], is a schema-based program for interpreting sketch maps. Its schema-based format has been inherited from Mapsee-2. Each object class is represented as a list of attribute-value pairs. The attributes determine its internal structure and its relations to other object classes. For example, each class has a "components'* and "super-components" attnbute which determines its location in a composition hierarchy of objects. The object's location in a discrimination graph, on the other hand, is determined by its "discriminations" and "generalizations'[1] attributes. An "instances" attribute lists the current instantiations of the object class. Each instance of an object class inherits the attributes of its parent. A special attribute "label" is used to store the current interpretation of the instance.

Mapsee-3 interprets sketch maps such as the one in figure 1. Both line segments and regions have meaning. Line segments can be interpreted in terms of elementary object classes such as roads, rivers, shores, towns, mountains, and bridges, whereas regions can be land or water. The input consists of a set of plotter commands which indicate the exact location of each line segment in the image.

The Mapsee-3 control is subdivided into three stages: segmentation, image-to-scene mapping, and interpretation. A segmentation process results in the creation of sets of connected line segments (called chains) and regions. The chains form the image features that need interpretation. Mapsee-3 has a fixed number of shape categories for describing chains. The image-to-scene process observes the shape of each individual chain and selects a category in which the chain is placed. Different categories are characterized by features such as closure, mountain-shape, bridge-shape, and blobs.

Discrimination graphs form a key feature in the Mapsee-3 design. Each shape category allows for many different elementary interpretations. The discrimination graphs are constructed such that there is a single (abstract) object class for the set of elementary classes allowed by each shape category. Figure 2 shows a simplified example of a Mapsee-3 discrimination graph. This graph would result from the existence of two shape categories. A closed chain depicts a coastline, lakeshore, or road. Any other line segment depicts a road or river. The leaves of the graph are elementary object classes (e.g. *lakeshore, road),* the other nodes represent more abstract classes, some of which are unnatural (e.g. *road/shore).* All descendents of *road /shore* can have the same appearance in the image. The nodes in the graph with more than one parent can have different appearances in the image. A road, for example, may or may not be depicted by a closed line segment. Thus, not all roads are road/shores, some may be road/rivers instead.

The image-to-scene process instantiates the appropriate scene object for each chain. Only one instance is created for every chain. A closed chain, for instance, gets represented by an instance of the class *road/shore.* The interpretation of this chain is represented by the "label" attribute of the instance. This label is also *road /shore,* which indicates that all of *road/shore's* successors in the discrimination graph are valid interpretations. The image-to-scene process also creates the beginning of the interpretation graph. The instances are the variables, their label the domain. Finding the constraints between instances, however, is the responsibility of the interpretation process.

This paper does not address the question of the construction of the interpretation graph. For such a discussion the reader is refered to [3]. We therefore only discuss the component of the interpretation process that is concerned with constraint propagation. In Mapsee-3 this process is called discrimination, and it uses the discrimination graphs.

Discrimination is an implementation of a network consistency algorithm called hierarchical arc consistency (h.a.c). This algorithm is a derivative of arc consistency as described in [4]. It maintains consistency between labels in the domain of adjacent variables in a constraint graph. Two adjacent variables are considered consistent if all labels in the domain of one variable are consistent with at least one label in the domain of the other. If this is not the case then the inconsistent label is replaced. In case of replacement the test is repeated for all variables adjacent to the one in the domain of which the replacement took place. H.a.c. assumes a hierarchical organization of the domain of each variable as is the case in discrimination graphs. If a label is inconsistent then h.a.c. will recursively replace it by one of its descendents in the hierarchy until a consistent descendent is found.

As an example, the *road/short* label of an instance becomes inconsistent if both regions surrounding the closed chain are constrained to be *land. Road/shore* is valid only as long as all of its descendents (see figure 2) are valid. The new constraint invalidates the shore interpretation. H.a.c. replaces this label by using the principle of least commitment, introduced to Computational Vision by Man and Nishihara [5]. First, *road/shore* is replaced by *road* and *shore*. Next, the consistency of each of these labels is tested. Consistent labels are kept, inconsistent labels are replaced by their successors. In this example, *shore*, *lakeshore* and *coastline* are all inconsistent. Thus *road* is the only label that remains. A more detailed discussion of h.a.c. is provided in [3].

Mapsee-3 solves the problem of uniformity and efficiency as described in the introduction. Competing interpretations are all represented as labels in the domain of a *single* variable. Efficiency is achieved in several ways. First, invalidation of a particular label does not result in a structural change in the interpretation graph. Second, the number of labels in the domain of each variable can be kept very small, thanks to the use of discrimination graphs and the principle of least commitment. Specialization hierarchies do not offer this capability to such an extent. Third, h.a.c. does not maintain an explicit administration of the compatibility between the labels of adjacent variables. Only if a label change in a neighboring variable took place, will a variable test its labels. It will stop this test as soon as a compatible label is found in the domain of a neighbor. In interpretation graphs with competing interpretations represented by different variables, the compatibility between the interpretation of adjacent variables is represented explicitly. As mentioned before, most interpretations tend to be eliminated during the course of the interpretation process.

4. Conclusions

We have described the concept of discrimination graphs as a uniform and efficient way for representing interpretations that are hypothetical and ambiguous. We have also described a hierarchical constraint propagation algorithm that uses discrimination graphs in an efficient manner to propagate constraints in an interpretation graph. Discrimination graphs and hierarchical constraint propagation have been implemented in Mapsec-3, a schema-based vision system that interprets sketch maps.

References

[1] R.J. Brachman, "What IS-A is and Isn't", *Proc. of the 4th Nat. Conf. of the Can. Soc. for Comp. Studies of Intelligence,* pp. 212-221, Saskatoon, Canada, 1982.

[2] W.S. Havens and A.K. Mackworth, "Representing Knowledge of the Visual World**, *IEEE Computer,* vol. 16, no 10, pp. 90-98, 1983.

[3] J.A. Mulder, "Representing Ambiguity and Hypotheticality in Visual Knowledge**, Forthcoming Ph.D. Thesis, Department of Computer Science, University of British Columbia.

[4] A.K. Mackworth, "Consistency in Networks of Relations", *Artificial Intelligence*, vol. 8, no 1, pp. 99-118, 1977.

[5] D. Marr and H.K. Nishihara, "Representation of the Spatial Organization of Three Dimensional Shapes", Report 377, A.I. Lab, M.I.T., 1976.

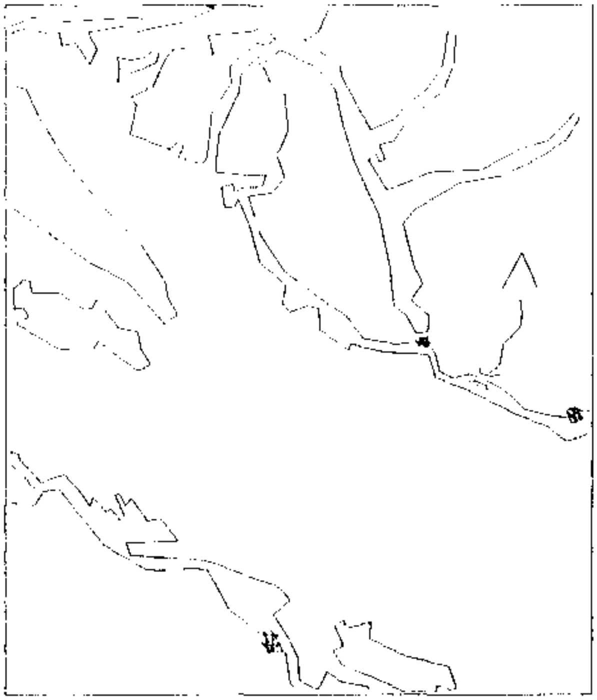

Figure 1: Sketch of Georgia Strait (B.C.)

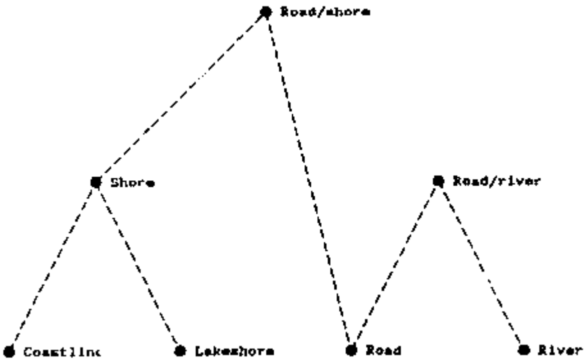

Figure 2: A Discrimination Graph