CVGIP: IMAGE UNDERSTANDING
Vol. 55, No. 2, March, pp. 212–218, 1992

WORKSHOP PANEL REPORT

Why Aspect Graphs Are Not (Yet) Practical for Computer Vision

Participants

OLIVIER FAUGERAS
INRIA, Sophia-Antipolis

JOE MUNDY
General Electric Corporate Research and Development

NARENDRA AHUJA
University of Illinois

CHARLES DYER
University of Wisconsin

ALEX PENTLAND
Massachusetts Institute of Technology

RAMESH JAIN
University of Michigan

KATSUSHI IKEUCHI
Carnegie Mellon University

Organizer: KEVIN BOWYER

Department of Computer Science and Electrical Engineering, University of South Florida, 4202 Fowler Avenue, Tampa, Florida 33620

Received November 20, 1991; accepted November 20, 1991

A panel discussion on the theme of this report was organized for the *1991 IEEE Workshop on Directions in Automated CAD-Based Vision*. This report contains the revised comments of the panel discussion participants.

1. INTRODUCTION

The *aspect graph* of an object is a graph structure in which

- each node represents a *general view* of the object as seen from some maximal, connected cell of viewpoint space,
- each arc represents an *accidental view* (or *visual event*) which occurs on the boundary between two cells of general viewpoint,
- there is a node for each possible general view of the object, and
- there is an arc for each possible visual event.

The aspect graph representation is often considered to have great potential for computer vision. In the last few years, algorithms have been developed to automatically compute the aspect graphs of polyhedra, general curved objects, and even objects with articulated connections between parts. However, much of the work in this area has a somewhat theoretical flavor and it is not clear that the aspect graph representation, at least as it is currently conceived, will find practical application.

For this panel discussion, several distinguished researchers have agreed to provide a critique of the aspect graph approach, and several other distinguished researchers who are working in the aspect graph area have agreed to respond to the critiques. It is hoped that this exchange will help to identify the essential issues involved in this area of research.

2. CRITIQUES

Olivier Faugeras, INRIA

I believe there are two important reasons that aspect graphs have not been heavily used so far in the computer vision community. The first reason is mostly mathematical but has incidences on the implementation. The second reason is mostly algorithmic but has a strong relationship to a yet unsolved problem in vision for which I believe some completely new ideas will be necessary. In what follows I consider only rigid objects.

As far as the first point goes, we know that the cells in viewpoint space that define the aspect graph of an object are separated by surfaces (or curves, if we stay on the Gaussian sphere) which signal a change in the topology of the silhouette of the object. Those visual events can in principle be computed from the equations defining the object, but mathematicians are presently in the blue for telling us at which scale on the silhouette the changes will happen. Some of them may happen at a large scale and some at a "microscopic" scale, but predicting the scale

1049-9660/92 $3.00

which would allow us not to compute everything and to group cells together seems to be a very hard mathematical question which is as yet unanswered. To quote a mathematician friend of mine, "mathematics has nothing to say about scale."

This has very serious implications for computer vision, since we know that the number of cells in the aspect graph of a real object can easily reach several million and many of those may be irrelevant at the scale at which the observations are made. But because of the previous mathematical lack of understanding of the scale at which the visual events occur, we must compute all of them before we can attempt to group them. Therefore, there does not seem to be any practical means in view to reduce the complexity of the computation of aspect graphs.

The second reason that I believe aspect graphs are difficult to use is the following. Suppose we have computed the aspect graph of an object and we observe that object from an unknown viewpoint (I assume that the object is isolated and will not deal with the even more difficult problem of occlusion). We extract, say the silhouette, and are faced with the task of matching it to one of the silhouettes stored in the aspect graph modeling that object. If the number of cells is high, let use say a few thousands, then this is a very difficult indexing problem: we probably do not want to try to match our measured silhouette to all those stored in the model, even if we have a highly parallel processor. Good solutions to this problem are unknown to me. Of course, it becomes even more difficult if we do not assume that we know which object we are looking at—we then also have to index in our data base of models.

I believe that this indexing problem is one of the key issues that the computer vision community has to face in the next few years and for which genuinely new ideas have to be found because none of the available ones will work.

To summarize, I believe there are two main reasons why aspect graphs are so difficult to use and, strangely enough, those two reasons correspond to two very big unsolved issues in computer vision. The first issue is our poor understanding of what scale means. This lack of understanding makes it extremely costly to compute the aspect graph of real objects. The second issue is our just as poor understanding of what model indexing is all about.

Joe Mundy, General Electric C, R, & D.

Aspect graphs have some appeal in that they provide an exhaustive catalog of the distinct views of a 3D object as projected onto an image plane. The great weakness of the approach is the use of feature topology to define the boundary between aspects. A view is defined in terms of the usual concepts of junction (or vertex), edge, and face topology. Whenever this structure changes as with respect to viewpoint, a new aspect is generated.

On the one hand, these topological structures are well defined and are well developed within the mathematical literature. On the other hand, there is little reason to believe that these topological relations can be reliably retrieved from an image, even without considering occlusion. For example, it is well known that the broadly used Canny edge detector is unable to recover junctions. In our own experience, it is also difficult to recover internal boundary edges between surface faces due to low contrast and self-shadows. One can rely only on recovering fragments of occluding boundaries as features to index into 3D descriptions.

In addition it is important to be able to determine the pose of the 3D model from image features so that the model can be projected onto the image to guide additional feature extraction. This requirement introduces the concept of an "effective viewpoint," which we have used in our model-based vision system. Briefly, an effective set of features are those features which can be used to determine model pose with good accuracy in the context of feature positional uncertainties caused by segmentation. Since pose recovery accuracy depends on viewpoint for a given set of features, additional features are required to cover the full range of viewpoints. The boundaries of effective performance for feature groups produce a kind of "aspect" graph but in terms not of topology, but of pose recovery accuracy. The boundaries between these "aspects" are not sharp and we represent the pose accuracy for a feature set as a variance distributed over the viewsphere. We select feature groups from a model which maximize performance for each viewpoint.

Current research on aspect graphs does not typically take into account these feature recovery and pose computation problems. (There has been some work on sensor modeling by Ikeuchi and Kanade in this context.) A much better direction for research on aspect graphs would be to form a set of "recoverable aspects," which quantify the image segmentation problem and potential self illumination effects. Such notions can be generalized to define an aspect graph which produces feature sets that are likely to be recovered from segmentation and are also effective in indexing model class and are accurate in pose recovery. These properties are difficult to quantify but this evolution of the aspect graph will be needed to make the idea of precomputed views an attractive approach for object recognition.

Narendra Ahuja, University of Illinois

The aspect graph is an intuitive, simple, and appealing method of representing an object, since it enumerates all appearances, or aspects, of the object, along with the viewpoints from which these aspects can be seen. In a

sense, this seems to be the minimal information about the object that any complete representation must be able to capture (for example, to match an image of an object to the object.) The different representations in vogue differ in (i) how the aspect is defined, and (ii) whether the different aspects are captured implicitly or explicity. According to the answers chosen to these questions, different representations have different mixes of advantages and disadvantages, and different domains of applicability. This implies that the question of whether a method of object representation (for that matter, any representation) is good or not can be answered only with respect to its desired usage.

In what has come to be known as the aspect graph representation, (i) an aspect is defined as the topological structure of an image of the geometric contours of the object, specifically, orientation and depth boundaries; and (ii) all different aspects and the associated sets of viewpoints are explicitly specified. The most common application of aspect graphs is object recognition (aspect graph is a misfit if the objective is to compute, say, the size of an object.) The discussion below is for this definition and application of aspect graphs; the shortcomings listed are direct consequences of the specific mix of object characteristics made implicit and explicit in the definition.

Shortcomings:

1. The implicit presumption that the geometric contour information is sufficient for recognition is questionable. For example, the gray level and relative size information may be necessary, or even crucial.

2. Generating an aspect graph is a very complex computation as a function of object complexity.

3. Aspect graphs are very large.

4. A consequence of (3) above is that the cost of using aspect graphs is high. Significant additions must be made to the representation to efficiently retrieve object information (e.g., methods for indexing the graph to access aspects of specific types). Without this, a huge search effort is necessary to match a view to an object.

5. Extraction of line drawings corresponding to geometric contours from noisy images is highly unreliable. Since lines are the source of all information in aspect graphs, any errors therein may lead to serious errors in results. Line drawing extraction may be made robust by carrying out extensive three-dimensional reconstruction, but this would mean solving the three-dimensional reconstruction problem first (at least partly), and thus collecting much more object information in the process than called for by the aspect graph representation. But this changes the problem fundamentally: with the three-dimensional structure already extracted, recognition may be easier, e.g., by using a more complete three-dimensional, surface-based representation rather than the aspect graph (i.e., by using a richer definition of the aspect).

6. Related to (5) above is another issue. Objects contain geometric as well as brightness contours, and confusing one type with the other would be catastrophic since each line means a lot. How does one distinguish between the two? It appears crucial that this distinction be made before any matching with aspect graphs begins. But it may not be easy to do so without extensive analysis, such as mentioned in (5), which would again make the use of aspect graphs superfluous in the presence of the extracted additional information.

3. RESPONSES TO THE CRITIQUES

Charles Dyer, University of Wisconsin

I see the major criticisms of the aspect graph to be of two general types: one is based on how an aspect graph is defined, and the second is based on how it will be used.

The first issue deals with questions such as what kinds of models and features define the visual events that identify a change in the aspect of an object. To date, the primary concern of most researchers has been on how a complete set of topologically-distinct views can be enumerated given a particular type of object model and image features generated by that model. The use of edge and vertex features generated by a polyhedral model was an important first step because it led to precise algorithms for computing visual events and aspect graphs. Polyhedral models are also important as approximations of natural, piecewise-smooth 3D shapes. As in computer graphics, the linear features of polyhedra permit faster algorithms than are possible using smooth models and complex numerical techniques. As long as the algorithms that use the representation (e.g., indexing and matching in object recognition) take into account that the model is an approximation of a smooth object, this is an important and practical tradeoff.

One of the negative results of work using polyhedral models was the realization that aspect graphs can be extremely large for complex objects containing many features. Consequently, new models and features have been used to define the aspects of an object. For example, piecewise smooth models (see work by Ponce, Kriegman, Bowyer, and others), models decomposed into parts (see Pentland below), and features generated by only the occluding contour of a model (see the paper by Seales and Dyer in the Workshop Proceedings) have been studied. These approaches are important because they focus on ways of (1) reducing the number of features, leading to a smaller aspect graph, and (2) restricting the types of features to the most relevant and detectable ones. Incorporating scale is another way to achieve

these same ends. The important point is that many types of features and object models can be used within this framework. The fact that researchers continue to change the way aspect graphs are defined only emphasizes the fact that the core ideas lead to multiple interesting realizations.

Even with a more selective set of object features, the size of the aspect graph can still grow very large. This is because aspect is a *global* property defined in terms of all visible features, and a visual event is produced by a change in any one of those features. Another consequence of this emphasis on creating a set of global descriptions (i.e., image structure graphs) of the views of an object, is that information about the visibility and geometry of individual features and groups of features is less accessible.

The aspect graph is but one type of the more general class of viewer-centered 3D object representations that incorporates a complete analysis of the continuous viewpoint space in order to explicitly describe the features that are detectable in an image. Mundy's effective feature sets and Ikeuchi's sets of visible features fit into this more general framework. Another alternative is to create structures that are organized in terms of the appearance of individual features. This approach emphasizes making explicit how interfeature relationships and geometric features in an image change with viewpoint (i.e., pose).

Our work on the "asp" and the "rim appearance representation" does just this—characterizing the range of viewpoints where each feature is visible and the geometry of its appearance over that cell of viewpoint space. For example, the geometry of T-junctions and curvature extrema of the occluding contour of a shape can be explicitly represented. What is novel about this approach is that not only is information organized in terms of individual features instead of the global image structure graph of features, but this organization (1) is usually much smaller than the aspect graph, and (2) explicitly describes how features dynamically change over viewpoint. The second point is especially important if one of the uses of the representation is with a dynamic vision system where spatiotemporal data are being used and therefore constraints are available from the way image features change over time.

The second broad criticism is based on how the aspect graph will be used. Primarily, this concerns the indexing problem for object recognition. Because each node in an aspect graph represents a set of views (or, equivalently, poses), the usual assumption is that recognition will use a brute-force, node-parallel search to find the best matching aspect for a given (segmented) set of image features. This aspect classification procedure can then be followed by a pose calculation step if necessary. This use of aspect graphs completely ignores aspect cell boundaries, cell adjacency information, and intracell geometry of the views. Successful classification therefore requires the unrealistic assumptions that the image has been segmented into the correct global image structure graph of features, and interobject occlusion is largely absent.

To avoid these problems, one can perform indexing and matching based on the appearance of individual features or relations between small groups of features. Selected feature configurations can be described in terms of the viewpoint cells in which they are visible and their changing geometry within a cell. For example, a T-junction, formed by the apparent intersection of two rim contours of a model, is stable over a bounded cell of viewpoint space. The boundaries of this cell can be precomputed in closed form from the model. The orientation and angle sizes of the "T" feature can be described as a function of viewpoint within this cell. A correspondence between an image T-junction and a model T-junction therefore constrains the possible viewpoints of the object to the associated cell of viewpoint space where this match is geometrically consistent. We have tested this approach using the rim appearance representation, implementing a system that finds the best consistent match with a set of image features and its associated cell of viewpoint space (see the Workshop Proceedings). Finally, because this representation framework is dynamic in the sense that features are represented as a function of viewpoint, when the viewer or object is moving, indexing could be based on how appearance is changing, matching model-based "spatioviewpointal" structures with spatiotemporal image data.

In summary, aspect graphs are an important first step in analyzing methods of encoding viewer-centered descriptions of 3D shape. In my view, future directions should deemphasize the issue of cataloging the topologically distinct views of an object. Greater emphasis is needed on determining geometric constraints on viewpoint, and modeling how detectable features change over viewpoint.

Alex Pentland, MIT

There are three central criticisms of the aspect graph approach that are raised, in various forms, by each of Professor Faugeras, Dr. Mundy, and Professor Ahuja. These criticisms are:

1. Aspect graphs can be very large and complex, leading to difficulties in matching, indexing, etc.
2. Matching image contours to aspect graphs can require a good segmentation, which is difficult to achieve.
3. Aspect graphs are ill-defined (especially with respect to scale and curvature) and impoverished (including no grey-level or sensor information).

I agree strongly with these points; however, I also see that there may be general methods of reducing or even avoiding these difficulties. In particular:

A general method for avoiding the complexity of full aspect graphs is to apply the aspect graph approach only to component parts, subassemblies or critical features. In our work (e.g., Dickinson, Pentland, and Rosenfeld, elsewhere in this workshop) we have applied them only to component parts, so that the resulting aspect graphs are extremely simple. The major difficulty of using components is that it forces you to confront the occlusion problem "head on"—which is perhaps not such a bad idea in any case.

The segmentation problem (e.g., edge and face finding) is a general problem that plagues all of machine vision, and so in one sense is not an objection to the aspect graph approach per se. However, in aspect graph applications knowledge about the *types* of aspect graph that can occur can be used to constrain the segmentation, so that segmentation becomes *model based* rather than generic. Thus if there are only a small set of possible aspect graphs, as in our case where only generic object parts are modeled, we have found that the problem of segmentation can become much easier.

Similarly, the criticism that aspect graphs are ill-defined and improverished is a problem common to many representations used in machine vision. I suggest, however, that many of these problems can be minimized by moving from *edge-based* aspect graphs to *face-based* aspect graphs, as was done in our work. An object's faces may be defined as a minimal set of "smooth" surfaces that approximate its 3-D surface to within some tolerance (e.g., a set of low-order polynomials plus boundaries). The approximation of surfaces in this manner is well understood (although some stability problems remain) and allows inference of grey-level appearance, texture foreshortening, etc. It also shifts the low-level processing from edge finding to region extraction, which for range imagery appears to be a more tractable problem.

Ramesh Jain, University of Michigan

The critiques of aspect graphs have raised several interesting issues. Clearly, there are many problems to be solved before we can use aspect graphs in object recognition systems. Let us first consider why aspect graphs are useful and then address specific issues raised above.

Object recognition has two distinct phases: learning or model formation, and recognition using images. The learning phase is off-line and, therefore, it does not matter much if this phase is slow. What is important is that the on-line recognition phase be completed fast.

Most objects to be recognized are three-dimensional. They must be recognized from their two-dimensional views. In addition to the well known problem that the intensity value at a point is the result of several factors, we have to consider the fact that the objects in images appear in "viewer-centered" representation. The models used for recognition of three-dimensional objects in a two-dimensional viewer-centered space will be multiple-view representations. If we try to use three-dimensional object-centered models, the on-line recognition time will be very slow. Thus, aspect graphs are useful only if they help us in solving the standard time-memory trade-off.

Now, let us consider the key issues raised by the critics.

1. Our poor understanding of what scale means, which makes it extremely costly to compute the aspect graph of real objects.

The scale issue is not only related to aspect graphs, but edges, corners, and all other features that we can think of. It starts much before we use a digital image. Did we sample an image at the rate to preserve all information? Researchers are trying to understand this issue in other contexts and aspect graphs will be no exception.

2. Our poor understanding of what model indexing is all about.

I agree with this completely. Indexing is a key issue that we have failed to address adequately. Whether we use aspect graphs or some other representations, if we want an object recognition system to work with a large number of objects, we will have to deal with indexing problem. Aspect graphs are responsible neither for our ignoring the indexing problem, nor for creating this problem.

Before going to other issues, I must point out that Professor Faugeras is aware of the above problems being general problems, he himself says, "I believe there are two main reasons why aspect graphs are so difficult to use and, strangely enough, those two reasons correspond to two very big unsolved issues in computer vision."

3. There is little reason to believe that the topological relations can be reliably retrieved from an image, even without considering occlusion.

Based on the amount of research effort spent on edge detection algorithms, and on their success, there is no reason to believe that we will be able to develop a "universal" edge detector to reliably recover edges in an arbitrary image. Thus, there is no hope of retrieving the aspect graphs from images. In fact, we have to learn to live in this imperfect world of unreliable edges. However, why should we worry about this problem in the context of aspect graph generation? Aspect graphs will be generated off-line either from models or under controlled conditions, even, with human interaction.

4. It is important to be able to determine the pose of the 3D model from image features so that the model can be

projected onto the image to guide additional feature extraction.

Indexing using aspect graphs is precisely going to do this. In fact, Mundy himself says, "A much better direction for research on aspect graphs would be to form a set of "recoverable aspects," which quantify the image segmentation problem and potential self illumination effects. Such notions can be generalized to define an aspect graph which produces feature sets that are likely to be recovered from segmentation and are also effective in indexing model class and are accurate in pose recovery."

5. The implicit presumption that the geometric contour information is sufficient for recognition is questionable. For example, the gray level and relative size information may be necessary, or even crucial.

Aspect graphs are not supposed to be based only on geometric contour information. For recognition, the surface information and other features will be required. I believe, however, that the gray level depends on illumination and it will be a mistake to include such scene dependent information in object models.

6. Generating an aspect graph is a very complex computation as a function of object complexity.

Since aspect graphs are computed off-line, we can afford to spend time on these computations. The complexity of an aspect graph will depend on the complexity of the corresponding object. However, this is true for every operation, including boundary detection.

Aspect graphs are just an intermediate representation for object recognition. The criticism that most work in this area has been theoretical is a valid one, but not a surprising one. Efforts are underway to develop algorithms for the generation of aspect graphs at several laboratories.

Katsushi Ikeuchi, Carnegie–Mellon University

The critiques of aspect graphs have raised several interesting issues. However, these critiques have been based on misunderstandings or narrow interpretations of the concepts of aspects and aspect graphs. The critiques assume that aspects are narrowly defined as topologically equivalent classes of line drawings of an object. This is an incorrect assumption.

The definition of aspects can be based on various visible features. Originally, an aspect was defined as a class of appearance with a common topological structure. However, the concept can be broadened by replacing "common topological structure" with "common set of visible features," and we can then evolve a family of aspects based on the features used. Using this expanded definition, aspects based on line-drawing topology are only one class within an entire family of aspects. We can define aspects based on visible faces, edges, or vertices. Moreover, the family of aspects also has a dimension characterized by sensors: aspects under photometric stereo are different from those under a light-stripe range finder. It is important to investigate the characteristics and structure of the entire family of aspects. I claim that such a broad family of aspects is a useful, practical, and essential tool for object recognition research.

The appearance of a 3D object varies as viewing direction varies. The changes in appearance fall into two classes: changes in aspect, and linear shape change. A change in aspect changes the overall appearance of the object, in terms of visible features. For example, a face may be visible in one aspect, then disappear when the aspect changes. On the other hand, a linear shape change preserves the overall appearance: the collection of features remains the same, but the apparent shapes and relationships may be skewed smoothly.

The goal of object recognition is to determine the presence/absence of an object in an image based on visible features; object localization aims to precisely determine object attitude using visible features. Aspects characterize sets of visible features, and the types of changes that can take place among visible features. It is natural to decompose an object recognition and localization task into two phases: aspect classification (AC), and determination of linear shape change (LC). The AC phase classifies an appearance into an aspect, and then the LC phase performs attitude-determination/existence-verification within an aspect using the visible features of the aspect. The purpose of the AC phase is to identify proper visible features and to simplify and stabilize the LC phase. Thus, we can say that the primary goal of object recognition and localization consists in the LC part.

An LC method usually consists of an evaluation function that measures the match between visible features and model features. For example, the EGI matching function measures the similarity between visible and model EGI's to determine the attitude of an object. Lowe's matching function minimizes the distances between projected model points and image edges to verify the existence of an object.

These evaluation functions are usually continuous and well-behaved within an aspect, provided that the set of visible features correctly matches the hypothesized set of model features. That is, given correct aspect classification, model features can be correctly matched to image features, and the evaluation function will converge to the correct solution. On the other hand, discontinuities exist across aspect boundaries since features appear and disappear. Thus, if an appearance is incorrectly classified into an aspect, then model and image features will be incorrectly matched, and the evaluation function may not converge to the correct solution and may instead converge to an incorrect local minimum.

Generalizing from the above discussion, the purpose of the AC phase is to determine the interval within which a

particular LC evaluation function is continuous and well-behaved. Thus, in the extreme case in which the LC function is continuous in all viewing directions, it is not necessary to have an AC phase at all. For example, when we determine the attitude of an ellipsoid using EGIs, the AC phase is not needed.

The relationship of the AC and LC phases guides us in the selection of aspects from the global aspect family:

• *Aspects should be defined based on the features that will be used in the corresponding LC function.* For example, if an LC evaluation function is based on the largest visible face under photometric stereo, then aspects should be defined based on largest visible faces under photometric stereo. If a LC function is based on the visible edges extracted from a TV camera, then aspects should be based on line drawings.

• *The resolution of aspects should be based on the robustness of the LC function.* If an LC function is robust enough to determine attitude given a few incorrect visible features, then a separate aspect does not need to be defined for every combination of visible features. The appearance/disappearance of minor features can be ignored, and the AC phase simplified.

In summary, aspects are important and convenient tools. However, we have to develop a general theory of aspects based on features detectable by sensors. The theory of aspects should have the capability to accommodate various LC methods and clarify the relationship between the AC and LC phases. The general aspect theory also should make clear the types and characteristics of aspect boundaries. General aspects can be used to solve the object recognition problem in efficient ways.

APPENDIX

Clearly the kind of misunderstanding which led to the critiques occurs due to a bug in the "general purpose vision" paradigm. This paradigm defines a vision module in a general and isolated condition, and develops theories independent of other modules. If this is done properly, it provides rich vision theories. However, it often occurs that along the course of research the effort loses sight of the original goal and instead generates unrealistic solutions from ill-defined assumptions. The bug in the paradigm is the lack of focus on the interaction between modules which allows poor assumptions to be introduced.

This kind of bug can be avoided by introducing the notion of a "task" within which a vision module works. A task specifies the purpose of a vision module, as well as the inputs and outputs, and forces the consideration of interaction between modules.

We are developing vision theories and vision modules within the task-oriented vision framework. We consider a vision system as a whole and develop not only intramodule vision theories but also intermodule vision theories for specific tasks. For example, in the previous discussion, the relationship between the AC and LC evaluation functions corresponds to an intermodule theory, while a method of generating aspect graphs corresponds to an intramodule theory.

Such a task-oriented vision framework determines the choice of vision modules and clarifies the assumptions and goals underlying each vision module. We have to promote this task-oriented vision framework to ensure the healthy development of the computer vision community.