Qualitative Tracking of 3-D Objects using Active Contour Networks*

Sven J. Dickinson
Department of Computer Science
University of Toronto, 6 King's College Rd.
Toronto, Ontario, Canada M5S 1A4
(sven@vis.toronto.edu)

Piotr Jasiobedzki
Department of Computer Science
University of Toronto, 6 King's College Rd.
Toronto, Ontario, Canada M5S 1A4
(piotr@vis.toronto.edu)

Göran Olofsson
Computational Vision and
Active Perception Laboratory
Royal Institute of Technology
S-100 44 Stockholm, Sweden
(gorano@garbo.bion.kth.se)

Henrik I. Christensen
Laboratory of Image Analysis, IES
Aalborg University
DK-9220 Aalborg, Denmark
(hic@vision.auc.dk)

1 Introduction

Research in 3-D object tracking is typified by approaches which attempt to recover the six degrees of freedom of an object in each frame, e.g., [11, 14, 6, 17]. Once a correspondence between image and model features is determined, changes in the positions of image features in successive frames is used to update the pose of the object. Although providing accurate pose of the object at each frame, these techniques require an exact geometric specification of the object. At the other extreme, active contour methods have been used to track objects in the image with little or no knowledge of the 3-D object, e.g., [8, 13, 3, 2]. However, these methods can only track the position of the object in the image and not its orientation.

In this paper, we track changes in the appearance of the object as it moves from one frame to the next. At a symbolic level, an aspect graph clusters all the views of an object into a set of topologically distinct classes in terms of which surfaces of an object are visible from a given viewpoint (Koenderink and van Doorn [9]). Two nodes (or aspects) in the aspect graph are connected by an arc if it is possible to directly move from a viewpoint in which the first aspect is visible to a viewpoint in which the second aspect is visible. Qualitatively, we can envision a tracking strategy which simply tracks an object as it moves from one node to another in the object's aspect graph. Although it does not provide us with accurate pose of the object, it does qualitatively describe the motion of the object *without* the need for a CAD representation of the object.

To track an object as it moves from one view to another requires that we can detect visual events in the image. Using a network of active contours, called an *adaptive adjacency graph* [7], we will track aspects from one frame to the next. Since visual events are characterized by the appearance or disappearance of faces in an aspect, we will use the adaptive adjacency graph to monitor the changes in each aspect's face from one frame to the next. When a face in an aspect is rapidly diminishing, a signal is sent to a monitoring process responsible for keeping track of the current position in the *aspect prediction graph* (APG), a derivative of the aspect graph. In addition to assigning probabilities to each node in a compressed aspect graph, the APG explicitly encodes at which contours of an aspect visual events occur, and maintains a correspondence between faces in adjacent nodes in the graph. When the monitor, or *symbolic tracker*, is informed by the image tracker that some face is disappearing, it decides to which node in the aspect prediction graph the object is moving. The symbolic tracker then sends a set of predictions to the image tracker in terms of structural changes to the network.

*The authors would like to especially thank Gene Amdur and Winky Wai for their assistance in the implementation of the tracking system. Funding for Piotr Jasiobedzki's contribution was provided, in part, by the ARK (Autonomous Robot for a Known environment) Project, which receives its funding from PRECARN Associates Inc., the Department of Industry, Science and Technology Canada, the National Research Council of Canada, Technology Ontario, Ontario Hydro, and Atomic Energy of Canada Limited. Henrik Christensen and Göran Olofsson would like to acknowledge funding support from EP-7108-VAP "Vision as Process".

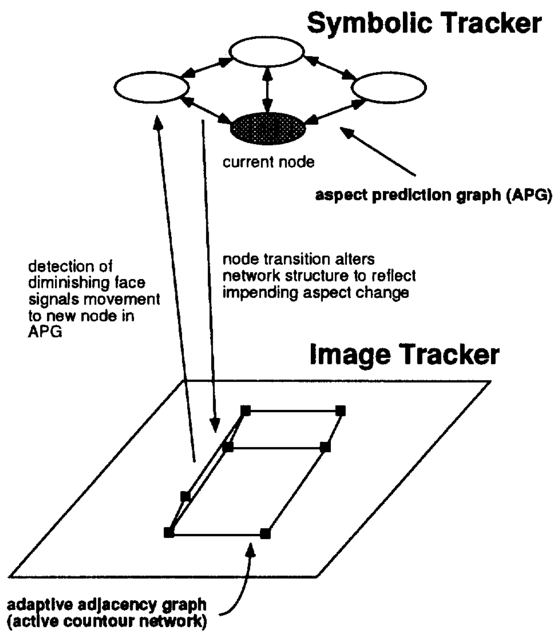
Symbolic Tracker
current node
aspect prediction graph (APG)
detection of diminishing face signals movement to new node in APG
node transition alters network structure to reflect impending aspect change
Image Tracker
adaptive adjacency graph (active countour network)

Figure 1: System Overview

2 Related Work

One approach to tracking a 3-D object in a 2-D image is to begin with a 3-D CAD model specifying the exact geometry of the object, as surveyed in [14]. Of more relevance to our approach are view-based strategies which encode a 3-D object as a set of views; examples include Wallace and Mitchell [15], Liu and Tsai [10], and Siebert and Waxman [12]. Although such approaches do not solve for the exact pose of an object, they do use aspects to capture an object's "qualitative" pose. If even the qualitative pose of the object is unnecessary, we can track only the boundary of the object in the image. Active contours (snakes) proposed by Kass et al. [8] posses the ability to adapt themselves to image features while maintaining their smoothness and rigidity. Different versions and implementations of active contours have been used for tracking moving contours, finding salient contours in images, and stereo matching, e.g., Amini et al. [1], Williams and Shah [16], Curven et al. [3], and Terzopoulos and Szeliski [13].

3 Symbolic Tracker

The symbolic tracker, as shown in Figure 1, tracks movement from one node to another in a representation called the aspect prediction graph [4]. Each of the nodes in this representation, derived from an aspect graph (Koenderink and van Doorn [9]), represents a topologically different viewpoint of the object, while arcs between nodes specify the visual events or changes in image topology between nodes. The role of the symbolic tracker is to:

1. Determine which view or aspect of the object is currently visible (current node).
2. Respond to visual events detected by the image tracker by predicting which node (aspect) will appear next (target node).
3. From the visual event specification defined by the current and target nodes, add or delete structure from the active contour network (predictions).
4. If predicted aspects cannot be verified by the image tracker or visual event predictions cannot be recognized by the symbolic tracker, the symbolic tracker must be able to bootstrap the system to relocate itself in the aspect prediction graph.

In the following sections, we explore the symbolic tracker in more detail, first examining the aspect prediction graph, and then describing its role in monitoring the image tracker.

3.1 The Aspect Prediction Graph

The aspect prediction graph (APG) is derived from two sources: the aspect graph and the aspect hierarchy. Our aspect prediction graph is a more efficient version of the aspect graph in which topologically equivalent nodes are grouped regardless of whether their faces map to different surfaces on the object. Next, the aspect prediction graph specifies the visual events in terms of which faces appear/disappear when moving from one aspect to another. Furthermore, the position of such a face appearance/disappearance from a source aspect to a target aspect is specified with respect to particular contours of faces in the source aspect (event contours). Finally, the transition between two nodes (aspects) encodes the direction(s) relative to the event contours that one must move in the image plane in order to observe the visual event.

The aspect prediction graph is also based on a hierarchical part-based aspect representation, called the *aspect hierarchy*, which uses aspects to represent a (typically small) set of volumetric primitives from which each 3-D object in a database is constructed, rather than representing entire objects directly [5]. Consequently, we use aspects to recover the 3-D volumetric primitives that make up an object in order to carry out a recognition-by-parts procedure, rather than attempting to use aspects to recognize the entire object. A unique feature of the aspect hierarchy is a set of conditional probabilities linking features at different levels of the hierarchy. The aspect prediction graph borrows from the aspect hierarchy both the $Prob(volume|aspect)$ and $Prob(aspect|volume)$ conditional probabilities, and assigns them to the nodes in the aspect prediction graph.

We have constructed an ***aspect prediction graph*** for a number of simple volumetric primitives. Given an aspect of a volume, the observer can usually move in more than one direction to get to some other aspect. To cover all these alternatives, the aspect prediction graph encodes multiple arcs between the aspects, each representing a qualitatively distinct view change direction. In addition, associated with each of these arcs are one or more *face events*. Each face event specifies what face will appear or disappear under the change of view corresponding to the arc, and where the event will occur relative to the source aspect.

To illustrate the above concepts, Figure 2 presents the aspect prediction graph for the block volume, illustrating the three possible aspects of the block. Between every two nodes (aspects) in the aspect prediction graph are a pair of directional arcs. The directional arc between aspect 1 and aspect 2 in Figure 2 is expanded in Figure 3. From aspect 1 in Figure 2, there are three ways to move to a view in which aspect 2 will be visible. Movement relative to contours 0 and 1 on face 2 will cause a visual event in which face 2 disappears at contour 1 on face 0 and at contour 3 on face 1. Or, movement relative to contours 0 and 1 on face 0 will cause a visual event in which face 0 will disappear at contour 0 on face 1 and contour 0 on face 2. Finally, movement relative to contours 0 and 3 on face 1 will cause a visual event in which face 1 will disappear at contour 0 on face 0 and contour 1 on face 2.

It should be noted that in the aspect hierarchy, each aspect has an indexing of its component faces, and each component face has a similar indexing of its bounding contours. By referring to the normals of such well-defined contours, we can qualitatively specify direction rules with respect to an aspect-centered coordinate system. The direction of view change is

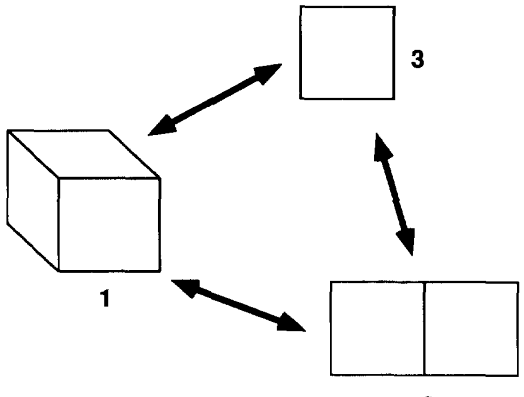

Figure 2: Aspect Prediction Graph (APG) for Block

specified as a vector sum of the normals to a particular set of recovered aspect contours that correspond to the model contours of an aspect prediction graph node.[1] The face events are also defined with respect to these specified contours. For example, we can predict along which contour in the current aspect a new face will appear or disappear when moving towards a new aspect.

3.2 Recognizing Visual Events

The symbolic tracker specifies the criteria for which a visual event will be detected by the image tracker. For the experiments in this paper, we will use region area as the single criteria. If at any time during the image tracking of an aspect, one or more of its faces' areas goes below some threshold, we interpret that to mean that the face is undergoing heavy foreshortening and will soon disappear. When a region's area drops below the threshold, the image tracker sends a signal to the symbolic tracker. Given its current position (node) in the aspect prediction graph, the symbolic tracker compares the outgoing arcs, or visual events, with the events detected by the image tracker. The arc in the aspect prediction graph matching the observed visual event defines a transition to a new aspect.

3.3 Predicting a New Aspect

The transition between the current APG node and the predicted APG node defines a set of visual events

[1]For concave and convex curve segments, the normal at the midpoint is used.

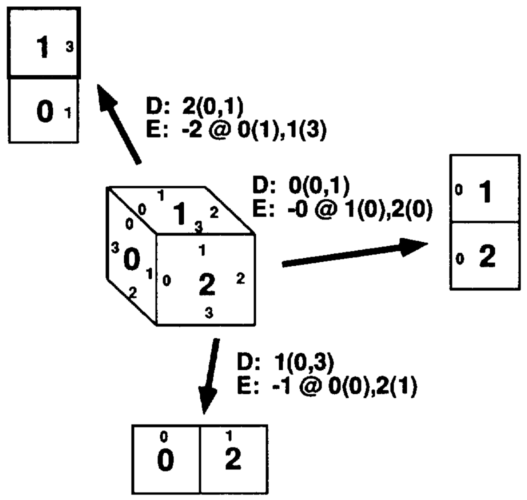

Figure 3: APG Transitions from Aspect 1 to Aspect 2 in Figure 2

in terms of the faces in the aspect defined by the current APG node. If one or more faces disappear from the current aspect to the predicted aspect, the symbolic tracker directs the image tracker to delete those contours from the adaptive adjacency graph which both belong to the disappearing faces and are not shared by any remaining faces. Alternatively, if one or more new faces are expected to appear, the symbolic tracker directs the image tracker to add structure to the adaptive adjacency graph. Since the symbolic tracker knows along which existing contours new faces should appear, it can specify between which nodes in the adaptive adjacency graph new contours should be added. The method in which new contours are added to the AAG will be addressed in Section 4.

3.4 Initialization and Error Recovery

An error state is reached when the visual events signaled by the image tracker are not consistent with any of the transitions emanating from the current node in the aspect prediction graph. When the error state is reached, the symbolic tracker terminates tracking and attempts to relocate itself in the aspect prediction graph. This initialization procedure involves acquiring an image and recovering the aspect of the volume. Further details on this process can be found in [5, 4].

4 Image Tracker

The image tracker employs a representation called an adaptive adjacency graph, or AAG. The AAG is initially created from a recovered aspect, and consists of a network of active contours (snakes) [8] which are connected at nodes. In addition, the AAG captures the topology of the network's regions, as defined by minimal cycles of contours. Contours in the AAG can deform subject to both internal and external (image) forces while retaining their connectivity at nodes. Connectivity of contours is achieved by imposing constraints applied at contour ends. If an AAG detected in one image is placed on another image that is slightly out of registration, the AAG will effectively "pull" itself into place.

4.1 Creating the Adaptive Adjacency Graph

As pointed out in Section 3.4, the initialization procedure recovers an object's (or part's) aspect from the image in order to locate its position in the APG. A recovered aspect encodes a set of faces and their connectivity. Furthermore, each recovered face defines a chain of bounding contour pixels partitioned at significant curvature discontinuities. In addition, each face defines a region mask specifying those pixels internal to the face. Nodes in the AAG are located at points where partitioned contours in the recovered aspect coterminate, while arcs are created from the recovered partitioned contours. The open rigid curves bounding a face are converted into "flexible" active contours by resampling the original chain, and are stored in the AAG together with information on their connectivity. The main feature of the AAG is its adaptability to internal and external forces while retaining its topology. This is achieved by specifying a set of constraints for active contours that preserve connectivity of the contours at their nodes [7].

4.2 Tracking with the AAG

The basic behavior of the AAG is to track image features while maintaining connectivity of the contours and preserving the topology of the graph. This behavior is maintained as long as the positions of active contours in consecutive images do not fall outside the zones of influence of tracked image features. This, in turn, depends on the number of active contours, the density of features in the image, and the disparity between successive images.

If either the tracked object or the camera moves between successive frames, the observed scene may change due to disappearance of one of the object faces. The shape of the region corresponding to the disappearing face will change and eventually the size of the region will be reduced to zero. The image tracker monitors the sizes and shapes of all regions in the AAG and detects such events.

4.3 Modifying the AAG topology

Topological modification of the AAG is performed by the symbolic tracker and represents the predicted changes in appearance from one APG node to another. Specifically, a modification consists of either removing the unshared contours and nodes of an object face predicted to disappear, or to add nodes and contours belonging to a new face predicted to appear. Decision as to which contours to remove/add is specified by the transition from the current APG node to a predicted APG node.

Creation of a new aspect face (region in the AAG) involves creating new nodes, creating new active contours between specified nodes, and linking specified contours with springs. Placement of new nodes and contours is specified according to the positions of existing nodes and contours in the AAG. If the predicted contours do not "lock-on" to new gradient ridges appearing in the image, the image tracker concludes that the prediction is wrong. The image tracker accomplishes this verification step by monitoring the area of the predicted face. If it fails to increase in area shortly after it is added to the AAG, then the prediction is not verified.

5 Results

In Figure 4, we demonstrate our tracking technique on a sequence of images taken of a rotating block. Note that for the first frame, the AAG was created from the recovered aspect. For subsequent frames, a blurred, thresholded, gradient image is used to exert external forces on the AAG. Moving left to right, top to bottom, we can follow the AAG as it tracks the image faces. When the foreshortened face's area falls below a threshold, the visual event is signaled to the symbolic tracker. Consequently, the nodes and contours belonging to the disappearing faces are removed while nodes and contours belonging to the face predicted to appear are added. Note that in order to ensure that new contours and old contours do not "lock on" to the same image gradient ridge, the contours are automatically "pulled apart", so that they will converge to the correct edges in the image. We are currently investigating the use of repulsion forces that would more effectively prevent network contours from converging.

6 Limitations

We are currently addressing a number of limitations of the current implementation. Although initialization can be performed in the presence of occlusion, there is currently no mechanism for detecting occlusion within the image tracker. Another limitation of the approach is the way in which the structure of the adaptive adjacency graph is altered to reflect new face predictions. Since predictions are located at specific model contours, the structure of the added face must reflect the structure of the model face in the model aspect. Finally, our image tracker currently only reacts to the disappearance of faces in the image. We are currently investigating the use of face detectors positioned at points along the AAG's outer boundary to detect the appearance of new faces.

7 Conclusions

We have presented a method for qualitatively tracking the pose of a 3-D object without the need for a model specifying the object's exact geometry. Use of an aspect prediction graph allows a symbolic tracker to track the object as it moves from one qualitatively different view to another. Using a network of active contours, the adaptive adjacency graph can quickly track an aspect using only gradient information computed from an image. By detecting certain kinds of deformations in the adaptive adjacency graph, the image tracker can predict impending visual events in the symbolic tracker. In addition to providing a means for qualitative object tracking, the aspect prediction graph extends previous active contour trackers beyond simply tracking an object's silhouette. The integration between the aspect prediction graph and the adaptive adjacency graph allows us to track both the external and internal discontinuities of an object. Furthermore, being able to predict how the structure of this network can change allows us to track the qualitative orientation of the object in terms of its aspect.

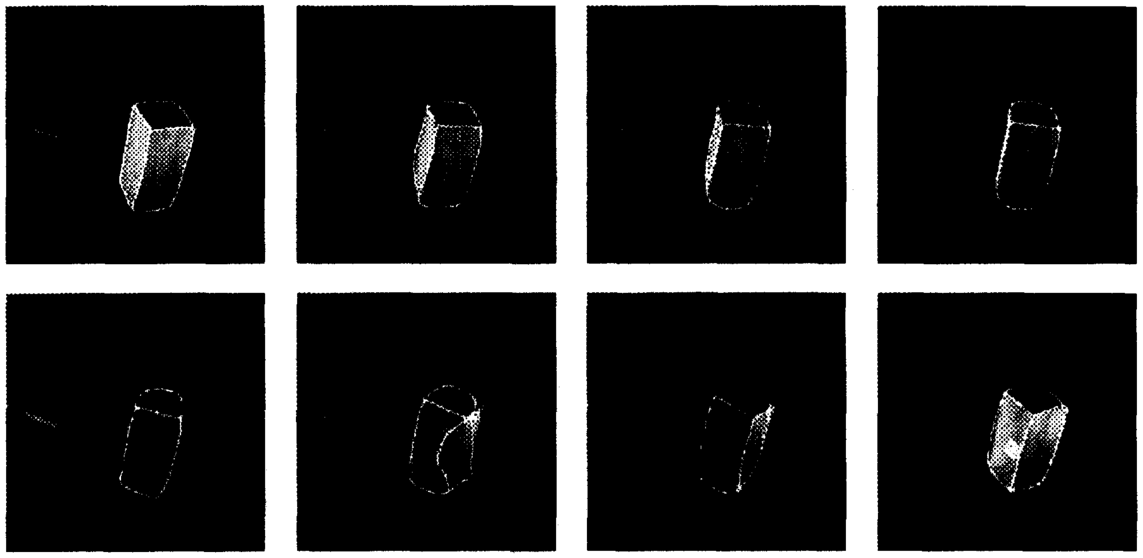

Figure 4: Tracking a Rotating Block. There were 11 images in the sequence with 10 iterations of the AAG per image for a total of 110 snapshots of the AAG. Working left to right and top to bottom, we show snapshots 1, 23, 32, 41, 70, 82, 90, and 110. Note that when the disappearing face is detected (70), the new face is predicted and contours are added (82). The added contours are automatically "pulled apart" to ensure that they do not converge to the same image edge; final position of the new edge is shown in frame 90.

References

[1] A. Amini, T. Weymouth, and R. Jain. Using dynamic programming for solving variational problems in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(9):855–867, September 1990.

[2] R. Cipolla and A. Blake. Motion planning using image divergence and deformation. In A. Blake and A. Yuille, editors, *Active Vision*, pages 189–201. MIT Press, 1992.

[3] R. Curven, A. Blake, and R. Cipolla. Parallel implementation of lagrangian dynamics for real-time snakes. In *Proceedings, British Machine Vision Conference (BMVC '91)*, pages 27–35, September 1991.

[4] S. Dickinson, H. Christensen, J. Tsotsos, and G. Olofsson. Active object recognition integrating attention and viewpoint control. In *Proceedings, ECCV '94*, Stockholm, May 1994.

[5] S. Dickinson, A. Pentland, and A. Rosenfeld. 3-D shape recovery using distributed aspect matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):174–198, 1992.

[6] D. Gennery. Visual tracking of known three-dimensional objects. *Int. J. Computer Vision*, 7(3), 1990.

[7] P. Jasiobedzki. Adaptive adjacency graphs. In *Proceedings, SPIE Geometric methods in Computer Vision*, pages 294–303, San Diego, CA, 1993.

[8] M. Kass, A. Witkin, and D. Terzopolous. Snakes: Active contour models. *Internation Journal of Computer Vision*, 1(4):321–331, 1988.

[9] J. Koenderink and A. van Doorn. The internal representation of solid shape with respect to vision. *Biological Cybernetics*, 32:211–216, 1979.

[10] C.-H. Liu and W.-H. Tsai. 3D curved object recognition from multiple 2D camera views. *Computer Vision, Graphics, and Image Processing*, 50:177–187, 1990.

[11] D. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5):441–450, 1991.

[12] M. Seibert and A. Waxman. Adaptive 3-D object recognition from multiple views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):107–124, 1992.

[13] D. Terzopolous and R. Szeliski. Tracking with kalman snakes. In A. Blake and A. Yuille, editors, *Active Vision*, pages 3–21. MIT Press, 1992.

[14] G. Verghese, K. Gale, and C. Dyer. Real-time, parallel tracking of three-dimensional objects from spatiotemporal sequences. In V. Kumar, P.S. Gopalakrishnan, and L.N. Kanal, editors, *Parallel Algorithms for Machine Intelligence and Vision*. Springer-Verlag, New York, 1990.

[15] T. Wallace and O. Mitchell. Analysis of three-dimensional movement using fourier descriptors. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 2(4):583–588, 1980.

[16] D. Williams and M. Shah. A fast algorithm for active contours and curvature estimation. *CVGIP:IU*, 55(1):14–26, January 1992.

[17] J. Wu, R. Rink, T. Caelli, and V. Gourishankar. Recovery of the 3D location and motion of a rigid object through camera image (an extended Kalman filter approach). In *Int. J. Computer Vision*, volume 3, pages 373–394, 1989.