# Delaying Commitment in Dynamic Stereo Vision

John D. Lewis
Ian H. Witten

Computer Science Department
The University of Calgary
Calgary T2N 1N4, Canada

## Abstract

To interpret a sequence of stereo views of a scene one must combine the information in each stereo pair and integrate the information over all pairs in the sequence. The former involves relating the parts of the two images, while the latter requires a correspondence between the models created from each view. Points that are fully determined in successive models are used to calculate, or refine, the camera motion between views. The models are then coalesced and components that are only partially determined in each model are disambiguated.

The major difficulty is in deciding between multiple plausible matches. A single stereo pair may have several registrations that can only be disambiguated after further information is provided. We describe a scheme that maintains a set of candidate models of the scene, each being as complete as possible given the views seen so far. These constrain the interpretation of subsequent views: any interpretation must be compatible with at least one candidate model, and every model must be compatible with some interpretation of a newly-obtained view.

The system has been implemented and tested on simple blocks scenes. It successfully integrates information from multiple ambiguous stereo pairs, given an accurate and noise-free specification of the lines in each view.

**Keywords**: stereo vision, scene interpretation, motion tracking, 3D correspondence, model matching, representing partial knowledge

## Motivation

The human visual system must often rely on information gleaned from successive views of a scene to provide an unambiguous interpretation. Obviously you cannot see all vertices and edges of even a simple scene, for instance a cube, from a single point, and in order to check that it *is* a cube you must move your head around, or move the cube around, to examine the hidden parts.

Each individual view is potentially ambiguous. When interpreting a single stereo pair the difficult part is solving the *stereo correspondence problem*, which involves determining which parts of one image correspond to each part of the other. Many clues in the images can be exploited, in particular the fact that stereo disparity varies smoothly almost everywhere and the fact that the image must have a consistent 3D interpretation (Marr and Poggio, 1979).

When interpreting a dynamic sequence of stereo images an analogous problem occurs, that of determining which part of the interpretation derived from one view corresponds to each part of the interpretation derived from another. We call this the *3D correspondence problem*, because unlike the stereo version of the problem a relation between 3D models is sought. Again, the correspondence may well be ambiguous, particularly if substantial camera or object motion occurs between images of the sequence and the estimate of this motion is inaccurate.

Schemes for dynamic stereo vision use statistical estimation procedures to derive camera motion parameters from a sequence of image pairs (Moravec, 1980; Matthies, 1989). They rely on finding enough information in each stereo pair to determine the correct registration between pairs. These techniques are likely to work satisfactorily so long as camera motion is slow and there is sufficient information in the images to allow reliable interpretation. Otherwise, heuristic methods (such as detecting and discarding outliers) are used to select a "best" registration. Systems that rely on these are generally unable to recover should their guess at registration ultimately prove wrong.

In principle it seems possible to carry forward multiple interpretations from one image to another, and use the information from succeeding images to disambiguate parts of the interpretation of a stereo pair. This resembles the "version space" approach to concept learning, which offers a structure for representing all possible generalizations consistent with the examples seen so far (Mitchell, 1982). If all conceivable generalizations are listed and stored, then, when an example is presented, the list can be pruned by removing those that are inconsistent with the example. Similarly, if all possible 3D interpretations of a stereo pair can be listed and stored, then subsequent images can be used to prune the set of candidate interpretations. We call these interpretations "models."

Of course, it is infeasible to store all possible models of an image—an unbounded number of variations can be produced by imagining different objects hiding behind a surface! Clearly it is essential to be able to represent partial information and use subsequent images to fill in more detail. Even so, there may well be multiple partial models that must each be updated with information gleaned from new images. When no consistent interpretation can be found of the new image with respect to an existing model, the model can be deleted from the set of candidates.

This approach is tantamount to delaying commitment to a particular interpretation as long as possible, a principle that has proven useful in many areas of computer science (Thimbleby, 1988). This paper describes a system that is intended to demonstrate its feasibility in the domain of image interpretation. To allow our efforts to be concentrated on the issues at hand, we operate in a greatly simplified visual domain and under ideal, noise-free, conditions. However, the ideas should be extensible to more realistic vision problems, although we have not tested this.

## Examples

We begin by giving simple examples of how ambiguities arise in stereo sequences and how our methodology assures a correct resolution in each case.

### VISUAL AMBIGUITIES

Stereo interpretation involves triangulating corresponding points from the left and right images of a stereo pair. Triangulation is a simple geometric operation that intersects the two lines defined by the image points and their associated focal points. Thus given the relative orientation of the two cameras in Figure 1a, corresponding points can be triangulated to establish their 3D location in eye coordinates.

The most natural interpretation of this stereo pair is a single cube directly in front of the camera. However the same images would be created by two smaller cubes, each falling inside the view cone of just one of the cameras. Figure 1b illustrates both situations. In general it is not easy to establish the correspondence between points in the left and right images. Although many techniques have been devised to find the correct stereo correspondence for arbitrary pairs of images, some scenes—like that of Figure 1—are inherently ambiguous and any attempt to discover the correct interpretation from just one view is futile.

Ambiguous interpretations abound in complex scenes. Figure 2a shows a stereo pair that supports many different interpretations. It seems natural to match the left block in the left field of view with the left block in the right field. However, the images can be interpreted as three cubes rather than two, where the leftmost is invisible to the right camera and the rightmost is invisible to the left camera. Under this interpretation the left cube in the right image corresponds to the right cube in the left image, as illustrated in Figure 2b. There is also a third possibility, illustrated in Figure 2c, where one block is very close to the eyes while the other is much further away and correspondingly larger, and the left block in the left field matches the *right* block in the right field. Many other configurations can be found as permutations of these three basic possibilities.

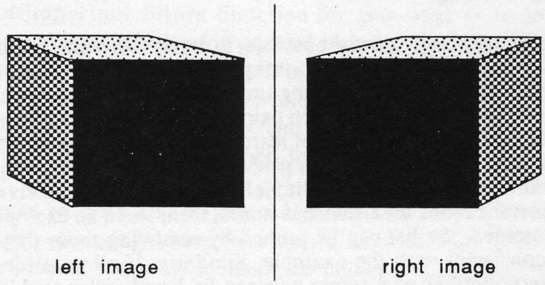
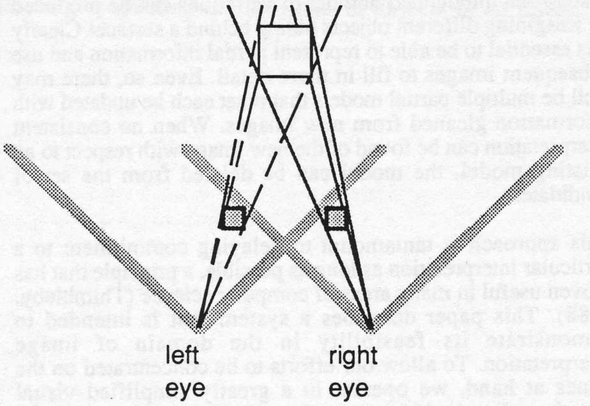


Figure 1 (a) Stereoscopic view of a simple scene



(b) Two possible interpretations

These examples illustrate the difficulty in interpreting an individual stereo view. There may be multiple interpretations that correspond to possible physical scenes, none of which can be discarded in the absence of further information. Eye motion provides the information needed to resolve the ambiguity.

### DISAMBIGUATING VIA STEREO SEQUENCES

Figure 1a can be interpreted correctly by simply moving the camera in any direction. Consider the view that results from moving it backwards. If the scene really does comprise a single cube, the 3D eye coordinates calculated from Figure 1a will be adjusted by an amount equal to the backward motion of the camera. If, however, the z-coordinate of some point in this single-cube interpretation grows much faster than the camera motion, or if two blocks appear in the field of view, the interpretation of the scene as a single cube can be discarded in favor of the second interpretation.

Similar arguments apply to the interpretation of Figure 2a. If the camera moves in any direction the resulting images will be very different for each of the three possible scenes. For example, suppose it moves straight backwards. If the scene comprises two blocks side by side, the new view will resemble Figure 2a, but the images of the cube will be slightly smaller and more towards the inside edge of both images. However, if it contains one block near the camera and another farther away, as portrayed in Figure 2c, then a different pair of images would result in which the left camera shows a small

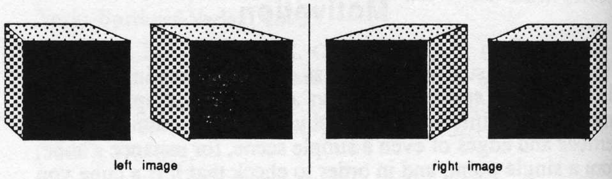
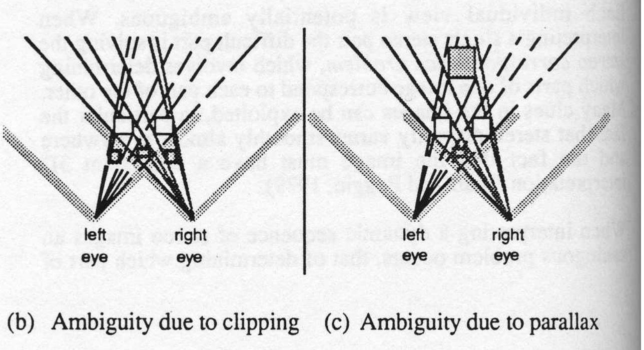


Figure 2 (a) Stereoscopic view of a more complex scene



(b) Ambiguity due to clipping (c) Ambiguity due to parallax cube on the right and a larger one to the left, while the right camera shows the reverse. The third possibility for the actual scene—three cubes—might reveal all three cubes in both images. In each case only one interpretation of the new view will be reconcilable with any of the interpretations of the original, allowing the correct correspondence to be established.

Our goal is to automate the reasoning that underlies these examples.

## Design Overview

We are presented with a dynamic sequence of stereo views of simple blocks-world scenes. By supplying line drawings rather than real images we avoid problems of line-finding, and by providing artificial images generated by a 3D computer graphics system (Wyvill *et al.*, 1986) we avoid problems of noise. These stratagems are obviously quite unrealistic in practice, but they allow us to concentrate exclusively on the issues raised by delaying commitment to any particular interpretation of an image.

The analysis of a dynamic sequence of stereo views begins with the interpretation of the first stereo pair as a 3D model. This is accomplished by finding a point match and then triangulating corresponding points. In general, however, multiple matches will result and some test must be used to eliminate incorrect ones. A strong criterion for eliminating these "false targets" is to check that the 3D models formed by triangulating corresponding points are physically possible. This is accomplished by ensuring that the interpretation of the individual monocular images as 3D objects is not violated by the correspondence. While this eliminates some false targets it will not necessarily resolve all the ambiguity. All remaining interpretations of the first view are retained as the initial set of *candidate models*.

Valid interpretations of each subsequent view must agree with this candidate model set and, conversely, the candidate model set must agree with each new view. Testing for agreement involves an overlay process which is controlled by an estimate of motion. If two models are identical except for occlusion and view boundaries, then they are considered a match. The actual motion can be computed from this match using any four non-planar point pairs. Next the candidate model is transformed by the amount of the actual motion, augmented by any new information from the view model, and added to the candidate model set for the next view.

This overlay process is done for every candidate model and every view interpretation. As views are analyzed their 3D models are tested against candidates in the model set and fused with each one with which they are consistent. If there are none, that model of the view is discarded. Conversely, if a candidate model is incompatible with all interpretations of the view, it is discarded. Thus the candidate model set is pruned by successive views, and augmented by any additional information available in them. This process ensures that at any point the candidate model set contains all interpretations that are consistent with every view to date. Figure 3 gives a flowchart of the procedure.

The result of the 3D matching stage is an updated model set, each member of which is described in the coordinate system of the most recent view and augmented by the information contributed by that view. Moreover, those candidate models that did not match a view interpretation are eliminated. Thus the new candidate model set is consistent with all previous view interpretations, as complete as possible, and pruned to the smallest possible set.

Interpretation of successive views can continue indefinitely. For a finite scene, if views are provided which show all aspects of the scene then the process will converge on a single complete and correct model. At this point no further information can be gained from further views. Nevertheless, successive views can be used to establish camera motion, and the model can be used to expedite the interpretation of new views.

The method applies equally to infinite scenes. A robot exploring an unfamiliar blockscape may use previous views to help interpret each new one and to correct for error in its motion estimate. Even though the model will never be completed it nonetheless serves to guide new interpretations. In fact even when ambiguity remains in the interpretation of a scene, the candidate model set can still support useful reasoning. For example, a robot may be able to find a motion that is safe in every candidate model, which will allow it to gain information that will help to determine the correct one.

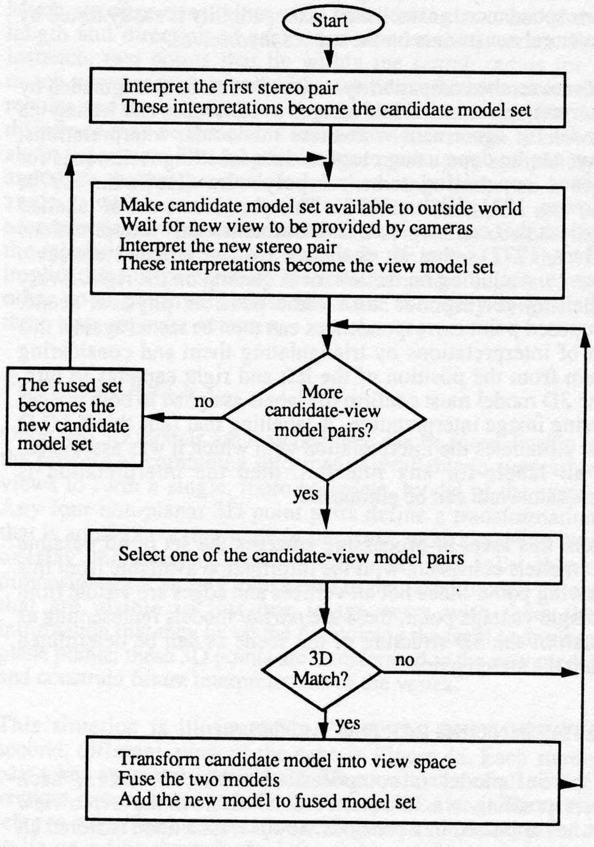


Figure 3 Flowchart of the overall interpretation procedure

## Details of Individual Steps

### INTERPRETING INDIVIDUAL STEREO PAIRS

The interpretation of a single stereo pair involves finding the correspondence between points in the left and right images and triangulating to find their 3D coordinate values. Since each camera receives a different image, this cannot be accomplished by seeking a direct match between the two images. However, a good preliminary filter results from the fact that matching points must have the same y-coordinate value (in eye coordinates), reducing the search for a correct match to one dimension.

Marr and Poggio (1977) suggested that at this point the only valid criterion for eliminating incorrect matches is that they are not physically realizable. They discussed three constraints derived from physical properties of the world:

- continuity—the disparity function varies smoothly except at edge boundaries;
- uniqueness—each image point corresponds to exactly one point in the scene;
- compatibility—the stereo interpretation corresponds to a physically possible scene.

Within the blocks world these tests are much simpler than in a real scene. A simple structural constraint—the fact that each surface is planar—ensures continuity. Uniqueness is guaranteed by extracting points from the image when a correspondence is established. Compatibility is easily tested by structural constraints on the scenes themselves.

Moreover, the compatibility constraint can be strengthened by interpreting the individual images as 3D scenes and testing the model for agreement with these monocular interpretations. This can be done using classical line-labelling techniques for scenes comprising trihedral polyhedra (Huffman, 1971; Clowes, 1971; Waltz, 1975). This produces a set of labelled vertices that can be given "2-1/2 dimensional" interpretations (Marr, 1977)—that is, each line can be labelled concave, convex, occluding on the left, or occluding on the right. Every labelling corresponds with some possible physical scene. Proposed point correspondences can then be tested against this set of interpretations by triangulating them and considering them from the position of the left and right cameras in turn. The 3D model must confirm the labels assigned to both images during image interpretation. A labelling that fails this 2-1/2 D test eliminates the interpretation with which it was associated. If all labels for any line fail, then the interpretation is impossible and can be eliminated.

From this level of processing emerges the set of all possible 3D models consistent with the information available from this viewing point. Since not all vertices and edges are visible from a single vantage point, these are *partial* models representing as much of the 3D structure of the scene as can be determined from the stereo pair.

### REPRESENTING PARTIAL MODELS

A partial model is composed of a list of *nodes*, each corresponding to a 3D vertex that is visible in the stereo view (or has appeared in a previous view). With a node is stored all the information that has been gleaned about it, namely

- 2D coordinates of the vertex in each of the images;
- constraints on the 3D position of the vertex;
- details of each edge that connects with the vertex.

If the point is visible in both images of a stereo pair its 3D coordinates can be calculated, in which case the second component records these values. If it is visible in one image only, it is constrained to lie on the line joining the eye position to the point's position on the image plane. If it is visible in neither view, no constraint is stored.

The third component lists information about edges incident on this vertex. Since the blocks world contains only trihedral vertices, each has exactly three connecting edges. With each edge is stored

- its 3D length;
- a 3D unit vector denoting the direction it leaves the vertex;
- an identification of the other node that lies on the edge;
- the line label attached to the edge during image analysis.

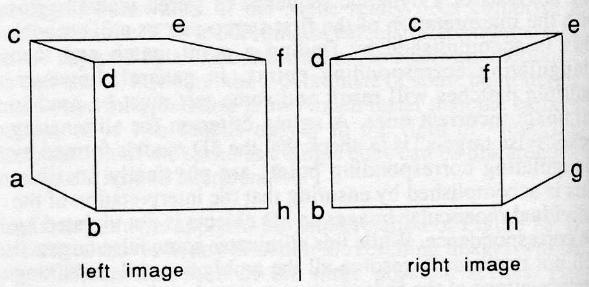


Figure 4 (a) Stereo view of a cube

```
(
                                    .
                                    .
                                    .
; vertex a
 ((((xal yal) (?xar ?yar)))         ; l and r image points
  (line ((1 0 0) (xal yal zip)))  ; point lies on this line
  ((?length ?direction              ; edge to b
    (((- > >)(xbl ybl))  (nil (xbr ybr))))
   (?length ?direction              ; edge to c
    (((< - <)(xcl ycl))  (nil (xcr ycr))))
   (?length ?direction              ; edge to unknown
    ((nil     (?x-l ?y-l))(nil (?x-r ?y-r)))))
                                    .
                                    .
         .

; vertex c
 ((((xcl ycl) (xcr ycr)))           ; l and r image points
   (point (xc yc zc))               ; 3D point is known
   ((?length ?direction             ; edge to a
     (((> -) (xal yal)) (nil       (?xar ?yar))))
    (1.021 (-0.001 -0.006 -0.999); edge to d
     (((+ +) (xdl ydl)) ((- > >) (xdr ydr))))
    (1.006 (-1 0 0)                 ; edge to e
     (((< -) (xel yel)) ((< - <) (xer yer)))))
                                    .
                                    .
                                    .
 )
```

(b) Part of the corresponding data structure

Figure 4b shows a partial node list that results from analyzing the view of Figure 4a as a cube. The first vertex, `a`, is visible in the left image only, and has 2D coordinates `(xal,yal)`—the "`l`" standing for left image.[1] It is constrained to lie on a line from `(1,0,0)`, the left eye position, to `(xal,yal,zip)`, where `zip` is the z-value of the image plane. The three edges from `a` are to vertices `b` and `c`, and to a third, unknown, vertex—actually `g` although the relationship is not visible in either image.[2] The permissible line labels are `{>, <, +, -}`, representing bidirectional occlusion, convex, and concave edges respectively, and each edge is multiply labeled because the scene is ambiguous. Thus, for example, `ab` and `ac` could be labeled - and < respectively, or > and -, or > and <.

The next entry, for vertex `c`, illustrates the case where a point appears in both images. Both sets of 2D coordinates are stored. The point is constrained to a particular 3D location, written `(xc,yc,zc)`. Now the length and direction of some edges can be identified: for example, edge `cd` has length 1.021 units and 3D direction cosines `(-0.005, -0.006, -0.999)`. The edge `cd` illustrates that the line labels will not necessarily be the same for both left and right image—it is labeled with `(+ +)` in the left and `(- > >)` in the right image to indicate that in the left image it must be interpreted as convex while in the right it may either be concave or occlude a surface that lies to its left.

The partial node list shown in Figure 4b is from the interpretation of Figure 4a as a cube. There is however a second interpretation—that of two cubes—which will be produced by the analysis. In this interpretation each vertex will be constrained to lie on the line formed by its image point and corresponding focal point.

## INTERPRETING STEREO SEQUENCES

The information necessary to select the correct model can only be gained by moving the camera. As the camera moves the views obtained must all correspond to the actual scene, and so to each other. By finding plausible correspondences between the view interpretations and the candidate model set the set can be completed and pruned. Pairings of all view interpretations with all candidate models are considered. At least one of the pairings will yield a correspondence since one of the candidate models corresponds to the actual scene. All matches that are found are used to update the candidate model set by augmenting the old model with any information available in the new view. Thus the candidate model set can become more complete with each successive view. Pruning occurs as a consequence of not finding a correspondence between interpretations for different views. Since the candidate model set represents all possible interpretations of the views seen so far, each interpretation for a new view must correspond to at least one of the candidate models. Thus a view interpretation is discarded if it does not correspond to any of the candidate models. Conversely, any valid candidate models must correspond to a view interpretation (one of the view interpretations corresponds to the actual physical scene). Thus if a candidate model does not correspond to any of the view interpretations it is discarded. Clearly there will be some sequence of views that disqualifies all but one of the initial models, so the candidate model set can be pruned until eventually a single interpretation remains.

Fortunately the 3D correspondence problem is much more constrained than the 2D one. Given an estimate of camera motion—perhaps from a mobile robot—with known uncertainty, the search for corresponding vertices can be constrained to a small radius.

Another simple structural constraint on model matching is provided by connectivity. If node *n* is connected to nodes *a, b,* and *c* in one model and a corresponding node *n'* is connected to *a', b',* and *c'* in the second model, then *a, b,* and, *c* must be matched with *a', b',* and *c'*. Thus, the first vertex match that is proposed can be used to constrain the rest of the match. In many cases this constraint is sufficient to eliminate all but the correct match. However, the 3D models will generally be incomplete, at least for the first several views, because some vertices and edges will be occluded. So when establishing or rejecting a registration on the basis of isomorphism between 3D models, it is necessary to allow for the possibility of nodes appearing from behind, or disappearing behind, an occluding surface. The result is that the connectivity test is in practice considerably weaker than a full isomorphism test would be, and also more expensive to perform.

Much stronger constraints arise from taking account of the length and direction of each edge that leaves a vertex. For instance, two points that lie within the search radius for a match may not correspond with each other. Sometimes it will require an extensive connectivity test to determine this, but the difference may be detectable immediately if enough is known about the vertex. The length of edges must agree completely, and since the motion error is known within limits, so is the angle at which an edge leaves a vertex. Occlusion can be accommodated within this framework as well, by considering the relative positions of the objects in the scene, and the 3D implications of performing the transformation suggested by other point pairings. Unless the scene is highly symmetric these tests are quite stringent.

## FUSING MODELS

Once two 3D models are matched they can be *fused*. Fusion is the process of combining information available in successive views to form a single, more complete, model of the scene. Any four non-planar 3D point pairs define a transformation that is applied to every node in the scene model, causing it to overlay the current view. Points that are fully three-dimensional are simply transferred to the fused model. Points that are visible in just one image carry with them the information that they lie on the line joining the focal and image plane points; these 3D points are transformed to the new model and constrain future interpretations of the vertex.

This situation is illustrated in Figure 5a, which shows a second, different, view of the cube in Figure 4a. Each stereo pair taken separately defines the 3D coordinates of all visible vertices except `a` and `g`. In both views point `a` is visible to the left eye only. The interpretation of Figure 4a constrains point `a` to lie on a line through the left origin and the image point,

---

[1] In the actual output from the program, coordinates like `xal`, `yal`, `zip`, and so on will all be floating-point numbers.

[2] Atoms beginning with a question mark are unbound variables. Different instances of them are different variables, despite the use of the same names in Figure 4.

which is a small, known, distance from focal point along the z-axis. Figure 5a yields a second, similar, constraint. Moreover, when Figure 5a is processed, the previous interpretation—including the constraint line for point a—is transformed into the new eye coordinate system. This yields two 3D lines which intersect in space at the coordinates that correspond to a. The procedure for g is similar. A portion of the data structure for Figure 4a was shown in Figure 4b; the corresponding portion after fusion is shown in Figure 5b. Notice that the fused model contains information not present in either view individually.

As each view is processed each candidate model is fused with any view interpretations with which it is consistent. The candidate model set for the next view comprises these fused models. This process tends to make the candidate models progressively more complete, which in turn makes 3D model matching more constraining.

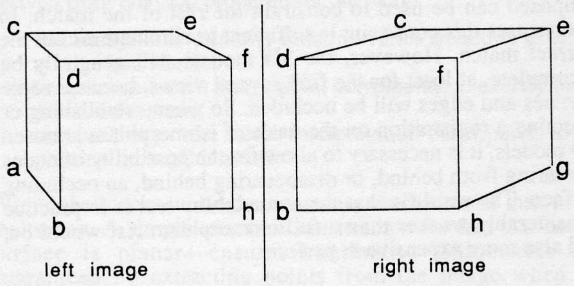


Figure 5 (a) A view of the scene of Figure 4a after moving up and right

```
(
                                        .
                                        .
                                        .
; vertex a
 ((((xal' yal') (?xar' ?yar')))      ; l and r image points
   ((xal yal) (?xar ?yar)))          ; image points from Fig 4b
  (point (xa' ya' za'))              ; 3D point is now known
  ((1.021 (-0.002 -0.003 -0.994)     ; edge to b
    (((- > >)(xbl' ybl'))  (nil (xbr' ybr'))))
   (1.013 (0 1 0)                    ; edge to c
    (((< - <)(xcl' ycl'))  (nil (xcr' ycr'))))
   (?length ?direction               ; edge to unknown
    ((nil    (?x-l' ?y-l'))(nil (?x-r' ?y-r')))))
                                        .
                                        .
                                        .
; vertex c
 ((((xcl' ycl') (xcr' ycr'))         ; l and r image points
   ((xcl ycl) (xcr ycr)))            ; image points from Fig 4b
  (point (xc' yc' zc'))              ; 3D point is known
  ((1.013 (0 -1 0)                   ; edge to a
    (((> -) (xal' yal')) (nil        (xar' yar'))))
   (1.021 (-0.002 -0.003 -0.994)     ; edge to d
    (((+ +) (xdl' ydl')) ((- > >)   (xdr' ydr'))))
   (1.006 (-1 0 0)                   ; edge to e
    (((< -) (xel' yel')) ((< - <)   (xer' yer')))))
                                        .
                                        .
                                        .
)
```

(b) Fusion of the new information with Figure 4b

## Conclusions

We have demonstrated the feasibility of delaying commitment in dynamic stereo vision within a highly structured domain. The approach described dispenses with heuristics to select the best registration and thus completely avoids the issue of how to recover when these fail. It provides a uniform and consistent way of integrating information from multiple stereo views of a scene, together with a reliable means of correcting errors in motion estimation.

It seems likely that the technique can be extended to non-rigid scenes, which cannot be accommodated by existing approaches (e.g. Matthies, 1989). To do so will necessitate extra domain knowledge—perhaps laws of dynamics. Retaining the ability to correct for motion estimation error, however, may be sacrificed—although the significance of such abilities in a dynamic environment would be greatly reduced.

Furthermore, since each candidate model fuses information from all views, the approach facilitates reasoning over a set of possible worlds. For many purposes it is enough to know a set of possible models without being able to determine which one is correct.

The crucial question is, will the method scale up to more realistic images? Certainly more comprehensive label sets should permit the incorporation of features such as shadows and cracks without real difficulty (Waltz, 1975). The constraint type on vertex positions—presently none, line, and point—suggests some interesting generalizations, such as allowing a vertex to be constrained to lie within a pyramidal region when it is known to be obscured by a face.

Accommodating real-world scenes requires significantly more effort. Continuity can be tested by checking that the disparity function for the set of point correspondence varies smoothly except at edge boundaries (Marr & Poggio, 1979). Compatibility, however, is somewhat more difficult. A form of depth map such as Coons surface patches or B-spline surfaces (Shirai, 1987) might be an appropriate representation. Proposed point correspondences would then form 3D surfaces, replacing the 2-1/2D test by one which tests for continuity and the 3D model matching stage by one that fits surfaces against each other. We are optimistic that these proposals will allow the basic idea of delaying commitment to a particular interpretation to be extended to deal with real-world scenes.

In conclusion, we believe that this work provides a well-founded basis for further research on the acquisition of knowledge about unfamiliar environments using autonomous navigation.

## Acknowledgements

This research is supported by the Natural Sciences and Engineering Research Council of Canada.

## References

Clowes, M.B. (1971) "On seeing things," *Artificial Intelligence 2* (1): 79–116; Spring.

Grimson, W.E.L, and Marr, D. (1979) "A computer implementation of a theory of human stereo vision," *Proc ARPA Image Understanding Workshop* 41–45.

Huffman, D.A. (1971) "Impossible objects as nonsense sentences," in *Machine Intelligence 6*, edited by B.Meltzer and D.Michie, Edinburgh University Press, pp.295–323.

Marr, D. and Poggio, T. (1979) "A computational theory of human stereo vision," *Proc Royal Society B 204*: 301–328.

Matthies, L. (1989) Dynamic stereo vision. PhD Thesis, Computer Science Department, Carnegie-Mellon University, October. Report CMU-CS-89-195.

Mitchell, T.M. (1982) "Generalization as search," *Artificial Intelligence 18*: 203–226.

Moravec, H.P. (1980) Obstacle avoidance and navigation in the real world by a seeing robot rover. PhD Thesis, Computer Science Department, Stanford University, September.

Shirai, Y. (1987) *Three–dimensional computer vision.* Springer–Verlag, Berlin.

Thimbleby, H. (1988) "Delaying commitment," *IEEE Software*, May, 78–81.

Waltz, D. (1975) "Understanding line drawings of scenes with shadows," in *The psychology of computer vision*, edited by P.H. Winston, McGraw Hill, New York, 1975.

Wyvill, B., McPheeters, C. and Garbutt, R. (1986) "The University of Calgary 3D Animation System," *Journal of the Society of Motion Picture and Television Engineer 95*, 629–636.