Geometric Reasoning for Constructing 3D Scene Descriptions from Images*

Ellen Lowenfeld Walker
Computer Science Department, Carnegie-Mellon University, Pittsburgh, PA 15213, U.S.A.

Martin Herman
Robot Systems Division, National Bureau of Standards, Gaithersburg, MD 20899, U.S.A.

ABSTRACT

There are many applications for a vision system which derives a three-dimensional model of a scene from one or more images and stores the model for easy retrieval and matching. The derivation of a 3D model of a scene involves transformations between four levels of representation: images, 2D features, 3D structures, and 3D geometric models. Geometric reasoning is used to perform these transformations, as well as for the eventual model matching. Since the image formation process is many-to-one, the problem of deriving 3D features from 2D features is ill-constrained. Additional constraints may be derived from knowledge of the domain from which the images were taken. The 3D MOSAIC system has successfully used domain specific knowledge to drive the geometric reasoning necessary to acquire 3D models for complex real-world urban scenes. To generalize this approach, a framework for the representation and use of domain knowledge for geometric reasoning for vision is proposed.

1. Generating 3D Descriptions from Images

The goal of a computer vision system is to derive a meaningful symbolic interpretation of a scene, given one or more images of that scene. A particularly useful interpretation is a three-dimensional geometric model of the scene. Such a model can be used for path planning for a robot, generation of alternate views, or change detection, as well as model-guided interpretation of additional images.

* This research was sponsored in part by AT&T Bell Laboratories' Graduate Research Program for Women, and in part by the Air Force Office of Scientific Research under contract F49620-83-C-0100. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Air Force Office of Scientific Research, or the US Government.

Artificial Intelligence 37 (1988) 275–290

In the process of image interpretation, the information is transformed through four levels of representation. These are:

(1) *Image(s)*. The original input image may be a single black and white image, a color image, a stereo pair, or a set of related views.
(2) *2D features*. Two-dimensional features extracted from the original images include vertices, lines, and regions of uniform intensity.
(3) *3D structures corresponding to 2D features*. Three-dimensional structures include 3D vertices, edges, and surface patches.
(4) *3D geometric models*. Three-dimensional geometric models may be edge-based, surface-based, or volume-based.

The system must be able to move between these levels, both extracting higher-level features from lower-level ones, and predicting lower-level features to match, using a higher-level model. Additionally, the current image-derived hypotheses may be matched to an internal model. Geometric reasoning is used both to move between levels, and to match representations at a given level.

2D feature extraction is exemplified by choosing a set of lines to fit edge points extracted from the image, or determining the best polygonal approximation of a region according to some criteria. When deriving 3D structures from 2D features, knowledge of the camera projection is combined with domain knowledge and the image data to derive the most reasonable explanation for each 2D feature. In the case of stereo or motion, additional constraints are derived from matching features from more than one image. When completing a model from a set of 3D structures, geometric reasoning is used to determine the type, position, and orientation of the structures necessary to complete each object, and to hypothesize additional structures or objects necessary for the model to conform to domain constraints. In the real world, for example, objects may not float in the air without support. In the domain of airplanes, all airplanes must have two symmetric wings.

Within this context, vision systems can be compared by asking the following questions:

- Which levels of representation are used?
- How is domain knowledge represented at each level?
- At which level is matching done?

Many early vision systems represented only 2D features, and their final model was a labeling of the regions in the image. For example, Ohta's system [8] divided a color image into regions, and labeled the regions as HOUSE, SKY, GRASS, or ROAD using region properties such as color, shape, position in the image, and adjacency. The domain knowledge was represented as a set of condition-action rules, and the matching was done at the 2D feature level. The ACRONYM system [2], in contrast, used all four levels of representation, represented its domain knowledge explicitly as graphs, and matched 2D features extracted from the image to 2D features predicted from the models.

The examples of geometric reasoning presented in this paper have been chosen from the 3D MOSAIC system, which is completely described in [4]. The 3D MOSAIC system also uses all four levels of representation. Its domain knowledge is represented implicitly within procedures, and matching is done at the 3D structure level.

The 3D MOSAIC system deals with complex real-world urban scenes, (e.g. Fig. 3(a)). These scenes contain many objects with a variety of shapes, surface textures and reflectances, as well as artifacts of the natural outdoor lighting conditions such as shadows and highlights. To deal with this complexity, multiple images obtained from multiple viewpoints are used. Objects extracted from each image are matched and the models are combined to derive a more complete model of the overall scene.

Figure 1 shows the levels of representation in the 3D MOSAIC system, and the processes which transfer between them. The input to the 3D MOSAIC system is a view of the scene, either a monocular image or a stereo pair. Processing occurs through the four levels of representation, from top to bottom in Fig. 1. First, 2D lines and junctions are extracted from the images. From these, 3D structures such as edges and vertices are derived, resulting in a

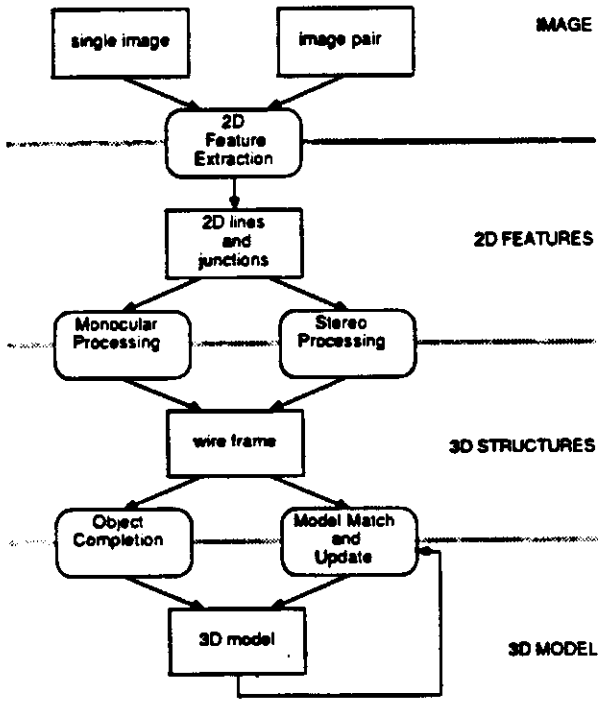


Fig. 1. Levels of representation in the 3D MOSAIC system.

sparse 3D wire frame description of the scene. The current scene model is represented as a graph of planar faces, edges, and vertices, and their topology and geometry. As each input image (or pair) is processed, the resulting wire frames are matched with the current scene model, and the model is modified to reflect them. Portions of the model for which there is no wire frame information are hypothesized using domain knowledge, in this case, knowledge about urban buildings.

The remainder of this paper discusses the use of geometric reasoning in the 3D MOSAIC system for the processes of monocular analysis and object completion. Some of the problems with the system are presented along with ideas for alleviating these problems in the next generation of the system.

2. Geometric Reasoning for Monocular Analysis

One method for determining 3D structures corresponding to 2D features extracted from a single image is to exploit strong geometric constraints from the domain along with the constraints of the projection to derive a unique interpretation for each 2D feature. This method is exemplified by the monocular analysis component of the 3D MOSAIC system [3–5], which generates a *wire frame* (see Fig. 3(c)) consisting of 3D descriptions of the edges and vertices corresponding to 2D lines and junctions that were extracted from the image.

In the 3D MOSAIC system under monocular analysis, all surfaces and edges are constrained to be either horizontal (parallel to the ground plane) or vertical (perpendicular to the ground plane). Therefore, the first step in generating the three-dimensional wire frame is to label each line as horizontal or vertical. The lines are labeled by exploiting a feature of the perspective projection: all vertical lines point at the vertical vanishing point [6]. Therefore, lines that point at the vertical vanishing point (a point specified by the user in the current implementation) are labeled vertical, and all others are labeled horizontal. In addition, since the views are known to be aerial, the end of each vertical line nearest the vertical vanishing point is labeled "bottom" and the other end is labeled "top".

Once the lines have been labeled, the 3D location of any endpoint of a line can be found given the 3D location of its other endpoint. There are two cases: horizontal lines and vertical lines. In Fig. 2, the 3D position of the vertex v_1 is known, line p_1p_2 is the image of a horizontal line in space, line p_1p_3 is the image of a vertical line in space, and $\boldsymbol{u}$ is a 3D vector in the vertical direction, found by calculating the vector from the focal point to the vertical vanishing point in the image plane [1]. The position of v_2, the far endpoint of the horizontal line, is found at the intersection of the ray from the focal point through p_2 and a plane parallel to the ground through v_1. The position of v_3, the far endpoint of the vertical line, is found at the intersection of the ray from the focal point through p_3 and a line parallel to $\boldsymbol{u}$ through v_1. Using these techniques, the position of any point can be derived, provided that a labeled

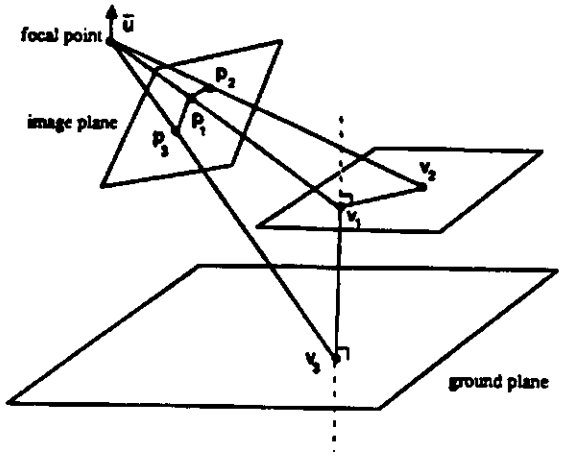


Fig. 2. Propagating 3D locations for monocular analysis.

line exists between it and a known point. Two techniques have been used to derive starting points.

The first technique uses points on the ground plane as starting points. The equation of the ground plane is $u \cdot p = -d$, where u is the vertical unit vector, p is a point on the plane, and d is the distance from the focal point to the ground plane. If d is not provided by the user, it remains a free variable in the plane equation, and only relative positions of vertices are obtained. The location of a point on the ground plane, such as v_3 in Fig. 2 is at the intersection of the ground plane and the ray through the point's image (in this case p_3). The points labeled "bottom" of vertical lines are considered to be on the ground for this analysis.

Not all junctions found in the image are connected to vertices on the ground. Some of these may be positioned using the second technique: if the 2D images of two lines are aligned, assume the 3D lines are aligned in the scene. This technique was also used in other systems [7]. When an unknown line aligns with a known one, the location of each endpoint on the unknown line may be found by intersecting the known line with the ray through the focal point and the endpoint's image.

This process of extracting wire frames from 2D lines and junctions depends on the following domain constraints:

- All surfaces and edges are either vertical or horizontal.
- An edge is vertical if and only if it points at the (known) vertical vanishing point.
- The bottom of a vertical edge lies on the (known) ground plane.
- Lines aligned in the image correspond to edges aligned in space.

The first two constraints allow each line to be labeled horizontal or vertical, ensuring that 3D locations may be propagated across each line. The remaining

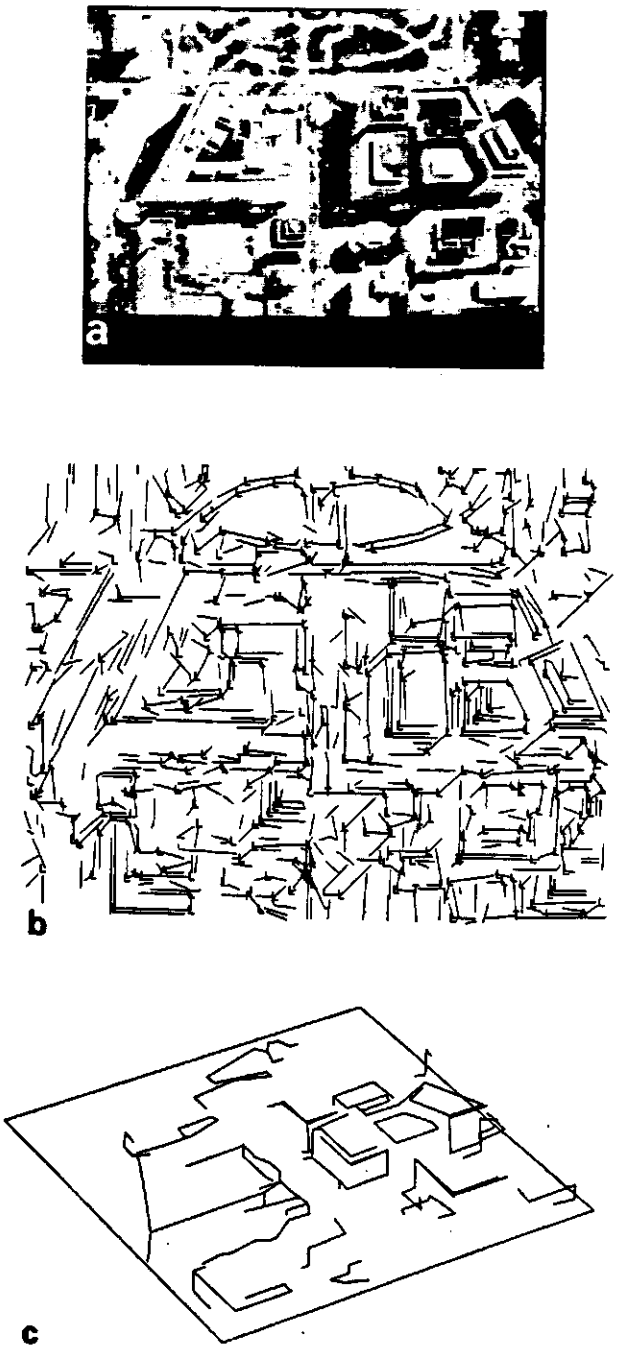


Fig. 3. Wire frame generation. (a) Initial image; (b) 2D lines and junctions; (c) wire frame.

two constraints allow for the two methods of determining initial starting points for the propagation.

Figures 3 and 4 are two examples of generating wire frames from monocular images using the techniques described in this section. Each figure contains (a) the grey scale image, (b) the extracted lines and junctions, and (c) a perspective view of the generated wire frame. Only junctions containing vertical lines and junctions whose 3D position could be obtained using the collinearity

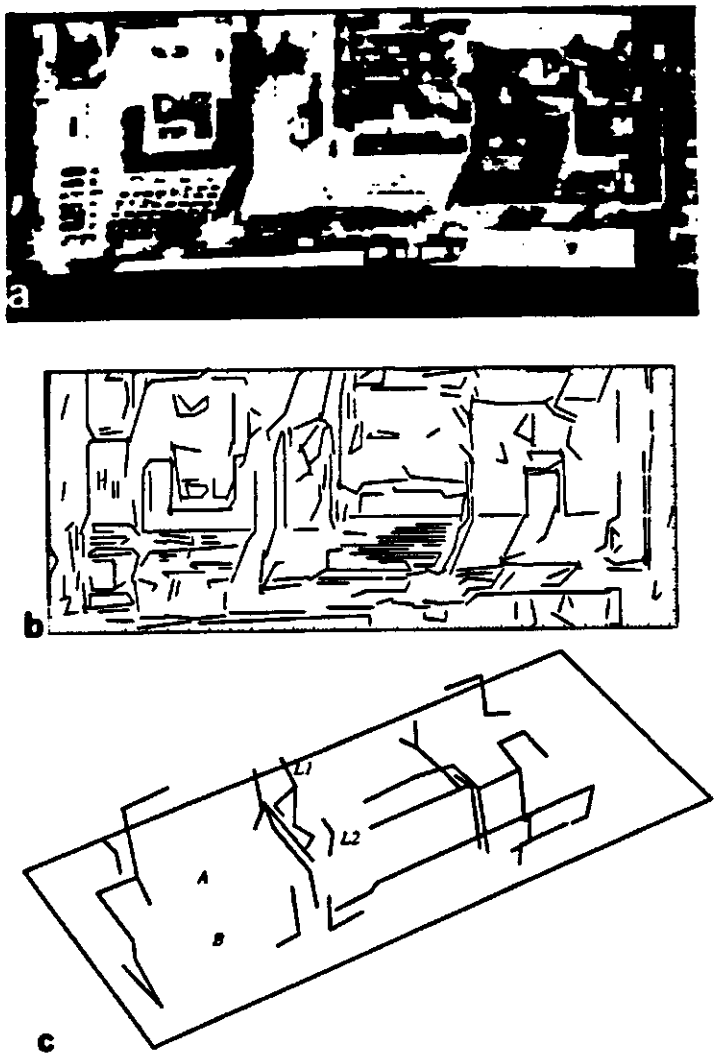


Fig. 4. Wire frame generation illustrating problems with monocular analysis. (a) Initial image; (b) 2D lines and junctions; (c) wire frame.

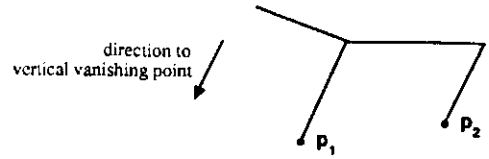


Fig. 5. The wire frame depends on the processing order.

constraint become vertices of the wire frame. The wire frame in Fig. 3 is a reasonable representation of the input image. Figure 4 illustrates several problems which can hamper the wire frame extraction.

One problem with monocular analysis is loss of information in the 2D feature extraction. For example, vertical lines in aerial images appear relatively short, so they are often ignored in the 2D feature extraction. In Fig. 4(c), the short vertical lines connecting the upper and lower roofs of the leftmost building were missed (area A). Therefore, no 3D structures were derived for the roof of that building. In addition, no 3D information is obtained for areas where there are no junctions, such as most of the front wall of the leftmost building in Fig. 4(c) (area B).

If short vertical lines are extracted, but their direction is off by only a few pixels, they will be labeled horizontal instead of vertical. An example is the line labeled $L1$ in Fig. 4(c). Occasionally a horizontal line (such as $L2$ in Fig. 4(c)) accidentally aligns with the vertical vanishing point and is labeled vertical. When lines are mislabeled, the errors in the wire frame propagate throughout the model, sometimes creating objects that should never occur in the domain. The current 3D MOSAIC system has no way to recover from these errors. Another problem is the possibility of generating inconsistent structures using this monocular extraction process. For example, in Fig. 5, if p_1 is processed first, then p_2 will lie above the ground. However, if p_2 is processed first, then p_1 will lie below the ground.

One goal of the 3D MOSAIC system was to use multiple views to recover reasonable three-dimensional models in spite of these problems. One possible area of research is to assign confidences to the 3D structures depending on their method of acquisition. Currently, hypotheses from higher-level processing are distinguished from structures found in the image, but structures from the image are not distinguished as to how they were derived. Another way to deal with the ambiguities in monocular images is to use more specific models of buildings: not just horizontal and vertical faces, but edge lengths, heights, etc., up to and possibly including complete CAD models of the buildings themselves.

3. Geometric Reasoning for Object Completion

To complete the generation of a 3D description from an image, the 3D structures extracted from the input views must be incorporated into a scene

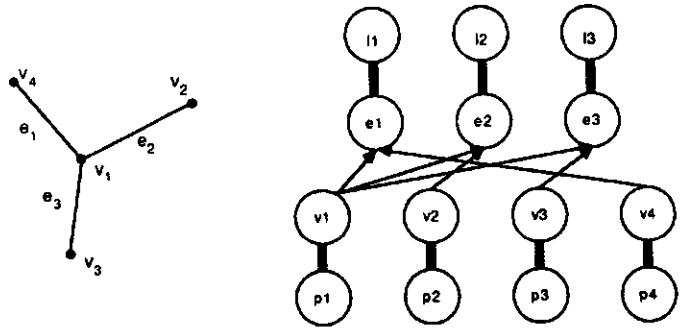
model. The completed scene model is the final result of the vision system. Therefore, the representation of the scene model is dependent on the use of the vision system. For some applications, for example, a space-filling representation might be most useful, while a surface-based representation would be best for other applications.

In the case of the 3D MOSAIC system, the model is designed to efficiently describe partially complete polyhedral objects, and to be easy to use in matching. The scene is represented as a graph of topological structures: vertices, edges, edge groups (rings of edges), faces, and objects, as well as the underlying geometric structures: points, lines, and planes. The topological features are linked to each other by part-of (topological) links, and to the geometric structures by geometric constraint links. This *structure graph* may be modified by adding or deleting nodes or links, or by changing the equations of geometric nodes. The effects of modifications are propagated to other parts of the graph. A complete description of the algorithms for updating the structure graph may be found in [4] or [5].

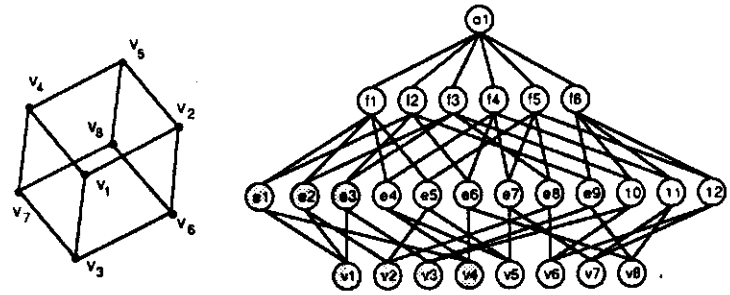
The wire frames are also represented by structure graphs, although the structure graphs of the wire frames contain only edge, vertex, line, and point nodes. The wire frame extracted from the first view of a particular scene forms the initial scene model. All vertices, edges, and points of the wire frame are tagged as confirmed. The initial scene model is augmented with additional vertices and edges, as well as edge rings, faces, and objects to derive a surface-based model of the scene, using knowledge of the domain. The elements of the final scene model that were not present in its initial state are tagged as unconfirmed. As an example, Fig. 6 shows the representations of an initial wire frame and a final model for a cube. In the initial wire frame, both topological and geometric nodes are shown, with the arrows representing part-of links and the thick lines representing geometric constraint links. The final model is simplified, showing only the topological nodes with confirmed nodes shaded.

To incorporate a wire frame extracted from a new view, the wire frame's structures are tagged as confirmed and matched to the current scene model. The wire frame is then merged into the model, "averaging" confirmed elements in the model with confirmed elements in the wire frame, and replacing unconfirmed elements in the model with their confirmed counterparts from the wire frame. The structures necessary to complete the model are then generated using domain knowledge, as for the first wire frame.

The steps taken by the 3D MOSAIC system for object completion are shown by an example in Fig. 7. The object to be completed (Fig. 7(a)) has one vertical edge and four coplanar horizontal edges. First, a face is hypothesized for each pair of edges that share a vertex. In Fig. 7(b) these faces are shown by "webs" between each face's edges. Then all pairs of faces that lie in the same plane and satisfy additional constraints are merged. One constraint is that faces that share



Initial wire frame



Final structure graph

Fig. 6. Completion of a cube.

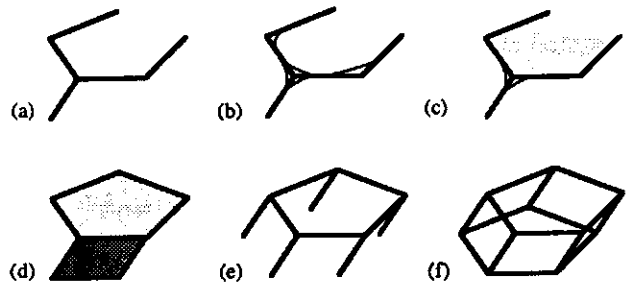


Fig. 7. Steps in object completion.

a single edge are merged (Fig. 8(a)), unless the shared edge separates the faces, as in Fig. 8(b). Another constraint is that if the confirmed edges of the faces do not touch, they must be within a threshold distance of one another. The result of this step is a set of partial faces. Only the "roof" face is represented in Fig. 7(c). Each partial face is then completed by adding one or two edges. If the partial face has two edges, two more are added, forming a parallelogram. If the face has more than two edges, then a single edge is added to complete the polygon formed by the edges of the face. These possibilities are exemplified by the roof and the front wall of the building in Fig. 7(d). Next, vertical edges are hypothesized to support all horizontal faces that lie more than a given distance above the ground (Fig. 7(e)). These edges are dropped from each vertex of the floating face to the next lower horizontal face, or to the ground. New faces are hypothesized for each pair of edges sharing a newly created vertex, and the process is repeated until no new faces can be hypothesized. Figure 7(f) shows the final building.

The knowledge used in completing wire frames is embedded in the procedures for hypothesizing, merging, and completing faces. Each wire is constrained to represent a boundary between two faces, and each vertex the intersection of three faces. Thus, a partial face is hypothesized for each pair of edges that meet at a vertex. This constraint is also used to derive the constraints on merging partial faces. For example, if two partial faces sharing an edge that separates them, as in Fig. 8(b), were merged, then the shared edge would not be the boundary between two faces. The distance threshold for merging nontouching edges is related to a constraint on the size of buildings in aerial images.

The domain constraint that buildings often have parallelogram faces is used to derive the strategy for completing faces. Finally, the real-world constraint that faces do not float in midair forces supporting edges to be hypothesized for floating faces.

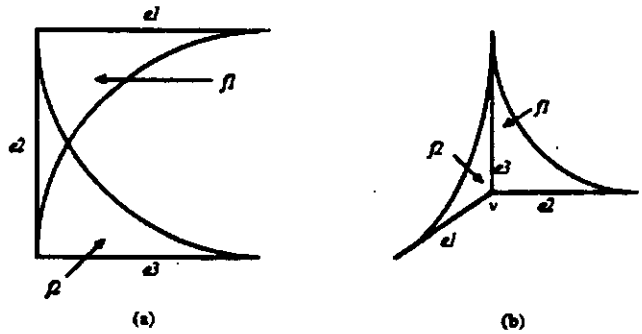


Fig. 8. Examples of heuristic for merging faces: (a) $f1$ and $f2$ should be merged because they share edge $e2$. (b) $f1$ and $f2$ should not be merged because $e3$ partitions them, rather than serving as a boundary. (This figure is adapted from [5].)

Errors in face completion are often caused by incorrectly assuming that a face with more than three edges should be completed by adding a single edge. Figure 9 shows two butterfly-shaped buildings that violate all the domain constraints. The dotted edge in each building was hypothesized by the system. In Fig. 9(a), the short spur *S* on one of the horizontal edges was connected to the opposite vertex with a single edge. In Fig. 9(b), vertical edge *L* was mislabeled as horizontal and connected to the other edges of the roof.

Even if all wires are correct, the true shape of the building can be lost if enough edges are not detected in early processing. For example, if only parts of the edges of a building were detected, the hypothesized edges would run through the middle of the building. Both types of errors could be prevented by accurately verifying hypothesized edges in the image.

Some experiments in verification have been run using an interactive version of the 3D MOSAIC system that allows the user to verify each edge and face hypothesis as it is made, by responding "yes", "no", or "change" the hypothesis. As an example, Fig. 10 shows a portion of the interactive completion of one building in Fig. 3(a). In the first two images, the web faces composing the roof are merged, with the user accepting each successive edge into the face. In the third image, the system proposes an incorrect completion of the face, and the user changes it to a more appropriate edge, in this case an extension of the previous edge. In the current version of the system, vertices are not deleted, so the vertex remains. The system then acceptably completes the face. The final two images show edges being dropped from the roof to the ground. The first is accepted by the user. The second edge, dropped from the vertex that should have been deleted, is rejected by the user.

In addition to providing hypothesis verification, the user may interactively create and modify the wire frames, and control the system's focus of attention. The interactive system is helpful in designing and debugging object completion algorithms, by observing real cases and generating relevant test cases. Watch-

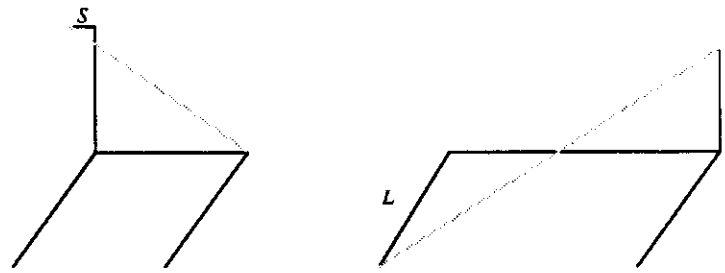


Fig. 9. Illegal "butterfly" buildings: (a) caused by "spur" (*S*) on horizontal edge; (b) caused by vertical line (*L*) mislabeled as horizontal.

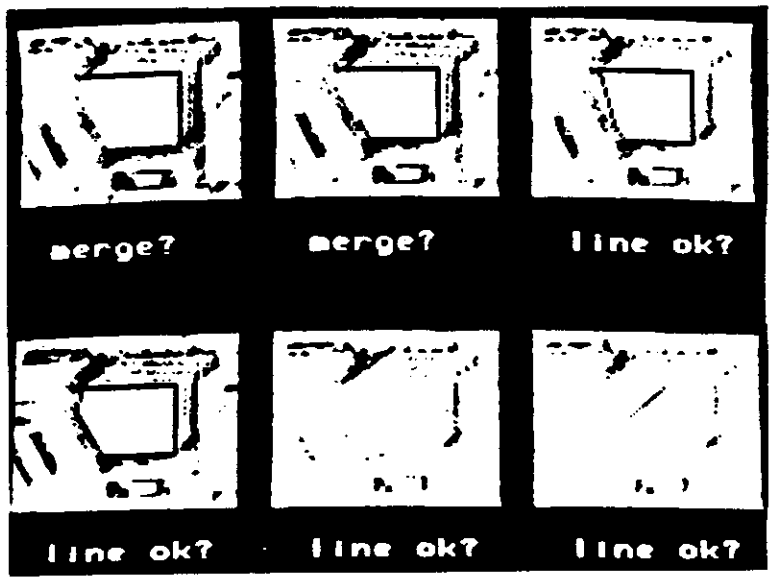


Fig. 10. Interactive object completion.

ing the system operate gives insight on domain assumptions which should be added to the system. These experiments have also shown the power of hypothesis verification, and a need for error correction when domain constraints are violated (such as the butterfly buildings in Fig. 9). The results of experiments with the interactive system are being used in designing the next generation of the 3D MOSAIC system.

4. Enhancing the System with Explicit Domain Knowledge

Much of the success of the 3D MOSAIC system can be attributed to its use of specific domain knowledge throughout its processing. For example, in monocular analysis, the knowledge that buildings have primarily horizontal and vertical faces is used, and during face completion, the information that many faces of buildings are parallelograms is used. Much of the geometric reasoning in the system exploits geometric constraints derived from the domain knowledge. However, all domain knowledge in the 3D MOSAIC system is represented implicitly in its interpretation and reasoning procedures.

The domain knowledge can be exploited more fully by combining an explicit representation of the domain constraints with geometric and symbolic reasoning. One system that has had success with such an approach is ACRONYM [2]. For example, consider the knowledge that walls are vertical rectangular faces. In the 3D MOSAIC system, this knowledge is used to derive a default method of completing faces, and contributes to the decision to label all edges as horizon-

tal or vertical when generating wire frames. If the system had an explicit model of a generic wall that would match any vertical rectangular face, this knowledge could also be used to alleviate the error caused by vertical edges of unequal length in Fig. 5. Although the initial wire frame in this example would have one leg either above or below the ground, after fitting the wire frame to the model, both legs would be on the ground. The butterfly buildings in Fig. 9 would not match a generic building model, causing some recovery action to take place.

Using explicit models, it would be easy to add new constraints to the system, such as a maximum height for buildings, or a minimum distance between them. Other relationships between buildings could be added, for example, buildings are generally aligned with each other and with roads. The relationships would be represented so that they might be used both for prediction and for verification. For example, the parallel relationship would take four arguments: two lines, the angle between them, and the distance. Any subset of the arguments could be specified, and the parallel relationship would fill in the rest. To verify that two lines are indeed parallel, the two lines would be specified, and the angle between them would be restricted. The parallel relationship would then determine whether this restriction is satisfied. To find a line parallel to a given one, only one line would be specified, but both the angle and distance would be restricted. The parallel relationship would provide a second line parallel to the first, satisfying the angle and distance restrictions.

In addition to knowledge about objects in the domain and relationships between them, the camera model and projection could also be explicitly represented. Explicitly representing both camera and object knowledge would facilitate the verification of object hypotheses in the image. For a hypothesized structure, the camera and object knowledge could be combined to derive a prediction of the structure's appearance, which could then be verified in the image. The predictions would be improved by including surface appearance information with the geometric information in the model. Appearance information would propagate through the structure graph in a similar manner to geometric information. Knowledge about the 3D structure and appearance of an object in a similar manner to geometric information. Knowledge about the 3D structure and appearance of an object could also be used to choose an appropriate image operator, such as a specific edge detector, to extract the low-level features for verifying that object. Scene knowledge might even be used to determine an initial strategy to extract useful 2D features from the image.

The next generation of the 3D MOSAIC system will explicitly represent and use domain knowledge including the camera model and projection, the shape and surface properties of scene objects, and multi-way relationships between scene objects. This system will be built on top of the 3D FORM system [9], which uses frames to represent geometric objects and the relationships between

them. By evaluating the relationships, the 3D FORM system does object completion in a more general way than the heuristics implemented in 3D MOSAIC allow. This allows a vision system using 3D FORM to be more flexible, organizing its reasoning to take advantage of the available information, while the 3D MOSAIC system must always operate according to its built-in heuristics. In addition, the frame representation is extensible to allow representations of surface properties of scene objects, camera models, and transformations between image and scene features. Finally, the frame representation allows easy extension to different building shapes, while this is difficult in the 3D MOSAIC system since knowledge of new building shapes would have to be embedded within the current procedures. Using 3D FORM, the processing in the current 3D MOSAIC system will be augmented with hypothesis verification and model-based predictions. The domain knowledge will be strengthened by adding appearance information, more objects, and more relationships between objects. The result should be a more robust, flexible, and extensible vision system.

5. Conclusion

This paper has described a method of generating 3D descriptions of a scene from images, using domain knowledge and geometric reasoning. Examples have been chosen from the 3D MOSAIC system that illustrate both the strengths and weaknesses of the current procedures. Finally, we have argued that the robustness, flexibility, and extensibility of the system can be improved by representing the domain knowledge explicitly, using the knowledge for prediction and verification, and incorporating geometric reasoning. We are developing a frame-based system to achieve this goal.

ACKNOWLEDGMENT

Takeo Kanade has provided excellent guidance and encouragement. The authors are also indebted to former members of the 3D MOSAIC project: Fumi Komura, Shigeru Kuroe, and Duane Williams.

REFERENCES

1. Barnard, S.T., Interpreting perspective images, *Artificial Intelligence* 21 (4) (1983) 435–462.
2. Brooks, R.A., Symbolic reasoning among 3-D models and 2-D images, *Artificial Intelligence* 17 (1981) 285–348; Special volume on computer vision.
3. Herman, M., Monocular reconstruction of a complex urban scene in the 3D MOSAIC system, in: *Proceedings ARPA Image Understanding Workshop* (1983) 318–326.
4. Herman, M. and Kanade, T., Incremental reconstruction of 3-D scenes from multiple, complex images, *Artificial Intelligence* 30 (1986) 289–341.
5. Herman, M., Representation and incremental construction of a three-dimensional scene model, in: A. Rosenfeld (Ed.), *Techniques for 3-D Machine Perception* (Elsevier Science Publishers, Amsterdam, 1986) 149–183, also: Carnegie-Mellon University Tech. Rept. CMU-CS-85-103, Pittsburgh, PA (1985).

6. Kender, J.R., Environmental labelings in low-level image understanding, in: *Proceedings IJCAI-83*, Karlsruhe, F.R.G. (1983) 1104–1107.
7. Lowe, D.G. and Binford, T., The interpretation of three-dimensional structure from image curves, in: *Proceedings IJCAI-81*, Vancouver, BC (1981).
8. Ohta, Y., *Knowledge-Based Interpretation of Outdoor Color Scenes* (Morgan Kaufmann, Palo Alto, CA, 1985).
9. Walker, E.L., Herman, M. and Kanade, T., A framework for representing and reasoning about three-dimensional objects for vision, *AI Mag.* **9** (2) (1988) 47–58.

Received October 1986; revised version received December 1987