

Detecting Buildings in Aerial Images*

A. HUERTAS AND R. NEVATIA

*Institute for Robotics and Intelligent Systems, University of Southern California,
Powell Hall Room 234, Los Angeles, California 90089-0273*

Detecting building structures in aerial images is a task of importance for many applications. Low-level segmentation rarely gives a complete outline of the desired structures. We use a generic model of the shapes of the structures we are looking for — that they are rectangular or composed of rectangular components. We also use shadows cast by buildings to confirm their presence and to estimate their height. Our techniques have been tested on images with density typical of suburban areas. © 1988 Academic Press, Inc.

INTRODUCTION

Making maps automatically from aerial images is a task of great importance for many applications. Currently, the major difficulty in automation is the extraction and description of cultural (man-made features and objects, such as buildings and transportation networks). In this paper, we describe a technique for extraction of certain types of buildings under certain conditions. We do not claim to be able to detect any raised structure in an arbitrarily complex scenes, but do feel that our method is a major step and adds significantly to the state of the art. More generally, our method shows an example of how generic model knowledge can be used to extract objects in real, outdoor scenes.

We can identify three dimensions of complexity of the proposed task:

1. *Density* of the structures in the scene. A rural scene has low density, an urban scene has high density and a suburban scene is in between (medium density).
2. *Shape of the structures*. Buildings come in many shapes from simple rectangular blocks with a flat roof to complex shapes with intricate, multi-based roof structures.
3. *Image quality*. Images vary in terms of contrast, resolution, and visibility.

Our method is applicable to scenes of medium density with relatively good image quality. Further, we assume that the buildings are rectangular or composed of rectangular components (thus “box,” “T,” “L,” and “E” shapes, for example, are allowed). Figure 1, an aerial image of the Los Angeles International Airport (LAX) (512 × 512 resolution), is typical of the domain for our system. Note that this image has buildings of varying complexity and size.

One of the major causes of difficulties in detecting building in scenes like Fig. 1 is that the lower level segmentation processes rarely give complete and accurate results. We use the model of the desired structures to overcome the errors of the lower levels. Another major source of difficulty is the lack of direct 3D data in a single 2D image. Using a stereo pair of images is one option, here we use shadows to

*This research was supported by the Defense Advanced Research Projects Agency, monitored by the Air Force Wright Aeronautical Laboratories under Contract F33615-84-K-1404, and by the Defense Mapping Agency under contract DMA800-85-C-0008

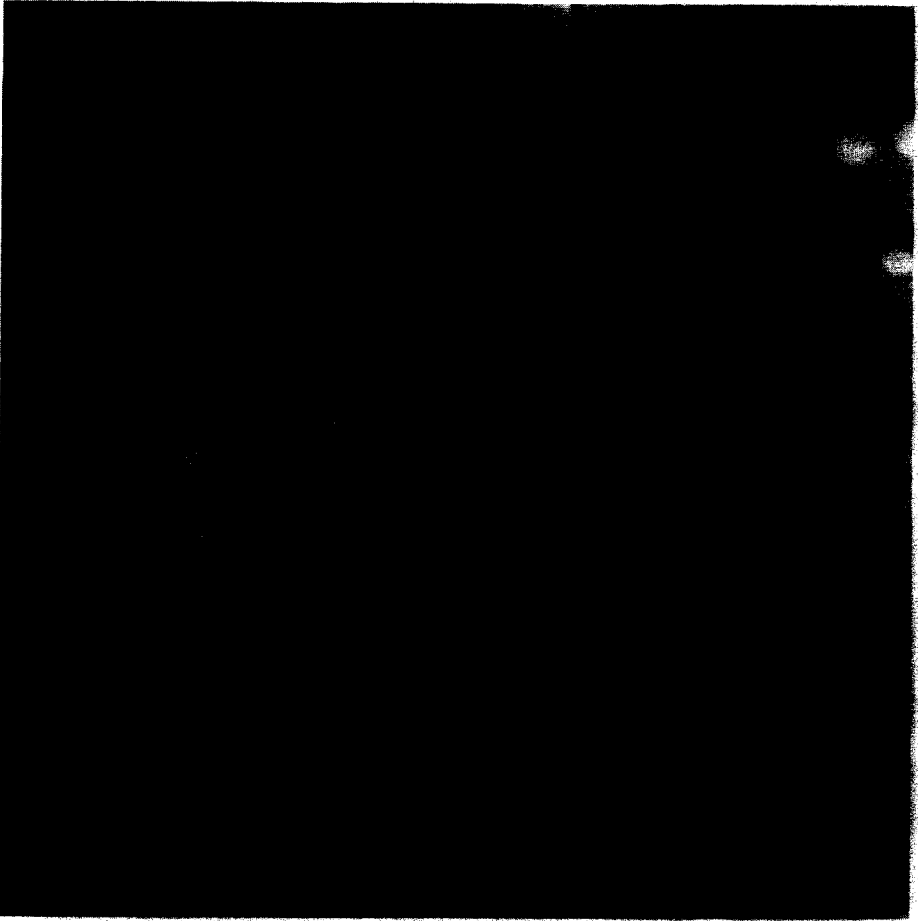


FIG. 1. LAX image (left view).

disambiguate raised structures from structures on the ground (such as parking lots) and also to measure the height of the structures where possible. Shadows have often been viewed as a problem in scene analysis, here we use them to our advantage in the analysis. We know of no other previous similar use of shadow analysis for real images (Shafer presents a very nice geometrical analysis of use of shadows in [1], however it applies to idealized cases only).

2. AN OVERVIEW AND ASSUMPTIONS

In this work we have chosen to work primarily with the line segments corresponding to the intensity edges in the scene. A building of interest is modeled as the union of one or more rectangular parallelepipeds in 3D. When viewed from above, our 3D model becomes a 2D model representing the union of one or more rectangular components. The problem then becomes one of finding sets of line segments that form the outline of a shape whose sides form near 90° L-junctions.

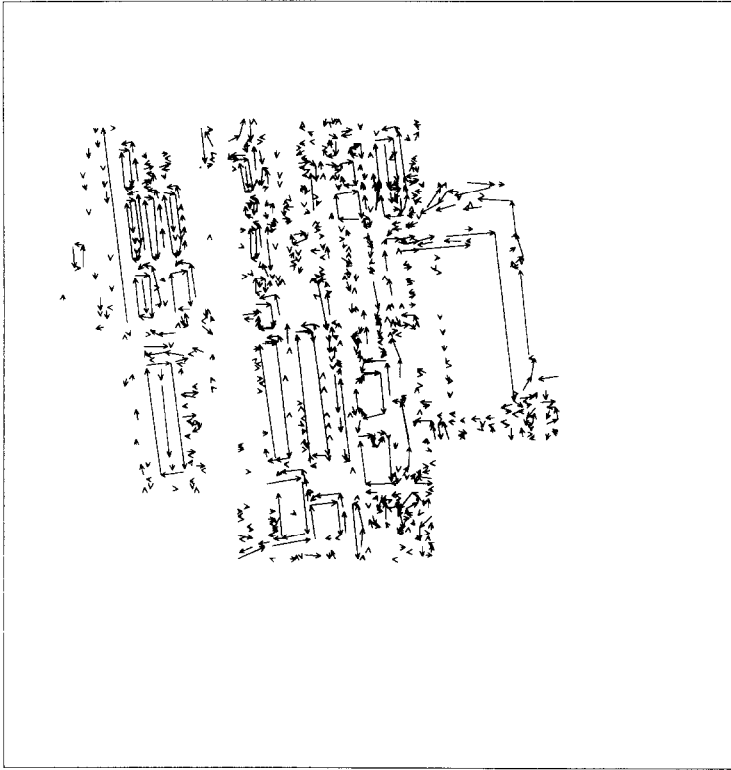


FIG. 2. Line segments from LAX image in Fig. 1.

The problem of finding such buildings is difficult, even in not very high density scenes, because the detected line segments may be highly fragmented. The fragmentation is not necessarily due to inadequacies of the line extraction process, though there is certainly room for improvement there. Many sources of problems are due to real structures in the image. For example, sides of buildings may have trees, vehicles, or road markings nearby. Roofs may have detailed super structures, and the contrast for some sides may be very low. To use the fragmented segments effectively, we utilize our knowledge of the expected building shapes and use shadows to confirm our hypotheses. (For example, Fig. 2 shows the lines detected in the LAX image by our method [2].)

We can often distinguish between 2D objects and 3D objects by analyzing shadows. A box representing a building, as opposed to a rectangular parking lot, for example, must cast a shadow, on sunny days, which is consistent with the building shape and with the direction of illumination. Thus, some of the observed line segments forming L-junctions along the buildings physical boundaries must have a corresponding shadow along the direction of illumination. If we are able to match corresponding building and shadow features then we can have increased confidence in our object hypothesis and also obtain an estimate of the building height and construct a 3D model of the building. We assume that the sun angles, relative to the image, are known a priori, and that the camera principal ray points to the center of the scene.

Our building model also assumes that the visible building surfaces consist of smooth regions and that building sides consist mostly of vertical structures. We do not perform detailed roof structure at this stage; our limited shadow analysis techniques assume that simple buildings have a nearly flat roof and that shadows are cast on level ground.

Our method consists of the following steps:

1. *Detect lines and corners.* We apply one of our edge detection techniques [2, 3] to obtain the line *segments* corresponding to the intensity edges in the image. Next we find all pairs of line segments forming *corners*. A corner is defined as a near orthogonal L-junction.

2. *Label corners based on shadows.* We assign an initial interpretation to the corners as object or shadow corners, on the basis of constraints imposed by the direction of illumination. We also look for correspondence between "bright" corners (presumably belonging to objects) and "dark" corners (presumably belonging to shadows) in the known direction of illumination. If these matches exist, we say that the matching corners are in "shadow correspondence" and make a record of these matches.

3. *Trace object boundaries.* Next we track the boundaries of those objects that satisfy our shape constraints as follows:

- (a) Corners which share a segment are called "strongly" compatible. We group these corners to form *chains* of corners representing continuous (complete or partial) boundaries of an object of interest. In some cases, this will lead to closed figures. A closed figure is called a *box*.
- (b) The shape of the chains of corners is then examined to check if it is consistent with our model (a composition of rectangles).
- (c) Most often, closed shapes are not found. In this case, we look for evidence of boundary continuation (segments and corners) near the ends of corner chains with looser tolerances. As before, in some cases, this will lead to slightly distorted closed figures.
- (d) In some cases, we find sets of chains that are not connected but are in proper positions and orientation to form a rectangular structure. If certain criteria are met, these chains are joined to form a box.
- (e) Finally we examine the open figures remaining. We analyze the missing or undetected features, and close the boundaries if appropriate. For example, in some cases, we are able to detect only two sides of an object; however, if the corner formed by these two sides has a matching shadow corner along the direction of illumination, that is, they are in shadow correspondence, then we can infer that we have a structure above ground. In cases like these, we simply close the "box" by adding the remaining two lines, parallel to the detected lines.

4. *Verify hypotheses.* Shadow information is used to verify an object boundary as a building. In cases where a closed 2D boundary has been found, the shadow correspondences serve to "verify" our hypothesis of having found a 3D structure and also give us the relative height of the structure. Verified objects are then described in terms of components, position, and size.

3. DETAILS OF THE METHOD

3.1. Line Detection

We use the USC “LINEAR” line detection system to obtain line *segments*. Edge detection is performed by applying either the Nevatia–Babu [2] or the Marr–Hildreth [4] edge detectors to the image. Edges are then thinned and linked to form continuous curves. These curves are approximated by piecewise linear segments. Each linear segment is described by its length, orientation, contrast, and position of its end points. Additionally we also know if a segment connects to another segment at either end.

3.2. Corner Detection

We find all pairs of line segments that form *corners* consisting of near orthogonal L-junctions. Some valid corners are shown in Fig. 3. If the average intensity of the region in the concave side of a corner is higher than the average intensity of the region on the convex side the corner is called a *bright* corner, otherwise the corner is called a *dark* corner. The set of detected corners is the strongest initial set of evidence as to the presence of objects having rectangular shapes in the image.

In most cases, edge detectors have inherent deficiencies at corners. As a result very few corners will be approximated by line segments in the configuration shown in Fig. 3a. Our corner detector processes corners allowing for the inherent imperfection of the lines according to the following criteria:

- Segments must have a minimum length (5 pixels), and the end points of the corner elements (line segments) must be within 5 pixels, as shown in Fig. 3b.
- The angle between the corner segments must be $90^\circ \pm \epsilon$, where ϵ is a function of the length of the shorter segment and the shape of the corner (ratio of the lengths of the two segments). The minimum tolerance is 5° for longer segments and corners having segments of similar lengths. The largest tolerance is 35° . (In general, ϵ should also be a function of the position in the image, but is not in the current implementation.)
- We allow for “rounded” corners as shown in Figs. 3c and d.

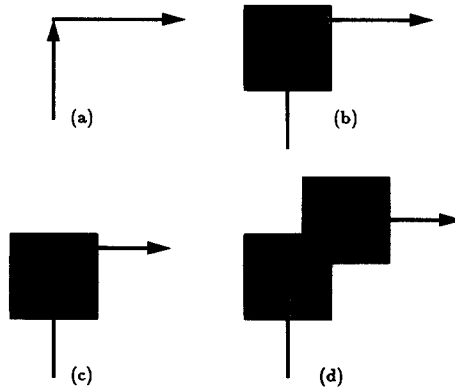


FIG. 3. Corner detection.

The search process is implemented by first examining connected segments as in Fig. 3a to see if they form a corner. Otherwise by searching in a 5×5 pixels window at the end point of one segment, for the begin point of another segment that forms a valid configuration. The search proceeds in a spiral fashion starting at the end point of a segment, and terminates when a corner is found. As shown in Fig. 4, a counterclockwise spiral search beginning at the end point of a segment will find first a segment whose end point is closest. It also favors the formation of bright corners, presumably belonging to buildings, over dark corners.

In Fig. 3c a short segment having a length smaller than the 5×5 search window allows a second segment to be detected within the window. In Fig. 3d, the length of the "middle" segment exceeds the window size but is shorter than the minimum allowed length for corner components. In this case the window is shifted to the end point of the middle segment.

In some cases, one of the corner segments extends beyond the other segment component to form a T-junction rather than an L-junction. In general T-junctions signify occlusion, where the top of the T belongs to the occluding feature and the stem of the T belongs to the occluded feature. More particular to our task of detecting buildings, we have observed that some tops of T-junctions are actually overextended corner components, resulting from coincidental alignments of features that are rather common around buildings. Shadows, roads, sidewalks, surface markings, changes in ground surface materials, access ramps, driveways, and landscaping are some of the examples of features that can be aligned with building sides. These kinds of alignment are likely to result in long line segments that in effect change from one interpretation to the other at the junction point. Also, in our future work, we will find that more complex buildings composed of "wings" that may have different heights will give rise to T-junction configurations. In this case, the top of the T has a consistent interpretation all along a building side but the height of the portions on both sides of the junction point may be different.

In order to avoid mistracing the boundaries of a building, we perform some early processing on the T-junctions that satisfy some criteria. In effect, we break the top (segment) of the T in two parts (segments) if:

- The stem of the T is longer than a minimum distance (5 pixels), and the top of the T is within a short distance (5 pixels) of the end of the stem, as shown in Fig. 5a.
- The angle between the top and the stem satisfies the criteria for corners.
- The length of the resulting parts is longer than a minimum length (5 pixels).
- The stem of the T is not connected to another segment, as shown in Fig. 5b.

As expected, breaking the tops of T-junctions results in additional corners. Figure 6 shows the (segments forming) corners extracted from the segments shown in Fig. 2.

3.3. Corner Descriptions

3.3.1. Interpretation of Corners Based on Shadows

We use shadows to assign an initial interpretation to the detected corners. We first determine from the image the average gray level of the shadow regions. This

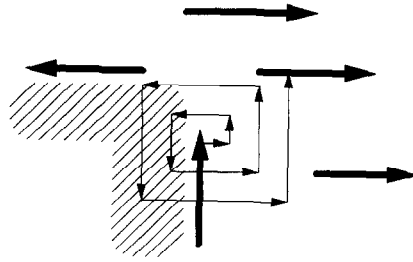


FIG. 4. Spiral search.

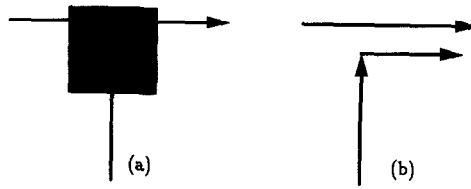


FIG. 5. Junctions.

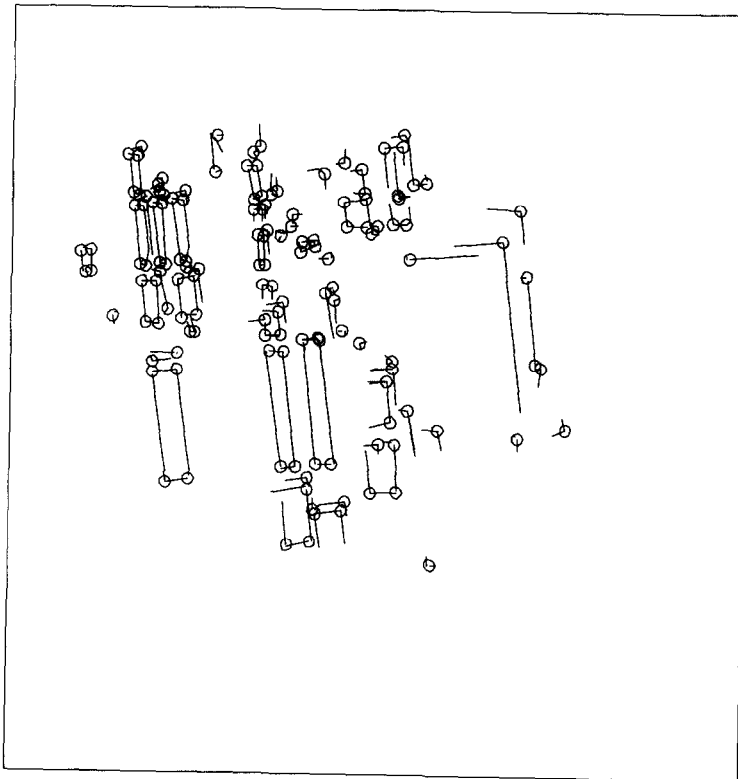


FIG. 6. Corners found from segments in Fig. 2.

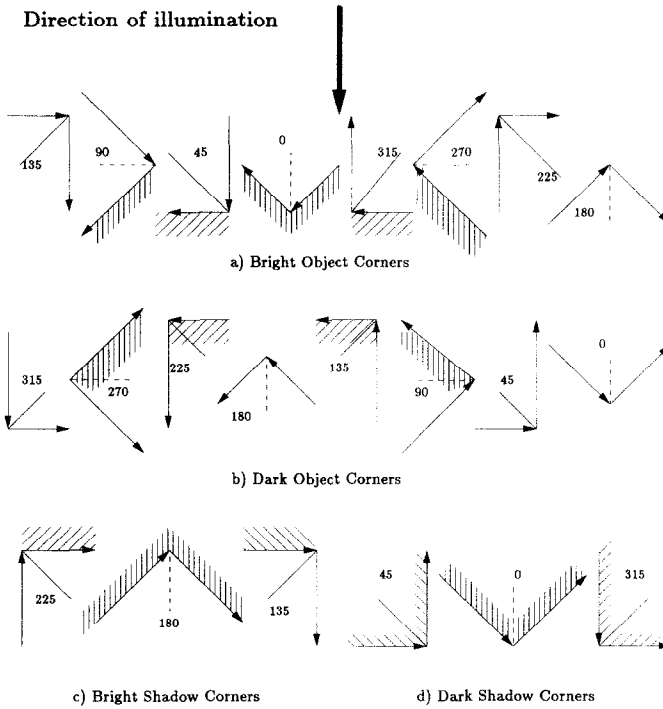


FIG. 7. Corner interpretation. Corner bisectors shown dashed. The angle shown is between the corner bisector and the projected direction of illumination.

level is selected manually from the histogram of the gray levels of the smooth regions in the image. (The edge “regions” are masked out by locating the pixels with high convolution output.)

Next we label the bright and dark corners as *object* or *shadow* as a function of the direction of illumination, the corner position and attitude, and the average gray levels of the regions on both sides of the corner segments.

Figure 7 summarizes the labeling combinations. The direction of illumination is shown as a thick arrow in the top part of the figure and is fixed for all the corners shown in the rest of the figure. The corner attitude with respect to the direction of illumination is given by the orientation of the corner bysectors (dashed lines). The

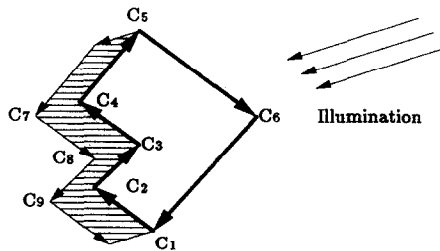


FIG. 8. Example.

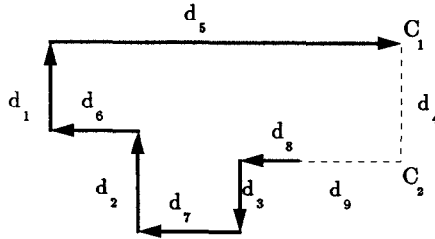


FIG. 9. Hypothetical chain.

shaded areas represent regions sufficiently dark to be part of a shadow. Recall that the arrows at the end of a segment indicate that the region to the right is brighter. In Fig. 7a, for example, the 0° bright corner must have a shadow region adjacent to both corner segments to be labeled an “object” corner. The 0° dark corner in Fig. 7b, on the other hand, must *not* have a shadow region adjacent to either of its two segments, to be labeled an “object” corner. Similarly, Fig. 7c and d show the labeling for dark corners as “object” or “shadow.”

3.3.2. Shadow Correspondence

Not all corners labeled “object” and that have a shadow region adjacent to it, are expected to have a corresponding shadow corner due to poor shadow definition at the corner. In these cases we look for object-to-shadow corresponding *lines* at a later stage. However, in some cases it is possible to find corresponding *corners* at this stage. The criteria for a match is as follows:

- The two corners must have a different color. A bright (dark) corner that belongs to an object must be closer to the source of illumination than a dark (bright) corner that belongs to a shadow.
- The corner bisectors must be collinear within a maximum tolerance of 20° for closer corners.
- In case of a multiple match, the closer corner is chosen.
- If a match is found, the distance between corresponding corners is recorded to generate 3D descriptions as discussed later.

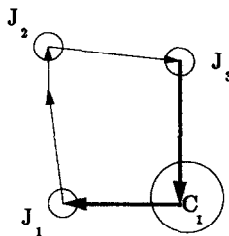


FIG. 10. Trace of continuous boundaries.

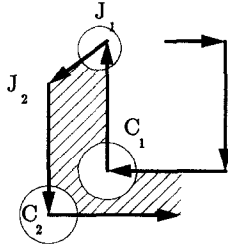


FIG. 11. Backtracking.

3.3.3. Corner Compatibility

We compute "compatibility" between corners indicating whether they belong to the same object as follows:

Based on continuity. If two corners share a line segment then they are said to be *strongly compatible*. In most cases, these two corners are very likely to be part of the outline of the same object in the image. Thus, the transitivity of the strong compatibility relationships is useful to group corners whose segments are continuous.

Based on attitude. If the bisectors of two corners are either parallel or orthogonal and the corners are located within a maximum distance (currently set to 100

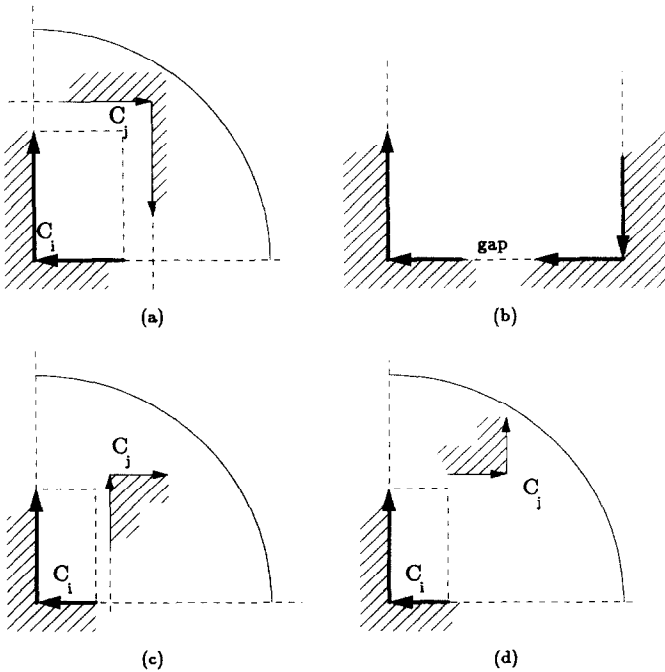


FIG. 12. Attitudes of interest between weakly compatible corners.

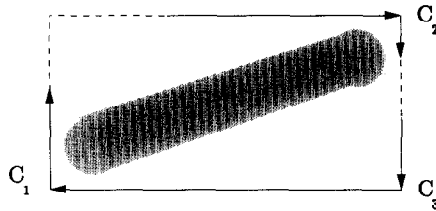


FIG. 13. Combination of disjoint chains.

pixels), then the two corners are said to be *weakly* compatible. Weak compatibility is therefore an n -to- m relationship with both n and m larger or equal to 1. As explained in the next subsection, we apply positional constraints to weakly compatible corners to try to join groups of strongly compatible corners corresponding to broken and fragmented boundaries.

Figure 8 shows some examples of the definitions given above. Corner C_3 is a dark object corner, Corner C_8 is a bright shadow corner. Corners C_7 and C_8 are dark shadow corners. Corners C_1 and C_2 are strongly compatible. Corners C_1 and C_3 , and corners C_1 and C_5 are weakly compatible. Corners C_4 and C_7 are in shadow correspondence.

3.4. Tracking Boundaries

a. Form Chains of "Strongly" Compatible Corners

We use the transitivity of strong compatibility to group corners that share segments into *chains*, representing the continuous boundaries of objects. Figure 9 shows schematically one such chain. We do not explicitly check that the labels assigned initially to the corners have a consistent interpretation along the chain until a later stage. At this point we give more importance to boundary continuity.

b. Test for Shape

Next we apply the following two shape measures to determine whether the elements in the chain geometrically fit our model. These measures also determine whether a chain forms a complete (closed) figure or not. If it does not, they give an

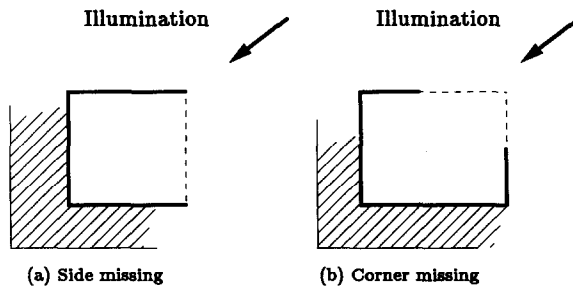


FIG. 14. Hypothesis of missing features.

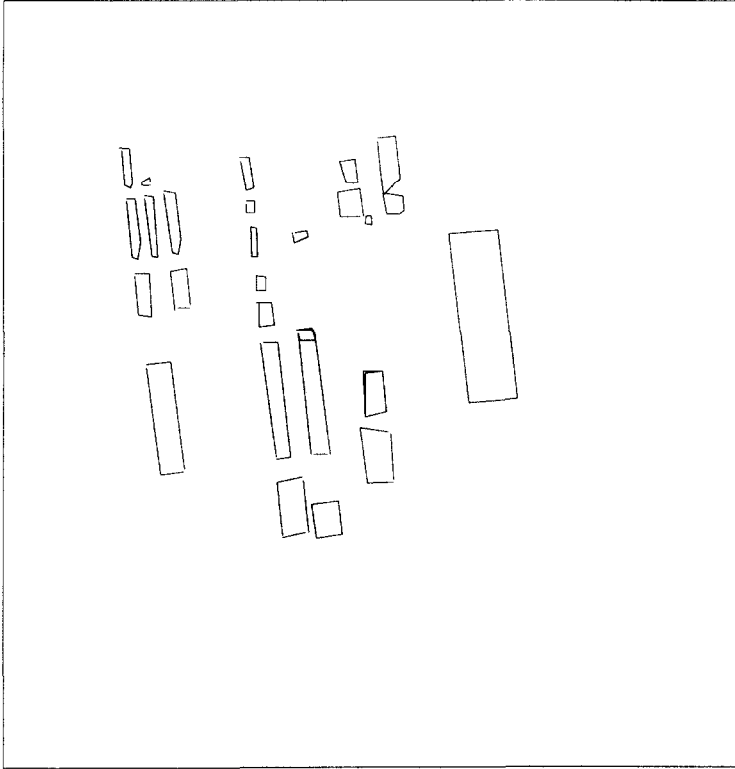


FIG. 15. Closed boundaries (boxes) in LAX image.

indication of the shape and size of the features (corners and segments) needed to have a closed figure.

1. *Corner consistency (CC)*, defined as the number of bright corners minus the number of dark corners along the boundary of a region. CC is always equal to $+4$ if the surrounded region is brighter than its background, or equal to -4 if the surrounded region is darker than its background. This is due to the fact that the simplest regular shape of interest, the rectangle, has four corners (all bright or all

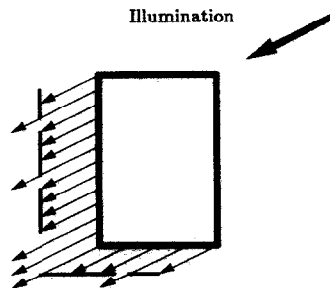


FIG. 16. Locating shadow boundaries.

dark). For each additional corner (pair of sides added) to this basic shape to form more complex shapes, we must add one bright plus one dark corner, which cancel each other in the summation. In the example of Fig. 9 there are 4 bright corners and 2 dark corners in the chain. The sum gives +2, indicating that a minimum of 2 additional bright corners (C_1 and C_2) are needed bring the summation to +4 and have a closed outline.

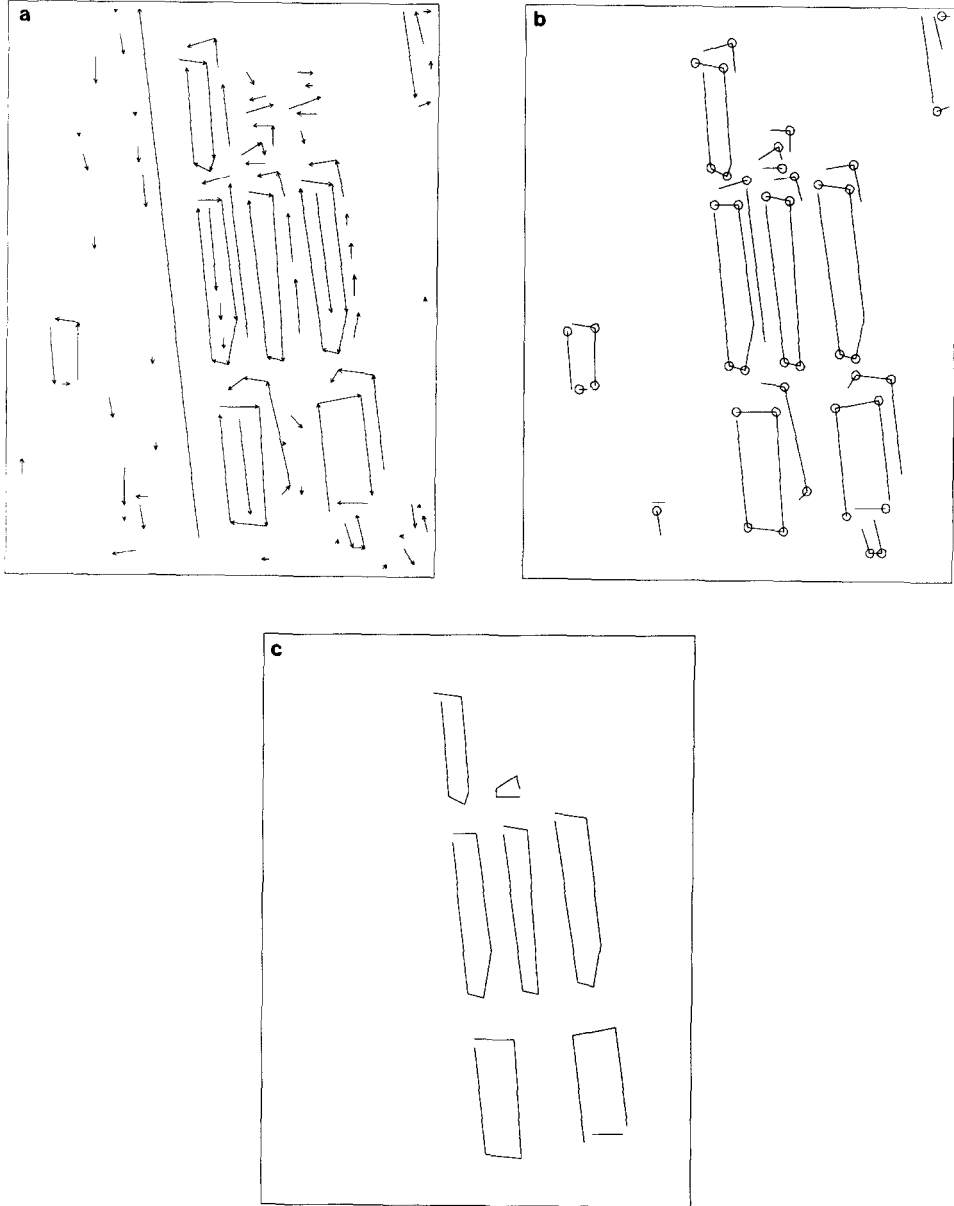


FIG. 17. (a) Line segments (upper left of LAX image); (b) corner segments from line segments in (a); (c) verified boxes (upper left of LAX image).

2. *Side consistency (SC)*, defined as the balance on the lengths of the object sides. Our objects of interest have rectangular shapes, thus the opposite sides of their minimum bounding rectangle must have the same lengths. If the object outline is not closed we can determine the length of the missing portions. In the example of Fig. 9, the segments \mathbf{d}_4 and \mathbf{d}_9 are needed to have a closed outline, where $\mathbf{d}_4 = \mathbf{d}_1 + \mathbf{d}_2 - \mathbf{d}_3$ in the vertical and $\mathbf{d}_9 = \mathbf{d}_5 - \mathbf{d}_6 - \mathbf{d}_7 - \mathbf{d}_8$ in the horizontal direction.

At this stage we mark the closed outlines as complete closed boxes. These are defined as those chains that satisfy our shape constraints, allowing for small gaps.

c. Track Continuous Boundaries

As discussed above, the CC and SC measures give an estimate of the segments and corners needed to have closed boundaries. In many cases some of these segments and corners are present in the segments set but not taken into account initially due to distortions exceeding the initial angle tolerance (ϵ) set for corner detection. We add these portions at the ends of chains by relaxing the angle tolerances to 2ϵ at junctions and by allowing for distortions along otherwise straight outline segments. In the process we update the CC and the SC measures and we test for closedness, marking those outlines that become closed. Figure 10 shows an

| | | | |
|-------------------|----------|-------------------|----------|
| Building | 1 | Building | 5 |
| Boundary Segments | 5 | Boundary Segments | 4 |
| Perimeter | 63.1 | Perimeter | 76.3 |
| Height | 4 | Height | 4 |
| Area | 184 | Area | 308 |
| Centroid | (28,49) | Centroid | (113,61) |
| Volume | 736 | Volume | 1232 |
| Orientation | 5° | Orientation | 4° |
| Building | 2 | Building | 6 |
| Boundary Segments | 5 | Boundary Segments | 4 |
| Perimeter | 95.1 | Perimeter | 76.3 |
| Height | 4 | Height | 3 |
| Area | 341 | Area | 315 |
| Centroid | (64,82) | Centroid | (110,87) |
| Volume | 1364 | Volume | 945 |
| Orientation | 189° | Orientation | 187° |
| Building | 3 | Building | 7 |
| Boundary Segments | 4 | Boundary Segments | 4 |
| Perimeter | 90.6 | Perimeter | 18.9 |
| Height | 4 | Height | 1 |
| Area | 240 | Area | 30 |
| Centroid | (66,67) | Centroid | (38,63) |
| Volume | 960 | Volume | 30 |
| Orientation | 4° | Orientation | 121° |
| Building | 4 | | |
| Boundary Segments | 5 | | |
| Perimeter | 89.8 | | |
| Height | 4 | | |
| Area | 293 | | |
| Centroid | (68,54) | | |
| Volume | 1172 | | |
| Orientation | 188° | | |

FIG. 18. Descriptions of seven buildings.

example of a closed outline for which only one corner (C_1) is initially detected. The remaining distorted junctions (J_1, J_2 , and J_3), are detected by looking for evidence of continuation at the ends of the chain formed by C_1 .

Adding segments and corners with a larger tolerance at the ends of an open chain may lead to the type of error shown in Fig. 11. This is detected by noting that a corner (C_1), labeled "shadow" was reached while tracing an object boundary. In this case the process backtracks to a "safe" point, currently defined as the last visited junction (J_1) in the chain where there was a change of "color," for instance, from bright to dark, in the corners in the chain.

d. Combine Disjoint Chains

We next determine whether two chains that remain open, but that contain weakly compatible corners, may be joined to form a longer chain. If these corners are in

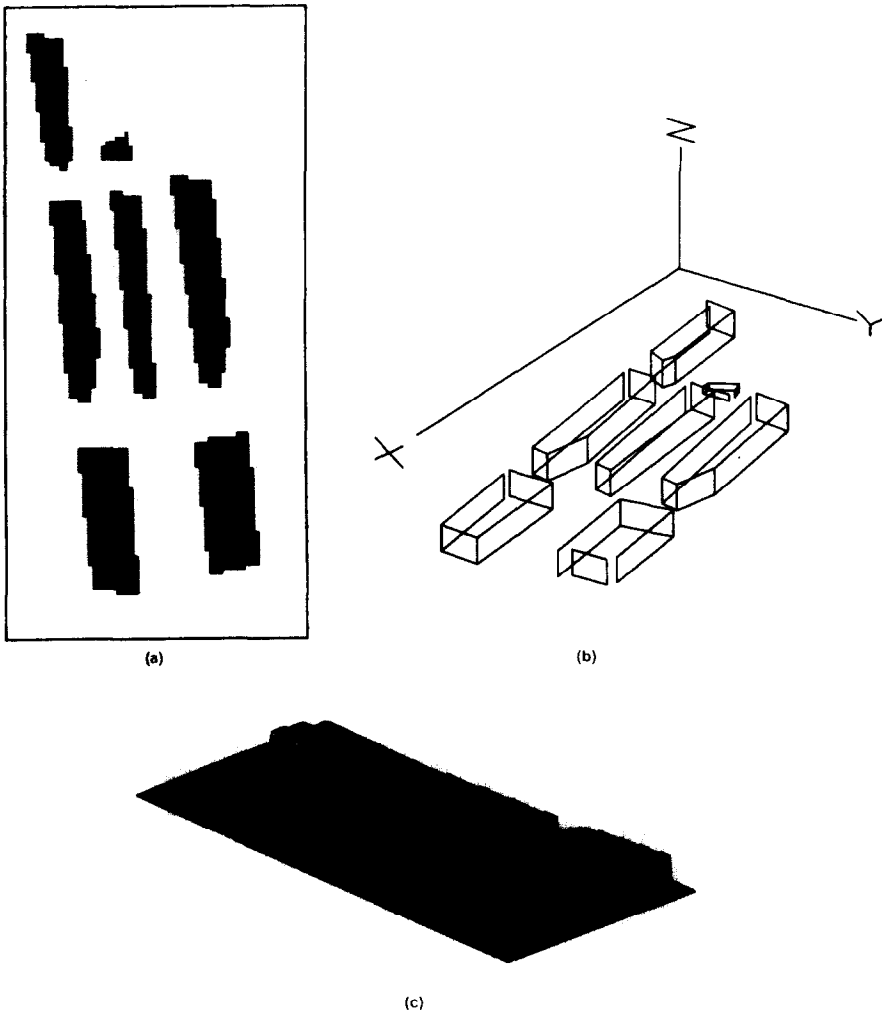


FIG. 19. Three views of seven buildings.

one of the four attitudes shown in Fig. 12 (see details below), then we apply positional constraints to determine whether the two chains are considered candidates for joining.

Figures 12a and b show a pair of bright weakly compatible corners belonging to two disjoint chains. In Figs. 12c and d the two corners have opposite color. These latter ones are useful when processing objects with E, T, L, and U shapes.

Let C_i be a bright corner in Figs. 12a, c, and d. For each C_i in one chain we look for a corner C_j in another chain until we find a candidate chain for joining. The position, size, and attitude of C_i determines the search window where we look for C_j ; the extension of the segments in C_i determine two of the boundaries of the search window. The remaining boundary is given by a circular arc centered at C_i and of radius r . Since weakly compatible corners are located within a maximum distance currently set to 100 pixels then r is also 100 pixels. The components of C_j (segments) must be entirely contained in the search window. In addition they must not intersect the minimum rectangle bounding the segment components of corner

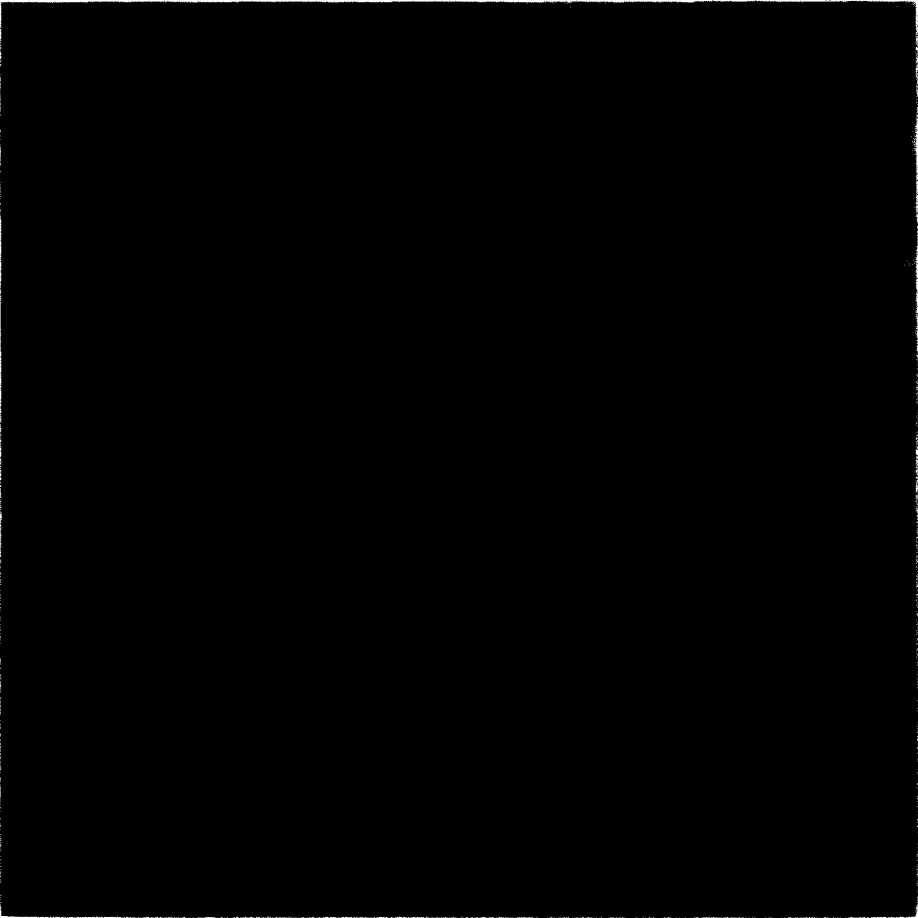


FIG. 20. Building boundaries overlaid on LAX image.

C_i . Among the possible matches we choose that one for which the distance between C_i and C_j is a minimum. In Fig. 12b, the gap between the two colinear segments must be shorter than the sum of the lengths of the colinear segments.

If a match is found then the region between the matching corners is tested for smoothness using the intensity data. A successful test results in the combination of the two chains which include the matching corners. In Fig. 13 corners C_1 and C_2 match. Also corners C_2 and C_3 match. If either match is successful the chains are joined (the corners in both chains now belong to one chain).

Since the chains (join candidates) are open chains, we examine the gaps between the corresponding ends of the combined chains. Either a portion of one side, or a corner is needed at the ends of the chains to have a closed box. If the gaps are small compared to the lengths of the chains, the missing segment or corner is hypothesized and the chains are then closed. The gaps should be smaller than one fifth of the sum of the lengths of the chains, up to 10 pixels. Chains with large gaps remain open.

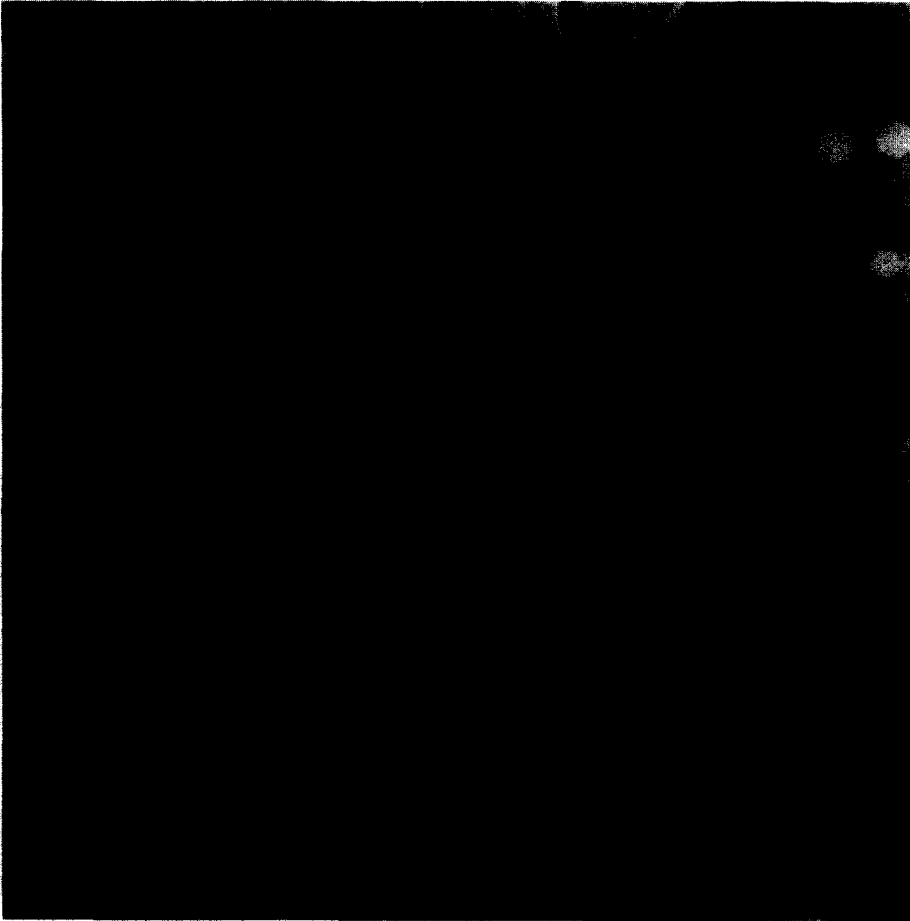


FIG. 21. LAX image (right view of stereo pair).

The CC and SC measures are updated and the resulting chain is tested for closedness as discussed above.

e. Missing or Undetected Features

In many cases it is difficult to extract complete building boundaries mainly due to low contrast or excessive segment fragmentation due to texture or small features adjacent to buildings. However, in general we can extract partial information on the side of the buildings that have a shadow region adjacent to it; the observed boundaries between the roof of a building and a shadow region usually have a high contrast and hence tend to be reliable features. If we have a record of shadow correspondence for at least one of the corners in the chain, we consider this sufficient evidence and proceed to reconstruct the missing elements to form a closed box, by adding a side or a corner (two sides) giving the smallest rectangular 3D object that would cast the observed shadow (see Fig. 14).

Figure 15 shows the line segments representing the closed boxes found in the LAX image shown in Fig. 1.

3.5. Verification and Descriptions

A closed outline satisfying our building model is considered verified if it contains a corner with shadow correspondence. Otherwise we explicitly search for a shadow

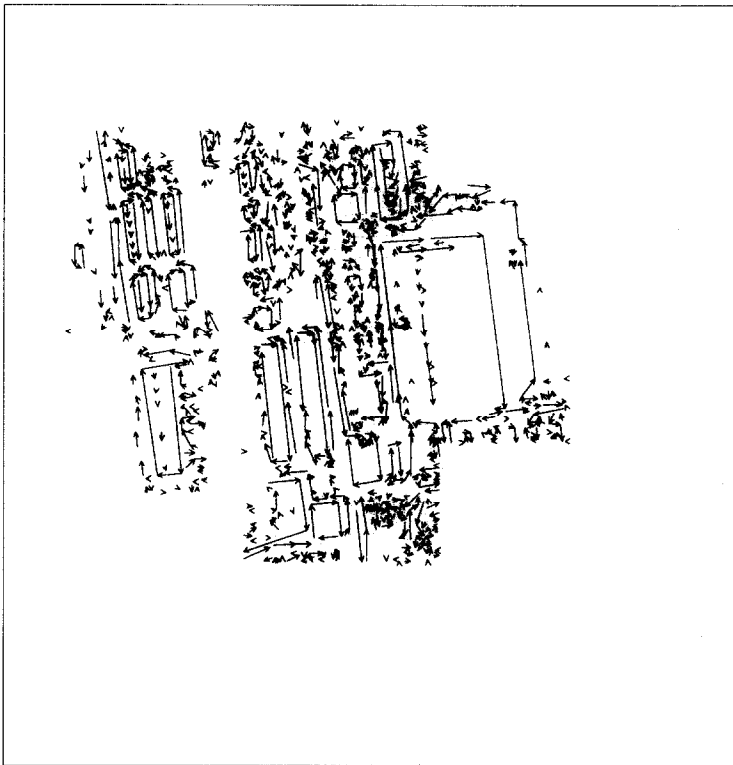


FIG. 22. Line segments from LAX image in Fig. 21.

boundary in the direction of illumination. The search is implemented by identifying those segments along the boundary that, in the case of a 3D object, would cast a corresponding shadow in the direction of illumination, as shown in Fig. 16. Beginning at points selected with regular intervals we search in the direction of illumination, pixel by pixel, for a segment representing a portion of the boundary of the shadow. At the same time we collect the gray level cumulative mean of the pixels in the intensity image. In order to prevent the search from proceeding beyond the shadow boundary if no corresponding segment is found due to fragmentation, we compare the cumulative mean to the average shadow gray level for the scene. We keep looking until the cumulative mean exceeds the average shadow gray level. The width of the shadow is used for height computation.

Figures 17a and b show, with a large scale, the line segments and corners, corresponding to the seven buildings in the upper left portion of the LAX image. Note that none of the building outlines is completely closed and most are quite distorted. Figure 17c shows the boxes detected and verified as possible buildings. Note that the rectangular feature on the left is not extracted as a possible building since the program is unable to verify the presence of a shadow. The other seven boxes, including a very small and quite distorted box, have corresponding shadow corners.

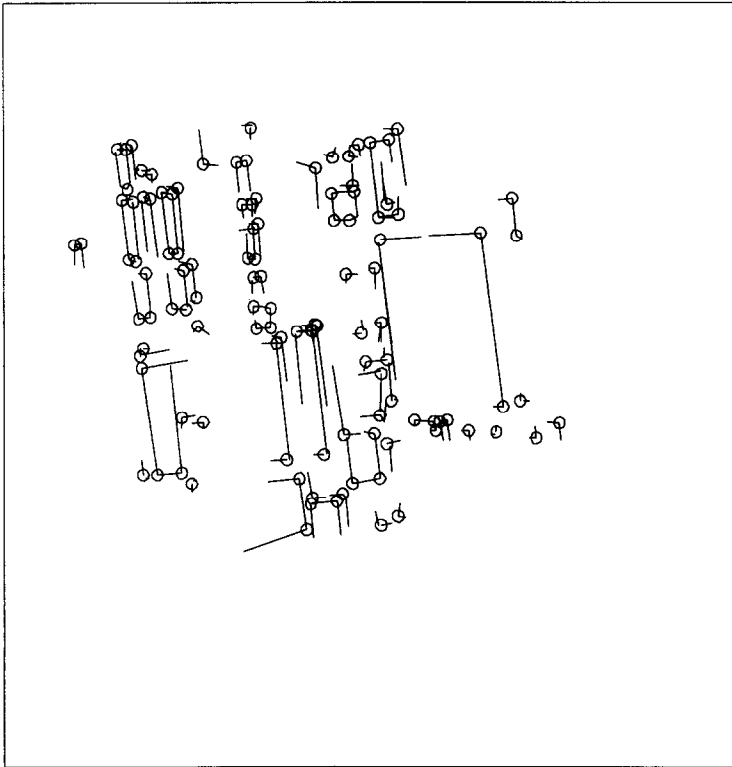


FIG. 23. Corner segments from line segments in Fig. 22.

Closed outlines are described as buildings by giving estimates of their perimeter, area, and height (obtained from the shadow correspondence). Also given are the building's orientation (the orientation of its longest segment), the coordinates of the centroid, the number of sides, and the number of corners.

The building descriptions for the seven boxes shown in Fig. 17c are given in Fig. 18. Figures 19a, b, and c show three other views of the buildings: the buildings viewed as regions, a 3D view of the buildings and ground surfaces, and a 3D wire-frame model of the buildings.

Figure 20 shows the detected building boundaries overlaid on the original LAX image.

4. MORE RESULTS AND CONCLUSIONS

Figure 21 shows the right view of the stereo pair of the LAX image (Fig. 1 is the left view). We have included two views to illustrate both the robustness of our method and the effect of the viewing angle on some important features. In this image many buildings have a simple shape, but in the context of a major airport, there are a multitude of objects adjacent to the buildings. Some of these include access ramps, rows of containers, stacks of containers and boxes, parked vehicles and trucks, etc. A small change in viewing angle can cause these objects to be more

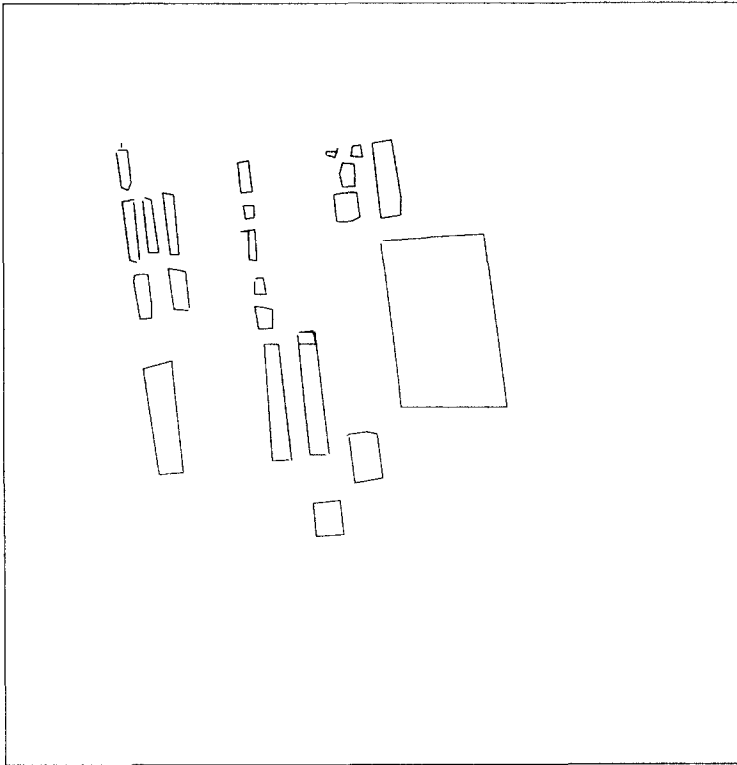


FIG. 24. Closed boundaries (boxes) for LAX image in Fig. 21.

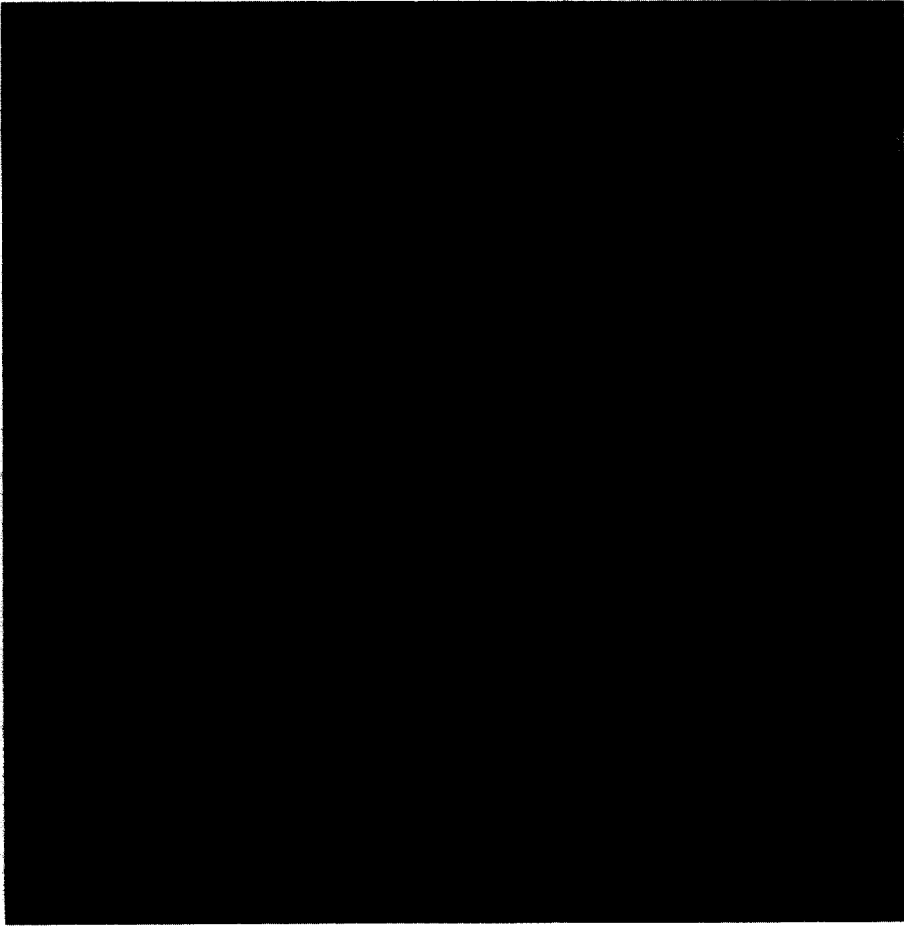


FIG. 25. Building boundaries overlaid on LAX image (right view).

or less visible. Also these objects are small compared to the buildings and most of the buildings are not tall and their shadows cast on these objects.

Five small windows in both, left and right, images were processed using the same parameters for feature extraction and box detection. We then combined the results in the five windows to form a composite. The line segments extracted from the right view are shown in Fig. 22, the corners in Fig. 23, and the structures found by our program are shown in Fig. 24. The results overlaid on the image are shown in Fig. 25.

Note that we find substantially the same structures as in the left view. An important difference is in the large building on the right; it is clearly extracted in the right view as its left edge now has a good contrast.

We think that the results shown here are highly promising, though many problems remain. The most apparent are the problems in the detection of the large building, caused due to poor contrast of its left-edge and the complex roof structure. Also we have not analyzed the tall building casting the large shadow, below the partially detected large building.

The difficulties can be considered to be due to the following causes; our future work will concentrate on these:

1. Poor contrast and/or texture—the buildings that are missed have poor contrast with the background or edges are fragmented due to many objects near them (such as vehicles parked nearby.) We believe that we can improve low-level edge detection and devise better line-linking methods. We are currently experimenting with detecting weak edges when the direction of the line is known.

2. Some of the buildings are missed just because of their small size and may be detected simply by improving the resolution. (The resolution of the digitized is not nearly the resolution of the original photograph.)

3. Complex roof structure—the large building in our example has a double roof with many structures on top. A more complex model of the buildings is needed for analysis. Stereo and improved shadow analysis should also help in such cases.

4. Density of buildings—as density of the buildings increases, shadow from one falls on others and segmentation becomes more difficult. Again, more complex models will be needed and stereo should help.

REFERENCES

1. S. A. Shafer, *Shadow Geometry and Occluding Contours of Generalized Cylinders*, Tech. Report, CMU Report CS-83-31, May 1983.
2. R. Nevatia and R. Babu, Linear feature extraction and description, *Comput. Vision Graphics Image Process.* **13**, 1980, 257–269.
3. A. Huertas and R. Navatia, Edge Detection in Aerial Images using $\nabla^2 G(x, y)$, Technical Report USCIPi No. 1010, University of Southern California, March 1981.
4. D. Marr and H. Hildreth, Theory of edge detection, *Proc. R. Soc. London Ser. B* **207**, 1980, 187–217.