COMPACT - A 3D SHAPE REPRESENTATION SCHEME FOR POLYHEDRAL SCENES

Pavel Grossmann

GEC Research Ltd.
Hirst Research Centre
East Lane
Wembley
Middlesex HA9 7PP

ABSTRACT

COMPACT is a high level surface representation particularly suitable for description of polyhedral objects and scenes and for recognition by matching to geometrical models. It builds on a line segment representation as developed in INRIA and converts a set of line segments into a set of surfaces (planes etc.). These are further converted into object / scene faces whose boundaries, internal features and mutual relationships are made explicit. The aim is to make matching to models of objects and scenes both faster and more constrained.

1. Introduction.

The problem of choice of the best representation for visible surfaces still remains to be solved. Much obviously depends on the application domain and on the type of sensor data we are working with. Objects and scenes that can be closely approximated by polyhedra can easily be described in terms of straight lines and planes, while more general surfaces have to be approximated locally by higher order surfaces in terms of surface 'patches' (Faugeras and Hebert 1986).

Most methods based on stereo can recover the depth information only for edges, i.e. parts of contours, surface markings etc. This is sufficient for 3D reconstruction of planar faces, but not for general surfaces. Some methods use surface information to constrain parameters of a chosen surface model like the superquadrics in (Bajcsy and Solina 1987).

Some sensing methods (like the laser range scanner) can yield a depth value for every point in the image and so enable construction of a consistent set of surface patches. Such representations, however, have been found to be neither unique nor very stable. Only when a whole surface can be entirely described by a simple (e.g. quadric) primitive - e.g. a cylindrical bottle (Faugeras and Hebert 1986) - can we achieve a stable description.

This work was undertaken as part of the ESPRIT programme P940 - Depth and motion analysis.

Due to the nature of manufacturing processes, many products have planar and cylindrical faces and a large class or objects can be described entirely in terms of planes and simple quadric primitives. This important category is somewhere between the 'planar world', the simplest of application domains, and the real world of general (but still reasonably smooth) surfaces.

The problem with devices like the laser range finder is that they require a (fairly) controlled environment for reliable operation and so their applicability is rather restricted. The edge-based stereo, on the other hand, is a robust method and as the recent results from INRIA (Ayache et al. 1985) show, the seemingly primitive line-segment representation is capable of describing even rather complicated surfaces.

2. The linear segment representation.

In the INRIA scheme (Ayache et al. 1985) the Canny operator is used to extract edges. These are linked together and a polygonal fit yields longer line segments. The 2D segments in two or three stereo views are matched to give a set of 3D segments in space. An example (corresponding to a laboratory interior at INRIA) is shown in Figure 8a. Each of these segments can offer only a weak support for any matching hypothesis but their large number builds up a powerful constraint leading to a unique interpretation.

It is clear, however, that for some simple (e.g. polyhedral) scenes we can easily recover the surface primitives and make them explicit. This is the aim of the research described in this paper. A representation scheme called COMPACT has been proposed and is currently being implemented at the Long Range Research Laboratory, GEC Research, Wembley.

This research is a part of a collaborative ESPRIT project P940: Depth and motion analysis. The aim of this project is to produce two working demonstrators: a robot vehicle capable of navigating in various man-made environments and a robot arm capable of recognizing and handling simple objects. Our aim is to design a representation for visible surfaces suitable for both demonstrators. As the 'planar (or polyhedral) world' is an acceptable approximation to the man-made environments in which

AVC 1987 doi:10.5244/C.1.33

the robot vehicle will operate, the present version of COMPACT already provides (or rather will provide, when it is completed) a representation suitable for one of the demonstrators. In the case of the robot arm, we want to be able to describe a wider range of surfaces and hence COMPACT will be extended to handle some non-planar (quadric) surfaces.

It is important to point out that by 'polyhedral scenes' we do not necessarily mean here scenes composed entirely of planar surfaces, but scenes containing a sufficient number of significant planar faces to make the number of interpretations (using planar primitives) acceptably small.

An alternative scheme, that is being developed at INRIA, also builds on the line segment representation and uses Delaunay triangulation to construct a polyhedral approximation to the free space and the visible surfaces. It is important to stress that the two schemes are highly complementary. While the INRIA approximation provides direct information on free space and obstacles without necessarily recognizing the scene, COMPACT is more suitable for scene recognition by matching to models.

A complete system for representation and 3D reconstruction of polyhedral scenes has recently been proposed by Herman and Kanade (1986). They consider both stereo and monocular aerial views of urban scenes and build up a model of the scene, which can be incrementally updated, to be used in model matching, display generation, path planning, etc.

COMPACT is not only different in the range of tasks it is designed to perform, but also its main ingredient, the direct extraction of surfaces from linear primitives (segments), is different. Herman and Kanade use corners extracted from images and task-specific knowledge about the scene to hypothesize surfaces (planes) and to confirm them (or otherwise) when new data is added.

3. Recovering planes.

Our method is based on direct recovery of a surface from the observed primitives (i.e. line segments). This contrasts with methods that imply the surface type from the observed boundary shapes (like planes and cylinders in (Bolles and Horaud 1984)), that construct a surface by using the boundary to constrain parameters of a chosen surface model (Bajcsy and Solina 1987) or that assume a priori that all visible surfaces are planes (Grimson and Lozano-Perez 1984). COMPACT does not require a complete connected boundary, and that contributes to its robustness.

While our direct recovery approach is quite general, it has so far been implemented in the simplest, planar, case. The extension to quadric surfaces will be discussed later on.

The segments in our low-level representation of the scene are grouped into sets corresponding to different planes by considering the coplanarity of segment pairs. In the graph-theoretic language we are looking for the maximum complete subgraphs (cliques) of a given undirected graph (Augustson and Minker 1970). There exist a number of clique-finding algorithms (Augustson and Minker 1970) and their implementation in COMPACT is currently being studied.

An alternative approach, which has already been implemented, could be called 'plane growing'. Each coplanar segment pair becomes a candidate plane which is tested for compatibility with the already existing planes by comparing their 'plane parameters' (i.e. the normal vector and the distance from the origin - see Appendix I). If compatibility is established, the existing plane is updated by the addition of the two segments and its parameters are recomputed.

In this way the segments are checked against planes (rather than other segments), which is more intuitive and hence more convenient at this experimental stage. Also, while the coplanarity is tested in the space of plane parameters, we also have to consider the errors of the segment endpoint coordinates. Even a small shift of edgel's endpoints out of the plane may result in a large change in the plane parameters of a pair if the segment is short.

Another method of grouping coplanar line segments which uses the Hough transform is currently being developed. The use of such a method will make it possible to compute the groupings in time proportional to the number of segments, n (rather than n^2). The basic ideas of Dual Space representation, recently investigated by Er(1985), are being used to construct a suitable Hough accumulator.

Two coplanar edges can be either *collinear, parallel* or *crossing* (i.e. the corresponding straight lines intersect). While two collinear segments do not determine a unique plane, both the parallel (P) and crossing (C) pairs can be used to compute the plane parameters.

Intuitively it might seem that the parallel pairs would provide a strong evidence for real surfaces as there are many man-made surfaces with parallel contour lines. Unfortunately, many parallel pairs can also arise accidentally, and they lead to false plane candidates. As an example, consider the image of a cube in Figure 1. Its edges will support 6 planes in the following way:

3 real planes, each supported by 4 C-pairs and 2 P-pairs
3 false planes, each supported by 2 P-pairs

By disregarding the P-pairs we somewhat reduce the support for the real planes but completely eliminate the false ones.

Hence in general (and also in all the examples given in this paper) we consider only the C-pairs, although for some special scenes (e.g. the camera calibration pattern in Figure 7) we may choose to use both P- and C-pairs.

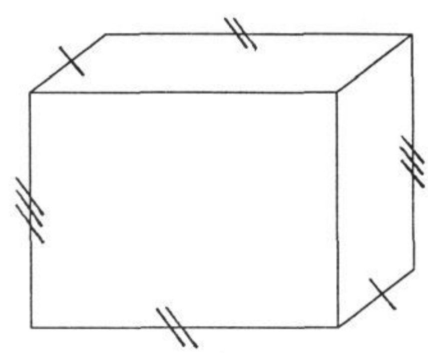


Figure 1. Image of a cube. Parallel edges marked (/), (//) and (///) support false plane candidates.

4. False plane candidates.

In general, however, we also get a number of false candidates due to C-pairs especially in such man-made environments as city-scapes or building interiors (like the model of an office interior in Figure 2). Such scenes often contain a number of lines in each of the principal directions (corresponding to the axes of some natural frame of reference). They are also likely to contain periodical patterns of parallel lines (like the window and the floor tiles in Figure 2) that increase probability of accidental alignment such as is shown in Figure 2 (the bold lines - also see Figure 5b). To eliminate such planes we can use both heuristics and strict geometrical reasoning.

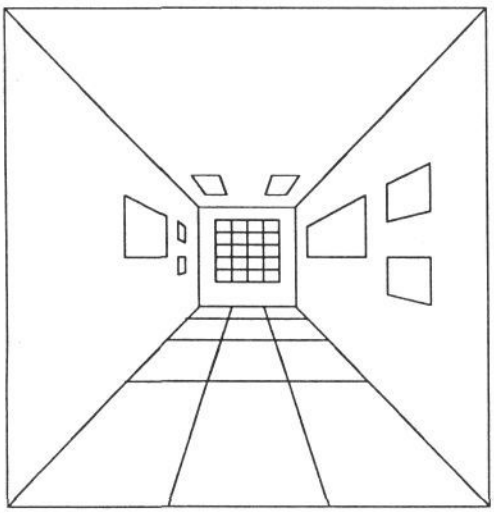


Figure 2. Wire frame model: Office scene 1 (empty).

Firstly we make an observation that the likelihood of finding a large number of segments that are accidentally coplanar is small and so we can impose a threshold on the size of the segment support for any candidate plane. This test is very fast and effective.

Secondly we make an assumption that any real surface is opaque. This is not true of windows, glass walls, etc., but these are in some sense exceptional and deserve special treatment. Thus, if a candidate plane corresponds to a real surface, no part of the scene that should be hidden (from the camera) by the surface ought to be visible in the image. This 'visibility test' in its simplest form (Appendix III) works for the empty office scene in Figure 2, but in general an infinite plane will obscure more parts of the image than the real surface the plane represents.

Clearly we have to extend our plane representation by considering the spatial distribution of segments within each plane and the finite physical surface they represent. For any point obscured by the (infinite) plane we construct the 'line of sight' and its intersection with the plane. Then we determine whether the intersection lies 'inside' or 'outside' of the relevant surface.

This task is well defined for a single surface with a closed convex boundary. If a closed connected boundary is available in the data, a line can be projected from the intersection to infinity and its intersections with the boundary counted (an even number implies 'outside').

In the usual case of a fragmented boundary one can use a less exact method based on the observation that if we connect the intersection point with the segment end points by straight lines, their distribution will be more or less isotropic for an 'inside' point and bunched close to one direction for an 'outside' point (Appendix II and Figure 3).

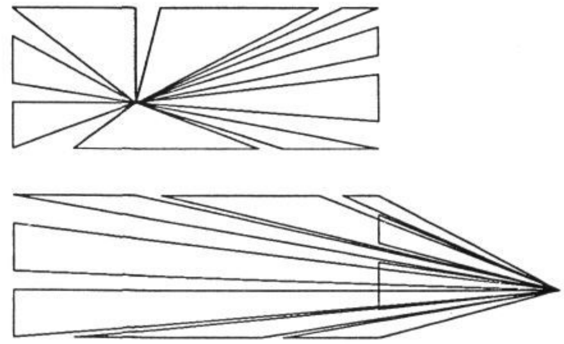


Figure 3. A simple way of determining the 'sidedness' of an interception point (see Appendix B).

In a general case the problem is highly non-trivial. The two cases in Figure 4 illustrate the two specific points:

1. There may be several physical surfaces in the same plane (e.g. several desktops in an office).
2. A single surface may have a non-convex boundary or a hole.

True, in the ideal case of closed boundaries there again is no problem in determining 'sidedness' of the intersection point, and there may indeed exist applications where the reconstruction of closed boundaries is the best way of approaching this problem. In many cases, however, there will be significant gaps in the boundary representation (e.g. see Figure 8a) that will make boundary reconstruction very difficult.

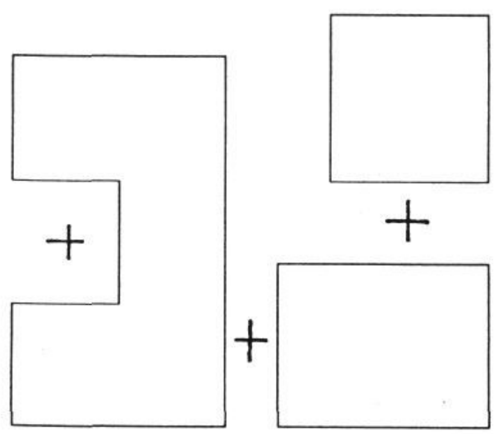


Figure 4. Candidate planes corresponding to several physical surfaces, one with non-convex boundary. The crosses correspond to points hidden by infinite planes. The camera is looking into the page.

5. Characteristics of physical surfaces.

Elimination of the false plane candidates leaves only planes corresponding to real physical surfaces. A small number of planes replaces a large number of segments as a representation of the image. The planes can now be matched to model faces using, for example, the method proposed by Grimson and Lozano-Perez (GLP) in (Grimson and Lozano-Perez 1985) where a particular plane-face assignment is checked using constraints derived from geometrical relationshiops between the model faces. In COMPACT this can be also done using various characteristics of the segment support ('features').

Here our representation primitives are upgraded from infinite planes to real surface candidates whose characteristics are made explicit. In the case of several physical surfaces corresponding to the same plane, we have to segment the plane's segment support into subsets corresponding to individual surfaces. Then the plane primitive is replaced by several surface primitives that share the same plane parameters.

The principal feature of a surface is its 'boundary', defined here as the set of segments that belong to the convex hull of the segment support. The boundary itself can be described by a number of features like eccentricity or aspect ratio, etc. The 'internal' set of segments may contain 'details' like the window, notice board and notices in Figure 2. Also 'adjacency' of surfaces can be made explicit when segments supporting 2 planes (and hence belonging to their intersection) are identified.

6. Matching strategy.

Matching algorithms for both line segments and planes already exist and have been used in a variety of situations (Murray 1987). They usually deal with one class of primitive and perform a tree search for the correct interpretation. In COMPACT we make the transition from planes to surface candidates that are described not only by the plane parameters but also by the corresponding features. The use of the features can make the search more efficient and ultimately can disambiguate otherwise degenerate interpretations.

Whenever a surface candidate is assigned to a model face, the assignment can be checked by comparing the relevant features. If the two sets are consistent, the segment support can be matched to the details of the model face.

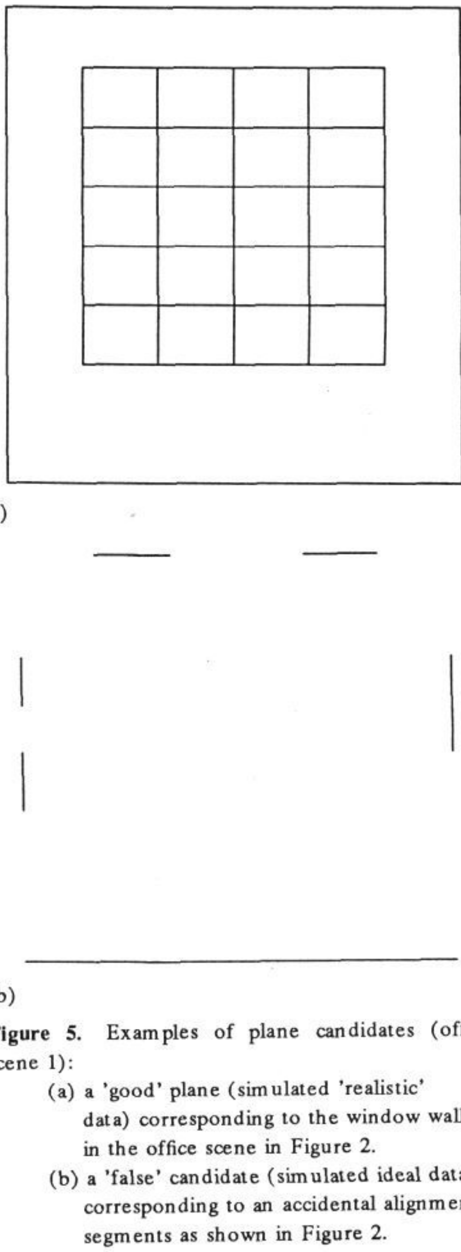
In some cases, however, it may be faster to consider the 'bare' planes without checking the features and so we should be able to choose different routes depending on the situation. The necessary information will be part of the model description. So, considering an assignment to the window wall in the office model in Figure 2, we would ask for a feature check. More than that, we could choose the window wall as the most prominent face in the model and look for a surface candidate with a compatible feature right at the beginning of the search. Thus the information about the significance of various model features, which is part of the model representation, will help to control the matching process.

7. Current implementation and results.

7.1. The modules in COMPACT.

COMPACT is currently implemented in Franz Lisp on our VAX-11 750. It has a highly modular structure that facilitates continuous development. Individual modules operate on two data structures, one containing segments and the other candidate planes together with all the relevant information (Appendix I).

The input module, MAKEDGES, processes either real image data in form of 3D edge segments (e.g. see Figures 8a and 7) or simulated data extracted from the wire-frame models of objects/scenes (Figures 2 and 6) and creates the segment data structure (Appendix I). The simulated segments can be 'ideal', i.e. taken directly from the model (Figure 2), or 'realistic', with gaps and superimposed noise (Figure 5a).



(a)

(b)

Figure 5. Examples of plane candidates (office scene 1):
(a) a 'good' plane (simulated 'realistic' data) corresponding to the window wall in the office scene in Figure 2.
(b) a 'false' candidate (simulated ideal data) corresponding to an accidental alignment of segments as shown in Figure 2.

The next module, FINDPLANES, then looks for the plane candidates using the 'plane growing' method described in section 3. Pairs of segments are tested for coplanarity and each coplanar pair becomes a 'new' plane candidate. Its plane parameters are computed and compared with the parameters of the 'established' plane candidates (if any). In the case of compatibility, the established candidate is updated. The new segments are included in its support set (unless one or both are already in) and its plane parameters are recomputed. If no established candidate compatible with the new pair is found, the new candidate itself becomes established. A special plane candidate data structure (Appendix I) is created and updated by creating a new entry for each new (different) plane candidate or by updating the segment support of an existing plane. A typical plane candidate is shown in Figure 5a: the window wall in the office scene (Figure 2), while Figure 5b shows a 'false' plane candidate: a set of accidentally coplanar segments whose position in the scene is also shown in Figure 2 (bold lines).

After all the plane candidates have been found, PURGEPLANES removes the ones that either have small segment support or that fail the visibility test described in section 4. Figure 6 shows another office scene with a desk (2 visible surfaces). The module correctly decided not to delete the desk faces although the associated infinite planes clearly 'obscure' points visible in the image.

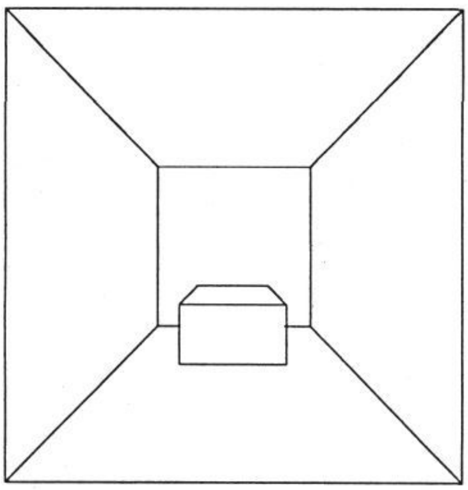
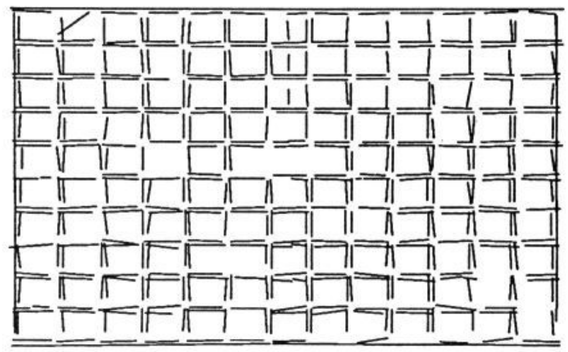


Figure 6. Wire frame model: Office scene 2 (including a 'desk' - 2 visible faces).

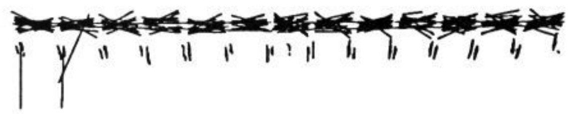
The next module, COMPATIBLE, creates, for each candidate plane, a list of segments whose endpoints are close to the plane. Once the plane candidates are known, the endpoints provide a more natural representation for a segment (rather than, for example, the direction vector) if we want to check its compatibility with a plane. This is particularly true if the segment is short and hence its direction error is large, although the endpoints are close enough to the plane. Our method for finding planes already produces the segment support of each plane and so COMPATIBLE can be seen mainly as a refinement, which is, however, important for short segments. At this stage, segment supports of different candidate planes may overlap to a large extent; it is even possible to get several different cliques consisting of the same set of segments with an addition of one that is different.

This situation is resolved by PRUNE which, for each 'overlapping' pair of planes, deletes the overlap from the 'smaller' one (according to a 'big eats small' principle). Subsequent PURGE then leaves a small number of significant, disjoint segment supports. A genuine overlap (i.e. an intersection of planes) can easily be restored by another run of COMPATIBLE. A genuine plane whose support consists exclusively of segments belonging to such plane intersections (e.g. a featureless wall in a room) could, theoretically, dissappear altogether and a safeguard against such (however unlikely) possibility will have to be implemented.

The calibration grid shown in Figure 7 has provided an interesting test for the modules. Due to a hardware fault in the image capture system, the reconstructed 'depth' values for the horizontal and vertical edges are different. The top view in Figure 7b indicates two planes separated by several centimetres. The 475 (mostly very short) segments that make up the grid representation yielded at first several hundred candidate planes that were eventually reduced to the two parallel planes shown in Figure 7c.



(a)



(b)

Figure 7. INRIA 3D segments: the calibration grid.
(a) front view.
(b) top view.

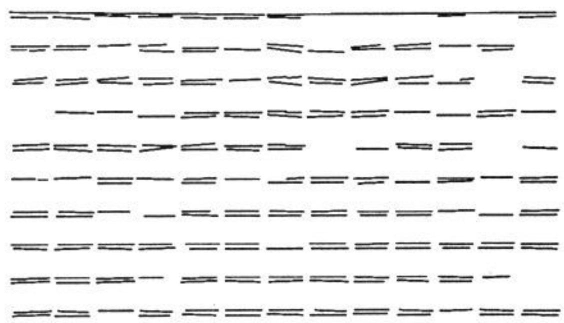
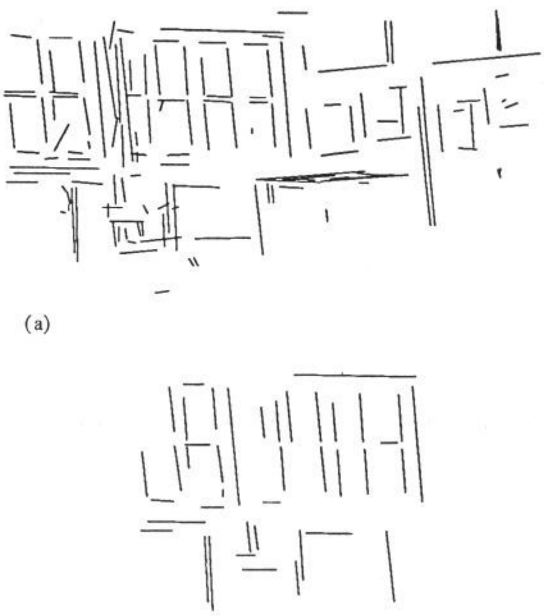


Figure 7c. Front view of the two planes extracted by COMPACT.

The ultimate aim of the program, however, is representation and interpretation of realistic indoor scenes like the one in Figure 8a. Figures 8 and 9 show two of the most significant planes extracted by COMPACT from the 3D segment data which correspond to a window (vertical) and a table top (horizontal).

When the extraction of planes is finished, FEATURES computes the 'boundary' and other characteristics of the segment distribution for each plane.

Support modules SHOWDATA and SHOWPLANE are used to monitor the progress of computation by displaying selected sets of segments at different stages of the program.



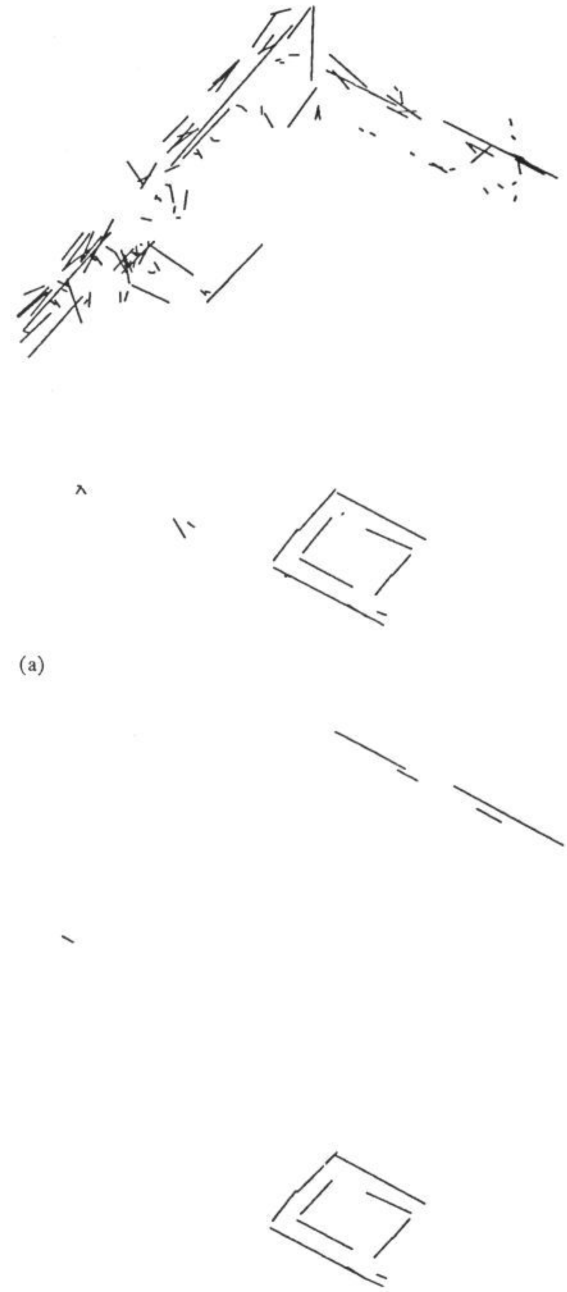
(a)

(b)

Figure 8. INRIA 3D segments: an office scene (front view).

a) all segments.

b) the most significant plane extracted by COMPACT



(a)

(b)

Figure 9. INRIA 3D segments: an office scene (top view).

a) all segments.

b) the most significant horizontal plane extracted by COMPACT.

7.1. Treatment of errors.

Reconstruction errors for all the segment endpoints are available in the form of an error matrix as part of the INRIA line segment representation. This error matrix will be used to compute the errors for all the segment and plane parameters in the usual way. At this experimental stage the reconstruction errors are represented by an anisotropic triplet $(\delta x, \delta y, \delta z)$ which is assumed to be independent of the position of the point in space.

These basic errors have to be combined with what could be called the 'perceptual resolution'. A physical surface, perceived (and modelled) as planar, may be planar only within a certain tolerance, e.g. some parts of a window, that we want to identify as a planar feature, may be a few centimeters out of the 'best fit' plane used to represent it.

This perceptual tolerance is at present being investigated, as it may be greater than the reconstruction errors and hence it can dominate the coplanarity tests.

7.2. Computing speeds.

By far the largest fraction of the total computing time is used by the FINDPLANE module. At present we perform an exhaustive search to consider all pairs of segments and for all the coplanar pairs we compute the plane parameters. At this research stage, the emphasis is on creating the complete system of representation and on solving all the problems in principle, leaving the speed and efficiency considerations for the later stages of the implementation. Thus the FINDPLANE stage for a typical office scene (Figure 8a, ~150 segments) takes several minutes on our VAX-11 750, and for the special image of the calibration grid (Figure 7, ~500 segments) it takes several hours.

The increase in speed will come from improvements in the basic method (e.g. considering for coplanarity only pairs of segments near in space), optimized code, use of 'C' language for the frequently used parts of the code and, eventually, from the use of parallel computation and specialized hardware.

8. Future work.

The next step in our implementation is the modification of a plane matcher of the GLP-type (Grimson and Lozano-Perez 1985) to include the use of 'features' in pruning the interpretation tree. As the significance of different features is very much context-dependent, it is a distinct advantage that in COMPACT it is provided as a part of the model description and need not be determined by the program.

The question of generalization of our scheme to other types of surfaces arises naturally, cylindrical and spherical surfaces being the obvious candidates. In both these cases we have to consider triplets of primitives (surface points with their associated tangents) that can be checked for consistency with a particular quadric surface. While the spherical case is relatively straightforward (Appendix IV), the cylinder turns out to be much more complicated. Also the combinatorics of the problem ($^nC_3$) suggest that a fast algorithm may be difficult to produce.

The most important task facing us, however, is to find a solution to the 'segmentation problem', which can be described in general terms as the problem of establishing the connection between each coplanar set of segments and the corresponding physical surfaces. More explicitly, we want to establish, given a set of segments and one or more intersections corresponding to the 'hidden points', (Figure 4):

1. whether the segment set represents more than one physical surface and if so, how to segment it,
2. whether a particular intersection point belongs to the interior of a given surface.

It is not difficult to see how the use of motion, colour or even some grey level (region) information might help here. When working with line segments alone, we have to look at the general task of interpreting the segment sets in terms of 'meaningful' surfaces, perhaps using shape models (e.g see Bagley 1985).

9. Summary.

We have developed a surface representation scheme that makes use of the large number of significant planar surfaces in man-made environments. The planar primitives that are directly extracted from line-segment data are further upgraded to contain explicit characteristics of the corresponding (finite) physical surfaces. A proposed plane matcher will make a flexible use of this additional information when it is compared to the explicit features of model faces which would form a part of our model representation.

The first part of the scheme has been implemented and the work on the model matcher is in progress.

10. Acknowledgements.

The author wishes to acknowledge discussions with Mike Brady and Bernard Buxton. The line segment data shown in Figures 7-9 were provided by our ESPRIT colleagues from INRIA.

REFERENCES.

1. Faugeras O.D. and M.Hebert, 'The representation, recognition and positioning of 3D shapes from range data', Techniques for 3D Machine Perception, A.Rosenfeld (ed), Elsevier Sci. Publ., 1986.

2. Ayache N., O.D.Faugeras B.Faverjon and G.Toscani, 'Matching depth maps obtained by passive stereo', IEEE 1985.

3. Bolles R.C. and P.Horaud, '3DPO: A three-dimensional part orientation system', Int.J.Robotics Res.

4. Bajcsy R. and F.Solina, 'Three dimensional object representation revisited', 1st Int. Conf. on Computer Vision, London (1987).

5. Grimson W.E.L. and T. Lozano-Perez, 'Model based recognition and localization from sparse range and tactile data', Int.J.Robotics Res. Vol.3 (1984) 3-35.

   Grimson W.E.L. and T. Lozano-Perez, 'Search and sensing strategies for recognition and localization of two and three dimensional objects', 3rd Int. Symp. of Robotics Research, Gouvieax (1985).

6. Augustson J.G. and J.Minker, 'An analysis of some graph theoretical cluster techniques', J. ACM, Vol.17, No.4 (1970) 571-588.

   Bron C. and J.Kerbosch, 'Finding all cliques of an undirected graph', Comm. ACM. Vol.16, No.9 (1973) 575-577.

7. Murray D.M., 'Model based recognition using 3D structure-from-motion: how good does the structure have to be?', Accepted for publication in IVC.

8. Bagley S.C., 'Using models and axes of symmetry to describe two-dimensional polygonal shapes', MIT thesis 1985.

9. Faux I.D. and M.J.Pratt, 'Computational geometry for design and manufacture', Ellis Horwood Ltd. (1981).

10. Er Meng Chian, 'Computer interpretation of engineering drawings', Ph.D. thesis, University of Essex, 1985.

11. Herman M. and T.Kanade, 'Incremental reconstruction of 3D scenes from multiple, complex images', Artificial Intelligence 30 (1986) 289-341.

Appendix I

Data structures in COMPACT.

The segment data structure consists of a 'list' of edgel names and their associated 'property lists', both standard features of LISP:

EDGES	(segment0, edgel1,..., edgeli,...) list of edgel names

Edgel properties:

END1	endpoint1 coordinates (x_1, y_1, z_1)
END2	endpoint2 coordinates (x_2, y_2, z_2)
LENGTH	segment length
UNIVEC	unit vector in segment's direction (u_x, u_y, u_z)
CDPL	list of cand. planes the segment supports (planei, planej,...)

Similarly for plane candidates:

PLANES	(plane1, plane2,..., planei,...)

Plane properties:

NORMAL	unit normal vector (n_x, n_y, n_z)
DISTANCE	distance from origin
EDSUP	list of segments suporting the plane (edgeli, edgelj,...)
BOUNDARY	list of segments that are part of the boundary

Treatment of errors:

At this stage the (anisotropic) errors associated with the segment end point coordinates are represented by a triplet $(\delta x, \delta y, \delta z)$ which is assumed to be independent of the image or space coordinates. Later on, the data errors and the corresponding segment and plane parameter errors will be incorporated in the data structures.

Appendix II

Determination of the 'sidedness' of a point.

Given a set of n edgels and a point P all in a plane, we construct all unit vectors $\vec{u}(P,i)$ along the lines connecting P to the edgel end points (see Figure 3). We define the following measure of 'directionality' D:

$$D = \frac{1}{2n} \left| \sum_{i=1}^{2n} \vec{u}(P,i) \right|$$

The value of D ranges from 0 to 1, D being small for P clearly inside the set and large when P moves outside to distances large compared with the size (diameter) of the edgel set. Hence $D < 0.3$, say, implies an inside point and $D > 0.7$ implies an outside point. There is no discrimination close to the boundary of the set.

Appendix III

The visibility test.

For a given point in the scene $P = (x,y,z)$ and a plane (described by the normal vector $\vec{n} = (a,b,c)$ and the distance from the origin d) we want to establish whether the point is 'hidden' from the camera (i.e. whether the point and the camera are on the opposite sides of the plane). The signed distance D of the point from the plane is:

$$D = ax + by + cz - d$$

If we find that $D(point)$ and $D(camera)$ have opposite signs, then the point and the camera are on the opposite sides of the plane, hence the plane 'hides' the point.

Appendix IV

Determination of a spherical surface from 3 tangents.

We assume here that we have 3 surface points and their associated tangents (and let us leave the question of how to obtain this information for later consideration). The tangents are normal to three planes passing through the points. These planes (in the usual vector notation): $\vec{r}\,\vec{u}_1 = p_1$, $\vec{r}\,\vec{u}_2 = p_2$ and $\vec{r}\,\vec{u}_3 = p_3$ intersect at the point $\vec{R}$ (Faux and Pratt 1981):

$$\vec{R} = \frac{p_1(\vec{u}_2 \times \vec{u}_3) + p_2(\vec{u}_3 \times \vec{u}_1) + p_3(\vec{u}_1 \times \vec{u}_2)}{\vec{u}_1(\vec{u}_2 \times \vec{u}_3)}$$

unless $\vec{u}_1(\vec{u}_2 \times \vec{u}_3) = 0$. If the distance D of the three surface points to the intersection $\vec{R}$ is the same (i.e. $D_1 = D_2 = D_3 = D$) the surface is spherical, $\vec{R}$ is its centre and D its radius.