On recognition of features in polyhedral scenes

Pavel Grossmann[*]
Long Range Research Laboratory
GEC-Marconi Hirst Research Centre
East Lane
Wembley
Middlesex HA9 7PP

A system that incorporates both high level representation of image data and a recognition and interpretation capability is described. Both competences are based on a structural, rather than metric, description of objects and image features and the corresponding models of object categories. The current implementation in the domain of polyhedral scenes transforms a set of 3D segments, our basic data representation, into a set of shape primitives such as planes and convex polyhedra (boxes) or planar configurations of segments (patterns). The interpretation module then uses its knowledge of the application domain to interrogate the representation in search for the expected evidence of a particular object category.

1 Introduction

Recognition of objects or features in the scene usually involves matching scene primitives to similar primitives used to describe object or feature models. Many different kinds of primitives have been used in the past, from significant points [1] to relational structures such as line junctions [2]. While the use of points or segments involves a high degree of ambiguity and complexity in the matching process, the less numerous and less ambiguous line junctions or, particularly, complete line drawings or wire frames are not particularly robust with respect to loss or degradation of data. In our work we find that a representation based mainly on (planar) surfaces and also on characteristic surface segment configurations (patterns) is more *compact* and *robust* and promises to be particularly suitable for interpretation of polyhedral scenes.

Most interpretation systems are concerned with recognition of particular instances of objects by geometrical matching of low level primitives [3, 4]. As an extension of this approach parametrized models have been used to recognize generalized *object families* [5]. More recently *superquadrics* have been used to describe complex articulated shapes (e.g. [6]) and the use of *modal dynamics* has been proposed in the description of deformable objects [7], although the suitability of such descriptions for recognition is yet to be demonstrated.

In this paper we investigate recognition of general object *categories* based on a structural, symbolic description of objects and features using simple shape primitives as proposed by Connell and Brady [8] and other authors.

Even a small number of highly symbolic primitives can lead to a complex matching problem if one attempts to match all the scene primitives to all the model primitives in our world domain. In our work we adopt a *task driven* approach and, instead of investigating all the possible matches, our recognition module *interrogates* the scene representation in search for evidence of a single desired object or feature.

Some aspects of our system (COMPACT) have been described earlier [9, 10, 11]. In this paper we shall give a brief outline of the whole structure and describe in more detail the recognition process.

2 The representation

The data input for our high level representation is a set of 3D line segments. A trinocular stereo vision system developed at INRIA [12] for the ESPRIT project P940 uses the standard image processing chain to generate 3D segments as a low level representation of the image data (e.g. see Figure 1).

The view of a calibration grid in Figure 1 is an example of a scene where the most significant structures are the planar surfaces (the two grid planes and the wall in the background) and the planar segment configurations (the grid patterns). These structures can also be recovered in a robust way : distortion or even removal of some of the segments will not significantly effect the feature extraction.

2.1 Planes

Our surface extraction method is based on testing small sets of segments for compatibility with a particular simple surface type. Although variants of the method have been developed to recover also spherical, cylindrical and conical surfaces [9], in this paper we shall concentrate on planes and polyhedral scenes [13]. This gives us a limited application domain that nevertheless corresponds to a large range of indoor scenes and man-made objects in which we are particularly interested. In the planar case

[*] This research was supported in part by the ESPRIT project P940.

BMVC 1990 doi:10.5244/C.4.54

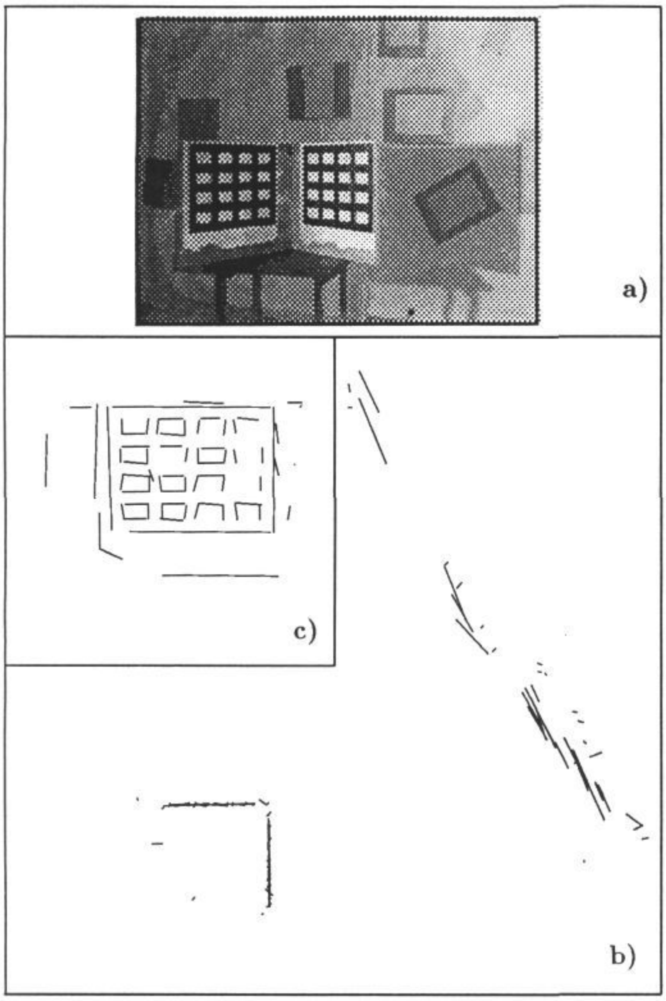

Figure 1: Indoor scene with a calibration grid (INRIA)
a) one of the triplet of images
b) top view of the 3D segments
c) one of the identified planes

pairs of segments are tested for coplanarity and plane candidates are grown by adding further segments [13].

2.2 Boxes and more complex shapes

When planes are extracted from the segment data, plane intersections are made explicit by labelling the segments that belong to two different planes. These intersections are labelled as *concave* or *convex* using our knowledge of the camera position. They are then used to establish links between the planes that belong to a connected convex surface of an object or a concave (inside) surface of a room. This finally enables us to reconstruct the 3D shape of a simple object (a *box*) or a simple space (a *room*) [10, 11].

Convex shape primitives (boxes) can be further combined to form a *composite shape* or *multibox* (e.g. a desk as shown in Figure 4b). This introduces an important question of the nature of a single object and the corresponding problem of identification of different boxes as convex subparts of the same object (scene segmentation). A set of heuristics have been developed that can be used in some simple cases [10, 11] but in general this problem requires further study.

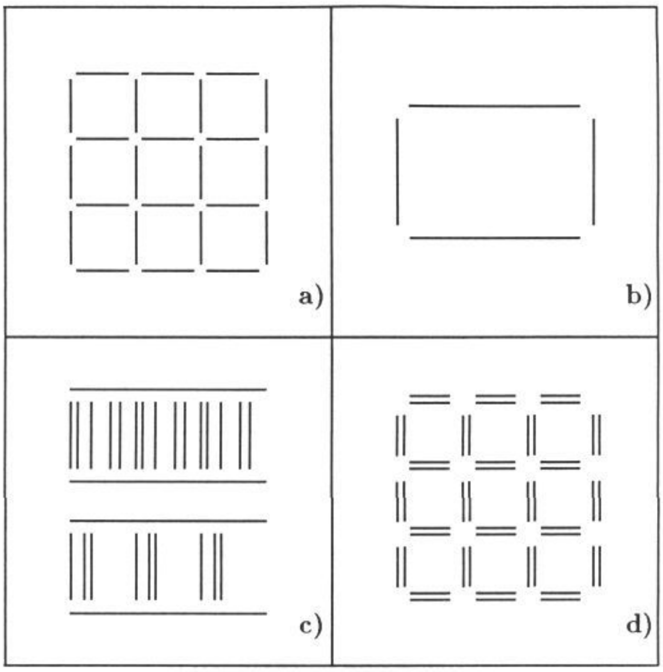

Figure 2: Types of rows and patterns
a) tiles
b) frame
c) books (top) and drawers
d) window

2.3 Segment configurations

Many objects and features in man-made environments can be recognized by a characteristic configuration of lines - a *pattern* - that either corresponds to the object itself (a window or a notice board) or an important part of it (a set of desk drawers).

Such patterns can be recovered [10, 11] by examining the line configurations within each plane extracted during the first stages of our analysis. Looking for a robust feature that can be recovered even from incomplete and imperfect data, we rejected the obvious rectangle or parallelogram that would require good information on corners (necessary to avoid the obvious connectivity problem, e.g. see Figure 2a) and chose instead as our pattern primitive a straight *row* of parallel segments that are, like the rungs of a ladder, equal in length and perpendicular to the *row* axis (Figure 2).

For each such *row* we can compute a range of properties such as its length, orientation, number of segments and the distribution of gap widths. Similarly we can investigate pair relations for coplanar *rows* that are parallel or perpendicular. *Rows* found to be related (usually adjacent or overlapping pairs) are then grouped to form larger characteristic *patterns*. At present we consider the following pattern classes that typically correspond to important physical objects or features : *window, tiles, drawers, books and frame*. Note that a rectangle is recovered here as a *frame* pattern after the larger connected patterns have been reconstructed which removed the corner ambiguity.

This aproach can obviously be extended to cover a larger range of patterns and other domains.

2.4 Sticks

There is one important object category in the domain of office scenes that often defies description as a polyhedron - the chair, particularly the molded plastic variety. A large number of chairs, however, still have legs that can aid recognition and that can be described as *sticks* (e.g. in Figure 4a). A *stick* is defined here as a close parallel pair of long segments and a set of *sticks* with certain symmetry and orientation may correspond to a set of table or chair legs (a *legset*). Hence *sticks or legsets*, that can be easily recovered during the plane extraction process, join *planes*, *boxes*, *multiboxes* and *patterns* as our representation primitives.

3 Errors and uncertainties

The treatment of errors in our method is based on the basic assumption that the uncertainties in the position of the 3D segment endpoints (our input data) can be adequately described by the corresponding *covariance matrices*. These uncertainties are then *propagated* to the higher-level representations using the standard formula [14] based on the first order Taylor expansion :

$$V_{kl}(\vec{y}) = \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{\partial y_k}{\partial x_i}\frac{\partial y_l}{\partial x_j} V_{ij}(\vec{x})$$

where $\vec{y}$ is a set of m functions which all depend on the set of n random variables $\vec{x}$ ($y_k = y_k(\vec{x})$). The error on any quantity is then given by the square root of the relevant variance.

Starting with the experimentally determined covariance matrices of the segment endpoints we first compute the errors for the segment parameters (unit vector and length), then for the intermediate quantities (e.g. vector products) and finally for the resultant surface parameters.

While the first assumption has been experimentally verified for the data used in our experiments, the use of the first order Taylor expansion becomes questionable for non-linear functions and large variable errors and therefore extra care is needed. The 3 planes shown in Figure 1 b) clearly demonstrate the considerable increase of point errors with the increasing distance from the cameras and the fact that all three have been correctly identified gives indication of the *dynamic range* of our plane extraction algorithm.

Uncertainties in interpretation are in general described in terms of conditional probabilities and are dealt with in the next section.

4 The interpretation module

Recognition as a process often involves an attempt to match all the data primitives to all the model primitives with a combinatorial explosion being the inevitable consequence, particularly when large numbers of low level primitives are involved.

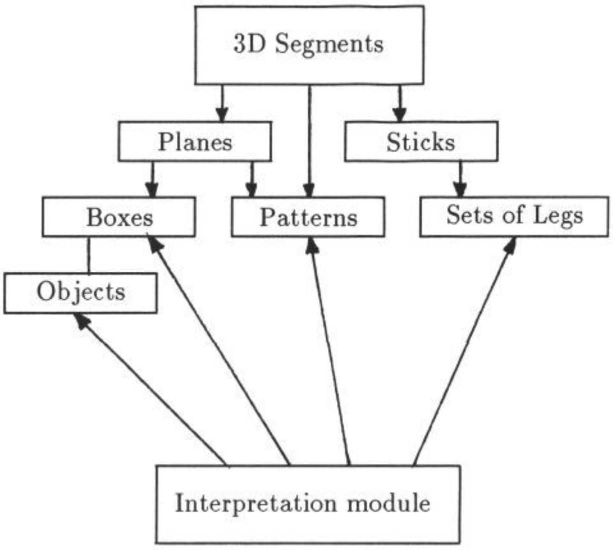

Figure 3: **COMPACT** - Representation and Interpretation Scheme

In our approach we replace typically hundreds of 3D segments by several planes, boxes and patterns to represent the scene; similarly the models of objects and features are described in terms of small numbers of such primitives.

Furthermore, instead of attempting to interpret all parts of the scene at once, we adopt a **task driven** approach and merely look for evidence of a particular object in the scene representation. In other words, instead of asking : "What do you see ?" we ask : "Can you find a table ?" etc. We see this as a more efficient way of using the available visual information - only the computation necessary for the completion of a particular task is done.

In a way we follow here Minsky's suggestion [15] and *assume* some knowledge of the domain we are dealing with, anticipating objects and features to be found in the observed scene.

In the language of bottom-up versus top-down processing we first create, in a bottom-up way, a high level representation of the image data. Subsequently the recognition module, in a top-down fashion, interrogates the data representation in its search for a particular object or feature using the domain-specific knowledge available to it (see the schematic representation in Figure 3).

While our aproach is quite general, for practical reasons we define, for the purposes of recognition, two classes of entities :

- 2D features (window, doorway etc.)
- 3D objects (desk, chair etc.)

In general an object or feature f_i from the domain set of models $F = (f_1 ... f_n)$ is described in terms of primitives p^j from the set of primitives $P = (p^1 ... p^m)$ and their relationships and characteristics.

The formal data structure we use here is the Lisp *property list*. Each model is defined by an ordered list of primitives. The order determines the sequence in which the primitives are searched for. Each primitive has a *requirement status* (RS) which determines the course of action when some expected information is not available and which takes the following values :

- 0 not required
- 1 desirable (required, absence not fatal)
- 2 essential (required, absence fatal)

and a number of characteristics each of which may again be a list.

4.1 2D features

The 2D features are usually described in terms of a single primitive. Their interpretation is more ambiguous that that of a 3D object and so it is treated in a slightly different way. For example a window is described as a 2D pattern in a vertical plane. Although we first search for the pattern *window*, patterns *tiles* and *frame* (Figure 2) are also considered. The search order is determined by the expected prior probabilities. For any type of scene we can prepare in advance tables of probabilities linking the 2D features and the corresponding segment patterns, i.e. the probability $P(p^j|f_i)$ that feature f_i will be observed as the pattern p^j :

$$\sum_{j=1}^{m} P(p^j|f_i) = 1$$

and similarly the probability $P(p^j|f_i)$ that the observed pattern p^j corresponds to the feature f_i :

$$\sum_{i=1}^{n} P(f_i|p^j) = 1$$

We clearly have also :

$$P(f_i|p^j) \cdot N(p^j) = P(p^j|f_i) \cdot N(f_i)$$

where $N(f_i)$ is the number of objects of type f_i in the scene and $N(p^j)$ number of instances of pattern p^j in the data.

The result of the search for feature f_i is then the pattern type p^j with the highest prior probability $P(f_i|p^j)$ found. Also returned are the other possible interpretations f_k and the associated probabilities $P(p^j|f_k)$.

4.2 3D objects

3D objects usually allow richer description not only in terms of any possible patterns but also in terms of their 3D shape - be it characteristic proportions of a single box or a more complex structure of a polyhedron such as a desk (Figure 4b) or a staircase. Usually more thanone of our set of primitives (*multibox, box, pattern and legset*) and their characteristics are used. As an example we list the primitives used to define the model *desk* and their characteristics :

- multibox
  - RS = 1
  - maximum number of constituent boxes = 2 (staircase has no such limit)
  - box configuration : boxes are aligned so that they share 2 planes (unlike the staircase)
- box
  - RS = 2
  - upper limits on the height, width and depth
  - relative box position in the scene (on the floor)
- pattern
  - RS = 2
  - ordered list of patterns most likely corresponding to desk drawers
  - pattern position (vertical plane)

In response to the user command **find desk** the interpretation module first searches for a *multibox* in the data representation. If one is found, its characteristics are determined and checked for consistency with those required by the *desk* model. Then the constituent *boxes* are analyzed.

If no multibox (RS=1) is found, we look for the next primitive type (box). If a box is found, its characteristics are checked. If there is no box (RS=2), the search for a *desk* is terminated.

Otherwise we look for a *pattern* in one of the box planes following the order of pattern types given by the model. Again the absence of a suitable pattern (RS=2) terminates the search.

In the general case, the search for a particular model M starts with the complete list of hypotheses. Following the model M description we look for the required primitives and use their characteristics determined from the data to rule out first the model M and then the other hypotheses. When an essential primitive is not found or a characteristic is found to be inconsistent with the model M, the search ends and the result is **M not found**.

Otherwise we can have one of two other outcomes. If only M survives, the interpretation is unique - **M found**. If also some of the other hypotheses survive, the interpretation is ambiguous - **M possibly found** - and the surviving list of hypotheses is given.

Is is easy to see that our aproach to object description and recognition is inspired by the human visual experience. In this pragmatic approach the model description is usually minimal - just sufficient to distinguish each model from all the others when good data is available (i.e. when all the model characteristics can be determined). If ambiguities cannot be resolved, we either have to refine the model description by including more features or to improve the data.

5 Implementation and results

The current research version of the COMPACT system is implemented in Common Lisp running on Sun-3 and Sun-4 workstations. The plane extraction module has also been implemented in "C", which considerably speeds up the execution.

5.1 Data structures

The basic data structure in our system is the Lisp *property list.* Starting with the input data, the 3D segments are represented as a list of segment names. With each name there is associated a number of *properties* such as the end point coordinates etc. In this way each data parameter can be accessed as a particular *property* of a named segment. Consecutive COMPACT modules operate on the existing property lists updating them and creating new lists in the process.

So the *segment* list is used to generate a list of *planes* and their parameters (including the names of all the segments assigned to each plane). At the same time a new property is created for each segment that contains the names of planes supported by it. In this way also lists of *sticks, boxes, multiboxes, patterns* and *legsets* are created, containing all the links between the representation primitives at different levels (Figure 3). This loose configuration of property lists is extremely flexible in accommodating new features as they become available. Finally, during the interpretation stage we found it useful to create and keep updating a list of all *interpretations*, so that repeated inquiries (as may occur in real life) could be answered with reference to this list without repeating the corresponding analysis.

5.2 Experiments

The first phase of experimentation was designed mainly to validate the principles of our scheme. A sample of both synthetic and real data sets was used to test the integrated chain of modules up to the interpretation stage. Presented with a particular scene (i.e. a set of 3D segments) the interpretation module was asked to find a particular object or feature. Various simple objects and features have been found and recognized in a range of synthetic room scenes. Several simulated scenes each containing a single object from the list *desk, table, chair, bookshelf, filing cabinet, staircase* (Figure 4) have also been used to test our method of object identification.

The next (current) series of experiments involves extensive testing of the system by a variety of real scenes provided by our ESPRIT P940 partners from INRIA (France) and ELSAG (Italy). The robustness of the procedures used to create the data representation and also of our scheme for representing object categories is being examined and evaluated. In Figure 5 we show a representation of a Rubik cube - a simple object easily recognized by its shape and the characteristic surface pattern. COMPACT easily extracted the three visible planes, established their connectivity and analyzed the surface pattern to reach the correct identification. Similarly the calibration object in Figure 1 was successfully reconstructed and identified.

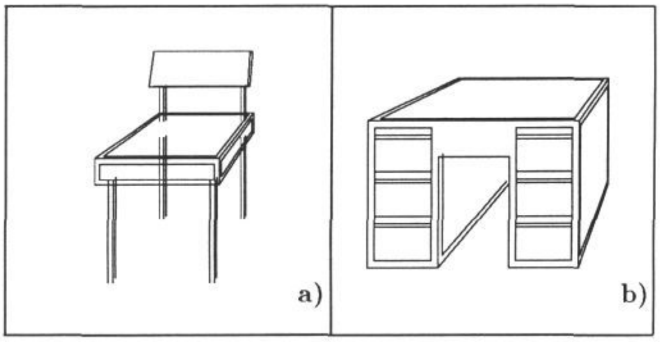

Figure 4: Examples of 3D objects (simulated data)
a) chair
b) desk

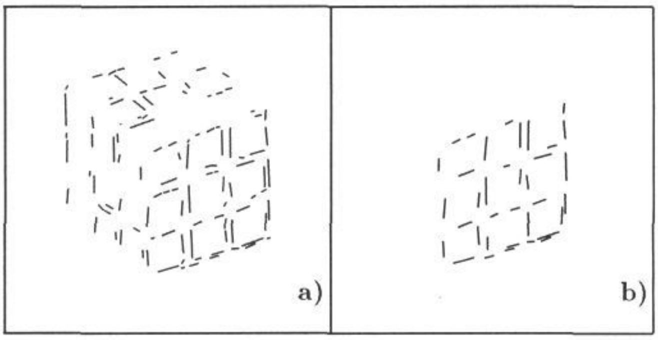

Figure 5: The 3D segment representation of a Rubik cube (ELSAG)
a) all segments
b) one of the 3 visible faces

The more realistic (i.e. not specifically prepared or selected) office scene with a barely recognizable desk in Figure 6 (plus an assortment of terminals and other objects) provides a much more severe test - the presence of clutter and the considerable errors in the segment position and orientation make even the first stage - plane extraction - a considerable task. Although COMPACT succeeded in extracting the front desk face (Figure 6 b) and also the top surface (not shown), this gave us only a single box structure. Hence identification of the drawers became of vital importance and here our present recognition criteria were not tolerant enough. We have to make a compromise between robust recognition and discrimination - an algorithm that recognizes any set of lines as *drawers* has little discrimination. In this sense, Figure 6 presents the kind of data that our present recognition module cannot quite handle. We expect, however, that a modest improvement in the data quality together with increased flexibility of our recognition algorithms will overcome this difficulty in the near future.

6 Conclusion

We have described an integrated system that incorporates both the representation of image data and an interpretation module. A bottom-up process of recovering structural description of the scene from 3D segments creates a high level representation in terms of surfaces, 3D

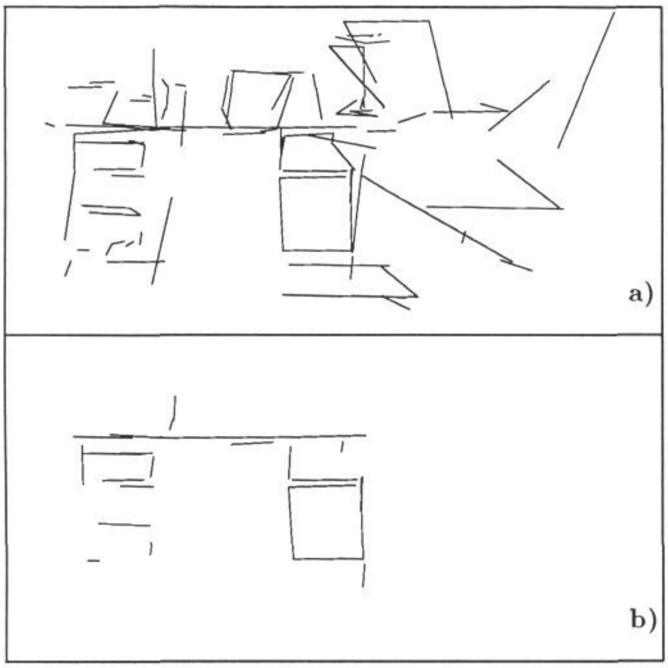

Figure 6: The 3D segment representation of a desk (INRIA)
a) all segments
b) one of the visible faces

shape primitives and 2D patterns. A top-down interpretation process, assuming some prior domain specific knowledge, interrogates the representation structures in search of data features that correspond to the expected objects or scene features.

7 Acknowledgements

Thanks are due to my colleagues in the ESPRIT P940 collaboration for the use of their data : INRIA at Sophia-Antipolis (Figures 1 and 6) and ELSAG in Genoa (Figure 5) and to the referees for their useful comments.

References

[1] C. G. Harris. Determination of ego-motion from matched points. In *Proceedings of the Third Alvey Vision Conference*, 1987.

[2] R.Horaud and F.Veillon. Finding geometric and relational structures in an image. In *Proceedings of the First European Conference on Computer Vision*, 1990.

[3] E.Grimson and T.Lozano-Perez. Model-based recognition and localization from sparse range or tactile data. *International Journal of Robotics Research*, 3, 1984.

[4] O.D.Faugeras and M.Hebert. The representation, recognition and locationg of 3d shapes from range data. *International Journal of Robotics Research*, 5, 1986.

[5] W.E.L.Grimson. Recognition of object families using parametrized models. In *Proceedings of the First International Conference on Computer Vision*, 1987.

[6] J.Lagarde F.P.Ferrie and P.Whaite. Recovery of volumetric object descriptions from laser rangefinder images. In *Proceedings of the First European Conference on Computer Vision*, 1990.

[7] Alex Pentland. Extraction of deformable part models. In *Proceedings of the First European Conference on Computer Vision*, 1990.

[8] J.H.Connell and M.Brady. Generating and generalizing models of visual objects. *Artificial Intelligence*, 31, 1987.

[9] Pavel Grossmann. Compact - a surface representation scheme. *Image and Vision Computing*, 1989.

[10] Pavel Grossmann. From 3D line segments to object and spaces. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1989.

[11] Pavel Grossmann. On recognition of object categories. In *Proceedings of the Fifth Alvey Vision Conference*, 1989.

[12] Nicolas Ayache and Francis Lustman. Fast and reliable passive trinocular stereovision. In *Proceedings of the First International Conference on Computer Vision*, 1987.

[13] Pavel Grossmann. COMPACT - a 3D shape representation scheme for polyhedral scenes. In *Proceedings of the Third Alvey Vision Conference*, 1987.

[14] O.Skjeggestad A.G.Frodesen and H.Tøfte. *Probability and Statistics in Particle Physics*. Universitetsforlaget Oslo, 1979.

[15] M.Minsky. A framework for representing knowledge. In P.H.Winston, editor, *The psychology of computer vision*. McGraw-Hill Book Company, 1975.