Image understanding and digital photogrammetry

TONI SCHENK, Columbus

ABSTRACT

Substantial progress has been made in digital photogrammetry during the last few years. Several products are operational, most notably softcopy workstations. Digital photogrammetry offers the potential to automate photogrammetric procedures. Despite of all the progress that has been achieved in digital photogrammetry, the degree of automation is quite low. Map compilation, for example, is still performed in the same way as in conventional photogrammetry. Most photogrammetric processes require first an analysis or interpretation of the imagery used, before measuring and digitizing can be performed. Image understanding and object recognition are a hot research topic. Most research is concentrated on analyzing indoor scenes, or is related to navigating autonomous land vehicles. In this paper we provide an overview of the image understanding problem and describe model-based object recognition as well as solutions for interpreting aerial scenes.

1. INTRODUCTION

Since time immemorial, mankind has been fascinated by the idea to create a machine that would somehow exhibit mental capabilities. The robot is a typical example of such dreams. With the attempt of endowing computers with information processing capabilities similar to those of humans, researchers in artificial intelligence pursue this dream in modern times. Ever since computers became available, researchers tried to mimic the mental faculty of seeing. Expectations were pushed far beyond what could be delivered and disillusion followed. The problem has been tremendously underestimated---like many other problems tackled by artificial intelligence. We see and interpret scenes without conscious effort, however, this does not mean that the task is easy. Clearly, the lack of a detailed understanding of vision is the reason why it is so difficult to make computers understand and analyze images.

Considerable progress has been made in digital photogrammetry during the past few years. Several products are operational and available to photogrammetrists: softcopy workstations, programs to produce digital orthophotos and DEMs, systems to precisely determine points in industrial applications, just to mention a few. By operational we mean that the products do not work only in laboratory environments under tight supervision of the research staff.

As impressive as the progress of digital photogrammetry is, the automation of photogrammetric procedures is still in its infancy. After all, working on a softcopy workstation is quite similar to operating an analytical plotter. Most photogrammetric tasks involve a fair amount of analyzing and interpreting the stereopairs. Think of compiling a map, for example. First, the human operator analyzes the images, makes a decision where objects to be mapped are, identifies and measures them, followed by changing their position and shape according to unwritten rules and specifications on how the map should look like. The first part of this process has to do with image understanding (scene analysis) and object recognition, while the second part is related to generalization.

In this paper we provide a short overview of work in image understanding. The next Section describes different terms and relates them to the vision paradigm. This is followed by a Section about a model-based object recognition approach in robot vision. Though it relates to indoor scenes, the technique to overcome the combinatorial explosion and the generation and verification of hypotheses is of general interest. Object recognition methods which are more amenable to outdoor scenes and aerial imagery, are presented in Section 4. We conclude with some remarks concerning

the present status and indicate on how to further advance the automation of photogrammetric procedures.

2. BACKGROUND

The purpose of this Section is to elaborate on the terminology and to associate the terms *image understanding, scene interpretation, object recognition, feature extraction* and *feature classification* to the computer vision paradigm. We begin with a concise summary of the paradigm.

2.1 Marr's vision paradigm

The most advanced and widely accepted paradigm of computer vision is based on Marr's theory about vision (Marr, 1982). His theory has a strong information processing underpinning. He argues for understanding an information process - vision - at three different levels.

computational theory specifies what the visual system must do. It answers the question about the purpose of the computation and the strategy for solutions.
representation and algorithm investigates the representation of input and output and the algorithm that transform one into the other.
hardware implementation answers the question of how the representation and the algorithm can be physically implemented by neurons.

The tenet of Marr's theory is that the shapes and positions of things can be made explicit from images without knowing what these things are and what role they play. However, this cannot be accomplished in one step, rather in a sequence of representations designed to facilitate the subsequent construction of physical properties of objects. The three main steps are generating the primal sketch, the 2.5-D sketch and representing 3-D models.
By and large, computer vision pursues the same goal as human vision: generate descriptions about the scene from images. The descriptions must be explicit and meaningful so as to allow other system components to carry out a task. In that aspect, computer vision is part of an entire system that interacts with the environment, say a robot. Consequently, tasks such as decision making, planning, executing decisions, are not part of computer vision.
Usually, the paradigm (see Fig. 1) begins with a raw image. We also include image formation, a point forcefully advocated by Horn and now accepted by many vision researchers. After all, machine vision may be viewed as the inverse process of image formation. Thus it only makes sense to obtain a thorough understanding of image formation.
The primal sketch is the result of edge detection. Edges are likely to have been caused by structures in the scene, such as object boundaries, markings and surface discontinuities. The unorganized edge fragments, bars and blobs are grouped into higher-level tokens, which are now processed by the independent modules *stereopsis, shading, motion, texture* to yield the 2.5-D sketch.
The 2.5-D sketch contains less data than the raw image, but more important, it is more explicit. An edge could be an object boundary or a shadow; a single pixel can be everything. Depth and 3-D shape information is particularly important. Shape and depth information is obtained independently from stereo, shading, motion and texture processes, also called shape-from-X processes. Note that the 2.5-D sketch is obtained purely from the raw images. It is the result of bottom-up processes, also referred to as *early vision.*
The 2.5-D sketch is the transition from image space to object space. Subsequent processes, termed *late vision*, are scene oriented rather than image oriented. Extracted features are grouped together, segmented and eventually parameterized. If the application of the vision system is object

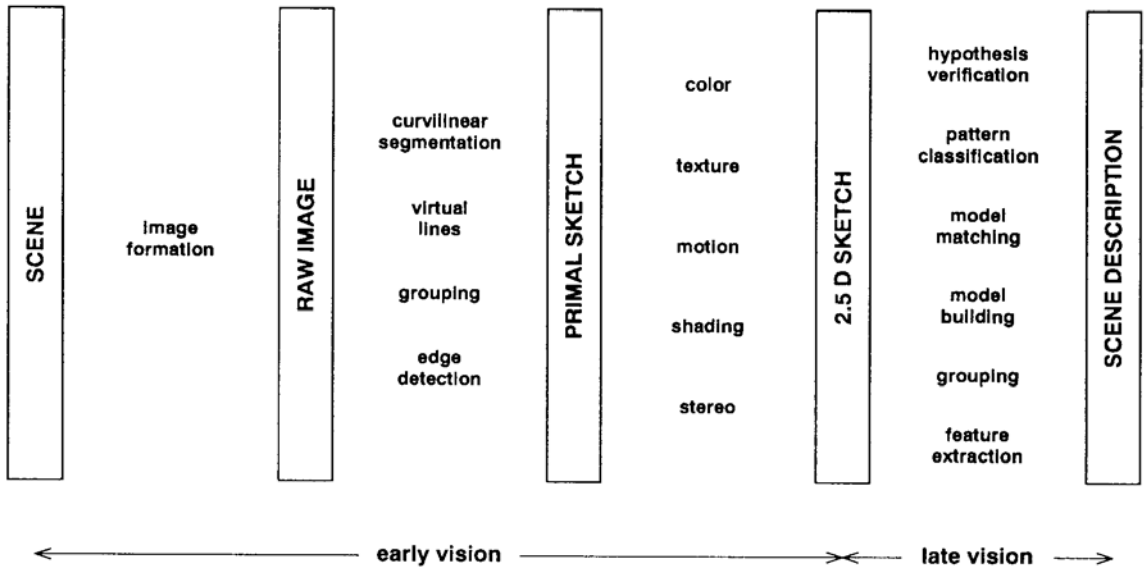

Figure 1: Paradigm overview of computer vision.

recognition then a data base with models of objects is generated. The parameterized features are now matched with the object library.

2.2 Image understanding and object recognition

For a person to respond properly to the environment he must analyze, interpret and understand visual stimuli. Ideally, the same feat must be accomplished by a robot or an autonomous vehicle. To pick a part from a bin or to navigate through a cluttered environment the robot must understand its environment from sensor data and stored knowledge. The net result of image understanding is a fully interpreted scene. Image understanding or image interpretation is application dependent.
By and large, image understanding and scene interpretation are used interchangeably. From Fig. 1 we conclude that they are late vision processes. Interpreting 3-D scenes is one of the most intensively researched areas in computer vision. Applications vary from locating and picking up tools from a bin, navigate a robot through a dynamically changing environment, guide an autonomous vehicle along a road or through unknown landscape, or extract features from aerial imagery.
Object recognition is an important subtopic of image understanding. Its objective is to find instances of objects in sensory data like digital images. For many robot applications models of objects are stored in library because indoor scenes comprise a finite number of objects that can be modeled. The recognition task proceeds by first extracting and grouping features from the images. These data sets are then compared with the stored models. Once an association is made, the geometrical relationship between data and models is established.
While image understanding is essential for many machine vision applications its role in digital photogrammetry is less important. Here, the emphasis is on recognizing and locating objects. Subsequent tasks, such as analyzing objects and their interrelationships, are typically performed by GIS.
Other areas of 3-D scene interpretation include 3-D sensing techniques, modeling and suitable representation of 3-D objects, and matching strategies to keep the combinatorial explosion under control. The representational issue of 3-D objects for aerial applications is of central interest in

digital photogrammetry. Considering the variety of natural and man-made objects in aerial scenes, their complexity in shape and size, requires a different approach to modeling and representing them. In summary, data that are related to objects, are extracted from images. This task, often called feature extraction, is carried out by edge and boundary detection, and region segmentation, including texture. The second task is to describe the objects, either by geometrical structures or by relational structures, or by some other means. Finally, extracted data sets are matched with the objects.

3. MODEL-BASED OBJECT RECOGNITION WITH GEOMETRICAL CONSTRAINTS

In this Section we describe model-based object recognition which is based on extracting features followed by matching them to a library of stored objects. Various geometrical constraints help to reduce the search space thus keeping the combinatorial explosion of searching a huge model base under control. Features to be matched include distinct points, edges, curves and surface patches. The following tasks must be solved:

1. Build a library of objects (model base). The geometric description of every object depicts shape characteristics in a local object coordinate system.
2. Extract features from the images. Group and segment them such that data features correspond to one object *(segmentation problem)*.
3. From the model base select those objects which are likely to correspond to a set of given data features *(indexing problem)*.
4. Find instances of objects in the data by establishing a correspondence between data features and object features *(correspondence problem)*.
5. Find instances of objects in the data by transforming the object to the image for checking global consistency *(hypothesize-and-test problem)*.

Extracting features, organize and represent them in a suitable form is a typical perceptual organization problem. In the most simple form, it would involve edge detection, edge formation and edge segmentation. There is a plethora of edge detecting operators available. Edge forming depends on the edge detector used. Curvi-linear edge segmentation results in straight line segments and curved parts of an edge. Yet another processing stage may analyze consecutive straight line segments for rectangular polygons.
A straightforward but naive solution to the indexing problem is simply to take every object from the model base and perform steps 3 and 4. A more intelligent approach would be to consider additional attributes such as color. Also the number of data features may be used to select only those objects that have more features, assuming that the data features are only a subset of all features describing an object. Finally, domain-specific knowledge may be used. For example, it may well be that certain objects do not occur in certain parts of the image or in the neighborhood of other objects.

3.1 Correspondence problem

We follow Grimson (1990) to explain the correspondence problem and the significance of geometric constraints by way of a simple 2-D example. Suppose the feature extraction process (e.g. edge detection, grouping and segmenting) generated the three data features f_1, f_2, f_3 (see Fig. 2) which we assume correspond to the boundaries of an object. Also shown in Fig. 2 is an object selected from the object library. The problem is to establish a correspondence between data features and object features without performing the costly transformation object to image.

Figure 2: Data features extracted from image (left) and object selected from the object library (right).

In a brute force approach we could pair each data feature with all object features and check all possible combinations for consistency. This would amount to an exhaustive search of the correspondence space (see Fig. 4). The n-dimensional correspondence space (n = number of data features) is tesselated by the number of object features. Referring to our example we obtain $5^3 = 125$ assignments most of which do not make any sense. Thus, the goal is to determine only likely pairings.

	F_1	F_2	F_3	F_4	F_5
F_1	0	150	250	150	250
F_2	250	0	100	0	100
F_3	150	300	0	300	0
F_4	250	0	100	0	100
F_5	150	300	0	300	0

Table 1: Angles in grads between features of object shown in Fig. 2.

Instead of exploring all pairings, geometrical constraints are used to eliminate those which make no sense. However, the method should require much less computing time than the rigorous solution of transforming the object to the image and checking if it matches. In our example angles are used as geometrical constraints. In Table 1 all angles between object features are listed. Table 2 contains the same for the data features.

	f_1	f_2	f_3
f_1	0	300	150
f_2	300	0	250
f_3	250	150	0

Table 2: Angles in grads between data features shown in Fig. 2.

As indicated in Fig. 3 data feature f_1 is assigned to every object feature F_i. Next, data feature f_2 is assigned to those object features which satisfy the angle constraint. From Table 2 we take the angle between f_1 and f_2 (300^g) and check in the appropriate rows of Table 1 for the same angle. For the assignment f_1 to F_3 two pairings satisfy the constraint. The same holds for f_1 to F_5. Now, only these pairings need be pursued further. Again, the assignment for f_3 is performed by considering the angular constraint $\angle(f_2, f_3) = 250^g$. We end up with the four sets of possible pairings for f_1, f_2, f_3,

namely, (F_3, F_2, F_1), (F_3, F_4, F_1), (F_5, F_2, F_1), (F_5, F_4, F_1). With additional constraints, for example, distances, the potential pairings could be further pruned.

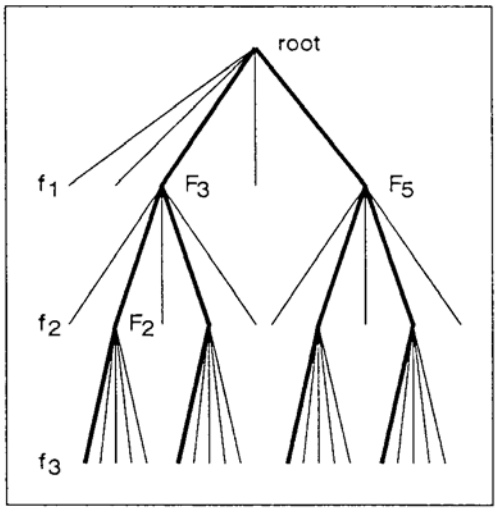

Figure 3: Correspondence problem cast as a tree search problem.

We note that the four solutions found are not globally consistent. The geometric constraints only assure local consistency. The angle constraint is a binary constraint. If enforced two consecutive nodes in the tree are consistent. A unary constraint would ensure a single node to be consistent. With unary and binary constraints the consistency between three or more nodes is not guaranteed. Therefore, the transformation of the object to image is necessary.
Every point in the discrete correspondence space constitutes a hypothesis for an assignment of data features to object features. Obviously, most assignments can be ruled out immediately. For example, two distinctly different data features cannot be assigned to one and the same object feature (assuming rigid objects). Unary and binary constraints further reduce the assignments to a feasible set.

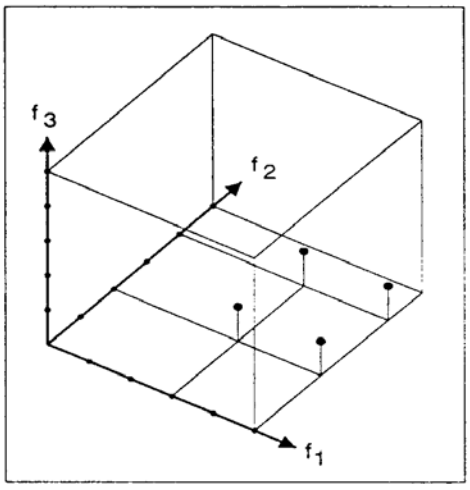

Figure 4: Correspondence space. Solid dots indicate possible correspondences found by applying geometric constraints.

As indicated in Fig. 3 the search for feasible assignments is cast as a tree search problem. On the first level of the tree, data feature f_1 is paired with all object features F_i, $i = 1,...n$, but only if unary

constraints are satisfied. An example of a unary constraint is the length. The same procedure is repeated on the next levels of the tree. Only those nodes are further expanded which satisfy the constraints. Every valid leaf node defines a path in the tree indicating feature assignments which must now be examined for global consistency.

3.2 Verifying hypotheses

Every leaf of the tree is a hypothesis about the correspondence of data features to object features. These hypotheses must be tested by transforming the object into the image. Global consistency is reached if all data features match the corresponding object features. Usually, there are more object features than data features. That is, there are some unpaired object features. They are now used to examine the gray levels for any evidence of data features which remained undetected in the feature extraction process.
In the general case of a 3-D transformation seven parameters must be determined. This transformation should be formulated in a general fashion; for example, it should not be restricted to points only, rather it should allow including straight line segments, curves or surface patches.
A hypothesis can be tested before the leaf of a tree is reached. In fact, as soon as enough data features are assigned to perform a 3-D transformation, it may well be advisable to verify the hypothesis immediately. If successful, the search in the correspondence space can be terminated. This variation of the general approach may be justified if the tree is very deep (e.g., many object features) or if a particularly salient matching is found. This method, termed local feature focus, was proposed by Bolles and Horaud (1986).

4. IMAGE UNDERSTANDING FOR AERIAL SCENES

The approaches to object recognition described so far are based on the assumption that objects can be described geometrically, for example, by defining their boundaries or surfaces. This may be impossible, however, if it comes to recognizing objects with complex shapes. Such is the case with aerial scenes. Even rather simple structures, such as buildings, come in such a variety of different sizes and shapes that it is a fruitless attempt to precisely describe and store all of them in a model library. For outdoor scenes other solutions must be found.

4.1 Context-based object recognition

Strat and Fischler (1991) describe a system, called Condor (context-driven object recognition), that deviates radically from the model-based paradigm. The authors argue that oftentimes objects of outdoor scenes can only be identified by taking their surroundings into account, because relationships among objects provide important clues. The notion of Condor is to embed objects in contexts rather than treating them as independent entities.
Condor is in an experimental stage. The goal is to recognize objects in natural scenes. The knowledge base is tailored for solving this task in a two-square mile area near the Stanford University campus with ground level images. As the authors put it, the recognition capability of Condor should equal that of a rabbit inhabiting the same environment.
At the heart of Condor is an object-oriented knowledge base, called core knowledge structure, that contains knowledge about the visual world. Spatial knowledge is represented as a multiresolution octree that allows recognizing objects at various scales. A semantic network is used for representing semantic knowledge at various levels of resolution.
Condor works like a production system. The main process types generate candidates, compare candidates, form and select cliques. The processes act as daemons and invoke themselves depending

on the contextual environment. The output of the system is a labeled 3-D model of the scene where a label is an object class. Examples are sky, foliage, raised object and ground.
Each class has an associated set of simple recognition procedures designed to work in specific contexts. An example is recognizing foliage which is very difficult considering the different situations and conditions. Thus, this general recognition task is divided into rather specialized subtasks like finding the silhouette of foliage against the sky, or finding foliage of one type of tree.
A context set is basically a rule. If the context is satisfied (e.g., sky is clear, camera is horizontal, color is available) an operator is invoked (e.g., segment blue regions) and the result forms a candidate hypothesis. Candidates are checked for global consistency in a process called clique formation. A clique is a set of mutually consistent candidate hypotheses. Inconsistencies of hypotheses are detected by context-specific procedures, again expressed as context sets. If the system labeled a region "ground", then it must satisfy the rule that it cannot extend above the skyline.

4.2 Function-based object recognition

Winston *et al.* (1984) point out that there are many different physical descriptions for objects, say, a cup. However, a single functional description can be used to represent all possible cups. Many man-made objects serve a specific purpose, and it may be possible to describe that purpose in a concise manner as functional descriptions which can be used together with other knowledge to recognize objects.
Stark and Bowyer (1991) describe a system to recognize chairs. The system takes 3-D polyhedral objects and recognizes whether the object belongs to category chair and, if yes, into which subcategory it falls. A first decision is made based on the size of the object.
First, all functional elements of an object are analyzed. Functional elements consists of (i) single surfaces, for example the seat of a chair, (ii) groups of surfaces serving one function, (iii) 3-D module of the structure. A function label (name of the functional property, e.g., *sittable surface*) is assigned to the functional elements.
Function labels are defined by procedural knowledge primitives. Stark and Bowyer use relative orientation, dimensions, stability, proximity and clearance as primitives. The dimension checks the size of surfaces if they are within reasonable sizes. The stability primitive checks for stable support by examining the contact points of the object with the ground plane.
A hierarchical tree represents the function-based descriptions. Associated to the nodes of the graph are frames with information about the name, type and functional plan. All function labels have specified constraint values.

4.3 Fusion of monocular cues

Over the past several years, considerable research was directed towards detecting buildings in aerial imagery. A number of interesting building detection techniques have been reported (see, e.g. Fua and Hansen (1989), Gülch *et al.* (1991), Harwood *et al.* (1987), Huertas and Nevatia (1988), Mohan and Nevatia (1989), Nicolin and Gabler (1987)). Several methods have been developed at the Digital Mapping Laboratory, Carnegy Mellon University. Shufelt and McKeown (1993) argue that no single detection method will correctly delineate buildings in every scene. Rather, the authors propose to combine the results obtained from different building extraction methods in what they call the cooperative-methods paradigm.
The idea is to use different methods to analyze the digital image for the occurance of buildings and then to combine the results (information fusion) in a generate and test hypotheses framework. The system described in Shufelt and McKeown (1993) uses four different building detection methods,

all of which are monocular. The first system analyses lines and corners by examining edges which might produce right angles. The sequence of corners defines boxes which are used as hypotheses for buildings. Since edge detection depends much on the gray level variations between buildings and their surroundings, the method fails if there is not enough contrast.
The other detection systems are based on the analysis of shadows cast by buildings. Edges comprising shadow regions are segmented into straight line segments, analyzed for corners and grouped to parallelograms which are then used as hypotheses for buildings. With the hypothesized buildings and known sun angle, cast shadows are computed and compared.
A simple information fusion schema takes all the building hypotheses generated by the individual extraction methods by converting the polygonal boundary descriptions into a raster system. This corresponds conceptually to overlaying each hypothesis and to check if the boundaries are in close proximity. The authors report that monocular fusion increases the detection of buildings from 58 % to 77 %, indicating that individual detection methods extract different information, which, when integrated, increases the performance.
The authors describe experiments with combining the monocular fusion results of the left and right image of a steropair. The process as described above is repeated for the right image and the results are registered to the left image by using an average elevation per region. With this technique the detection rate is further increased. One should note, however, that the method presented by Shufelt and McKeown is two dimensional; the extraction, in particular the analysis, is not carried out in 3-D. Thus, additional cues resulting from 3-D data, are missing.

5. CONCLUDING REMARKS

Image understanding has gained much attention from researchers in different fields, mainly in computer vision. Image understanding or scene interpretation, is very much application dependent. An important application is robotics, where most tasks require 3-D scene interpretation. For example, to pick a part out of a bin, or to navigate a robot through a cluttered environment requires scene interpretation. Examples for outdoor scenes is the navigation of autonomous land vehicles which includes road following capabilities as well as analyzing landscapes and terrain modeling.
Comparing to the examples mentioned, the interpretation of aerial imagery has received less attention. Most of the work reported concentrates on extracting buildings, airports, or road networks. No system reaches the maturity of, say, model-based object recognition. If the ultimate goal is to compile maps automatically, then we are far from having even experimental systems. A number of reasons account for this situation.

- most of the work of analyzing aerial scenes is conducted by researchers in computer vision (usually under the title *automatic cartography*). It seems that a joint effort with photogrammetrists and cartographers would substantially increase domain experience and application know how.
- most of the research is conducted on a "tool level", that is on the algorithmic and implementation level in Marr's terminology (see Section 2). The problem of making maps automatically must first be solved on the conceptual level, however. As strange as it may be, but we do not know exactly how to make maps. That is to say, we cannot explicitely describe the map making process. Consequently, the selection and design of algorithms is characterized by by trial and error.
- the attempt of producing maps automatically is typically driven by a bottom-up approach which begins with the raw image, proceeds to edge detection and image segmentation, and ends with generating and verifying hypotheses. The verification process, as well as the generation of

useful hypotheses about occurances of objects, should involve domain knowledge and existing information to a much greater extent.

- even though many papers report about non model-based object recognition, the problem of describing and representing aerial scenes with natural and man-made objects is far from being solved.
- in order to reduce the problem, most research in interpreting aerial scenes concentrates on one class of objects, such as, buildings, airports, or roads. Thus, interrelations between classes are neglected, and a rich source of knowledge about spatial and functional relationships cannot be utilized.
- additional spectral information, available in color photography, should be used.
- most, if not all aerial scene interpretation is performed in image space, although supported by depth information. It seems that recognition would benefit of being conducted in object space, especially when the gray levels are transformed into object space as well (orthophoto).

If the endresult of image understanding and object recognition in digital photogrammetry is a digital map, then additional aspects must be considered. Recognizing an object, say a building, is only the first step. The delineation or the representation of the object in the digital map, may add a plethora of new problems. If we think of medium or small scale applications, then generalization aspects come into play. Even for large scale applications, the boundaries of objects hardly ever coincide with intensity changes in the gray level image. Oftentimes, boundaries are subjective contours which may slightly deviate from their true positions because of cosmetic reasons. For example, buildings are squared and perhaps aligned to an easement, and roads are shown perfectly parallel.

6. REFERENCES

Fua, P., and A.J. Hanson, 1989. Objective functions for feature discrimination: Theory. Proc. DARPA Image Understanding Workshop, pp. 443-460.

Grimson, W.E.L., 1990. Object recognition by computer. The MIT Press, Cambridge, Massachusetts.

Hanson, A.R., and E.M. Riseman, 1978. Segmentation of natural scenes. In Hanson and Riseman (Ed) Computer vision systems, Academic Press, New York.

Harwood, D., S. Chang, and L. Davis, 1987. Interpreting aerial photographs by segmentation and search. Proc. DARPA Image Understanding Workshop, pp. 507-520.

Huertas, A., and R. Nevatia, 1988. Detecting buildings in aerial images. Comput. Vision Graphics Image Process., Vol. 41, pp. 131-152.

Lee, D.C., and T. Schenk, 1992. Image segmentation from texture measurement. Int. Archives of Photogrammetry and Remote Sensing, Congress Washington D.C., Comm. III.

Marr, D., 1982. Vision. W.H. Freeman and Company, New York.

Mohan, R., and R. Nevatia, 1989. Using perceptual organization to extract 3-D structures.

Nicolin, B., and R. Gabler, 1987. A knowledge-based system for the analysis of aerial images. IEEE Trans.Geosci.Remote Sensing, GE-25, pp. 317-329.

Shufelt, J., and D. McKeown, 1993. Fusion of monocular cues to detect man-made structures in aerial imagery. CVGIP: Image Understanding, Vol. 57, No. 3., pp. 307-330.

Stark, L., and K. Bowyer, 1991. Achieving generalized object recognition through reasoning about association of function to structure. IEEE Trans. Pattern Recognition and Machine Intelligence, PAMI, Vol. 13, No. 10, pp. 1097-1104.

Strat, T.M., and M.A. Fischler, 1991. Context-based vision: recognizing objects using information from both 2-D and 3-D imagery. IEEE Trans. Pattern Recognition and Machine Intelligence, PAMI, Vol. 13, No. 10, pp. 1050-1059.

Winston, P.H., T.O. Binford, B. Katz, and M. Lowry, 1984. Learning physical decription from functional definitions, examples, and precedents. In Proc. Int. Symp. Robotics Research, The MIT Press, vol. 1.