Efficient Recognition of 3-D Rigid Solid Objects from 2-D Projective Images

T. Gevers and A.W.M. Smeulders

Faculty of Mathematics & Computer Science, University of Amsterdam

Kruislaan 403, 1098 SJ Amsterdam, The Netherlands

E-mail: gevers@fwi.uva.nl

Abstract

In this paper, we address the problem of recognizing 3-D rigid, solid objects in cluttered scenes from 2-D grey-value images obtained from arbitrary unknown viewpoints. The task of the recognition method is to find instances of known object models in projective images. Geometrical properties of solid objects are used which are invariant to a change in the point of view. To obtain these projective invariants, the classical projective geometry has been taken as starting point. It is classically known that the cross ratio is projectively invariant. The invariants describing the model can be used as "keys" to index the model into a hash table. In this way, a table of object model is constructed where each model is indexed according to its key(s). Because these keys remain the same under the specific transformation they are computed for objects found in an image and used to lookup models to determine which object is present in an image.

We propose an efficient method to recognize 3-D rigid, solid objects from 2-D projective images in the presence of object overlapping and occlusion which is robust to noise, location accuracy and able to deal with multiple instances of a model in a scene.

Experiments show excellent performance and together with the inherent parallelism of the recognition method makes the method a promising one.

keywords: 3-D solid objects, 2-D projective images, projective invariants, object recognition, stability, efficient indexing and retrieval.

1 Introduction

In a general context, object recognition involves the task to identify correspondence between a 3-D object model and some part of a 2-D grey-value image taken from an arbitrary viewpoint of a cluttered real-world scene containing the object.

Because the physical world is three dimensional and an image contains a two-dimensional projection of the reality, it is a key problem to infer 3-D information and to recognize 3-D objects from 2-D projective images because the projection looses some 3-D information.

Many practical object recognition systems are model-based [2], [4], [9], [12]. To succeed they address two major interrelated problems: object representation and object matching. The representation should be good enough to allow for reliable and efficient matching. The recognition consists of matching the stored models, encapsulated in a representation scheme, against the image to determine which model corresponds to which portion of an image. Several systems have been developed to deal with the problem of model-based object recognition by solving the correspondence problem by tree search [7]. However, the computational complexity is exponential for nontrivial images, because there are infinitely many points of view from which the image is obtained.

In this paper, we propose a representation scheme and an object recognition method based on projective invariant shape descriptors for efficient recognition of 3-D objects from a 2-D projective image. We address the case where neither the models nor the image data have to be complete. Also the viewpoint is unknown and arbitrary and the objects are rigid, solid 3-D objects (e.g. polyhedra, blocks and some industrial parts).

The paper is organized as follows. In section 2 a projective invariant representation scheme is proposed for 2-D planar objects and in section 3 a hash table is suggested to provide the essential mechanism for efficient indexing and retrieval. A method for the recognition of 2-D planar objects from 2-D projective images is proposed in section 4 and in section 5 the method is extended to recognize 3-D objects in 2-D projective images. The implementation of the method is presented in section 6 and experiments carried out with the proposed recognition method are discussed in section 7.

2 Projective Invariant Representation Scheme for 2-D Planar Objects

For our purpose, the issue arises which geometrical properties of the shape of a planar object are invariant under a change in the point of view. Although, the shape of an object itself is independent of the point of view from which it is seen, its visible image is viewer-dependent. For example, the parallelism of lines is not necessarily invariant under perspective projection because the projective image is changing along with the change of position and orientation of the object. Therefore, in order to recognize objects present in the scene, a search has to be made in parameter space to determine the relative point of view. This very time consuming. Using invariants can eliminate the search and reduce the necessary information needed in the representation scheme.

The projective invariance of cross ratios and its generalization to cross ratios of areas of triangles and volumes of tetrahedra has been proven by Möbius [10] as early as 1827. We use cross ratios as the invariants of the perspective projection in the representation scheme for 2-D planar objects.

From the classical projective geometry we have the following theorem [5], [13]:

- Theorem 1: Consider an object O in R^3, five points of which lie in a plane. Let A, B, C, D and E be that five points, no three collinear; let A_1, B_1, C_1, D_1 and E_1 be their images on the image plane. Select one point, A, and extend rays from A to each point B, C, D, and E. Let $\theta_1, \theta_2, \theta_3$ be the the angles at A between AB and AC, AC and AD, and AD and AE respectively. Let θ_i^1 be the corresponding angles in the image plane. Then:

$$\frac{\sin(\theta_1+\theta_2)\cdot\sin(\theta_2+\theta_3)}{\sin(\theta_2)\cdot\sin(\theta_1+\theta_2+\theta_3)} = \frac{\sin(\theta_1^1+\theta_2^1)\cdot\sin(\theta_2^1+\theta_3^1)}{\sin(\theta_2^1)\cdot\sin(\theta_1^1+\theta_2^1+\theta_3^1)} \quad (1)$$

independent of the projection viewpoint.

So the cross ratio of sines between five points in a plane is projective invariant. Because the cross ratios express the vertex configuration with respect to one another, we describe the shape of a five point planar object in 3-D space using cross ratios from a single projective image.

For 2-D planar objects consisting of more than five points, we use the boundaries as their natural description. For each object, a polygon is obtained by approximating the boundary. Because the polygonal approximation depends on the line fitting tolerance, the result is not uniquely defined. In addition, for a practical situation, noise in the image formation will also introduce uncertainty in the approximation. Therefore, we employ several polygonal approximations simultaneously for different line fitting tolerances. In this way, a robust planar object representation is obtained which allows for recognition of objects at different scales.

2.1 Projective Invariant Shape Description of 2-D Planar Polygons

The goal is now to compute a projective invariant representation of the polygons describing the objects.

Planar 2-D object models are described by their polygonal approximations. We assume that polygons approximating the planar objects are given in terms of their vertices in a counterclockwise order. Polygons are considered of at least five vertices.

Let P be a set of polygons. $P_i \in P$ denotes a polygon with m_{P_i} vertices $\{(v_1^{P_i}, v_2^{P_i}, ..., v_{m_{P_i}}^{P_i})\}$, $m_{P_i} \geq$

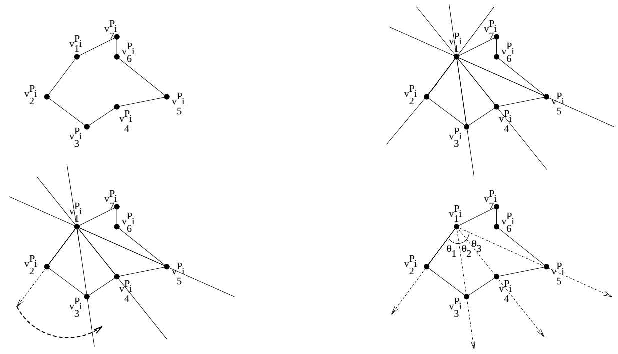

Figure 1: Definition of the projection lines.

5. We calculate the cross ratio of the vertices $v_j^{P_i}$, $v_{[j+1]}^{P_i}$, $v_{[j+2]}^{P_i}$, $v_{[j+3]}^{P_i}$ and $v_{[j+4]}^{P_i}$ where the notation $[k] = k \bmod m_{P_i}$, that is vertices are indexed modulo m_{P_i} to account for the cyclic nature of polygons.

The method to compute the cross ratio $c_j^{P_i}$ of vertex j is as follows:

- Extend straight lines going through $v_j^{P_i}$ and each of the following four vertices $v_{[j+1]}^{P_i}$, $v_{[j+2]}^{P_i}$, $v_{[j+3]}^{P_i}$ and $v_{[j+4]}^{P_i}$.
  If two lines overlap, the cross ratio for vertex j is undefined.

- Take the line from $v_j^{P_i}$ to $v_{[j+1]}^{P_i}$ as the first projection line. Rotate the first projection line in a counterclockwise manner. The first three lines encountered in this way are regarded as the other three projection lines, see Fig. 1.

- Calculate the angles between the four projection lines and compute $c_j^{P_i}$, as in formula (1).

By applying the algorithm to each vertex of P_i, we obtain an ordered set C_{P_i} with m_{P_i} cross ratios:

$$C_{P_i} = \{(c_1^{P_i}, c_2^{P_i}, ..., c_{m_{P_i}}^{P_i})\} \quad (2)$$

The set is projective invariant apart from its modulo order. It reflects the configuration of the vertices. It gives a description of the shape of the object enabling shape distinction with other objects.

3 2-D Projective Hashing

Now, we consider a model set O. For each object $O_i \in O$, polygons $P_i \in P$ are computed. In turn, polygons P_i are converted to ordered lists C_{P_i}.

When the data base of models is converted into their cross ratio representation, attention shifts to establishing correspondence between a model and the projective image.

We address the object recognition problem in a realistic environment. The object may be partially occluded, multiple instances of a model may be present in the image and a large number of object models is allowed. To reduce computational complexity, a hash table provides efficient indexing and fast retrieval.

3.1 The Hash Table

Given is a set of M planar polygons $P = \{P_1, P_2, ..., P_M\}$, where each $P_i \in P$ is given in terms of their cross ratios C_{P_i}. We use the code as a key for the hash table where we record P_i as an entry. Let $f()$ be defined as an indexing function. For each address $f(c_j^{P_i})$ we once record the polygon P_i as a slot.

The compilation of $P_i \in P$ into the hash table can be executed off-line because it is dependent on models only. New polygons may be added independently without recomputing the hash table.

4 The 2-D Recognition Process

For the recognition of 2-D planar objects in R^3 from 2-D images images by using the prepared hash table, the recognition method consists of two stages:

- (1) compute candidate solutions by fast filtering.
- (2) verify each candidate solution against the corresponding model.

4.1 Generating Candidate Solutions

First, the Canny edge detector [3] is applied on grey-value images holding the unknown object's image (any other edge finder will do). Then a line fitting algorithm, [11], is executed on the edge image to compute the polygonal approximations. For each polygon extracted from the image the sequence of cross ratios is computed.

Suppose the image contain N different polygons $U = \{U_1, U_2, ..., U_N\}$, where each $U_j \in U$ have n_{U_j} vertices represented by their cross ratios:

$$C_{U_j} = \{(c_1^{U_j}, c_2^{U_j}, ..., c_{n_{U_j}}^{U_j})\} \tag{3}$$

The recognition problem is to determine which $U_j \in U$ is an instance of which $P_i \in P$ where extraneous or missing vertices are admissible in U_j. Stated differently, the problem is to determine the number of similar cross ratios between P_i and U_j preserving the sequence order, with extraneous or missing vertices allowed.

For each $c_k^{U_j}$ of $U_j \in U$ the entry address in the hash table is $f(c_k^{U_j})$. Then a vote is given to each model at that entry. If a model P_i lies unoccluded in the image, the polygon will get at least m_{P_i} votes. Even when k vertices are missing, P_i receives at least $m_{P_i} - k$ votes. Thus, a given vote contributes to the presence of a certain model. A vote accumulator looks for pairs (P_i, U_j) that received more votes than a certain threshold, relative to the number of vertices m_{P_i} of P_i. Let t_r be a relative threshold and P_i has m_{P_i} vertices (i.e. cross ratios), then the voting threshold t_v is defined as:

$$t_v = t_r \cdot m_{P_i} \tag{4}$$

where t_r is the tolerated occlusion, interpreted as the percentage vertices out of sight.

For P_i some detected polygons surpass the threshold generating in total k candidate solutions $H^{P_i} = \{h_1^{P_i}, h_2^{P_i}, ..., h_k^{P_i}\}$, $k \geq 0$. It may so that some $P_i \in P$ have generated no candidate solutions, others one or more. If $k = 0$, the number of votes indicates no presence of a model.

4.2 Verifying Candidate Solutions

The candidate solutions are verified to be an instance of P_i.

The verification of $h_j^{P_i}$ and P_i is based on the minimum least square match between the cross ratio sequences of U_j and P_i. Only those cross ratios of U_j which have been voting during the voting scheme will contribute to the least square match.

We define the dissimilarity of two given cross ratios c_q and c_r in the following way:

$$d(c_q, c_r) = (c_q - c_r)^2 \tag{5}$$

The minimum least square match $MLSM()$ between P_i and U_j of $h_j^{P_i}$ is defined as:

$$MLSM(P_i, U_j) = \min_q \{\frac{1}{\gamma}\sqrt{\sum_{r=1}^{n_{U_j}} d(c_{[q+r]}^{P_i}, c_r^{U_j})}\} \tag{6}$$

and $[q + r] = (q + r) \bmod m_{P_i}$ again, where $q = 1, ..., m_{P_i}$ to make the matching independent of the first vertex for computing the cross ratio sequences. γ denotes the total number of votes of U_j voting for P_i.

4.3 Noise Analysis

One type of error introduced in the recognition phase is the ill positioning of the polygon vertices due to measurement error. In turn, this leads to an error in the cross ratio. The sensitivity of the cross ratio to image noise depends largely on the configuration of the vertices with respect to one another. Thus, the error of the cross ratio is not only based on the strength of the noise but also on the shape and scale of the polygon considered in the cross ratio computation.

Assume A, B, C, D and E are five points as defined as in Fig. 2, taken with respect to A.

In polar coordinates (r, θ) in the presence of noise, we assume that the points falls within a maximum circle of radius R_ϵ. The tolerance of points as measured is Δr. For the distance we find $r_\epsilon \in (r - 2R_\epsilon, r + 2R_\epsilon)$. The upper bound on the variation in θ is given by [8]:

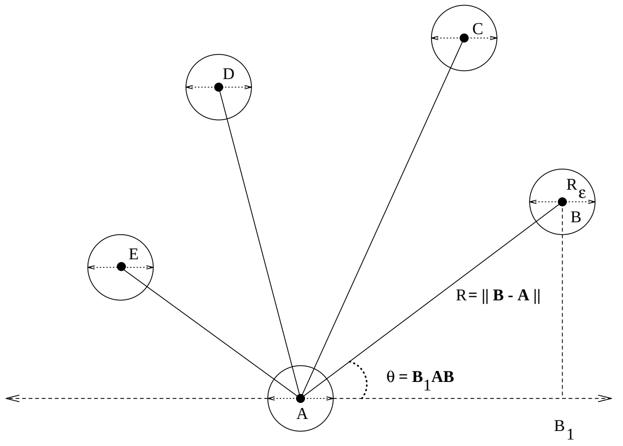

Figure 2: Noise Analysis

$$\Delta\theta \leq 2R_\epsilon(\frac{1}{l} + \frac{1}{r}) \qquad (7)$$

Let X be the noise free version of a cross ratio, then we need to calculate ΔX, the possible range of cross ratio values. Because the displacement of vertices affects the computation of the cross ratio X through their three angles, we have the following formula:

$$\Delta X = \frac{\sin((\theta_1 \pm \Delta\theta_1) + (\theta_2 \pm \Delta\theta_2)) \cdot \sin((\theta_2 \pm \Delta\theta_2) + (\theta_3 \pm \Delta\theta_3))}{\sin(\theta_2 \pm \Delta\theta_2) \cdot \sin((\theta_1 \pm \Delta\theta_1) + (\theta_2 \pm \Delta\theta_2) + (\theta_3 \pm \Delta\theta_3))} \qquad (8)$$

where $\Delta\theta_i$ is defined as the range of maximum angle variations and ΔX is the possible range of cross ratio values.

In the recognition phase, we vote for all hash entries lying in a neighborhood of $f(\Delta X)$ instead of voting for all pairs (model, planar section) at one hash or address as we did in the noise free voting scheme. By accessing all these entries $f(\Delta X)$, it is assured that voting for the correct solution is not lost. Although the correct solution is not lost, the need to access a neighborhood of hashes yields an increased number of candidate solutions affecting the computational efficiency of the filtering technique. These false matches will be discarded in the verification step.

5 Recognition of 3-D Objects from 2-D Images

So far, we have discussed the case where the object models and the images have been given in 2-D. We now consider the recognition problem of 3-D objects from 2-D photographic images. The additional problem is the reduced dimension of the image space with respect to the model space. One way to cope with this problem is based on the observation that many 3-D objects contain planar planes. The extension of 2-D to 3-D object recognition is simply the description of a 3-D model into a number of polygonal approximations of the planes the model consists of.

The 3-D models can be acquired manually by a CAD/CAM system or automatically by taking each plane of the 3-D object model from a 2-D image from a single or different perspective viewpoints. Notice that a plane as discussed above need not be a surface of a 3-D object which is physically planar, but rather be a planar section.

6 Implementation

The proposed method to recognize 3-D solid objects from 2-D projection images has been implemented.

The recognizer takes a 2-D grey-value image and a set of 3-D models as an input. The output is a correspondence between a part of an image and a model producing that portion of the image.

Because the method is based on boundary information, we assume that the objects are distinguishable and identifiable by their boundaries or edge contours. Further, there are no major constraints on the imaging process (perspective projection), the objects (3-D solid objects without smoothly changing surfaces identifiable by their edge contours) and scenes (very cluttered physical 3-D world with object occlusion and object overlapping).

The recognition method has been implemented in C on a SPARC-10 station with UNIX as operating system. The model representation stage consists of describing object models by means of a CAD/CAM system, computing the cross ratio sequences of each planar section, encoding them and entering the model in the hash table.

After image acquisition, a number of image processing routines were applied on the raw image to obtain the polygonal vertices. In the first step, edges are extracted from the images using the Canny edge detection operator [3]. The output is a binary image indicating the presence or absence of an edge at each pixel. A line fitting technique [11] is applied to group edges to obtain polygonal approximations.

The key operators of the recognition phase is accessing the hash table and the voting. Therefore, the hash table and accumulator should be in the main memory. The accumulator counts the number of votes for a certain (model, planar section) in the 3-D case. Hence, the memory requirements are $M \cdot PS$ bytes, where M is the number of object models and PS is the average number of planar section per object model and for each (model, planar section) we need 1 byte. The memory requirements for the hash table is $b \cdot M \cdot PS$ bytes, where b is the number of buckets of the hash table hb. Thus, the memory requirements of both the accumulator as well as the hash table are not a major problem even for a large database and can fit easily in the main memory of a SPARC station.

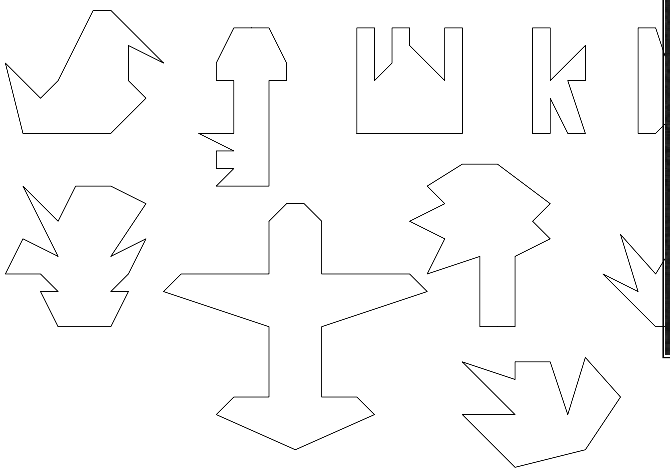

Figure 3: The set of polygons from top left to bottom right: P_1, P_2,...,P_{10}.

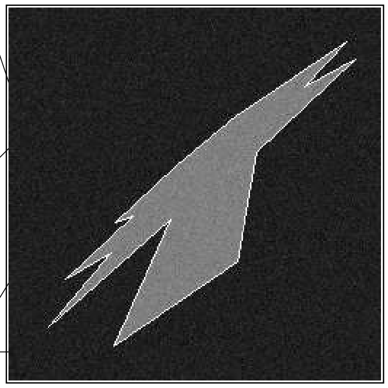

Figure 4: The projective image

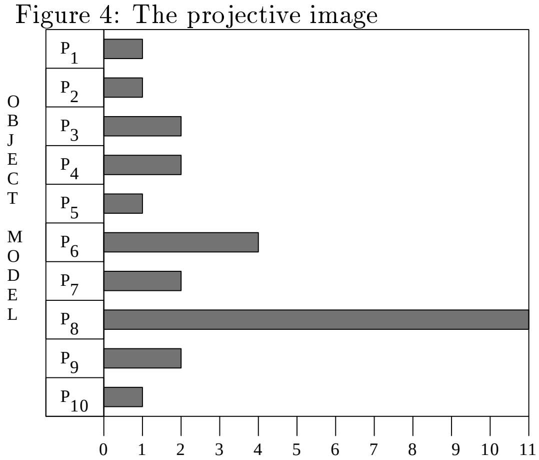

Figure 5: The number of votes.

7 Experimental Results

Experiments have been carried out on 2-D polygonal objects as an example of one of the applications on which the proposed object recognition can be applied.

7.1 Polygonal Objects

Several methods have been proposed to the problem of polygonal object recognition in 2-D images under perspective projection [1], [6]. Although they all consider an orthographic projection of the image, the worst case complexity is exponential.

In order to investigate the complexity and filtering ratio of the voting scheme, the typical hash element distribution in the hash table and the recognition capability of the method, experiments have been performed with a set of polygonal objects, see Fig. 3. The polygons have been encoded and entered into the hash table.

Then we took a projective image of one of the polygons, see Fig. 4.

From this image we extracted the polygonal vertices and calculated the cross ratios. The aim was the recognition of the isolated objects viewed with the optical axis not parallel to polygonal surface normal. The cross ratios of the image object were used to check the appropriate entry in the hash table, and for every model that appears there a vote has been given. In Fig. 5 the number of votes for each model is shown.

Obviously, there is one object model which have been given a distinctive number of votes which appears to be the model from which the image has been taken from. Therefore, the filtering ratio is satisfactory. The minimum least square match confirms the correspondence of the image object and the corresponding object model. The user time of the the recognition of the image object with the small database of 10 polygonal models on a SPARC-10 station was 0.4 seconds.

8 Conclusions

In this paper, we addressed the problem of identifying a correspondence between a 3-D object model and some part of a 2-D projective image.

A representation scheme and an object recognition method have been proposed which enable efficient recognition of a 3-D object from a 2-D projective images taken from an arbitrary viewpoint of a cluttered real-world scene containing the object.

The models are three dimensional and are represented by the set of polygons, approximating each independent boundary of a planar section of the three dimensional object. Each polygon has been represented as a sequence of projective invariant shape descriptors, yielding a projective invariant representation scheme of an object model. Model polygons are then encoded. The code is used as a key for the hash table where the specific polygon is recorded as an entry. The hash table provides the essential mechanism for efficient indexing and fast retrieval. The encoding and storing of the database of three dimensional object models is done off-line. The on-line object recognition consists of segmenting the image into polygons, encode them and use them as a key to vote for the model to be present at the corresponding entry. Finally, the verification step consists of a least square match between the instance and the model.

Excellent performance is shown, even in noisy images, by the experiments carried out on a database of 2-D polygonal objects.

References

[1] Ballard, D. H. and Sabbah, D., *Viewer Independent Shape Recognition* , IEEE Trans. on Pattern Analysis and Machine Intelligence, 5(6), pp. 653-660, 1986.

[2] Besl, P. J. and Jain, R. C., *Three-Dimensional Object Recognition* , ACM Computing Surveys, 17(1), pp. 75-154, 1985.

[3] Canny, J., *A Computational Approach to Edge Detection*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 8, No. 6, pp. 679-698, 1986.

[4] Chin, R. T. and Dyer, C. H., *Model-Based Recognition in Robot Vision* , ACM Computing Surveys, 18(1), pp. 67-108, 1986.

[5] Coxeter, H. S. M., *Projective Geometry* , Univ. of Toronto Press, Toronto, 1974.

[6] Grimson, W. E. L. and Lozano-Perez, T., *Model Based Recognition and Localization from Sparse Range or Tactile Data* , Int. J. Robotics Research, 3(3), pp. 3-35, 1984.

[7] Grimson, W. E. L., *Object Recognition by Computer-The Role of Geometric Constraints* , Cambridge, MA: MIT Press, 1990.

[8] Grimson, W. E. L. and Huttenlocher, D. P., *On the Sensitivity of the Hough Transform for Object Recognition* , MIT AI Memo No. 1044, 1988.

[9] Kalvin, A., Schonberg, E., Schwartz, J. T. and Sharir, M., *Two-dimensional, model-based, boundary matching using footprints* , Int. J. Robotics Research, 5(4), pp. 38-55, 1986.

[10] Möbius, A. F., *Der Barycentrische Calcul* , Verlag von Johann Ambrosius Barth. Leipzig, 1827; in A. F. Möbius, "Gesammeltewerke, Vol. 1," Dr. Martin Sandig, OHG, Wiesbaden, 1967.

[11] Sharaiha, Y. M. and Christofides, N., *An Optimal Algorithm for the Straight Segment Approximation of Digital Arcs* , CVGIP: Graphical Models and Image Processing, Vol. 55, No. 5, pp. 397-407, 1993.

[12] Thompson, D. W. and Mundy, J. L., *Three-Dimensional Model Matching from an Unconstrained Viewpoint* , Proc. of the IEEE Int. Conf. on Robotics and Automation, Raleigh, North Carolina, pp. 208-220, 1987.

[13] Veiblen, O. and Young, J. W., *Projective Geometry* , Ginn. Boston, 1910.