A 3-D RECOGNITION AND POSITIONING ALGORITHM USING GEOMETRICAL MATCHING BETWEEN PRIMITIVE SURFACES

O.D. PAUGERAZS and M. HEBERT
I.N.R.I.A
Domaine de Voluceau, Rocquencourt
B.P. 195, 79133 LE CHESNAY CEDEX
FRANCE

ABSTRACT

In this paper, we describe an efficient algorithm for 3-0 scene anal>sis. This algorithm uses a segmentation of the surfaces to be identified into geometrical primitives, the original data being obtained by a laser range finder. Moreover, the algorithm estimates precisel> the location and orientation of an identified object of the scene. Results are presented on real objects.

I INTRODUCTION

In applications of Scene Analysis techniques to Robotics it is interesting not only to recognize objects in the scene but also to estimate as accurately as possible their position and orientation. In both cases symbolic descriptions of objects must be available. Such descriptions are usually obtained by segmenting the scene into regions homogeneous with respect to some criterion and representing the result as a graph where nodes are regions labelled with features and arcs are relations between regions. In many cases the relations are topological (neighbor, inside....) and sufficient for recognition but. not for positioning. In order to achieve recognition and positioning we need to introduce the constraint of rigidity which is global and therefore not very well suited to the graph representation. Our representation is purely geometrical in the sense that objects are segmented into regions approximated by simple parameterized primitives such as planes. and quadiics [l-5l which implicitely embed the rigidity constraint. The geometrical matcher presented here is part of a general 3-D Vision system including automatic data gathering, segmentation, model construction and Scene Analysis.

II GEOMETRICAL MATCHING

A geometrical description G of an object is a set $(P(U_i))_{i=1..I}$ of regions which are approximations of geometrical primitive surfaces, (e.g. planes or quadrics), U_i being the parameter vector of the i^{th} primitive. A matching between two descriptions G and G' is a set of pairs $(P(U_i),P'(U'_j))$ where P and P' are primitives of G and G' respectively. The parameters U_i and U'_j are not expressed in the same coordinate system because G' and G are usually the descriptions of a model and of the observed scene, respectively. More precisely, the parameters of G' are calculated in an object-centered frame while those of G are calculated in a viewer-centered frame. We shall say that the matching M is consistent if the two paired sets of primitives describe the same object. Since we are dealing with rigid solids, the matching is perfectly consistent if and only if there exists a rigid transformation T which maps each primitive P' of G' onto its corresponding primitive P with respect to the matching M. This transformation T provides the orientation and location of the identified object of the scene.

The basis of the recognition algorithm is a measure of consistency which allows us to control a tree search procedure. This measure must be compatible with the condition of perfect consistency as defined above. So, we define our consistency measure by :

$$(1) \quad g(M)=\min_T \sum \|U_i - T(U'_j)\|^2$$

where $T(U'_j)$ is the vector of the parameters of the the primitive $P'(U'_j)$ transformed by T.

This measure of consistency involves the estimation of the best transformation defined by the matching in the least-squares sense. The algorithms which perform this estimation are presented in part III.

In our case, we use a simple geometrical description in which the scene is described by a list of almost planar regions with the parameters of their fitting planes. These descriptions are obtained by algorithms of segmentation [5-7] which use the 3-D data obtained by a laser range finder [7]. Moreover, the acquisition system provides a complete model of each object which is a segmentation of the whole surface of the object into planar regions. Figs.2 and 3 show two examples of segmentation, each shade of gray corresponding to a planar region identified by the algorithms of segmentation in a view of the object of fig.1. Figs.2 and 3 correspond to two different methods of segmentation which are called the "Hough transform" and the "region growing method" [5-7] respectively. The reference model of the complete object is obtained in the same way and is shown on fig.4.

III ESTIMATION OF THE BEST TRANSFORMATION

In this part we present an algorithm which computes the best estimate of the transformation T of relation (1). A plane P is represented by two parameters w and d where w is the unit vector normal to P and d is the signed distance of the plane P to the origin O. Since the surfaces that we have segmented are boundaries of solids, we assume that w is always oriented to the outside of the object. For computational purposes, it is convenient to associate to P a point M of P, therefore : d=w.OM (where "." is the inner product).

Let T=t*R be a transformation with its associated translation and rotation. The notation "t*R" indicates that we apply the rotation first. This decomposition is not unique in general but we assume that the axis of rotation contains the origin of the reference coordinate system and this condition implies the uniqueness of the decomposition. When we apply T to the plane P(w,d) we obtain a new plane $P_1(w_1,d_1)$ where :

$$(2) \qquad w_1=Rw \quad \text{and} \quad d_1=w_1.t+d$$

If $M=(P(v_i,d_i),P'(v'_i,d'_i))_{i=1..N}$ is a matching with N pairs, we seek t and R which minimize the sum (1) which is given by

$$(3) \qquad \sum_i \|v_i-Rv'_i\|^2+W.\left|d_i-d'_i-v_i.t\right|^2$$

where W is a weighting factor. This sum can be decomposed in two terms, a sum over $\|v_i-Rv'_i\|^2$ which allows us to determine the best rotation and a sum over $\left|d_i-d'_i-v_i.t\right|^2$ which determines the translation. We present the two algorithms used for minimizing these two sums.

Estimation of the rotation

The problem of the estimation of the rotation is the most difficult because a rotation cannot be linearly represented by natural parameters such as the Euler's angles and classical least-square methods cannot be directly applied.

The first natural representation of rotations is the composition of three elementary rotations around the three axis of coordinates.

The second representation is obtained by using the axis w and the angle r of the rotation, the rotation is then given by the Euler formula :

$$(4) \quad Rv=v+(1-\cos(r))w\wedge(w\wedge v)+\sin(r)w\wedge v$$

Where $\wedge$ is the outer product. In both cases, we have to minimize an expression of the form :

$$(5) \qquad F=\sum_i \|v_i-f(p,v'_i)\|^2$$

where p is the vector of the parameters of R and f is a non-linear function.

The resolution of this minimisation problem uses gradient-like algorithms whose convergence is not ensured. At last, the rotation can be represented by a 3 by 3 matrix R subject to the constraint $R^tR=Id$, this leads to an iterative minimisation of a quadratic criterion subject to six quadratic constraints.

We present now a representation which seems to be the simplest one according to the minimization problem and uses the notion of quaternion. A quaternion q is a pair (v,s) where v is a vector of E3 and s is a scalar (many other definitions exist, see [8]). The set of quaternions H is isomorphic to E4 and has a stucture of non-commutative algebra, the product of two quaternions being defined by :

$$(6) \quad (v,s) * (v',s')=(v\wedge v'+s'v+sv',ss'-v.v')$$

In the sequel, we identify the vectors v of E3 with the quaternions (v,0) and the scalars s with (0,s). The conjugate $\bar{q}$ of a quaternion is : $\bar{q}=(-v,s)$ if q=(v,s) and its norm is given by $\|q\|^2=q * \bar{q}=\|v\|^2+s^2$. An important property of this norm is that it is "multiplicative" :

$$(7) \qquad q_1*q_2 = |q_1| \cdot |q_2| \cdot$$

Notice that if q is regarded as a vector of E4, $|q|$ is its euclidian norm.

The quaternions can be regarded as a generalization of the complex numbers, the vector v in q corresponding to the complex part of a+ib. Since the complex numbers of module 1 represent the rotations of E2, it is natural that the same property holds in H. More precisely, there exists an inomorphism h from the group of rotations of E3 to the group UH/Eq where UH is the group of the quaternions of unit norm and Eq is the equivalence relation :

q Eq q' iff q=-q' or q=q'.

Moreover if R is a rotation and q is a element of h(R), then the following relation holds for every vector u in E3 :

$$(8)\ Ru=q * u * \bar{q}\ .$$

(the vector u and the quaternion (u,0) are identified in this relation.)

This property means that rotations and normalized

The equivalence relation Eq reflects the fact that a rotation has two axis/angle representations, (w,r) and (-w,-r). The mapping h is very simple. Let R be a rotation of axis w and angle r, the associated quaternions are :

$$(9)\ q=(\sin(r/2)w,\cos(r/2)) \text{ and } -q.$$

In the sequel, we only consider the determination of h(R) such that cos(r/2) is positive and for notation convenience, the vectors of E3 or E4 are identified with 1-row matrices. Notice that the rotation of null angle is the quaternion (0,1).

Conversely, if q=(v,s) is a quaternion of UH such that s is positive then the corresponding rotation is given by :

$$(10)\ r=2.\mathrm{Arcos}(s) \text{ and } w=v/\sin(r/2) \text{ if } r\neq 0.$$

Our minimization problem (5) is now a minimization over the set UH, the expression to be minimized is :

$$(11)\ P=\sum_i \|v_i-q*v'_i*\bar{q}\|^2\ .$$

This new form is legal because the norm of the quaternions is an extension of the euclidian norm of E3. Relation (7) can be applied by right multiplying the expression (11) by q which is of unit norm, so expression (11) becomes :

$$(12)\ P=\sum_i \|v_i*q-q*v'_i\|^2\ .$$

Therefore, the problem is to minimize (12) subject to the constraint q =1. This leads to a simple eigenvalue problem because the expression $v_i*q-q*v'_i$ is a linear function of the 4-vector q, so it exists a 4 by 4 matrix A_i such that :

$$(13)\ v_i*q-q*v'_i=q.A_i\ .$$

(q is now a 1 by 4 matrix.) therefore the criterion P of expression (11) is a quadratic criterion :

$$(14)\ P=\sum_i q.A_i.A_i^t.q^t=q.B.q^t$$

where $B=\sum_i A_i^t.A_i$ is a symmetric matrix.

So, we have to minimize P subject to the constraint $\|q\|=1$. This is a classical problem, [9] and P_{min} is the smallest eigenvalue of B. The solution q_{min} of (5) is the eigenvector of unit norm and of positive fourth coordinate corresponding to the eigenvalue P_{min}.

Thus, we have reduced our initial problem to the computation of the eigenvalues of a symmetric matrix of dimension four. In order to complete the algorithm, we have to compute the matrix B which is defined by relation (14). We first compute the matrices A_i of relation (13).

Let q=(w,s) be a quaternion, and v and v' two vectors of E3, if q'=v*q-q*v' then :

$$(15)\ q'=((v'+v)\wedge w+s(v-v'),w.(v'-v))$$

a matrix u^o is associated to each vector u of E3, this matrix is defined by :

$$(16)\ u\wedge u'=u'.u^o \text{ for all } u' \text{ of E3.}$$

Moreover, if $u=(u_1,u_2,u_3)$ then :

$$(17)\qquad u^o=\begin{bmatrix}0\ , & u_3, & -u_2\\ -u_3, & 0 & ,u_1\\ u_2, & -u_1, & 0\end{bmatrix}$$

we derive from relation (4) and definition (16) that the matrix A such that q'=q.A is :

$$(18)\ A=\left[\begin{array}{c|c}0 & (v-v')\\ \hline (v'-v)^t & (v'+v)^o\end{array}\right]$$

so, the matrix B of (14) is the sum of the matrices $C_i=A_i^t.A_i$ (19) which are easily derived from relation (18).

The matrix R_{min} corresponding to the quaternion $q_{min}=(w,s)$ is given by :

$$(20)\ R=(Id+(1-\cos(r))w^{o2}+\sin(r)w^o)^t$$

<u>Estimation of the translation</u>

We have to determine the vector t which minimizes the sum :

$$(21) \quad S = \sum_i |d'_i - d_i + w_i . t|^2 \quad .$$

Where d'_i, d_i and w'_i are the parameters of the planes of the matching M of relation (3). The minimization of S is a simple least-squares problem and is solved by the pseudo-inverse method :

Let N be number of pairs , A the N by 3 matrix : $\left[w_1,\ldots,w_N\right]^t$ and Z the N-vector : $\left[d'_1-d_1,\ldots,d'_N-d_N\right]^t$ then :

$S = \|Z - At\|^2$.

And the estimate of the best translation according to the criterion (21) is :

$$(22) \quad t_{min} = (A^t A)^{-1} A^t Z \quad .$$

And the error is given by :

$$(23) \quad S_{min} = Z^t (Z - At_{min}) \quad .$$

The estimation of the translation completes the algorithm of estimation of the best transformation relative to a matching M. The corresponding consistency measure g(M) is given by :

$$(24) \quad g(M) = P_{min} + W . S_{min}$$

where W is a weighting factor, P_{min} is the smallest eigenvalue of the matrix B defined by (14), and S_{min} is defined by (23). Simple and powerful as it is, this technique has several weaknesses which are analyzed now. The estimation procedures are not iterative in the following sense : if we add a new pair (p_{N+1}, P'_{N+1}) to a given matching M, we can't use the transformation T(M) to compute $T(M,(p_{N+1}, P'_{N+1}))$ because although the matrix B of (14) can be easily updated, its eigenvalues must be recomputed. This increases the time of computation in the tree-search algorithm. The second problem is that the transformation T computed by the two previous algorithms is not necessarily unique because the determination of the rotation (resp. translation) requires at least two pairs of non parallel planes (resp. three pairs of independent planes). These constraints can give rise to indeterminations in the tree search algorithm. This problem is that we use "infinite" planes as primitives.

IV THE MATCHING ALGORITHM

We present now the recognition algorithm which is a simple tree-search procedure [10]. For the sake of clarity, we denote the two descriptions to be matched by two sets of labels G=(1..N) and G'=(1..N') where each label corresponds to a plane as described above. So, a matching M is a list of pairs (i,j). Moreover, we assume that no label of G or G' appears more than once in M and we denote by (M,(i,j)) the new matching obtained by adding a new pair (i,j) to M. At last, we assume that G and G' are the descriptions of the observed scene and of the model of an object, respectively. The output of the algorithm is a matching M which contains the primitives of the model identified in the scene, the corresponding optimal transformation which gives the position of the object in the scene and the quality measure g(M).

The basic algorithm can be described by a recursive procedure MATCH(M,i) whose arguments are the current matching M and the current label i of G. The procedure updates a matching M_{min} which is the best one according to the criterion g (eq. 24).

```
Procedure MATCH(M,i)
    . Try to find a label j which
      has not been already tested such
      that the error g(M,(i,j)) is
      acceptable.
      If none exists then Return
      Else
          If i=N then
              If (M,(N,j)) is better
              than M_min then

              M_min:=(M,(N,j))

          Else MATCH((M,(i,j)),i+1)
```

The standard algorithm must be modified in order to take into account the problems inherent to the estimation of g(M). At first, the evaluation of g(M) is not very fast because it requires some matrix computations such as eigenvalues calculations, moreover if a pair (i,j) is strongly inconsistent with a matching M, it is not necessary to compute a precise error. So, we use a new parameter c(M,i,j) which is called the "local consistency". This parameter is used as a rough estimate of the geometrical consistency of the new pair (i,j) with the matching M and it enables us to perform a fast elimination of strongly inconsistent pairs.

In our case, the primitives are planes and the local consistency can be defined by :

$$(25) \quad c(M,i_0,j_0) = |v_i . v_{i_0} - v'_j . v'_{j_0}|$$

where (i_0,j_0) is the pair to be tested, (i,j) is some pair in M and the vectors v are the normals to the corresponding planes. This relation comes from the fact that there exists a rotation consistent with the pairing $((i,j),(i_j,j_j))$ if and only if $v_i.v_{i_j}=v'_j.v'_{j_j}$. Another possibility is to define c by :

$$(26) \quad c(M,i,j)=\left|1-v_i.Rv'_j\right| \quad ,$$

where R is the rotation defined by the matching M. Although they are very simple, these two functions provide a good measure of inconsistency and speed up the algorithm drastically.

Another problem is the possible indetermination in the computation of the transformation T defined by (1). As previously mentioned, we need two pairs of non-parallel planes in order to estimate the rotation and three for the translation. It means that we have to explore the first three levels of the tree before checking the geometrical consistency. This is the main difference with the 2-D case in which a transformation can be precisely estimated with only one pair of primitives [11]. Thus we use only the "rotational consistency" F_{min} defined by relations (5) and (14) for the control of the search, the complete consistency g(M) being used for selecting the best solutions among several ones at the bottom of the tree.

The last problem is that the estimation of the rotation is not "recursive" which means that we cannot use the estimated rotation R of M in order to estimate the rotation R' of (M,(i,j)). This is also an important difference with the 2-D problem as presented in [11]. Nevertheless, we can perform some calculations in an iterative way. More precisely, if B is the matrix defined by relation (14) for the matching M, the new matrix B' corresponding to the matching (M,(i,j)) is : B'=B+C(i,j), where C(i,j) is the matrix of relation (19) for the pair (v_i,v'_j). Conversely, if we remove a pair (i,j) from the matching M, the new matrix is : B'=B-C(i,j), so we can easily update the least-squares matrix B at each level of the search. Moreover, we store the matrices C(i,j) that have been already computed so that each matrix C is computed only once. At last, if $M=((1,j_1),\ldots,(L,j_L))$ is the current matching at some level of the search, we store the rotation matrices $(R_p)_{p=2..L-1}$ and the errors $(F_p)_{p=2,\ldots,L-1}$ corresponding to its ancestors in the search,

$M_2=((1,j_1),(2,j_2)),\ldots,$

$M_{L-1}=((1,j_1),\ldots,(L-1,j_{L-1}))$.

This last improvement allows us not to recompute the parameters of the ancestor matchings when backtracking occurs. In summary, matrix computations occur when we add a new pair to a partial matching, in this case we have to compute the new estimate of the rotation and the corresponding error, the matrix C defined by relation (19) being computed only if the new pair has never been encountered before. These improvements give rise to a very fast algorithm in spite of the complexity of the numerical algorithms of Part. III.

V THE RESULTS

Results are now presented on a simple scene built with the object of fig. 1. Fig. 5 shows the segmentation of the scene in planar regions, only regions of high quality are considered. Figs.6 and 7 show the result of the identification of the first part. The estimated orientation of the object is shown on fig.6 where the black regions are the identified regions of the model. Fig.7 shows the superposition of the identified regions of the scene (solid lines) with the corresponding regions of the model (dotted lines), the superposition uses the estimation of the transformation. Finally, the regions 12,2 and 1 have not been recognized but are eliminated by a simple superposition test.

The computation time is about 15 seconds that may appear very fast,but it illustrates the efficiency of the geometrical compatibility tests. Finally, the precision of the transformation estimation is rather good, the mean angle between v and Rv' is about 0.04 radians and the translation error is about 3 mm. These values are related to the accuracy of the acquisition which is about 1 mm.

VI EXTENSION TO QUADRIC SURFACES

As previously mentioned, the recognition method is quite general and can be extended to other classes of primitive surfaces provided that there exists an algorithm for the estimation of the transformation . In this Section, we briefly show how the previous algorithm could be extended to the case of quadric surfaces.

The general equation of a quadric surface is :

$$(27) \quad X^tQX+C.X+D=0$$

A transformation T=t*R maps a quadric (Q,C,D) onto a quadric (Q_1,C_1,D_1) such that :

(28) $Q_1=RQR^t$, $C_1=C+2R^tQt$, $D_1=D+t^tQt$.

So, if $M=(P(Q_i,C_i,D_i),P'(Q'_i,C'_i,D'_i))_{i=1,..,N}$ is a matching with N pairs of quadrics, we have to minimize a sum similar to (3) :

$$(31)\ F=\sum_i \|Q_i-RQ'_iR^t\|^2$$

$$+W_1.\|C_i-C'_i-2R^tQ'_it\|^2+W_2.|D_i-D'_i-t^tQ'_it|^2 ,$$

where the Wi's are weighting factors and . Is a matrix norm defined by :

$$(30)\ \|Q\|^2=Tr(Q^tQ)$$

where Tr(A) denotes the trace of the matrix A.

The translation can be easily estimated by a linear least-squares method. The most difficult problem is the estimation of R. The expression can be simplified by using the definition (30) and the relations $Q=Q^t$ and $RR^t=Id$. So, the new expression to be minimized is :

$$(33)\ F'=\sum_i Tr(Q_iRQ'_iR^t) .$$

The theory of the Lagrange multipliers can be applied to this problem. Precisely, a quaternion with unit norm is a local extremum of F' only if there exists a real k such that :

$$(34)\ H_i(q,k)=\frac{\partial F'}{\partial q_i}-kq_i=0 \text{ for } i=1,..,4 .$$

The partial derivative is a polynomial function of degree three of the q_i's. We assume that we know a rotation q^o, then, we seek q'_1 and q'_2 which maximize the criterion (33), the other variables being fixed. This maximization is an elimination between the polynomials Hi's and the constraint. Notice that we compute the absolute maximum of $F'(q_1,q_2,q_3{}^o,q_4{}^o)$ with respect to the variables q_1 and q_2. So, we obtain a new solution $(q'_1,q'_2,q_3{}^o,q_4{}^o)$ which is better than q^o.

This method can be iteratively applied to the other variables. At each step, the new solution q is better than the previous one according to the criterion (33). So, the method converges to a local maximum of the function F'. The only assumption that we have made is the existence of an a-priori estimate q^o.

This estimate could *be* obtained by computing the transformation using only planner regions, the matching of the quadlies beinq used to validate or reject the matching of planes.

VII CONCLUSION

We have presented an algorithm of 3-D recoqnition using primitive surfaces which gives a precise estimation of the positions of the objects. We are in the process of extending the algorithm to more complex primitive surfaces such as quidric surfaces. Our future work is to include this recognition module in a complete vision system including automatic data acquisition, segmentation, model construction and scene analysis.

[l] D.L. Milgram, CM. Bjorklund, "Range Image Processing: Planar Surface Extraction", Proc. of the fifth ICPR, Dp. 912-919, Miami, Dec l-'4, 1939.

[2] R.O. Duda, D. Nitzan, P. Barret, ",L)se of Range and Reflectance Data to find Planar Surface Regions", IEEE Trans. On Pattern Analysis and Machine Intelligence, Vol. PAMI-l, No 3, pp. 259-271, Jul) 1979.

[3] M. Oshima,Y. Shirai, "A Scene Description Using Three-Dimensionnal Information", Pattern Recognition, Vol. 11, Dp. 9-17, 1979.

[4] C.A. Dane, "An Object-Centered Three-Dimensionnal Model Builder", PhD Dissertation, University of Pennsylvania, 1932.

[5] O.D. Paugeras and M. Hebert, "Segmenting range data into planar and quadratic patches", CVPR, Washington, June 83.

[6] M. Hebert , J. Ponce , "A New Method for Segmenting 3-D Scenes", Proc. Of the 3ixth ICPR, Dp. 836-338, Munich, October 1982.

[7] O.D. Faugeras et a1 ., "Towards a Flexible Vision System" in Robot_vision A. Pugh Editor, Springer-Verlag , 1933.

|8] N. Jacobson, "Lectures in Abstract Algebra", Van Nostrand Ed,1951

[9] G.W. Steward, " introduction to the matrix Computations" » Academic Press, 1973.

[IO]L. Shapiro, R. Haralick, "Structural Descriptions and Inexact Matching", PAMI, Vol. 3, No 3, Sept. 81, pp. 504-519.

[11] N. Ayache , "A Model Based Vision System to Identify and Locate Partially Visible Parts", CVPR, Washington, June 83.

Fig.1. Photograph of an automobile part.

Fig.2. Segmentation using the Hough Transform.

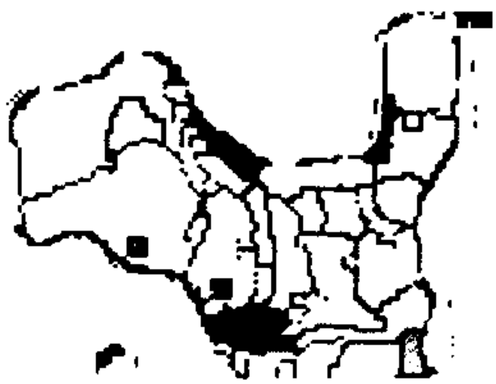


Fig.3. Segmentation using the region growing method.

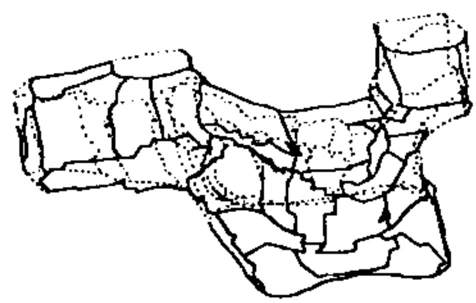


Fig.4. Reference model of the object of Fig.1.

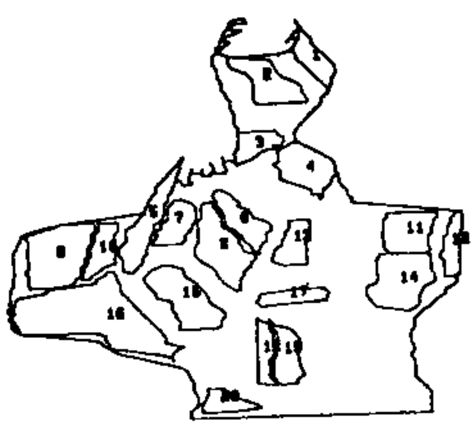


Fig.5. Segmentation of a scene with two objects.

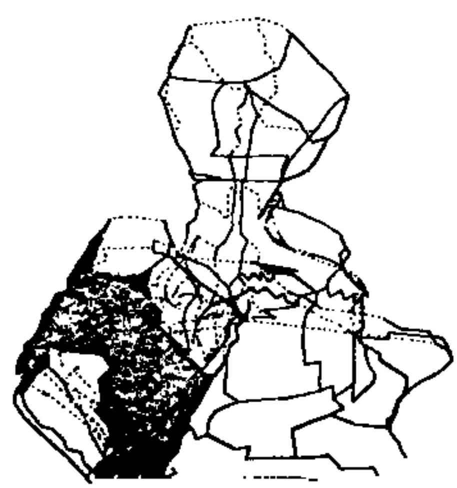
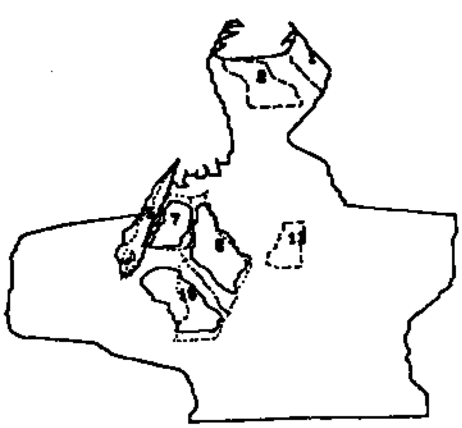


Fig.6. Estimated Orientation of the first object.



——— : recognized regions of the scene.
..... : corresponding regions of the model.
----- : regions eliminated after the recognition.

Fig.7. Superposition of the scene and the recognized model.