

RECOGNITION OF 3-D OBJECTS
USING THE EXTENDED GAUSSIAN IMAGE

Katsushi Ikeuchi

Computer Vision Section
Electrotechnical Laboratory
Umezono-1-1-4, Sakura-mura
Niihari-gun, Ibaraki, 305
Japan

ABSTRACT

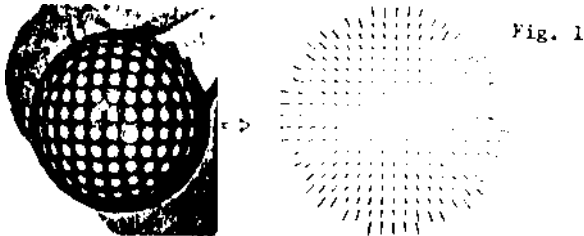
We propose to use an extended Gaussian image (EGI) for interpreting 2-1/2-D representations for recognition of 3-D objects. The EGI is constructed by mapping each surface normals of an object to the Gaussian sphere.

The freedom in viewer directions caused by incomplete observation is greatly reduced by applying constraints derived from a global distribution of surface normals on the EGI. One constraint on the viewer direction is derived from the ratio of the projected area to the original surface area. The other constraint comes from the direction of the principal axis. After reducing the possible viewing directions with these constraints, we will apply a matching function to ESIs of a candidate set for a final decision.

We also propose an algorithm for reconstruction of the original shape of a convex polyhedron from its EGI. This algorithm is based on the analysis-by-synthesis method.

¹ WHAT IS THE EXTENDED GAUSSIAN IMAGE

A collection of local surface normals [1,2,3,4,5], sometimes referred to as a 2-1/2-D representation of an object [6], is often provided by machine vision at the low level. For example, an algorithm based on the propagation-of-constraints technique [2] provides local surface orientation from shading and occluding information. The same algorithm can also produce surface orientation from apparent distortion of known patterns based on a regular-pattern gradient map [A]. The distortion of these small circles on the golf ball in Fig. 1 can be used to recover local surface orientation.



In the above cases, the next problem encountered is how to interpret these local representations for a global recognition of an object. One of the most important issues in this process is how to convert a local representation based on the viewer-centered coordinate system into

a global description based on the object-centered coordinate system such as the generalized cylinder [7,8] or the spike model [1]. Each local representation depicted as a needle in the above example is obtained at a particular point expressed in viewer-centered coordinate system on the image plane. On the other hand, an object model is expressed in a particular coordinate system usually based on an object center and natural axis of the object. These two coordinate systems are independent each other. It is quite difficult to recover the original object-centered coordinate system from the 2-1/2-D representation observed in the viewer-centered coordinate system. We propose to use the extended Gaussian image (EGI) as a tool for this conversion process.

An EGI of an object may be derived from a spike model of the object [1]. A spike model is a collection of surface normals on each surface patch in the 3-D world. Let us assume that there is a fixed number of surface patches per unit surface area and that a unit normal is erected on each patch. The collection of these normals is called a spike model of the object. These normals are like porcupines's quills [1]. These normals on surface patches in a spike model are moved to a common point of application. The locus of points consisting of their end points lies on the surface of a unit sphere. This mapping is called the Gauss map; the unit sphere is called the Gaussian sphere. If we attach a unit mass to each end point of the normal vector, we will observe a distribution of mass on the Gaussian sphere. This distribution of mass will be normalized. The resulting distribution of mass on the Gaussian sphere is called the extended Gaussian image (EGI) of the object [1,9].

An EGI is independent on both the position of the origin and the scale of axes of the coordinate system. A coordinate system may be characterized using three components; the position of origin, the direction of coordinate axes, and a scale factor of a coordinate axis. Among these three components, an extended Gaussian image is independent on both the position of the origin and the scale factor of coordinate axis. An EGI is independent on the position of the origin, because Surface normals on patches will be projected on the Gaussian sphere in parallel transformation without regarding the position of the origin. Also the representation is independent on the scale of the coordinate. For example, a 2x2x2-inches cube has the same representation as a 1x1x1 inches cube, provided that total mass of distribution on the sphere is

normalized. Namely, both images from these cubes consist of impulses of 1/6 unit mass at the north pole, the south pole, and each quarter point on the equator of the Gaussian sphere.

The EGI may be regarded as a unique description, on treating only convex objects. An object has a unique EGI, when the object is a convex polyhedron [10]. Minkowski showed that: two convex polyhedra are identical if each pair of corresponding faces has an equal area and the same surface normal.

IT RECOGNITION OF AN OBJECT USING AN EGI

A. Data Structure

We have to tessellate the Gaussian sphere into meshes, when we treat objects which have continuous surfaces (eg. cylinder, sphere,...). A continuous surface has a continuous distribution of infinitely many points on the Gaussian sphere. We have to convert a continuous distribution into a discrete distribution.

For tessellation, we have to divide the surface of the Gaussian sphere in a uniform manner. Since we look at an object from different directions, a tessellation method is required to have the same resolution of angle in every direction. In other words, each cell on the tessellated sphere is required to have the same area and the same distance from its adjacent cells.

One way to satisfy this condition is to use a regular polyhedron to divide the Gaussian sphere into meshes [11]. A regular polyhedron has faces of equal area. Also a regular polyhedron has faces evenly distributed in direction with respect to the center of gravity. Thus, we can divide a sphere into meshes of a fixed area by projecting edges of a regular polyhedron inscribed in the sphere onto the surface of the sphere with respect to the center of the sphere. The resulting tessellation on the sphere is called as a geodesic dome [12].

Since we have no polyhedron of higher order than the icosahedron, we use a semi-regular geodesic dome constructed from a two frequency dodecahedron. (See Fig. 2.) Our tessellated sphere also has a hierarchical structure. Each cell on one level contains both pointers to sub-cells on the next level and the direction of the center point of the cell. Distance measures between cells is also maintained in this structure. Since we use a two frequency dodecahedron for tessellation, the top level cell contains pointers to twelve surfaces of a geodesic dome based on a dodecahedron. The second level has 60 triangle areas which come from a one frequency dodecahedron. The lowest level contains 240 triangle cells of a two frequency dodecahedron.

Quantization of an object in all directions uniformly is also required when making a prototype. For example, if you sample surface orientations of an ellipsoid along its long axis at every observed picture element, you will obtain very rough information on the view from the direction of the long axis even though you get fine information on

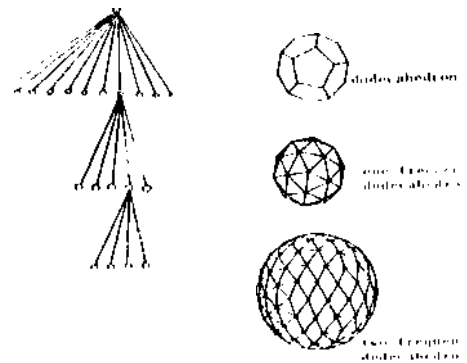


Fig. 2 Data structure. Each cell, depicted as a circle, contains both pointer to sub-cells on the next level and the surface orientation there.

the view from the direction perpendicular to the long axis. Thus, this sampling method is quite unsuitable in this case. We will use the tessellated sphere for sampling directions when making prototypes as well as for registration of the EGI. The tessellated sphere samples the whole spatial directions in an even manner. This schema will give an even sampling with respect to every direction.

Note that two spheres are used; One for sampling spatial directions and the other for registering an EGI on its spherical surface. The former sphere for sampling is assumed to be located at the center of gravity of an object. Surface orientation of patches on an object are sampled at every direction corresponding to each cell on the tessellated sphere. In other words, the surface normal on each cell of the sampling sphere is extended until it reaches the surface of the object. At that surface patch where the extended surface normal hits, surface orientation of the object is measured. The measured surface area will be registered on the corresponding cell of tessellated Gaussian sphere for expressing the EGI of the object. Thus, the cumulative image on the sphere for registration is constructed as the EGI for the prototype of the object.

B. Constraints on Viewer Directions

Matching an EGI from observed data with its prototype involved three degrees of freedom, since there are three degrees of freedom to altitude of an object in space. If the whole surface of an object was visible, the directions of the three principal moments of inertia would determine the spatial alignment of the object [9]. Unfortunately, however, the typical situations allow us only a partial observation of an unknown object. This partial observation causes three degrees of freedom in altitude of the object. Viewer direction gives two degrees of freedom; that is, there are two degrees of freedom corresponding to which cell on the Gaussian sphere of a prototype is perpendicular to the direction of the line of sight. The remaining degree of freedom comes from rotation of

the Gaussian sphere of the prototype around the line of sight when matching the prototype with the observed sphere.

We will use two kinds of constraints to reduce the search space for matching. Although a brute force technique, such as search through the space of possible attitudes, can be applicable to matching directly, we will reduce this search space using constraints before beginning comparison. One constraint reduces the freedom of viewing direction. This constraint is derived from a ratio of visible area projected onto a plane perpendicular to the line of sight against surface area. The other one constrains freedom of rotation. This constraint for rotation of the sphere is based on the symmetry axis on the plane perpendicular to the line of sight.

The ratio of area projected onto the image plane against the original surface area is a constraint on the viewing direction. For example, observing an ellipsoid from the direction of the long axis gives a smaller projected area than looking at the same ellipsoid from the direction perpendicular to the axis. Yet the surface area is the same.

Projected area is normalized by surface area. The area is calculated at each triangle v before comparison. See Fig. 3. This ratio is denoted as

$$A_v = \text{sum} (Z_i \times M_i) / \text{sum} (M_i),$$

where the summation is done only if the mesh i is visible from the mesh v , M_i is mass at the cell i , and Z_i is a inner product between the viewer direction v and the surface normal at the cell i . This ratio is calculated only over a visible hemisphere from the viewing direction v . This ratio removes only one degree of freedom Just like the reflectance ratio on the Gaussian sphere [2], Thus, even though this constraint reduces the possible set of of viewer directions, we still have to search for the real view point on the Gaussian sphere along one of these iso-ratio lines observed from an image. For example, Fig. 4 denotes iso-ratio contours drawn on the stereographic plane for graphical clearness. This diagram is used when finding viewer direction in the same way as the reflectance map is used to recover the 2-1/2-D representation from a brightness value.

We also must find the rotation of the EGI of a prototype around each candidate cell for the viewer direction. We use the direction of the principal axis of inertia over a hemisphere from the viewer direction. We will use the 2-D axis rather than the 3-D axis for simplicity of calculation. In other words, we use the direction of the axis of least inertia on the tangential plane at cell v (or the image plane on seeing the object from the direction of v).

The following calculation will give the direction of this axis. Before calculation, each cell i is projected onto the tangential plane at the cell v . Let (X_i, Y_i) be the coordinate obtained on the plane. We will use

$$\begin{aligned} I_{xx} &= \text{Sum}(M_i \times X_i \times X_i), \\ I_{xy} &= \text{Sum}(M_i \times X_i \times Y_i), \\ I_{yy} &= \text{Sum}(M_i \times Y_i \times Y_i), \end{aligned}$$

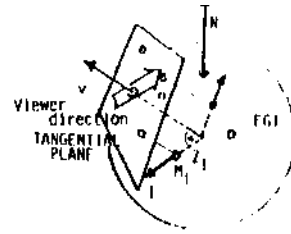


Fig. 3 The relationship between the EGI and the viewer direction.

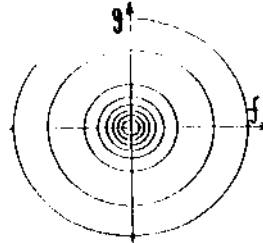


Fig. 4 The ratio of the surface area to the principal axis of inertia projected area. These ratios are depicted as contours on the stereographic plane. The prototype and the observed directions.

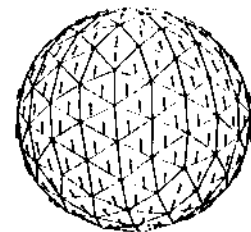


Fig. 5 The direction of surface area to the principal axis of inertia at each image plane on seeing the object from the direction. The stereographic plane. The prototype and the observed directions.

where these summation are also done only if cell i is visible from the cell. From these values of I_{xx}, I_{xy}, I_{yy} , the principal direction of inertia is obtained.

$$\alpha = (\text{atan}(2I_{xy}/(I_{yy}-I_{xx}))) / 2,$$

This α is the direction of the principal axis on the tangential plane.

This direction may be regarded as a symmetry axis of the object observed from the direction denoted with the cell v . Note that, if the object has a symmetry axis in the 3-D space, the principal direction of inertia in 2-D obtained above will correspond to the direction of 3-D symmetry axis projected onto the plane [9]. In other words, the direction is unique at each cell in almost all case. Thus, this direction uniquely determines the manner in which way a prototype should be aligned on comparing the prototype with an observed data. In this case, the constraint reduces the possible area perfectly; there is no need for search involved. In Fig. 5 the direction of principal axis of inertia at each viewing direction is depicted as a needle on each cell v . These needles constraint the rotation of prototype around the line of sight.

So far each cell on the Gaussian sphere has to contain three kinds of information; how much weight the cell contains (mass), how much an object has projected area on being viewed from the direction of the cell (surface ratio), which way the axis of inertia exists (axis). The second one (surface ratio) and the third one (axis) constrain freedom of viewing direction before matching. The first one

(mass) is used for final decision based on a matching function mentioned in the next section.

C. Matching Function

The following matching function is used for comparing an EGI obtained from a 2-1/2-D representation with images of prototypes.

```

if M-proto(cell-number) equal 0.0 do nothing
if M-proto(cell-number) not equal 0.0 then
  setq difference (- M-proto(cell-number)
                  M-observed(cell-number+e))
  distance (distance cell-number
            cell-number+e)
if (diff difference-standard
    and
    dist distance-standard)
  then
    add exp ((- diff x g1)+(- dist x g2))
    to total-point
  
```

where g_1 and g_2 are coefficients. In other words, if a cell in the prototype has some contents, the algorithm checks around the cells in the observed data to see whether the observed data contains similar amounts of mass or not. In case there exists a similar amount of mass within a neighboring area, a weight will be added to the point-total for the prototype. After calculating this point-total for all candidates, the candidate obtaining highest point-total will be determined as the object observed.

D. An Illustrative Example

Consider the simple example of an ellipsoid ($X^2+Y^2+Z^3/5-1$) observed from the direction inclined from Z axis by angle 10 degrees. The surface normal will be derived analytically and used to generate a needle diagram. The algorithm will be applied to the synthesized needle diagram. The result will be used as a basic basis for judging the performance of the algorithm.

For the example, a 40 x 40 synthesized needle-diagram was generated. At each image point, elementary calculus gave an outer surface normal of the ellipsoid. For graphical clearness, each surface normal is depicted as a needle.

The first task is to obtain an EGI from this needle diagram. Each surface normal at an image point is determined to which cell on the gaussian sphere the surface normal corresponds. Then, its surface area is added to the cell. Since each image point is assumed to have the same area, corresponding surface area is calculated as

$$(1 / \cos \alpha).$$

where α is the angle between the surface normal and the direction of the viewer. After obtaining surface areas, cumulative values registered on cells of the Gaussian sphere are normalized by the surface area observed. Fig. 7 shows the resulting EGI obtained from Fig. 6.

We also determine the ratio of the projected area to the surface area, while making the EGI from

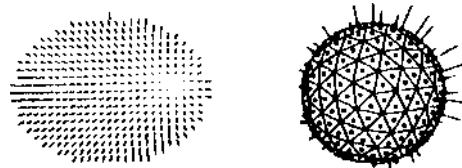


Fig. 6 A synthesized needle diagram. Fig. 7 The obtained EGI from Fig. 6

the needle diagram. This ratio reduces possible viewer directions as mentioned above. The surface area is calculated simply by summing up $(1 / \cos \alpha)$, while the projected area is as same as the number of image points within the occluding boundary. *Speaking, we use the points within a boundary ($\sqrt{f^2+g^2} < 0.9$) when obtaining the surface area, because points near the occluding boundary cause a dramatic effect to this value. For example, if a point sits at the occluding boundary, $(1 / \cos \alpha)$ becomes infinite; we cannot obtain the real surface area.*

Fortunately, the constraint, the ratio of the projected area to the surface area, also works within some boundary. More precisely, the ratio is a function of both a spatial angle on the gaussian sphere and the viewer direction. For example, if we take into account all points on a hemisphere of the Gaussian sphere, the ratio is the constraint to the viewer direction as the one mentioned above. On the other hand, if we use only points within a cell, the constraint is identical to the judgment whether the cell contains some amount of mass or not. Between these two extremes, we can also derive various kinds of constraints, depending upon which area on the gaussian sphere we use for calculation. In this example, we use the ratio of the projected area to the surface area whose corresponding points on the Gaussian sphere have azimuth angles less than 85 degrees based on the direction of the viewer ($\sqrt{f^2+g^2} < 0.9$). We obtained 0.27 as the ratio within the boundary in this illustrative example.

We prepared two kinds of EGIs as candidates for matching. One is from the ellipsoid and the other one is from an cylinder whose radius is 1.0 and height is 1.5. Their EGI is shown in Fig.8.

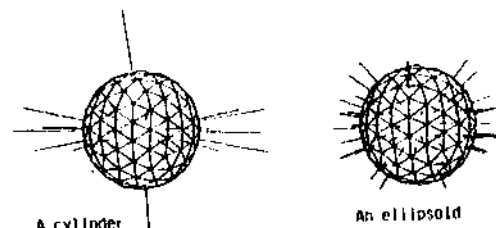


Fig. 8 EGIs of candidates.

The ratio of the projected area to the surface area (0.27+10.1) reduces 240 possible viewer directions to 21 directions in the case of the ellipsoid and to 43 directions in the case of the cylinder. Since both objects have symmetry axes, the constraints from the axis determines the direction of alignment uniquely.

In the ellipsoid case, cell 16 (and cell 220 - the opposite side of 16) has the highest value, 6.59

and cell 0, 8 has the Second largest value 6.07. In the cylinder case, cell 0 has the largest value 1.40. Namely, the algorithm tells us that the cell number 16 of the ellipsoid is most likely as the viewer direction. This cell has the direction (0.1426, -0.1036, 0.9843) based on the direction of the long axis of the ellipsoid. In other words, the algorithm successfully tells us that it is most likely that the ellipsoid is observed from the direction inclined by 10.15 degrees from the long axis of the ellipsoid,.

III RECONSTRUCTION OF A CONVEX POLYHEDRON FROM ITS EGI

If an object is a convex polyhedron, it is possible to reconstruct the original shape from its EGI. This result can be predicted from Minkowski's theorem [10]. This theorem is quite important. Because of it, we need not store the adjacency relationships among faces. Only the surface orientation of each face and its area need be remembered. Actually, this fact is the basic motivation for using an extended Gaussian image.

Unfortunately, Minkowski's proof of the theorem is not constructive and does not provide an algorithm for reconstruction. We propose here an algorithm for this reconstructing of a convex polyhedron from its EGI. Our algorithm for reconstruction is based on the analysis-by-synthesis method.

The problem for reconstructing a shape from its EGI is to obtain the distances of the faces (C_0, \dots, C_n) from an origin, provided that we have known areas (M_0, \dots, M_n) of faces and the surface orientations $((X_0, Y_0, Z_0), \dots, (X_n, Y_n, Z_n))$. We may set $C_0=0$ without loss of generality, because a polyhedron can be put at an arbitrary place in a particular coordinate system without altering its shape.

We can divide the problem into n cases. There always exists a maximum value C_{\max} among G_1, \dots, G_n . That is, there is a farthest face from the origin among face 1, ... face n . Thus, we will consider n sub problems. Namely, in case i , we assume C_i has the maximum value; face i is the farthest face from the origin. We set the farthest distance, $C_i - 1.0$, in case i . The reason for dividing the problem into n cases is to fix the possible values of the C_i between 0.0 and 1.0. Otherwise, C_i may take a value from 0.0 through an infinite value. Since we do not know which case is true among the n cases, we will check all n cases to see whether the case contains a solution point or not. For each case, we will quantize the remaining $n-2$ dimensional space into cells where $C_1, \dots, C_{i-1}, C_{i+1}, \dots, C_n$ changes from 0 through 1.0. Then, it is possible to calculate M_0, \dots, M_n at each mesh point (C_0, \dots, C_n) and to compare with the given values.

We will show how to calculate M_0 as an illustration. Obviously, another face area M_i can be obtained in the same way.

For simplicity, we assume (X_0, Y_0, Z_0) $(0.0, 0.0, 1.0)$, where (X_0, Y_0, Z_0) denotes the unit

normal vector corresponding to the surface orientation of face 0. This can be established by rotating the Gaussian sphere so that the vector (X_0, Y_0, Z_0) points to the north pole. In other words, $(T(X_0, Y_0, Z_0)) - (0.0, 0.0, 1.0)$, where T is a matrix which corresponds to one of the suitable rotations.

The other points on the Gaussian sphere should also be rotated in the same way; i.e. $(T(X_i, Y_i, Z_i))$. In the following part we denote $T(X_i, Y_i, Z_i)$ by (X_i, Y_i, Z_i) for the sake of simplicity.

We next determine whether one intersection line constitutes an edge of the face for (X_0, Y_0, Z_0) or not. Every face corresponding to (X_i, Y_i, Z_i) has an intersection line with the infinite plane corresponding to (X_0, Y_0, Z_0) . The desired face corresponding to (X_0, Y_0, Z_0) consists of a partial plane cut out from the infinite plane by some segments of these intersection lines.

Since a face of a convex polyhedron is also convex, the algorithm can connect only those segments which constitute a convex figure. If a small bug is assumed to travel along edges of a convex figure counter-clockwise, the bug would always turn to the left at each corner of the convex figure. In other words, the normal vector from each edge (and also the projection of the surface normal of a face which intersects the infinite plane constituting the edge) also rotates clockwise. For this, we will order the intersection line according to the azimuth angle $(\text{atan}(Y_i/X_i))$ of the normal vector (X_i, Y_i, Z_i) of each face i . (we will denote the n -th intersection line as L_n), then the bug will visit intersection lines in the order sometimes skipping some lines.

We calculate the intersection point between the n -th line and the $n+1$ -th line (we will denote this point $P_{n,n+1}$). These points are candidates for vertices on the face of the polyhedron. We will connect $P_{n-1,n}$ with $P_{n,n+1}$. In other words, we will cut out a segment S_n between $P_{n-1,n}$ and $P_{n,n+1}$ from the intersection line L_n , where " S_n " has the direction from $P_{n-1,n}$ to $P_{n,n+1}$.

---> We will connect S_n from $P_{0,m}$ in sequence. If S_{n+1} turns around $P_{n,n+1}$ counter-clockwise, based on the direction of the segment " S_n ", then the segment " S_n " is an edge of the face. If S_{n+1} turns around $P_{n,n+1}$ clockwise, then S_{n+1} is not an edge of the face. We call this segment S_{n+1} as an imaginary edge and do the imaginary-edge operation in order to find a real edge.

The imaginary-edge operation recalculates a segment which causes an imaginary-edge. Let us assume that S_i is detected as an imaginary-edge. It means that the last segment S_{i-1} should be also re-calculated, because the last segment S_{i-1} is assumed to intersect with S_i but S_i is not a real edge. The imaginary-edge operation will recalculate each intersection point between L_{i-1} and L_j , where j is larger than i and make a new segment. The imaginary-edge operation will find a segment S_{i-1} whose length is smaller than S_{i-1} for the first time after L_i along the sequence. (See Fig. 9.) This segment S_{i-1} between $P_{i-2,i-1}$ and $P_{i-1,j}$ is

returned to the main operation as an edge of the face. This imaginary-edge operation is called recursively, if necessary.

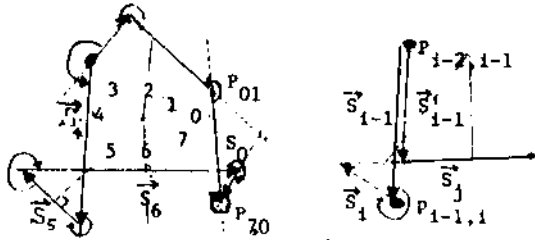


Fig. 9

(A) At first, all adjacent segments will be connected on their azimuth angles. (B) An imaginary-edge operation re-calculates a segment which causes an imaginary edge.

After finding all real segments, the face area will be obtained as

$$M_0 = \sum_i \alpha_i (C_i^2 + 2C_i Z_i + (C_i Z_i)^2) + \beta_i (C_{i-1} C_i + C_0 (C_i Z_{i-1} + C_{i-1} Z_i) + C_0^2 Z_{i-1} Z_i) + \gamma_i (C_{i-1} C_i + C_0 (C_i Z_{i-1} + C_{i-1} Z_i) + C_0^2 Z_{i-1} Z_i),$$

where

$$\alpha_i = \frac{y_{i-1} z_i - y_i z_{i-1}}{\Delta_i}, \quad \beta_i = -1/\Delta_{i-1}, \quad \gamma_i = -1/\Delta_i$$

$$\Delta_i = x_i y_{i-1} - x_{i-1} y_i$$

0-th and i-th plane are denoted as

$$-Z = C_0$$

$$x_i x + y_i y + z_i z = C_i$$

In the above calculations, i visits only real edges in sequence. After obtaining all areas M_0, \dots, M_n , the areas will be normalized. Comparison with the goal will be done as

$$S = \sum_k (M_k - M_{k-proto})^2$$

where $M_k, M_{k-proto}$ are areas corresponding to the face (X_k, Y_k, Z_k) .

IV CONCLUDING REMARKS

We proposed the use of the EGI for interpreting 2-1/2-D representations obtained from raw data for recognition of 3-D objects. While a 2-1/2-D representation is based on the viewer-centered coordinate system, a 3-D description is expressed in an object-centered coordinate system. This discrepancy between the two kinds of representations may be overcome by using the EGI, since the extended Gaussian image is essentially independent of the coordinate system.

Incomplete observations cause freedom in viewer directions and rotations in the space on matching. The freedom is greatly reduced if we apply constraints derived from a global distribution of mass to the matching process. So far each cell contains the mass of an EGI at that point, surface

ratio representing how much a surface is projected on a plane perpendicular to a viewing direction, and the direction of a symmetry axis relative to the viewing direction. After reducing the possible directions of observation, a matching function is applied to the observed distribution of mass with the prototypes of candidate objects. The best fitting candidate will be selected as the object observed at that time.

Even though a couple of non-convex objects among a candidate set may have the same EGI [13], an EGI is still a helpful tool; after reducing the possible set of candidates and also determining possible spatial alignments of the candidates by using the EGI, the interpreter can make a final decision based on the other features such as position information, which the EGI discards in order to overcome the discrepancy between the viewer-centered coordinate system and the object-centered coordinate system.

ACKNOWLEDGMENTS

The author would like to extend his sincere appreciation to Dr. Y. Shirai of ETL and Prof. B.K.P. Horn of MIT for their helpful discussions. Thanks go to the referees to their helpful comments.

REFERENCES

- [1] Horn, B.K.P., "SEQUINS and QUILLS representation for surface topography." AI Memo No. 536, AI Lab., MIT, 1979.
- [2] Ikeuchi, K and Horn, B.K.P., "Numerical Shape from Shading and Occluding Boundaries." Artificial Intelligence 16:Special Issue on Vision (1981).
- [3] Ikeuchi, K., "Determining Surface Orientations of Specular Surfaces by Using the Photometric Stereo Method," IEEE Trans. on PAMI (accepted).
- [4] Ikeuchi, K., "Shape from Regular-patterns An Example of Constraint Propagation in Vision." in Proc. IJCP-80, Miami Beach, Dec, 1980, 1032-1039.
- [5] Barrow, H.G. and Tenenbaum, J.M., "Recovering Intrinsic Scene Characteristics from Images." Hanson, A.R. & Riseman, E.M., (Ed.) Computer Vision Systems, Academic Press, New York, 1978.
- [6] Kender, J.R., "Shape from Texture," Ph.D. Thesis, Computer Science Dept., CMU, 1980.
- [7] Marr, D. and Nishihara, K.H., "Representation and Recognition of the Spatial Organization of Three-dimensional Shape." Proc. R. Soc. Lond., B:200, (1978).
- [8] Binford, T.O., "Visual Perception by Computer." IEEE Conf. on Systems and Control, Miami, Dec. 1971.
- [9] Smith, D.A., "Using Enhanced Spherical Images for Object Representation," AI Memo No. 530, AI Lab., MIT, 1979.
- [10] Lysternik, L.A., Convex Figures and Polyhedra, Dover, New York, 1963.
- [11] Horn, B.K.P., A Personal Communication, Dec. 1980.
- [12] Wenninger, M.J., Spherical Models, Cambridge University Press, Cambridge, 1979.
- [13] Bajcsy, R. "Three-dimensional Scene Analysis," in proc IJCP-80, Miami Beach, Dec. 1980,