Robotics and Autonomous Systems 8 (1991) 291–309
North-Holland

Exploiting parallelism in 3-D object recognition using the Connection Machine

Suchendra M. Bhandarkar

Dept. of Computer Science, 415 Graduate Studies Research Center, University of Georgia, Athens, GA 30602, USA

Ravi V. Shankar

Dept. of Computer and Information Sciences, Syracuse University, Syracuse, NY 13244, USA

Minsoo Suk

Dept. of Electrical & Computer Engineering, Syracuse University, Syracuse, NY 13244, USA

Abstract

Bhandarkar, S.M., Shankar, R.V. and Suk, M., Exploiting parallelism in 3-D object recognition using the Connection Machine, Robotics and Autonomous Systems, 8 (1991) 291–309.

This paper addresses the problem of parallelism in 3-D object recognition and localization using a fine-grained SIMD architecture such as the Connection Machine™. The input images are range images of partially occluding three-dimensional objects. The objects are made up of piecewise compositions of curved surfaces. The surfaces of interest are cylindrical, conical and spherical surfaces since they cover a large portion of objects encountered in an industrial environment. Qualitative classification of surfaces based on the signs of the Mean and Gaussian curvature is used to come up with *dihedral feature junctions*. Dihedral feature junctions are shown to be robust to occlusion and offer a viewpoint independent scheme for modeling the object models and are well suited for matching and pose determination. Hough clustering is chosen as the constraint propagation mechanism on account of the ease of parallelization. Issues regarding parallelism on the Connection Machine™ for every stage of the recognition and localization process such as segmentation, feature extraction matching and pose determination are discussed. Experimental results on the Connection Machine™ bring out the advantages of exploiting parallelism for 3-D object recognition and localization on a fine-grained SIMD architecture.

Keywords: Model-based vision; 3-D object recognition; Hough clustering; Range image processing; Range image segmentation; Parallel processing.

1. Introduction

Representation and control strategy are two important issues that 3-D object recognition systems deal with. The issue of representation deals with (a) the choice of features with which to represent image data and the object models, and (b) the choice of appropriate representation of constraints which arise out of matching scene features with model features. The issue of control strategy deals with the choice of appropriate constraint propagation/constraint satisfaction technique such that a consistent interpretation for the scene can be obtained. Most existing 3-D object recognition systems can be classified based on the choices made for representation and control strategy [1].

Recognition-via-localization is a commonly used paradigm is 3-D object recognition. The image features used are primitive and matching takes place early in the recognition and localization process. Geometric constraints arising out of matching of local geometric features are propagated using a constraint propagation/constraint

satisfaction technique. A consistent set of constraints represents a valid scene interpretation. In this sense a scene interpretation problem can be thought of as a constraint propagation/constraint satisfaction problem.

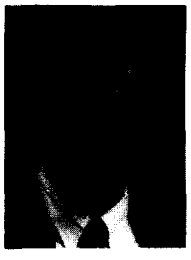
The two commonly used techniques for constraint propagation/constraint satisfaction in the recognition via localization paradigm are: (i) Searching through an Interpretation Tree and (ii) Hough (Pose) Clustering.

(i) The Interpretation Tree (I.T.) approach matches a scene feature with a model feature at every stage in the recognition/localization process. Local geometric constraints such as pairwise angle and distance measurements are used to discard inconsistent matches. Each subset of consistent matches can be represented by a path in the I.T. The local geometric constraints are used to prune paths in the I.T. When a path of sufficient length is found, a global transformation is used to determine and verify the pose of the object. The control structure of the algorithm is that of hypothesize-and-test with backtracking.

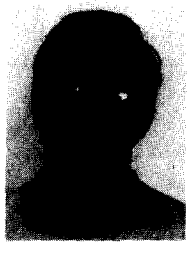
(ii) The Hough (Pose) Clustering approach matches each scene feature to each possible model feature. The match is constrained by local geometric constraints based on angle and distance measurements. Under the assumption of rigidity, each match enables one to compare a geometric transform which can be represented as a point in Hough (parameter) space. For 3-D objects with six degrees of freedom the Hough space is six-dimensional. Clustering of points in the Hough space yields a globally consistent hypothesis regarding the pose of the object.

Both the Interpretation Tree approach and the Pose Clustering approach have been used extensively in 3-D object recognition. Grimson and Lozano-Perez [2] and Faugeras and Herbert [3] have used the Interpretation Tree approach whereas Stockman [4], Boyter and Aggarwal [5] and Dhome and Kasvand [6] have used the Pose Clustering approach in their experiments in 3-D object recognition. Most of the work cited in the literature deals with the recognition and localization of polyhedral objects or curved objects approximated by planar surfaces. Vemuri and Aggarwal's article [15] is one of the few published works which deals directly with the problem of recognition and localization of curved objects.

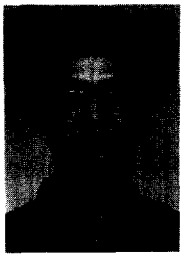
Both the Interpretation Tree approach and the Pose Clustering approach are seen to work fairly well in single-object scenes. Multiple-object scenes with partial occlusion result in a combinatorial explosion in the search space of valid scene interpretations. Conventional implementations of the Interpretation Tree approach and the Pose Clustering approach on a serial architecture exhibit a severe degradation of performance in terms of the time taken to come up with a valid scene interpretation. Exploring the scope for par-



Suchendra M. Bhandarkar received his B. Tech in Electrical Engineering from the Indian Institute of Technology, Bombay, India in 1983, and his M.S. and Ph.D in Computer Engineering from Syracuse University, Syracuse, New York in 1985 and 1989 respectively. He is currently an Assistant Professor in the Department of Computer Science at the University of Georgia, Athens, Georgia. He was nominated University Fellow for the academic years 1986–87 and 1987–88. He is a member of the Institute of Electrical and Electronic Engineers (IEEE), American Association of Artificial Intelligence (AAAI), Association of Computing Machinery (ACM) and Society of Photo-Optical and Instrumental Engineers (SPIE). He is also a member of the honor societies Phi Kappa Phi and Phi Beta Delta. His research interest include computer vision, pattern recognition, image processing, artificial intelligence and parallel algorithms and architectures for computer vision and pattern recognition.



Ravi V. Shankar is a doctoral candidate in Computer Science at Syracuse University. He received his B.E. in Computer Science and Engineering from Anna University, Madras, India in 1987. His research interests are in the areas of computer vision, images processing, and parallel algorithms in vision and robotics.



Minsoo Suk received his B.S., M.S. and Ph.D degrees, all in Electrical Engineering in 1970, 1972 and 1974 respectively, from the University of California at Davis. He is currently an Associate Professor in the Department of Electrical and Computer Engineering at Syracuse University, Syracuse, New York. Prior to joining Syracuse University, he was a member of the Technical Staff at Rockwell International, an Associate Professor of Electrical Engineering at the Korea Advanced Institute of Science and Technology and a visiting Associate Professor at the University of California at Davis and the University of California at Santa Barbara. His current research interest are in the areas of computer vision, neural networks and parallel computing.

allelism in the recognition and localization process is thus very essential. The Pose Clustering approach due to its potential ease of parallelization is therefore a natural candidate for such a study.

A considerable amount of effort in recent times has been devoted to the design and development of architectures for computer vision problems. The architectures vary from fine-grained SIMD architectures such as the mesh [22], the pyramid [23], the tree [24] the Connection Machine™ [25] and the polymorphic-torus [26] to coarse-grained MIMD architectures such as the hypercube [27] and the hierarchical bus architecture [28]. Given the complexity of vision problems and the absense of rigorous benchmarks there is no clear consensus is the research community regarding the relative merits or demerits of the various architectures.

Fine-grained SIMD architectures such as the Connection Machine™ have been found to be well suited for many vision problems. Little et al. [29] have demonstrated the usefulness of the Connection Machine™ for several early vision problems such as edge detection, intermediate vision problems such as connected component labeling, and determination of the convex hull and the Voronoi diagram of a set of planar points and high-level vision problems such as the generalized Hough transform for the recognition and localization of 2-D objects. A set of primitive parallel operations such as general permutations, grid permutations and scan operations on a vector model are defined. These primitives are shown to be the basic building blocks for implementing procedures for low-, intermediate- and high-level vision.

In this paper we consider a more general problem in high-level vision, i.e. that of recognition and localization of 3-D objects from range data. The input images contain multiple objects with objects partially occluding each other. The objects that have been considered are those which could be expressed as piecewise composition of cylindrical, conical and spherical surfaces. This restriction is not unreasonable since most objects encountered in an industrial environment can be modeled thus. The combinatorial complexity of the search space of possible scene interpretations necessitates parallelism. This paper shows how parallelism can be exploited at various stages in the recognition and localization of 3-D objects from range data. These stages are edge detection, segmentation, feature extraction, matching and pose determination. Qualitative classification of surfaces based on the signs of the Mean and Gaussian curvature is used to come up with *dihedral feature junctions* as features for matching and pose determination. Dihedral feature junctions are shown to be fairly robust to occlusion and offer a viewpoint independent modeling technique for the curved objects under consideration. This offers a considerable saving in terms of storing the object models as compared to the viewpoint dependent modeling techniques which need to store multiple views of a single object model. Dihedral feature junctions are quite easy to extract and do not require very elaborate segmentation. Our experimental results on the Connection Machine™ bring out the advantages of exploiting parallelism in 3-D object recognition.

2. Representation of curved surfaces

The desired characteristics for the representation of curved surfaces are: (i) viewpoint invariance so that they can be used for matching, (ii) richness of local support which ensures robustness to occlusion, (iii) ease of extraction so that the effort involved in segmentation is minimal and (iv) generality of representation so that a wide range of surface types can be covered. The representation of surfaces in terms of surface curvature properties exhibit the above characteristics.

2.1. Invariant surface features

It is a well-known result from surface differential geometry [7] that the surface curvature properties such as the Mean curvature (H), Gaussian curvature (K), the maximum principal curvature (κ_1) and the minimum principal curvature (κ_2) values are invariant to rigid body motion (i.e. rotations and translations) in 3-D Euclidean space. Moreover, the unit vectors in the directions of the principal curvatures κ_1 and κ_2 denoted by $\boldsymbol{t}_1$ and $\boldsymbol{t}_2$, respectively and the unit normal $\boldsymbol{n}$ form an orthogonal coordinate system at every non-umbilic and non-planar point on the surface as shown in *Fig. 1*.

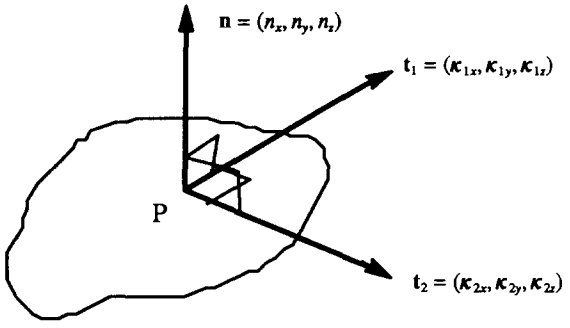


Fig. 1. Curvature properties of a range surface.

The invariance of these surface curvature properties forms the underlying basis of existing surface segmentation and feature extraction techniques. Haralick et al. [9] in their Topographic Primal Sketch come up with a description of the surfaces in terms of the first and second derivatives of the intensity surface. Each image pixel is uniquely classified as one of the six qualitative surface types: peak, ridge, ravine, saddle, flat and hillside. Besl and Jain [8] describe range surfaces based on the Mean and Gaussian Curvature computed at every point in the range image. The signs of the Mean and Gaussian curvature (*HK* sign map) are used to classify each range pixel as one of eight qualitative surface types: peak, pit, ridge, saddle valley, planar, minimal and valley. A region growing algorithm is used to segment the range image surface into homogenous surface patches based on the *HK* sign map. Hoffman and Jain [11] segment the range surface into surface patches which are labeled as convex, concave or planar. The range image is segmented into surface patches by a squared error criterion clustering algorithm using surface points and associated surface normals. Non-parametric statistical tests for trend, curvature values and eigenvalue analysis are used to classify the underlying surface. Boundaries between adjacent surface patches are classified as crease or non crease edges. Fan et al. [12] detect surface discontinuities by determining the location and magnitude of the zero crossing of the Gaussian curvature and the extrema of the principal curvature. This is followed by a surface fitting algorithm which takes into account the surface discontinuities. Vemuri and Aggarwal [13] classify each range image pixel based on the principal curvature values into one of five qualitative surface types: planar, parabolic, elliptic, hyperbolic and umbilic.

2.2. Extraction of surface curvature features form range images

Since the digital range images are typically noisy, direct approximation of the surface curvatures and the surface normals from the raw range data is very unreliable. A more reliable technique for surface curvature extraction is by approximating the digital range surface by an analytic surface in a local window using a least-squares surface fitting technique or a functional optimization technique. The equation of the fitted analytic surface is used to compute the values of the surface curvatures and the surface normals. Typical surface fitting techniques include least-square surface fitting using bivariate polynomial approximations [10] or parametric surface approximation techniques such as B-splines [14], tensor products of splines under tension [13] or tensor product of discrete Chebyshev polynomials [16].

In our experiment the digital range surface is approximated by using discrete bi-orthogonal Chebychev polynomials [16] as basis functions in a local $N \times N$ (where N is odd) window centered about the point of interest. The orthogonality of the basis functions enables efficient computation of the coefficients of the functional approximation. The first four orthogonal polynomials are:

$$\phi_0 = 1,\ \phi_1 = u,\ \phi_2 = u^2 - \frac{\mu_2}{\mu_0},\ \phi_3 = u^3 - \frac{\mu_4}{\mu_2}u,$$

$$\text{where } \mu_k = \sum_{u=-M}^{u=+M} u^k,\ M = \frac{N-1}{2}. \qquad (2.2.1)$$

A discrete bi-orthogonal basis is created from the ϕ_i's

$$\phi_{i,j}(u, v) = \phi_i(u)\phi_j(v). \qquad (2.2.2)$$

The surface function estimate that minimizes the sum of squared surface fitting error within the window is given by:

$$\hat{f}(u, v) = \sum_{i,j=0}^{3} a_{i,j}\phi_i(u)\phi_j(v), \qquad (2.2.3)$$

where the coefficients of the functional approximation are given by

$$a_{i,j} = \sum_{(u,v)=(-M,-M)}^{(+M,+M)} f(u, v)b_i(u)b_j(v). \qquad (2.2.4)$$

The $b_i(u)$ are the normalized versions of the polynomials $\phi_i(u)$ [8].

The estimates of the first and second order derivatives of the surface are given by:

$$f_u = a_{10} - \frac{\mu_2}{\mu_0}a_{12} - \frac{\mu_4}{\mu_2}a_{30} + \frac{\mu_4}{\mu_0}a_{32},$$

$$f_v = a_{01} - \frac{\mu_2}{\mu_0}a_{21} - \frac{\mu_4}{\mu_2}a_{03} + \frac{\mu_4}{\mu_0}a_{23},$$

$$f_{uu} = 2a_{20} - 2\frac{\mu_2}{\mu_0}a_{22},$$

$$f_{vv} = 2a_{02} - 2\frac{\mu_2}{\mu_0}a_{22},$$

$$f_{uv} = a_{11} - \frac{\mu_4}{\mu_2}a_{31} - \frac{\mu_4}{\mu_2}a_{13} + \frac{\mu_4^2}{\mu_0}a_{33}.$$

The estimates of the partial derivatives are used to compute the coefficients of the first and second fundamental forms of the surface [7].

$g_{11} = 1 + f_u^2$, $g_{22} = 1 + f_v^2$, $g_{12} = g_{21} = f_u^2 f_v^2$,

where $\boldsymbol{G} = \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix}$ is the first fundamental form of the surface,

$$d_{11} = \frac{f_{uu}}{\sqrt{1 + f_u^2 + f_v^2}}, \quad d_{12} = d_{21} = \frac{f_{uv}}{\sqrt{1 + f_u^2 + f_v^2}},$$

$$d_{22} = \frac{f_{vv}}{\sqrt{1 + f_u^2 + f_v^2}},$$

where $\boldsymbol{D} = \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix}$ is the second fundamental form of the surface. (2.2.5)

The principal curvatures are the roots of the quadratic equation [7]

$$|\boldsymbol{G}|\kappa_n^2 - (g_{11}d_{22} + d_{11}g_{22} - 2g_{12}d_{12})\kappa_n + |\boldsymbol{D}| = 0, \quad n = 1, 2. \tag{2.2.6}$$

The directions of the principal curvatures are given by $t_n = (1, \lambda_n, f_u + \lambda_n f_v)$ [7] where

$$\lambda_n = -\frac{d_{11} - \kappa_n g_{11}}{d_{12} - \kappa_n g_{12}} = -\frac{d_{21} - \kappa_n g_{21}}{d_{22} - \kappa_n g_{22}}, \quad n = 1, 2. \tag{2.2.7}$$

The Mean curvature H and the Gaussian curvature K are given by

$$H = \frac{\kappa_1 + \kappa_2}{2}, \quad K = \kappa_1 \kappa_2. \tag{2.2.8}$$

Table 1
Qualitative surface types

K	H		
	−	0	+
−	Saddle ridge Type = 4	Minimal surface Type = 6	Saddle valley Type = 8
0	Ridge surface Type = 3	Planar surface Type = 2	Valley surface Type = 7
+	Peak surface Type = 1	None	Pit surface Type = 5

The feature extraction algorithm extracts the surface curvature properties at each point on the range surface. The output of the segmentation process is a data structure which describes each point on the surface in terms of its Mean, Gaussian and principal curvature values and the directions of the principal curvatures and the surface normals.

2.3. Qualitative surface descriptions from H–K signs

The signs of the Mean and Gaussian curvature are used to come up with eight qualitative surface types [8]. These surface types are tabulated in *Table 1*. The surface types that are of interest to us are cylindrical, conical and ellipsoidal surfaces. Cylindrical and conical surfaces are characterized by zero Gaussian curvature and non-zero Mean curvature (types 3 and 7) whereas ellipsoidal surfaces are characterized by positive Gaussian curvature (types 1 and 5). The ellipsoidal surfaces that are of special interest are spherical surfaces for which $H^2 = K$. The qualitative surface descriptions enable us to use localization techniques which are specific to that particular surface type.

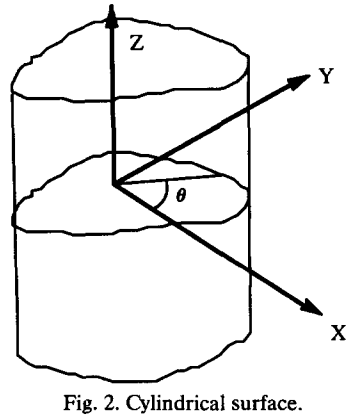


Fig. 2. Cylindrical surface.

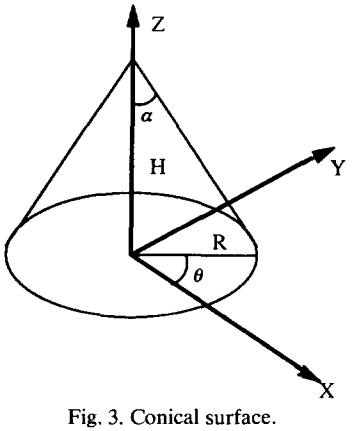


Fig. 3. Conical surface.

2.4. Cylindrical and conical surfaces

With reference to *Fig. 2* cylindrical surfaces are represented by the equation $x(\theta, z) = (x(\theta), z)$ [7]. The equation for the surface normals is given by

$$n(\theta, z) = \frac{x_\theta \times x_z}{|x_\theta \times x_z|}. \tag{2.4.1}$$

For right circular cylindrical surfaces $x(\theta) = r \cos\theta$ and $y(\theta) = r \sin\theta$. Hence $n(\theta, z) = (\cos\theta, \sin\theta, 0)$. Mapping the surface normals (needle map) onto the Gaussian sphere yields a great circle with the orientation of the plane of the great circle in the direction of the axis of the cylindrical surface.

Conical surfaces are represented by the equation $x(\theta, z) = (r(z)\cos\theta, r(z)\sin\theta, z)$ [7]. For right circular conical surfaces, (*Fig. 3*) $r(z) = R(1 - z/H)$, hence the equation for the surface normals is given by

$$n(\theta, z) = \frac{x_\theta \times x_z}{|x_\theta \times x_z|} = \frac{\left(\cos\theta, \sin\theta, \frac{R}{H}\right)}{\sqrt{1 + \left(\frac{R}{H}\right)^2}}. \tag{2.4.2}$$

The needle map maps onto a small circle on the Gaussian sphere with the orientation of the plane of the small circle, the same as the axis of the conical surface. The distance of the small circle from the origin is given by $d = R/\sqrt{R^2 + H^2} = \sin\alpha$. Thus detecting the orientation of the axis of the cone or the cylinder boils down to estimating the orientation of the plane of the great or small circle on the Gaussian sphere on which the needle map of the surface is mapped. The value of d can be used to distinguish conical surfaces from cylindrical surfaces since for cylindrical surfaces d is zero whereas for conical surfaces it is non-zero.

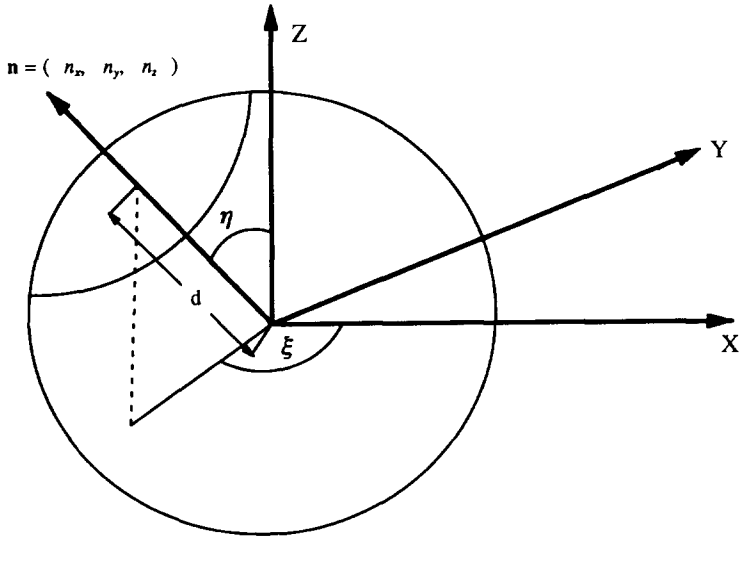


Fig. 4. Mapping the needle map on the Gaussian sphere.

2.5. Detection of planes on the Gaussian sphere

As shown in *Fig. 4* the equation of the plane on the Gaussian sphere is given by

$$d = n_{xi} \cos \xi \sin \eta + n_{yi} \sin \xi \sin \eta + n_{zi} \cos \eta, \tag{2.5.1}$$

where $\boldsymbol{n}_i = (n_{xi}, n_{yi}, n_{zi})$ is the input needle map. The plane on the Gaussian sphere can be uniquely characterized by the triple (d, ξ, η). A Hough transform technique is the most obvious way of estimating the parameters (d, ξ, η) from the input needle map. The parameter d distinguishes cylindrical surfaces from conical surfaces since $d = 0$ for a cylindrical surface whereas for a conical surface d is non-zero. The parameters ξ and η give the orientation of the axis of the conical or cylindrical surface on the Gaussian sphere.

3. The use of qualitative features for localization of curved surfaces

In our earlier experiments on the localization of curved surfaces [1] we had used the values of the Mean and Gaussian curvatures to match points on the scene and the model surface. The directions of the principal curvatures and the surface normals are used to compute the geometric transformation which would place the point on the model surface in registration with the corresponding point on the scene surface. For each match of a point on the model surface with a point on the scene surface, the geometric transformation can be represented by a 6-tuple where the 6-tuple represents the six parameters of rigid body motion – three translations one along each of the coordinate axes and three rotations one about each of the coordinate axes. This approach though has several drawbacks which could be listed as: (i) presence of several spurious hypotheses in the Hough space due to the presence of multiple objects with objects partially occluding each other resulting in subsequent loss of accuracy of localization, (ii) inability to detect planar and umbilic surfaces, (iii) inability to take into account the axes of symmetry of the modeled objects. These drawbacks could be alleviated using qualitative features. The use of qualitative features enables the use of a parameter space specific to a particular surface type. This enables one to take into account the axes of symmetry of the modeled objects and enables one to localize planar and umbilic surface types. It also reduces the combinatorial complexity of the search space and thereby reduces the number of spurious hypotheses in the Hough space.

The procedure for using qualitative features in the recognition and localization process could be described in the following step-wise manner:

(1) Feature extraction: The analytic surface fitting technique described in the previous section was used to compute the Mean, Gaussian and principal curvature values and the directions of the principal curvatures and the surface normals at each pixel in the range image.

(2) Detection of surface discontinuities:

(i) Step edges: Step edges were detected and localized in order to segregate the various objects in the scene. Since the depth discontinuities denote a high value for the surface gradient, the square root of the determinant of the first fundamental form matrix was used as a criterion for the detection of step edges.

$$\sqrt{|\boldsymbol{G}|} = \sqrt{g_{11}g_{22} - g_{12}^2} = \sqrt{1 + f_u^2 + f_v^2}. \tag{3.1}$$

The range image pixels where the values of $\sqrt{|\boldsymbol{G}|}$, exceeded a predefined threshold were selected as step edge points.

(ii) Roof edges: The maximum angular difference between adjacent surface normals gives the roof edge magnitude M_{roof} at a pixel location in the image.

$$M_{\text{roof}}(x, y) = \max\left(\cos^{-1}[\boldsymbol{n}(x, y) \cdot \boldsymbol{n}(x + k, y + 1)],\right.$$
$$\left. -1 \leq k, 1 \leq +1\right). \tag{3.2}$$

The range image pixels where the value of M_{roof} exceeds a specific threshold were selected as roof edge pixels.

(3) Pixel classification: Based on the signs of the Mean and Gaussian curvature (H–K signs) the range pixels were classified as belonging to one of eight qualitative surface types described in *Table 1*.

(4) Pixel grouping: Pixels were grouped spatially into surface patches homogenous in the H–K signs using a region-growing algorithm. The knowledge of the location of the step and roof edges was used to prevent accidental region-

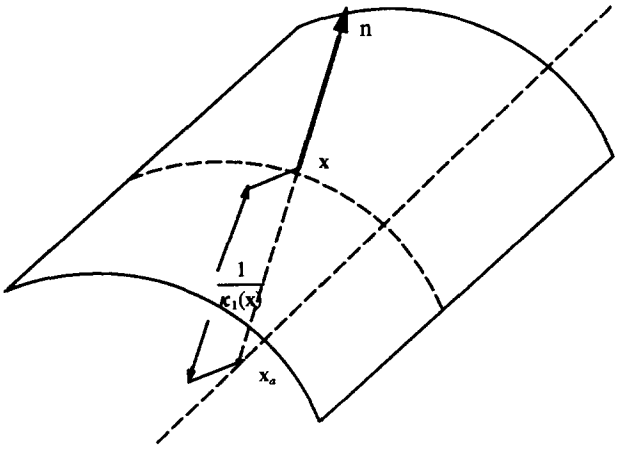


Fig. 5. Determining the axis point of a cylindrical surface.

growing over the surface discontinuities.

(5) Identification and localization of surface patches: Surface patches were localized in 3-D space using identification and localization techniques specific to the qualitative surface type. Since the surface types of interest were conical, cylindrical and spherical, localization techniques specific to these surface types were used in the experiment.

Localization techniques specific to conical, cylindrical and spherical surface types are discussed in the following subsections.

3.1. Localization of a cylindrical surface

A cylindrical surface is localized by the orientation of its axis, the value of its radius and the location of its centroid. The orientation of the axis is estimated as mentioned in the previous section. The radius of the cylinder is obtained by averaging the maximum principal curvature value κ_1 over the cylindrical surface. The centroid is obtained by finding the mean of the axis points. With reference to *Fig. 5*, for each point x on the cylindrical surface the corresponding axis point is given by:

$$x_a(x) = x - n(x)\left(\frac{1}{\kappa_1(x)}\right). \tag{3.1.1}$$

The centroid is given by:

$$x_0 = \frac{1}{N}\sum x_a(x) \tag{3.1.2}$$

and the radius is given by:

$$R = \frac{1}{N}\sum \frac{1}{\kappa_1(x)}. \tag{3.1.3}$$

Thus the cylinder is completely identified and localized by the feature vector $(c_x, c_y, c_z, \xi, \eta, r)$ where (c_x, c_y, c_z) is the center of the cylinder, (ξ, η) is the orientation of the axis and r is the radius of the cylinder.

3.2. Localization of a conical surface

A conical surface is localized by the orientation of its axis, the value of α and the location of the apex of the cone. With reference to *Fig. 6* the

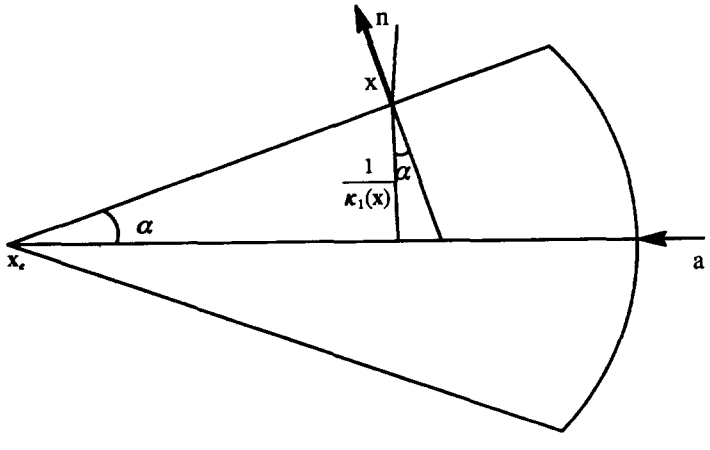


Fig. 6. Determining the apex of the conical surface.

location of the apex with respect to a single point x on the conical surface is given by:

$$x_a(x) = x - n(x)\left(\frac{1}{\kappa_1(x)\cos\alpha}\right) + a\left(\frac{2}{\kappa_1(x)\sin 2\alpha}\right), \quad (3.2.1)$$

Where $n(x)$ is the unit surface normal at point x on the conical surface, $\kappa_1(x)$ is the maximum principal curvature at x and a is the unit vector in the direction of the axis of the conical surface. The points x_a computed for each surface point x form a cluster in 3-D space. The actual value of the apex is taken to be the center of the cluster of x_a values.

$$\hat{x}_a = \frac{1}{N}\sum x_a(x). \quad (3.2.2)$$

Thus the cone is completely specified by the feature vector $(a_x, a_y, a_z, \xi, \eta, \alpha)$ where (a_x, a_y, a_z) is the location of the apex, (ξ, η) is the orientation of the axis and α is the angle of the cone.

3.3. Localization of a spherical surface

The needle map of a spherical surface is uniformly distributed on the Gaussian sphere. Consequently, the process of histogramming on the Gaussian sphere does not yield the parameters of the spherical surface. A conventional technique of determining the parameters of the spherical surface is by directly using the equation of the spherical surface

$$(x-a)^2 + (y-b)^2 + (z-c)^2 = r^2. \quad (3.3.1)$$

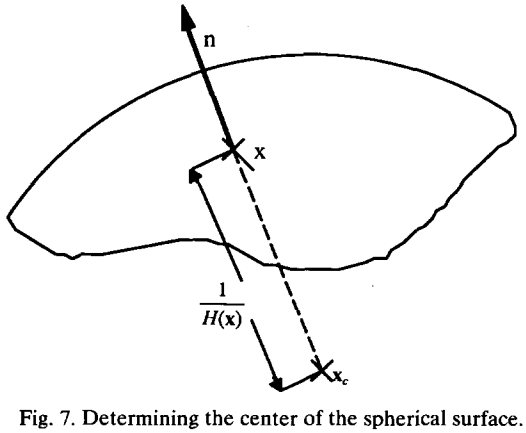


Fig. 7. Determining the center of the spherical surface.

Since a spherical surface is localized by the location of its center and the value of its radius, histogramming in the (a, b, c, r) parameter space would localize the spherical surface.

Alternatively, the spherical surface can be localized by clustering in the image domain instead of the parameter domain. With reference to *Fig. 7*, the location of the center with respect to the point x on the spherical surface is given by

$$x_c(x) = x - n(x)\left(\frac{1}{H(x)}\right). \quad (3.3.2)$$

The points x_c computed for each point on the spherical surface form a cluster in 3-D space. The center of this cluster is taken to be the center of the spherical surface.

$$\hat{x}_c = \frac{1}{N}\sum x_c(x). \quad (3.3.3)$$

The value of the radius is given by

$$R = \frac{1}{N}\sum\frac{1}{H(x)}. \quad (3.3.4)$$

Thus the spherical surfaces are uniquely identified and localized by the feature vector (c_x, c_y, c_z, r) where (c_x, c_y, c_z) is the center of the sphere and r is its radius.

The use of qualitative features in the recognition and localization process has the following advantages:

(1) A representation which is most specific for the surface type in question can be used for recognition.
(2) The choice of localization parameters can be made specific to the representation of that particular surface type.

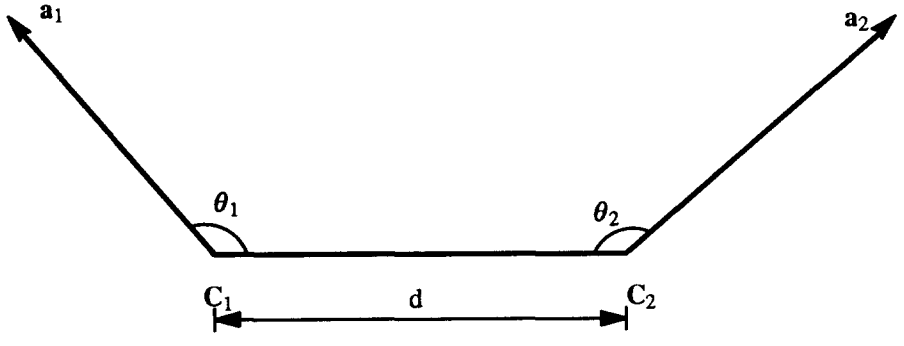
4. Matching and pose determination

As discussed earlier, the process of matching could be looked upon as a constraint propagation/constraint satisfaction problem. In fact the problem of 3-D object recognition most often involves a tradeoff between the complexity of representation and the complexity of constraint propagation/constraint satisfaction technique used. A higher-order relational description of the image data would imply a graph-theoretic constraint propagation technique based on subgraph

isomorphism or maximal clique detection whereas a description in terms of primitive geometric features implies a constraint propagation technique based on Hough clustering or searching through an Interpretation Tree. A more complex representation reduces the time complexity of the constraint propagation/constraint satisfaction process and vice versa.

In our earlier experiments [1] we tried to match surfaces by matching each point on the model surface with an appropriate point on the scene surface. The values of the Mean and Gaussian curvature were used to restrict the possible matches and the local coordinate system at each point formed by the directions of the principal curvatures and the direction of the surface normal was used to compute the transformation that would place the point on the model surface in registration with the point on the scene surface. Hough clustering in the six-dimensional parameter space of rigid body motion parameters was used to compute the global pose. The segmentation effort in this case was minimal since all that was involved was fitting an analytic surface in a local window centered around a surface point and computing the surface curvature properties from the parameters of the analytic surface. This approach apart from being intensive in memory and time also caused a number of spurious peaks in the Hough space especially when dealing with a multiple object scene with partial occlusion. This made recognition and localization very difficult.

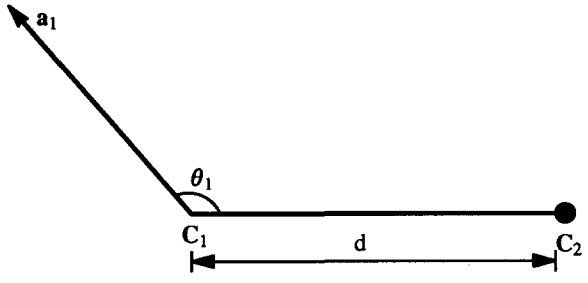
The approach followed in this paper involves, in addition to computation of the curvature values at every point, classification of surfaces into one of the afore-mentioned qualitative surface types based on the signs of the Mean and Gaussian curvature (H–K signs), grouping of surface pixels identical in H–K signs into homogenous regions, extraction of recognition and localization features characteristic of that particular surface type and the formation of dihedral feature junctions. Although this approach involves a greater effort in segmentation, it is far less compared to the segmentation effort required to come up with a relational description. The fact that most of the segmentation techniques in this paper are either based on pixel-level operations in a local window or based on histogramming makes the segmenta-



C_1 : apex of conical surface
centroid of cylindrical surface
centroid of planar surface
a_1 : axis of conical or cylindrical surface
normal to planar surface

C_2 : apex of conical surface
centroid of cylindrical surface
centroid of planar surface
a_2 : axis of conical or cylindrical surface
normal to the planar surface

Fig. 8. Type I: Dihedral feature junction pair.



C_1 : apex of conical surface
centroid of cylindrical surface
centroid of planar surface
a_1 : axis of conical or cylindrical surface
normal to planar surface

C_2 : centroid of spherical surface

Fig. 9. Type II: Dihedral feature junction.

tion easily amenable to parallelism on the Connection Machine™.

4.1. Dihedral feature junctions

The recognition and localization features for conical, cylindrical and spherical surfaces can be used to form what we term as *dihedral feature junctions*. Dihedral junctions are junctions with a single vertex and two incident edges. The use of dihedral junctions in the recognition and localization of polyhedral solids has been explored with satisfactory results [19]. *Dihedral feature junctions* differ from dihedral junctions in polyhedra in the sense that dihedral feature junctions do not correspond to physical junctions in the scene but are junctions formed from the recognition and localization features for the afore-mentioned surface types. Representation of the object models in the form of dihedral feature junctions in viewpoint independent. This offers a considerable advantage in terms of memory storage requirements

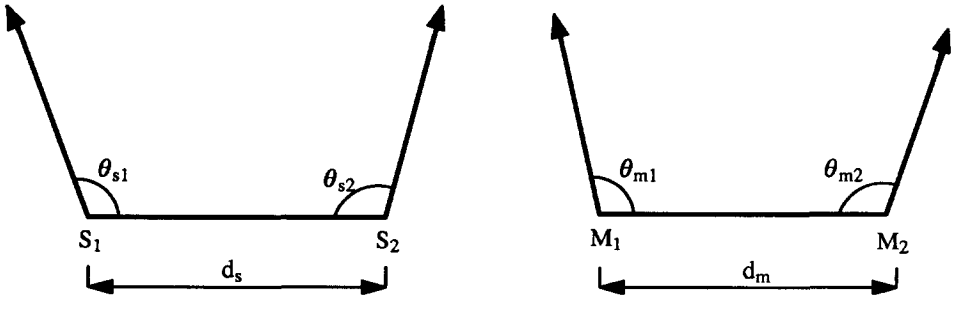


Fig. 10. Matching dihedral feature junction pairs (Type I).

over representation techniques which are viewpoint dependent and need to store several characteristic views of the objects [18].

4.1.1. Dihedral feature junction types

Two types of dihedral feature junctions are encountered in the recognition and localization of curved objects made up of piecewise combinations of conical, cylindrical, spherical and planar surfaces.

Type I: *Dihedral Feature Junction Pair* which is encountered when dealing with pairwise combinations of conical, cylindrical and planar surfaces. A typical dihedral feature junction pair is shown in *Fig. 8*.

Type II: *Dihedral Feature Junction* which is encountered when dealing with a spherical surface in combination with a conical, cylindrical or planar surface. A typical dihedral feature junction is shown in *Fig. 9*.

A pair of spherical surfaces does not constitute a valid dihedral feature junction and is not considered for the purpose of matching and pose computation. Both, the range image data and the model data are represented in terms of Type I and Type II dihedral feature junctions.

4.2. Matching of dihedral feature junctions

The matching procedure for each of the dihedral feature junction types is as discussed below:

Type I: With reference to *Fig. 10* the two dihedral feature junction pairs are said to match if they satisfy the following unary and binary constraints.

Unary constraints:
Surface type S_1 = Surface type M_1
Surface type S_2 = Surface type M_2
Angle θ_{s1} = Angle θ_{m1}
Angle θ_{s2} = Angle θ_{m2}
If S_1 and M_1 are cylindrical then
radius of S_1 = radius of M_1
If S_1 and M_1 are conical then
angle of S_1 = angle of M_1
If S_2 and M_2 are cylindrical then
radius of S_2 = radius of M_2
If S_1 and M_2 are conical then
angle of S_2 = angle of M_2.
Binary constraints:
Distance d_s = Distance d_m.

Type II: With reference to *Fig. 11*, the two dihedral feature junctions are said to match if they satisfy the following unary and binary constraints:
Unary constraints:
Surface type S_2 = Surface type M_2 = spherical
Radius of S_2 = Radius of M_2
Surface type S_1 = Surface type M_1
Angle θ_{s1} = Angle θ_{m1}
If S_1 and M_1 are cylindrical then
radius of S_1 = radius of M_1
If S_1 and M_1 are conical then
angle of S_1 = angle of M_1.
Binary constraints:
Distance d_s = Distance d_m.

4.3. Pose determination

The technique for pose determination for dihedral feature junction of Type I is discussed first. The pose determination technique for the dihedral feature junction of Type I is shown to be an extension of the pose determination technique for the dihedral feature junction of Type II.

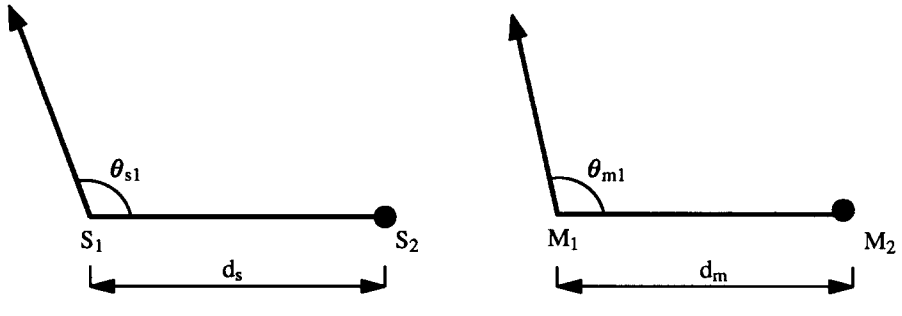


Fig. 11. Matching dihedral feature junctions (Type II).

4.3.1. Pose determination for Type II dihedral feature junction

For a successful match between a scene dihedral feature junction and a model dihedral feature junction the viewpoint parameters are computed as described in the remainder of this section. The description is given in homogeneous coordinate systems and transformations [20]. The coordinates (x, y, z) refer to the model coordinate system and (u, v, w) to the scene coordinate system. The operations $\times$ and $\cdot$ denote the vector cross product and the vector scalar product respectively.

With reference to *Fig. 12* let $\boldsymbol{m}_1$ be the unit vector in the direction $\boldsymbol{BA}$ and $\boldsymbol{m}_2$ be the unit vector in the direction $\boldsymbol{BC}$. Similarly, let $\boldsymbol{s}_1$ be the unit vector in the direction $\boldsymbol{ED}$ and $\boldsymbol{s}_2$ be the unit vector in the direction $\boldsymbol{EF}$. The homogeneous coordinates of B in the model coordinate system are given by the column vector $[x_0, y_0, z_0, 1]^T$ are the homogeneous coordinates of E in the scene coordinate system are given by the column vector $[u_0, v_0, w_0, 1]^T$. The goal is to find a transformation $\mathbf{T}$ such that

$$\mathbf{T}[x_0, y_0, z_0, 1]^T = [u_0, v_0, w_0, 1]^T. \quad (4.3.1.1)$$

The transformation $\mathbf{T}$ is determined in a stepwise manner as outlined below:

(1) Points B and E are translated to their respective origins. Let TRANS$(-B)$ and TRANS$(-E)$ denote the respective homogeneous transformations. This ensures that both junctions have their vertices translated to the origin.

(2) The vectors $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$ are rotated about an axis $\boldsymbol{k} = (k_x, k_y, k_z)$ (where $\boldsymbol{k}$ is the unit vector in the direction of the axis of rotation) by a scalar magnitude of rotation θ. The corresponding homogeneous transformation is denoted by ROT$(\boldsymbol{k}, \theta)$. ROT$(\boldsymbol{k}, \theta)$ aligns $\boldsymbol{m}_1$ with $\boldsymbol{s}_1$ and $\boldsymbol{m}_2$ with $\boldsymbol{s}_2$. The values of $\boldsymbol{k}$ and θ are computed as follows [21]:

$$\boldsymbol{k} = \frac{(\boldsymbol{m}_1 - \boldsymbol{s}_1) \times (\boldsymbol{m}_2 - \boldsymbol{s}_2)}{\|(\boldsymbol{m}_1 - \boldsymbol{s}_1) \times (\boldsymbol{m}_2 - \boldsymbol{s}_2)\|} \quad (4.3.1.2)$$

$$\cos\theta = 1 - \frac{[1 - (\boldsymbol{m}_1 \cdot \boldsymbol{s}_1)]}{[1 - (\boldsymbol{k} \cdot \boldsymbol{m}_1)(\boldsymbol{k} \cdot \boldsymbol{s}_1)]} \quad (4.3.1.3)$$

$$\sin\theta = \frac{[(\boldsymbol{k} \times \boldsymbol{s}_1) \cdot \boldsymbol{m}_1]}{[1 - (\boldsymbol{k} \cdot \boldsymbol{m}_1)(\boldsymbol{k} \cdot \boldsymbol{s}_1)]}. \quad (4.3.1.4)$$

(3) The final transformation can be thus written as:

$$\text{ROT}(\boldsymbol{k}, \theta)\text{TRANS}(-B)[x_0, y_0, z_0, 1]^T = \text{TRANS}(-E)[u_0, v_0, w_0, 1]^T. \quad (4.3.1.5)$$

From (4.3.1.1) and (4.3.1.5)

$$\mathbf{T} = \text{TRANS}^{-1}(-E)\text{ROT}(\boldsymbol{k}, \theta)\text{TRANS}(-B). \quad (4.3.1.6)$$

The transformation $\mathbf{T}$ from the model coordinate system to the scene coordinate system could be thus written as:

$$\mathbf{T} = \text{ROT}(\boldsymbol{k}, \theta)\text{TRANS}(t_x, t_y, t_z) = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (4.3.1.7)$$

where

$$r_{11} = k_x^2(1 - \cos\theta) + \cos\theta$$

$$r_{12} = k_x k_y(1 - \cos\theta) - k_z \sin\theta$$

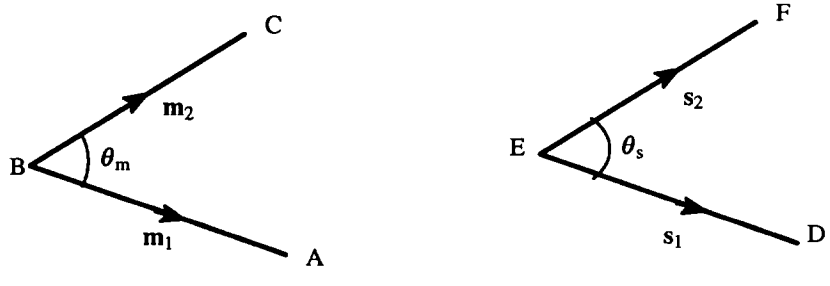


Fig. 12. Matching candidate model and scence feature dihedral junctions.

$$r_{13} = k_x k_z(1 - \cos\theta) + k_y \sin\theta$$
$$r_{21} = k_x k_y(1 - \cos\theta) + k_z \sin\theta$$
$$r_{22} = k_y^2(1 - \cos\theta) + \cos\theta$$
$$r_{31} = k_x k_z(1 - \cos\theta) - k_y \sin\theta$$
$$r_{32} = k_y k_z(1 - \cos\theta) + k_x \sin\theta$$
$$r_{33} = k_z^2(1 - \cos\theta) + \cos\theta \qquad (4.3.1.8)$$

and

$$t_x = u_0 - r_{11}x_0 - r_{12}y_0 - r_{13}z_0$$
$$t_y = v_0 - r_{21}x_0 - r_{22}y_0 - r_{23}z_0$$
$$t_z = w_0 - r_{31}x_0 - r_{32}y_0 - r_{33}z_0. \qquad (4.3.1.9)$$

The axis of rotation k could be alternatively expressed by the pair (ξ, η) where

$$k_x = \cos\xi \sin\eta,\ k_y = \sin\xi \sin\eta,\ k_z = \cos\eta, \qquad (4.3.1.10)$$

where $-\pi < \eta < \pi$ and $0 < \xi < 2\pi$.

The transformation **T** is thus uniquely specified by the 6-tuple $(t_x, t_y, t_z, \xi, \eta, \theta)$.

4.3.2. Pose determination for Type I dihedral feature junction pair

In order to determine the pose for a dihedral feature junction pair, the pose is computed for each of the constituent dihedral feature junctions as described in Section 4.3.1. The pose of the dihedral feature junction pair is the average of the poses computed for the individual dihedral feature junctions.

5. Implementation on the Connection Machine

The Connection Machine (CM-1) is a fine-grained SIMD architecture with 64K processors. It is particularly well suited for *data parallel* computation where several processors carry out simultaneous operations across large sets of data as contrasted with *control parallel* computation wherein parallelism results from multiple processors executing multiple threads of control. The two principal features of CM-1 that we have exploited in this work are (i) ability to treat the CM-1 processors as *virtual* processors which can be activated or deactivated under software control and (ii) programmability of the communication between processors.

Each of the 64K processors in CM-1 contains 4K bits of memory and a serial ALU. The processors are connected by a packet-switched network based on a boolean n-cube topology. In CM-1 there is a router for every group of 16 processors. The router handles communication between processors within the same group as also the communication between groups of processors. The router which are 4K in all, are configured as a 12 dimensional boolean hypercube. Each vertex of the hypercube could be looked upon as a fully connected set of processors.

In our work we have made use of the fact that a K-dimensional boolean hypercube can be mapped on an N-dimensional wrap-around mesh and vice versa in $O(1)$ time. The mapping algorithm can be summarized as follows:

Given a K-dimensional boolean hypercube and an N-dimensional wrap-around mesh $D_1 \times D_2 \times \ldots \times D_N$ where each $D_i = 2^{m_i}$ and $\sum_{i=1}^{N} m_i = K$. Let G_{m_i} be the m_i bit gray code representation of each processor along each dimension of the N-dimensional wrap-around mesh. Any processor in the N-dimensional wrap-around mesh can be represented by a sequence of gray codes $\langle G_{m_i} \rangle$ where each m_i bit gray code value represents the coordinate value of the processor along the ith dimension. It is also known that any processor in the K-dimensional boolean hypercube can be assigned a unique K bit gray code value. These K-bit gray code values maintain the adjacency property of the K-dimensional boolean hypercube in that any two K bit gray codes which differ in a single bit (i.e. at unit hamming distance) map onto processors which are adjacent in the K-dimensional boolean hypercube. It can be shown that the concatenation of the gray codes in the sequence $\langle G_{m_i} \rangle$ of a processor in the N-dimensional wrap-around mesh yields the K-bit gray code of a processor in the K-dimensional boolean hypercube. This mapping, moreover, preserves the adjacency property, i.e. adjacent processors in the N-dimensional wrap-around mesh map onto adjacent processors in the K-dimensional boolean hypercube. Similarly, a processor in the K-dimensional boolean hypercube can be mapped onto an N-dimensional wrap-around mesh by appropriately partitioning the K bit gray code along each of the dimensions of the mesh. This partitioning yields the gray code sequence $\langle G_{m_i} \rangle$ for the processor in the mesh. This

inverse mapping also preserves the adjacency property i.e. adjacent processors in the hypercube map onto adjacent processors on the mesh. Since most of the recognition techniques discussed in this paper are based on histogramming in N-dimensions, the ability to configure CM-1 as an N-dimensional wrap-around mesh proved extremely valuable.

Issues regarding implementation on the Connection Machine for the various phases of the recognition and localization process are discussed in this section.

(1) Feature extraction: The process of feature extraction includes surface fitting, curvature computation, step edge detection, roof edge detection, pixel classification and pixel grouping. All these operations are carried out by configuring the Connection Machine as a two-dimensional grid of the same size as the image. Thus each pixel in the range image is mapped onto a single processor. The surface fitting is carried out in a local $N \times N$ window (where N is odd) centered around each pixel. The computations involved communications in a local pixel neighbourhood using the NEWS communications. Curvature computation, roof and step edge detection and pixel classification were strictly on a single pixel basis and involved no communication between neighbouring pixels. Pixel grouping was carried out using an asynchronous region growing algorithm. Each pixel (i, j) was assigned an initial region label MAX$*i+j$ where MAX is the image size. For each pixel which is not an edge pixel the neighbouring pixels are checked for H–K signs and their corresponding region labels. If any of the neighbouring pixels has the same H–K sign and a lower region label, the region label of the center pixel is updated to the value of the lower region label. This process is carried out in parallel for all pixels in the image. The termination condition is when no pixel updates its region label.

(2) Localization of curved surfaces: The axis of the conical and cylindrical surfaces were determined by histogramming in the (d, ξ, η) space. Each point on the conical or cylindrical surface patch had to vote for the appropriate buckets in the (d, ξ, η) space based on the value of the components of the surface normal at that point. This voting process was carried out in parallel for each point on the surface patch. The maxima in the histogram was used to compute the orientation of the axis and so also distinguish between conical and cylindrical surfaces. A parallel averaging process was used to determine the centers and radii of spherical and cylindrical surfaces and the apexes of conical surfaces. Points in each of these regions was represented by a linked list. The averaging process was essentially a parallel divide-and-conquer technique that computed the average of a pair of elements. The complexity of the algorithm was $O(\log N)$ where N is the length of the list.

(3) Computation of dihedral feature junctions: The surface patches which were identified and localized were allocated to the first row and column of an $M \times M$ grid (where M = no. of surface patches). At each processor (i, j) in the grid the dihedral feature junction corresponding to surface patches i and j is computed. It should be noted that only the upper half or the lower half of the grid along any of the major diagonals is effectively utilized. The mapping therefore is not optimal in the sense of processor allocation. Only valid dihedral feature junctions (Type I or II) are considered for matching and pose computation.

(4) Matching and pose computation: The valid dihedral feature junctions were allocated to the first row and column of an $L \times L$ grid (where L = no. of valid dihedral feature junctions). At each processor (i, j) in the grid the dihedral feature junctions i and j are matched and if the match is successful the pose is computed. As before, only the upper half of or the lower half of the grid along any of the major diagonal is effectively utilized.

(5) Pose clustering: Pose clustering in the six-dimensional parameter space of rigid body motion is carried out by the process by histogramming in a six-dimensional accumulator. The Connection Machine software allows the processors to be configured in an n-dimensional grid [30]. Limited memory considerations necessitated a coarse initial quantization. After the initial histogramming, the contents of the accumulator were thresholded to identify the cells that contained at least a certain number of votes. These cells were tested recursively for peaks. Finer quantizations enabled a more accurate localization of peaks in parameter space.

6. Experimental results

The matching and localization technique based on dihedral feature junctions as described in the previous sections was experimentally verified. Since real range sensing facilities were not available to us, the experiments were carried out on synthesized range images. This to some extent weakens the conclusions of this paper but nevertheless brings out the advantages of exploiting parallelism in the process of 3-D object recognition and localization on a fine-grained SIMD machine such as the Connection Machine.

The flow chart of the entire recognition and localization process is shown in *Fig. 13*. Three candidate scenes shown in *Figs. 14*, *15* and *16* were considered. Each of the scenes contained three objects from different viewpoints. The objects are shown to partially occlude each other. Table 2 shows the experimental results in which both the theoretical and the experimentally computed poses are compared.

Our experimental results show that the rotation parameters in the pose computation are quite sensitive to the difference in the angles between the scene and the model dihedral feature junc-

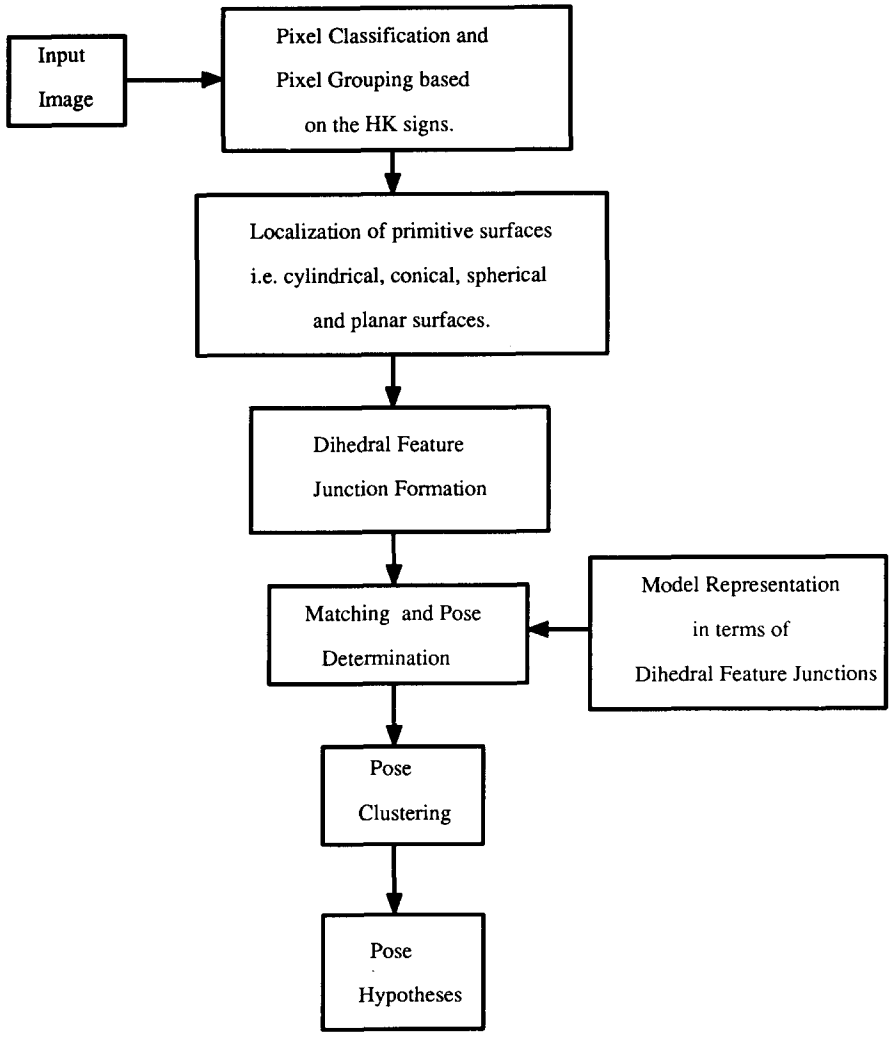


Fig. 13. Flowchart of the 3-D object recognition process.

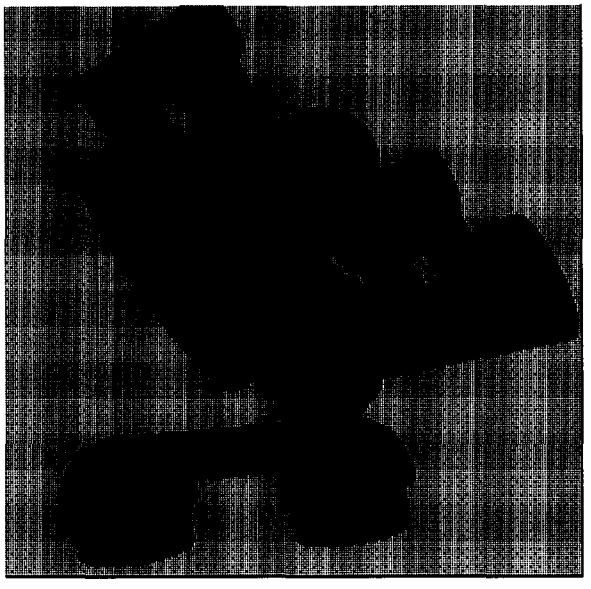


Fig. 14. Range image of Scene I.

Fig. 15. Range image of Scene II.

tions. On the other hand the estimation of translation parameters was found to be far more robust. The experimental findings were found to be in conformity with a formal sensitivity analysis reported elsewhere [31]. In scene II object 2 is shown to be fairly occluded. Yet, the experimentally computed pose is in close conformity with the theoretical pose. This brings out the robustness of the dihedral feature junctions when used for matching and localization. In scene III experimentally computed pose for object 3 is seen to differ considerably from the theoretical pose. This could be accounted for by the object symmetry. Since the object is symmetrical with respect to a rotation of π about the model z axis, the transform $\mathbf{T}$ is indistinguisable form the transform $\mathbf{T}\boldsymbol{R}_z(\pi)$ where $\boldsymbol{R}_z(\pi)$ is the matrix corresponding to the rotation about the model z axis with an angular magnitude of rotation $= \pi$. For the theoretical viewpoint of (150.0, 160.0, 1000.0, 347.23, 121.44, 91.74) the corresponding symmetric pose is (150.0, 160.0, 1000.0, 77.236, 138.669, −136.023) which agrees well with the experimentally computed viewpoint of (149.081, 160.763, 997.619, 75.938, 139.219, −132.188). The histogramming technique used is not capable of disambiguating symmetric poses.

7. Conclusions

In this paper we have shown how the use of a fine-grained SIMD machine such as the Connection Machine enables use to exploit parallelism in the process of 3-D object recognition and localization. The input images are range images of 3-D curved objects composed of piecewise combinations of conical, cylindrical and spherical sur-

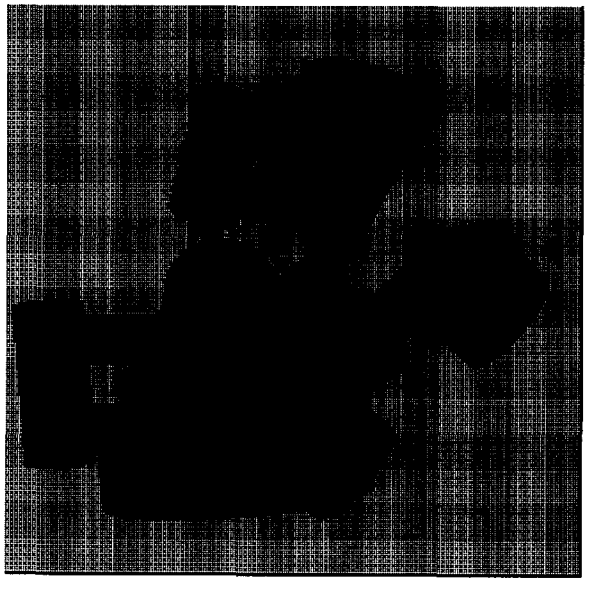


Fig. 16. Range image of Scene III.

Table 2
Experimental results

Scene	Viewpoint parameters	Object 1		Object 2		Object 3	
		Theoretical viewpoint	Experimental viewpoint	Theoretical viewpoint	Experimental viewpoint	Theoretical viewpoint	Experimental viewpoint
I	t_x	100.000	95.974	160.000	156.802	200.000	200.308
	t_y	100.000	102.689	160.000	165.329	100.000	99.936
	t_z	1000.000	1000.183	2000.000	1999.084	3000.000	2999.694
	ξ(deg)	2.329	2.633	5.264	2.633	327.240	329.063
	η(deg)	114.910	116.719	114.597	116.719	133.930	130.781
	θ(deg)	−113.483	−109.688	−84.673	−81.563	116.497	126.563
II	t_x	100.000	95.974	160.000	161.435	130.000	129.007
	t_y	100.000	102.689	160.000	159.824	180.000	180.502
	t_z	1000.000	1000.183	3000.000	3011.658	2000.000	1997.375
	ξ(deg)	2.329	2.633	5.264	14.063	354.736	357.188
	η(deg)	114.910	116.719	114.597	108.281	114.597	116.719
	θ(deg)	−113.483	−109.688	−84.673	−70.313	84.673	87.188
III	t_x	180.000	180.972	120.000	121.152	150.000	149.081
	t_y	100.000	98.392	140.000	137.936	160.000	160.764
	t_z	3000.000	2999.695	2000.000	2003.357	1000.000	997.620
	ξ(deg)	32.330	30.938	32.260	30.938	347.230	75.938
	η(deg)	127.370	127.969	132.770	133.594	121.440	139.219
	θ(deg)	−145.220	−143.438	−113.060	−109.688	91.740	−132.188

faces. Such objects constitute a significant majority of the objects encountered in an industrial environment. Hough clustering is chosen as the constraint propagation technique on account of its ease of parallelization. Qualitative surface classification based on the signs of the Mean and Gaussian curvature is used to come up with *dihedral feature junctions* which are used for matching and pose computation. Dihedral feature junctions are shown to be robust to occlusion and offer a viewpoint independent modeling scheme for the object models. Experimental results on the Connection Machine bring out the advantages of the proposed technique.

Acknowledgments

This research was partially supported by the US Airforce Grant F30602-C-0008 to the Northeast Artificial Intelligence Consortium monitored by the Rome Air Development Center, Rome, NY. The use of the Connection Machine by the Northeast Parallel Architectures Center at Syracuse University is also gratefully acknowledged.

References

[1] S.M. Bhandarkar and M. Suk, Recognition of multiple 3-D objects from qualitative surface descriptions, submitted to *International Journal of Computer Vision*.

[2] W.E.L. Grimson and T. Lozano-Perez, Localizing overlapping parts by searching the interpretation tree, *IEEE Trans. Pattern Analysis and Machine Intelligence* 9 (4) (1987) 469–482.

[3] O.D. Faugeras and M. Herbert, The representation, recognition and locating of 3-D objects, *The Int. Journal of Robotics Research* 5 (3) (1986) 27–52.

[4] G. Stockman, Object recognition and localization via pose clustering, *Computer Vision, Graphics and Image Processing* 40 (1987) 361–387.

[5] B.A. Boyter and J.K. Aggarwal, Recognition of polyhedra from range data, *EEE Expert* (1986) 47–59.

[6] M. Dhome and T. Kasvand, Polyhedra recognition by hypothesis accumulation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 9 (3) (1987) 429–438.

[7] I.D. Faux and M.J. Pratt, *Computational Geometry for Design and Manufacture* (Ellis Horwood Publishers, New York, 1972).

[8] P. Besl and R. Jain, Invariant surface characteristics for 3-D objects recognition in range images, *Computer Vision, Graphics and Image Processing* 33 (1986) 33–80.

[9] R.M. Haralick, L.T. Watson and T.J. Laffey, The topographic primal sketch, *Int. Journal of Robotics Research* 2 (1) (1983) 50–72.

[10] P. Besl and R. Jain, Segmentation through variable order surface fitting, *IEEE Transactions Pattern Analysis and Machine Intelligence* 10 (2) (1988) 167–192.

[11] R. Hoffman and A.K. Jain, Segmentation and classification of range images, *IEEE Transactions Pattern Analysis and Machine Intelligence* 9 (5) (1987).

[12] T.J. Fan, G. Medioni and R. Nevatia, Segmented descriptions of 3-D surfaces, *IEEE Journal on Robotics and Automation* 3 (6) (1987) 527–538.

[13] B.C. Vemuri and J.K. Aggarwal, Curvature-based representation of objects from range data, *Image and Vision Computing* 4 (2) (1986) 107–114.

[14] S.M. Naik and R.C. Jain, Spline-based surface fitting on range images for CAD applications, *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (1988) 249–253.

[15] B.C. Vemuri and J.K. Aggarwal, Representation and recognition of objects from dense range maps, *IEEE Trans. Cicuits and Systems* CAS-34 (11) (1987) 1351–1363.

[16] R.M. Haralick, Digital step edges from zero crossing of second directional derivatives, *IEEE Trans. Pattern Analysis and Machine Intelligence* 6 (1) (1984) 58–68.

[17] T. Nagata and H.B. Jha, Determining orientation, location and size of primitive surfaces by a modified hough transformation technique, *Pattern Recognition* 21 (5) (1988) 481–491.

[18] K. Ikeuchi, Generating an interpretation tree form a CAD model for 3-D object recognition in bin-picking tasks, *Int. Jnl. Computer Vision* (1987) 145–165.

[19] S.M. Bhandarkar and M. Suk, Hough clustering technique for surface matching, *Proc. IAPR Workshop on Computer Vision* Tokyo, Japan (Oct 12–14, 1988) 82–85.

[20] R.P. Paul, *Robot Manipulators: Mathematics, Programming and Control* (MIT Press, Cambridge, MA, 1981).

[21] G.A. Korn and T.M. Korn, *Mathematical Handbook for Scientists and Engineers* (Mc. Graw Hill, New York, 1972).

[22] R. Miller and Q. Stout, Geometric algorithms for digitized pictures on a mesh connected computer, *IEEE Trans. Pattern Analysis Machine Intelligence* 7 (1985) 216–218.

[23] R. Miller and Q. Stout, Data movement techniques for the pyramidal machine, *SIAM Journal of Comp.* 16 (1) (1987) 38–60.

[24] F.T. Leighton, Parallel computations using mesh of trees, Technical Report, MIT (1982).

[25] W.D. Hillis, *The Connection Machine* (MIT Press, Cambridge, MA, 1985).

[26] H. Li and M. Maresca, Polymorphic-torus architectures for computer vision, *IEEE Trans. Pattern Analysis Machine Intelligence* 11 (3) (1989) 244–257.

[27] R. Cypher, J.L.C. Sanz and L. Synder, Hypercube and shuffle-exchange algorithms for image component labeling, *Proc. 1987 Workshop on Computer Architecture for Pattern Analysis and Machine Intelligence* (1987) 5–10.

[28] R.S. Wallace and M.D. Howard, HBA vision architecture: Built and benchmarked, *IEEE Trans. Pattern Analysis and Machine Intelligence* 11 (3) (1989) 227–232.

[29] J.J. Little, G.E. Blelloch and T.A. Cass, Algorithmic techniques for computer vision on a fine-grained parallel architecture, *IEEE Trans. Pattern Analysis and Machine Intelligence* 11 (3) (1989) 244–257.

[30] G. Ramamoorthy, R.V. Shankar and M. Suk, Pose determination using model-vertex pairs on the connection machine, *Proc. IEEE Int. Workshop for Tools on AI* (1989).

[31] S.M. Bhandarkar and M. Suk, *Pattern Recognition* 24 (6) (1991) 505–513.