AVC89: Polyhedral Object Recognition with Sparse Data - Validation of Interpretations

Derrick Holder[1] and Hilary Buxton[2]

[1] City of London Polytechnic, Department of C.M.S.M.S., 100 Minories, London EC3N 1JY.

[2] Queen Mary College, Department of Computer Science, Mile End Road, London E1 4NS.

The method of Grimson, Lozano-Pérez and others, for the generation of feasible interpretations of scenes with sparse data, has been developed and implemented by the present authors on a distributed array processor, the AMT DAP, which operates in SIMD mode. Measurements involving the location vectors and the surface normals at m data points, considered in pairs, are compared with the maximum and minimum values associated with the n×n pairs faces of a polyhedral object model, in a process that exploits n×n parallelism.

This paper discusses the subsequent validation of the interpretations, in which data points have been assigned provisionally to object model faces.

In many applications, the key task for a robot's vision system is to supply the control unit with a quantitative and symbolic description of its surroundings, telling the robot what is where in the scene being viewed[1]. One approach to the problem is to enable a robot to identify an object from a set of known objects, and to locate it relative to the robot's sensors.

Murray and Cook[1], Grimson and Lozano-Pérez[2] Faugeras, Ayache and Faverjon[3], and others, consider objects in the form of separate, possibly non-convex, polyhedra, for which there are accurate geometric models. An object may have up to six degrees of freedom relative to the robot's sensors, which are assumed to be capable of providing three-dimensional information about the position and orientation of a small set of points on the object.

The general approach to the problem is based on the hypothesis, prediction and verification paradigm that is widely used in AI. First we generate feasible interpretations by means of simple, generally pairwise, geometric comparisons between object models and sensor data, and then we test the interpretations, in detail, for compatibility with the surface equations of a particular object model.

Several authors have implemented sequential algorithms, in which measurements involving the location vectors and the surface normals at m data points, considered in pairs, are compared with the values associated with points in n×n pairs of object model faces. It is found that, with simple geometric constraints applied to sparse data independently of the coordinate frame of reference, the possibilities can generally be whittled down to just a few, frequently only one, feasible interpretation. This is done without resort to a detailed solution of the surface equations. Nevertheless, in sequential form, the algorithms are not generally fast enough to offer a practical solution to the problem.

Until recently the best parallel algorithm for the generation of feasible interpretations was one implemented by Flynn and Harris on the Connection Machine at MIT[4]. This achieved parallelism at the expense of processor numbers which grew exponentially with problem size. However, a similar degree of parallelism has been achieved by the present authors[5] on a distributed array processor, the AMT DAP 510, which operates in SIMD mode, with a processor set that is only quadratic in the problem size. This enables problems to be handled that would otherwise far outstrip the capacity of the Connection Machine.

We note that these algorithms can equally well be applied to measurements at points on the edges of a polyhedron, with edge direction replacing surface normal.

Instead of using a small number of discrete measurements, edge matching generally involves the processing of a substantial volume of grey level data, and the production of a 2½D sketch[6,7]. Nevertheless, this form of input is efficiently provided by the ISOR system[8] developed at GEC Hirst Research Centre and currently being implemented on the AMT DAP at Queen Mary College.

The purpose of this paper is to discuss parallel algorithms for the validation of both face matching and edge matching interpretations of visual data. This is required because the interpretations are based only on simple geometric constraints, with no guarantee that the object model description is entirely consistent with the data.

THE VALIDATION METHOD

The problem is to validate an interpretation in which, in the case of face matching, sensory data, expressed in terms of position vectors

$$\mathbf{r}_i' = (x_i', y_i', z_i'),$$

measured to within some volume of error relative to the sensor, and *outward* unit normals

AVC 1989 doi:10.5244/C.3.4

$$\mathbf{n}_i^{\prime} = (a_i^{\prime}, b_i^{\prime}, c_i^{\prime}),$$

measured to within some cone of error, have been provisionally assigned, for $i = 1,2,\ldots,m$, to particular faces of a given polyhedral object model.

The models are described within a database, in terms of the Cartesian coordinates of their vertices, together with equations for the faces, of the form

$$a_j x + b_j y + c_j z = d_j,$$

i.e.

$$\mathbf{n}_j.\mathbf{r} = d_j,$$

where $\mathbf{r}$ is the position vector of a point in face j, in a localised coordinate system, $\mathbf{n}_j$ is the *outward* unit normal, d_j is the perpendicular distance from the origin and $j = 1,2,\ldots,n$.

An interpretation may be regarded as valid, only if it is possible to find a rotation and translation of the given object model that would result in each data point being placed sufficiently accurately on the surface of the object, and, in the case of visual rather than tactile data, with every data point visible, not being obscured by any part of the given object. The possibility of scaling an object model to match the data is not considered in the algorithms presented here.

The validation of an interpretation proceeds as follows:-

(i) For every feasible interpretation, establish the location and orientation of the given object model that is most compatible with the data;

(ii) Confirm that every data point is visible.

(iii) Confirm that every data point lies sufficiently close to, and within the perimeter of, the object model face to which it has been assigned;

In the case of edge matching, sensory data expressed in terms of position vectors and edge direction vectors

$$\mathbf{t}_i^{\prime} = (a_i^{\prime}, b_i^{\prime}, c_i^{\prime})$$

have been provisionally assigned to particular edges. The perpendicular vector $\mathbf{p}_i^{\prime}$ from the origin replaces $d_i^{\prime}$, but the object model database and the validation method are essentially the same.

LOCATION AND ORIENTATION OF THE OBJECT MODEL

A rigid body rotation and translation may be expressed in terms of a 3×3 orthogonal rotation matrix $\mathbf{R}$, and a translation vector $\mathbf{r}_0$,

$$\mathbf{r}_0 = (x_0, y_0, z_0),$$

so that

$$\mathbf{r}_j \to \mathbf{R}\mathbf{r}_j + \mathbf{r}_0,$$

$$\mathbf{n}_j \to \mathbf{R}\mathbf{n}_j$$

and

$$d_j \to d_j + \mathbf{R}\mathbf{n}_j.\mathbf{r}_0.$$

We have to determine $\mathbf{R}$ and $\mathbf{r}_0$ in such a way that, if j relates to the object model face to which the data point with subscript i has been assigned,

$$\mathbf{R}\mathbf{n}_j \approx \mathbf{n}_i^{\prime}$$

and

$$d_j + \mathbf{R}\mathbf{n}_j.\mathbf{r}_0 \approx d_i^{\prime}.$$

i.e.

$$d_j \approx d_i^{\prime} - \mathbf{n}_i^{\prime}.\mathbf{r}_0.$$

The orthogonality condition $\mathbf{R}^T\mathbf{R} = \mathbf{I}$ imposes 6 non-linear constraints on the elements of $\mathbf{R}$, and the application of the method of constrained least squares to the residual vectors $\mathbf{R}\mathbf{n}_j - \mathbf{n}_i^{\prime}$ leads to the condition

$$\mathbf{R}(\mathbf{S}+\mathbf{L}) = \mathbf{T},$$

where $\mathbf{S}$ and $\mathbf{T}$ are the 3×3 matrices whose elements are defined as follows:-

$$S_{11} = \sum_{i=1}^{m} a_j^2,\quad S_{12} = \sum_{i=1}^{m} a_j b_j,\quad S_{13} = \sum_{i=1}^{m} a_j c_j,$$

$$S_{21} = S_{12},\quad S_{22} = \sum_{i=1}^{m} b_j^2,\quad \text{etc.}$$

and

$$T_{11} = \sum_{i=1}^{m} a_i^{\prime} a_j,\quad T_{12} = \sum_{i=1}^{m} a_i^{\prime} b_j,\quad T_{13} = \sum_{i=1}^{m} a_i^{\prime} c_j,$$

$$T_{21} = \sum_{i=1}^{m} b_i^{\prime} a_j,\quad T_{22} = \sum_{i=1}^{m} b_i^{\prime} b_j,\quad \text{etc.,}$$

and $\mathbf{L}$ is a symmetric matrix of Lagrangian multipliers.

On the left hand side of the equation the matrix $\mathbf{S}$ is also symmetric, and it follows that $\mathbf{R}^T\mathbf{T}$ must be symmetric. We thus have three further linear conditions on the elements of $\mathbf{R}$.

It has been demonstrated [9] that the solution of these equations may be expressed in terms of singular value decomposition, with the best result selected from 4 possible rotations. However, the Newton-Raphson process readily lends itself to a parallel implementation. A good first approximation, obtained from the relationship

$$\mathbf{R}\mathbf{S} = \mathbf{T},$$

that applies when the data exactly fit the object model, avoids the likelihood of convergence to a spurious solution. In practice, the process converges to sufficient accuracy after just one or two iterations.

The solution for $\mathbf{r}_0$ is obtained much more easily, with the method of least squares applied to the residuals $d_j - d_i^{\prime} + \mathbf{n}_i^{\prime}.\mathbf{r}_0$.

We note that Faugeras, Ayache and Faverjon, work rather more compactly with the quaternions $(d_i^{\prime}, \mathbf{n}_i^{\prime})$ and $(d_j, \mathbf{n}_j)$ to achieve what appears to be an equivalent result, but presumably their algorithm is implemented in sequential form.

DATA VALIDATION

Having established the location and orientation of the given object model that is most compatible with the data, we may easily determine whether the locations of the data points are consistent with the object model face equations, but it remains to be verified that every data point lies within the perimeter of the face to which it has been assigned, and that it is not hidden from view by another part of the object model.

In order to do this we consider the intersections with the edges of a polygon, when a line is drawn from a given data point to some external point. There will be an odd number of intersections if the first point is inside the polygon, and an even number of points if it is outside. For example, the line joining Q to S in Figure 1 has 3 intersections with the edges of the polygon.

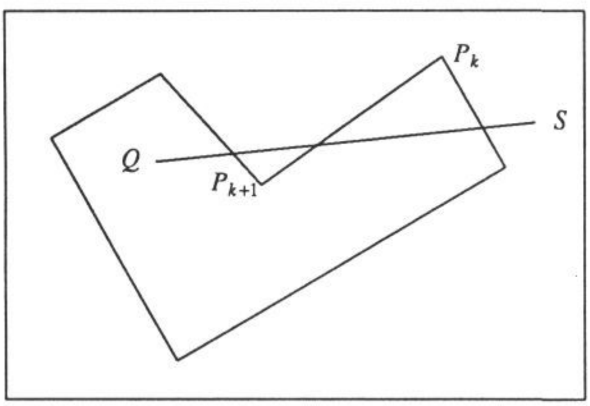

Figure 1. A Point within A Non-Convex Polygon

For the purpose of counting intersections, we choose a viewing plane that, for simplicity and efficiency, is orthogonal to one of the coordinate axes, looking towards the object, with the external point S at an infinite distance to the right of the viewing plane ($X_S \to \infty$). We then write the equation of a line segment, such as that joining P_k to P_{k+1}, in the form

$$AX + BY = C$$

where

$$A = (Y_{k+1} - Y_k),$$

$$B = (X_k - X_{k+1})$$

and

$$C = (X_k Y_{k+1} - X_{k+1} Y_k),$$

If point Q has coordinates (X_{Q,Y_Q}) in the viewing plane, the line joining Q to S intersects the line segment if, and only if, either

$$Y_{k+1} > Y_Q > Y_k$$

and

$$AX_Q + BY_Q < C,$$

in which case Q is to the left of the line segment, or

$$Y_{k+1} < Y_Q < Y_k$$

and

$$AX_Q + BY_Q > C,$$

so that Q is to the right of the line segment, when looking from P_k to P_{k+1}.

Then, for a given data point to be visible from the position of the sensor, it must lie in a face that is not directed away from the sensor, and its projection on the viewing plane must not fall within the perimeter of another face that is nearer to the sensor.

These requirements lead to the conditions

$$\mathbf{Rn}_j.\mathbf{r}_i \geq d_j + \mathbf{Rn}_j.\mathbf{r}_0,$$

i.e.

$$\mathbf{Rn}_j.(\mathbf{r}_i - \mathbf{r}_0) \geq d_j,$$

for all j such that the projection of data point i falls within the perimeter of the projection of face j onto the viewing plane.

The same visibility check applies when values of $\mathbf{r}_i'$ are derived from edge matching data, but it then has to be established that every data point is sufficiently close to the edge segment to which it has been assigned.

SIMD IMPLEMENTATION

There are three main directions in which parallelism may be exploited in the validation processes described above:-

(i) processes applied simultaneously to each data point;

(ii) processes applied simultaneously with regard to every object model face or edge;

(iii) processes applied simultaneously with regard to all of the edges associated with a given model face.

Parallelism of type (i) occurs throughout, from the summations required in determining the location and orientation of the object model and checking against face equations to the final data visibility check, whereas types (ii) and (iii) occur only during the visibility check.

The need for efficient calculation of the 3×3 matrices involved in the Newton Raphson process clearly calls for the exploitation of type (i) parallelism, which also leads to substantial gains in the latter stages of validation. A uniformity of approach is considered important, as this avoids the need for the restructuring of the database that would be necessitated by a shift of emphasis to fully exploit type (ii) or (iii).

However, there is a substantial improvement in overall performance when a limited restructuring of the object model and data are undertaken at run time, with a view to achieving $m \times m$ parallelism during the visibility check.

The first task in an SIMD implementation of the validation process is to map the model against the data in accordance with a given interpretation of the form

j = face(i), i=1,2,...,m.

so that

mapped_object(i,) = model(j,),

and to set the unused rows of the mapped object and data matrices to zero. The initial rotation matrix, the solution of the Newton Raphson equations and the translation vector for best fit are then computed using standard DAP Library subroutines. SHEP and SUM operations are used to maximise parallelism in setting up the equations for the Newton-Raphson process. The rotation matrix and the translation vector are replicated before rotation and translation of the object model.

Before proceeding with the validation of individual data points, the x,y and z coordinates of the vertices associated with given faces, originally stored sequentially in rows, are moved into the columns of separate DAP matrices, and the coordinates of the data points are replicated in columns using a simple but effective binary algorithm. As a consequence, when a given row of vertex coordinates is replicated and related to a matrix of data coordinates, the process simultaneously relates every data point to every object model face, and $m \times n$ parallelism is thus achieved. The organisation of the information within DAP matrices at this stage is illustrated in Figure 2.

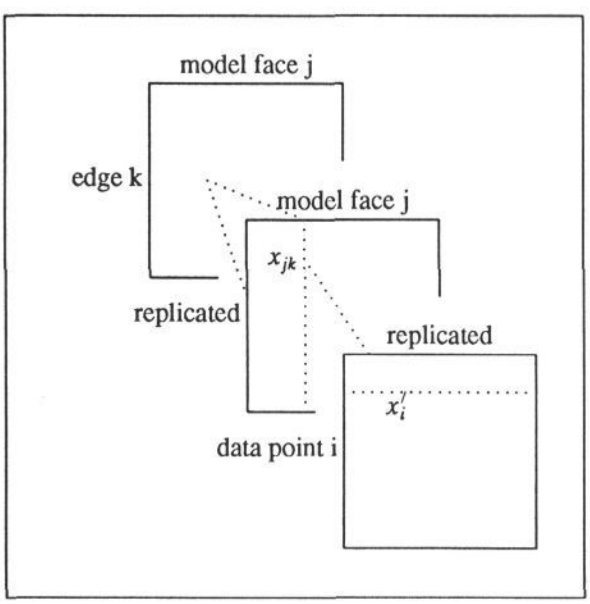

Figure 2. The Organisation of Information within DAP Matrices

Transforming into viewing coordinates and initialising a DAP logical matrix inside = .FALSE. we proceed to investigate intersections of the line joining each data point to the external point S with successive edges of every face, switching inside between .TRUE. and .FALSE. whenever an intersection occurs. It is thus rapidly established which data points fall inside which faces when these are all projected onto the viewing plane.

Then, with a fairly obvious nomenclature in DAP FORTRAN:-

```
in_face = inside(face,)
   .AND.(ABS(delta).LT.tolerance)
```

and

```
visible = (.NOT.back_face)
   .AND. ANDCOLS(in_front.OR.(.NOT.inside))
```

where the real matrix delta and the logical matrices back_face and in_front are the results from straightforward parallel calculations, and the function ANDCOLS collates results within a given row. Thus the process efficiently determines which data points lie sufficiently near to the face to which they have been assigned, and which if any are not visible from the position of the sensor.

TEST RESULTS

The validation process has been applied successfully to an L-shaped block with 8 faces, viewed as illustrated in Figure 3, and with 8 data points, three of which are in fact visible, at the centre of each face. The total processing time required to determine the location and orientation of the object and to validate the data is about 42 milliseconds.

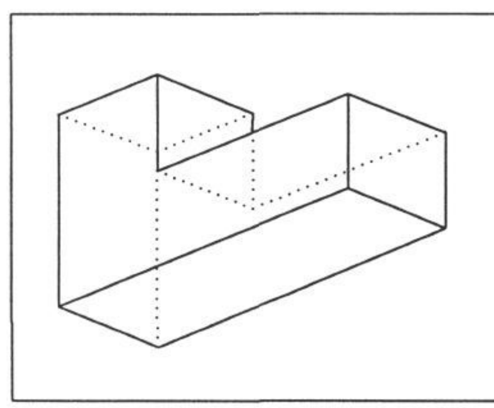

Figure 3. A View of an L-Shaped Block.

Further tests on the validation process involved three different views of a three-pin electric plug with 27 faces. The first view of the plug, with 14 visible faces, is illustrated in Figure 4. In the second view, looking upwards with 12 visible faces, the pins were partially obscured. The third view, was looking directly down onto the pins, with 4 faces clearly visible and 5 more at an oblique angle to the sensor.

The data for View 1 consisted of 14 data points at the centre of each visible face, the one on the visible side of the plugbeing obscured by the flange. Three back face points were also included in the test data. There were 17 data points for View 2, including two in faces of the earth

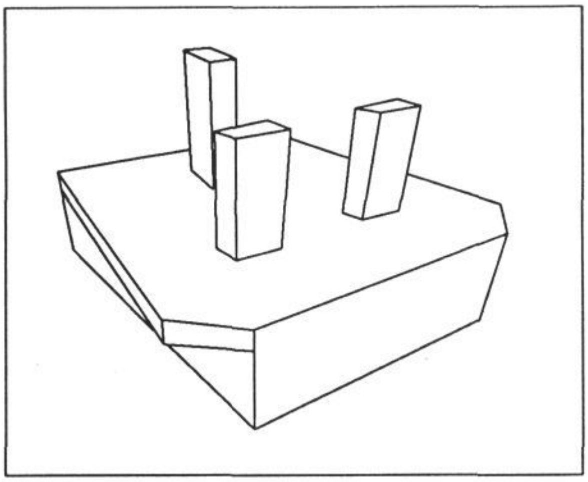

Figure 4. - The Electric Plug (View 1)

pin that were obscured by the neutral pin and one in the underside of the flange on the far side of the plug, that was totally obscured by the rest of the plug. There were also four back face data points for View 2. For View 3 there were 12 data points, three in back faces but none otherwise obscured.

The location and orientation of the plug were determined, in each case, within about 26.5 milliseconds, and the back face and obscured data points were identified by the validation process in a further 12 milliseconds. This brought the overall processing time, including that required to generate a feasible interpretation, to about 90 milliseconds. A complete breakdown of the validation processing times for the electric plug is given in Table 1.

	processing time (milliseconds)	
Convert to DAP format		0.4
Map object against data		0.9
Compute $\mathbf{S},\mathbf{T}$ and initial $\mathbf{R}$	5.9	
Newton-Raphson for $\mathbf{R}$	11.1	
Rotate object	4.3	
Compute $\mathbf{U}$ and $\mathbf{v}$	1.6	
Solve for $\mathbf{r}_0$	2.3	
Translate object	1.3	
Total to locate object		26.5
Replicated viewing coords	2.0	
Check for inside face	8.2	
Check for in face, in front, in back face	2.0	
Total to check data		12.2
Total to validate interpretation		40.0

Table 1. Processing Times for the Electric Plug

Further tests were then made with simulated errors in the spatial coordinates of the data points, and the surface normal directions.

It was found that, whereas coordinate errors of about 0.05 inches might simply result in the rejection of the offending data points, with the electric plug being about 1.5 inches across and viewed from a distance of about 5 inches, errors of the order of 0.25 inches resulted in substantial errors in $\mathbf{r}_0$, leading to the rejection of several valid points. The orientation of the plug, and the run times for validation, were not affected by errors in spatial coordinates.

On the other hand, errors ranging from 0.1 to 0.2 in the direction cosines of the surface normals led to errors in both $\mathbf{R}$ and $\mathbf{r}_0$, with the subsequent rejection of several valid points. Again, there was no change in run times, because the errors were not sufficient to provoke further iterations of the Newton Raphson process, in computing $\mathbf{R}$.

A possible strategy for dealing with errors such as these would be to recompute $\mathbf{R}$ and $\mathbf{r}_0$ with the suspect points removed and, if necessary, to check them again, against the revised location and orientation of the object model. When the position of the plug was recomputed with just four of the data points from View 1, the maximum discrepancy in the elements of $\mathbf{R}$ was about 0.02, the majority were much smaller, and the maximum discrepancy in the components of $\mathbf{r}_0$ was 0.01.

CONCLUDING REMARKS

It has been established in this paper that the hypothesis, prediction verification paradigm that is widely used in A.I. can be applied in SIMD parallel processing mode to the problem of object recognition, within a very realistic time scale.

Given object model descriptions stored in a database, together with a small number of feasible interpretations, in which sensory data points have been provisionally assigned to the faces of a given object model on the basis of simple geometric comparisons, we have shown how to establish the location and orientation of the object model that is most consistent with the data. We have also shown how to confirm that every data point is visible and lies sufficiently close to the face to which it has been assigned.

It may thus be established which of the interpretations is most closely consistent with the data, and the location and orientation of the object corresponding to the best interpretation is then available for input to a control system.

The overall timescale for interpretation and validation, with obscured data points, back face data and errors in the spatial coordinates of data points correctly identified, is about 90 milliseconds for the electric that is used as an exemplar.

We have been concerned herein with the static recognition problem, with data derived from a snapshot in time, but in principle the results might be applied to the dynamic situation in which previous interpretations can be used to narrow down the search for feasible interpretations, and possibly speed up the validation process, with prior knowledge of the location and orientation of certain objects in the scene gained from previous runs of the recognition system.

In this case, failure to arrive at a definite interpretation, at the outset, is not likely to be a serious problem, for we may distinguish between exploratory mode, in which we are simply gathering data with potential feedback to the sensor of requests for further information, and active mode when a valid interpretation is required by the control system.

In the meantime, work is proceeding on the interpretation of edge matching data, and the subsequent validation of interpretations, to meet the demands of interfacing with the parallel version of the ISOR system that was mentioned in the introduction to this paper.

REFERENCES

1. **Murray D.W. and Cook D.B.** "Using the Orientation of Fragmentary 3D Edge Segments for Polyhedral Object Recognition" *International Journal of Computer Vision* Vol 2 (1988) pp 147 - 163.
2. **Grimson W.E.L. and Lozano-Pérez T.** "Model-Based Recognition and Localisation from Sparse Range or Tactile Data" *International Journal of Robotics Research* Vol 3 (1984) pp 3-35.
3. **Faugeras O.D., Ayache N. and Faverjon B.** "A Geometric Matcher for Recognising and Positioning 3-D Rigid Objects" *Proc. A.I. Applications Conference* IEEE (1984) pp 218 - 224.
4. **Flynn A.M. and Harris J.G.** "Recognition Algorithms for the Connection Machine" *Proceedings of the Ninth International Joint Conference on Artificial Intelligence,* Morgan and Kaufman (1985) pp 57 - 60.
5. **Holder D.R. and Buxton H.** "Polyhedral Object Recognition with Sparse Data in SIMD Processing Mode" *Proceedings of the Fourth Alvey Vision Conference* (September 1988) pp 103-110, also published in *Image and Vision Computing* (February 1989).
6. **Castelow D.A., Murray D.W., Scott G.L. and Buxton B.F.** "Matching Canny Edgels to Compute the Principal Components of Optic Flow" *Proceedings of the Third Alvey Vision Conference* (September, 1987) pp 193-200.
7. **Murray D.W., Castelow D.A., and Buxton B.F.** "From an Image Sequence to a Recognised Polyhedral Model" *Proceedings of the Third Alvey Vision Conference* (September, 1987) pp 201-210.
8. **Buxton H. and Wysocki J.** "The SIMD Parallel Image Sequence Object Recognition (SPISOR) System" (in draft).
9. **Buxton B.F.** *Personal communication* (February, 1989).