SIMD GEOMETRIC MATCHING

Derrick Holder
City of London Polytechnic, Department of C.M.S.M.S.,
100 Minories, London EC3N 1JY.
and
Hilary Buxton
Queen Mary Westfield College, Department of Computer Science,
Mile End Road, London E1 4NS.

INTRODUCTION

It is generally agreed that the primary task for any true machine vision system, when confronted with geometric data (i.e. location, orientation and hence shape) from low level vision sources, is to identify what is where within the field of view. One approach, that has been adopted by Grimson and Lozano-Pèrez[1], Faugeras, Ayache and Faverjon[2], Murray and Cook[3] and others, is to consider objects in the form of separate, possibly non-convex, polyhedra, for which there are accurate geometric models. First they generate feasible interpretations by means of simple, generally pairwise, geometric comparisons between object models and sensor data. Then they test the interpretations, in detail, for compatibility with the surface equations of a particular object model, bearing in mind the fact that an object may have up to six degrees of freedom relative to the robot's sensors. The method is thus based on the hypothesis, prediction and verification paradigm that is widely used in Artificial Intelligence.

Numerous sequential algorithms have been implemented for the generation of feasible interpretations. Measurements involving location vectors and surface normals at m data points, considered in pairs, are compared with corresponding values that are associated with n×n pairs of object model faces. It is found that, when simple geometric constraints that are independent of the coordinate frame of reference are applied to sparse data, the possible asignments of data points to object model faces can generally be represented by just a few, frequently only one, feasible interpretation. This is done entirely without resort to a detailed solution of the surface equations. Even so, the algorithms are not generally fast enough, in sequential form, to offer a practical solution to the problem.

A parallel algorithm for the generation of feasible interpretations has been implemented by Flynn and Harris[4] on the Connection Machine at the Massachusetts Institute of Technology. This algorithm exploits the parallelism in the problem at the expense of processor numbers which grow exponentially with problem size. However a similar degree of parallelism has been achieved by the present authors[5,6] with a processor set that is only quadratic in the problem size. Using a distributed array SIMD processor, the AMT DAP 510, problems are handled that would previously have far outstripped the capacity of the Connection Machine. The algorithm can equally well be applied to measurements relating to the edges of a polyhedron. 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. Nevertheless, this form of input is efficiently provided by the ISOR system[7,8,9] developed at GEC Hirst Research Centre, and currently being implemented on the AMT DAP at Queen Mary Westfield College.

The purpose of this paper is to present an overview of the algorithms for both face matching and edge matching interpretations of visual data, together with some of the results that have been achieved to date.

THE GENERATION OF FEASIBLE INTERPRETATIONS

The generation of feasible interpretations in the face matching problem proceeds as follows:-

(i) For each pair of data points, trial assignments to the faces of a particular object model are recorded in an interpretation tree, with each node representing a given assignment, and with alternative paths representing the sequences of assignments embodied in different interpretations of the data set.

(ii) A geometric match is said to be achieved when the values of certain primatives, such as the distance between two points or the angle between two surface normals, associated with a given pair of data points, are compatible with the ranges of values associated with the object model faces to which they have been assigned. The interpretation tree is pruned, i.e. the path representing a given interpretation is terminated, wherever there is a failure to achieve a geometric match.

(iii) Finally the interpretation tree is pruned wherever a trial assignment would be inconsistent with assignments already made at higher levels in the interpretation tree.

By far the most important single step in the quest for parallelism is to note that pairwise geometric matching is totally independent of the preceding partial interpretations and can be implemented as a parallel process, leaving the global consistency of interpretations to be taken into account at a later stage. We note that the sub-trees from the nodes at a particular level in the interpretation tree are all the same until they are pruned for consistency, and will be reproduced many times over. The results of the geometric matching process may therefore be best represented by a network rather than a tree structure, and stored compactly in an array such as is illustrated in Figure 1, where all paths downwards through true values have to be explored.

	face$_1$	1	2	3	4	5
datum$_1$	datum$_2$ / face$_2$	12345	12345	12345	12345	12345
1	2	T·T··	·····	·····	···T·	···T·
1	3	·····	···T·	·····	·····	·T···
1	4	·····	·····	·····	·····	T····
2	3	···T·	·····	··T··	·T···	·····
2	4	·····	·T···	·····	T····	·····
3	4	·····	T····	·····	·····	·····

Figure 1 *The Matching Array*

In fact pairs of data points may be considered in any order, and we note that there is only one feasible assignment of data points 1 and 4, in the hypothetical example above, namely to object model faces 5 and 1, respectively, so this is obviously a good place to start. We can generally avoid a proliferation in the number of alternatives to be considered at a given level in the interpretation, by sorting the data pairs into ascending order of geometric match, and a simple tag sort procedure using standard functions in DAP FORTRAN may be used for this purpose.

Although at any stage the check for consistency is dependant on the preceding partial interpretation, it can be performed as a parallel process within a recursive procedure, and the subsequent assignments of data points to object model faces can be made conditional on the outcome. The conditional processing within the loop is of a sequential nature, but this seems inevitable if the demands on processing elements are to be kept within reasonable bounds. Nevertheless, highly effective pruning of alternative interpretations is thus achieved, because the matrix of consistent matches at a given level in the interpretation is generally very sparse, and the selection of true values is efficiently implemented in DAP FORTRAN.

In the case of edge matching, sensory data expressed in terms of position vectors and edge direction vectors are assigned to particular edges of an object model, but the object model database and the method of generating feasible interpretations are essentially the same.

VALIDATION

Having generated an interpretation in which sensory data have been provisionally assigned to particular faces of a given polyhedral object model, on the basis of simple geometric constraints, there is no guarantee that the object model description will be entirely consistent with the data.

The validation process involves the following three steps:-

(i) establishing the location and orientation of the object model that is most compatible with the data;
(ii) confirming that every data point then lies sufficiently close to and within the perimeter of the object model face to which it has been assigned;
(iii) confirming that every data point is visible in the given interpretation.

Now, a rigid body rotation and translation may be expressed in terms of a 3x3 orthogonal rotation matrix $\mathbf{R}$, and a translation vector $\mathbf{r}_0$, and we may determine $\mathbf{R}$ and $\mathbf{r}_0$ in such a way that first the object model surface normals after rotation, and then the perpendicular distances from the origin after translation, match the data as closely as possible. The orthogonality condition $\mathbf{R}^T\mathbf{R} = \mathbf{I}$ imposes 6 non-linear constraints on the elements of $\mathbf{R}$, and a further three equations are obtained when the method of constrained least squares is applied to the residual differences between normal directions. The solution for $\mathbf{r}_0$ is obtained more easily, with the method of least squares applied to the residual differences in perpendicular distance from the origin.

It has been demonstrated[10] that the solution of the equations for $\mathbf{R}$ 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, with a good first approximation obtained from the relationship that applies when the data exactly fit the object model The process converges to sufficient accuracy after just one or two iterations. Faugeras, Ayache and Faverjon, work rather more compactly with quaternions to determine $\mathbf{R}$ and $\mathbf{r}_0$, achieving what appears to be an equivalent result, but presumably their algorithm is implemented in sequential form.

Having established the appropriate location and orientation of the given object model, 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. 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. 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. We note that the equations for intersections take a particularly simple form when the external point is located at an infinite distance along the positive x-axis.

The first task in an SIMD implementation of the validation process is to map the object model against the data and to set the unused rows of the mapped object model 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 FORTRAN Library subroutines, and Standard 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 their application to the object model.

Before proceeding with the validation of individual data points, the Cartesian 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 mxm parallelism is thus achieved.The organisation of the information within DAP matrices at this stage is illustrated in Figure 2.

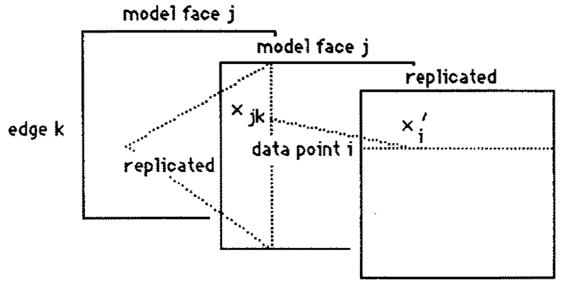

Figure 2. *The Organisation of Information within DAP Matrices*

Transforming into viewing coordinates and initialising a DAP logical matrix, we set up a logical matrix **inside**, and proceed to investigate intersections of the line joining each data point to the given external point with the edges of every face, successive edges of each face being considered in turn face. We switch an element of **inside** between TRUE and FALSE whenever an intersection occurs, and it is thus rapidly established which data points fall inside which faces when these are all projected onto the viewing plane. The perpendicular distances from data points to the object model faces, and the backface condition , are determined from straightforward parallel calculations, and a standard function collates results within a given row. In this way, 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.

Essentially the same algorithms apply to the edge matching problem, with perpendicular vectors from the origin to observed edge segments used in computing $\mathbf{r}_0$, but it then has to be established that every data point is sufficiently close to the edge segment to which it has been assigned.

TEST RESULTS

The method works well with synthetic data related to simple object models, and a representation of a three-pin electric plug, similar to that used by Murray and Cook[3], has been adopted with a view to further performance tests. The plug is viewed from three different positions, with data points at the centre of each visible face. The first view, looking towards the face of the plug, has 14 visible faces and 91 pairwise comparisons are involved in the generation of feasible interpretations. The second view, looking towards the back, has 12 visible faces requiring 66 comparisons, whereas the third view, looking directly down on the pins, has only 4 faces that are clearly visible involving only 6 comparisons. The three views are illustrated in Figure 3.

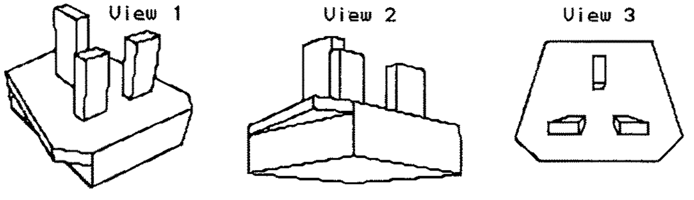

Figure 3 *The Three Views of the Electric Plug*

There are more that 10^{20} possible interpretations to be considered with regard to View 1, and the method of Flynn and Harris would require either a separate processor for each one or totally unacceptable segmentation of the problem. In the meantime, the present method allows the problem to be accomodated easily on a 32x32 DAP, and the process converges quickly to a single interpretation for View 1 and View 2, with run times of 65ms and 37ms, respectively. However, not too surprisingly, the process fails to distinguish between the ends of the two short pins in View 3.

Three additional back face points were included with the View 1 test data for the purpose of testing the validation process, and we note that the point at the centre of the visible side of the plug was in fact obscured by the flange. Four back face data points were included with the data for View 2, two points in faces of the earth pin were obscured by the neutral pin, and the one in the underside of the flange on the far side of the plug was obscured by the rest of the plug. For View 3, three data points were in back faces but none were obscured otherwise.The location and orientation of the plug were determined, in each case, within about 26.5 milliseconds, and the back faces and obscured data points were identified by the validation process in a further 12 milliseconds.

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 $\mathbf{R}$.

CONCLUDING REMARKS

Interpretation and validation, with obscurred data points and simulated errors, is achieved in about 90 milliseconds, for the given exemplar. Work is continuing with regard to the interpretation of real, as opposed to synthetic, edge matching data, and the subsequent validation of interpretations, to meet the demands of interfacing with the parallel version of the ISOR system.

REFERENCES

1. Grimson W.E.L. and Lozano-Pèrez T. "Model-Based Recognition and Localisation from Sparse Range or Tactile Data". *International Journal of Robotics Research,* Vol 3 (1984) pp 3 -35.
2. 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.
3. Murray D.W. and Cook D.B. "Using the Orientation of Fragmentary 3D Edge Segments for Polyhedral Object Recognition". *International Journal Computer Vision,* Vol 2 (1988) pp 147 - 163.
4. Flynn A.M. and Harris J.G. "Recognition Algorithms for the Connection Machine". *Proc. Ninth International Joint Conference on A. I.,* Morgan and Kaufman (1985) pp 57 - 60.
5. Holder D.R. and Buxton H. "Polyhedral Object Recognition with Sparse Data in SIMD Processing Mode". *Proc. Fourth Alvey Vision Conference,* (September 1988) pp 103 - 110, also published in *Image and Vision Computing* (February 1989).
6. Holder D.R. and Buxton H. "Polyhedral Object Recognition with Sparse Data - Validation of Interpretations". *Proc. Fifth Alvey Vision Conference* (September 1989).pp 19 - 24
7. Castelow D.A., Murray D.W., Scott G.L. and Buxton B.F. "Matching Canny Edgels to Compute the Principal Components of Optic Flow". *Proc. Third Alvey Vision Conference* (September, 1987) pp 193-200.
8. Murray D.W., Castelow D.A., and Buxton B.F. "From an Image Sequence to a Recognised Polyhedral Model". *Proc. Third Alvey Vision Conference* (September, 1987) pp 201 - 210.
9. Buxton H. and Wysocki J. "The SIMD Parallel Image Sequence Object Recognition (SPISOR) System" (In draft).
10. Buxton B.F. *Personal communication* (February, 1989).