*J. Indian Inst. Sci.*, Jan.–Feb. 1996, **76**, 93–107.


# Reverse engineering of solid models

B. GURUMOORTHY
Department of Mechanical Engineering, Indian Institute of Science, Bangalore 560 012, India.

Received on October 15, 1995.

**Abstract**

This paper presents an overview of the work done in the Robotics and CAD Laboratory on developing tools for automatic construction of CAD models from existing information. These include engineering drawings and point data measured off the physical prototype. Some of the tools developed are of value not only in CAD as a flexible and efficient means of constructing 3D computer models of existing parts, but also in product development especially for outsourcing of tooling and components, automatic inspection and verification in manufacturing, and model-based vision.

**Keywords:** Reverse engineering, model reconstruction, legacy data, solid models, CAD.

## 1. Construction of solid models—motivation

The solid model of a part has become an essential requirement for the application of computer-based tools that are available for automating various tasks in the design–manufacture cycle. The conventional method of constructing a solid model is to use any of the CAD systems that are currently available. Typically in these systems the user is provided with a set of tools that allow construction of curves/surfaces that can then be used to form solids through operations such as sweep and extrusion. These solids can then be combined with other solid primitives that are available in the system to construct complex solid models. The user interaction is restricted to picking the primitives (curves, surfaces, solids) and sizing/locating these as necessary and choosing the appropriate combinations. This process is supportive of *ab-initio* design.

In cases where models have to be constructed for existing parts, the above process involves an additional effort on the part of the user in terms of interpreting the existing representation of the part geometry and then translating it to the appropriate commands/inputs to the modelling system; making the speed and accuracy of the construction dependent on the user's skill. For existing parts, therefore, a more direct method is desirable. This problem is also referred to as the reverse engineering problem because the starting point is the end product, and steps are retraced to obtain the starting model. With specific reference to mechanical design, it is the production of the solid model (initial product) from the manufactured part (end product).

There are two broad approaches possible for reverse engineering. In situations where the drawings of the part are available, these could be scanned and the resulting data could then be processed to generate solid models. The other approach would be to take measurements off the physical prototype and process them to construct the solid model.

## 2. Construction from engineering drawings

Even as solid modelling systems and CAD systems based on creating and manipulating solid models are finding increasing acceptance and use, it is found that in most industries, the part representation is predominantly in the form of blueprints containing the orthographic views of the part (these representations are referred to as legacy data in the literature). Also, it is believed that there are more than an order of magnitude number of computer-aided drafting systems than solid modelling systems in use[1].

In order to use the computer-based tools for the tasks in product development such as FEM, NC Path generation, it is necessary to integrate the drafting systems or engineering drawings with the solid model-based CAD systems. This requires conversion of the 2D representations (as in paper drawings or drafting models) to 3D solid models.

While the rules for forming the projection of a solid are well defined, the reverse procedure is not straightforward as the projective transformation is not invertible. In the projections, collinear vertices may project as a single vertex and many coplanar edges may form a single edge in the projected view. The complexity increases while dealing with objects containing curves and space curves in particular. The exact profile of the space curves is not obtained from their orthographic projections. Thus the exact number of edges or vertices in the solid cannot be inferred from a given projection.

The process of reverse engineering solid models from 2D projections of a part is as follows. In cases where this information is available in the form of engineering drawings, scanning the drawing would yield pixel data that have to be processed to obtain vector data. This is not a trivial task. Drawings contain both text and graphics. These must be separated. Even within the graphical part of the scanned data, distinction has to be made between the part that conveys the shape information and that which is associated with annotations. For instance, a line can be part of the view or a dimension attatched to the view. At the end of this conversion (also referred to as layer separation) we would have a set of 2D wireframes.

A computer-aided drafting system would output the 2D wireframe directly (though this may require some data conversion). Our problem then is of interpreting these 2D wireframes (vector/line data) to realize the solid model. Our work assumes that the input views are available as vector data (true of computer-generated engineering drawings) thereby side-stepping the problems of errors and uncertainties introduced by the process of digitizing and vectorization of paper drawings.

Two approaches have been used to construct the solid model from engineering drawings. The first approach is based on the more popular 'bottom-up approach'[2] and the second[3,4] incrementally constructs the solid model using elementary solids identified from the views.

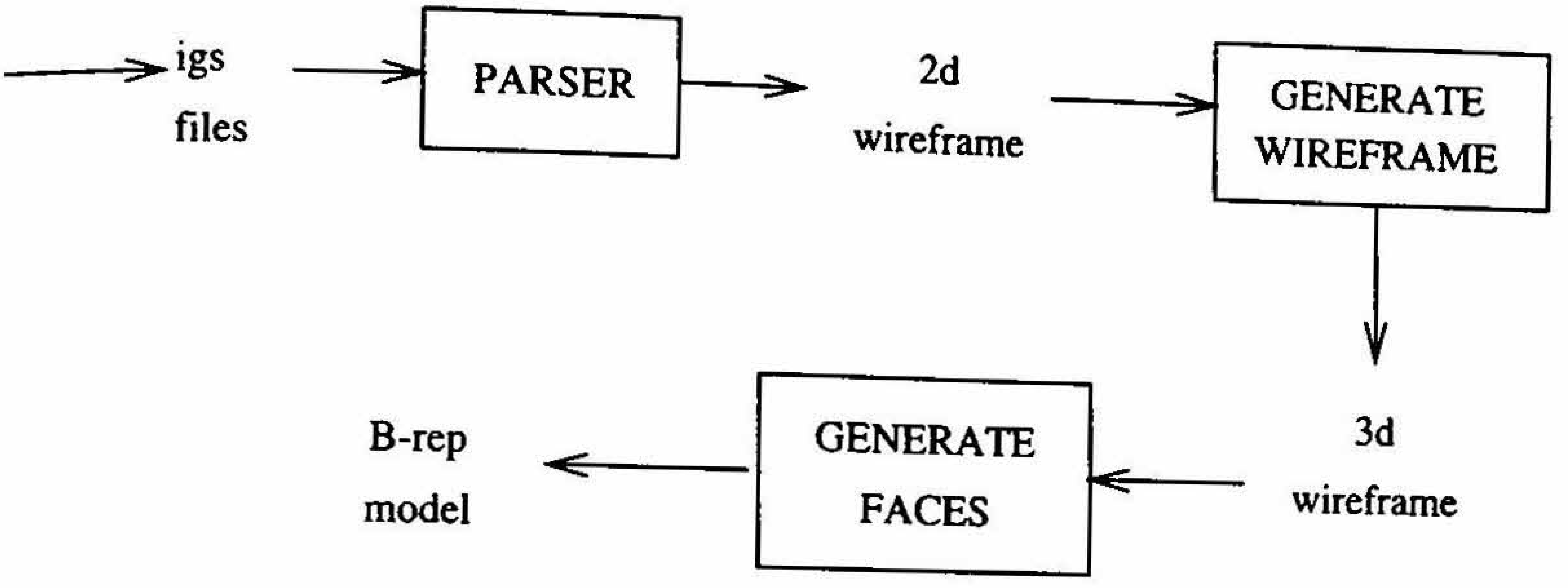


FIG. 1. Bottom-up construction of solid models from three views.

## 2.1. *Bottom-up construction*

In this approach[2, 5, 6] it is assumed that the representation available contains all three orthographic views of the object. These are assumed to be available in the form of IGES files corresponding to each of the three views of the object, namely, elevation, side and plan. The output is the boundary representation of the part. Figure 1 shows the schematic of the steps involved in the construction.

The first task is to realise the 2D wireframes by parsing the given IGES files.The 3D vertices in the object are then obtained. From these the 3D edges are generated. Arranging these in a proper order gives the 3D wireframe. The face information is then generated from the wireframe. Moebius rule is then implemented to arrange the faces in order to get the boundary representation[2].

3D vertices of the object are obtained by searching for a given vertex in any view, its likely projection in the other two views. The vertex coordinates are then obtained by picking the appropriate coordinate of the matching 2D vertices. The 3D wireframe is obtained by using a generate and test procedure. Potential edges are generated based on the 2D connectivity in a view and tested for validity against the connectivity information in the other views. A variation of Markowsky and Wesley's algorithm[7] has been developed to obtain the solid model from the 3D wireframe[2, 6, 8]. This algorithm uses a connectivity matrix derived from the 3D wireframe to identify planes/surfaces that contain the faces of the object. Vertices lying on each of these planes/surfaces are then sorted to form closed loops. These loops are then classified with respect to each other to identify the faces and the inner loops. The outward normals for the faces are assigned by implementing the Moebius rule, thus constructing the boundary representation (BRep) of the part. A typical object constructed by this algorithm is shown in Fig. 2. The main limitation of this approach is that it cannot handle sectional views. This is because all edges and loops in the sectional view may not correspond to edges and faces in the solid. The *generate and test* strategy to obtain the 3D wireframe will therefore fail. Also this approach requires all three orthographic views to be available for the part. This is rarely the case in actual practice. In most situations, a component is represented by one or two

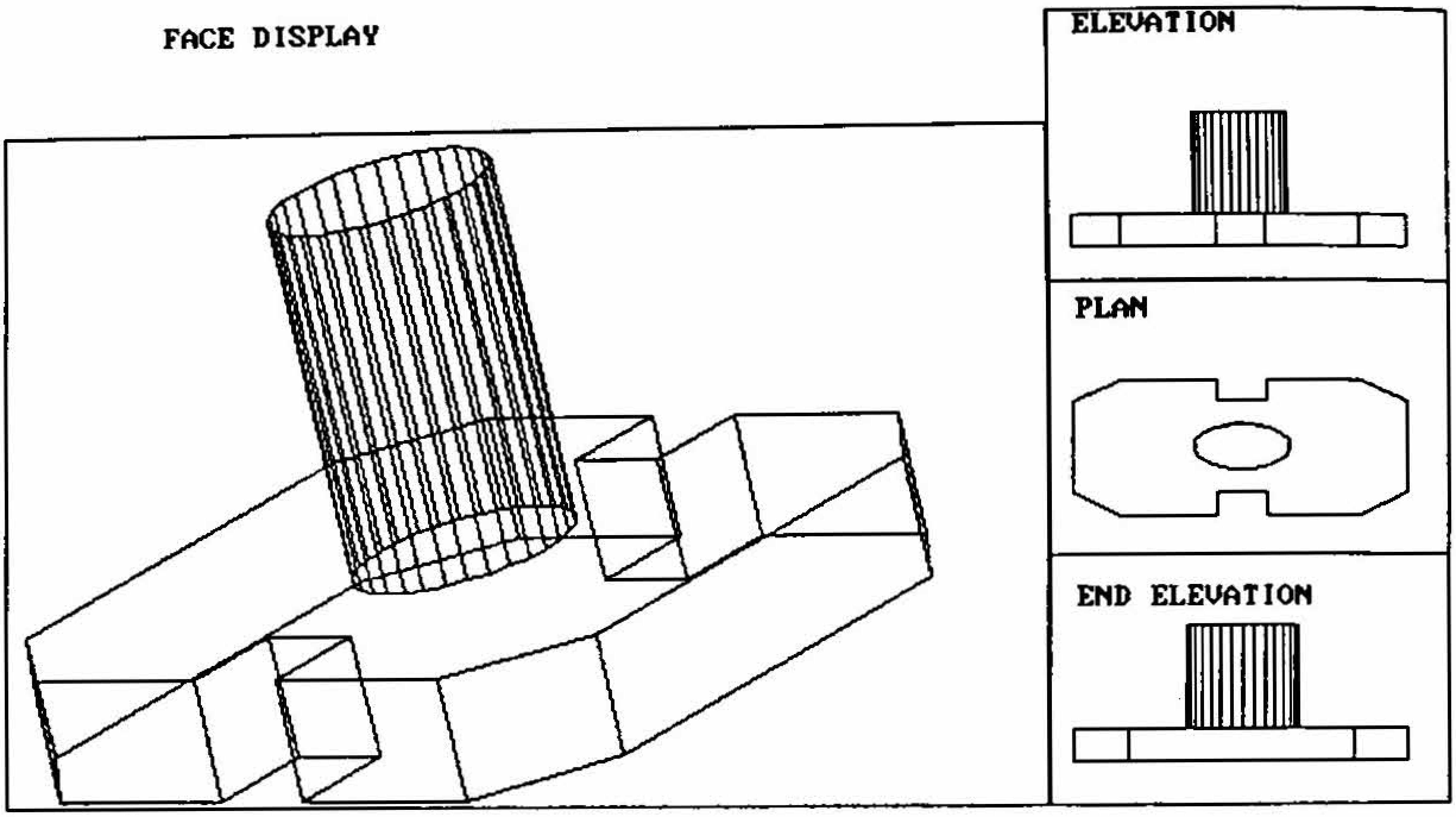


FIG. 2. Example solid constructed from three views.

orthographic views and several sectional views to highlight hidden details. Some work has been reported in the literature[9, 10] that constructs model from two views using variations of this approach but they need the third view for checking completeness. A different approach had therefore to be used to be able to handle sectional and auxiliary views.

## 2.2. *Incremental approach*

The underlying idea of this approach is as follows[4,11]: Given the 2D wireframes describing each view, the solid model is constructed incrementally by adding or removing elementary solids from a base solid. The base and the elementary solids are swept solids identified from edge loops in the views. The two main issues are:

- Identification of loops that can form an elementary solid.
- Finding the operation to combine the base and the elementary solids.

The main attraction of this approach is that it works with a pair of views at a time. Figure 3 illustrates the schematic of the approach. Figure 5 demonstrates the working of the algorithm for an example object, given two orthographic views as shown in Fig. 4. The algorithm works by first identifying the outer loop in a view and splitting the view into quasi-disjoint loops (loops that intersect only at their boundaries). Splitting is required only if there are vertices with more than two edges incident on them in a view. The span of sweep for a loop is determined from the matching vertices of the loop in the second view. The choice of the boolean operation to be performed on the swept solids is based on the local behaviour of the edges in a loop at vertices with more than two edges in the view incident on them. The algorithm deals with a pair of views at a time. After a solid has been obtained from a pair of views, its projection in the other direction(s) can be

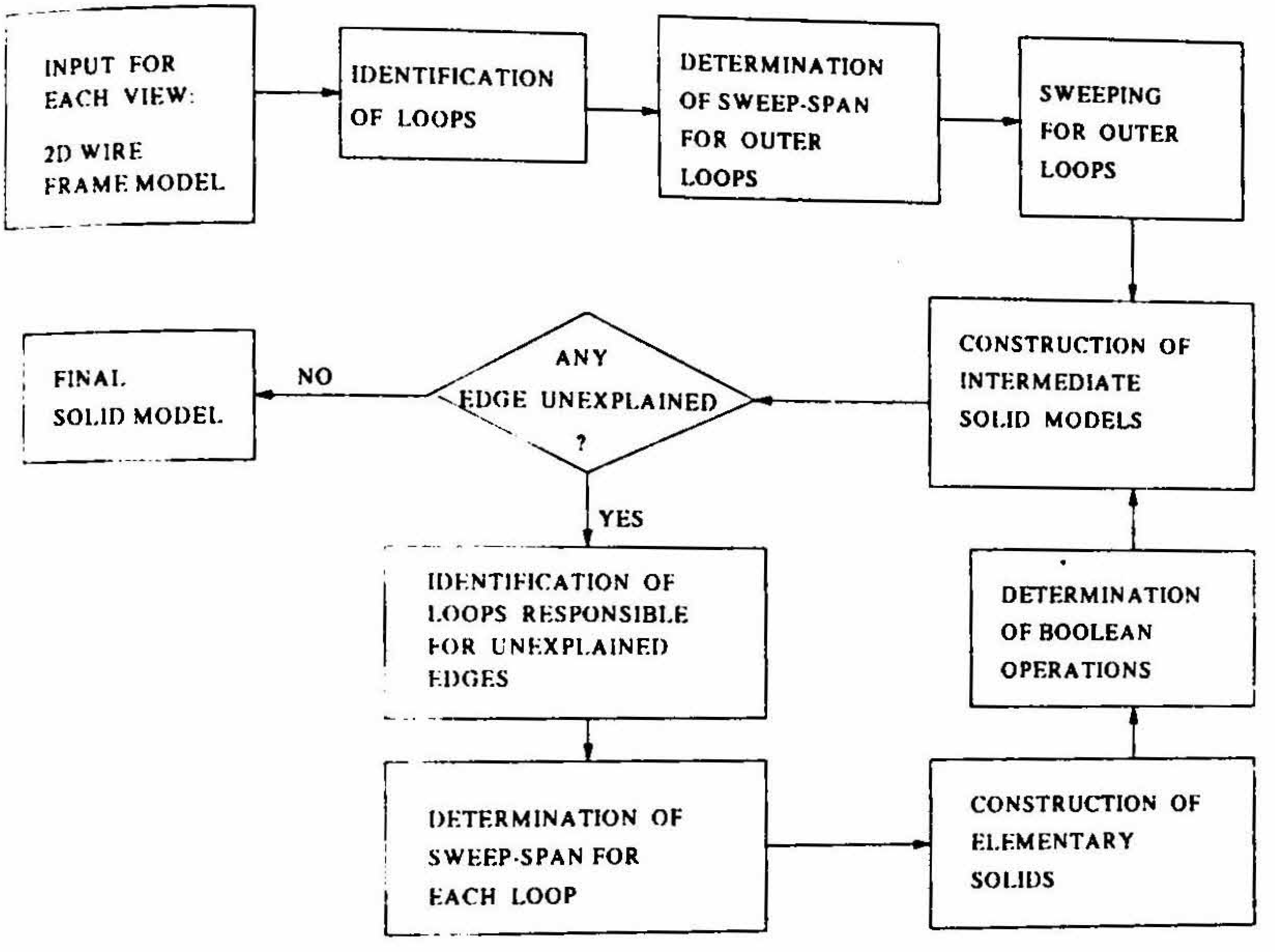


FIG. 3. Incremental approach to construct solid models from two views.

compared with given view(s) to identify the loops that have not been accounted for. It has to be investigated if solids corresponding to new loops cause inconsistencies in the views already considered. A rendered image of the solid model constructed is shown in Fig. 6.

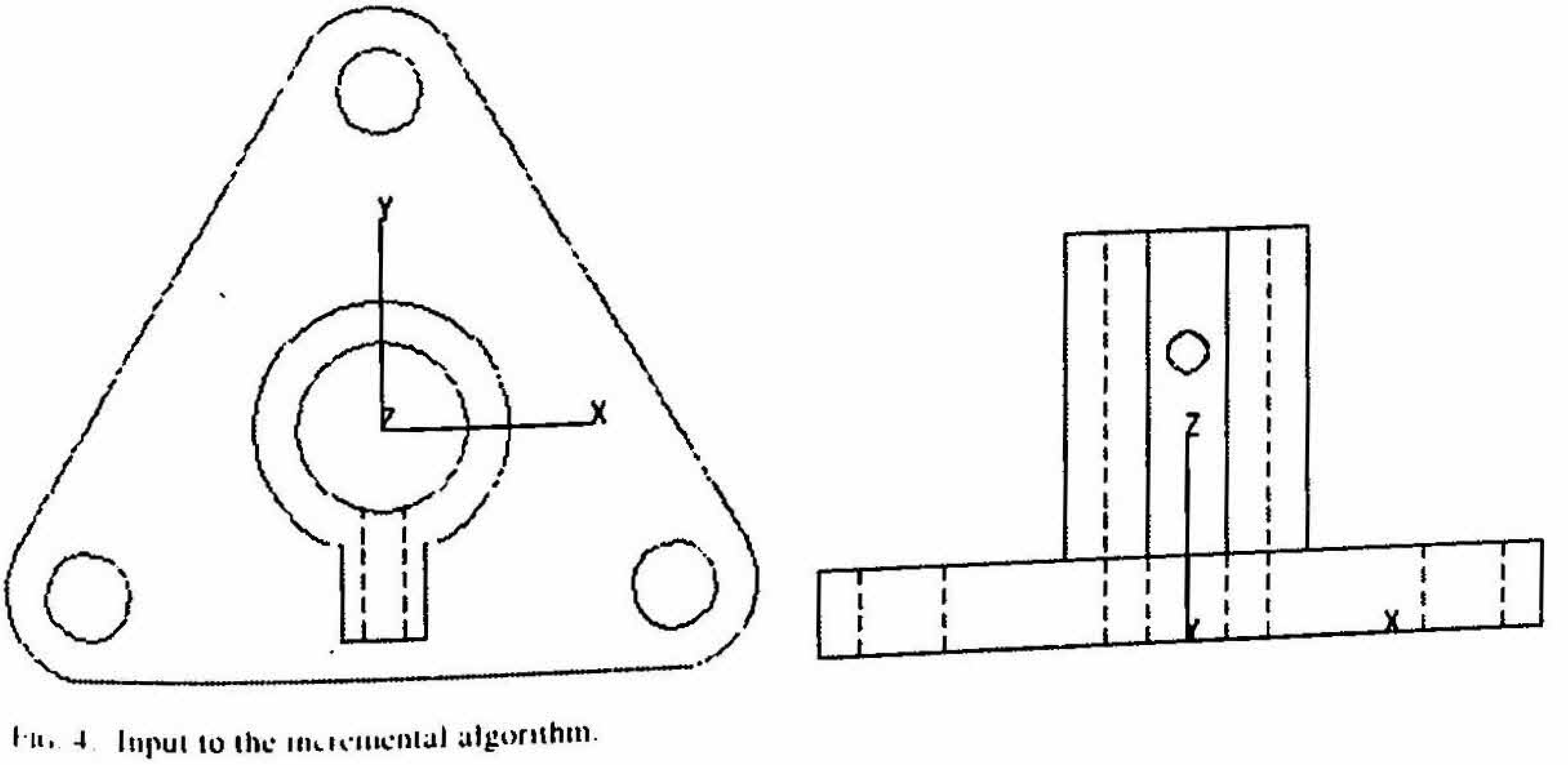


FIG. 4. Input to the incremental algorithm.

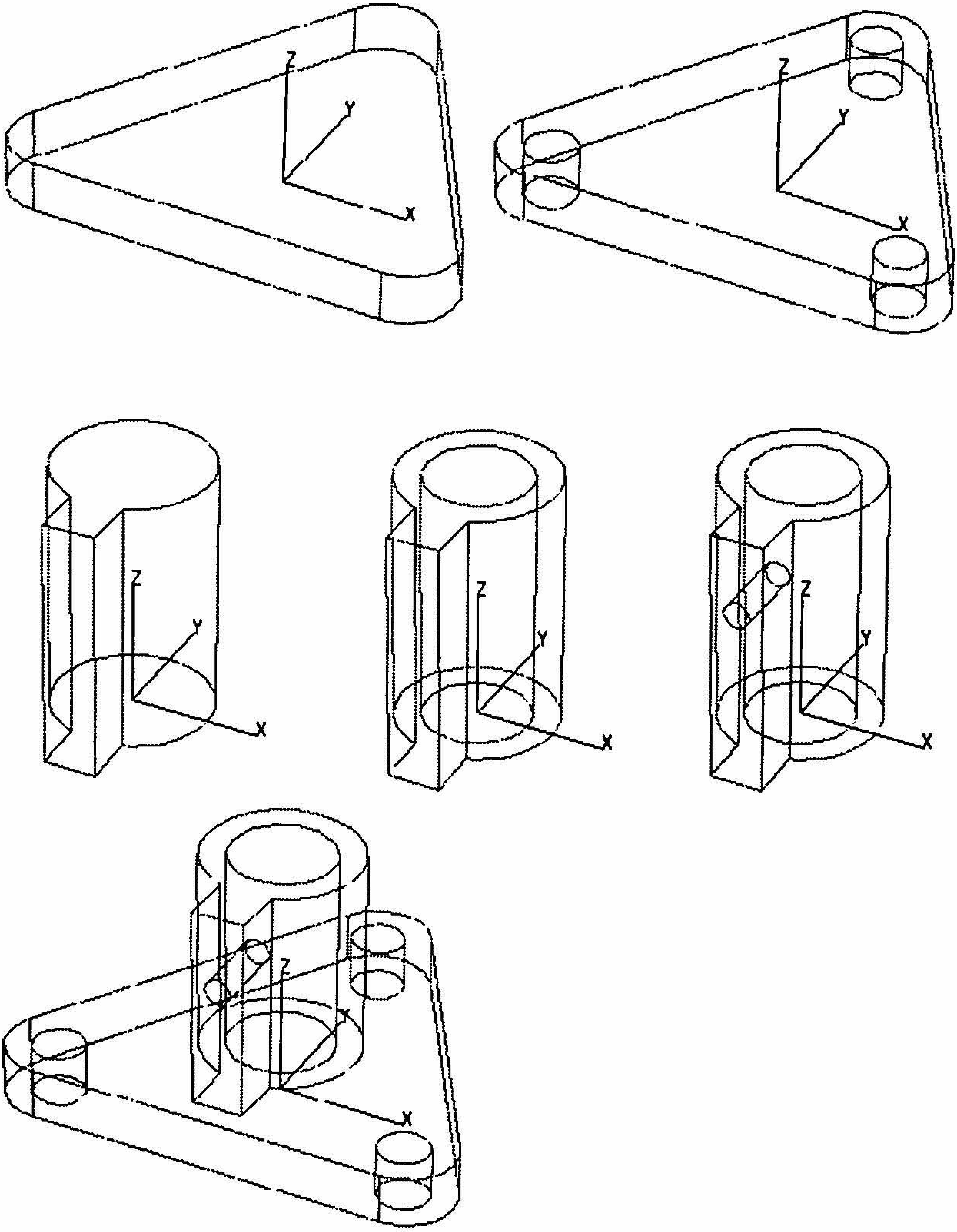


FIG. 5. Construction of solid model by the incremental algorithm.

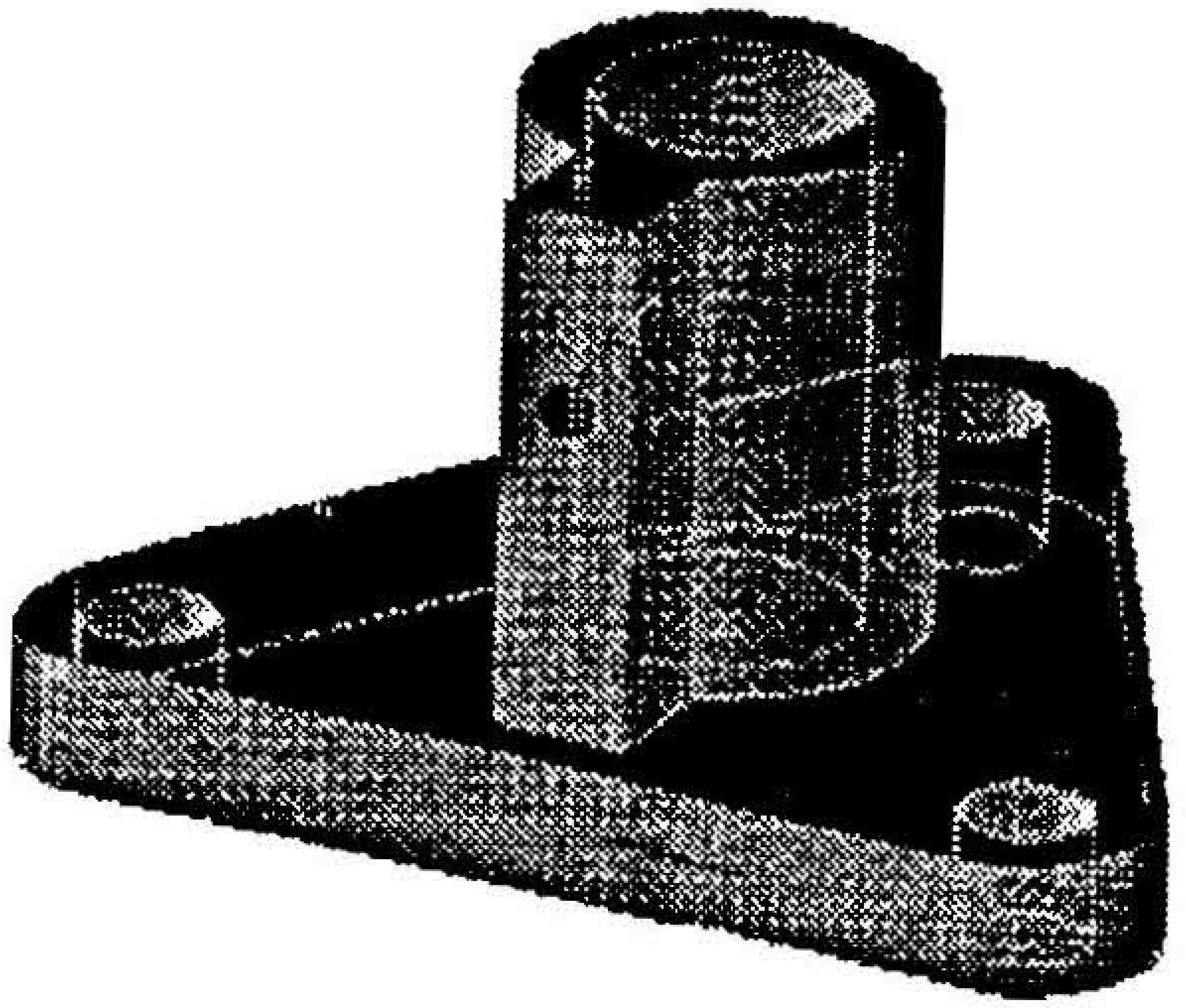

FIG. 6. Rendered image of the solid model constructed by incremental algorithm.

Unlike the approach proposed by Aldefeld and Richter[12] the elementary solids in this algorithm are not predefined, but are determined from the information in the drawings. The construction process therefore is not limited by the library of elementary objects available, thereby making the reconstruction process generative.

Since the algorithm works by identifying (and appropriately combining) solids that correspond to loops in any view, it is believed that this algorithm can handle sectional views and auxiliary views. If the view is an auxiliary view (plane of projection is not one of the orthographic planes) or a sectional view in an auxiliary direction, then determining the matching vertices requires further development. This is because one reason for using sections or auxiliary views is to avoid having dotted lines in the other views. The absence of dotted lines would render the task of finding the matching vertices, and thereby the span of sweep of loops to form the elementary solids, difficult.

Another extension required to enhance the capabilities is to allow elementary solids to be formed by sweep other than extrusion. The additional types of sweep should include rotation about an axis, sweep along a curved path and sweep with varying crosssection. These would allow handling objects with non-iso oriented edges and 3D curved entities.

While computer-generated drawings will output clean 2D wireframes, it is unrealistic to expect the same from digitization of paper drawings. In order to be able to handle information from paper drawings, it is therefore necessary to examine the effect of uncertainties/noise in the 2D geometry on the algorithm.

## 3. Construction from physical prototypes

The problem here is to construct a representation of a part from point data obtained from a physical prototype. The objective of construction from measured data sets apart the work done in the Robotics and CAD Laboratory[13, 14] from efforts ongoing elsewhere. The

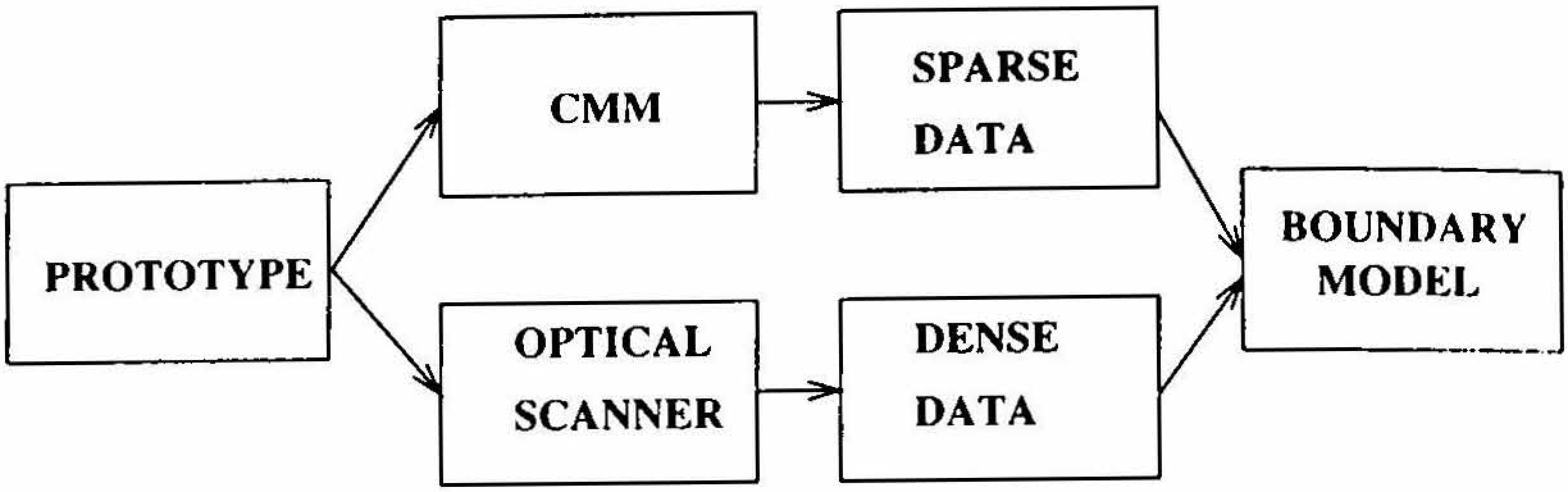


FIG. 7. Construction of solid models from physical prototypes.

main aim of construction of models (as reported in literature)[15, 16] appears to be visualization. Consequently, a tessellated model is found to be acceptable for representing the solid. In contrast, our goal has been to construct models that can be used with the CAD/CAM systems currently in use.

Construction of solid models from physical prototypes is of great significance in the situation where the part is available but a computer model of the part is not available. This is particularly the case where vendors/collaborators do not provide design details. Reverse engineering of solid models from physical prototypes is also useful in updating the part model with modifications made in the prototyping/manufacture stages. Solid model constructed from measured data can also be used in model-based vision and automatic inspection (for comparing the manufactured part with the original solid model). Also, starting from the physical object is the only option available in the design of prosthetic implants.

There are two options possible in measuring points on the part (Fig. 7). These could be generated by a touch probe mounted on a robot/CMM or the data could be generated by means of an optical scanner or digitizer. The former method results in *sparse* data while the data obtained by scanning or digitizing is referred to as *dense* data.

Work on this problem in our Laboratory originated with construction from structured point data along with neighbourhood information[13]. The requirement of neighbourhood information was relaxed next[14] and currently work is in progress on construction from unstructured point data.

The problem of obtaining mathematical representation of surfaces given point data is encountered in the construction of solid models from point data. Reverse engineering of surface models is a research area in itself and some work[17, 18] on this has been done in our Laboratory. In the following, construction of surface models is described first, followed by the work on construction of solid models.

### 3.1. *Reverse engineering of surface models*

Obtaining a surface model from a cloud of points involves finding a best fit surface. This problem in interpolation is solved as a nonlinear optimization problem as the desired form of the interpolating surface is NURBS (polynomials of order 3 and above). The error of interpolation is sought to be minimized in the optimization. The approach developed[17] departs from those reported in literature in two significant ways. The parametric values are assigned independently and not averaged over the row/column. This helps to localize changes during error optimization. The measure used for the interpolation error is the shortest distance between the input points and the surface and not the distance between the input point and its corresponding point as used in literature. This, it is believed, is more intuitive physically and has been shown to result in a better interpolation[17]. This measure reflects the actual requirement of the interpolation more closely than the distance between the input point and its corresponding point on the surface, which is an artifact introduced by the user.

Another problem that is encountered in constructing surface models from a cloud of points is that of fitting multiple surface patches to the set of points while maintaining desired continuity between these patches. Methods have been developed that first interpolate the points with multiple patches and then alter these patches to improve the continuity without altering the interpolation performance[18]. Further work is required to validate this approach.

### 3.2. *Construction of solid model using neighbourhood information*

Points measured on each face along with adjacency information for each face are the inputs in this approach. It must be mentioned here that even if the faces are adjacent along more than one edge, the adjacency information between the pair of faces needs to be input only once. It is therefore believed that this additional input can be provided by the operator while collecting the point data. The sets of points measured on each face of the part are used to determine the best fit plane/surface. The orientation of the normal of the surface is derived using the origin of the measuring device.

The main problem, when the neighbourhood information is given, is to find chains from graphs that represent the neighbourhood information. The NP algorithm[13] works on adjacency graphs where all nodes (faces) have degree exactly equal to 2 (all faces share exactly one edge with other faces) and finds the correct chain. The algorithm for general solids (NP and SM algorithms[13]) works by decomposing the input adjacency graph to subgraphs where NP algorithm will work and then merging them suitably. The decomposition is done at nodes (faces) that have degree $d_j > 2$ and the NP algorithm generates chains that terminate at such nodes. The merging process works by either searching for matching head and tail of the chains identified or identifying vertices that have to be introduced to be able to merge these chains corresponding to these nodes (faces) and face *i*. As these vertex chains are ordered (to maintain correct orientation of the normal of face *i*), it is sufficient to merge the tail of a chain with an appropriate head of another chain. When there are only two chains with the same parent node, the choice of appropriate chain is obvious and unique because the vertices in the chain are ordered.

Figure 8 shows an object that has been constructed by the NP and SM algorithms. While this is an improvement over earlier work[19] in that the existence of a CAD model was not required, it nevertheless does not work on measured data alone (the ultimate

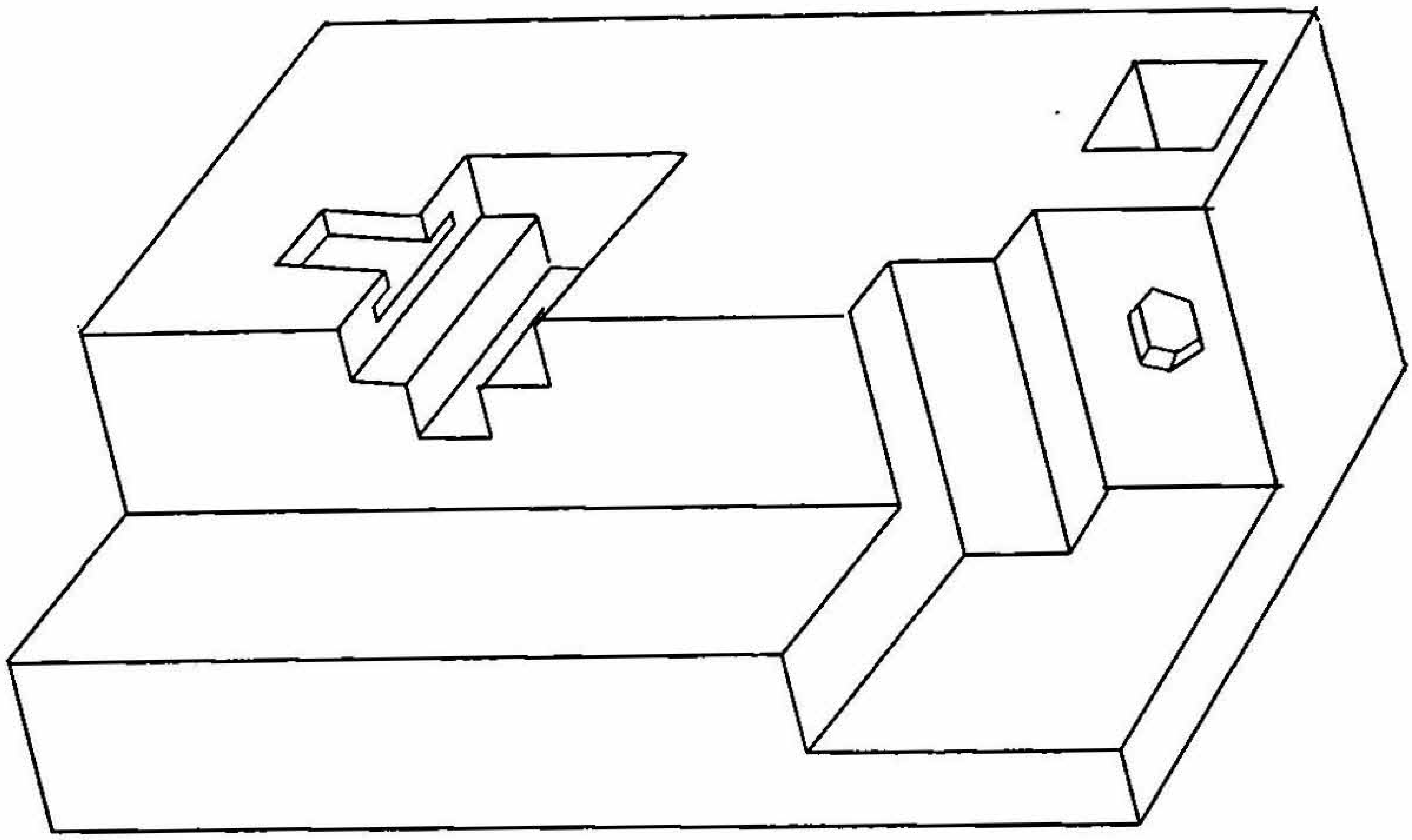

FIG. 8. Example object constructed by NP and SM algorithms.

goal of reverse engineering). As the next step the requirement of neighbourhood information was removed.

## 3.3. *Construction of solid model from structured data*

In this approach[14, 20] the input are points which are required to be clustered with respect to the faces in the object. However, no other neighbourhood information is required. The sets of points measured on each face of the part are used to determine the best fit plane/surface. The orientation of the normal of the surface is derived using the origin of the measuring device. The algorithm (CXCOVER) works with planes (surfaces are represented as planes by the algorithm for determining the connectivity).

The algorithm is based on the hypothesis that any nonconvex polyhedron can be represented as a combination of several convex polyhedra. These convex polyhedra are constructed from the set of planes obtained from measured points. Combinations of the input planes are used to construct convex polyhedra. It may be noted that all combinations may not result in a convex polyhedron. Each convex polyhedron generated is then checked for containment against the input point data. The containment check involves checking that no input point is in the interior of a convex polyhedron. These are then combined using boolean union operations to realize the solid model.

Since this approach is based on enumerating all possible convex objects that can be formed by the planes, it is computationally expensive. The number of convex solids enumerated is reduced by checking for the containment of the input point data in the boundary of the model constructed at any stage. If all the points are contained in the

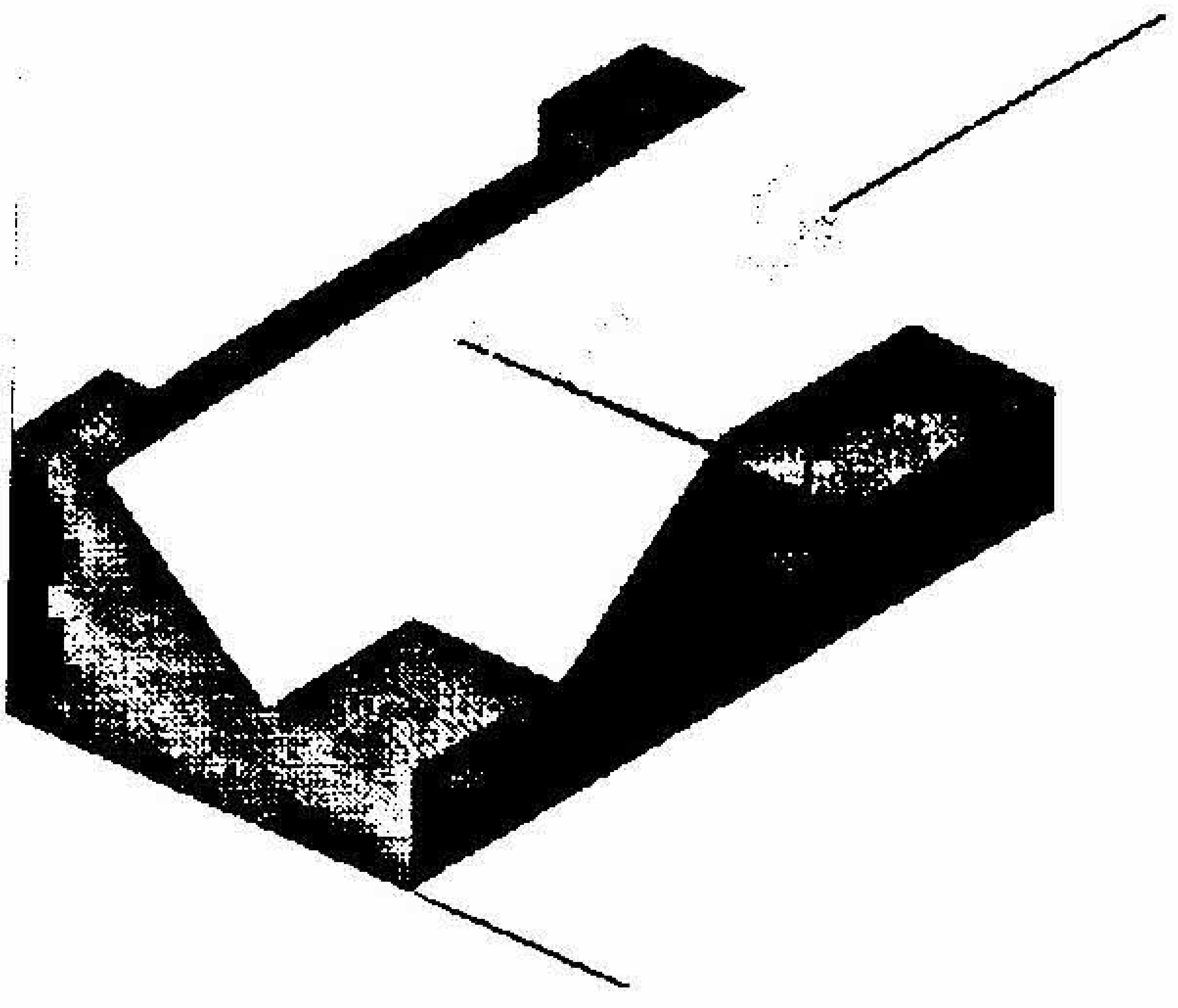

FIG. 9. Pivot block.

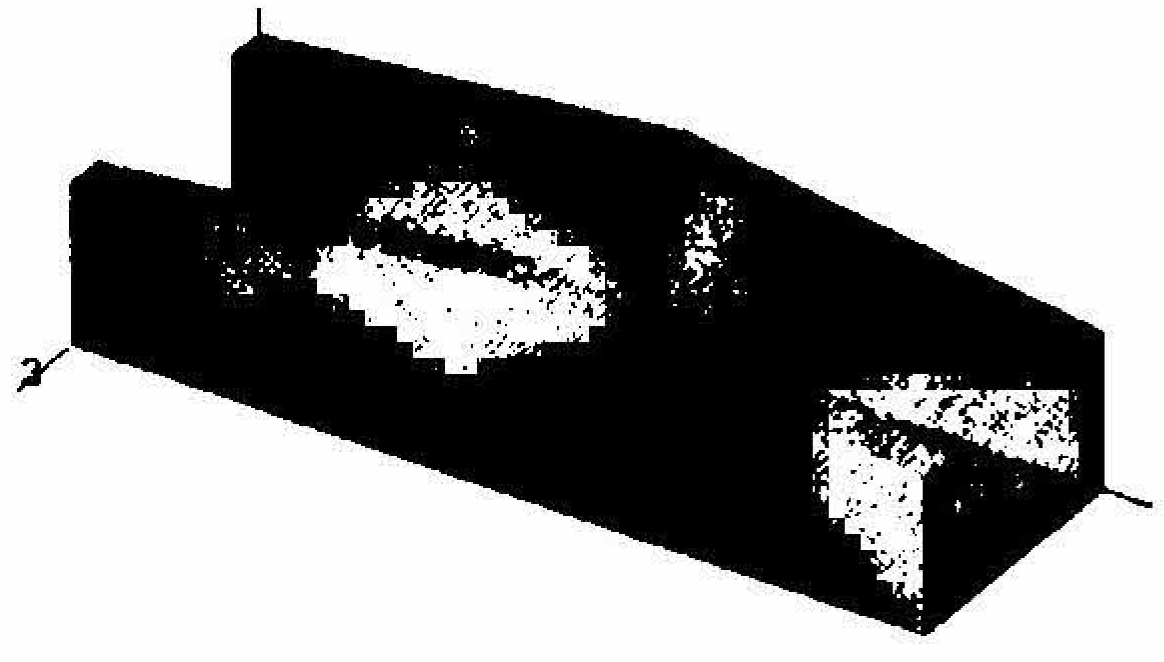

FIG. 10. Solid model constructed from inspection data.

boundary, the algorithm terminates. At this point, however, it is not possible to determine a priori the number of combinations that have to be generated. The computational effort can be further reduced by considering only those planes that are not on the bounding polyhedron (inner planes) for enumerating the convex solids. In this case, the resulting convex solids would have to be subtracted from the bounding polyhedron.

Some example solids constructed by the algorithms presented are shown in Figs 9 and 10. Point data for object in Fig. 9 has been generated from an in-house modeller. Figure 10 shows a model generated from inspection data generated in CATIA.

Noise in the input data will affect the interpolating routine. This will not affect the determination of the topology/connectivity by the algorithm, provided the geometric checks (containment of a point on a plane, etc.) take into account the presence of noise in the data.

Measurement of boundary points (of a face) of most engineering solids is difficult as the corners are rounded off by fillets or blends. So the proposed method is the most suited for reverse engineering of parts as it is the only approach that uses only internal points measured on a face. Blends and fillets in the original part can be added after construction of the solid model of the part with sharp edges/corners.

The construction of convex solid is not possible when the object contains curved surfaces. For determining the connectivity, the surface is approximated by a plane[14] and the convex covering algorithm is used. The exact representation of the surface is used while determining the geometric entities—edges and vertices, and while combining the convex solids. Figure 11 shows the solid model of a part with curved surfaces that has been constructed from point data.

The above discussion is restricted to analytic surfaces only, because interpolation of points would yield an implicit representation that also defines the surface in the region

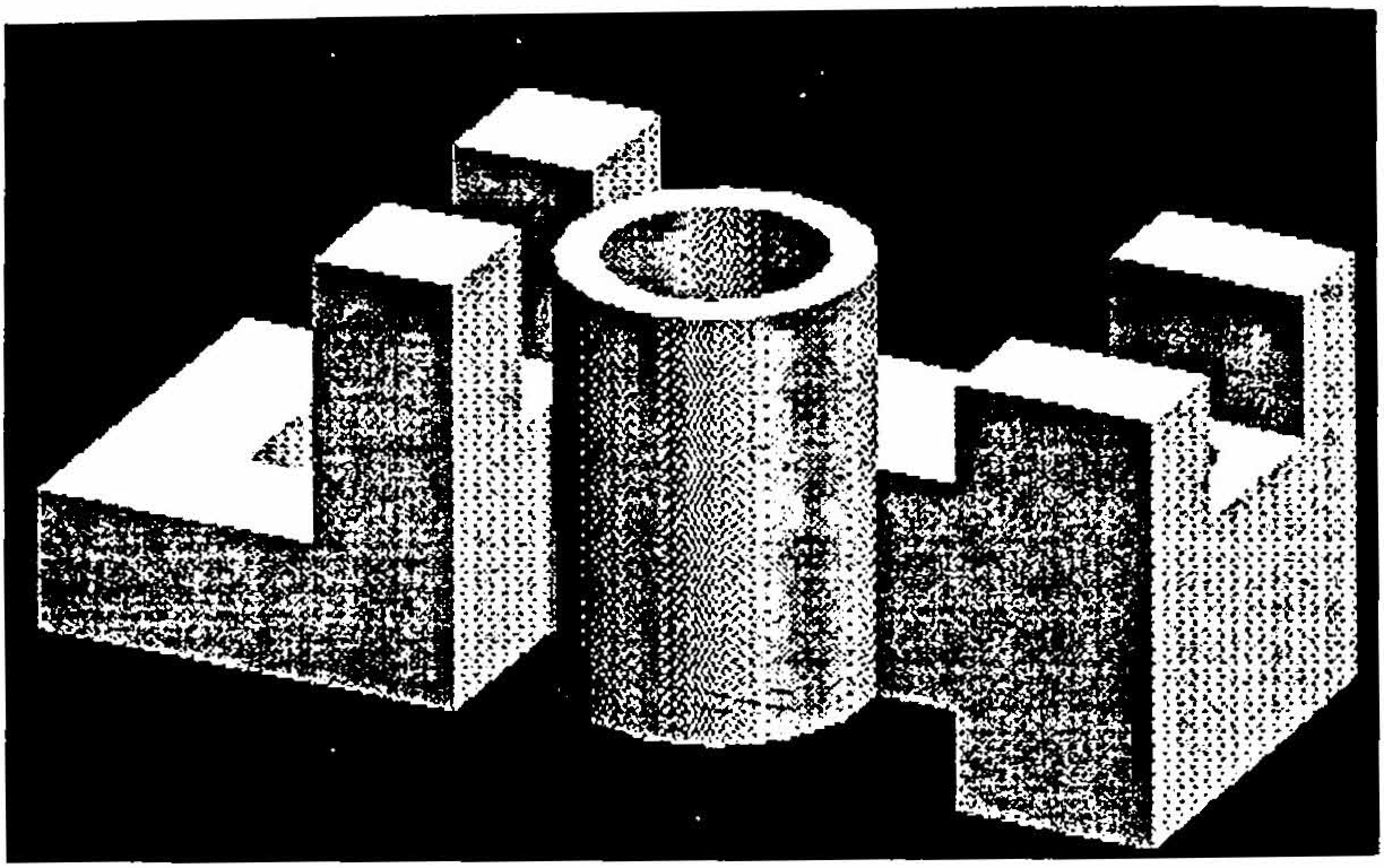

FIG. 11. Test solid with curved faces.

beyond the extent of the measured point data. However, if the surface is a free-form surface, the interpolation results in a parametric form that is bound by the extreme points measured. Therefore, the interpolation will be correct only if the points on the boundary of the face are also included in the input set. Definition/representation of a parametric surface beyond the extent of the point set is an open problem.

In reverse engineering from physical prototypes, currently work is in progress to handle dense data that are obtained by optical scanning. One problem with sparse data is that the measurement process is tedious and the clustering of the data requires intervention by the operator. Automation in reverse engineering can be enhanced greatly by using dense data as is obtained by optical probes/digitizers. As in the generation of sparse data here too multiple orientations of the part (and the probe) are required to scan the complete part. Merging multiple range image data is itself a nontrivial problem.

In a departure from current practice (of constructing tessellated models for visualization), construction of 3D solid model from the merged data is being explored. Here the input data has to be first clustered. Delaunay triangulation of the point set would yield an approximate surface (represented by triangles) that interpolates the input points. This approximate tessellated surface can now be processed to determine the regions of sharp/not so smooth transitions. Once these regions are identified the data can be clustered to represent regions of no/little transitions. These regions would correspond to the faces in the object. The input points in this region can then be interpolated to determine planes/surfaces. The CXCOVER algorithm can now be used to obtain the solid model of the part.

The uniqueness of the representation of an object by points on its faces has not been proved yet. This could be an interesting problem in computational geometry.

## 4. Concluding remarks

Contrary to the obsolescence that appears to be intrinsic to this class of problems (clearly if all legacy data are converted to CAD models, whither reverse engineering), it is now apparent that the underlying problem of constructing CAD solid models from point or 2D data will continue to be of significance in the area of CAD.

In keeping with the emerging view of CAD/CAM as a collection of enabling tools rather than as tools for automation, it appears that in model construction, it would be desirable to have the designer model the part geometry in 2D. This is consistent with the training of most designers and eliminates the difficulties in handling 3D data in a 2D medium (computer screen). The problem of constructing 3D models from 2D data has resulted in tools that are now being considered as an alternate means of inputting geometry into the CAD system. Tools developed here can now be used to allow the user to design the part in 2D (as most designers are trained) and obtain the 3D model as output. Therefore this problem is likely to remain significant even if all the legacy data gets converted to CAD models.

In a like manner (as in legacy data), reverse engineering from point data has resulted in tools that may find application in the interface between CAD and rapid prototyping(RP). Techniques for surface interpolation and solid model construction are likely to yield a more robust solution to the problem of bad STL[21] files that are now used to transit from CAD to the rapid prototyping apparatus.

The integration of rapid prototyping with CAD is through the slicing software. Irrespective of the RP process, the slice software can handle a tessellated model only. This is because slicing an arbitrary, nonconvex object is not straightforward and requires considerable effort. The CXCOVER algorithm generates a set of convex polyhedra that combine to form the object. Slicing of a convex polyhedron is straightforward and is not prone to the kind of problems that occur with arbitrary solids. By eliminating the overlapping portions in the set obtained from CXCOVER, a set of convex volumes can be obtained that form the solid. These volumes can then be sliced in a straightforward manner and used as input to the rapid prototyping apparatus.

As the realization of the physical prototype is now possible in hours, quick validation of the reverse engineered model is possible. A schematic of the whole process in shown in Fig. 12. This integrated setup will also be of value in rapid product development, especially to tooling manufacturers. It would also form an excellent teaching faclilty in the areas of product design and manufacture.

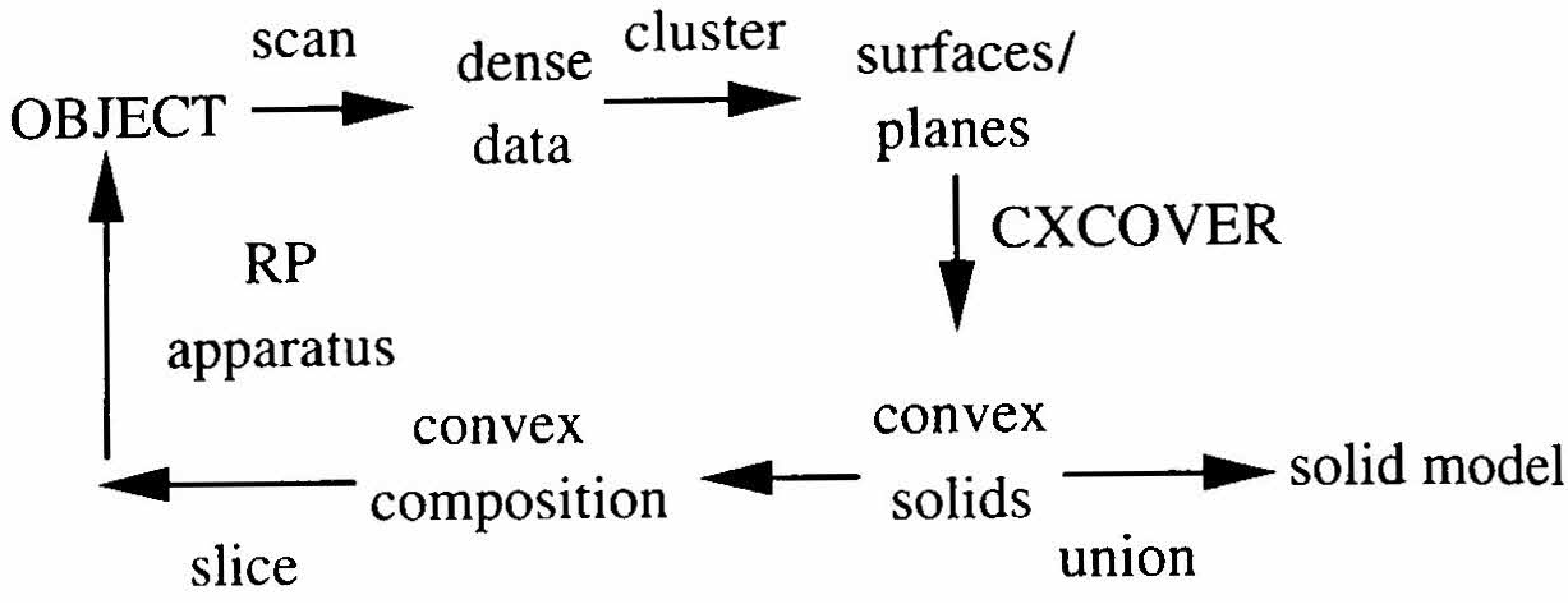


FIG. 12. Integration of reverse engineering with rapid prototyping.

The algorithms developed for reverse engineering of solid models, (irrespective of the source of input—drawings or measured point data) work by constructing simpler objects that are then combined to form the actual object. The simple objects first obtained are mostly convex and always a swept solid. The use of swept solid also forms the underlying basis for the work on features technology in the laboratory. It remains to be seen if this will lead to some form of a unifying representation scheme for modelling and manipulating part geometry.

## Acknowledgements

All the work described here has been done in the Robotics and CAD Laboratory by students and project staff, over a period of five years. The work on algorithms to handle legacy data has been largely due to S. Srinivasan (M.E., '90), D. Rastagi (M.E., '91), B. S. Annapurna (Project Assistant), B. S. Subrahmanyam (M.E., '93), K.Saha (M.E., '94) and Manoj (M. Sc). The work reported on reverse engineering from measured point data is from the dissertations of B.Ganesh (M.E., '93), N. Shyamsundar (M. Sc '95) and the ongoing work of A.Soni (M. E.). Most of this work has been financially supported by the Aeronautical Development Agency, Bangalore, through two sponsored projects.

## References

1. PRIESS, K. AND MERON, O. Variant and generative methods for constructing solid data from 2-D projections, Geometric modelling for product engineering, *IFIP Conf.*, 1990, pp. 197–203.
2. GURUMOORTHY, B. *Automatic generation of and reasoning about geometric models for manufacturing*, TR/IISC/ME/RCL/C-1993, Department of Mechanical Engineering, Indian Institute of Science, Bangalore, 1993.
3. SUBRAHMANYAM, S. *Solid model construction from 2-D information*, M. E. Thesis, Department of Mechanical Engineering, Indian Institute of Science, Bangalore, 1993.
4. SAHA, K. *Solid model construction from 2-D information*, M. E. Thesis, Department of Mechanical Engineering, Indian Institute of Science, Bangalore, 1994.

5. SRINIVASAN, S. *Automatic generation of 3-D mechanical object shapes*, M. E. Thesis, Department of Mechanical Engineering, Indian Institute of Science, Bangalore, 1991.

6. RUSTAGI, D. *Generation of solid model from 3-D wireframe*, M. E. Thesis, Department of Mechanical Engineering, Indian Institute of Science, Bangalore, 1991.

7. MARKOWSKY, G AND WESLEY, M. A. Fleshing out wireframes, *IBM J. Res. Dev.*, 1980, **24**, 582–597.

8. RUSTAGI, D. AND GURUMOORTHY, B. Generation of solid model from 3-D wireframe, *Proc. National Conf. on Design Engineering*, NACOMM-91, 1991, pp. 386–393.

9. LEQETTE, R. Automatic construction of curvilinear solids from wireframe views, *CAD*, 1988, **20**, 171–180.

10. DUTTA, D. AND SRINIVAS, L. Reconstruction of curved solids from two polygonal orthographic views, *CAD*, 1992, **24**, 149–159.

11. SAHA, K. AND GURUMOORTHY, B. Automatic generation of 3D solid models from orthographic projections, *Proc. ASME Design Automation Conf.*, 1994.

12. ALDEFELD, B. AND RICHTER, H. Semi-automatic three dimensional interpretation of drawings, *Comput. Graphics*, 1984, **8**, 371–380.

13. GANESH, B. *Reverse engineering from physical prototype*, M. E. Thesis, Department of Mechanical Engineering, Indian Institute of Science, Bangalore, 1993.

14. SHYAMSUNDAR, N. *Automatic construction of solid models from measured point data*, M. Sc. (Engg.) Thesis, Department of Mechanical Engineering, Indian Institute of Science, Bangalore, 1995.

15. DEROSE, T, *et al.* Surface reconstruction from unorganized points, *Comput. Graphics*, 1992, **26**, 71–78.

16. MÜCKE, E. P. M. AND EDELSBRUNNER, H. Three-dimensional alpha shapes, *ACM Trans. Graphics*, 1994, **13**, 43–72.

17. MUKHERJEE, S. AND GURUMOORTHY, B. Surface approximation in reverse engineering. *Proc. ASME Design Automation Conf.*, 1994.

18. RAMAKRISHNA, V. *Interpolation of surfaces with multiple patches for reverse engineering*, M. E. Thesis, Department of Computer Science and Automation, Indian Institute of Science, Bangalore, 1995.

19. CHIVATE, P. N. AND JABLOKOW, A. G. Solid model generation from measured point data, *CAD*, 1993, **25**, 587–600.

20. SHYAMSUNDAR, N. AND GURUMOORTHY, B. Reverse engineering of CAD models from physical prototypes, *Proc. Ninth World Cong. on the Theory of Machines and Mechanisms*, 1995.

21. JACOBS, P. F. *Rapid prototyping & manufacturing—Fundamentals of stereolithography*, SME, 1992.