Volume Based Approach to Extract 3-D Parts from 2D Assembly Drawing

Bharat Bhushan [1] and B. Gurumoorthy[2]

[1]Indian Institute of Science, bharatiisce98@rediffmail.com
[2]Indian Institute of Science, bgm@mecheng.iisc.ernet.in

ABSTRACT

This paper extends an earlier volume-based approach to automatically reconstruct 3D parts from the 2D orthographic views of an assembly drawing. Most of the work in automatic reconstruction of 3D solids using volume-based approach from 2D orthographic views has been done only for single component drawings. The proposed method identifies elementary solids called candidate blocks from the orthographic views and then combines these solids to obtain different parts in the assembly. The candidate blocks are combined on the basis of some conditions defined based on the validity of solids and characteristics of an assembly. Earlier efforts towards addressing multi-part assembly drawing either require additional inputs in the form of labeling of parts in the drawing, additional special cross-sectional views or are restricted to assemblies with standard parts. Results have been achieved for some typical engineering assembly drawings. Future work will address extension of this algorithm to assembly drawings with sectional views.

Keywords: assembly drawings, 3D parts, orthographic views, volume-based

1. INTRODUCTION

Conventionally an engineering drawing has been a means to store information about complex 3D parts in 2D form. The drawing contains all the required information about the 3D object in an accurate, concise and systematic way. In industries, there is huge legacy of data and product information available in the form of engineering drawings (either in the hard copy form of blue prints of drawings or in the digital/soft form generated by available computer-aided drafting tools), which are continuously used in new product development. This form of shape representation continues to increase due to its familiarity for users and consequent ease in manipulation. Understanding engineering drawing requires a thorough knowledge of the rules and conventions of engineering drawing and industry/organization specific protocols. As the complexity of the drawing increases (sectional views, auxiliary views), the skills required for interpretation also increases. A 3D solid model is becoming essential for other design tasks such as CFD/FEM analysis, rapid prototyping, and simulation. Automatic construction of 3D solid model from 2D drawing is of great importance given the penetration of 3D models in product development tasks and the specialized skill required to manually interpret the drawing. Automatic conversion of 2D views of engineering drawings into 3D model has been a challenging and popular problem in CAD area. Since 1980, many methods for reconstruction of solid model from orthographic views have been proposed and implemented. The approach for automatic reconstruction of a solid model from a 2D engineering drawing can be broadly classified as:

Wire frame approach or bottom-up approach [2] [4] [11]
The bottom-up approach first identifies the 3D vertices from the vertices in the views in the drawing. Connectivity of the vertices in the views is used to then construct 3D edges. A 3D wireframe is then constructed with the 3D edges and vertices based on the connectivity information available in the views. Loops of edges are constructed from the 3D edges and elementary solids are formed. These are then combined to obtain the 3D solid.

Volume based approach or top-down approach [7] [8]
The volume based approach works in an opposite manner. Here elementary solids are directly obtained by sweeping of matching loops in the input views. By sweeping each of the loops forming the matching set, three intersecting solids are obtained. Intersection of these solids gives an elementary solid corresponding to these three matching loops.

Similar elementary solids are obtained from other sets of matching loops in the views. The final solid is then obtained by combining these elementary solids.

While these address the conversion of a large amount of legacy data available in the form of drawings to solid model. It has been found that a significant portion of the legacy data pertains to drawing of assemblies. It is also believed that from the point of view of reuse of designs, it is the assembly drawings that are more critical than the single component drawings. Most of the work till now has only addressed the automatic construction from engineering drawings of single components. Very little work has been done for multi-component assembly drawings. In practice, there are large numbers of engineering drawings which are not single component drawings but drawings of assemblies. There is a need therefore for some method by which 3D parts can be identified and reconstructed from the 2D views of an assembly.

1.1 Related Work

The problem of identifying 3D parts in an assembly from a drawing of the assembly has been addressed by Tanaka *et. al* [9][10]. They first proposed a method using *solid elements*[9] to isolate the 3D parts. A solid element is a closed region of faces and is the same as the elementary solids obtained in the wireframe approach. They define four conditions to combine the solid elements to form 3D parts corresponding to the assembly drawing. Of these, two are for combining solid elements and the other two are for eliminating spurious solid elements. Their method requires additional labeling of part numbers (which is standard in assembly drawings) to perform the combination of solid elements correctly. Additional input in the form of cross-sectional views is required for capturing interior details and pruning the number of solutions. The location and number of the cross-sectional views is also critical as it is not clear how many cross-sectional views will be required to capture all the internal details and the number of cross-sectional views required increases as the complexity of the 2D drawing increases.

In a subsequent paper Tanaka *et. al* [10] restrict their approach to assembly drawing consisting of standard parts such as bars and plates to obtain a single solution. The input assembly drawing contains dimensions, part numbers and parts list. Using this information the portions of the drawing corresponding to the standard parts are first identified (referred to as *virtual 2D part drawings*). Solid elements are generated from these virtual 2D part drawings and combined using conditions in their earlier paper[9] to get the 3D parts.

1.2 Present Work

In the proposed method an existing volume based approach[8][3] has been used and extended for obtaining candidate blocks from the 2D views of an assembly. Conditions are defined for grouping/combining these candidate blocks into different 3D parts of the assembly. These conditions are derived from the validity conditions for a solid and characteristics of an assembly. Proposed method is more general as it is not limited to assembly consisting of only standard bars and plates. If the information contained in the 3 views is not sufficient to resolve ambiguities in the interior of the assembly the method generates all possible valid solutions. Given an additional side view (that is different from the other input views), a unique solution is identified. In contrast to earlier efforts [9][10], the present method does not require cross-sectional views to capture interior details and prior information about the part numbers in the drawing.

The input is assumed to be available as a 2D edge-vertex graph (vector representation). This data could be obtained by image processing of scanned paper drawings [1] or as the output of some computer - aided drafting tool. The present work assumes that all the input data is available as a noise free vector data. It is also assumed the input vector data is complete, sufficient and consistent as per the standard rules of engineering drawing. In this algorithm, types of faces are limited to planar, cylindrical, conical and spherical faces. This limitation is only in the construction of the candidate blocks and not in the algorithm to combine candidate blocks to form the parts.

2. DEFINITIONS

Candidate block: A candidate block is a region closed by faces which has NULL volume of intersection with other candidate blocks. These are analogous to the blocks used by Markowsky and Wesley [6]. These candidate blocks when combined form the 3D parts. Candidate blocks are further classified as:

- *True candidate block*: Candidate blocks that actually exist in the final parts. True candidate blocks combine to form the parts in the assembly. For example, S1, S3, S4, S5 of Fig. 1 are true candidate blocks.

- *False candidate block*: Candidate blocks that do not actually exist in any part but are formed during the generation stage are called false candidate blocks. These are due to the matching of loops which may result in some unnecessary candidate blocks. These are identified and removed from the list of candidate blocks. S2 in Fig. 1 is a false candidate block.

- *Internal candidate block*: A candidate block whose boundary (*all faces*) is shared by other candidate blocks is called an internal candidate block. In the example in Fig. 1, blocks S1, S2 and S3 are internal candidate blocks.

- *External candidate block:* A candidate block that has some part of its boundary (face or part of it) not overlapping with any other candidate block is called an external candidate block. Blocks S4 an S5 in Fig. 1 are external candidate blocks.

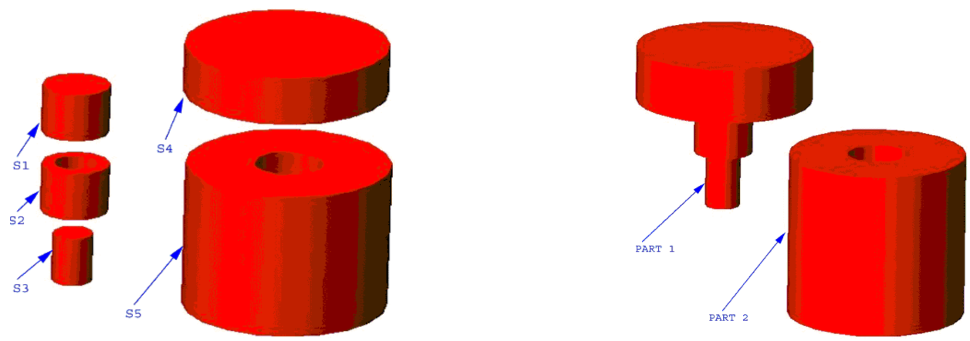

Fig. 1. Classification of candidate blocks.

Free face: A face of a candidate block which does not overlap with any face of any other candidate block.

Tangential faces: Two faces of a pair of candidate blocks are said to be tangential, if they share an edge and there is C^1 continuity at that edge.

Non -Tangential faces: Two faces of a pair of candidate blocks are said to be non tangential, if they share an edge and there is only C^0 continuity at that edge.

Classification of loops:
A loop is classified to be IN, ON-IN or ON-OUT with respect to another loop depending on its edges being all inside, none outside or none inside the other loop respectively.

A loop is called *solid* if all its edges are solid (not hidden) and *dotted* if it contains at least one dotted (hidden) edge.

Sign convention for loop: A loop is classified to be positive or negative with respect to its immediate solid loop. The following are the sign convention for a pair of matching loops:

Positive Loop: Any loop which is IN or ON-IN for even number of solid loops is called positive loop.
Negative Loop: Any loop which is IN or ON-IN for odd number of solid loops is called negative loop.

3. OVERVIEW OF THE PROCEDURE

The procedure works by first finding all possible elementary volumes, which are consistent with the input views of given assembly drawing. Being consistent implies that the volumes do not add any extra edge in the input views when projected along the respective view directions. The elementary volumes can be obtained by using either wireframe approach[2][4][6] or volume based approach[3][7-8]. In the present method elements are obtained using volume based approach[8]. These elementary volumes are used for getting candidate blocks. The procedure then combines these candidate blocks on the basis of defined conditions to form parts of the assembly. The volumes that need to be combined are determined based on ideas derived from the notion of a valid solid.

After having generated necessary candidate blocks for the given assembly drawing, these blocks are combined on the basis of some defined conditions to produce the required 3D parts. The steps involved in getting 3D parts are as follows:

i. All the candidate blocks are flagged as internal and external. This is required for further processing of the candidate blocks.
ii. All the false external candidate blocks are identified and are removed from the list of candidate blocks so that these does not come into consideration while further analyzing other candidate blocks.
iii. Combine all the external candidate blocks sharing dotted edges in a view and flag them as decided.
iv. Combine all the internal - external candidate blocks pair based on Continuity condition. The combined solid is flagged as external and modified.
v. Combine all the external candidate blocks sharing solid edges using the two conditions and flag them as decided.
vi. Combine the remaining internal - external candidate blocks based on minimum overlapping faces condition.
vii. Remove false internal candidate blocks.

4. DETAILED STEPS IN THE PROCEDURE

In this section the detailed steps in the procedure is described.

4.1 Construction of Candidate Blocks

In the proposed method all the loops in a matching set (*onto- formation*[8]) are swept along the respective view directions. The number of loops in the matching set is the same as the number of input views. The length of sweep is taken as the maximum length present in any of the views. Elemental solids bounded by cylindrical, toroidal or spherical faces are constructed using rotational sweep. For this information about lamina to be swept, axis of rotation, point from where axis will pass and span of rotation is required which comes from loops in the given input views [8]. The intersection of these swept solids yields an elementary solid.

These elementary solids are then classified with respect to each other. Elementary solids whose volume of intersection is NULL with all other elementary solids are taken as candidate blocks. Elementary solids which have finite volume of intersection with the other elementary solids are then converted to candidate blocks (with quasi-disjoint volumes) as follows. If A and B are two elementary solids having a finite volume of intersection (Fig. 2, left), then their volume of intersection, A∩B is obtained (Fig. 2, centre). Subtracting this from A and B yields A-B and B-A. The number of faces of A∩B that overlap with A-B and B-A respectively, are counted. If A-B has minimum number of overlap with A∩B, then A and B-A are taken as the candidate blocks otherwise B and A-B are taken as candidate blocks (Fig. 2, right).

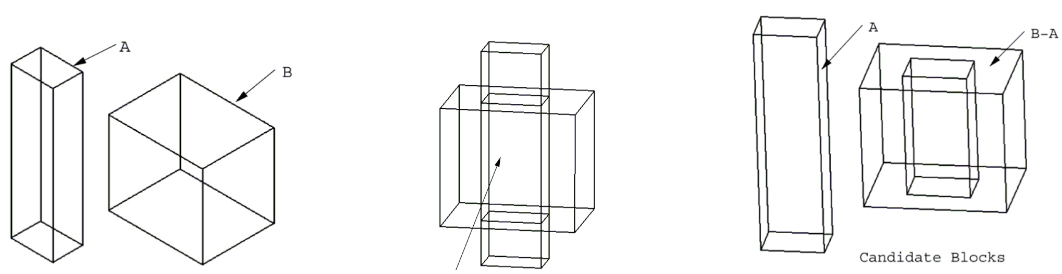

Fig. 2. Candidate blocks from elementary solids.

4.2 Classification of Faces of Candidate Blocks

For classification and combination of candidate blocks, classification of faces of two candidate blocks with respect to each other is required. As the candidate blocks are quasi-disjoint, faces of these can either overlap or not. The following types of classification are possible:

No overlap: In this case both faces do not make any contact (they do not touch each other) or their intersection results in one dimensional entity.

Complete overlap: In this case each face in a pair of face is completely contained by other face.

Partial overlap: Here for a pair of faces, one face completely contains the other face while other does not.

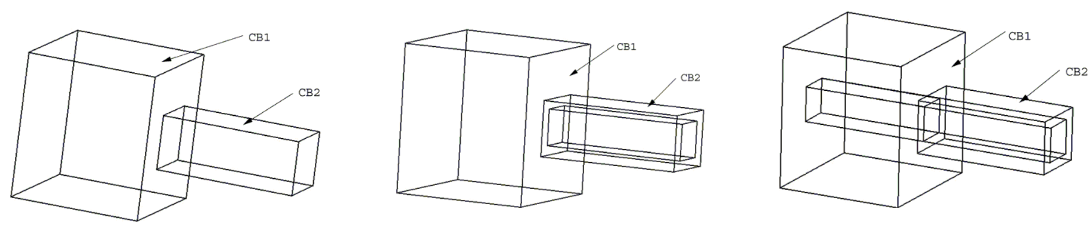

Fig. 3. Classification of overlapping faces.

These are of two types:
Partial overlap of type 1: In this case the overlapping portion is a simply connected entity when both the overlapping faces are simply connected type and will be multiply connected if both the overlapping faces are multiply connected type (Fig. 3, left and right).
Partial overlap of type 2: In this case overlapping portion of faces is a multiply connected entity only. Here one face is simply connected whereas other is multiply connected (see Fig. 3, center).

4.3 Algorithm for Flagging Candidate Blocks as Internal or External

Each candidate block is flagged as internal or external. All the faces of a candidate block are checked if they overlap with faces of other candidate blocks. If all faces (entire boundary) of a candidate block are overlapped by the faces of other candidate blocks, then the candidate block is flagged as internal. Otherwise the block is flagged as external. For example in Fig. 4 all the six (2planar and 4 cylindrical) faces of S2 are covered by the faces of S1 and S3, therefore it is flagged as internal. Whereas only one face of S1 overlaps with S2 and S3 and remaining five are free faces, therefore it is external. In the case of S3 only four cylindrical and 2 planar faces overlaps, remaining 5 faces are free, therefore it is flagged as external.

4.4 Conditions for Removing a False Internal Candidate Block

A false internal candidate block is an internal candidate block that has at least one loop with all its edges dotted and which remains internal and undecided (candidate block S2 in Fig. 6).

4.5 Conditions for Removing False External Candidate Block

A false candidate block is one that will not be a part of the final solution. These candidate blocks get generated due to the ambiguity in the wireframe. The straightforward way of identifying and removing these blocks is to check if the views obtained by projecting all the other candidate blocks matches exactly with the input views. If the projections thus obtained exactly match with the given orthographic views, then the temporarily removed external candidate block is considered as false. It is important to remove the false candidate blocks from further consideration because otherwise, they will continue getting processed under the other rules without contributing to the solution.

As the above check is expensive to implement, we do the following instead. The set of loops responsible for the construction of the candidate block is maintained. If a candidate block is associated with a solid loop that has no boundary edges, then the candidate block is flagged as false. This is because when a candidate block has a solid loop that has no boundary edges and faces of other candidate blocks also share these edges. If we now remove the first candidate block then the views of the remaining candidate blocks will still contain all the solid edges due to the presence of the adjacent candidate blocks and there will not be any violation of input information. Hence we can remove the candidate block associated with a solid loop that has no boundary edges by labeling it as a false candidate block.

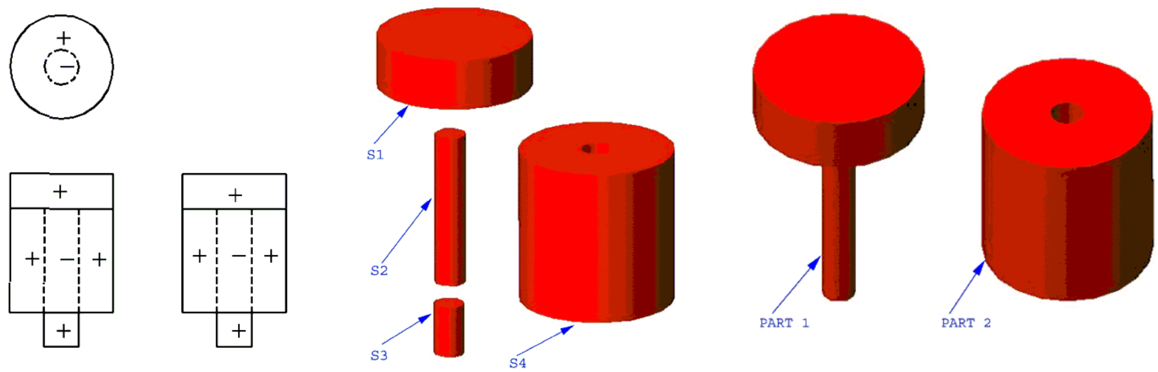

Fig. 4. Combining candidate blocks to form parts.

4.6 Combining Candidate Blocks

All the true external and undecided internal candidate blocks are combined to form the 3D parts that will compose the assembly. The candidate blocks are combined using the following conditions. These conditions are based on the adjacency of faces in the blocks and the adjacency of the corresponding loops in the input views. Faces of two blocks can be tangential or non-tangential. There are two types of adjacencies possible between loops. The loops in a view share a dotted edge or they share a solid edge.

I. *External candidate blocks with dotted edges*
This condition is applied only for external candidate blocks. If there is a pair of tangential faces belonging to two different candidate blocks and whose corresponding loops share exactly one dotted edge in at least one input view, then these two candidate blocks can be merged into one block.

II. *External candidate blocks with solid edges*
Two different conditions for combining such solids are defined.
i. Condition 1 - If there is a pair of faces belonging to two different candidate blocks that are non-tangential with their corresponding loops in at least one view sharing a solid edge, then these two candidate blocks can be merged into one block. When there is a solid edge between two tangential faces of a pair of external candidate blocks then these 2 candidate blocks cannot be merged. This is because; merging will remove the solid edge and thereby violate the input information.

The above two conditions have also been used by Tanaka et. al[9] to combine solid elements. However they do not distinguish between internal and external solid elements.

ii. Condition 2 - A pair of candidate blocks having only one partial overlap of type 1 can be merged into one block.

III. *Continuity condition for union of internal and external candidate blocks*
A valid solid can be composed by combining a pair of candidate blocks that have overlapping faces. Two candidate blocks forming a solid cannot be connected only at vertices or edges. The continuity condition is applied to a pair of one internal and one external candidate block. This pair can be merged into one solid if there is only one face in each block that completely overlaps the other. If more than one external candidate blocks are satisfying continuity condition with one internal candidate block, then first all the external candidate blocks are merged with the internal candidate block and then this modified merged solid is flagged as external. This condition is only applied to a pair of one internal and one external candidate block. The case of two external candidate blocks is handled by condition 1 for external

candidate blocks sharing solid edge. Continuity condition is not applied between a pair of internal candidate blocks, as it will omit one dotted edge from the input views.

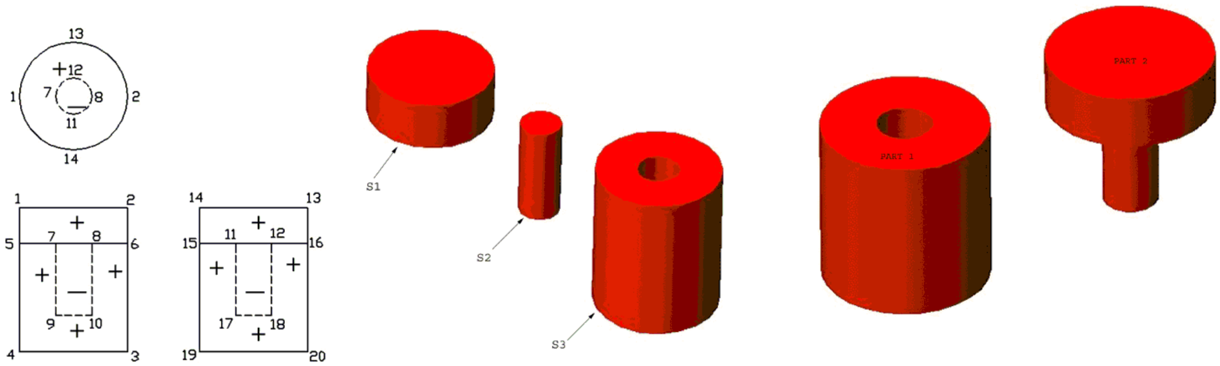

Fig. 5. Illustrative example – Test solid 1.

IV. *Minimum overlapping faces condition*
This is the last condition applied to decide further combinations of the candidate blocks. Among the candidate blocks that have not been flagged thus far, a candidate block is merged with that candidate block with which it has minimum number of overlapping faces. This condition is first applied to the internal blocks that have not been flagged for combining them with the flagged external candidate blocks. Then the condition is applied to decide if there are unflagged internal candidate blocks that can be combined to each other. This is required only when there is some part of the assembly is made of only internal candidate blocks. Finally, this condition is applied to the external candidate blocks to check for and generate multiple solutions if any.

This condition is based on the premise that the assembly results in restricting the degrees of freedom between parts. Therefore the blocks that restrict other blocks (by overlapping) should belong to different parts.

In addition to the above conditions there are some heuristics that have been developed based on the signs of loops associated with the candidate blocks to decide when to combine candidate blocks?

- Two candidate blocks that have faces associated with different solid loops can be united on the basis of minimum overlapping faces condition.
- Two candidate blocks are not combined if they contain positive and negative loops for the same solid loop. As this merging will lead to the disappearance of some of the dotted edges.
- Two candidate blocks, having same sign loops with respect to the same solid loop in at least one view, can be added only when the common faces are not completely overlapping. This has priority over minimum overlapping faces condition.

5. RESULTS AND DISCUSSION

The procedure described has been implemented as an extension of an existing system for automatic construction of 3D solids from 2D drawings [8][3]. Input to the system is a .DXF file containing the drawing. The .DXF files are parsed [8] and processed to obtain the vertex-edge graph for each view [5]. The existing system is used to construct the elementary solids from the vertex-edge graph. The proposed procedure for identifying, classifying and combining candidate blocks has been implemented in `C' with the ACIS kernel used for geometric operations and display of results.

5.1 Illustrative Example

The input views are shown in Fig. 5. For this example, two sets of matching loops are found to be – ({1,13,2,14,1}, {1,2,6,8,7,5,1} and {14,13,16,12,11,15,14}) and ({7,12,8,11,7}, {7,8,10,9,7}, and {11,12,18,17,11}). A third set of matching loops ({1,13,2,14,1}, {5,7,8,6,3,4,5}, {15,11,12, 16,20,19,15}) is obtained by merging loops in the views. Sweeping each loop from the sets results in three intermediate solids. Intersection of these intermediate solids results in an elementary solid. The first two sets result in solids S1 and S2. The third set results in a cylinder which has an overlapping region with S2. Following the procedure to obtain disjoint regions, S3 along with S1 and S2 are obtained as the candidate blocks. Classification of the candidate blocks results in S2 classified as internal and, S1 and S3 as external blocks. There are no false candidate blocks. The two external blocks cannot be combined under any of the three conditions. Applying the minimum face condition next, it is found that S2 and S3 cannot be combined as they contain positive and negative loops with respect to different solid loops. S1 and S2 are combined to form part 1 as they satisfy the minimum face condition of one external and one internal block with loops from different solid loops. Candidate S3 is the other part (part 2) forming the assembly, which is the desired result (Fig. 5).

5.2 Results

The candidate blocks generated from the input views given in Fig. 6 (left frame) are shown in the middle frame. S4 and S5 are external and remaining three are internal candidate blocks. Loop sign convention restricts union of S1 and S5, S2 and S5 and, S3 and S2 respectively. Based on the minimum face condition blocks S4, S1 and S3 are combined into part 1. S2 remains internal and undecided. It is therefore removed from the list of candidate blocks. S5 alone makes part 2. The resulting parts are shown in the right frame in Fig. 6.

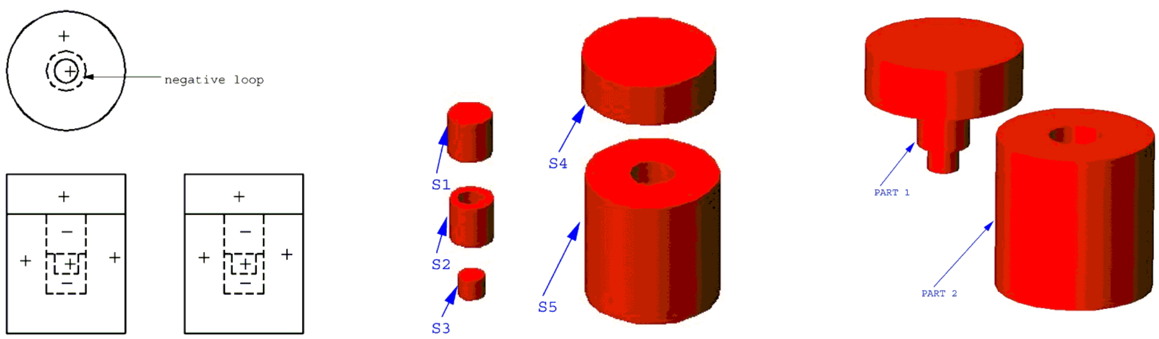

Fig. 6. Results for Test solid 2.

The input views for a cotter joint are shown in the left frame of Fig. 7. The three parts obtained from the 9 candidate blocks generated are shown in the same figure (right frame). 2D assembly drawing of unprotected type of flange coupling is shown in Fig. 8. In this case 34 candidate blocks are generated. Combining these candidate blocks using the rules developed results in two segments of shafts, four bolts, two flanges and four nuts totaling twelve 3D parts and are shown in the right frame of Fig. 8.

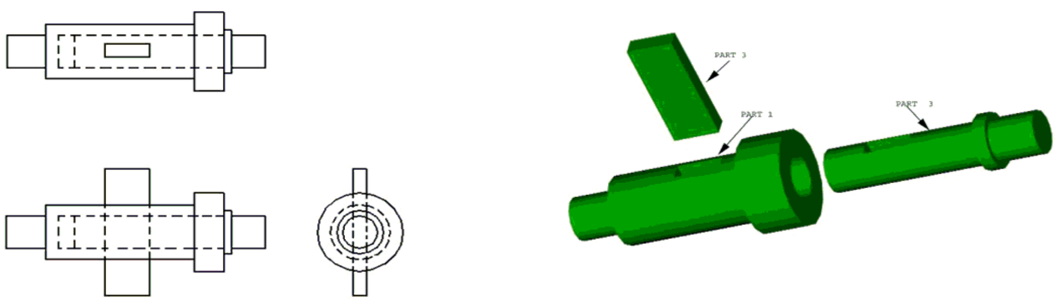

Fig. 7. Construction of 3D parts in a cotter-joint assembly.

The procedure presented here can also generate multiple solutions where they are possible. From the input views shown in Fig. 9, twelve external candidate blocks are generated. In this case there are a few undecided external candidate blocks after the rules for combining candidate blocks have been applied. The minimum face condition then generates four different sets of parts that can be used to compose the given assembly.

Fig. 8. Construction of 3D parts in a flange joint assembly.

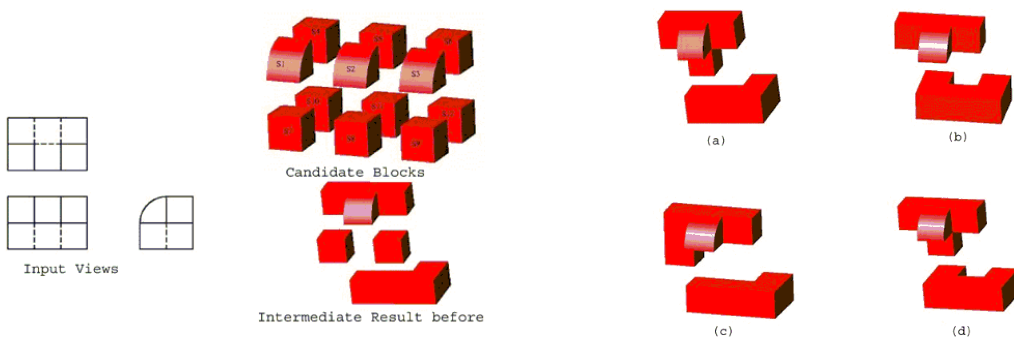

Fig. 9. Multiple solutions.

5.3 Discussion

All these examples are solved without having cross-sectional views and part labeling in the input views. The proposed method is able to construct 3D parts forming an assembly without any labeling of the parts in the assembly drawing and without any information about the size and shape of the parts. The method also does not require cross-sectional views. However, additional sectional or auxiliary views can be used to obtain a unique solution in cases where there are undecided external candidate blocks that would otherwise result in multiple solutions (as in Fig. 9).

The characteristic size of the problem to assess the computational complexity is the number of candidate blocks (n_{cb}) and the number of faces in the candidate blocks($n_{f\text{-}cb}$). The formation and classification of the candidate blocks are O($n_{f\text{-}cb}{}^{2}$). The computational effort in combining the candidate blocks and applying the continuity condition and mass condition depends on n_{cb} and is O($n_{cb}{}^{2}$) and O(n_{cb}) respectively.

5.4 Future Work

Present work can be made more efficient and its scope can be increased by handling sectional views in the assembly drawing. Additional information available in the sectional views can be used to better prune the list of candidate blocks. While the task of identifying elementary solids given sectional views has been addressed [3], handling sectional views in multi-part drawings requires further research. A formal proof of both existence of solution (given correct input is construction possible) and correctness of the solution has not been addressed thus far. Given the use of volumes to compose the solution, this work can also be extended to recognize features in multi-component drawings. This might be useful in tools evaluating assemblability or planning for assembly.

6. CONCLUSIONS

A volume based approach to construct 3D parts that constitute an assembly given the orthographic views of the assembly has been described. The method proposed does not require additional information such as part labeling or cross-sectional views as is required in current art. The method has been applied on typical assembly drawings with encouraging results. Further testing and handling sectional views in the assembly drawing are the focus of future work.

7. REFERENCES

[1] Dori, D., and Tombre, K., From engineering drawings to 3D CAD models: are we ready now? *Computer-Aided Design*, Vol. 27, No. 4, 1995, pp 243-254.

[2] Idesawa, M., A system to generate a solid figure from three views, *Bull. Of JSME*, Vol. 16, No. 2, 1973, pp 216-225.

[3] Jitendra Dimri and Gurumoorthy, B., Handling sectional views in volume-based approach to automatically construct 3D solid from 2D views, *Computer- Aided Design*, Vol. 37, 2005, pp. 485-495.

[4] Liu, S. X., Hu, S., Chen, Y. J. and Sun, J. G., Reconstruction of curved solids from engineering drawings. *Computer-Aided Design*, Vol. 33, No. 14, 2001, pp 1059-1072.

[5] Lysak, D. B., Devaux P. M. and Kasturi R. View Labeling for Automated Interpretation of Engineering Drawings," *Pattern Recognition*, Vol. 28, No. 3, 1995, pp. 393-407.

[6] Markowsky, G. and Wesley, M. A., Fleshing out Wire frames, *IBM Journal of Research and Development*, Vol. 24, No. 5, 1980, pp 582-597.

[7] Shum, S. S. P., Lau, W. S., Yuen, M. M. F., and Yu, K. M., Solid reconstruction from orthographic views using 2-stage extrusion, *Computer-Aided Design*, Vol. 33, No. 1, 2001, pp 91-102.

[8] Sourabh, S. and Gurumoorthy, B., Handling solids of revolution in volume based construction of solid models from orthographic views, *Transactions of ASME, Journal of Computing and Information Science in Engineering,* Vol. 3, No. 3, 2003, pp 250-259.

[9] Tanaka, M., Iwama,K., Atsushi, H., and Watanabe,T., Decomposition of a 2D assembly drawing into 3D part drawings, *Computer-Aided Design*, Vol. 30, No. 1, 1998, pp 37-46.

[10] Tanaka, M., Anthony, L., Kaneeda, T. and Hirooka, J., A single solution method for converting 2D assembly drawings to 3D part drawings, *Computer-Aided Design*, to appear.

[11] Wesley, M. A. and Markowsky, G., Fleshing out projections, *IBM Journal of Research and Development*, Vol. 25,No. 6, 1981, pp 934-954.