Automated Layer Decomposition for Additive/Subtractive Solid Freeform Fabrication

Y. C. Chang, J. M. Pinilla, J. H. Kao, J. Dong, K. Ramaswami, F. B. Prinz

micchang@Stanford.edu, jmp@Stanford.edu, jhkao@Stanford.edu, jdong@Stanford.edu, krishram@Stanford.edu, fbp@Stanford.edu

Department of Mechanical Engineering, Stanford University

Abstract

The new additive/subtractive technology (Shape Deposition Manufacturing) enhances Solid Freeform Fabrication (SFF) capability in producing near net-shape surface finish. This technology also builds parts in fewer layers compared with conventional layered manufacturing technology. However, to decompose a part into freeform layers usually requires expensive geometric computation. Also, to plan build sequences often requires human intervention because of the complicated spatial relationships among the freeform layers. At present decomposition and build sequence planning are both performed by experienced designers/users. In this paper, a novel decomposition approach based on surface splitting is proposed to facilitate computation and planning of the additive/subtractive SFF processes. The results shown in this paper are from models with 3D planar geometry. Continuous effort is devoted into extending and implementing this new approach for models with 3D freeform geometry.

1. Introduction

Traditional layered manufacturing techniques have a shorter cycle time of realizing a part from a CAD model compared with that of typical machine shop practices. Cycle time is shortened mainly because the technology employs a slicing strategy which greatly simplifies the planning task of building a part. Under this strategy, a given solid model is sliced horizontally into a set of planar layers, and these planar layers are then built one at a time from bottom up. In this case, planning the build sequence of a given model is no more than listing the sliced layers along the build direction, which is straightforward and can be automated easily.

On the other hand, certain issues limit the application of this technology. First, parts built by this technology generally show stair steps on the layer-to-layer boundaries. This lack of C2 continuity on the part surfaces is undesirable especially for parts made of ceramic materials, because each stair step can serve as a crack initiation site. In addition, the mechanical properties of parts will be affected by the bonding between layers. Clearly, reducing the number of layers by increasing the slicing interval can improve the mechanical properties. Nonetheless, the surface finish of the built parts may well become worse as the interval becomes larger. As a consequence, a trade-off has to be made case by case.

In contrast to traditional layered manufacturing technology, the new additive/subtractive SFF technology not only builds parts in fewer layers, but also achieves much finer surface finish. Instead of horizontal slicing, this new technology decomposes a solid model along a set of parting lines into freeform layers [3]. However, such decomposition usually requires expensive computation. In addition, the spatial relationship among these freeform layers is much more complicated than that among the planar layers. As a result, the build sequence can not be determined without carefully examining the geometry of each freeform layer. Therefore, the layer build sequence under this new technology is still completed mainly by experienced designers.

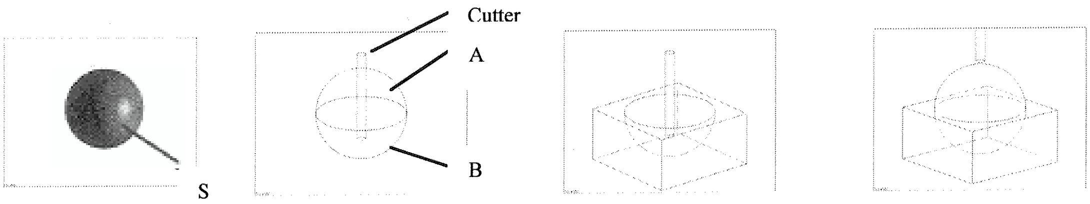
As shown later in this paper, a modified approach is developed for the additive/subtractive SFF technology in order to release designers from the time consuming task of decomposing a model and arranging build sequences. This new approach has been implemented successfully for models with 2.5D features. With limited testing results so far, it is shown to be able to work for models with 3D freeform features as well.

2. Surface Splitting

Currently decomposition of a given part is done in a "solid" level. That is, a given solid model is split into smaller manufacturable solid portions. An algorithm has been proposed [3] to automatically decompose a 3D part into a set of freeform layers for Shape Deposition Manufacturing (SDM) processes. In addition, all the feasible build sequences can be generated through a compact graph algorithm [6]. However, performing operations in a "solid" level usually requires expensive geometric computation.

To lower computation load, the feasibility of utilizing surfaces of a solid to partition the solid into manufacturable freeform layers for SDM processes is investigated. Since the shape of any part is represented completely by its surfaces, theoretically we can realize a solid object according to its surfaces. Simply stated, the object can be shaped out of a stock of material by performing machining operations on all surfaces of the object in such a sequence that each surface is machined without damaging any previously built surface. With this concept, each surface on the solid serves as a build unit which "partitions" the whole solid. There is no need for splitting the solid.

However, one critical manufacturing characteristic has to be considered before the concept can be applied to SDM processes. In SDM processes, the build direction of a part is pre-selected and remains unchanged for all machining stages of the part, which means the cutters always access the target machining areas from the top along the build direction. As a result, some surfaces can not be machined in a single step. As it can be seen in Fig. 1, we can not shape area B of surface S together with area A at one step without damaging A. But surface S can be fully shaped if we machine out area B first, deposit more part material on top of B, and then machine out area A. Under this reasoning, all surfaces of a part can be machined along the build direction as long as the surfaces containing both undercut and non-undercut regions are properly split into subsurfaces where each subsurface contains only either undercut or non-undercut region.



(a) (b) (c) (d)

Fig. 1 (a) Sphere S (b) Surfaces A and B on the sphere (c) Machine Surface B (d) Machine Surface A

In the following sections we will first define the terms used in this approach and then describe the procedures for surface splitting.

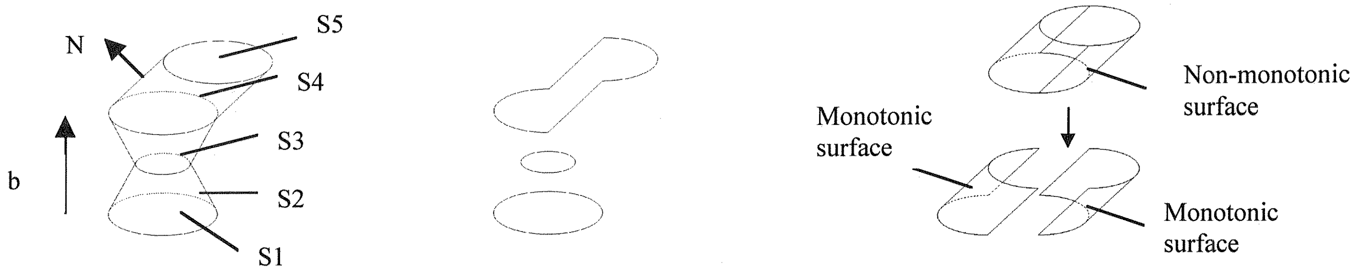
2.1 Definitions

Undercut, non-undercut, and non-monotonic surface

The type of a surface of a given solid model can be classified into three categories [3] as shown in Fig. 2(a):

Assume that N(u,v) is the normal vector at a point (u,v) on surface S, and b is the selected build direction, then

- S is an undercut surface if $N(u,v) \bullet b < 0$ for all (u,v) in S domain.
- S is a non-undercut surface if $N(u,v) \bullet b >= 0$ for all (u,v) in S domain.
- S is a non-monotonic surface if it contains both undercut and non-undercut portions.



(a) Undercut surfaces: S1, S3
Non-undercut surfaces: S2, S5
Non-monotonic surface: S4

(b) Silhouette curves

(c) Non-monotonic surface splitting

Fig .2 (a) Three surface types (b) Silhouette curves (c)Non-monotonic surface splitting

Silhouette curve

A silhouette curve is defined as the curve along which $N(u,v) \bullet b=0$. In other words, silhouette curves serve as the boundary between undercut and non-undercut portions of the surface. Thus one non-monotonic surface can be split into two or more monotonic (undercut or non-undercut) surfaces along the silhouette curves as shown in Fig. 2(c).

Overlapping surfaces

For any two surfaces, if there exist areas where a ray intersects both surfaces along the build direction, these two surfaces are "overlapping" surfaces.

Self-overlapping surface

If a ray intersects a monotonic surface more than once along the build direction, this surface is said to contain self-overlapping features. Fig. 3 illustrates the situation.

2.2 Procedure description

Five steps are taken to split surfaces of a solid model:

- Generate silhouette curves on all the non-monotonic surfaces of the given solid model.
- Split these non-monotonic surfaces into monotonic surfaces through the silhouette curves obtained in the previous step.
- Split self-overlap surfaces, if there exists any, into surfaces without self-overlap features [3].

- Split overlapping surfaces so that each resulting surface is either totally overlapped or totally not overlapped with other surfaces [3], then
- Mark each resulting surface as either an undercut or a non-undercut surface.
  A simple example is given in Fig. 4.

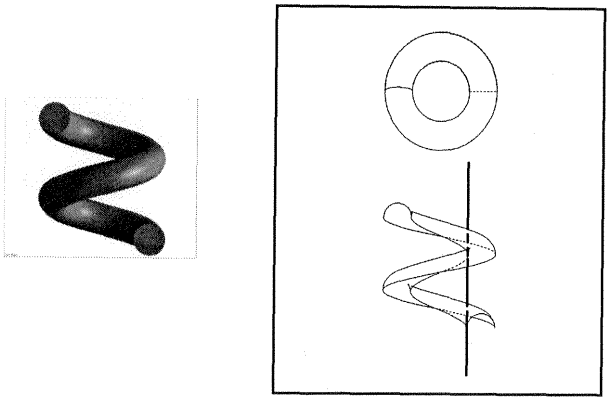


Fig.3 Self-overlapping surface

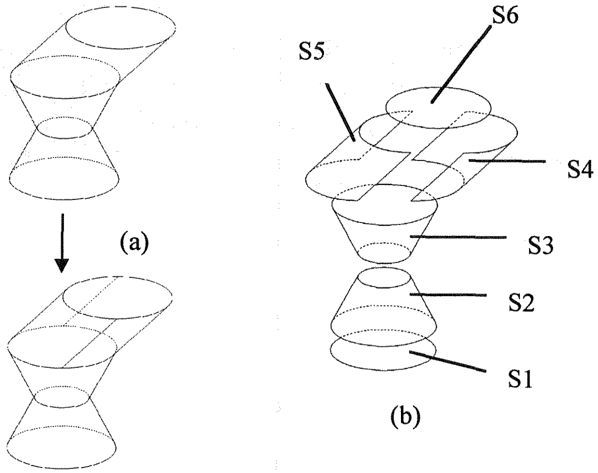


Fig.4 (a) Generate silhouette curves (b) Six surface compacts

After performing the five steps, the surfaces of the solid have been split into subsurfaces which serve as the basic build units.

3. Surface Compact Graph Construction

Up to this stage, all the surfaces of the target solid have been split into smaller subsurfaces such that each subsurface can be processed in one additive/subtractive build cycle. The target solid can now be produced by shaping these subsurfaces in certain build sequences. As mentioned earlier, a feasible build sequence is basically a spatial arrangement of subsurfaces which assures the cutters will not run into the already built subsurfaces while the target subsurface is being machined. In order to identify such sequences, the spatial relationships among all these freeform subsurfaces have to be examined and recorded. A special graph structure named Compact Graph [4] is utilized to record these spatial relationships because it not only offers a concise format of keeping these spatial information, but also provides useful algorithms to explore alternative build sequences afterwards. Finally, a "preliminary surface compact graph" is obtained after the subsurface entities and their spatial relationships are recorded through the Compact Graph structure. Again, the terminology and procedures are stated in the following sections.

3.1 Definitions

Surface compact

A surface compact represents a set of undercut or non-undercut surfaces which can be machined/processed together in one SDM cycle. In its simplest form, a surface compact is either an undercut surface or a non-undercut surface.

Surface compact graph

A graph in which each node represents one surface compact and each edge connecting two nodes indicates certain spatial relationship (either precedence or adjacency) between these two nodes.

Precedence relationship

One surface compact (or node) is "precedent" to another surface compact if there exists a ray which intersects both compacts along the build direction. Thus if surface compact A is "precedent" to surface compact B, compact A has to be built before compact B can be built. In other words, the precedence relationship between two compacts represents the build order between them.

Adjacency relationship

Two surface compacts (or nodes) are "adjacent" if they have common boundary.

3.2 Procedure description

Three steps are taken to construct a preliminary surface compact graph:

- One node is created in correspondence with one surface compact obtained through the surface splitting stage.
- Precedence relationship between each undercut surface compact and each non-undercut surface compact is examined. If there exists a precedence relationship between the two surface compacts, a directed edge is created between their associated nodes.
- The adjacency relationship between any two surface compacts (or nodes) is then examined and an undirected edge is created if there exists adjacency relationship between the two compacts.

After all surface compacts are processed through the three steps above, the spatial relationship among all surface compacts is preserved in the resulting preliminary surface compact graph, as shown in the example below. (Fig. 5)

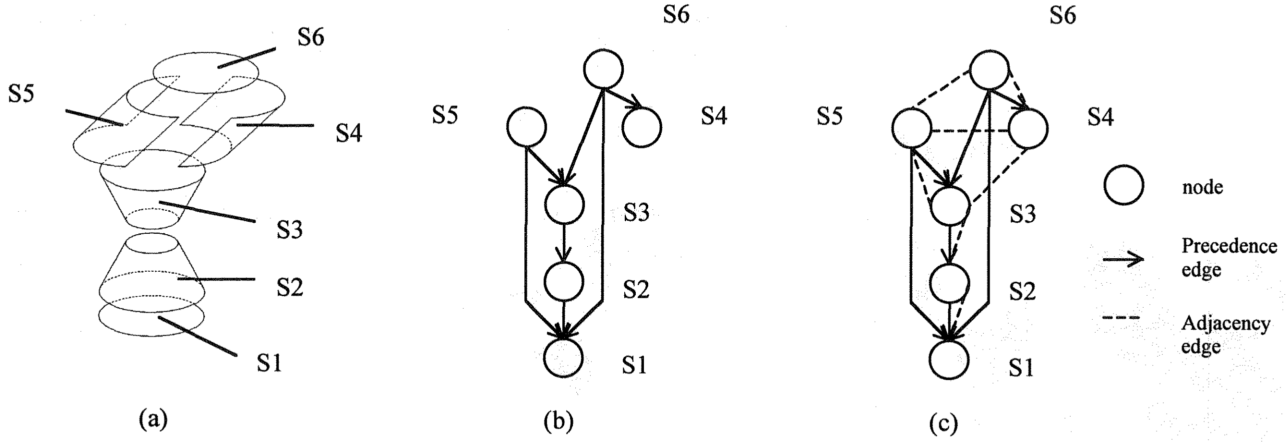


Fig. 5 (a) Surface compacts (b) Surface compact nodes and precedence edges (c) Preliminary surface compact graph

4. Surface Compact Graph Consolidation

The preliminary surface compact graph obtained above represents the spatial relationships between every two surface compacts. However, redundant spatial relationships may well exist in this preliminary graph. Because the computation cost of finding build sequences will be affected by these redundant spatial relationships, an intermediate step to consolidate the preliminary graph is taken before we search for the build sequences. At this stage, the preliminary surface compact graph is consolidated into a minimal surface compact graph.

4.1 Definitions

Redundant precedence edge

If the precedence relationship represented by a precedence edge can be also represented through a combination of other precedence edges, this precedence edge is called a "redundant" edge since it repeatedly expresses an existing precedence relationship.

Minimal surface compact graph

Through the two steps described below, a preliminary surface compact graph can be consolidated into a more concise graph named minimal surface compact graph. This minimal graph contains fewer nodes and edges but still describes the spatial relationship among the surfaces of a given solid model. Formally speaking, a minimal surface compact graph results when the nodes in a given preliminary graph can not be further merged and all the redundant precedence edges are removed. For instance, a solid with six surfaces can be represented by a minimal graph of four nodes and six edges rather than six nodes and thirteen edges as shown in Fig. 6.

4.2 Procedure description

Two steps are required to consolidate a preliminary surface compact graph into a minimal graph:

- Any two adjacent nodes can be merged into a new node if no precedence edge exists between them.
- Locate and remove all redundant precedence edges in the graph with certain checking algorithm [5].

After the adjacent node merge and redundant edge removal steps, a minimal surface compact graph containing the least number of nodes and edges is formed.

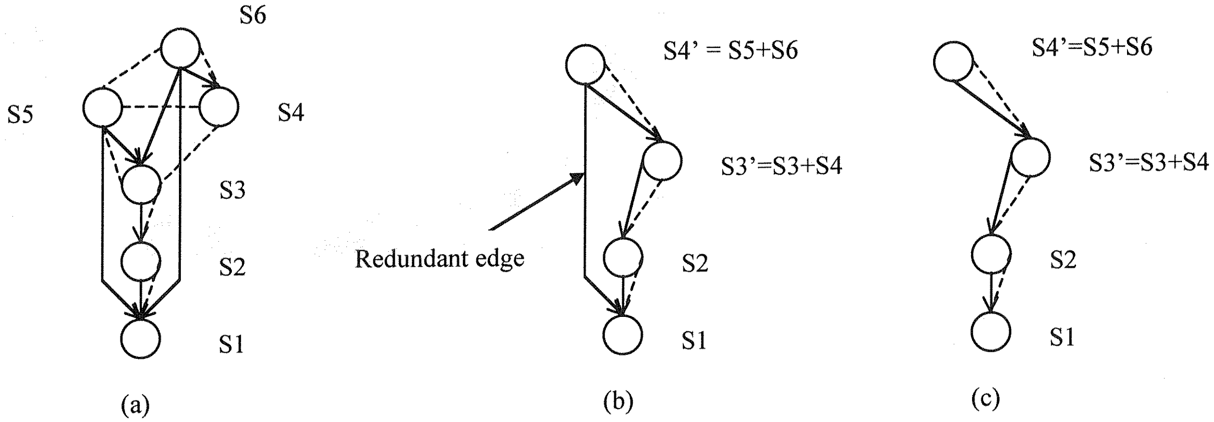


Fig.6 (a)Preliminary graph (b)Node merge and redundant edge removal (c)Minimal graph

5. Build Sequence Generation

As explained above, the minimal surface compact graph records all the build precedence relationships among nodes. Alternative build sequences can be explored by partitioning this minimal graph properly. Because the structure of a minimal surface compact graph is equivalent to that of a Compact Adjacency Graph (CAG) [4], the algorithms developed for CAG [6] can be also applied to a minimal surface compact graph. By employing these algorithms, the build sequence will be extracted automatically from the minimal graph as shown below. (Fig. 7)

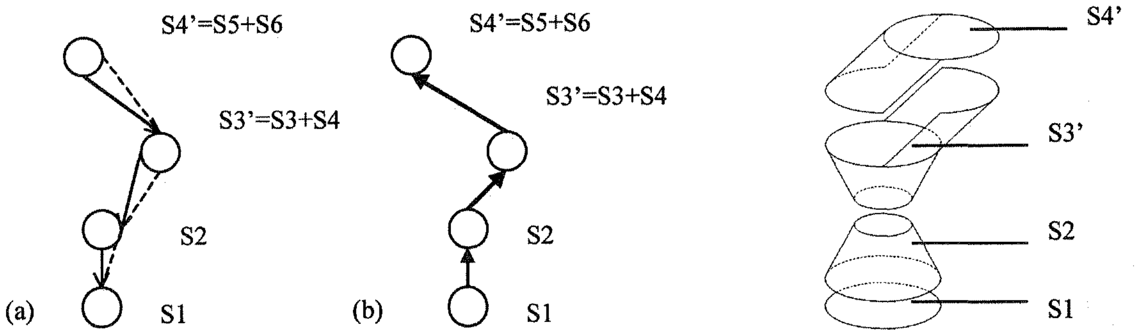
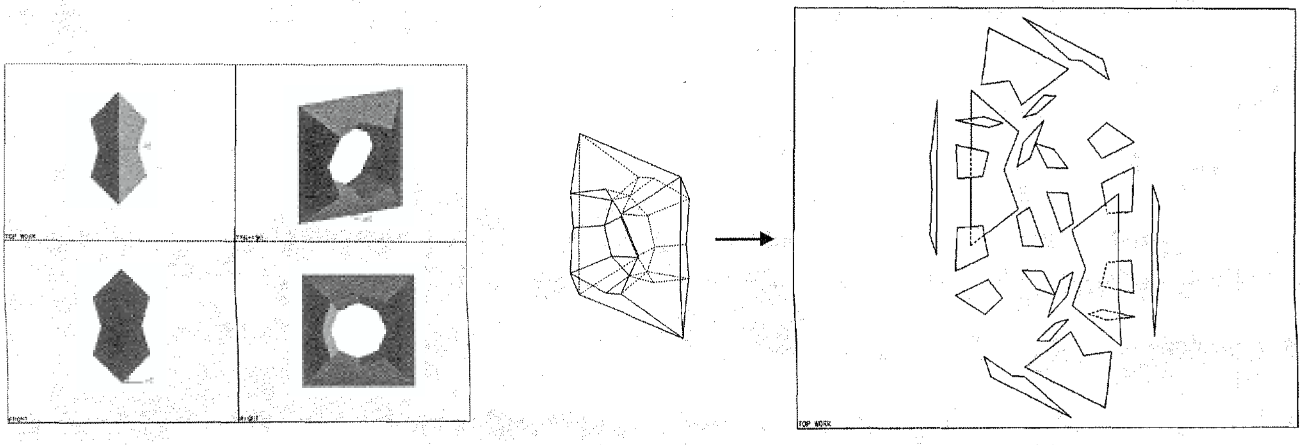


Fig.7 (a) Minimal graph (b) Build sequence: S1 -> S2 -> S3' -> S4'

6. Result

The example part has twenty-four surfaces originally. Ten undercut surfaces and fourteen non-undercut surfaces are identified after performing surface splitting on the part. The preliminary surface compact graph contains twenty-four nodes representing these twenty-four surfaces. After graph consolidation, the resulted minimal surface compact graph contains only four nodes, which suggests this part can be built in four layers as illustrated in Fig. 10. In short, we first deposit support material and machine surface compact S1, then deposit part material and machine S2. More support material is deposited on top of S2 and S3 area is machined. Finally, we deposit part material on top of S3 and machine S4 area.



(a)

(b)

Fig.9 (a) Example part (b) Decomposed surfaces

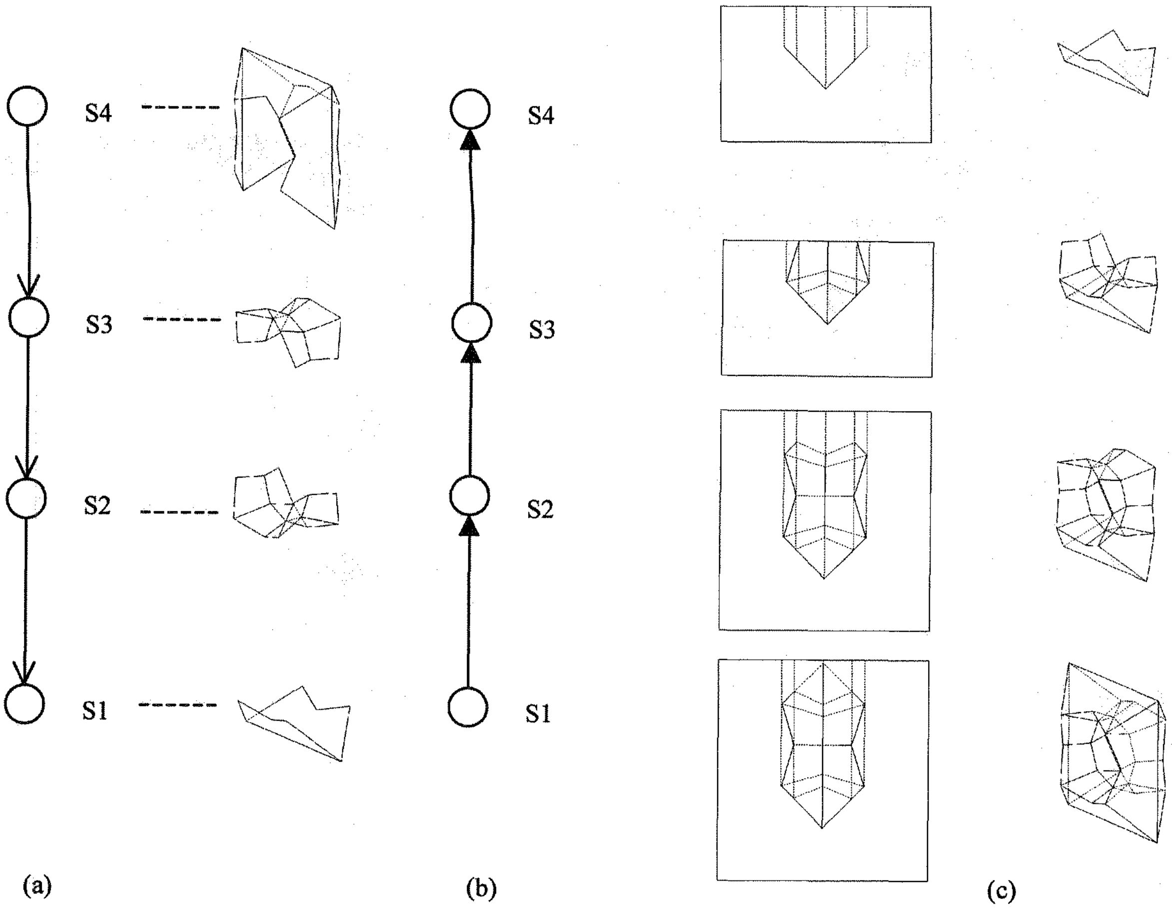


Fig. 10 (a) Minimal surface compact graph (b) Build sequence: S1->S2->S3->S4 (c) Build sequence illustration

7. Conclusions

Instead of decomposing a solid model horizontally, the proposed decomposition strategy splits and distinguishes the surfaces of the given solid into undercut and non-undercut surface compacts. Based on the surface compacts, the compact graph algorithms can then construct build sequences of the given model automatically. Compared with traditional planar slicing approach, this new surface compact approach results in fewer build layers for 2.5D and 3D freeform solid models.

8. Acknowledgement

We wish to acknowledge the financial support of DARPA under the 3D Reasoning project and Prof. C. Sequin of UC Berkeley for providing us the example part.

9. References

1. Harris L. Marcus and David L. Bourell. (1993) Solid Freeform Fabrication. Advanced materials and Processes, 144(3) pp28-35

2. Merz R., Prinz F. B., Ramaswami K., Terk M., and Weiss L. E. (1994) Shape Deposition Manufacturing. SFF symposium '94 Austin, Texas.

3. Ramaswami, K., Y. Yamaguchi, et al. (1997). Spatial Partitioning of Solids for Solid Freeform Fabrication. ACM/SIGGRPAH Symposium on Solid Modeling and Applications.

4. Pinilla, J. M., Kao J. H., et al. (1997). The Compact Graph Format: An Interchange Standard for Solid Freeform Fabrication. NIST workshop on Rapid Prototyping, Maryland, NIST.

5. Pinilla J. M., Kao J. H., Prinz F.B. (1998). Process Planning and Automation for Additive-Subtractive Solid Freeform Fabrication. Process planning and execution for SFF processes, SFF symposium '98 Austin, Texas.

6. Pinilla J. M., Kao J. H., Prinz F.B. (1999). Compact Graph Representation for Solid Freeform Fabrication (SFF). Submitted for publication.