Integration of a Solid Freeform Fabrication Process into a Feature-Based CAD System Environment

Yong S. Suh† and Michael J. Wozny‡

†Center for Advanced Technology in
Robotics, Automation and Manufacturing
Rensselaer Polytechnic Institute
Troy, NY 12180

‡Manufacturing Engineering Laboratory
National Institute of Standards and
Technology
Gaithersburg, MD, 20899

ABSTRACT

The design and implementation of a feature-based software system for the Solid Freeform Fabrication (SFF) process is discussed. For the SFF process to be an effective design tool in a manufacturing system, the tight integration of the process into a feature-based CAD system is desired. A feature set for determining the optimal part orientation is identified, and the procedure is described.

1 INTRODUCTION

1.1 SFF software: feature vs. geometric reasoning

To cope with the increasing market competition, the *concurrent engineering* (CE) concept is being adopted by many companies to reduce the cost and the cycle time for manufacturing quality parts. To build a successful CE system where designers and manufacturing experts work simultaneously, the appropriate management of the product information flow among the users is essential. The *product information* include high-level data such as design intent, part functionality and manufacturing processes, which traditional CAD systems cannot support. To support such high-level information beyond geometric data in the CE system, feature-based CAD systems have been introduced to associate engineering meaning to the shapes of the CAD model components. In these systems, users can manipulate the CAD models in terms of *features*, and software algorithms can simulate the human behavior by manipulating the high-level feature entities, as oppose to the low-level geometric reasoning processes with blind searching algorithms.

One of the primary application of the current SFF processes is to fabricate design prototypes for fast design verification. The process is identified to be a valuable tool in the CE environment because it can reduce the significant amount of design cycle time. Therefore, it is desirable that the SFF process software is fully integrated into the environment by taking a feature-based approach. As the process requires extensive geometric reasoning procedures that are time consuming and require complex algorithms, the feature-based approach is appropriate, and more intelligent processing is possible. Also, an algorithm can be easily customized in the feature-based approach by simply adding new features or modifying the existing features.

1.2 Design of a feature-based approach for the SFF

The feature-based systems have been classified into two big categories; feature extraction systems and design-with application feature systems.[hen94, hou93, kyp80, lu86]

In the feature extraction systems, features are extracted from pure CAD model data based on the predefined feature descriptions. However, it is very difficult, if not impossible, to implement a universal algorithm that extracts all types of features that are required by various applications. Another disadvantage is that most of the feature extraction algorithms can only find the features that exactly match the feature descriptions topologically and geometrically. The algorithms are not generally extendible to find general-shape features such as thin-walls, slender features and lateral protrusions. Most of the features that are required by the SFF processes are such general features in contrast to the machining features that have relatively concrete forms.

In the design-with SFF feature systems, a designer builds models using the features specific to the SFF process so that the features required by SFF software are embedded in the CAD model. As the primary usage of the SFF systems is for fabricating prototypes, however, this method is not practical because designers will focus on the final manufacturing processes such as injection molding and machining. This approach is not adequate for the CE system where multiple manufacturing systems should be supported.

As both traditional approaches are not suitable to the SFF process, a hybrid system design is desired to integrate the process into a feature-based system. The new system must support multiple feature representations, because the feature representation for the SFF processing will be different from the designer's feature representation. Therefore, the system software must provide a mechanism to extract customized features (SFF features) from the part model described by design features. Even though each user works on different views (different feature representations), any information provided by one of the users (such as dimensional constraints specified by a designer) must be shared among the users through features.[bro93]

1.3 Requirements for the feature-based system for the SFF process

The new system must satisfy the following functional requirements to support the SFF process in the CE environment.

- New features must be easily defined and extracted without system modification.
- General-shaped features must be defined and extracted.
- Design information such as dimensional constraints and surface finish must be supported.

A flexible feature-based system that supports the above requirements is designed and implemented. In section 2, this system architecture will be briefly discussed. In section 3, we will discuss a feature-based approach to the selection of the optimal part orientation.

2 SYSTEM ARCHITECTURE

The detail description of the system design will not be discussed in this paper. Instead, the basic idea of the system design and its features will be described very briefly.

2.1 Overview of the system

A part described by design features are converted to an internal feature representation in the system. A finite number of *fundamental features* (FF) and *fundamental spatial relations* (FSR) are defined, and the internal feature structure is formed by using them as building blocks. This is based on the assumption that any form feature can be decomposed to the FF's, and again, any form features can be represented by rearranging those FF's. Therefore, the application form features are extracted from the intermediate feature description converted from the design features using the FF's. The FSR's specify the spatial relationship between the FF's. Therefore, any design constraints specified by the designer can be applied to the application feature model. Using the FF's and FSR's, algorithms can be standardized without regard to the details of user-customized features.

2.2 Advantages over traditional feature extraction systems

A part with one feature representation can be transformed (extracted) to another through the two-step algorithm -- *Feature Shape Code Matching* step and *Feature extraction* step. Through these steps, designer's feature information is used when it's possible. The term, "*feature extraction*" will be used in this context in this paper. As the detail of the method is beyond the scope of this paper, only the advantages of this approach over traditional feature extraction systems are highlighted below.

- Design feature information can be used from other feature views.
- Features can be dynamically defined.
- A newly defined feature can be immediately extracted from the model without implementing the extraction algorithm specific to the feature.
- General-shape feature can be defined and extracted.

2.3 An example of feature extraction

In this system, features are defined in a declarative way as shown in Figure 2.1 in which a *through-slot* machining feature is defined. Writing the feature description in a text file using the predefined system keywords, the feature can be immediately extracted without coding additional program. Figure 2.2 shows a part described by several design features such as ribs, protrusions and depressions, and Figure 2.3 shows one of the three through-slot features extracted from the part as described in Figure 2.1.

```
face id 0 name wall-1 type HB-planar const { ^3 ^4 };
face id 1 name wall-2 type HB-planar const { ^3 ^5 };
face id 2 name bottom type HB-planar const { ^4 ^5 } open;
confront-paral id 3 entity ^0 ^1 hint value 15 +0.01 -0.01;
confront-angle id 4 entity ^0 ^2 value 90;
confront-angle id 5 entity ^1 ^2 value 90;
profile-open ^0 ^2 ^1 both-sides;
```

Figure 2.1 Declarative through-slot feature description

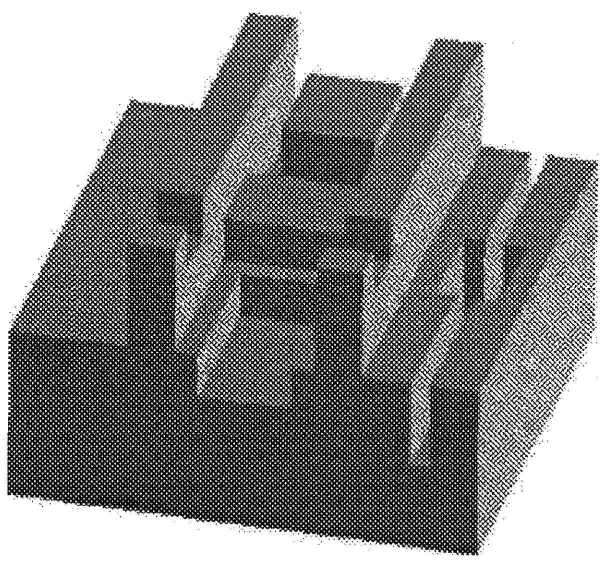

Figure 2.2 Example part

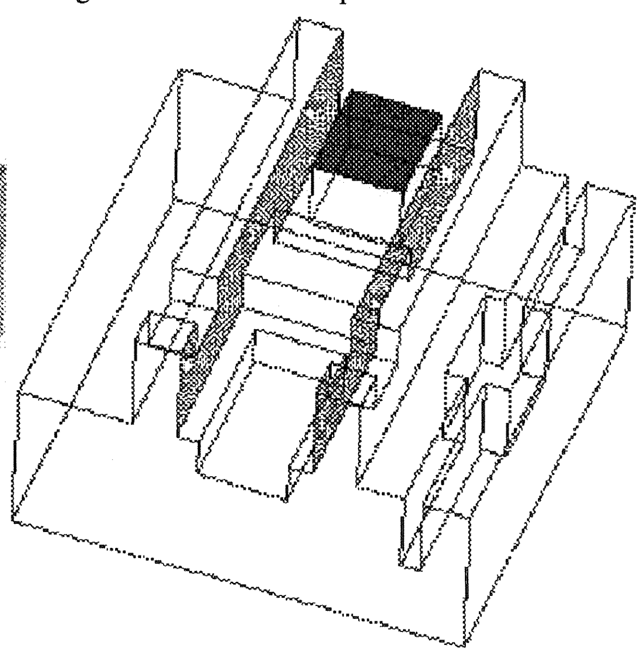

Figure 2.3 Extraction of a thru-slot feature

3 FEATURES FOR SOLID FREEFORM FABRICATION PROCESSES

The proposed system enables the SFF process to exploit the advantages of the feature-based CAD system in the concurrent engineering environment. An SFF process operator can define any features of their interests based on the characteristics of his/her SFF machine. The defined features are extracted from the designer's CAD model. In this section, we will discuss a feature-based approach to evaluation the optimal part building orientation as an example.

3.1 Features for determining the optimal part orientation

The part orientation within the building chamber of an SFF machine has a primary impact on the building time, part resolution and surface finish[jac92]. Comparing with the low-level geometric reasoning processes for evaluating the part orientation taken by several researchers[set94, kim94, sre94], the feature-based approach has the following advantages.

- **Faster processing**: As the features are directly extracted from the design features, many of the time-consuming geometric reasoning steps are reduced.
- **Intelligent processing**: The information supplied by designers can be used.
- **Flexibility**: The features can be easily added and removed for better results, customizing to the needs of a company.

In the following subsections, we will discuss the criteria for determining the optimal part orientation with the brief descriptions of relevant features. For the brevity of this paper, we will not discuss the detail of the algorithm for processing each type of features.

3.1.1 Trapped liquid

In the liquid-based SFF processes such as the SLA, trapped liquid is one of the major concerns in determining the part orientation. The trapped volume is the space (formed during the building process) within a part that hold the liquid isolated from the liquid in the vat.[1]

In this section, the features for the trapped volume are described borrowing the terminology of the machining features to represent the shapes of the features, as follows.

- Pocket
- Blind step
- Blind slot
- Hole
- General depressions

Once these trapped liquid features are extracted, the vertical height of the trapped liquid can be evaluated easily without extensive geometric reasoning. For each feature type, an algorithm for calculating the liquid height, $\mathcal{H}_i(\Theta)$, at an arbitrary orientation, $\Theta = \theta(\alpha,\beta,\gamma)$[2], can be defined easily. An example is shown in Figure 3.1 that shows a pocket feature at an arbitrary orientation. A local coordinate frame is attached to each feature, and its principal axes are represented as X',Y' and Z', and $\hat{\mathcal{X}}', \hat{\mathcal{Y}}', \hat{\mathcal{Z}}'$ denote the unit vectors of the local coordinate system with respect to the global coordinate system.

The liquid height, $\mathcal{H}_{pocket}$, of the pocket is calculated with a simple algorithm as follows.

$$\text{if } \hat{\mathcal{Z}}' \bullet \hat{\mathcal{Z}} \geq 0.0 \text{ then } \mathcal{H}_{pocket} = h_{\min}(\{\mathcal{V}_i^{top}\}) - h_{\min}(\{\mathcal{V}_j^{bottom}\}) \quad (3.1)$$

where

$h_{\min}(\{\mathcal{V}\})$: the minimum Z component value of the set of vertices, $\mathcal{V}$.

---

[1]In the powder-based processes such as SLS, the effect of the trapped volume is on the warpage of the part. If the open end of the trapped volume is downward, the heat is built up inside the trapped volume, which causes uneven temperature distribution.

[2]To specify an orientation of a solid in 3D space, three independent variables are required. The roll,pitch and yaw angles are used in this implementation.

$\mathcal{V}_i^{top}$, $\mathcal{V}_j^{bottom}$: the top and bottom vertices of the pocket respectively (i,j=1...4)

Similar algorithms can be defined for each feature type. A general algorithm has also been developed for evaluating the trapped liquid height of a general arbitrary depression feature. After evaluating each feature, a cost function, $\mathcal{F}(\Theta)$, can be calculated at a certain part orientation, Θ, as in Eq.(3.2).

$$\mathcal{F}(\Theta) = \sum_{i=1}^{n} \mathcal{H}_i(\Theta) / \mathcal{D}_{max} \tag{3.2}$$

where

n : number of the relevant features in the part

$\mathcal{H}_i(\Theta)$: sum of the trapped liquid height of every feature at Θ

$\mathcal{D}_{max}$: maximum diagonal size of the bounding box of the part at initial orientation

The cost function will be used in the nonlinear optimization procedure described in section 3.2.

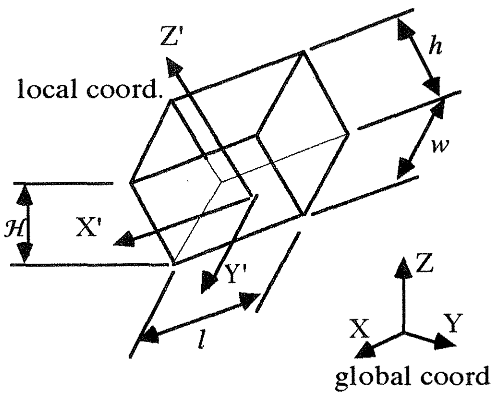

Figure 3.1 Evaluating trapped liquid height

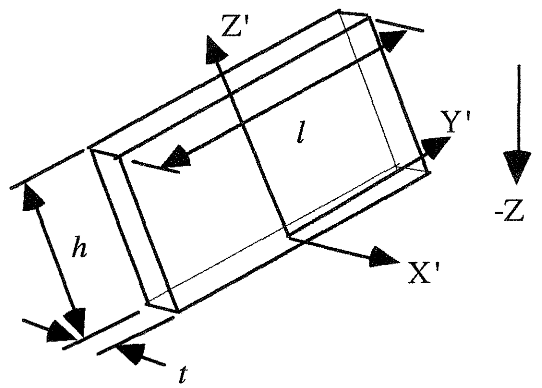

Figure 3.2 Evaluating support structure

3.1.2 Support structure

Support structures must be generated to hold the part during the fabrication in several processes. The complexity of the support structure of a part depends on the part orientation, and it must be as simple as possible so that they can be removed easily at the post processing step. The following features are extracted from the part to evaluate the support structure complexity.[3]

- Rib
- Boss
- Land
- Hole
- Pocket
- Slot
- Step
- General protrusion
- General depression

Given a part orientation at Θ, the area of the surfaces that require the support structure can be calculated according to the feature types. For example, as shown in Figure 3.2, the area, $\mathcal{A}_{rib}$, of the surfaces that need support structure can be calculated as follows for a rib feature.

$$\text{if } \hat{\mathcal{X}}' \bullet \hat{\mathcal{Z}} \neq 0.0 \text{ then } \mathcal{A}_x = l * h \tag{3.3}$$

$$\text{if } \hat{\mathcal{Y}}' \bullet \hat{\mathcal{Z}} \neq 0.0 \text{ then } \mathcal{A}_y = t * h \tag{3.4}$$

[3]The features can also be used in the automatic generation of the support structures. For example, a gusset-type support structure can be generated for a rib feature, while a honeycomb-shaped support structure is appropriate for a flat protrusion feature.

$$\text{if } \hat{\mathcal{Z}}' \bullet \hat{\mathcal{Z}} < 0.0 \text{ then } \mathcal{A}_z = t * l \tag{3.5}$$

$$\mathcal{A}_{rib} = \mathcal{A}_x + \mathcal{A}_y + \mathcal{A}_z \tag{3.6}$$

A general algorithm for an arbitrary feature has also been developed.

The cost function, $\mathcal{G}(\Theta)$, is calculated as follows.

$$\mathcal{G}(\Theta) = \sum_{i=1}^{n} \mathcal{A}_i(\Theta) \Big/ \mathcal{A}_{part} \tag{3.7}$$

where

n : the number of the relevant features in the part

$\mathcal{A}_i(\Theta)$: the area of the surfaces that requires surface finish in the i-th feature

$\mathcal{A}_{part}$: the overall area of the surfaces of the part

3.1.3 Part strength

In some processes, some features need to be appropriately oriented because of the strength problems.[sub94] An example of the features is a small diameter pin pointing up in the Z direction. The following features are identified:

- Thin wall
- Slender protrusion

The cost function, $\mathcal{J}(\Theta)$, returns either 1 or 0 as in Eq.(3.7).

$$\mathcal{J}(\Theta) = \begin{cases} 1 & \text{if properly oriented} \\ 0 & \text{otherwise} \end{cases} \tag{3.8}$$

3.1.4 Surface finish

The *'stair-stepping'* effect that is the trace of the layer can be minimized by properly orienting the part, focusing on the following features.

- Cylindrical hole
- Cylindrical boss
- Curved features

The cost function, $\mathcal{L}(\Theta)$, is defined as in Eq.(3.8)

$$\mathcal{L}(\Theta) = \sum_{i=1}^{n} \mathcal{A}_i(\Theta) / \mathcal{A}_{part} \tag{3.9}$$

where

$\mathcal{A}_i(\Theta)$: the area of the surfaces which are subject to the stair-stepping effect

$\mathcal{A}_{part}$: the overall surface area of the part

3.1.5 Dimensional accuracy

Most of the SFF processes have the characteristic that the dimensional accuracy in the vertical direction is inferior to that of the horizontal directions. Therefore, the part should be oriented so that the feature with a critical dimension lies on the horizontal X-Y plane. To build an accurate functional part, we must consider the tolerance information specified on the design features. The tolerance information can be easily extracted from the model using the proposed system, even though the design features are not explicitly visible from the SFF application side. The features with critical tolerance are extracted, and the part is oriented so that the feature lies on the X-Y plane.

The cost function, $\mathcal{I}(\Theta)$, is described in Eq.(3.10).

$$\mathcal{I}(\Theta) = \begin{cases} 1 & \text{if the feature is oriented appropriately} \\ 0 & \text{otherwise} \end{cases} \qquad (3.10)$$

3.1.6 Build time

The build time can be roughly estimated by the height of the part to be built. The cost function, $\mathcal{H}(\Theta)$, returns the normalized height of the part from the bounding box as follows.

$$\mathcal{H}(\Theta) = h(\Theta) / \mathcal{D}_{max} \qquad (3.11)$$

where

$h(\Theta)$: the height of the part at the orientation Θ

$\mathcal{D}_{max}$: the maximum diagonal dimension of the bounding box of the part

3.2 Evaluation of the optimal part orientation

The optimal part orientation is evaluated by setting up an equation as in Eq.(3.12), and solving it using a nonlinear optimization procedure.

$$\mathcal{M}in\ \mathcal{O}(\Theta) = w_1\mathcal{F}(\Theta) + w_2\mathcal{G}(\Theta) + w_3\mathcal{H}(\Theta) + w_4\mathcal{I}(\Theta) + w_5\mathcal{J}(\Theta) + w_6\mathcal{L}(\Theta) \qquad (3.12)$$

where

$w_1, w_2, \ldots, w_6$: user defined weighting factors ($0.0 \le w \le 1.0$)

$\mathcal{F}(\Theta), \mathcal{G}(\Theta), \mathcal{H}(\Theta), \mathcal{I}(\Theta), \mathcal{J}(\Theta), \mathcal{L}(\Theta)$: the cost functions of the criteria

The convergence to the solution depends on the initial orientation of the part. Therefore, the procedure is repeated several times with different initial orientations. Among the converged solutions, the orientation with the minimum design function value, $O(\Theta)_{min}$, is chosen as the optimal orientation of the part.

3.3 Examples

Two simple cases are tested as shown in Figure 3.3. For the simplicity and to verify the result intuitively, only the trapped volume, support structure and build time criteria are considered in the example. The figure shows the configurations at the optimal orientations found for the weighting factors as represented in the figure respectively. The configurations are selected from the converged orientations with six different initial orientations for each case.

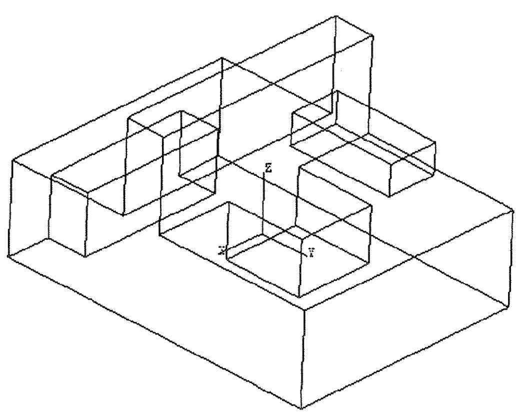

Figure 3.3 $w_1 = 0.3$ $w_2 = 0.6$ $w_3 = 0.1$

(Support > Trapped liquid > Build time)

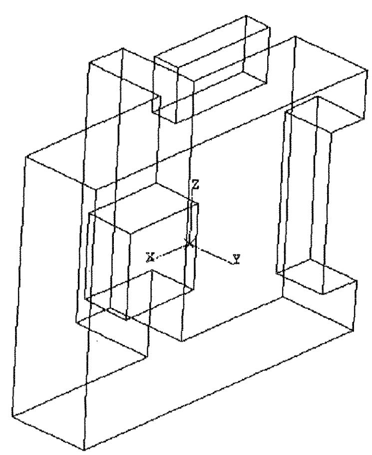

Figure 3.4 $w_1 = 0.6$ $w_2 = 0.3$ $w_3 = 0.1$

(Trapped liquid > Support > Build time)

4 SUMMARY

A feature-based approach to the SFF process software is discussed. In order for a SFF process to be integrated into a feature-based CAD system environment, the CAD system must provide a mechanism to map design features to the features that are meaningful to the SFF process. The feature-based approach simulates the behavior of human operators. In determining a part orientation, a human operator investigates the part and focuses on several outstanding features that are relevant to guidelines at a global point of view. By using the features, a faster and more reliable result is expected than traditional approaches based on geometric reasoning. The proposed system software architecture is implemented, and the SFF process software is integrated to the system to build a design-manufacturing system. The system is applied to the automatic determination of optimal part orientation process.

5 REFERENCES

[bro93] Willem F. Bronsvoort and Frederik W. Jansen, "Feature modelling and conversion - key concepts to concurrent engineering," Computers in Industry, 21:61-86, 1993

[dol94] A. Dolenc and I. Makela, "Slicing procedures for layered manufacturing techniques," Computer-Aided Design, 26(2):119-126, February 1994

[hen94] Mark R. Henderson, Gopal Srinath, Roger Stage, Kim Walker, and William Regli, "Boundary representation-based feature identification," Advances in Feature Based Manufacturing, chapter 2, pp.15-38, Elsevier Science B.V., 1994

[hou93] F.J.A.M. van Houten, O.W. Salomons and H.J.J. Kals, "Review of research in feature-based design," Journal of Manufacturing Systems, 12(2):113-132, 1993

[jac92] Paul F. Jacobs, "Rapid Prototyping & Manufacturing, Fundamentals of StereoLithography," Society of Manufacturing Engineers, 1992

[kim94] Kim, J. Y., Lee, K., and Park, J.C.,"Determination of optimal part orientation in stereolithographic rapid prototyping," preprint, July 1994

[kyp80] L. K. Kyprianou, "Shape classification in computer aided design," PhD thesis, Computer Science Laboratory, Cambridge University, 1980

[lub86] S. C. Luby, J. R. Dixon, and M. K. Simmons, "Creating and using a features data base," Computers in Mechanical Engineering," 25-33, November 1986

[set94] Seth Allen and Deba Dutta, "On the computation of part orientation using support structures in layered manufacturing," Proceedings of Solid Freeform Fabrication Symposium, pp.259-269,Austin, TX, August 1994

[sre94] Sreeram, P., and Dutta, D., "Determination of Optimal orientation based on variable slicing thickness in layered manufacturing," Technical Report UM-MEAM-TR-94-14, Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI, July 1994

[sub94] P. Kamatchi Subramanian, N. K. Vail, J.W. Barlow and H.L. Marcus, "Anisotropy in alumina processed by SLS", Proceedings of Solid Freeform Fabrication Symposium, pp.330-338, Austin, TX, August 1994

[suh94] Yong Seok Suh and Michael J. Wozny, "Adaptive slicing of solid freeform fabrication processes," Proceedings of Solid Freeform Fabrication Symposium, pp.404-411, Austin, TX, August 1994