ADAPTIVE LOCAL SLICING IN STRATOCONCEPTION BY USING CRITICAL POINTS

Houtmann, Y. *; Delebecque, B. *; Barlier, C. **
* Cirtes, 29 bis rue d'Hellieule - 88100 Saint-Dié-des-Vosges, France
** ERMeP, GIP-InSIC, 27 rue d'Hellieule - 88100 Saint-Dié-des-Vosges
Email : yves.houtmann@laposte.net

Abstract:
The aim of this paper is to present the use of critical points for adaptive local slicing in rapid prototyping, especially Stratoconception. First, we introduce the context of use of critical points in rapid prototyping. Then, we define them on surfaces and triangular meshings and extend this definition to other geometric entities, such as edges, facets and faces. Finally, we use the critical points to optimise usual slicing. We compare this new slicing with the usual one on a few models manufactured by Stratoconception.

Key Words: Rapid prototyping, Decomposition, Critical points, Adaptive slicing, Stratoconception

1. CONTEXT
Rapid prototyping allows the physical manufacturing of a numerical 3D model, with time saving compared to the classical manufacturing process. Rapid prototyping processes are used to make models layer by layer, each of them being manufactured in 2D [4].

Visually, critical points look like basin's bottoms, tops of hill and mountain's passes (Fig. 1). If we slice the model at their heights, the pieces of the model obtained can be made more efficiently with adapted parameters (accuracy, speed, slicing direction...). For example, each of the hills of a massif can be isolated by setting a cutting plane at the height of the basin's bottom of the massif.

In this paper, we present an optimization of rapid prototyping manufacturing, especially Stratoconception, by using critical points.

2. DEFINITION OF CRITICAL POINTS

2.1 Different definitions on surfaces
The use of critical points is a classic tool in rapid prototyping [11], they might be defined in several ways.

Definition by topologic variations of the intersecting curves
Let $\Sigma$ be a surface of Euclidian space $E^3$.
Let $O$ be a point of the space, called origin.

Let $\Pi_i$ be a plane with normal $\tau$, such that $\forall M_i \in \Pi_i$, $\overline{OM_i} \cdot \tau = h_i$. Thus we call $h_i$ the height of the plan $\Pi_i$ thanks to the slicing direction $\tau$ with respect to the origin $O$.

The result of the slicing of $\Sigma$ by a plane $\Pi_i$ is a set of contours $C_i$.

When computing the intersection of a plane with a surface, the change of topology from the set of curves $C_i$ to the set $C_{i+1}$ is the expression of the presence of a critical point:
- if there is a local maximum at the height $h_i$, a curve disappears (Figure 1, a)
- if there is a local minimum at the height $h_i$, a curve appears (Figure 1, b)
if there is a saddle point, two curves disappear at the same times as two others appear (Figure 1, c)

**Definition by looking at the height of the neighbourhood**

Let $\zeta$ be the neighbourhood of the point $p_i$.

Thus, we obtain:

- $p_i$ is a local maximum $\iff \forall v \in \zeta, Ov \cdot r < h_i$
- $p_i$ is a local minimum $\iff \forall v \in \zeta, Ov \cdot r > h_i$
- $p_i$ is a saddle point $\iff$ there is a partition of $\zeta$ in at least four distinct parts so that they are alternatively below and above $h_i$.

![Figure 1](image)

**Connection with the Gauss curvature**

The Gauss curvature [5] allows us to detect critical points of a model without considering a specific direction. The literature contains several definitions and computations techniques of the Gauss curvature, particularly [2, 12, 13]. There is a relation between critical points and points with extreme curvature. For example, hyperbolic points, with negative curvature, correspond to saddle points. Within the framework of this work, we don't need to compute Gauss curvature to detect critical points because we have to consider a given direction. Consequently, points with extreme curvature are more numerous than useful critical points. Also if we use points with extreme curvature to set automatically slicing plane. This requires an additional filtering, which is time consuming and prone to approximation.
2.2 Adaptation of the definition on STL mesh

The STL format, standing for STereoLithography, is the de facto standard used to represent models in the rapid prototyping world. An STL model can be considered as a polyhedron of $E^3$ with triangular faces.

Classically, an STL model is defined as a finite set of triangles, called facets. A facet comprises three non-aligned points called vertices, and three edges joining vertices by pairs.

We have to adapt the second definition of critical points on an STL mesh. Thus, we define the idea of neighbourhood of a vertex $p_i$ on an STL mesh. The neighbours of a vertex $p_i$ are the vertices of the mesh connected to $p_i$ through one edge.

We consider that the height $h_p$ of a point $p$ is its $z$ component in a three-dimensional axes system. So, the slicing direction $\tau$ and the unit vector $z$ merge together.

Consequently, we obtain:

- $p_i$ is a local maximum $\iff$ the whole of its neighbours is below $p_i$ (Figure 2, a)
- $p_i$ is a local minimum $\iff$ all its neighbours are upon $p_i$ (Figure 2, b)
- $p_i$ is a saddle $\iff$ we count an even number, more or equal to 4, of edges connecting $p_i$ to neighbours alternating above and below it (Figure 2, c)

![Figure 2: Three types of critical points on an STL mesh: local maximum (a), local minimum (b) and saddle (c).](image)

There is a problem with STL mesh. Indeed it results from the numerical chain and can be disturbed for various reasons (in particular by digitizing). Consequently, the mesh is full of insignificant critical points, e.g. not very deep basin's bottom or not very high top of hill. It is then necessary to filter the critical points we find in order to preserve only the most important. Several solutions are possible, some of them relying on the analysis of the stability of the critical points by using persistence diagrams [1, 6, 10] or interval persistence [9].

In our work, we set up a few filters looking at minimal variations between critical points and the area of the cutting contour obtained by slicing the model at a height given by a critical point.
3. CONTRIBUTION : EXTENSION OF CRITICAL POINTS CONCEPT

We try to cut the model orthogonally to the slicing direction at the height of critical points. For this, the traditional definition of critical points on STL model is quickly limited.

Figure 3 shows this problem: if you look at the plane edges, i.e. which have their two vertices at the same height, each vertex individually is not critical, because its neighbours are above or at the same height. On the other hand, if we regard the whole of the edges as a vertice, this one is a basin’s bottom, since its neighbours are all above it.

To extend the concept of critical point to edges, facet and faces (group of planar edges and facets, neighbours ones of the others), we need to identify the possible grouping of planar entities (Figure 4, red zone). Various methods exist, among which the variational shape approximation [6]. More simply, we try to proceed by propagation starting from a facet orthogonal to the slicing direction. We then gather vertices which are neighbours of the planar zone, i.e. vertices joined to vertices of the planar zone and not belonging to it (Figure 4, blue vertices). Then, on this whole of vertices, we use the same indexing as for a critical point on an STL mesh (§ 2).
There are two types of intersecting curves obtained by slicing the model at a given height: outer and inner. An outer contour includes one or more inner contours. When a critical zone is limited by an outer contour including inner contours, a specific classification for this zone must be set (Table I).

Critical points are a classic tool in rapid prototyping [11], they might be defined in several ways. We present the one using the notion of neighbourhood.

4. APPLICATION FOR LOCAL ADAPTIVE SLICING IN STRATOCONCEPTION

Stratoconception is a Rapid Prototyping process with solid/solid layers. It consists in the decomposition of the piece by calculating a set of elementary layers called "strata" and by placing reinforcing pieces and inserts in the strata. The elementary layers are displayed and manufactured by rapid micromilling or laser-cutting. The strata are then assembled with inserts to rebuild the final object (Figure 6).
4.1 Different types of slicing: constant, adaptive and local adaptive

**Constant slicing**
It is the slicing classically used in rapid prototyping. We cut the model with a constant step $\xi$ chosen by the user (Fig. 7, a). That is the cutting planes are regularly spaced without taking the model into account.

**Adaptive slicing**
This slicing consists in the adaptation of the slicing step $\xi$ according to the slope of the model at a given height. For this, we look at the slope of the facets in comparison with the slicing direction $\tau$. This slicing allows to save slicing when the model is vertical and to add ones when needed (Figure 7, b). Adaptive slicing is commonly used in Stratoconception [8].

**Local adaptive slicing**
If there are several independent contours after a slicing at a given height, it is due to the model which is composed by several parts at this height. Adaptive local slicing consists in adapting the slicing step to each identified part of the model individually and not to its whole (e.g. the cylinder and the half sphere Fig. 7, c) [14, 15].

The birth and death of these entities can be underlined by critical points. Therefore they can help to make use adaptive local slicing (Algorithm 1).
Algorithm 1. Adaptive local slicing in Stratoconception by using critical elements:

1. Detection of the plane elements (facets and edges) orthogonal to the slicing direction
2. Grouping of these elements in zones by propagation with a planar tolerance
3. For each identified zone (composed by one or more elements)
   (a) Computation of the type of the zone regarding its neighbourhood
   (b) If it is a critical zone (local minimum or saddle)
      i. Mark the elements of the zone (facets, edges and vertices) as critical
      ii. Set a cutting plane at this height

4. For each vertice of the STL
   (a) Computation of its type regarding its neighbourhood
   (b) If it is a critical vertice (local minimum or saddle)
      i. Mark the vertice as critical
      ii. Set a cutting plane at this height

5. Filtering of the planes
   (a) If the difference in height between cutting plane is below a threshold
      i. Keep the plane with the maximum cutting contours area

6. Compute the slicing of the model by taking into account the critical points and adapting
   the slicing step to the parts of the model isolated by the critical slicing (e.g. the cylinder
   and the half sphere Figure 7, c)

4.2 Examples and savings

We use seven models (Figure 8) to show the results of local adaptive slicing by using critical
points and to quantify savings in term of the length of manufacturing path. These models are
either mechanical, i.e. made with simple geometric entities (planes, cylinders, drilling...) and
eventually with symmetries, or artistic, which are free form models.
Figure 8: Models for tests with cutting planes automatically set by local adaptive slicing algorithm.

The main advantage of the use of local adaptive slicing is the saving of 25 to 92% in manufacturing path length against usual constant step slicing (Table I and Figure 9).

Regarding adaptive slicing, the saving is not so marked. It depends mainly on the model's geometry. The most important saving is 51% high with the model Figure 8, a. The reason is that after critical slicing the cylinder becomes independent from the half sphere and can therefore be made with one cutting contour only.

Table I: Manufacturing path length with different slicings.

<table>
<thead>
<tr>
<th>Model</th>
<th>Size (mm)</th>
<th>Facets</th>
<th>Slicing accuracy (mm)</th>
<th>Method of slicing</th>
<th>Manufacturing path length (mm)</th>
<th>Savings towards constant slicing (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Piéce type</td>
<td>84<em>44</em>60</td>
<td>117</td>
<td>0.50</td>
<td>Constant</td>
<td>64 117</td>
<td>83.58</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Adaptive</td>
<td>10 529</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Local adaptive</td>
<td>5 129</td>
<td>92.00</td>
</tr>
<tr>
<td>Support</td>
<td>116<em>135</em>97</td>
<td>152</td>
<td>0.50</td>
<td>Constant</td>
<td>197 152</td>
<td>15.09</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Adaptive</td>
<td>167 405</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Local adaptive</td>
<td>133 880</td>
<td>32.09</td>
</tr>
<tr>
<td>Eurocast</td>
<td>85<em>193</em>72</td>
<td>405</td>
<td>0.50</td>
<td>Constant</td>
<td>259 546</td>
<td>46.04</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Adaptive</td>
<td>140 057</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Local adaptive</td>
<td>129 733</td>
<td>50.01</td>
</tr>
<tr>
<td>Horse</td>
<td>84<em>183</em>153</td>
<td>238</td>
<td>0.50</td>
<td>Constant</td>
<td>292 027</td>
<td>13.20</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Adaptive</td>
<td>253 490</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Local adaptive</td>
<td>162 050</td>
<td>44.51</td>
</tr>
<tr>
<td>Standford bunny</td>
<td>65<em>48</em>61</td>
<td>405</td>
<td>0.50</td>
<td>Constant</td>
<td>56 181</td>
<td>13.76</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Adaptive</td>
<td>43 278</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Local adaptive</td>
<td>38 396</td>
<td>27.47</td>
</tr>
<tr>
<td>Hippo</td>
<td>71<em>201</em>102</td>
<td>308</td>
<td>0.50</td>
<td>Constant</td>
<td>208 030</td>
<td>28.71</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Adaptive</td>
<td>148 293</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Local adaptive</td>
<td>140 309</td>
<td>32.55</td>
</tr>
<tr>
<td>Pharao</td>
<td>60<em>145</em>51</td>
<td>308</td>
<td>0.50</td>
<td>Constant</td>
<td>72 366</td>
<td>19.13</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Adaptive</td>
<td>58 524</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Local adaptive</td>
<td>53 966</td>
<td>25.43</td>
</tr>
</tbody>
</table>

66
5. CONCLUSION
The critical points presented and redefined above are useful and efficient when applied to rapid prototyping, especially with Stratoconception. Thanks to them, we can easily and robustly make use of local adaptive slicing on STL models.

REFERENCES