Advances in Production Engineering & Management 4 (2009) 1-2, 59-68
ISSN 1854-6250 Scientific paper

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 Σ be a surface of Euclidian space E^3.

Let O be a point of the space, called origin.

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

The result of the slicing of Σ by a plane Π_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 ζ_i be the neighbourhood of the point p_i.

Thus, we obtain :

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

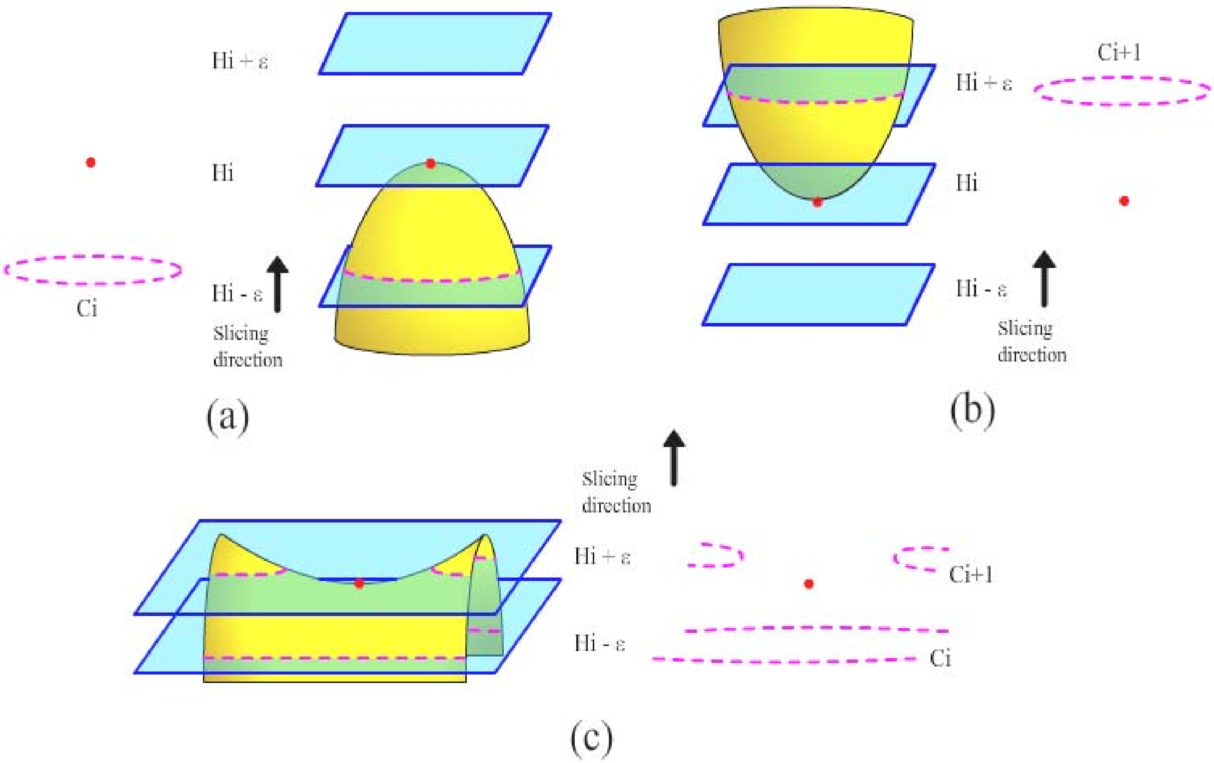

Figure 1: Three types of critical points : local maximum (a), local minimum (b) and saddle (c).

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 ζ_i on an STL mesh. The neighbours of a vertice p_i are the vertices of the mesh connected to p_i through one edge.

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

Consequently, we obtain :

- p_i is a local maximum $\Leftrightarrow$ the whole of its neighbours is below p_i (Figure 2, a)
- p_i is a local minimum $\Leftrightarrow$ all its neighbours are upon p_i (Figure 2, b)
- p_i is a saddle $\Leftrightarrow$ 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)

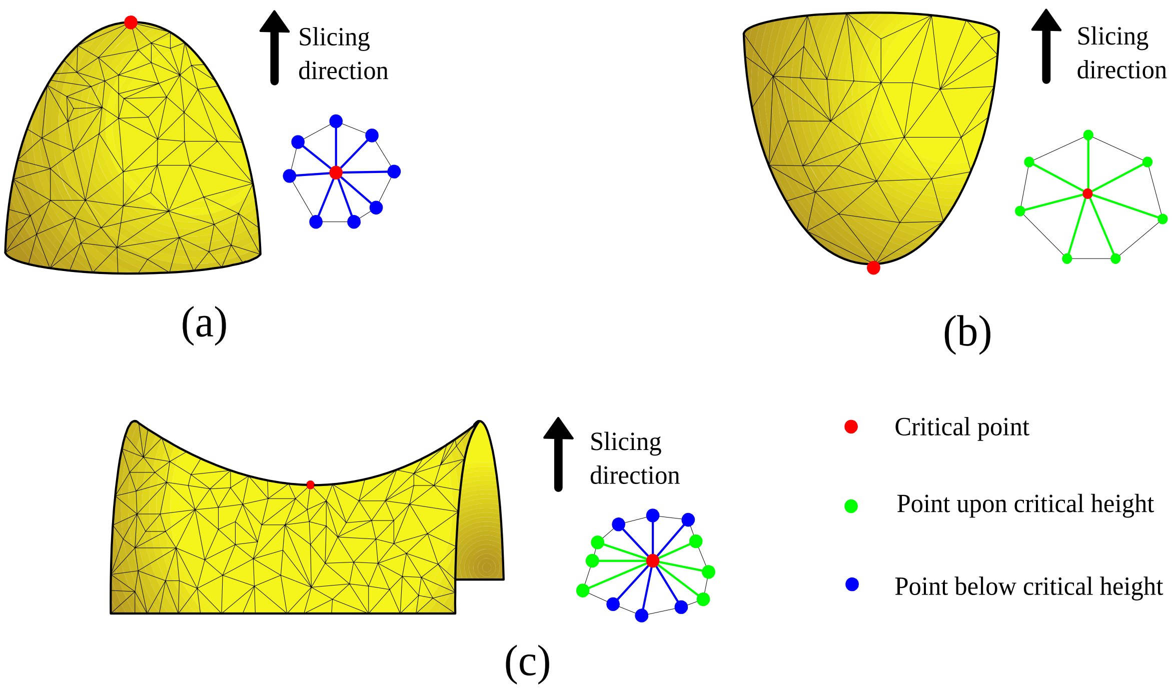

Figure 2: Three types of critical points on an STL mesh : local maximum (a), local minimum (b) and saddle (c).

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 vertice 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.

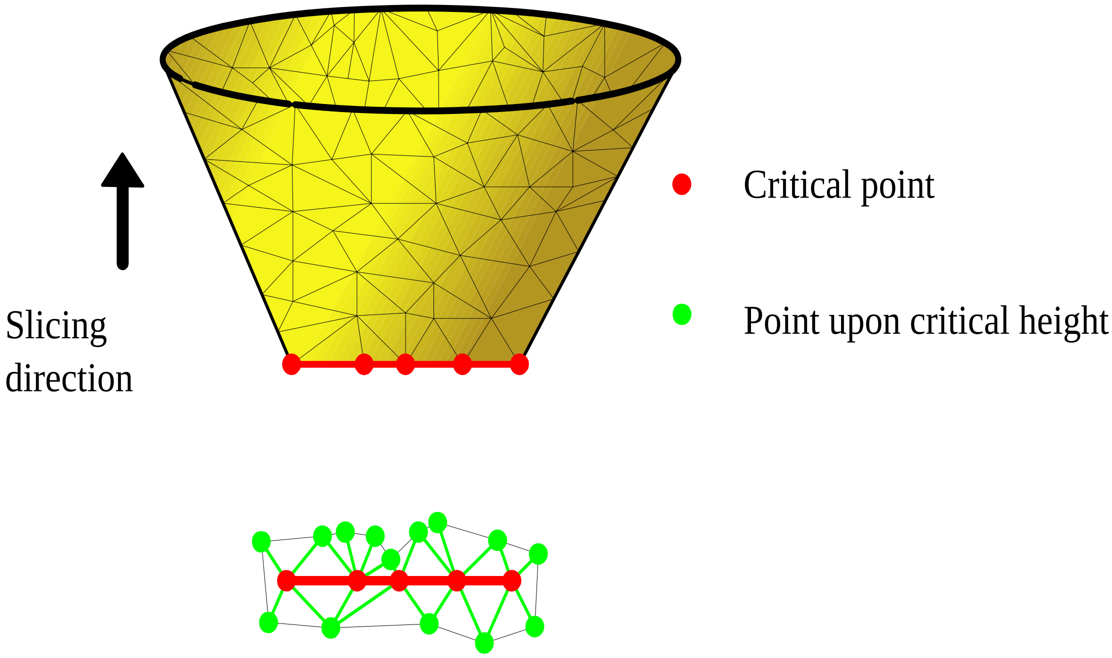

Figure 3: Extension of critical point concept to a whole of edges.

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).

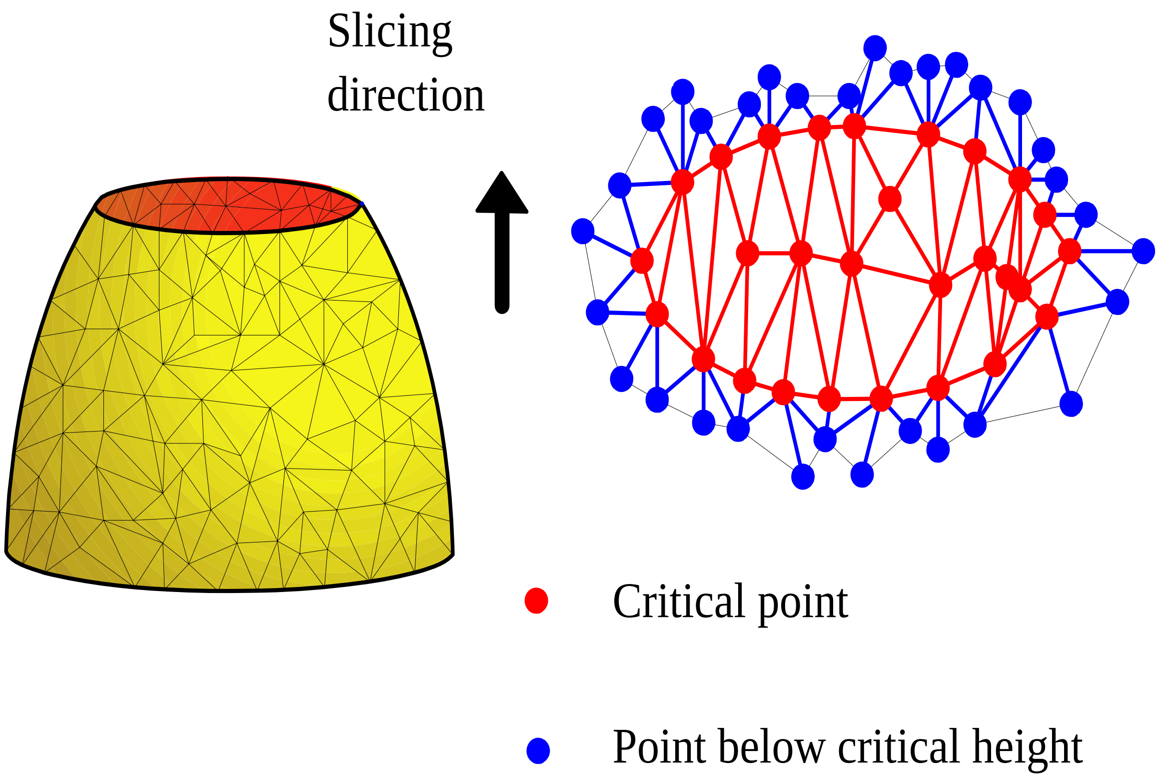

Figure 4: Extension of local maximum definition to plateau.

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 define 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).

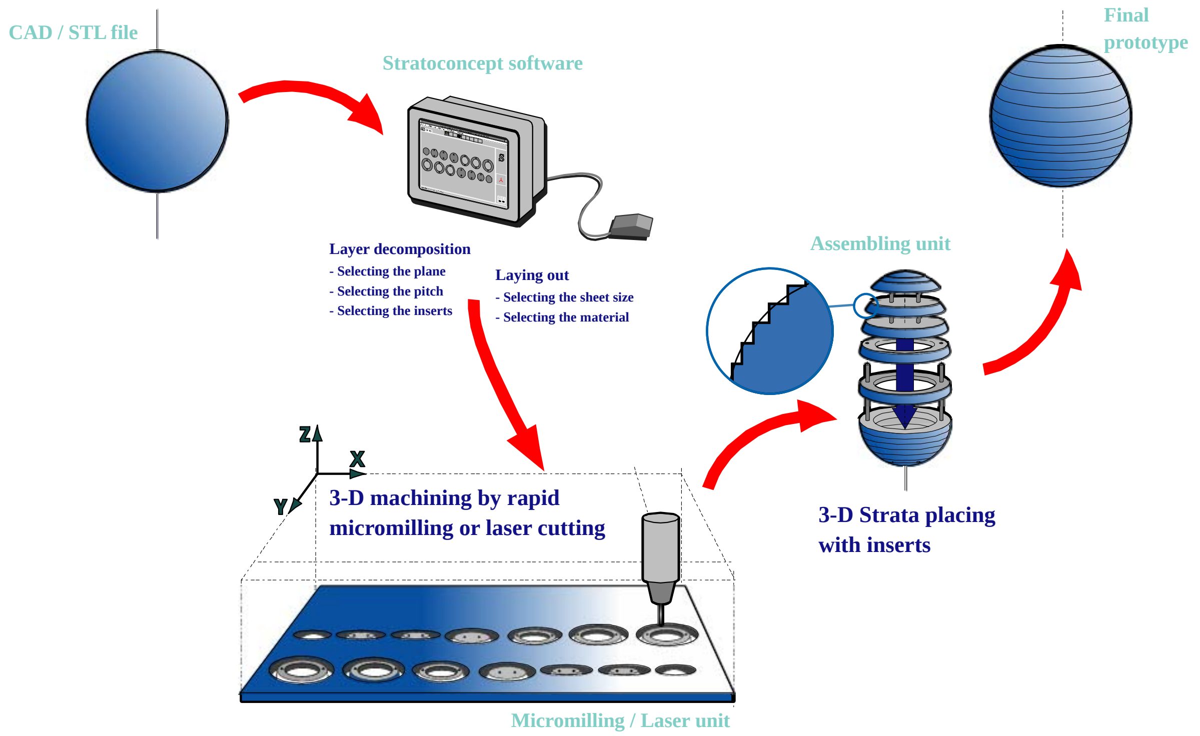

Figure 6: Principles of Stratoconception [3].

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 ς choosen 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 ς 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 τ. 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 underline by critical points. Therefore they can help to make use adaptive local slicing (Algorithm 1).

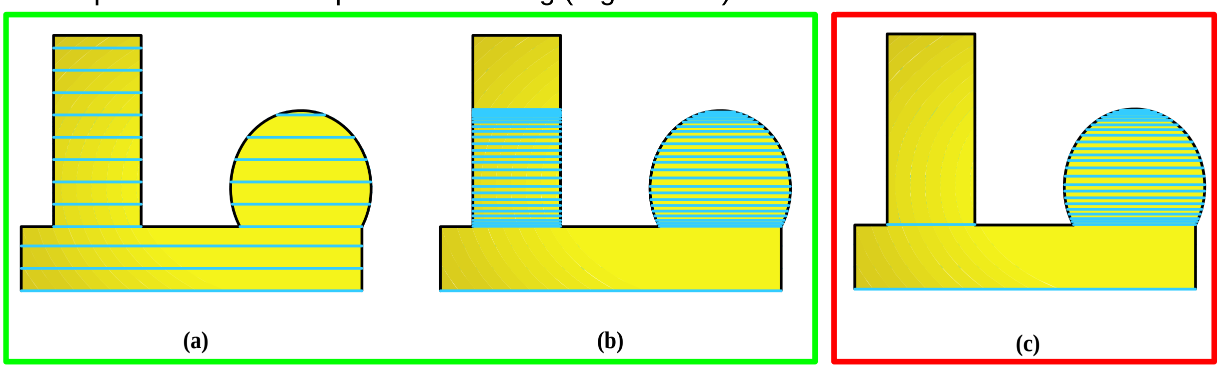

Figure 7: Different types of slicing: constant (a), adaptive (b) and local adaptive (c).

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.

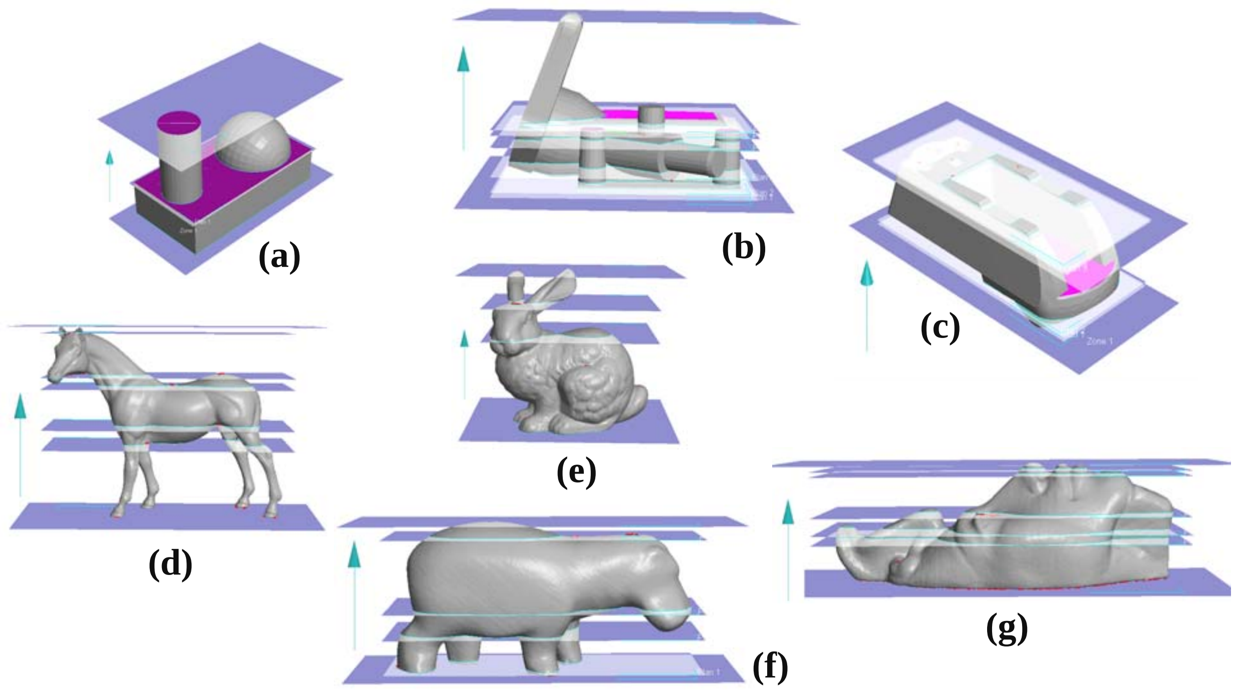

Figure 8: Models for tests with cutting planes automatically set by local adaptive slicing algorithm.

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.

	Model Size (mm) Facets	**Slicing accuracy (mm)**	**Method of slicing**	**Manufacturing path length (mm)**	**Savings towards constant slicing (%)**
	Piéce type		Constant	64 117	
(a)	84*44*60		Adaptive	10 528	83,58
	864	0,50	Local adaptive	5 129	92,00
	Support		Constant	197 152	
(b)	116*135*97		Adaptive	167 405	15,09
	3 392	0,50	Local adaptive	133 880	32,09
	Eurocast		Constant	259 546	
(c)	85*193*72		Adaptive	140 057	46,04
	10 770	0,50	Local adaptive	129 733	50,01
	Horse		Constant	292 027	
(d)	84*183*153		Adaptive	253 490	13,20
	39 238	0,50	Local adaptive	162 050	44,51
	Standford bunny		Constant	50 181	
(e)	65*46*61		Adaptive	43 278	13,76
	64 126	0,50	Local adaptive	36 396	27,47
	Hippo		Constant	208 030	
(f)	71*201*102		Adaptive	148 293	28,71
	154 986	0,50	Local adaptive	140 309	32,55
	Pharaon		Constant	72 366	
(g)	60*145*51		Adaptive	58 524	19,13
	166 372	0,50	Local adaptive	53 966	25,43

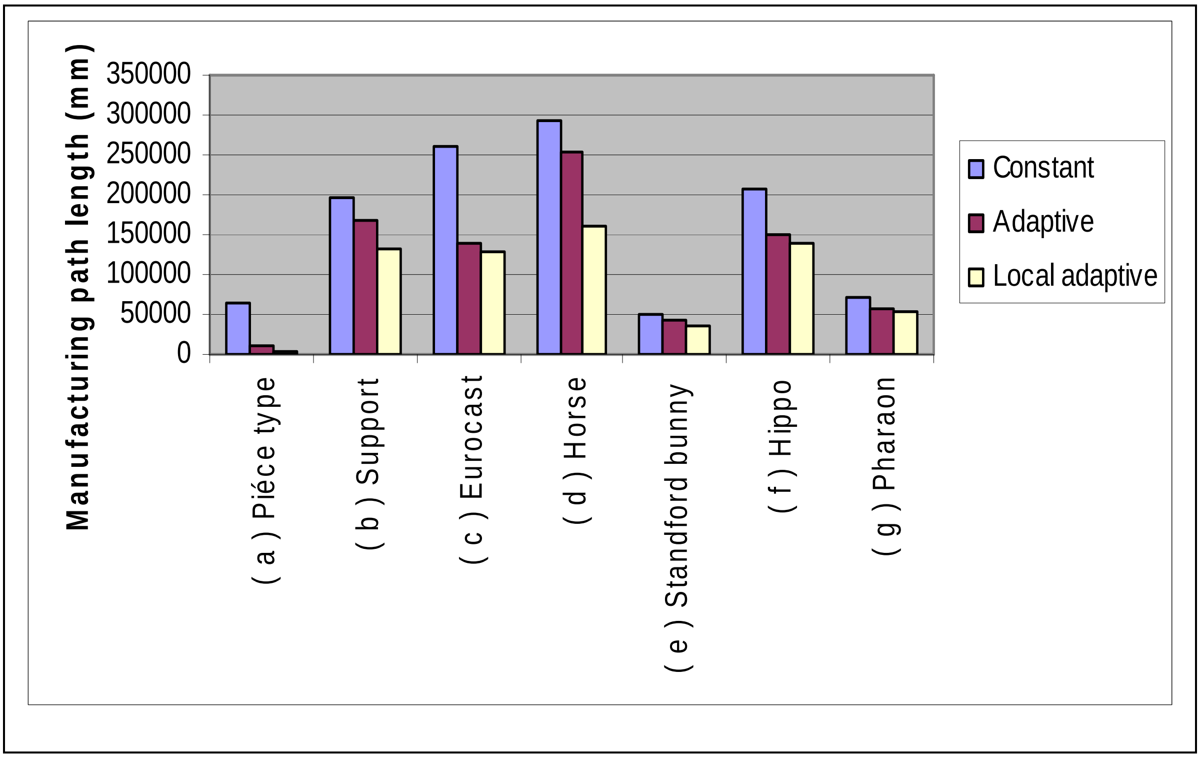

Figure 9: Manufacturing path length with different slicings (among which our automatic local adaptive slicing) with test models.

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

[1] Agarwal, P.K.; Edelsbrunner, H.; Harer, J.; Wang, Y. (2004) Extreme elevation on a 2-manifold, 20th Annual symposium on computational geometry, 357-365, Brooklyn, New York, USA

[2] Alboul, L.; Echeverria, G.; Rodrigues, M. (2005) Discrete curvatures and Gauss maps for polyhedral surfaces, 21st European Workshop on Computational Geometry, Eindhoven

[3] Barlier, C. (1991) Procédé pour la création et la réalisation de pièces par C.A.O et pièces ainsi obtenues, brevet n°91-02437, France

[4] Barlier, C. (2002) Le prototypage rapide dans la filière numérique de conception de produit, Revue annuelle des arts et métiers, 58-65

[5] Barre, S. (1994) La courbure de Gauss, Le journal de maths des élèves de l'école normale supérieure de Lyon, Volume 1, No. 1, 51-59

[6] Cohen-Steiner, D.; Alliez, P.; Desbrun, M. (2004) Variational shape approximation, Technical Report INRIA n°5371, November 2004

[7] Cohen-Steiner, D.; Edelsbrunner, H.; Harer, J. (2005) Stability of persistence diagrams, 21st Annual symposium on computational geometry, 263-271, Pisa, Italy

[8] Delebecque, B.; Lauvaux, G.; Barlier, C. (2000) Stratification adaptative, 8ème Assises Européennes du Prototypage Rapide, CNIT, Paris

[9] Dey, T.K.; WENGER, R. (2006) Stability of critical points with interval persistence, Tech Report OSU-CISRC-4/06-TR47, April 2006

[10] Edelsbrunner, H.; MOROZOV, D.; PASCUCCI, V. (2006) Persistence-sensitive simplification of functions on 2-manifolds, 22nd Annual symposium on computational geometry, 127-134, Sedona, Arizona, USA

[11] Jun, C-S; Kim, D-S; Hwang, J.S.; Chang, T-C. (2000) Surface slicing algorithm for rapid prototyping and machining, Geometric Modeling and Processing 2000, 373-382

[12] Maltret, J-L.; Daniel, M. (2002) Discrete curvatures and applications : a survey, Technical report, Laboratoire des Sciences de l'Information et des Systèmes, n° LSIS.RR.2002.002

[13]Meyer, M.; Desbrun, M.; Schröder, P.; Barr, A.H. (2002) Discrete differential-geometry operators for triangulated 2-manifolds, VisMath '02, Berlin, Germany
[14]Tyberg, J.T. (1998) Local adaptive slicing for layered manufacturing, Thesis of the Virginia Polytechnic Institute and State University, Department of Mechanical Engineering
[15]Tyberg, J.; Bohn, J.H. (1998) Local adaptive slicing, Rapid prototyping journal, Volume 4, Number 3,