A priori Computation of a mesh size for Adaptive Loop Subdivision

Sandrine Lanquetin, Marc Neveu
LE2I, UMR CNRS 5158
Université de Bourgogne, BP 47870
21078 DIJON Cedex, France
{slanquet, mneveu}@u-bourgogne.fr

Abstract

Subdivision schemes provide efficient algorithms both for the design and processing of arbitrary control meshes. Theoretically, these techniques are often considered as an elegant algorithmic way to approximate a desired surface from a given surface. In practice, controlling the accuracy of control meshes with regard to the limit surface remains difficult. In this paper, from a bound of the distance between a subdivision surface and its control polyhedron, we *a priori* compute the subdivision depth according to a given accuracy. From this result, we can exactly predict the number of faces of the control polyhedron for a given accuracy during full subdivision. As full subdivision generates too many faces, we focus in this paper on adaptive subdivision. We work with the adaptive scheme which generates the fewest number of faces and explain how to predict a bound of the number of faces of the subdivided mesh because it is not possible to a priori give the final number of faces during adaptive subdivision.

***Keywords: Subdivision surface, Loop scheme, polyhedral mesh, distance, adaptive subdivision, accuracy, mesh size*.**

1.INTRODUCTION

In computer graphics, subdivision surfaces are used to smooth meshes. These surfaces are defined by an initial coarse mesh named control polyhedron and refinement rules of subdivision. These rules describe how to move existing control points (old points) and how to create new points in order to reach the limit surface. The application of refinement rules generates a sequence of increasingly fine control meshes. The sequence of control meshes converges to a smooth surface: the limit surface. Among approximation schemes, the schemes of Loop [6] and Catmull-Clark [2] are the most popular. A particularity of approximation schemes is that control meshes approach the limit surface at each step of refinement (Figure 1). The maximum distance D at the initial level is always larger than the distance d at the next level and so on because the subdivision process 'smoothes' the mesh. In this paper, we focus on the Loop scheme.

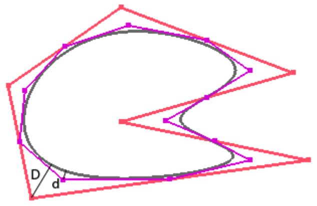


Figure 1: Distance in approximation schemes.

Each subdivision step provides a more accurate approximation of the limit surface. An upper-bound of this distance can be used as a stop criterion during the subdivision process [11][9]. However, the number of faces can be reduced by subdividing only particular faces. There are many criteria to determine faces to subdivide: faces in the viewport [3], angle between face normals [1], flatness of the surface [7], normal cones of vertices [8], curvature [12], distance [11][5]. We choose to use the distance between the control mesh and the limit surface as a geometric criterion for adaptive subdivision. This criterion, combined with local properties of subdivision, allows us to subdivide the surface only where this distance is greater than a given accuracy, saving a lot of triangles.

From a bound of the distance between a subdivision surface and its control polyhedron, the number of subdivisions to perform can be known *a priori* according to a wished accuracy. We can thus estimate the exact number of faces of the final mesh. During full subdivision, this is easy due to the definition of Loop.subdivision. Indeed, a face is split in four at each step of subdivision. When the mesh is partially subdivided, the number of faces cannot be predicted, we can only determinate an upper-bound.

The purpose of this paper is to a priori determine an upper bound of the number of triangles that will be generated during adaptive subdivision. It allows to estimate memory requirements and rendering rates, which is very usefull for applications in video games - mainly on portable playstations- or in virtual reality.

The paper is organized as follows. Section 2 reminds the reader with the necessary background that we use in our work: Loop subdivision and a bound for the distance of a subdivision surface to its control polyhedron. Section 3 explain the adaptive subdivision used with a distance criterion. In section 4, we determine the number of subdivisions to perform from an accuracy given at the initial level of subdivision and thus a bound for the number of faces of the mesh for adaptive subdivision and then in section 5, we give some practical results. Finally, directions for future work are proposed in the conclusion.

2.BACKGROUND

2.1. Loop subdivision surfaces

Subdivision surfaces are among the easiest ways to generate smooth surfaces. They preserve both the advantages of NURBS and those of polygonal meshes. We choose the Loop scheme for our results because the majority of meshes are currently triangular (triangular meshes provided by geometric modellers, triangulated meshes reconstructed from laser range images…). The Loop scheme generalizes quadratic triangular B-splines and the limit surface obtained is a quartic Box-spline. This scheme is based on splitting faces: each face of the control mesh at refinement level k is subdivided in four new triangular faces at level k + 1. This first step is illustrated into Figure 2. Consider a face: new vertices are inserted in the middle of each edge, they are named odd vertices and those of the initial face are named even vertices.

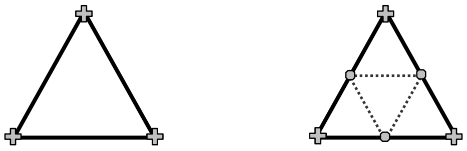


Figure 2: Left, an initial face. Right, the 4 new faces.

In the second step, all vertices are displaced by computing a weighted average of the vertex and its neighbourhood vertices. These averages can be substituted by applying different masks according to vertex properties: even, odd, valences (Figure 3). The sub-figure (a) represents the interior even vertex mask where n denotes the vertex valence and β is chosen to be [10]:

$$\beta = \begin{cases} \frac{3}{16} & \textit{if } n = 3 \\ \frac{1}{n}\left(\frac{5}{8} - \left(\frac{3}{8} + \frac{1}{4}\cos\left(\frac{2\pi}{n}\right)\right)^2\right) & \textit{if } n > 3 \end{cases}$$

The sub-figure (c) illustrates the interior odd vertex mask.

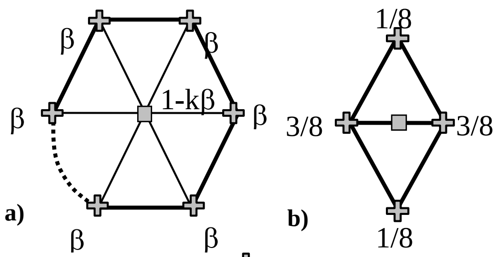


Figure 3: Loop masks where ✚ represent old vertices and ■ the new position of an even vertex (a) and of an odd vertex (b) respectively.

2.2. Loop distance to limit surface

Loop surfaces are defined as the limit of an infinite refinement process. However, it is easy to compute exact points on the limit surface. Indeed, the images of a vertex (from the initial mesh) on the successive subdivision meshes depend only on this vertex and its neighbourhood. The following results are derived from [6].

Having the image P_0^∞ of the vertex P_0^0 as a function of P_0^0 and its neighbourhood allows us to compute the distance between P_0^0 and P_0^∞ :

$$d\left(P_0^0, P_0^\infty\right) = \left\| \frac{-8.n.\beta}{8.n.\beta+3} P_0^0 + \frac{8.\beta}{8.n.\beta+3} \sum_{i=1}^{n} P_i^0 \right\|$$ with the valence of P_0^0 and β the Loop coefficient.

From the distance between the control mesh vertices and their image on the limit surface allows us to determine the maximum distance between the control mesh and the limit surface. Indeed, subdivision surface properties are such that the control mesh vertices are the most distant points from the limit surface because refining coefficients are all positive. If a vertex is inserted between two vertices of the control mesh, the distance between this vertex and the limit surface will inevitably be smaller (Figure 4). This property is valid for almost any surface: convex or concave, open or closed. This bound is suitable for all usual mesh, however an exotic mesh can be generated to contradict it. This is shown in [9].

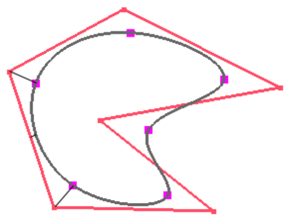


Figure 4: Distance from a midpoint to the limit surface.

Notations used for the vertices $P_{i,j}^k$ of the control mesh $\mathbf{P}^k$ at subdivision level k are described in Figure 5.

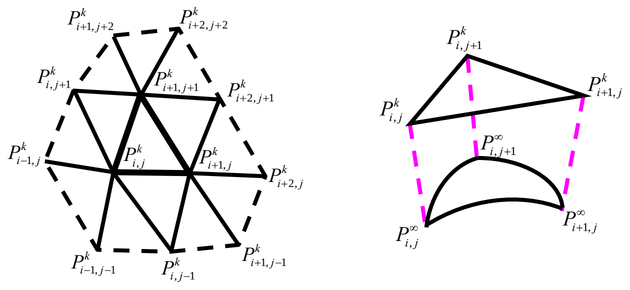


Figure 5: Notations used for the vertices of the control mesh. Left: in bold, Loop triangle at level k with its neighbourhood. Right: Loop triangle at level k with corresponding limit surface triangular patch.

In [6], the following theorem is proven:

Theorem 1. Let $S^k(u,v)$ be the polyhedral surface associated to the control mesh $\mathbf{P}^k$ at subdivision level k and $S^\infty(u,v)$ be the limit surface.

$$\left\|S^k(u,v) - S^\infty(u,v)\right\| \le M_k \le \left(\frac{5-8N\beta_N}{8}\right)^k M_0 \qquad \text{where}$$

$$M_k = \max_{P_{i,j}^k \in \mathbf{P}^k} \left(\left\| \frac{8n_{i,j}^k \beta_{i,j}^k}{8n_{i,j}^k \beta_{i,j}^k + 3} P_{i,j}^k - \frac{8\beta_{i,j}^k}{8n_{i,j}^k \beta_{i,j}^k + 3} \sum_{P_{i',j'}^k \text{ adjacent to } P_{i,j}^k} P_{i',j'}^k \right\| \right),$$

$n_{i,j}^k$ is the valence of the vertex $P_{i,j}^k$, $\beta_{i,j}^k$ is the Loop coefficient associated to this valence and N is the maximum valence of the successive control meshes.

3.ADAPTIVE SUBDIVISION

The adaptive subdivision scheme used in this paper is described in this section. Then the chosen scheme is analyzed and compared to other schemes.

3.1. Adaptive scheme

When the same rules are applied on the whole input mesh, the number of faces quickly increases. Indeed, for Loop scheme, a face produces four faces after one subdivision, 4² after 2 subdivisions and 4^n after n subdivisions. A well-known method to reduce the number of faces is to subdivide only a part of the mesh.

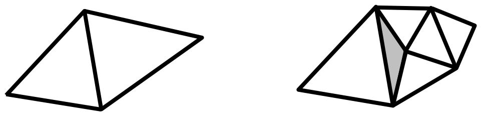


Figure 6: Left: two adjacent faces of the mesh. Right: one face is subdivided and the other is not, the crack between the faces is represented in grey.

When the surface is not entirely subdivided, cracks appear between faces with different subdivision depth as shown in Figure 6.

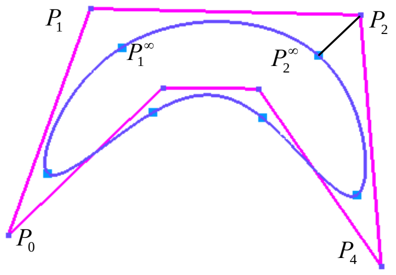


Figure 7: Upper-bound of the distance between a control vertex and the limit surface.

The area to subdivide can be selected by different ways. We choose to select it according to the accuracy of the control polyhedron from the limit surface. The adaptive subdivision used here generates a minimum number of faces. The chosen criterion selects faces according to the distance of its vertices to the limit surface. This distance cannot easily be computed, so the distance between a vertex and its image on the limit surface is used as an upper-bound (Figure 7).

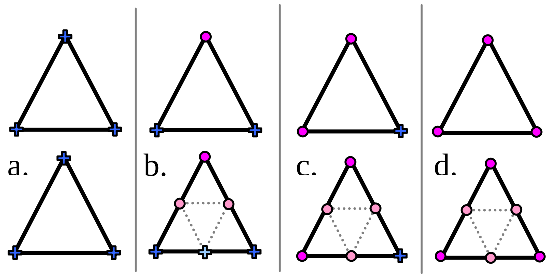


Figure 8. Different cases of face subdivision during adaptive subdivision.

In this paper, we choose to work with the first adaptive subdivision introduced in [4]. Vertices are classified into two categories: static and mobile vertices. A mobile vertex is a vertex which is displaced because the distance between this vertex and the limit surface is more than a given accuracy. Other vertices are static. The topological rules used to avoid cracks in this scheme are chosen as follows. Faces are classified into 4 categories according to the number of mobile vertices. Mobile vertices are depicted by circles in Figure 8 (top). When all vertices are static, the face is not subdivided Figure 8.a.). Figure 8.b. illustrates the case where only one vertex is mobile; only two among three new (odd) vertices are then mobile in order to avoid cracks. When there are two mobile vertices, face subdivision is almost normal except the fact that one among the old vertices is static (Figure 8.c.). Finally when all vertices are mobile, subdivision is carried out in a normal way (Figure 8.d.).

By applying this adaptive subdivision scheme, cracks are avoided because the inserted vertex is in the middle of the edge from the face to subdivide. Indeed, the middle vertex inserted on the edge between a face that is subdivided and a face that is not subdivided is not mobile any more. This scheme does not allow a correct computation of a neighborhood for all vertices for further subdivisions. Indeed faces are generated but correspondences between edges are not updated. Nevertheless, a neighborhood is not necessary because the distance for a static vertex to the limit surface does not change. We just need to save it from one level to another. For the other cases, the neighborhood is correct; the distance can thus be computed. Figure 9 illustrates a case where the new neighborhood should include both new and old vertices.

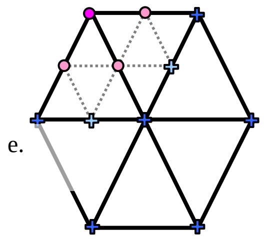


Figure 9: New neighbourhood.

3.2. Results and comparison

It is not easy to compare adaptive subdivision schemes because criteria are very different according to the required effect. For instance, comparing a view dependent scheme with ours, is not very interesting because they do not have the same goal at all.

One focuses in the part of the model which is in the view cone and the other on the accuracy of the entire model. Other criteria, such as those based on curvature or normal cone, are more similar to the above criterion of distance because they are dependent as shown in Figure 10. Indeed, for a vertex of the control mesh, the larger the angle of the normal cone the larger the distance between this control vertex and the limit surface. The distance is represented by a red line, the limit surface is represented by a blue shape and the normal cone is in grey color in Figure 10.

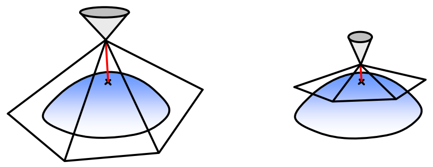


Figure 10: Relation between the size of the normal cone at a control vertex and the distance between this vertex and the limit surface.

Curvature, normal cone and distance are similar even in special cases such as that presented in Figure 11. Indeed, the distance between P_2 and the limit surface is negligible but it is the same for the curvature and the normal cone. On the other hand, the distance from P_1 or P_3 to the limit surface is large as the curvature or the normal cone.

For every adaptive scheme, topological rules are modified to avoid cracks. This modification involves that the limit surface is different from this obtained in subdividing the whole mesh. This is not important because subdivision is stopped according to the criterion, so the limit surface is never reached.

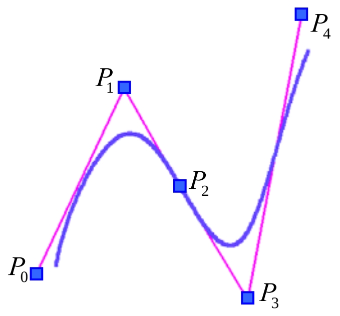


Figure 11: Similarity between distance curvature and normal cone.

The main advantage of our adaptive subdivision scheme is that the criterion is handier for the user. Moreover, for fields like computer aided geometric design, it is very significant to know the accuracy of the model. The results obtained with this distance criterion gives results similar to those obtained with other criterion such as curvature, normal cone but it is more intuitive and thus easier to use. Contrary to others adaptive scheme, the scheme introduced in this section do not generate a "good" mesh according to the mesh definition. Indeed, between subdivided and not subdivided faces, the mesh topology is particular. The odd vertex introduce and not displaced to avoid cracks (in grey in Figure 12 left) do not belong to the left face otherwise the face would not be more triangular. Thus, the mesh topology is represented in Figure 12 (right). The light grey line represents the edge of the left face which is in one part and the dark grey lines represent the edges of the right faces adjacent to the left face.

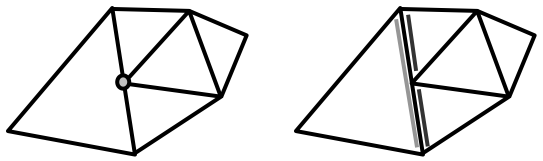


Figure 12: The generated mesh is not "good".

This kind of topology is sufficient to visualize mesh but it would be not sufficient in some particular cases. For example, in the case where the user modifies the tag of the grey vertex in Figure 12 (mobile instead of static) because he decides to change the needed accuracy or selects faces to subdivide with an other criterion. Computing intersection between two meshes can also be a problem because the data structure is not updated.

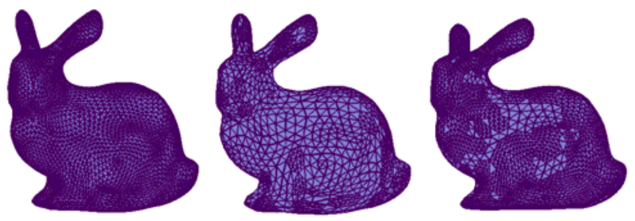


Figure 13: From left to right: respectively full subdivision, adaptive subdivision with an accuracy of 0.1, adaptive subdivision with an accuracy of 0.05.

In Figure 13, one can see on the one hand the results obtained for a given accuracy (here 0.1) and that the number of faces of the subdivided model in a non adaptive way corresponds to the number of faces which one can obtain in adaptive subdivision but for a higher accuracy (here 0.05). If the number of faces is smaller for the adaptive subdivision with a precision of 0.05, one can however note that time necessary to obtain the subdivided mesh is longer than adaptive subdivision with an accuracy of 0.1 because there is an additional level of subdivision. Table 1 gives numerical results corresponding to Figure 13 where a time value of 1 refers to the time necessary for a full subdivision.

	Plain $\varepsilon = 0.1$	Adaptive $\varepsilon = 0.1$	Adaptive $\varepsilon = 0.05$
Level of subdivision	3	3	4
Number of faces	9472	3841	7930
Minimum distance (.10^{-4})	4,16	2614	15.35
Maximum distance (.10^{-4})	831.95	831.95	256.01
Mean distance (.10^{-4})	136.58	333.55	165.40
Time (in ratio)	1	0.1	0.33

Table 1. Comparison of the results obtained on the bunny model.

4.SIZE OF MESHES

To manage the storage of the control mesh, we need to know a priori the size of this mesh. We have already seen in a previous section that the number of faces is different according to the chosen subdivision of the mesh. Indeed, adaptive subdivision generates fewer faces than full subdivision.

4.1. Full subdivision

From [6], we have:

Corollary 1. With M_0 and N as in Theorem 1, the control mesh approximates the limit surface with the accuracy ε if the number of recursive subdivision is at least k with $k > -\ln\left(\frac{M_0}{\varepsilon}\right) \Big/ \ln\left(\frac{5}{8} - N\beta_N\right)$ where $N = \max(6, N^0)$ and N^0 is the maximum valence of the initial control mesh.

<u>Note</u>: The number 6 comes from the subdivision. Indeed, during the subdivision process, the valence of each inserted vertex is 6. This is develop later in the paper.

When the subdivision of the control mesh is full, it is easy to determine the exact number of faces of this mesh at a given accuracy. Indeed, Corollary 1 gives the subdivision level according to a given accuracy. From this computation, the mesh has to be subdivided k times. However, in the full subdivision, we know that a face generates 4 sub-faces at the first subdivision level, 4^2 at the second subdivision level and 4^k at the k^{th} subdivision level. The control mesh at the k^{th} subdivision level will thus have $nf_k = 4^k \times nf_0$ faces with nf_0 the number of faces of the initial control mesh.

4.2. Adaptive subdivision

When the subdivision of the control mesh is adaptive, it will be interesting to *a priori* know the size of the control mesh at a given accuracy. Only faces the vertices of which have a distance to the limit surface greater than the given accuracy are subdivided. Thus, we cannot *a priori* exactly determine the number of faces of the control mesh for this accuracy. Nevertheless, we can upper-bound the number of faces of the control mesh without subdividing the initial control mesh.

Theorem 2. A vertex P_i^0 of valence n_{v_i} approximates the limit surface with the accuracy ε if the number of recursive subdivision is at least k_{v_i} with $k_{v_i} > -\ln\left(\frac{\left\|P_i^0 - P_i^\infty\right\|}{\varepsilon}\right) \Big/ \ln\left(\frac{5}{8} - n_i\beta_{n_i}\right)$ where $n_i = \max(6, n_{v_i})$.

Proof. For each vertex, we have:

$$\left\|P_i^k - P_i^\infty\right\| \le \left(\frac{5-8n_i\beta_{n_i}}{8}\right)^{k_{v_i}} \left\|P_i^0 - P_i^\infty\right\| < \varepsilon$$

The last part of the inequality can be written:

$$\left(\frac{5-8n_i\beta_{n_i}}{8}\right)^{k_{v_i}} \left\|P_i^0 - P_i^\infty\right\| < \varepsilon$$

$$\Leftrightarrow k_{v_i} > -\ln\left(\frac{\left\|P_i^0 - P_i^\infty\right\|}{\varepsilon}\right) \Big/ \ln\left(1 \Big/ \frac{5}{8} - n_i\beta_{n_i}\right)$$

Corollary 2. A face f_i approximates the limit surface with the accuracy ε if the number of recursive subdivision is at least k_i with $k_i = \max_{j\in[1,2,3]}\left(k_{v_j}\right)$ where k_{v_j} are computed for each vertex P_j^0 of f_i .

Corollary 3. The total number of faces after adaptive subdivision is upper-bounded by $|F| = \sum_{i=1}^{k_i} 4^i \times |f_i|$ where $|f_i|$ is the number of faces that needs to be subdivided k_i times.

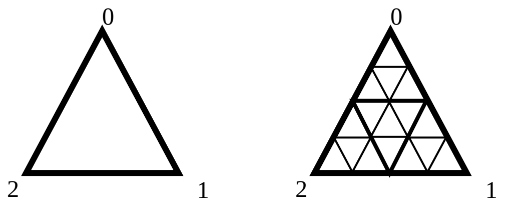


Figure 14: Left: Subdivision levels of face vertices. Right: Number of new sub-faces.

From the computation of the subdivision level k_{v_j} of each vertex of a face f_i, the subdivision level k_i of the face can be determined. We choose the subdivision level of the face to be the maximum of the subdivision level of its vertices. For instance, if the subdivision levels of the face vertices are respectively 0, 1, 2, the subdivision level of the face will be 2. In Figure 14 (on the left), subdivision levels of the vertices are represented by coefficients near the vertices.

As we wish an a priori upper bound of the mesh, the number of sub-faces created depends only on the initial subdivision level. Such as in the full subdivision, a face generates 4 sub-faces at the first subdivision level, 4^2 at the second subdivision level and 4^k at the k^{th} subdivision level. The generated sub-faces for the previous example are shown in Figure 14 (on the right). If we do not pay attention to special cases, we can have the following problem. Let the box in Figure be the initial control mesh. The subdivision level found for each initial vertex with a given accuracy $\varepsilon = 0.2$ is 2; however by successive subdivisions the mesh reaches the wished accuracy only after 3 subdivisions (Figure 15).

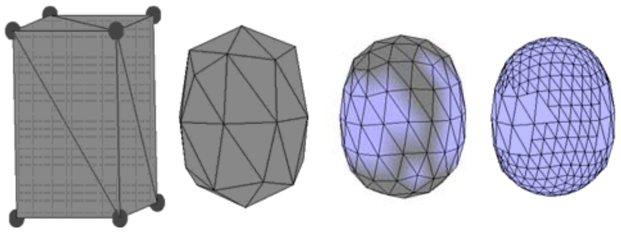


Figure 15: Subdivision depth: 2 subdivisions for each vertex and 3 subdivisions by successive refinement.

From Corollary 1, the computation of k_{v_i} uses $n_i = \max(6, n_{v_i})$. The importance of this maximum is shown in Figure 15 (a to d). Let the box in Figure 15.a. be the initial control mesh. The subdivision level found for each initial vertex with a given accuracy $\varepsilon = 0.2$ is 2; however by successive subdivisions the mesh reaches the wished accuracy only after 3 subdivisions.

The problem encountered is illustrated in Figure 16. In fact, some new vertices need one subdivision more than for the initial vertices. This problem comes from the valences of the new vertices which are always 6 whereas the initial vertices valences are 4 or 5. It can be avoided by taking 6 for the valence at the initial level when the computed valence is less than 6. For such vertices, the computation will thus be done with valence 6 and the corresponding Loop coefficient ($\beta = 1/16$).

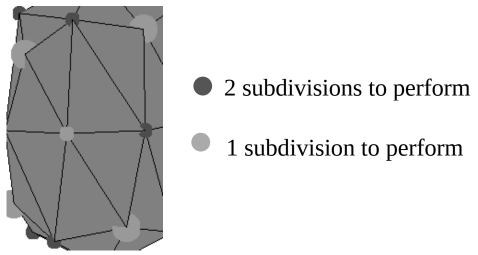


Figure 16. Focus on the problem: higher valence (6) of vertices after subdivision yields to additional subdivision

5.PRACTICAL RESULTS

In this section, results obtained on the penguin model with an accuracy $\varepsilon = 0.2$ are presented.

5.1. Full subdivision

For the full subdivision, the mesh has to be subdivided three times according to section 4.1. As the control mesh includes 500 faces at the initial level, the control mesh will have 32 000 faces at the third subdivision. Figure 17.b. illustrates the control mesh of the penguin model after full subdivision and Graph 1 represents the number of faces which have to be subdivided once, twice or more times according to the result presented in section 4.2.

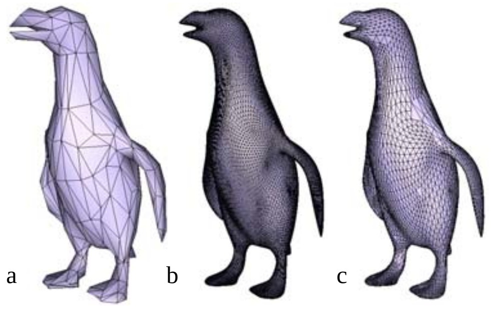
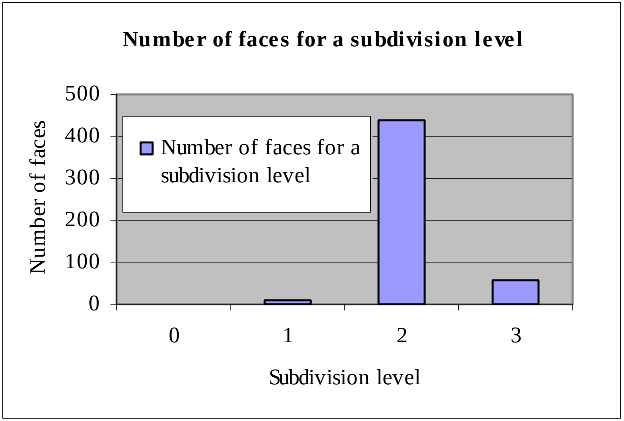


Figure 17: a. the penguin model. b. full subdivision with an accuracy $\varepsilon = 0.2$. c. adaptative subdivision with an accuracy $\varepsilon = 0.2$



Graph 1. Number of faces for successive subdivision levels in applying the formula on the penguin model with $\varepsilon = 0.2$.

5.2. Adaptive subdivision

For adaptive subdivision, we first compute the number of subdivision to perform for each vertex of the penguin mesh. The subdivision level of each face is then the maximum subdivision level of its vertices. Graph 1 shows the number of faces for each level of subdivision to perform. Subdividing each face the corresponding number of times, we can upper bound the real number of faces of the final adaptive mesh. In this case, by successive subdivisions, the adaptive mesh counts 8213 faces and our forecast is $10544 = \sum_{i=1}^{3} 4^i |f_i|$. Figure 17.c. represents the control mesh of the penguin model after adaptive subdivision. We cannot improve this upper-bound without subdividing the mesh.

5.3. Comparison on different meshes and accuracy

In this section, the number of faces are computed and estimated on various meshes and accuracies such as the penguin model with $\varepsilon = 0.1$ and $\varepsilon = 0.2$ (Figure 17), the cat model with $\varepsilon = 0.1$ and $\varepsilon = 0.2$ (Figure 18), the torus model with $\varepsilon = 0.1$ (Figure 20), the box model with $\varepsilon = 0.05$ (Figure 21) and the top model with $\varepsilon = 0.15$ (Figure 22).

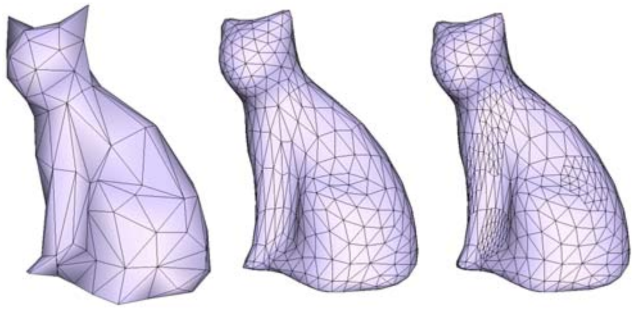


Figure 18: Successive control mesh in adaptive subdivision for the cat model with the accuracy 0.2.

Table 2 gives the subdivision depth according to the given accuracy on a regular surface (valence is 6 for every vertex), the torus. In figures 9 to 14, the dark (resp. light) faces represent the faces whose distance to the limit surface is greater (resp. less) than ε .

Accuracy ε	0.5	0.1	0.01	0.05	0.001
Subdivision depth	2	3	4	5	6

Table 2. Subdivision depth necessary for accuracy ε for the torus.

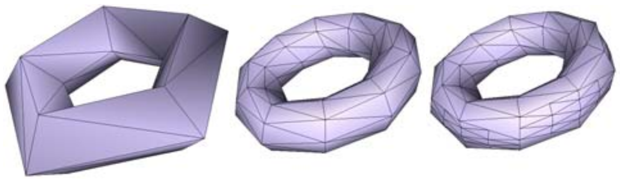


Figure 19: Subdivision of the torus ($n = 6$) with an accuracy ε of 0.5.

Figure 19 shows the number of subdivisions necessary to obtain an accuracy ε of 0.5 for the torus model which has regular valences everywhere. Figure 20 illustrates successive levels of subdivision to have an accuracy ε of 0.1. We can easily verify that subdivision level k gives a correct result whereas level $k-1$ leaves non accurate vertices (dark areas).

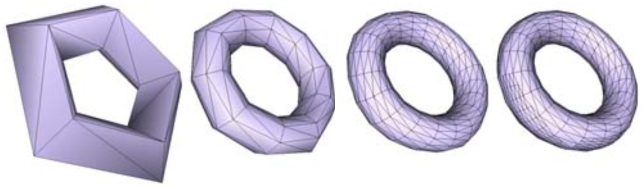


Figure 20: Subdivision of the torus ($n = 6$) with an accuracy ε of 0.1.

For a mesh with arbitrary valences, we have to pay attention to the role of valence n. n denotes the maximum valence of the mesh including the subdivision steps. When valences are strictly less than 6 on the initial control mesh, the maximum valence is 6 because all vertices inserted during the subdivision have valence 6. For instance, initial valences of the box are 4 or 5 so $n = 5$. For $\varepsilon = 0.05$, the subdivision depth k found for $n = 5$ is 3. But k is 4 when the maximum valence n is considered to be 6 due to subdivisions. Figure 21 shows successive subdivisions of the box. At level 3, some vertices are still further than ε from the limit surface. Indeed, subdivision depth is 4.

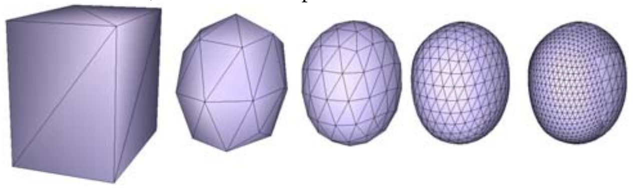


Figure 21: Box. Valence of inserted vertices must be taken into account.

In some cases, using the mean valence 6 is not sufficient. For instance, the surface in Figure 22 has a minimum valence of 4 and a maximum valence of 18. With an accuracy of 0.15, using mean valence gives a subdivision depth of 2, whereas the maximum valence gives 3, which is the true result.

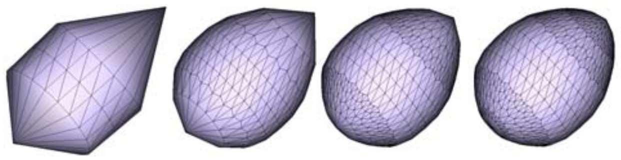


Figure 22: Spinning top. Mean valence (6) is not enough. An error remains at level 2 near the pole (vertex of valence 18).

It is then very important to choose N to be $\max(6, N^0)$ where N^0 is the maximum valence of the initial control mesh.

Table 3 gives the subdivision depth as a function of a given accuracy in the bunny model with arbitrary valences ($n \in [\![3,10]\!]$). For an accuracy ε with a difference of 2.10^{-3}, the subdivision depth increases by 1 (from 4 to 5) which represents an increased cost (memory and computation) even with adaptive subdivision.

Accuracy ε	0.5	0.1	0.05	0.012	0.01
Subdivision depth	1	2	3	4	5

Table 3. Subdivision depth necessary as a function of accuracy ε in the bunny model.

Figure 23 shows the number of subdivisions necessary to obtain an accuracy ε of 0.15 in the bunny model.

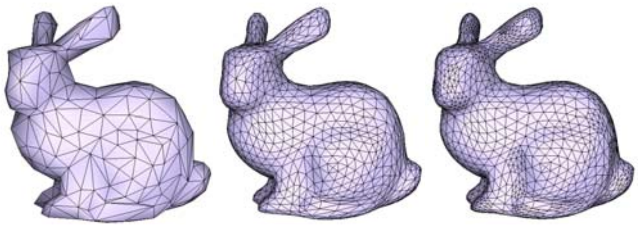
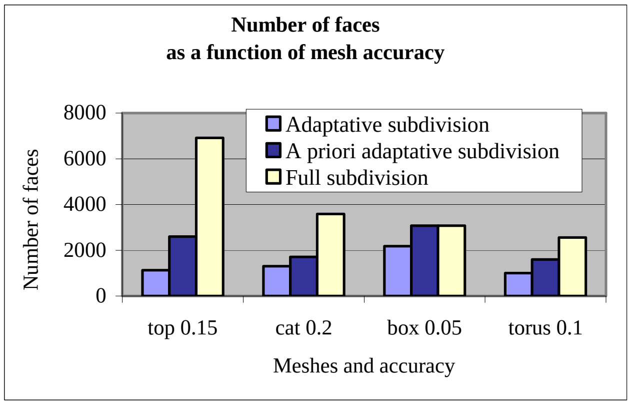
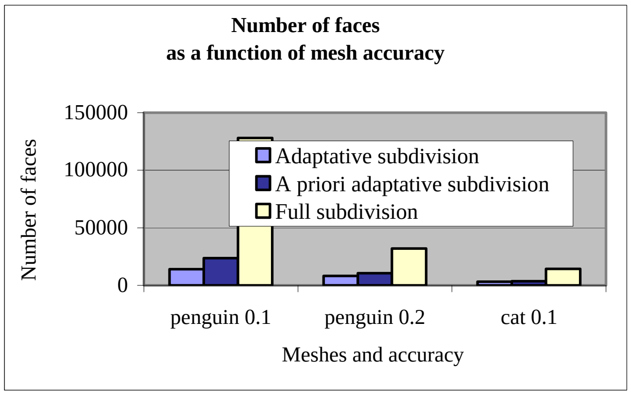


Figure 23: The bunny ($n \in [\![3,10]\!]$) with an accuracy of 0.1.



Graph 2. Number of faces found for top, cat, box and torus model according to the way of computation: successive adaptive subdivision, a priori estimated adaptive subdivision and a priori full subdivision.



Graph 3. Number of faces found for penguin and cat model according to the way of computation: successive adaptive subdivision, a priori estimated adaptive subdivision and a priori full subdivision.

It is easy to determine the number of faces of a mesh subdivided at a given accuracy without performing subdivisions. On the other hand, we do not know how to determine this number for adaptive subdivision without subdividing. So we proposed a formula in section 4.2 to upper-bound this number from the initial level. This formula gives too high a number of faces by construction but it represents the best upper-bound we can perform without subdividing. However the number of faces obtained is smaller than that of full subdivision.

The results are presented in Table 4 and Graph 2 and Graph 3. Graph 2 and 3 show for each mesh the number of faces found according to the way of computation: successive adaptive subdivision, a priori estimated adaptive subdivision and a priori full subdivision. Meshes used in Graph 2 are the following: the top model with a 0.15 accuracy, the cat model with a 0.2 accuracy, the box model with a 0.05 accuracy and the torus model with a 0.1 accuracy whereas Graph 3 is applied on the penguin model with a 0.1 then 0.2 accuracy and the cat model with accuracy 0.1. These results are completed with Table 6 where the first column gathers the additional percentage of faces in the upper-bounded subdivision compared to the real adaptive subdivision and the second column gives the percentage of faces saved in the upper-bounded subdivision compared to the full subdivision.

	Additional percentage of faces in the upper-bounded subdivision compared to the real adaptive subdivision	Percentage of faces saved in the upper-bounded subdivision compared to the full subdivision
penguin 0.1	69	81
penguin 0.2	28	67
cat 0.1	15	75
top 0.15	130	62
cat 0.2	31	52
box 0.05	41	0
torus 0.1	60	38

Table 4. Comparison of the number of faces in percentage.

6.CONCLUSION

From a distance bound between a subdivision surface and its control polyhedron we can adaptively subdivide meshes. This allows us to *a priori* determine the number of times that the control mesh has to be subdivided in order to approximate the limit surface with a given accuracy. From this number of subdivisions, mesh size can be determined for full subdivision and upper-bounded for adaptive subdivision. The determination of the mesh size, without subdividing, allows us to estimate properties of computers which are used to compute the scene. Indeed, in animation movies, there can be many meshes in a scene and they can be more or less significant. From this formula and from the data structure used, we can compute the necessary memory. Thus, if it exceeds the RAM memory of the computer (personal computer, portable playstation…), memory can be added. If the memory size cannot be increased (portable playstations for instance) scenes can be edited in order to realize the necessary compromise between quality (increasing the accuracy and thus the levels of subdivision) and memory occupation. Also, one usually considers that if the memory amount exceeds 4 GB, work can be parallelized. Moreover, for real time applications, such as virtual reality, the frame rate is a crucial criterion, which depends on the number of triangles to be displayed. A priori knowledge of this number can avoid further decimations. In this paper, the forecast of the number of faces is done only for Loop subdivision. As other subdivision schemes are currently used, we need to do the same as on this scheme. The distance used cannot always be the same. Indeed, for dual approximating schemes, we cannot upper bound the distance in the same way. Moreover for interpolating schemes, vertices of control meshes are on the limit surface, so this distance will always be null.

7.REFERENCES

[1] Amresh A., Farin G. and Razdan A., *Adaptive subdivision schemes for triangular meshes*, in *Hierarchical and Geometric Methods in Scientific Visualization*, H.H. G. Farin, and B. Hamann, editors, Editor. 2003: pp. 319-327.

[2] Catmull, E. and Clark J., *Recursively generated B-spline surfaces on arbitrary topological meshes.* Computer Aided Design, 1978. **9**(6): pp. 350-355.

[3] Hoppe H., *View-Dependent Refinement of Progressive Meshes.* Proceedings of ACM SIGGRAPH 97, 1997: pp. 189-198.

[4] Lanquetin S. and Neveu M., A *new non-uniform Loop scheme*. International Conference on Computer Graphics Theory and applications (GRAPP), Setubal, Portugal, 2006: pp. 134-141.

[5] Lanquetin S. and Neveu M., *A priori computation of the number of surface subdivision levels.* Proceedings of Computer Graphics and Vision (GRAPHICON 2004), Moscow, September 6-8, 2004: pp. 87-94.

[6] Loop, C. *Smooth Subdivision Surfaces Based on Triangles*. Department of Mathematics: Master's thesis, University of Utah, 1987.

[7] Meyer M. *et al.*, *Discrete differential-geometry operators for triangulated 2-manifolds.* VisMath, 2002.

[8] Müller K. and Havemann S., *Subdivision Surface Tesselation on the Fly using a Versatile Mesh data Structure.* Eurographics'2000, 2000. **19**(3): p. 151-159.

[9] Peters J., Wu X., *The distance of a subdivision surface to its control polyhedron,* Journal of Approximation Theory (2008), doi:10.1016/j.jat.2008.10.012.

[10] Wang H., Sun H. and Qin K., *Estimating Recursion Depth for Loop Subdivision.* International Journal of CAD/CAM, 2004, **4**(1) n°453: p. 11-18.

[11] Wu X. and Peters J., *An accurate error measure for adaptative subdivision surfaces*. Shape Modeling International, may 2005: p. 51-56.

[12] Zorin D., Schröder P. and Sweldens W., *Interactive multiresolution mesh editing.* SIGGRAPH'98 Proceedings, 1998: p. 259-268.

About the author

Sandrine Lanquetin is an assistant professor at Burgundy University, Dijon, France, Laboratory of Electronics, Informatics and Image (LEII). Research on subdivision surfaces.
Contact email : sandrine.lanquetin@u-bourgogne.fr.

Marc Neveu is a full professor at Burgundy University, Dijon, France, Laboratory of Electronics, Informatics and Image (LEII). His research concerns geometric modelling with a focus on deformable surfaces, subdivision surfaces and fractal surfaces.
Contact email : marc.neveu@u-bourgogne.fr.
Home page: http://danae.u-bourgogne.fr/Equipe/Pages/neveu/