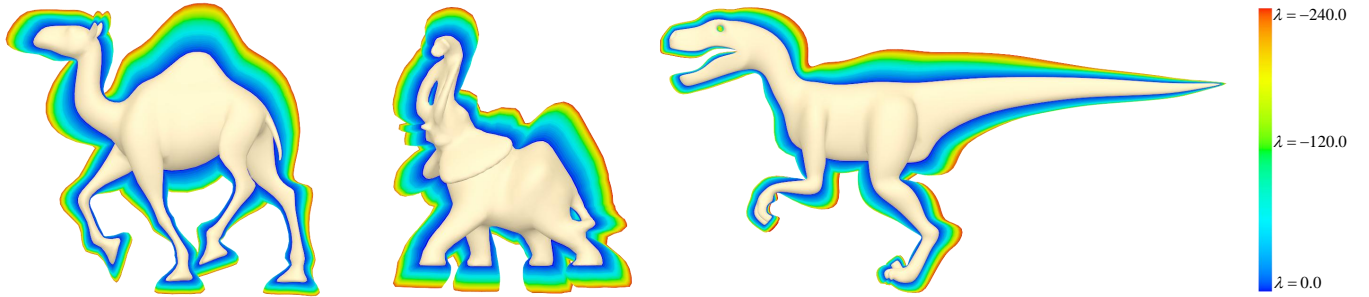# Curvature Enhance With Laplacian Operator For Hybrid Quad/Triangle Meshes

Alexander Pinzon*
Cimalab Research Group

Eduardo Romero†
Cimalab Research Group

Figure 1: *A set of 48 successive curvature enhance shapes, from $\lambda = 0.0$ in blue to $\lambda = -240.0$ in red, with steps of -5.0.*

Abstract

This paper proposes a novel method for modeling polygonal mesh using curvature enhancing. This method use our extension of Laplace Beltrami operator for hybrid quad/triangle meshes to enhance global curvature in the model. This work present a novel applications of curvature enhancement in sculpting and modeling with subdivision surfaces and weight vertex groups. We show a series of graphics examples that demonstrate the quality, predictability and flexibility of our results in a real production environment with software blender.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling —Modeling packages

Keywords: laplacian smooth, curvature, sculpting, subdivision surface

Links: ◆DL 🔧PDF

1 Introduction

Over the last years have been developed novel techniques of modeling that can generate a variety of shapes to look natural and realistic [Botsch et al. 2006]. Editing techniques have evolved from affine transformations to advanced tools like sculpting [Coquillart 1990; Galyean and Hughes 1991; Stanculescu et al. 2011], editing and creation from sketches [Igarashi et al. 1999; Gonen and Akleman 2012], complex interpolation techniques [Sorkine et al. 2004; Zhou et al. 2005], among others.

Traditional methods for smooth surfaces from coarse geometry like Catmull-Clark have been widely developed [Catmull and Clark 1978; Stam 1998], these works generalize uniform B-cubic splines knot insertion to meshes, some of them add control of the results with the use of creases to produce sharp edges [DeRose et al. 1998], or the modification of weights on the vertices that control locally the zone of influence [Biermann et al. 2000], instead our method performs a feature enhancement of the model allowing parameterize the curvature of the surface creating a family of different versions of the same object preserving detail and realistic natural look of the original model.

---
*e-mail: apinzonf@gmail.com
†e-mail:edromero@unal.edu.co

Many types of brushes have been developed to sculpt meshes, brushes that perform inflation lose detail when inflating the vertices [Stanculescu et al. 2011], our method allows inflation of the mesh vertices moving in the opposite direction to the curvature preserving the shape and sharp features of the model.

This work is organized as follows: In the section 2 show works related to the Laplacian mesh processing, digital sculpting, and offsetting methods for polygonal meshes; In the section 3 we describe the theoretical framework of the Laplacian operator for polygon meshes; In the section 4 we present our proposed method for curvature enhancement and applications on subdivision surfaces and sculpting; Lastly some results are shown in graphic examples of Laplacian operator for polygonal meshes composed of triangles and quads. Also shows some results of curvature enhancement applications in sculpting, subdivision and modeling methods.

Contributions Our work present an extension of the Laplace Beltrami operator for meshes of arbitrary topology composed for triangles and quads representing a larger spectrum of mesh that works with today eliminating the need for preprocessing and allows the preservation of the original topology. With this operator we proposed a method to generate family of parameterized shapes, in a robust and predictable way. Our method enable to customization of smoothness and curvature obtained during subdivision surfaces process. We proposed a new brush for enhance silhouette features of mesh in modeling and sculpting.

2 Related work

Many tools have been developed for modeling based on Laplacian mesh processing. Thanks to the kindness of the Laplacian operator these tools have in common the need for preservation of the geometric details of the surface for the different processes such as: free-form deformation, fusion, morphing and other applications [Sorkine et al. 2004].

Methods for offset polygon meshes based on the curvature defined by the Laplace Beltrami operator have been developed. These methods allow adjusting shape offset by a constant distance with high enough precision to minimize Hausdorf error. The problem with these methods is the loss of detail caused by smoothing, which depends on the size of the offset [Zhuo and Rossignac 2012]. In volumetric approaches on computing the offset boundary that are based on distance field computation in point-based representation,

this methods the topology of the offset model can be different from that the original geometry [Chen and Wang 2011].

[Gal et al. 2009] proposes automatic features detection and shape edition with feature inter-relationship preservation. In analysis step they define salient surface features how ridges and valleys with base on first and second order curvature derivatives [Ohtake et al. 2004], and angle-based threshold. In feature characterization step the curves are classified by several properties as planar or non-planar, approximated by line, circle or ellipse shapes, and so on. In edit step the user define initial change over several feature and then this edit is propagated over other features with base in your inter-relationships. This method works fine with objects that have sharp edges composed of basic geometric shapes such as lines, circles or ellipses but this method has difficulties when models are smoother with organic forms and cannot find the features to edit and preserve.

Digital sculpting is divided into two principal methods: based on polygonal methods and voxel grids-based methods. Brushes for inflate operations in polygonal methods only depends on the normal at each vertex [Stanculescu et al. 2011], in grids-based some operations permit add or remove voxels and then have that processing isosurfaces from volume to produce polygonal meshes representation [Galyean and Hughes 1991]. The problem whit this type of operations is the difficult to maintain surface details during larger scale deformation.

3 Laplacian Smooth

The Laplacian Smooth techniques allows you to reduce noise on a mesh's surface with minimal changes on its shape. Computer graphics objects which have been reconstructed from real world, contain undesirable noise. A Laplacian smoothing removes undesirable noise while still preserves desirable geometry as well as the shape of the original model.

The functional used in many Laplacian smoothing approach to constrain energy minimization is based on a total curvature of a surface S.

$$E(S) = \int_S \kappa_1^2 + \kappa_2^2 dS \qquad (1)$$

Where κ_1 and κ_2 are the two principal curvatures of the surface S.

3.1 Gradient of Voronoi Area
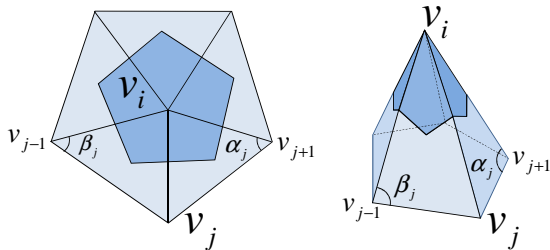
Figure 3: *Area of Voronoi region around v_i in dark blue. v_j 1-ring neighbors around v_i. α_j and β_j opposite angles to edge $\overrightarrow{v_j - v_i}$.*

Consider a surface S compound by a set of triangles around vertex v_i. We can define the *Voronoi region* of v_i as show in figure 3, The

change in area produced by move v_i is named gradient of *Voronoi region* [Pinkall et al. 1993; Desbrun et al. 1999; Meyer et al. 2003].

$$\nabla A = \frac{1}{2} \sum_j \left(\cot \alpha_j + \cot \beta_j \right) \left(v_i - v_j \right) \qquad (2)$$

If we normalize this gradient in equation (2) by the total area in 1-ring around v_i, we have the *discrete mean curvature normal* of a surface S as shown in equation (3).

$$2\kappa \mathbf{n} = \frac{\nabla A}{A} \qquad (3)$$

3.2 Laplace Beltrami Operator

The *Laplace Beltrami operator* LBO denoted $\triangle_g$ is used for measures mean curvature normal of the Surface S [Pinkall et al. 1993].

$$\triangle_g S = 2\kappa \mathbf{n} \qquad (4)$$

The LBO has desirable features, one feature of the LBO is in direction of surface area minimization, allowing us to minimize energy using it on a total curvature of a surface S at equation (1).

4 Proposed Method

Our method allow the editing of geometric features using the curvature enhancement and smoothing. Generating a parameterized family of shapes using a set of vertices representing a coarse sketch of the desired model. Our approach can be mixed with traditional or uniform subdivision surfaces methods and is iterative and converges towards a continuous and smooth version of the original model.

Unlike other methods, our method allows to use mixed arbitrary types of mesh representation as triangles and quads, exploiting the basic geometrical relationships facilitating and ensuring convergence of the algorithm and similar shapes consistent with the original shape against the other methods.

Our method allows the use of soft constraints weighting the effect of smoothing at each vertex based on a normalized weight, the weights are assigned to the control vertices of the original mesh or. The weights of the new vertices resulting from the subdivisions are calculated by interpolation, allowing to modify the behavior of the method on exact regions of the original model.

Our approach contain an extension of the Laplace Beltrami operator for meshes composed by triangles and quads. Using meshes composed by triangles and quads has been increasing in recent years due to the flexibility of modeling tools as Blender 3D [Blender-Foundation 2012]. Today many artists manually connecting vertices such that its edition allows simplest way to perform animation processes and interpolation [Mullen 2007]. For these reasons it is very important to develop an operator that allows working with this type of mesh immediately, eliminating the need to preprocessing the mesh to convert to triangles and losing the original design made by users.

### 4.1 Laplace Beltrami operator over triangular and quadrilateral meshes TQLBO

Given a mesh $M = (V, Q, T)$, with vertices V, quads Q, triangles T.

The area of 1-ring neighborhood (N_1) with shared face to vertex v_i in M is.
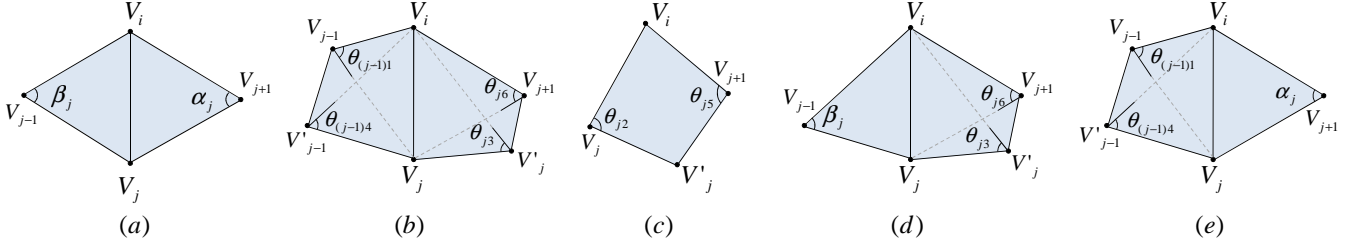
Figure 2: *The 5 basic triangle-quad cases with common vertex V_i and the relationship with V_j and V'_j. (a) Two triangles [Desbrun 1999]. (b) (c) Two quads and one quad [Xiong 2011]. (d) (e) Triangles and quads (TQLBO).*

$$A(v_i) = A\left(Q_{N_1(v_i)}\right) + A\left(T_{N_1(v_i)}\right).$$
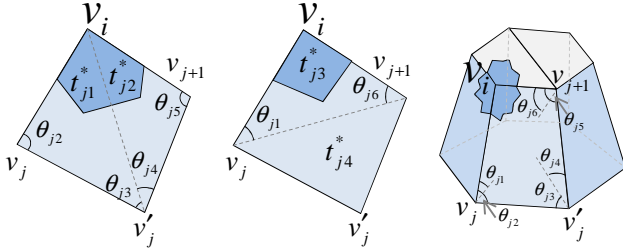
Figure 4: $t^*_{j1} \equiv \triangle \ v_i v_j v'_j$, $t^*_{j2} \equiv \triangle \ v_i v'_j v_{j+1}$, $t^*_{j3} \equiv \triangle \ v_i v_j v_{j+1}$ *Triangulations of the quad with common vertex v_i proposed by [Xiong 2011] to define Mean LBO.*

Applying the mean average area according to [Xiong et al. 2011] of all possible triangulations for each quad to $A\left(Q_{N_1(v_i)}\right)$ as show in figure 4.

$$A(v_i) = \frac{1}{2^m} \sum_{j=1}^{m} 2^{m-1} A(q_j) + \sum_{k=1}^{r} A(t_k)$$

Where $q_1, q_2, ..., q_i, ..., q_m \in Q_{N_1(v_i)}$ and $t_1, t_2, ..., t_k, ..., t_r \in T_{N_1(v_i)}$.

$$A(v_i) = \frac{1}{2} \sum_{j=1}^{m} \left[A(t^*_{j1}) + A(t^*_{j2}) + A(t^*_{j3}) \right] + \sum_{k=1}^{r} A(t_k) \quad (5)$$

Applying the gradient operator to (5).

$$\nabla A(v_i) = \frac{1}{2} \sum_{j=1}^{m} \left[\nabla A(t^*_{j1}) + \nabla A(t^*_{j2}) + \nabla A(t^*_{j3}) \right] + \sum_{k=1}^{r} \nabla A(t_k) \quad (6)$$

According to (2), we have.

$$\nabla A(t^*_{j1}) = \frac{\cot\theta_{j3}(v_j - v_i) + \cot\theta_{j2}(v'_j - v_i)}{2}$$

$$\nabla A(t^*_{j2}) = \frac{\cot\theta_{j5}(v'_j - v_i) + \cot\theta_{j4}(v_{j+1} - v_i)}{2}$$

$$\nabla A(t^*_{j3}) = \frac{\cot\theta_{j6}(v_j - v_i) + \cot\theta_{j1}(v_{j+1} - v_i)}{2}$$

$$\nabla A(t_k) = \frac{\cot\alpha_k(v_k - v_i) + \cot\beta_{k+1}(v_{k+1} - v_i)}{2}$$

All triangles and quads configurations of the 1-neighborhood faces adjacent to v_i can be simplified in five simple cases how show in figure 2.

Then according to equation (3), (4), and five simple cases defined in figure 2 the TQLBO (Triangle-Quad LBO) of v_i is.

$$\triangle_g(v_i) = 2\kappa\mathbf{n} = \frac{\nabla A}{A} = \frac{1}{2A} \sum_{v_j \in N_1(v_i)} w_{ij}(v_j - v_i) \quad (7)$$

$$w_{ij} = \begin{cases} (\cot\alpha_j + \cot\beta_j) & \text{case } a. \\ \frac{1}{2}\left(\cot\theta_{(j-1)1} + \cot\theta_{(j-1)4} + \cot\theta_{j3} + \cot\theta_{j6}\right) & \text{case } b. \\ (\cot\theta_{j2} + \cot\theta_{j5}) & \text{case } c. \\ \frac{1}{2}\left(\cot\theta_{j3} + \cot\theta_{j6}\right) + \cot\beta_j & \text{case } d. \\ \frac{1}{2}\left(\cot\theta_{(j-1)1} + \cot\theta_{(j-1)4}\right) + \cot\alpha_j & \text{case } e. \end{cases} \quad (8)$$

We define a Laplacian operator as a matrix equation

$$L(i,j) = \begin{cases} -\frac{1}{2A_i} w_{ij} & \text{if } j \in N(v_i) \\ \frac{1}{2A_i} \sum_{j \in N(v_i)} w_{ij} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

Where L is the $n \times n$ matrix, n is the number of vertices of a given mesh M, w_{ij} is the TQLBO defined in equation (8), $N(v_i)$ is the 1-ring neighbors with shared face to v_i, A_i is the ring area around v_i.
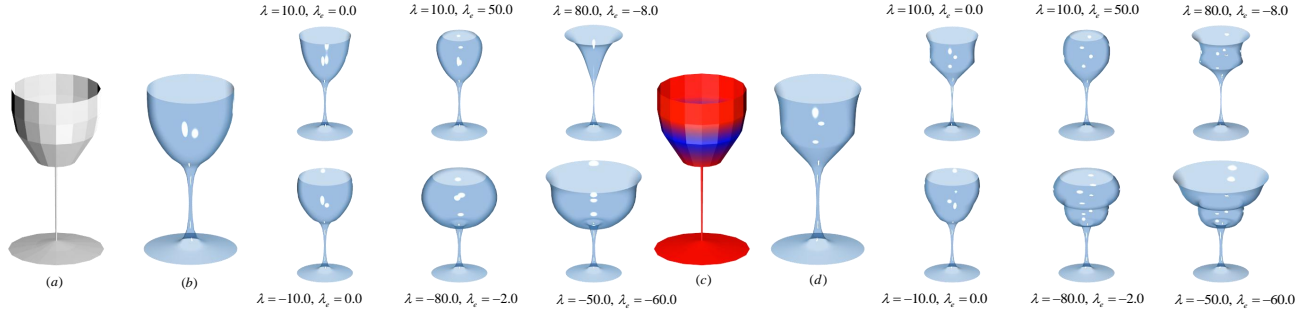
Normalized version of the TQLBO as a matrix equation

$$L(i,j) = \begin{cases} -\frac{w_{ij}}{\sum_{j \in N(v_i)} w_{ij}} & \text{if } j \in N(v_i) \\ \delta_{ij} & \text{otherwise} \end{cases} \quad (10)$$

Where δ_{ij} being the Kronecker delta function.

4.2 Curvature Enhancing

The curvature enhancing use the change produced by Laplacian smoothing in the inverse direction of the curvature flow for moves the vertices in the portions of the mesh with most curvature. In this equation we use a diffusion process:

$$\frac{\partial V}{\partial t} = \lambda L(V)$$

$\lambda=10.0,\ \lambda_e=0.0$ $\lambda=10.0,\ \lambda_e=50.0$ $\lambda=80.0,\ \lambda_e=-8.0$ $\lambda=10.0,\ \lambda_e=0.0$ $\lambda=10.0,\ \lambda_e=50.0$ $\lambda=80.0,\ \lambda_e=-8.0$

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

$\lambda=-10.0,\ \lambda_e=0.0$ $\lambda=-80.0,\ \lambda_e=-2.0$ $\lambda=-50.0,\ \lambda_e=-60.0$ $\lambda=-10.0,\ \lambda_e=0.0$ $\lambda=-80.0,\ \lambda_e=-2.0$ $\lambda=-50.0,\ \lambda_e=-60.0$

Figure 5: *Family of cups generated with our method from coarse model to enhancing the curvature obtained from Catmull-Clark Subdivision and the use of constraints over coarse model with weight vertex group in red.*

For solve the equation above we use implicit integration and a normalized version of TQLBO matrix.

$$(I - |\lambda dt|\, W_p L)\, V' = V^t \qquad (11)$$

$$V^{t+1} = V^t + \text{sign}\,(\lambda)\,(V' - V^t)$$

The vertices V^{t+1} are enhance along their inverse curvature normal directions by solving this simple linear system: $Ax = b$, where $A = I - |\lambda dt|\, W_p L$, L is the Normalized TQLBO defined in the equation (10), $x = V'$ are the smoothing vertices, $b = V^t$ are the actual vertices positions, W_p is a diagonal matrix with weight vertex group, $\text{sign}\,(x)$ is the sign function, and λdt is the enhance factor that support negative and positive values, negative for enhancing positive for smoothing.

At the borders of the meshes that are not closed, you can not calculate the curvature, for that reason we use the scale-dependent operator proposed by [Desbrun et al. 1999].

Our method was designed for use with weight vertex groups to specify the degree of impact on the solution, the weights vary between 0 and 1 with a value of 0 makes no changes and with values 1 applies the total change. The weights modifying influence zones where the Laplacian is applied as shown in the equation 11. The weights on each vertex will produce a different solution for that reason are put before obtaining the solution of the linear system. Families of shapes that are generated may change substantially with the weights of specific control points.

The model volume increases as the lambda is most negative, this can be countered by a simple method of preserving volume. In [Desbrun et al. 1999] present a simple method to resize the mesh but have a problem the model suffer large displacements with $\lambda < -1.0$ or perform multiple iterations. We propose the following solution: If we have v_i^{t+1} is a mesh vertex of V^{t+1} in the $t + 1$ iteration.

$$\overline{v} = \frac{1}{n} \sum_{v_i \in V} v_i,$$

$\overline{v}$ is the center of the mesh, vol_{ini} is an initial volume, and vol_{t+1} is the volume at the iteration $t + 1$, then we have that scale factor for resize the volume is

$$\beta = \left(\frac{vol_{ini}}{vol_{t+1}}\right)^{\frac{1}{3}}$$

and the new vertices positions are:

$$v_{i\,new}^{t+1} = \beta\,(v_i^{t+1} - \overline{v}) + \overline{v}$$

4.3 Sculpting

We design a new brush that allows enhancement of the curvatures of a polygon mesh in real time. Our brush work well with the stroke method *Drag Dot* developed in the sculpt mode in Blender [Blender-Foundation 2012] which allows you to preview the change that occurs in the model until you release the mouse button or tablet, also enable moves the mouse over the model to fit exactly where you want to perform the enhancement of the curvature.

Brushes that perform similar work as inflate can create distortions in the mesh and can also produce self-intersections of the mesh, as this brushes only moves the vertices along the normal and does not take into account the global information. Whereas our method look for the best way to make inflation while keeping the global curvature for preserving the shape and sharp features of the model.

Our method simplifies the work required for the enhancement, which would be to use some different brushes for inflating and some other to soften and styling. With our enhanced brush can be performed in one step.

For the real-time work brush is necessary that the matrix is constructed with the vertices that are within the radius defined by the user involvement, which dramatically reduces the size of the matrix to be processed, the center of this sphere depends on the place where the user clicks on the canvas and three-dimensional mesh place where the click is projected.

Furthermore special handling is necessary with the vertices are at the boundary of which have neighbors that are not within the radius of impact, these vertices must be marked as boundary and not the curvature calculated for them, but must be present in the matrix to allow the vertices that have all their neighbors within the selection radius to calculate correctly the curvature, this change allows the results to be much smoother on the border. The Laplacian matrix for sculpt mode as a matrix equation.

$$L\,(i,j) = \begin{cases} -\dfrac{w_{ij}}{\sum\limits_{j \in N(v_i)} w_{ij}} & \text{if } \|v_i - u\| < r \wedge \|v_j - u\| < r \\ 0 & \text{if } \|v_i - u\| < r \wedge \|v_j - u\| \geq r \\ \delta_{ij} & \text{otherwise} \end{cases}$$

Where $v_j \in N\,(v_i)$, u is the center of sphere the radius r. Then the matrix should remove rows and columns of vertices index that are not within the radius.

$\lambda = 60.0, \lambda_e = 12.0$ $\lambda = 90.0, \lambda_e = 80.0$ $\lambda = -200.0, \lambda_e = 5.0$ $\lambda = -40.0, \lambda_e = -100.0$
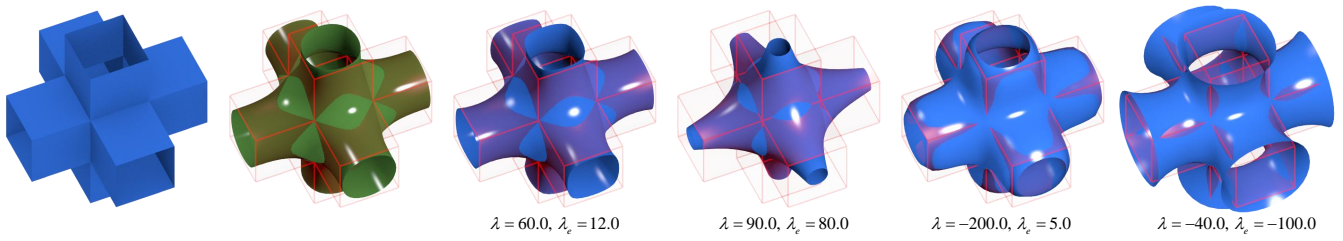
Figure 6: *Left: Original Model, in green color model with Catmull-Clark Subdivision. Models with Laplacian smoothing: $\lambda = 60.0$, $\lambda_e = 12.0$ and $\lambda = 90.0$, $\lambda_e = 80.0$. Models first filter with Laplacian smoothing $\lambda = 60.0$, $\lambda_e = 12.0$ and before applied curvature enhancing: $\lambda = -200.0$, $\lambda_e = 12.0$ and $\lambda = -40.0$, $\lambda_e = -100.0$.*

4.4 Subdivision surfaces

The Catmull-Clark subdivision transformation is used to smooth a surface as the limit of sequence of subdivision steps[Stam 1998]. This method do a recursive subdivision transformation that refines the model into a linear interpolation that is a approximate smooth surface. The process of Catmull-Clark is govern by properties of B-spline curve from multivariate spline theory[Loop 1987]. In many subdivision surfaces methods Catmull-Clark, Loop so on. The smoothness of the model is automatically guaranteed[DeRose et al. 1998].

Catmull-Clark subdivision surface method generates smooth and continuous models from a coarse model and produce results quickly due to the simplicity of implementation, but with these methods is not easy to make changes to the global curvature of the model. If we use Catmull-Clark subdivision surfaces and curvature enhancement for modeling from sketches with few vertices can generate families of shapes with just a parameter, which would allow an artist to choose the model of a similar set of options that best meets your needs without having to change each one of the control vertices.

Our method allows the use of vertex weight paint over the control points. The weights can be applied to the coarse model, then to perform subdivision weights are interpolated, producing weights with smooth changes in the zones of influence, so the curvature that is obtained is much softer at those areas. Using vertex group weight and subdivision surfaces is shown in figure 5.c.

In the equation 11, W_p is a diagonal matrix with weights corresponding to each vertex. The weights at each vertex produce a different solution for this reason the matrix is placed in the diffusion equation, as families that are generated may change substantially with weighted of specific control points.

5 Results

In this section we describe the results of our curvature enhance method that used our extension of Laplace Beltrami operator for triangles and quads TQLBO with several example models in figures 1,8. We test the curvature enhance with TQLBO method on a PC with AMD Quad-Core Processor @ 2.40 GHz and 8 GB RAM.

Figure 7 show the results when applying Laplacian smoothing operator TQLBO of equation (9) in a model with a simple subdivision. In column (c) Laplacian smoothing was applied to the model consists of only quads. In column (d) the model was converted to triangles and then Laplacian smoothing was applied. In column (e) the model is converted randomly some quads into triangles and then Laplacian smoothing was applied, showing similar results to those composed only of triangles or squares.

Figure 8 show the generation of different version of camel with the

(a)   (b)   (c)   (d)   (e)
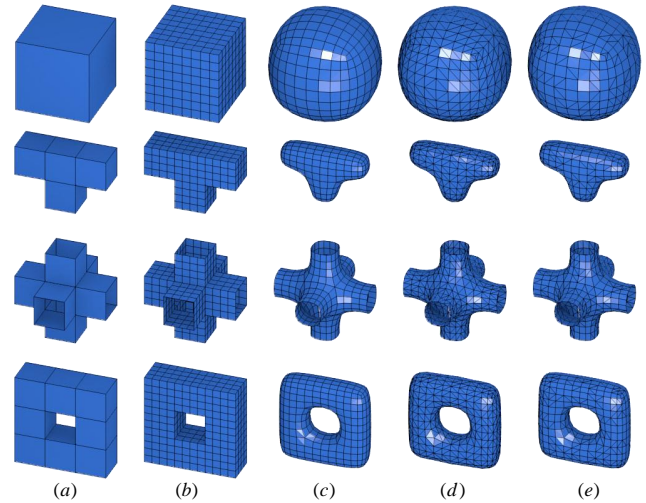
Figure 7: *(a) Original Model. (b) Simple subdivision. (c), (d) (e) Laplacian smoothing with $\lambda = 7$ and 2 iterations: (c) for triangles, (d) for quads, (e) for triangles and quads random chosen.*

Original $\lambda = -30.0$ $\lambda = -100.0$ $\lambda = -400.0$

$\lambda = -50.0$, 2 iterations $\lambda = -200.0$, 1 iteration
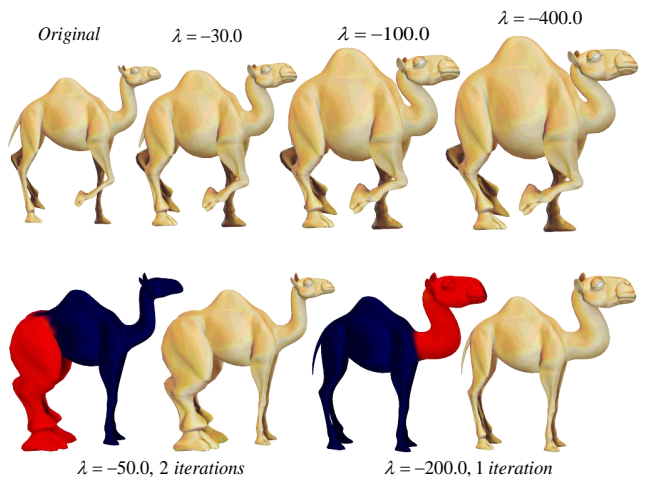
Figure 8: *Top row: Original camel model in left. Curvature enhancing with $\lambda = -30.0$, $\lambda = -100.0$, $\lambda = -400.0$. Bottom row: Curvature enhancing with weight vertex group, $\lambda = -50.0$ and 2 iterations at legs, $\lambda = -200.0$ and 1 iteration in head and neck.*

variation of parameter lambda. In the top row you can see results of do curvature enhance over all model, as the lambda becomes more negative curvature in this model makes close in the concave parts and inflates on the convex parts as shown in figure 1. We use negative values to enhance model silhouette features, among more negative the lambda, the model will be further enhanced the silhouette features. In the bottom row of the figure 8 can be observed using weighted vertex groups, which allows you to specify which areas you want to model enhancement, on the left is the enhancement of the legs of the camel produces enhancement of organic aspect, also notes that the border is not distorted and smooth in the union.

We did tests with the Laplacian operator to equation (9) and its normalized version equation (10), the two produce similar results if the triangles that compose the mesh are the same size on average, but the normalized version is more stable and predictable because it is not divided by the area of the ring which may be zero or very small and cause problems due to floating point calculation errors as shown in figure 10.c bottom row in which the mesh is deformed because the TQLBO is susceptible to the size of the triangle. The enhancement of curvature of the model with the normalized Laplacian operator has a more regular behavior. The model can deform in TQLBO normalized version with large lambdas> 400 intersects self, but produces no peaks observed with the TQLBO. Figure 10 observed different results because the area of the triangles in this model is not so regular triangles where greater area enhancement is less (figure 10.c skull), and where the smaller triangles are produced greater enhancement (figure 10.c chin).

Our method for enhancement of the silhouette features is predictable and invariant under isometric transformations as those present in some animations (see Figure 11) in this animation shows some poses of a camel doing a walk. In this animation the legs and neck are the parts that enhancement is performed as shown in the bottom left part of the figure 11. Local modifications produced by methods such as pose interpolation or animation rigging not significantly affect the result as with camel legs in each sample pose different flexion of the leg joints enhancement camel keeps flesh-like shapes in the pattern produced original by the artist. This is due to the diffusion process which is subjected to the mesh so that small local changes are treated without significantly affecting global solution. Our method is invariant of rotation, since depends solely the normal field of the mesh, wish is invariant under global rotations.

Figure 9 shows the use of curvature enhancement locally in real time. One pass was used with the brush as shown in the figure with the blue and red radius. In the bottom of the figure 9.b shows how the camel leg self-intersecting and looks like two bubbles stuck, so do the fingers on the bottom of figure. Using silhouette features enhancement in figure 9.c we see better results because not lost the shape of the silhouette and kept the details of the fingers and leg. Similar results can be obtained by an artist but it would take several steps and the use of several brushes with curvatures enhancement only takes one step. With this new method can easily enhance organic features how muscles during the sculpting process.

Methods using Catmull-Clark subdivision surface and the enhancement allows to modify the curvature that is obtained with the process of subdivision as shown in figure (5) this test used a coarse model of a cup, after the model was performed subdivision then was performed Laplacian smoothing and enhancement by modifying the parameters λ y λ_e corresponding to the lambda for rings and edges respectively. In figure (5).c, (5).d shows also the use weight vertex groups over coarse models with subdivision surfaces that allowed to generate the weights for the new interpolated vertices, these new weights were used for the enhancement obtained the 6 cups that are to the right of the figure (5).d.
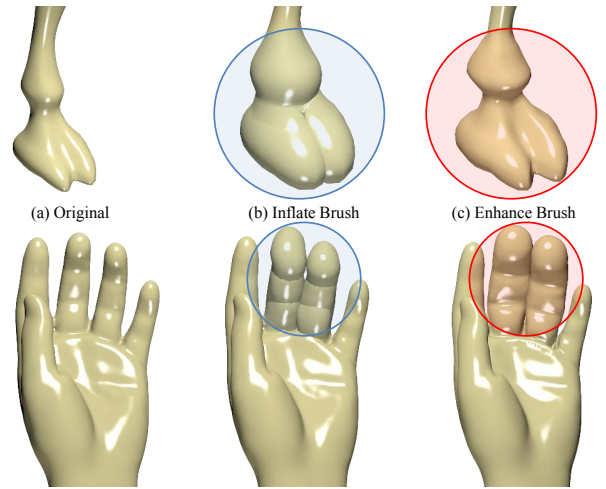
Figure 9: *Top row: (a) Leg Camel, (b) Inflate brush for leg into blue circle, (c) Enhance curvature brush for leg into red circle. Bottom row: (a) Hand, (b) Inflate brush for fingers into blue circle, (c) Enhance curvature brush for fingers in red circle.*
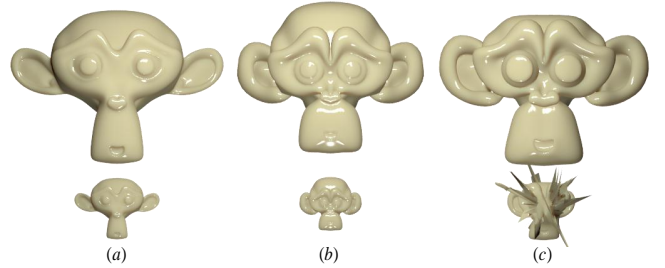

Figure 10: *(a) Top row: Original model scaled by 4. Bottom row: Original Model (b) Top and bottom row: enhancing with Normalized-TQLBO $\lambda = -50$ (c) Top and bottom row: enhancing with TQLBO $\lambda = -50$.*

Laplacian smoothing applied with simple subdivision may produce results similar to obtained with Catmull-Clark whose models are on average equal triangles as shown in figure 6 green model and that obtained with Laplacian smoothing $\lambda = 60.0$, $\lambda_e = 12.0$, but also can modify the curvature obtained after applying Catmull-Clark as shown in the three columns to the right of the figure 6.

5.1 Implementation

Our method was implemented how a modifier for modeling and brush for sculpting on the Blender software [Blender-Foundation 2012] with C and C++. Working with Blender allowed us to test the method interactively with others as Catmull-Clark, weight vertex groups and sculpting system in Blender.

To improve performance, we worked with the Blender mesh struct calculating all possible data each time visiting each triangle or quad and adding to its corresponding index in a list that stored the sum of the Laplacian weights of the ring, in this way only had to visit two times the list of faces of the mesh list and the two times the list of the edges if the mesh was not closed. To brush sculpting mode was necessary to create a list that will map the index of the selected vertices to a list from 1 to N where N is the number of vertices selected and the number of rows in the linear system to solve, with this drastically reduced the calculations should make working with
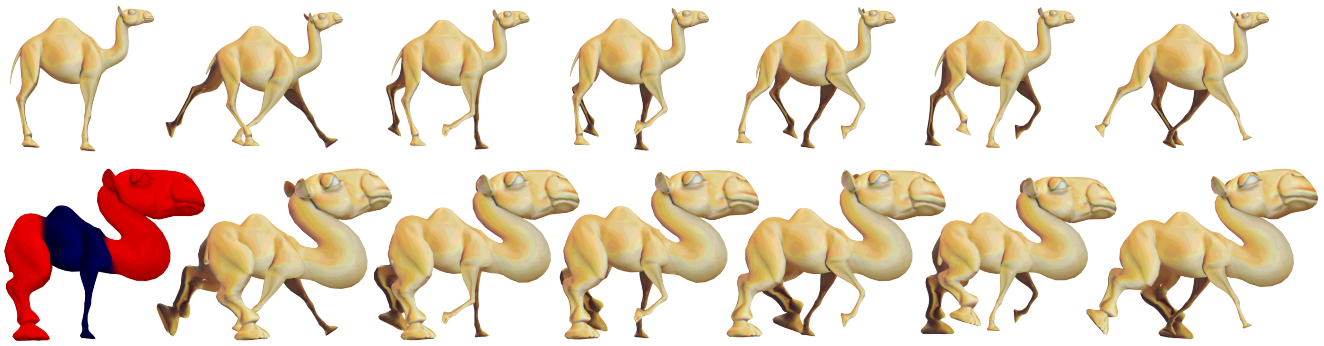
Figure 11: *Our method is pose insensitive. The enhanced for the different poses are similar in terms of shape. Top row: Original walk cycle camel model. Bottom row: Curvature enhancing with weight vertex group,* $\lambda = -400$ *and 2 iterations.*

the tool allowing real-time. In the construction of our Laplacian matrix were blocked at vertices having faces areas or lengths edges with value zero that can cause spikes and bad results.

The matrix 9 is sparse since the number of neighbors per vertex corresponding to the number of data per row is small compared to the total number of vertices in the mesh. To solve the linear system equation 11 was used OpenNL which is a a library for solving sparse linear system.

6 Conclusion and future work

We present an extension of the Laplace Beltrami operator for triangles and quads that can work in production environments without conversion that provides results similar to those obtained by working only with triangles or quads.

We introduced a new way to perform enhancement silhouettes in a mesh for modeling or sculpted in a few steps allows the modification of the curvature of a model while maintaining its overall shape. We introduce a new method of modeling and show some of its possible uses. We show that the method behaves in predictable ways which facilitate the learning process, it also showed that the method works well with isometric transformations opening the possibility of introducing it on the stages of animation.

We demonstrate that this tool serves to work in the early stages in which are used coarse models to permit modifying the curvature generated by the Catmull-Clark subdivision surfaces, avoiding the edition of the vertices to have only to change a few parameters.

As future work we would like to analyze theoretically the relationship between the Catmull-Clark subdivision surfaces and Laplacian smoothing because in some cases can produce very similar results. But the subdivision surface is a fast method thereby reducing computation times for calculate curvature in the mesh.

Acknowledgments

References

BIERMANN, H., LEVIN, A., AND ZORIN, D. 2000. Piecewise smooth subdivision surfaces with normal control. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, SIGGRAPH '00, 113–120.

BLENDER-FOUNDATION, 2012. Blender open source 3d application for modeling, animation, rendering, compositing, video editing and game creation. http://www.blender.org/.

BOTSCH, M., PAULY, M., ROSSL, C., BISCHOFF, S., AND KOBBELT, L. 2006. Geometric modeling based on triangle meshes. In *ACM SIGGRAPH 2006 Courses*, ACM, New York, NY, USA, SIGGRAPH '06.

CATMULL, E., AND CLARK, J. 1978. Recursively generated b-spline surfaces on arbitrary topological meshes. *Computer-Aided Design 10*, 6 (Nov.), 350–355.

CHEN, Y., AND WANG, C. C. L. 2011. Uniform offsetting of polygonal model based on layered depth-normal images. *Comput. Aided Des. 43*, 1 (Jan.), 31–46.

COQUILLART, S. 1990. Extended free-form deformation: a sculpturing tool for 3d geometric modeling. *SIGGRAPH Comput. Graph. 24*, 4 (Sept.), 187–196.

DEROSE, T., KASS, M., AND TRUONG, T. 1998. Subdivision surfaces in character animation. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, SIGGRAPH '98, 85–94.

DESBRUN, M., MEYER, M., SCHRÖDER, P., AND BARR, A. H. 1999. Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, ACM Press Addison-Wesley Publishing Co., New York, NY, USA, SIGGRAPH '99, 317–324.

GAL, R., SORKINE, O., MITRA, N. J., AND COHEN-OR, D. 2009. iwires: An analyze-and-edit approach to shape manipulation. *ACM Transactions on Graphics (Siggraph) 28*, 3, #33, 1–10.

GALYEAN, T. A., AND HUGHES, J. F. 1991. Sculpting: an interactive volumetric modeling technique. *SIGGRAPH Comput. Graph. 25*, 4 (July), 267–274.

GONEN, O., AND AKLEMAN, E. 2012. Smi 2012: Short paper: Sketch based 3d modeling with curvature classification. *Comput. Graph. 36*, 5 (Aug.), 521–525.

IGARASHI, T., MATSUOKA, S., AND TANAKA, H. 1999. Teddy: a sketching interface for 3d freeform design. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, SIGGRAPH '99, 409–416.

LOOP, C. 1987. *Smooth Subdivision Surfaces Based on Triangles*. Department of mathematics, University of Utah, Utah, USA.

MEYER, M., DESBRUN, M., SCHRÖDER, P., AND BARR, A. H. 2003. Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and Mathematics III*, H.-C. Hege and K. Polthier, Eds. Springer-Verlag, Heidelberg, 35–57.

MULLEN, T. 2007. *Introducing character animation with Blender*. Indianapolis, Ind. Wiley Pub. cop.

OHTAKE, Y., BELYAEV, A., AND SEIDEL, H.-P. 2004. Ridge-valley lines on meshes via implicit surface fitting. *ACM Trans. Graph. 23*, 3 (Aug.), 609–612.

PINKALL, U., JUNI, S. D., AND POLTHIER, K. 1993. Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics 2*, 15–36.

SORKINE, O., COHEN-OR, D., LIPMAN, Y., ALEXA, M., RÖSSL, C., AND SEIDEL, H.-P. 2004. Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, ACM, New York, NY, USA, SGP '04, 175–184.

STAM, J. 1998. Exact evaluation of catmull-clark subdivision surfaces at arbitrary parameter values. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, SIGGRAPH '98, 395–404.

STANCULESCU, L., CHAINE, R., AND CANI, M.-P. 2011. Freestyle: Sculpting meshes with self-adaptive topology. *Computers & Graphics 35*, 3, 614 – 622. Shape Modeling International (SMI) Conference 2011.

XIONG, Y., LI, G., AND HAN, G. 2011. Mean laplace-beltrami operator for quadrilateral meshes. In *Transactions on Edutainment V*, Z. Pan, A. Cheok, W. Muller, and X. Yang, Eds., vol. 6530 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 189–201.

ZHOU, K., HUANG, J., SNYDER, J., LIU, X., BAO, H., GUO, B., AND SHUM, H.-Y. 2005. Large mesh deformation using the volumetric graph laplacian. *ACM Trans. Graph. 24*, 3 (July), 496–503.

ZHUO, W., AND ROSSIGNAC, J. 2012. Curvature-based offset distance: Implementations and applications. *Computers & Graphics 36*, 5, 445 – 454.