Curvature Enhanced Shape With and Adapted Laplacian Operator For Hybrid Quad/Triangle Meshes

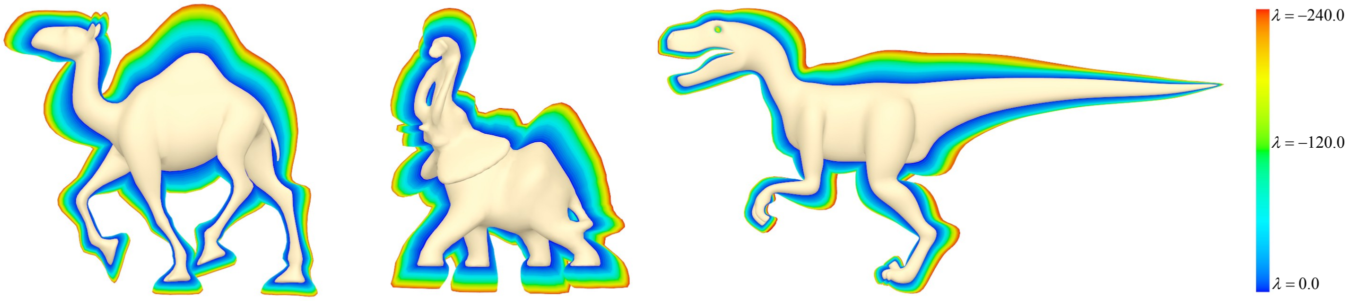

Figure 1: *A set of 48 successive curvature enhance shapes, from* $\lambda = 0.0$ *in blue to* $\lambda = -240.0$ *in red, with steps of* -5.0.

Abstract

This paper proposes a novel modeling method for a hybrid quad/triangle mesh by enhancing the global object curvature. This method uses our adapted extension of the Laplace Beltrami operator. Along with the method, this work presents new applications in sculpting and modeling, with subdivision of surfaces and weight vertex groups. A series of graphics examples demonstrates the quality, predictability and flexibility of the method 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

1 Introduction

Over the last years several, modeling techniques able to generate a variety of realistic shapes, have been developed [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, creation from sketches [Igarashi et al. 1999; Gonen and Akleman 2012], and complex interpolation techniques [Sorkine et al. 2004; Zhou et al. 2005]. Catmull-Clark based methods however require to interact with a minimun number of control points for any operation to be efficient, or in other words, a unicity condition is introduced by demanding a smooth surface after any of these shape operations. Hence, traditional modeling methods for subdividing surfaces from coarse geometry have become widely popular [Catmull and Clark 1978; Stam 1998]. These works have generalized a uniform B-cubic spline knot insertion to meshes, some of them adding some type of control, for instance with the use of creases to produce sharp edges [DeRose et al. 1998], or the modification of some vertex weights to locally control the zone of influence [Biermann et al. 2000]. Neverteless these methods are difficult to deal with since they require a large number of parameters and a very tedious customization. Instead, the presented method requires a single parameter that controls the global curvature, which is used to maintain realistic shapes, creating a family of different versions of the same object and therefore preserving the detail of the original model and a realistic appearance.

Interest in meshes composed of triangles and quads has lately increased because of the flexibility of modeling tools such as Blender 3D [Blender-Foundation 2012]. Nowadays, many artists use a manual connection of a couple of vertices to perform animation processes and interpolation [Mullen 2007]. It is then of paramount importance to develop operators that easily interact with such meshes, eliminating the need of preprocessing the mesh to convert it to triangles. The curvature enhancement and shape exaggeration can thus be used as such operator in the sculpting process, when inflating a shape since current brushes end up by losing detail when moving vertices [Stanculescu et al. 2011]. In contrast, the presented method inflates a mesh by moving the vertices towards the reverse curvature direction, conserving the shape and sharp features of the model.

Contributions This work presents an extension of the Laplace Beltrami operator for hybrid quad/triangle meshes, representing a larger mesh spectrum from what has been presented so far. The method eliminates the need of preprocessing and allows preservation of the original topology. Likewise, along with this operator, it is proposed a method to generate a family of parameterized shapes, in a robust and predictable way. This method enables customization of the smoothness and curvature, obtained during the subdivision surfaces process. Finally, it is proposed a new brush for enhancing the silhouette mesh features in modeling and sculpting.

This work is organized as follows: Section 2 presents works related to the Laplacian mesh processing, digital sculpting, and offsetting methods for polygonal meshes; In section 3 , it is described the theoretical framework of the Laplacian operator for polygon meshes; In section 4, it is presented the method for curvature enhancement and applications of subdivision of surfaces and sculpting; finally some Laplacian operator results, to hybrid quad/triangle meshes are graphically shown as well as results of the curvature enhancement applications in sculpting, subdivision and modeling.

2 Related work

Many tools have been developed for modeling, based on the Laplacian mesh processing. Thanks to the advantages of the Laplacian operator, these different tools preserve the surface geometric details when using them for different processes such as free-form deformation, fusion, morphing and other application[Sorkine et al. 2004].

Offset methods for polygon meshing, based on the curvature defined by the Laplace Beltrami operator, have been developed. These methods adjust the shape offset by a constant distance, with enough precision as to minimize the Hausdorff error. Nevertheless, these methods fail to conserve sufficient detail because of the smoothing, a crucial issue which depends on the offset size [Zhuo and Rossignac 2012]. In volumetric approaches, in case of point-based representations, the offset boundary computation is based on the distance field and therefore when calculating such offset, the topology of the model may be different to the original [Chen and Wang 2011].

Gal et al. [Gal et al. 2009] propose automatic feature detection and shape edition with feature inter-relationship preservation. They define salient surface features like ridges and valleys, characterized by their first and second order curvature derivatives [Ohtake et al. 2004], and angle-based threshold. Likewise, curves have been also classified as planar or non-planar, approximated by lines, circles, ellipses and other complex shapes. In such case, the user defines an initial change over several features which is propagated towards other features, based on the classified shapes and the interrelationships between them. This method works well with objects that have sharp edges, composed of basic geometric shapes such as lines, circles or ellipses. However, the method is very limited when models are smooth since it cannot find the proper features to edit.

Digital sculpting have been traditionally approached either under a polygonal representation or a voxel grid-based method. Brushes for inflation operations only depend on the vertex normal [Stanculescu et al. 2011]. In grid-based sculpting, some other operations have allowed to add or remove voxels since production of polygonal meshes require a processing of isosurfaces from a volume [Galyean and Hughes 1991]. The drawback comes from the difficulty of maintaining the surface details during larger scale deformations.

3 Laplacian Smooth

Computer objects, reconstructed from the real world, are usually noisy. Laplacian Smooth techniques allow a proper noise reduction on the mesh surface with minimal shape changes, while still preserving a desirable geometry as well as the original shape.

Many smoothing Laplacian functionals regularize the surface energy by controling the total surface curvature$S$.

$$E\left(S\right) = \int_S \kappa_1^2 + \kappa_2^2 dS \tag{1}$$

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

3.1 Gradient of Voronoi Area

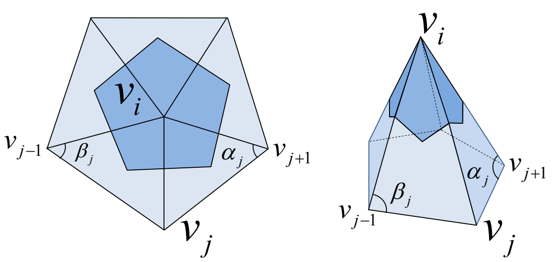

Figure 3: *Area of the Voronoi region around v_i in dark blue. v_j belong to the first neighborhood around v_i. α_j and β_j opposite angles to edge $\overrightarrow{v_j - v_i}$.*

Consider a surface S composed of a set of triangles around vertex v_i. Let us define the *Voronoi region* of v_i as show in figure 3, The area change produced by the movement of v_i is called the gradient of *Voronoi region [Pinkall et al. 1993; Desbrun et al. 1999; Meyer et al. 2003].*

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

If the gradient in equation (2) is normalized by the total area of the first neighborhood around v_i, the *discrete mean curvature normal* of a surface S is obtained, as shown in equation (3).

$$2\kappa\mathbf{n} = \frac{\nabla A}{A} \tag{3}$$

3.2 Laplace Beltrami Operator

The *Laplace Beltrami operator* LBO noted as $\triangle_g$ is used for measuring the mean curvature normal to the Surface S [Pinkall et al. 1993].

$$\triangle_g S = 2\kappa\mathbf{n} \tag{4}$$

The LBO has desirable properties: the LBO points out to the direction of the minimal surface area, minimizing the energy of the total surface curvature S at equation (1).

4 Proposed Method

The method allows shape magnification and exaggeration using the laplacian smoothing operator in the reverse direction, generating a parameterized family of shapes from a set of vertices which represent a coarse sketch of the desired model. Our approach can be mixed with traditional or uniform subdivision surfaces methods, is iterative and converges towards a continuous and smooth version of the original model.

Our method allows the use of soft constraints weighting the effect of smoothing at each vertex based on a normalized weight, which 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 modifying the behavior of the method on specific regions of the original model.

Our work presents an extension of the Laplace Beltrami operator for hybrid quad/triangle meshes, allows using mixed arbitrary types of mesh representation, exploiting the basic geometrical relationships facilitating and ensuring convergence of the algorithm and similar shapes consistent with the original shape against the other methods.

4.1 Laplace Beltrami Operator for Hybrid Quad/Triangle 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 quad or triangle to vertex v_i in M is.

$$A\left(v_i\right) = A\left(Q_{N_1(v_i)}\right) + A\left(T_{N_1(v_i)}\right).$$

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.

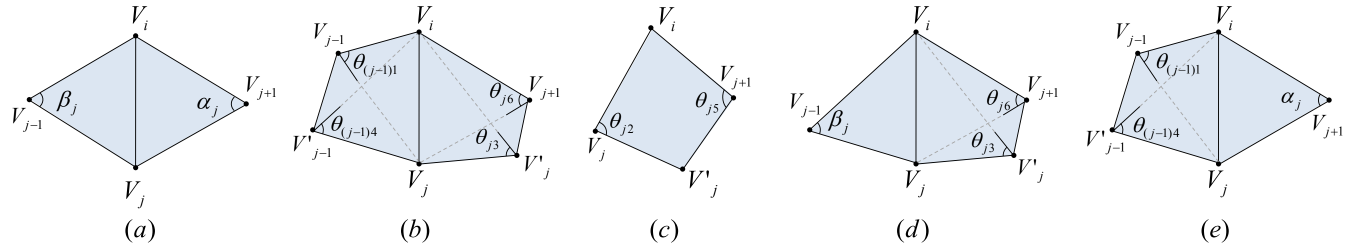

Figure 2: *The 5 basic triangle-quad cases with common vertex V_i and the relationship with V_j and V'_j. (a) Two triangles [Desbrun 1999]. (b) (c) Two quads and one quad [Xiong 2011]. (d) (e) Triangles and quads (TQLBO).*

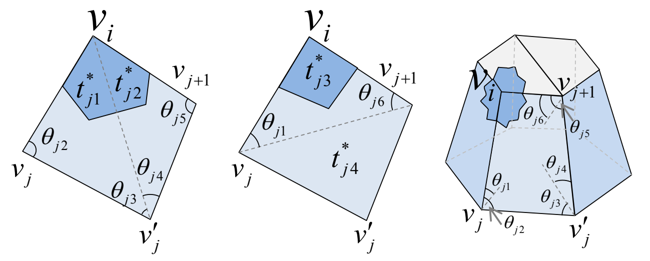

Figure 4: $t^*_{j1} \equiv \triangle\ v_i v_j v'_j$, $t^*_{j2} \equiv \triangle\ v_i v'_j v_{j+1}$, $t^*_{j3} \equiv \triangle\ v_i v_j v_{j+1}$ *Triangulations of the quad with common vertex v_i proposed by [Xiong 2011] to define Mean LBO.*

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

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

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

Applying the gradient operator to (5).

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

According to (2), we have.

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

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

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

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

Thus triangles and quads configurations of the 1-ring neighborhood faces adjacent to v_i can be simplified to five cases shown 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.

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

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

We define a Laplacian operator as a matrix equation

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

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

Normalized version of the TQLBO as a matrix equation

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

Where δ_{ij} being the Kronecker delta function.

4.2 Curvature Enhancing

The curvature enhancing uses the inverse direction of the curvature flow to move the vertices in the portions of the mesh with most curvature. In this equation we use a diffusion process:

$$\tfrac{\partial V}{\partial t} = \lambda L\left(V\right)$$

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

$$\left(I - \left|\lambda dt\right| W_p L\right) V' = V^t \quad (11)$$

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

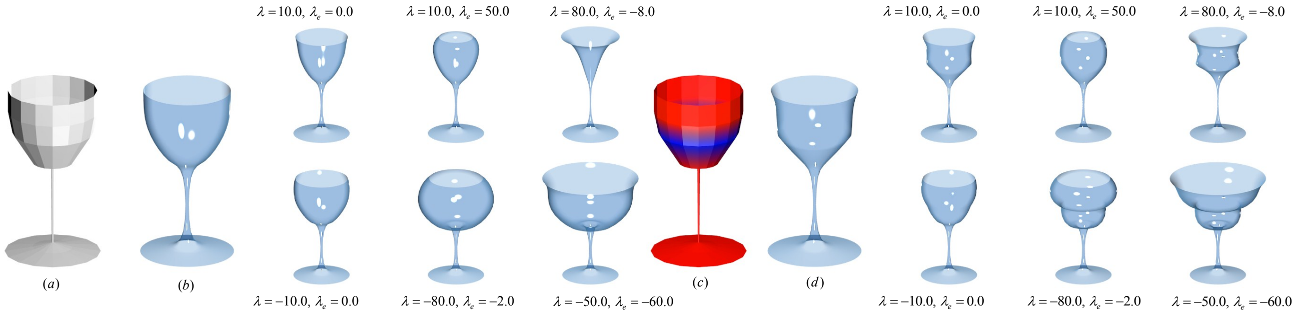
$\lambda = 10.0,\ \lambda_e = 0.0$
$\lambda = 10.0,\ \lambda_e = 50.0$
$\lambda = 80.0,\ \lambda_e = -8.0$
$\lambda = 10.0,\ \lambda_e = 0.0$
$\lambda = 10.0,\ \lambda_e = 50.0$
$\lambda = 80.0,\ \lambda_e = -8.0$
(a)
(b)
(c)
(d)
$\lambda = -10.0,\ \lambda_e = 0.0$
$\lambda = -80.0,\ \lambda_e = -2.0$
$\lambda = -50.0,\ \lambda_e = -60.0$
$\lambda = -10.0,\ \lambda_e = 0.0$
$\lambda = -80.0,\ \lambda_e = -2.0$
$\lambda = -50.0,\ \lambda_e = -60.0$

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.*

The vertices $V^{t+1}$are enhanced along their inverse curvature normal directions by solving the 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, and λdt is the enhance factor that support negative and positive values, negative for enhancing positive for smoothing.

The curvature cannot be calculted at the border of the meshes that are not closed, for that reason we use the scale-dependent operator proposed by [Desbrun et al. 1999].

Our method was designed for use with weighted vertex groups which specify the final curvature enhancement on the solution, these weights vary between 0 and 1 with a value of 0 makes no changes and with values 1 applies a maximal change. The weights modify influence zones where the Laplacian is applied as shown in the equation 11. Families of shapes that are generated may change substantially with the weights of specific control points.

The models volume increases as the lambda is larger and negative, this can be go against by a simple method of volume preservation. In [Desbrun et al. 1999] presents a simple method to resize the mesh however the model suffers large displacements when $\lambda < -1.0$ or after multiple iterations. We propose the following solution: If v_i^{t+1} is a mesh vertex of V^{t+1} in the $t+1$ iteration, we define $\overline{v}$ as:

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

$\overline{v}$ 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 define a scale factor.

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

The new vertices positions are:

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

4.3 Sculpting

We designed a new brush that allows the enhancement of the curvatures of a polygon mesh in real time. Our brush works well with the stroke method named *Drag Dot* which allows to preview the changes in the model before you release the mouse button or pen tablet, also allows to move 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 in the mesh, as these brushes only move the vertices along the normal and do not take into account the global information. Whereas our method looks for the best way to make inflation while preserving the global curvature retaining the several 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 in one step can be performed.

For the real-time brush it necessary that the Laplacian matrix is constructed with the vertices that are within the radius of the sphere defined by the user, which 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 required for the vertices at the boundary which have neighbors that are not within the radius of the brush. These vertices must be marked as boundary and the curvature will not be calculated for them, but must be present in the matrix so that every vertex has their neighbors within the selection, this change allows the results to be much smoother on the border. The Laplacian matrix for sculpt mode as a matrix equation.

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

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

4.4 Subdivision surfaces

The Catmull-Clark subdivision transformation is used to smooth a surface as the limit of a sequence of subdivision steps[Stam 1998]. This process is governed by properties of B-splines curve from multivariate spline theory[Loop 1987]. This method does a recursive subdivision transformation that refines the model into a linear interpolation that approximates a smooth surface. The smoothness of the model is automatically guaranteed [DeRose et al. 1998].

Catmull-Clark subdivision surface methods generate 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 coarse models with few vertices can generate

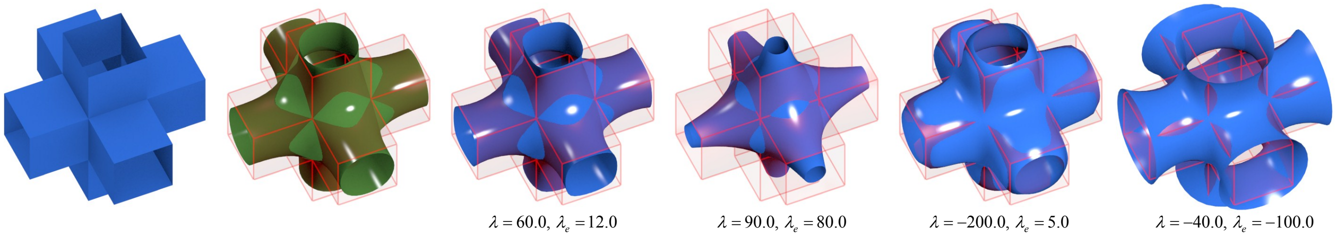
$\lambda = 60.0,\ \lambda_e = 12.0$
$\lambda = 90.0,\ \lambda_e = 80.0$
$\lambda = -200.0,\ \lambda_e = 5.0$
$\lambda = -40.0,\ \lambda_e = -100.0$

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* $\lambda = -40.0$, $\lambda_e = -100.0$.

families of shapes changing a single parameter, this would allow an artist to choose the model from a similar set of options that best meets his/her 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 a coarse model, then to perform a Catmull-Clark subdivision where weights are interpolated, producing weights with smooth changes in the zones of influence, so the obtained curvature is much softer at those areas as shown in figure 5.c.

In the equation 11, W_p is a diagonal matrix with weights corresponding to each vertex. 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 with the extension of the Laplace Beltrami operator for hybrid quad/triangle meshes with several example models (see figures 1, 5, 6, 7, 8, 9, 10, 11, 12). 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 Laplace Beltrami 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 is applied, showing similar results to those composed only of triangles or squares.

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.

Laplacian smoothing applied with simple subdivision may produce similar results to those 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.

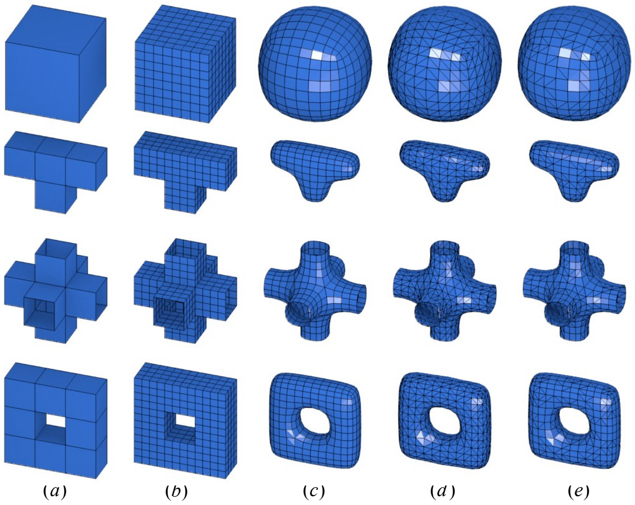

Figure 7: *(a) Original Model. (b) Simple subdivision. (c), (d) (e) Laplacian smoothing with* $\lambda = 7$ *and 2 iterations: (c) for triangles, (d) for quads, (e) for triangles and quads random chosen.*

Figure 8 show the generation of different versions of a camel according to parameter lambda. In the top row you can see results of do curvature enhance over all model, as the lambda becomes larger and negative in this model it inflates the convex parts as shown in figure 1. The larger λ the larger enhancements on the model features. The bottom row of figure 8 shows the use of weighted vertex groups, which allows to specify which areas will be enhanced. On the left the enhancement of the legs of the camel is shown which produces enhancement of organic aspect, notice that the border is not distorted and smooth in the union.

Our method for enhancement of the silhouette features is predictable and invariant under isometric transformations as those present in some animations (see Figure 12). This animation shows some poses of a camel during a walk, enhancement is performed in the neck and legs as shown in the bottom left part of the figure 12. Local modifications produced by pose interpolation or animation rigging do 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, this since depends exclusively on the normal field of the mesh, wish

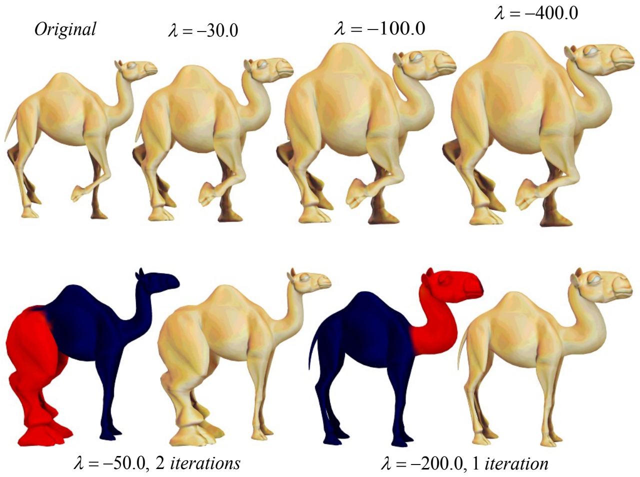
Original
$\lambda = -30.0$
$\lambda = -100.0$
$\lambda = -400.0$
$\lambda = -50.0$, 2 iterations
$\lambda = -200.0$, 1 iteration

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.*

is invariant under global rotations.

Figure 9 shows the use of curvature enhancement brush for sculpting in real time. One pass was used with the brush as shown in the figure with the blue and red radius. In figure 9.b camel shows how the leg self-intersecting and looks like two bubbles stuck, the same happens to the fingers on the bottom of the figure. Using silhouette features enhancement in figure 9.c we get better results sd the are retained shape of the silhouette and the details of the fingers and leg. Similar results can be obtained by an artist however it would take several steps and require the use of several brushes. With curvature enhancement it only takes one step. This new method can easily enhance organic features like muscles during the sculpting process. In figure 10 we show the performance of curvature enhancement brush, in this experiment we use three models with 12K, 40K and 164K vertices, this models were sculpted with curvature enhance brush in each step the brush sphere contain a variable number of vertices for processing. The processing time for 800 vertices in the camel paw (40k model) only take 0.1 seconds, for 2600 vertices in leg and neck (model 40k) take 0.5 seconds, these times are good, because usually an artist sculpts a model for parts, and each part is represented by an average of about 3000 vertices in the models we use.

Tests with the Laplacian operator equation (9) and its normalized version equation (10), produce similar results if the triangles that compose the mesh are the same size on average. The normalized version is more stable and predictable because it is not divided by the area of the ring which may be very small and cause problems during floating point calculations as shown in figure 11.c bottom row. The enhancement of curvature of the model with the normalized Laplacian operator has a more regular behavior. The model can be deformed in TQLBO normalized version with large lambdas ($\lambda > 400$) while it intersects itself it produces no peaks. Figure 11 shows different results due to the area of the triangles in the model. Triangles with larger area have smaller enhancement (figure 11.c skull), and smaller triangles have larger enhancements (figure 11.c chin).

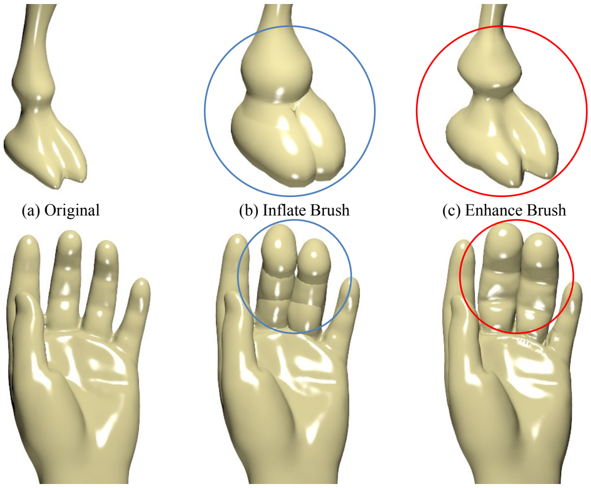

Figure 9: *Top row: (a) Leg Camel, (b) Inflate brush for leg into blue circle, (c) Enhance curvature brush for leg into red circle. Bottom row: (a) Hand, (b) Inflate brush for fingers into blue circle, (c) Enhance curvature brush for fingers in red circle.*

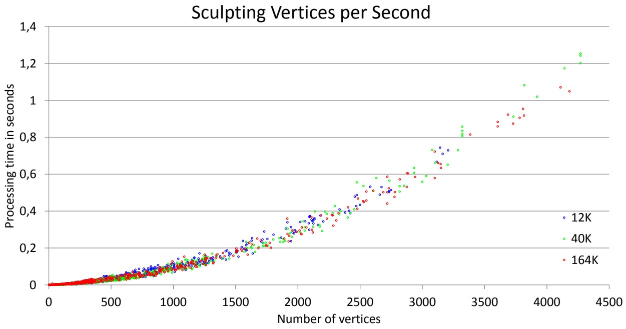

Figure 10: *Performance of our dynamic curvature enhance brush in terms of vertices sculpted per second. Three models with 12K, 40K, 164K vertices used for sculpting in real time.*

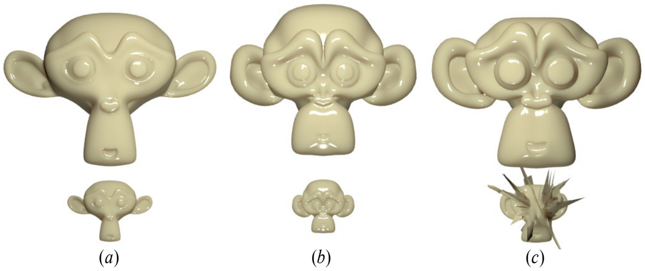

Figure 11: *(a) Top row: Original model scaled by 4. Bottom row: Original Model (b) Top and bottom row: enhancing with Normalized-TQLBO $\lambda = -50$ (c) Top and bottom row: enhancing with TQLBO $\lambda = -50$.*

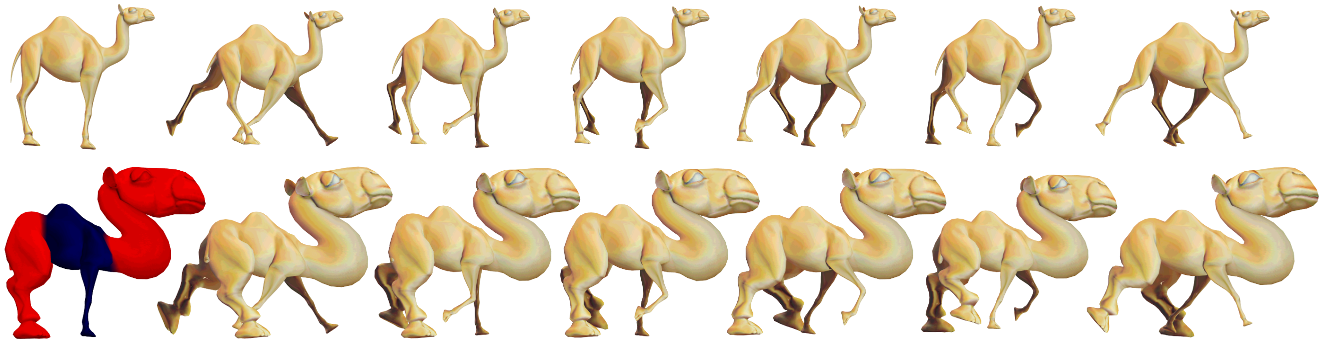

Figure 12: *Our method is pose insensitive. The enhanced for the different poses are similar in terms of shape. Top row: Original walk cycle camel model. Bottom row: Curvature enhancing with weight vertex group, $\lambda = -400$ and 2 iterations.*

5.1 Implementation

Our method was implemented as 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 against others as Catmull-Clark, weight vertex groups and sculpting system in Blender.

To improve the performance, we worked with Blender mesh struct visiting each triangle or quad and adding its corresponding index in a list that stores the sum of the Laplacian weights of the ring, in this way we only had to visit two times the list of faces of the mesh list and two times the list of the edges if the mesh was not closed. The brush sculpting mode required to create a list that maps 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. This drastically reduced the calculations enabling real-time processing. In the construction of our Laplacian matrix several index were locked at vertices having faces areas or lengths edges with value zero that can cause spikes and bad results.

Under this conditions matrix at the equation9 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

This work presented an extension of the Laplace Beltrami operator for hybrid quad/triangle meshes that can be used in production environments without modification that provides results similar to those obtained by working only with triangles or quads.

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

We show that this tool could work in the early stages in where coarse models are used, allowing to modify the curvature generated by the Catmull-Clark subdivision surfaces, avoiding the edition of the vertices with the change of a few parameters.

As future work we 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

We wold like to thank anonymous friends for their support of our research.

This work was supported in part by the Blender Foundation, Google Summer of code program at 2012.

Livingstone elephant model is provided courtesy of INRIA and ISTI by the AIM@SHAPE Shape Repository. Hand model is courtesy of the FarField Technology Ltd. Camel model by Valera Ivanov is licensed under a Creative Commons Attribution 3.0 Unported License. Dinosaur and Monkey models are under public domain, courtesy of Blender Foundation.

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 &amp; 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 &amp; Graphics 36*, 5, 445 – 454.