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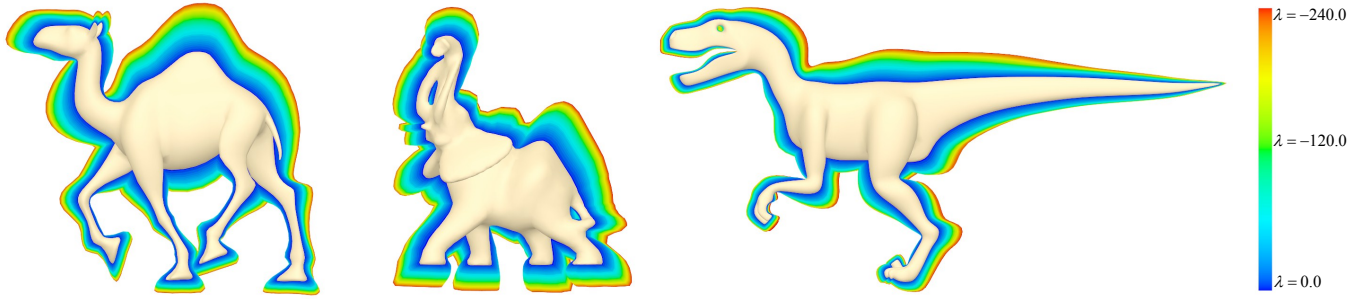

Figure 1: *A set of 48 successive shapes enhanced, from $\lambda = 0.0$ in blue to $\lambda = -240.0$ in red, with steps of -5.0.*

1 Abstract

This paper proposes a novel modeling method for a hybrid quad/triangle mesh that allows to set a family of possible shapes by controlling a single parameter, the global curvature. The method uses an original extension of the Laplace Beltrami operator that efficiently estimates a curvature parameter which is used to define an enhanced shape after a particular operation performed in certain mesh points. 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]. Nevereless 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 shape enhancement and shape exaggeration can thus be used as such brush 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 shape 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 shape 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 inter-relationships 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.

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

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

3.1 Gradient of Voronoi Area

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]*.

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

If the gradient in equation (2) is normalized by the total area of the 1-ring neighborhood around v_i, the *discrete mean curvature normal*
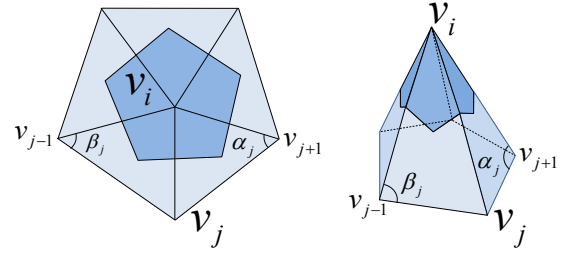

Figure 3: *Area of the Voronoi region around v_i in dark blue. v_j belong to the first neighborhood around v_i. α_j and β_j opposite angles to edge $\overrightarrow{v_j - v_i}$.*

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

This method exaggerates a shape using a Laplacian smoothing operator in the reverse direction, i.e., the new shape is a modified version in which those areas with larger curvature are magnified. The operator amounts to a generator of a set of models which conserves the basic silhouette of the original shape. In addition, the presented approach can be easily mixed with traditional or uniform subdivision of surfaces. This method is based on an original extension of the Laplace Beltrami operator for hybrid quad/triangle meshes, mixing arbitrary types of meshes, exploiting the basic geometrical relationships and ensuring algorithm convergence.

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 $A(v_i)$ corresponds to a sum of the quad faces $A(Q_{v_i})$ and the areas of the triangular faces $A(T_{v_i})$ adjacent to vertex v_i.

$$A(v_i) = A(Q_{v_i}) + A(T_{v_i}).$$

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

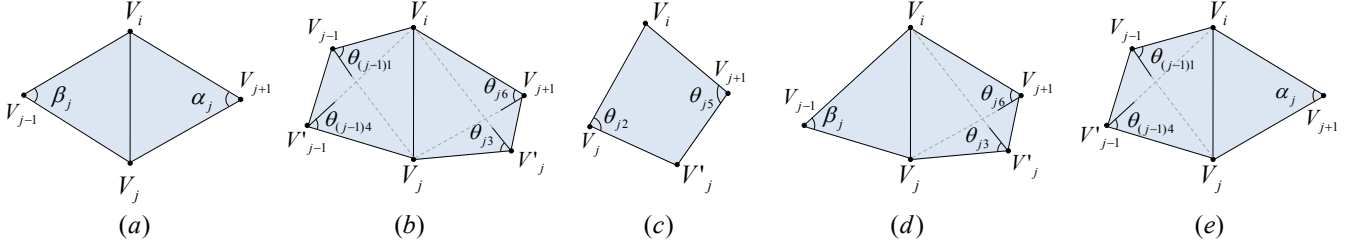$$A(v_i) = \frac{1}{2^m} \sum_{j=1}^{m} 2^{m-1} A(q_j) + \sum_{k=1}^{r} A(t_k)$$

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).*
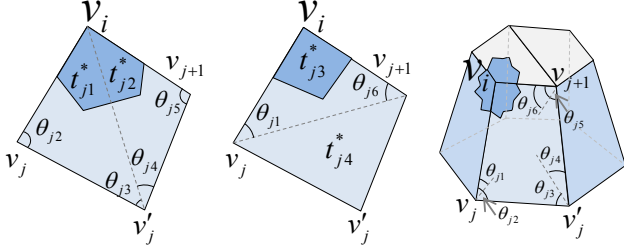

Figure 4: $t^*_{j1} \equiv \triangle \, v_i v_j v'_j$, $t^*_{j2} \equiv \triangle \, v_i v'_j v_{j+1}$, $t^*_{j3} \equiv \triangle \, v_i v_j v_{j+1}$ *Triangulations of the quad with common vertex v_i proposed by [Xiong 2011] to define Mean LBO.*

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

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

Applying the gradient operator to (5).

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

According to (2), we have.

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

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

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

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

Triangle and quad configurations of the 1-ring neighborhood faces, adjacent to v_i, can be simplified to five cases, as shown in figure 2.

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 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), $N(v_i)$ 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

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

Where δ_{ij} being the Kronecker delta function.

4.2 The Shape Enhancement

The shape is enhanced by using the reverse direction of the curvature flow, moving the vertices towards those mesh portions with larger curvature. A standard diffusion process is applied:

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

To solve this equation, implicit integration is used as well as a normalized version of TQLBO matrix.

$$(I - |\lambda dt| W_p L) V' = V^t \quad (11)$$
$$V^{t+1} = V^t + \text{sign}(\lambda)\left(V' - V^t\right)$$

The vertices V^{t+1} are enhanced, along their reverse curvature direction, 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 enhancement factor that supports negative and positive values: negative for enhancement and positive for smoothing.
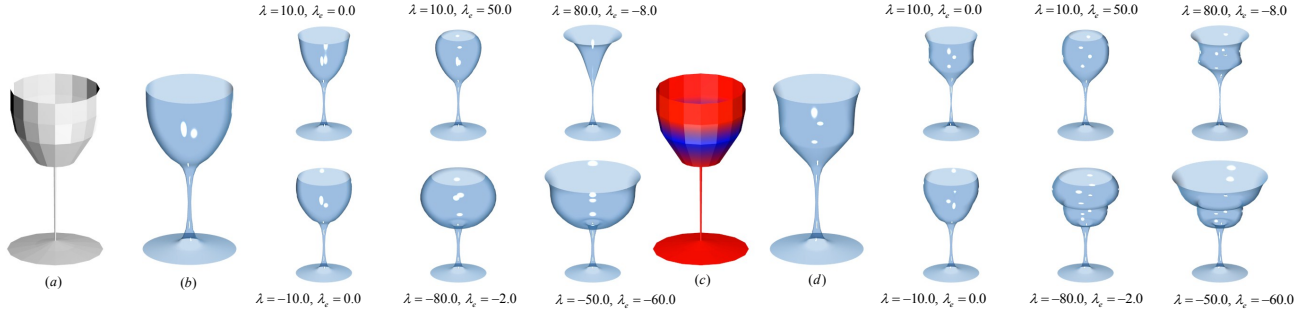
Figure 5: *Family of cups generated with our method, from a coarse model (a), (c): the shape, obtained from the Catmull-Clark Subdivision (b), (d), is enhanced. Soft constraints, over the coarse model, is drawn in red and blue (c).*

The method was devised to use with weighted vertex groups, which specify the final shape enhancement of the solution, meaning 0 as no changes and 1 when a maximal change is applied. The weights modify the influence zones, where the Laplacian is applied, as shown in equation 11. Interestingly, the generated family of shapes may change substantially with the weights of specific control points.

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

The model volume increases as the lambda is larger and negative, this can be counteracted with a simple volume preservation. However, the mesh may suffer large displacements when $\lambda < -1.0$ or after multiple iterations. A simple volume conservation algorithm is: 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 mesh center, vol_{ini} is an initial volume, and vol_{t+1} is the volume at the iteration $t+1$, then the scale factor.

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

allows to move the new vertices to:

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

4.3 Sculpting

A new sculpting brush is herein proposed and aims to inflate the shape, magnifying the shape curvatures of a polygon mesh in real time. This brush works properly with the stroke method *Drag Dot*, allowing to pre-visualize the model changes before the mouse is released. Also, it allows to move the mouse along the model to match the shape zone which is supposed to be enhanced.

Brushes that perform a similar inflation can introduce mesh distortions or produce mesh self-intersections, provided these brushes only move the vertices along the normal without any global information. In contrast, the present method searches for a proper inflation while preserving the global curvature, retaining the original shape and main model features. In addition, this method simplifies the work required for the enhancement since it needs not different brushes for inflating, softening or styling. The enhanced brush can make all these operation in a single step. Real-time brushes require the Laplacian matrix is constructed with the vertices that are within

the sphere radius defined by the user, reducing the matrix to be processed, the center of this sphere depending on the place where the user clicks on the canvas and the three-dimensional mesh placed where the click is projected. Special handling is required for the boundary vertices with neighbors that are not within the brush radius: these vertices are marked as boundary and the curvature is not there calculated, but they must be included in the matrix so that every vertex has their corresponding neighbors within the selection. The sculpting Laplacian matrix reads as.

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

Where $v_j \in N\,(v_i)$, u is the sphere center of radius r. The matrices should remove rows and columns of vertices 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 a B-spline curve [Loop 1987], performing a recursive subdivision transformation that refines the model into a linear interpolation that approximates a smooth surface. The model smoothness is automatically guaranteed [DeRose et al. 1998].

Catmull-Clark subdivision surface methods generate smooth and continuous models from a coarse model and produce quick results because of the simplicity of implementation. Nevertheless, changes to the global curvature are hardly implantable. The Catmull-Clark subdivision surfaces together with shape enhancement can easily generate families of shapes by changing a single parameter, allowing to handle a model with very few vertices. In practice, this would allow an artist to choose a model from a similar set of options that would meet his/her needs without having to change each of the control vertices. Likewise, the presented method allows the use of vertex weight paint over the control points. The weights can be applied to a coarse model, followed by a Catmull-Clark subdivision where weights are interpolated, producing weights with smooth changes in the influence zones, as shown in figure 5.c.

In equation 11, W_p is a diagonal matrix with weights corresponding to each vertex. Weights at each vertex produce a different solution so that the matrix must be placed in the diffusion equation since families that are generated may change substantially with weighted of specific control points.
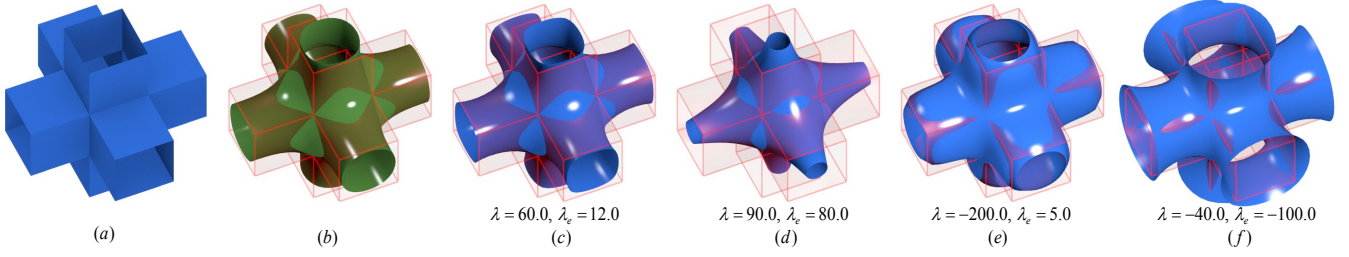
$\lambda = 60.0, \lambda_e = 12.0$ $\lambda = 90.0, \lambda_e = 80.0$ $\lambda = -200.0, \lambda_e = 5.0$ $\lambda = -40.0, \lambda_e = -100.0$

(a)   (b)   (c)   (d)   (e)   (f)

Figure 6: *(a) Original Model, (b) Model with Catmull-Clark Subdivision. Models with Laplacian smoothing: (c) $\lambda = 60.0$, $\lambda_e = 12.0$ and (d) $\lambda = 90.0$, $\lambda_e = 80.0$. Models with a first Laplacian filtering $\lambda = 60.0$, $\lambda_e = 12.0$ and before applying shape enhancement: (e) $\lambda = -200.0$, $\lambda_e = 12.0$ and (f) $\lambda = -40.0$, $\lambda_e = -100.0$.*

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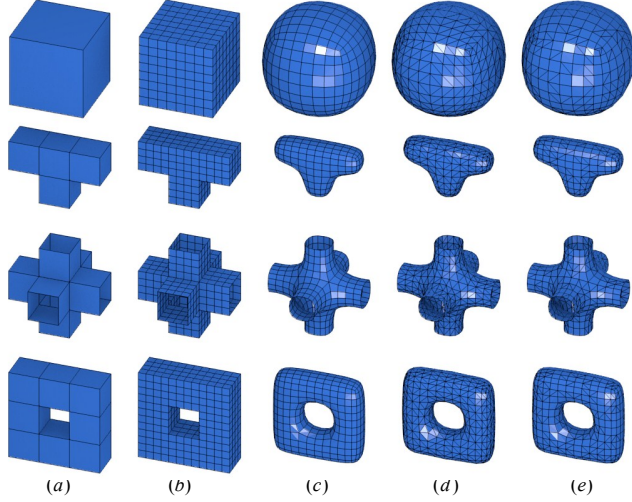
Figure 7: *(a) Original Model. (b) Simple subdivision. (c), (d) (e) Laplacian smoothing with $\lambda = 7$ and 2 iterations: (c) for triangles, (d) for quads, (e) for triangles and quads random chosen.*
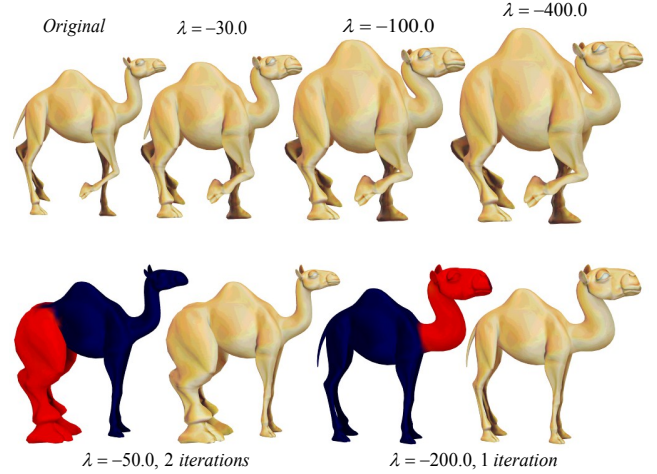

Original $\lambda = -30.0$ $\lambda = -100.0$ $\lambda = -400.0$

$\lambda = -50.0$, *2 iterations* $\lambda = -200.0$, *1 iteration*

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

5 Results

The results of the shape enhancement 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). The shape enhancement was assessed with TQLBO method on a PC with AMD Quad-Core Processor @ 2.40 GHz and 8 GB RAM.

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

Methods using the 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 cup model, in which the subdivision was performed, followed by a Laplacian smoothing and enhancement. In figure (5).c, (5).d shows also the use of 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 on the 6 cups that are at 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 in average equal triangles. Figure 6.b. shows Catmull-Clark subdivision, 6.c. shows Laplacian smoothing. The one obtained with the Laplacian smoothing is shown in panel (c), (d) and those curvature modified versions are in (e) and (f). As can be observed, different versions of the original sketch can be obtained by parameterizing a single model value, a great advantage of the presented method. Figure 8 shows the generation of different versions of a camel according to the λ parameter. In the top row, it is shown the shape enhancement results, as λ becomes larger and negative, the resultant shape is observed as if the model would inflate the more convex parts, as shown in figure 1. The larger the λ parameter the larger the model feature enhancements. The bottom row of figure 8 shows the use of weighted vertex groups, specifying which areas will be enhanced. On the left, the enhancement of the camel legs produces an organic aspect, notice that the border is not distorted and smooth.

The enhancement of the silhouette features is predictable and invariant under isometric transformations, as those classically used in some animations (see Figure 12). In this figure, the animation shows some camel poses during a walk, the enhancement is performed at the neck and legs, as shown in the bottom left camel in figure 12. Local modifications produced by the pose interpolation

or animation rigging practically do not affect the result. In spite of at any pose of the camel legs there is a clear difference, the enhancement method allows a flesh-like shape in the original pattern produced by the artist. This is due to the mesh restricted diffusion process so that small local changes are treated without affecting the global solution. The method therefore is rotation invariant since it depends exclusively on the normal mesh field, which is rotation invariant.

Figure 9 shows the use of a shape 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 the camel foot shows the inflation intersection that looks like two bubbles, a similar pattern to what is observed to the fingers on the bottom of the same figure. The silhouette enhancement is observed in figure 9.c since the main shape is retained together with its finger and foot details. Similar results can be obtained by a user, however it would take several steps and require the use of several brushes, while the shape enhancement took a single step. Likewise, this new method can easily enhance organic features like muscles during the sculpting process. In figure 10 the shape enhancement brush performance is illustrated, in this experiment three models with 12K, 40K and 164K vertices, were used. These models were sculpted with the shape enhance brush, at each step the brush sphere containing a variable number of vertices for processing. The processing time for 800 vertices in the camel paw (40k model) only took 0.1 seconds, for 2600 vertices in the leg and neck (model 40k) took 0.5 seconds, these times are suitable in real applications since an artist sculpts a model for parts, and each part is represented by an average of about 1800 vertices.

Tests with the Laplacian operator (equation 9) and its normalized version (equation 10), produce similar results if the triangles that compose the mesh are about the same size. 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 causes numeric problems, as shown in figure 11.c bottom row. The shape enhancement of the model with the normalized Laplacian operator results in a more regular pattern. The model can be deformed with a TQLBO normalized version with large λ ($\lambda > 400$) while intersecting itself with no peaks. Figure 11.c shows different results due to the triangle areas in the model. Triangles with larger area have smaller enhancement (figure 11.c skull), and smaller triangles have larger enhancements (figure 11.c chin).

5.1 Implementation

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

To improve the performance, it was worked with the Blender mesh struct, visiting each triangle or quad and storing its corresponding index and the sum of the Laplacian weights of the ring in a list so that only two visits were required for the list of mesh faces and two times the edge list, if the mesh was not closed. This drastically reduced calculations, enabling real-time processing. In the construction of the Laplacian matrix, several index were locked at vertices having face areas or edge lengths with null value that could cause spikes and bad results.

Under these conditions, the matrix at equation 9 is sparse since the number of neighbors per vertex, corresponding to the number of data per row, is smaller 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.
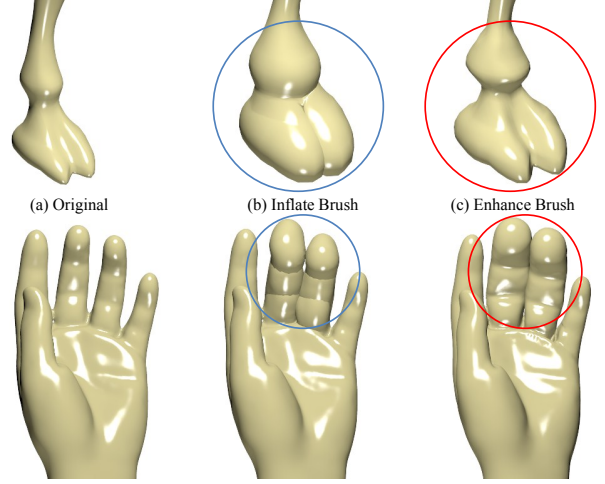

Figure 9: *Top row: (a) Leg Camel, (b) Inflate brush for leg into blue circle, (c) Enhance shape brush for leg into red circle. Bottom row: (a) Hand, (b) Inflate brush for fingers into blue circle, (c) Shape enhancement brush for fingers in red circle.*
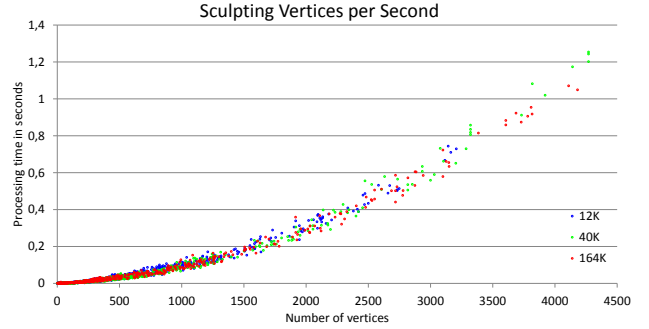

Figure 10: *Performance of our dynamic shape enhancement brush in terms of the sculpted vertices per second. Three models with 12K, 40K, 164K vertices used for sculpting in real time.*
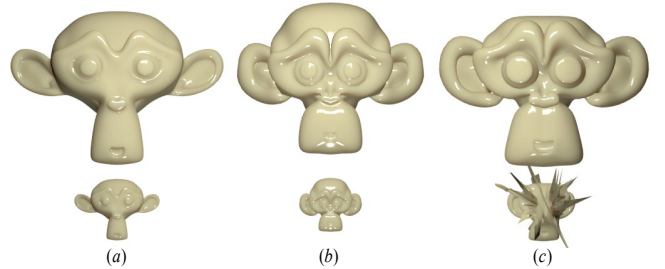

Figure 11: *(a) Top row: Original model scaled by 4. Bottom row: Original Model (b) Top and bottom row: enhancing with Normalized-TQLBO $\lambda = -50$ (c) Top and bottom row: enhancing with TQLBO $\lambda = -50$.*
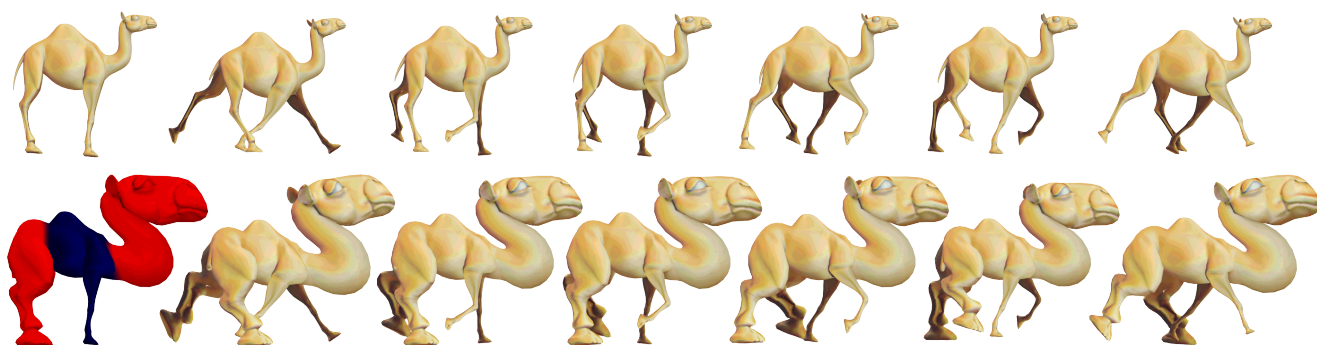
Figure 12: *The method is pose insensitive. The enhancement for the different poses are similar in terms of shape. Top row: Original walk cycle camel model. Bottom row: Shape enhancement with weight vertex group, $\lambda = -400$ and 2 iterations.*

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 and provides results similar to those obtained by working only with triangles or quads. This paper has introduced a new way to change silhouettes in a mesh for modeling or sculpting in a few steps by means of the curvature model modification while preserving its overall shape. In addition, a new modeling method has also been presented some possible applications have been illustrated. The method works properly with isometric transformations, opening the possibility of introducing it on the process of animation.

We show that this tool may work in early modeling stages, case in which coarse models are used, allowing to modify the shape generated by the Catmull-Clark subdivision surfaces, and thereby avoiding edition of the vertices with change of a single parameters.

Future work includes the analysis of theoretical relationships between the Catmull-Clark subdivision surfaces and the Laplacian smoothing since they can produce very similar results.

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 SIG-GRAPH 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.