Adapted Laplacian Operator For Hybrid Quad/Triangle Meshes

Alexander Pinzón Fernández

Universidad Nacional de Colombia

Facultad de Ingenieria, Departamento de Sistemas e Industrial

Grupo de Investigación CIM@LAB

Bogotá, Colombia

2014

Adapted Laplacian Operator For Hybrid Quad/Triangle Meshes

Alexander Pinzón Fernández

A thesis submitted in partial fulfillment of the requirements for the degree of:

Master in Systems Engineering and Computer Science

Advisor:
Eduardo Romero Castro , Ph.D.

Research Area:
Computer Graphics

Universidad Nacional de Colombia
Facultad de Ingenieria, Departamento de Sistemas e Industrial
Grupo de Investigación CIM@LAB
Bogotá, Colombia
2014

Dedicación

A Beatriz y Campo Elias mis padres que siempre me dieron la libertad de escoger, con su comprensión y apoyo me permitieron dedicar mi tiempo a la ciencia.

A mis padres

Inteligente es aquella persona que sabe ser feliz.
Alexander Pinzón Fernández

Acknowledgement

I wold like to thank my advisor professor Eduardo Romero and CIM&LAB research group for his support in this thesis.

I would like to thank all the workers in Colombia who fund the public education with their work.

This work was supported in part by the Blender Foundation, Google Summer of code program at 2012 - 2013.

Abstract

In the last two decades three-dimensional modeling methods have been evolving and developing rapidly, thanks in part to the use of vector operators of differential geometry. The Laplacian operator has been one of the most widely studied and used thanks to the properties exhibited by their eigenvectors in novel applications such as noise reduction, enhancement, remeshing, UV mapping, posed, skeletonization, among other. The differential operator is defined in a continuous and smooth domain named manifold, manifolds are often approximated by discrete polygon meshes composed by triangles and quadrangles which represent the real world three-dimensional objects. In these meshes spectral structure is calculated using a discrete Laplacian operator, the discrete version of the Laplacian operator given by Pinkall in 1993 works only with meshes of triangles, and Xiong in 2011 working exclusively with quads. This thesis proposes an original extension of the Laplacian operator that allows to work with hybrid meshes composed by triangles and quadrangles.

Along with the operator, this work presents new sculpting and modeling applications with base on the enhancement, applications on subdivision surfaces using smoothing, mesh posing using differential coordinates and skeletonization using iterative contraction. This series of applications demonstrates the quality, predictability and flexibility of the proposed operator.

The proposed operator was successfully used in real production environments within software Blender how new tools. Currently these tools are available as open source software.

Resumen

En las dos últimas décadas los métodos de modelado tridimensional han ido evolucionando y desarrollándose rápidamente, en parte gracias al uso de operadores vectoriales de geometría diferencial. El operador Laplaciano ha sido uno de los más ampliamente estudiado y usado gracias a las propiedades exhibidas por sus vectores propios en novedosas aplicaciones como: reducción de ruido, realce, remallado, mapeo UV, posado, esqueletonización, entre otras. Este operador diferencial es definido en un dominio continuo y suave llamado variedad, las variedades son a menudo aproximadas por mallas discretas de polígonos compuestas por triángulos y cuadrángulos que a su vez representan objetos tridimensionales del mundo real. En estas mallas se calcula la estructura espectral con el uso de algún operador Laplaciano discreto, la versión discreta del operador Laplaciano propuesta por Pinkall en el 1993 trabaja únicamente con mallas compuestas por triángulos, y la de Xiong en el 2011 trabaja exclusivamente con cuadrángulos. Esta tesis propone una extensión original del Operador Laplaciano que permite trabajar con mallas híbridas compuestas por triángulos y cuadrángulos.

Junto con el operador, este trabajo presenta nuevas aplicaciones en esculpido y modelamiento con base en el realce, aplicaciones en subdivisión de superficies con el uso de suavizado, posado de mallas con el uso de coordenadas diferenciales y esqueletonización usando contracción iterativa. Esta serie de aplicaciones demuestra la calidad, predictibilidad y flexibilidad del operador propuesto.

El operador propuesto fue usado de forma exitosa en ambientes reales de producción dentro del software Blender como nuevas herramientas. Actualmente estas herramientas están disponibles como programas de código abierto.

Keywords: laplacian operator; smooth; enhance; sculpting; subdivision surface

Contents

List of Figures

List of Tables

1 Introduction

The discrete versions of Laplace Beltrami Operator have been used in the last years for the develop of new geometric modeling tools. In the work of Pinkall [22] was introduced the cotangent version of Laplace operator, that allow to find the minimal surface when compute a discete harmonic map with the Laplacian operator, this version has been widely studied and applied in various problems of computer geometric modeling. The manifolds that are continuous domains homeomorphic to $\mathbb{R}^n$ are represented in computers by polygon meshes. These polygons are generally composed of triangles and quadrangles, while work with laplacian operator in this hybrid composition are not a mathematical challenge, most research uses only meshes composed by triangles [22, 11, 19, 25, 1, 2, 15], in other recent studies [17, 30] the Laplacian operator is working exclusively with quadrangles. But in the artistic scope topology and the way the edges are distributed, the triangle and quadrangles, directly affects the processes of animation, interpolation, textured, etc. As discussed by [20] who use a manual connection of a couple of vertices to perform animation processes and interpolation. 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 and change the original topology.

Modeling techniques able to generate a variety of realistic shapes, have been developed [5]. Editing techniques have evolved from affine transformations to advanced tools like sculpting [8, 13, 29], editing, creation from sketches [16, 14], and complex interpolation techniques [25, 31]. 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 [6, 28]. 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 [10], or the modification of some vertex weights to locally control the zone of influence [3]. Neverteless these methods are difficult to deal with since they require a large number of parameters and a very tedious customization. Instead, the presented applications 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.

The shape inflation 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 [29]. In contrast, the presented enhance 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 inflating the silhouette mesh features in modeling and sculpting.

This work is organized as follows: Chapter 2 presents works related to the Laplacian operator, applications in digital sculpting, deformation, and offsetting methods for polygonal meshes; In chapter 3, it is described the theoretical framework of the Laplacian operator for polygon meshes; In chapter 4, it is presented the extension of the Laplacian Operator for hybrid meshes and applications of shape inflation ,subdivision of surfaces , deformation, skelenotization and sculpting; finally some Laplacian Operator results, to hybrid quad/triangle meshes are graphically shown as well as results of the shape inflation applications in sculpting, subdivision, deformation, skelenotization 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 applications [23].

The most used discretization of Laplace Beltrami operator was proposed by Pinkall et. al. in 1993 [22]. In this work the discretize laplacian operator was used to find the minimal surface based on minimization energy strategy using the Dirichlet's energy of the function u over a manifold represented by triangulated mesh Ω.

$$E_D(u) = \frac{1}{2}\int_\Omega \left|\nabla u\right|^2$$

In the work of Desbrun et. al. in 1999 [11] a curvature normal ($\bar{\kappa}\mathbf{n}$) calculation based on cotangen Laplace operator version is proposed

$$-\bar{\kappa}\mathbf{n} = \frac{1}{A}\sum_{j\in N_1(i)}(\cot\alpha_j + \cot\beta_j)$$

Where A is the area surrounding vertex i, $N_1$is the 1-ring neighboorhood, α and β are the opposite angles to edge between vertex i and vertex j. This laplacian operator L was used for reduce the noise in a mesh Ω over a difussion process.

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

In the work of Sorkine [26] the laplacian operator was used to repose a meshe while preserving geometry details of the surface. The details was stored in differential coordinates δ_i for every vertex v_i.

$$\delta_i = \sum_{j\in N_1(i)}\frac{1}{2}(\cot\alpha_j + \cot\beta_j)(v_i - v_j)$$

The differential coordinates represents the difference between the absolut coordinate of $v_i$and the center of mass of its immediate neighboors.

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. Nevertheless, these methods fail to conserve sufficient detail

because of the smoothing, a crucial issue which depends on the offset size [32]. 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 [7].

[12] propose automatic feature detection and shape edition with feature inter-relationship preservation. They define salient surface features like ridges and valleys, characerized by their first and second order curvature derivatives, see [21], 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 [29]. 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 [13]. The drawback comes from the difficulty of maintaining the surface details during larger scale deformations.

Skeleton extraction methods permits simplify the dimension of the object preserving the topological structure [9], Au et. al. [1] present a skeleton extraction method based on iterative smooothing-contraction, in this method the several constraints are used to waranty that the process converge to skeleton formed by branchs and joins, the constraints based on laplacian operator, the low frequencies of the mesh are preserving with the use of attractor to original mesh, while the iterative smoothin process remove high frecuencies.

3 Mathematical foundation

This chapter studies bascic mathematical foundation on differential geometry to understand the differential operators and the laplace beltrami opetor.

The differential geometry studies curvatures and geodesics [15]. The differential operators show a deep relationships betwen the geoemetry (curvatures, geodesics) and topology of the manifold ——and this properties are awsome applications on the last yesr abrand studies with this opertator[24]

A manifold is a topological space M with the follwing properties:

If $x \in M$, then there is some neighborhood $N(x)$ and some integer $n \geq 0$ such that $N(x)$ is homeomorphic to $\mathbb{R}^n$ [27].

3.1 Laplace Operator

In computer graphics a manifold is often approximated by a discrete mesh [23], then is necessary to define a discrete laplacian operator that act on functions defined on such meshes.

Consider a smooth compact manifold M of dimension m isometrically embedded in a Euclidean space $\mathbb{R}^d$.

Given a twice continuously differentiable function $f \in C^2(M)$, let $\nabla_{\mathrm{M}} f$ denote the gradient vector field of f on M.

The Laplace-Beltrami operator Δ_{M} of f is defined as the divergence of the gradient; that is [27],

$$\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial z^2} = 0$$

$$\nabla^2_{\mathrm{M}} f = \Delta_{\mathrm{M}} f = 0$$

$$\Delta_{\mathrm{M}} f = div\left(\nabla_{\mathrm{M}} f\right) \tag{3-1}$$

3.1.1 Discrete Laplace Operator Setting

Discrete Laplacian operators are linear operators that act on functions defines on meshes. These functions are specified by their values at the vertices.

Thus, if a mesh M has n vertices, then functions on M will be represented by vectors with n components and a mesh Laplacian will be described by an $n \times n$ matrix [23].

The Laplacian operator locally takes the difference between the value of a function at a vertex and a weighted average of its values at the first-order or one-ring neighbor vertices, then a Laplacian matrix L has a local form given by

$$L(f)_i = b_i^{-1} \sum_{j \in N(i)} w_{ij} (f_i - f_j)$$

Where $w_{ij}$are the weights between the vertex i and the vertex j. b_i^{-1} are the factors depending of the boundary region over vertex i. $N(i)$ are the neighbors that share a edge with vertex i.

3.1.2 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 **3-1**, The area change produced by the movement of v_i is called the gradient of *Voronoi region* [22, 11, 19].

$$\nabla A = \frac{1}{2} \sum_j (\cot \alpha_j + \cot \beta_j)(v_i - v_j) \tag{3-2}$$

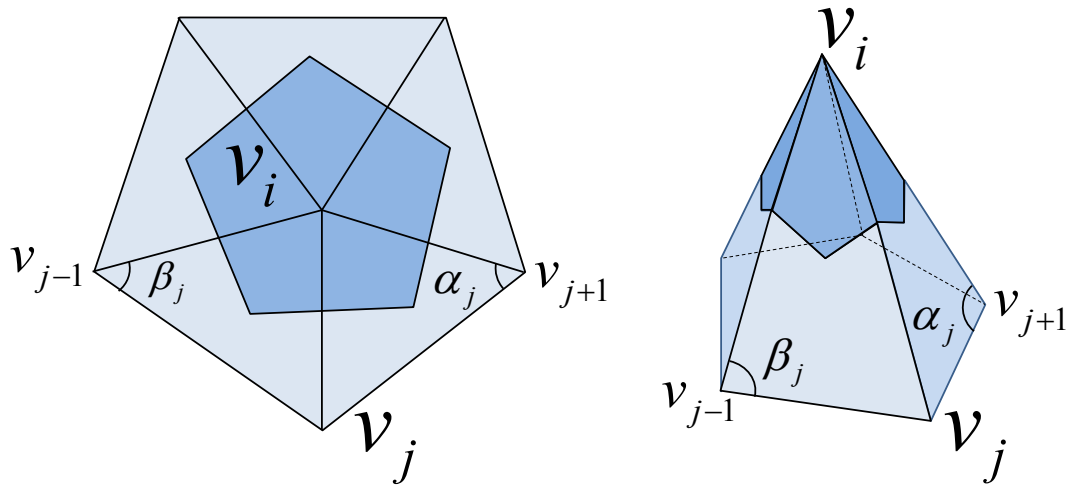

Figure **3-1**: Area of the Voronoi region around v_i in dark blue.v_j belong to the first neighborhood around v_i. α_j and β_j opposite angles to edge $\overrightarrow{v_j - v_i}$.

If the gradient in equation 3-2 is normalized by the total area of the 1-ring neighborhood around v_i, the *discrete mean curvature normal* of a surface S is obtained, as shown in equation 3-3.

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

3.2 Laplace Beltrami Operator

The *Laplace Beltrami operator* LBO noted as Δ is used for measuring the mean curvature normal to the Surface S [22].

$$\Delta_S = 2\kappa\mathbf{n} \tag{3-4}$$

4 Proposed Laplacian Operator

This thesis propose 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.

With the use of the proposed Laplacian Operator this work show succesful uses in: Smoothing, Enhancing, Smooth Subdivision, Deformation, Sculpting, and Skeletonization.

4.1 Proposed 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 areas of the quad faces $A(Q_i)$ and the areas of the triangular faces $A(T_i)$ adjacent to vertex v_i.

$$A(v_i) = A(Q_i) + A(T_i)$$

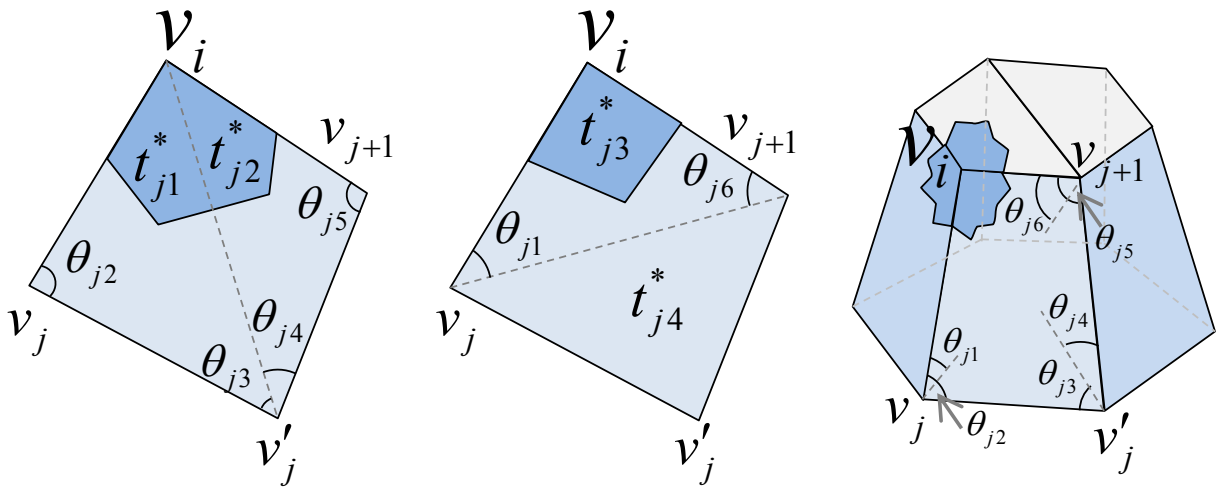

Figure **4-1**: $t^*_{j1} \equiv\triangle\ v_i v_j v'_j$, $t^*_{j2} \equiv\triangle\ v_i v'_j v_{j+1}$, $t^*_{j3} \equiv\triangle\ v_i v_j v_{j+1}$ Triangulations of the quad with common vertex v_i proposed by [Xiong 2011] to define Mean LBO.

Applying the mean average area, according to Xiong et. al. [30], from all possible triangulations, as show in figure **4-1**, the area for quads $A(Q_i)$ and triangles $A(T_i)$ is.

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

Where $q_1, q_2, ..., q_j, ..., q_m \in Q_i$ are quads adjacent to v_i, and $t_1, t_2, ..., t_k, ..., t_r \in T_i$ are triangles adjacent to v_i.

Applying triangulations of the quad with common vertex v_i proposed by [30].

$$A(v_i) = \frac{1}{2}\sum_{j=1}^{m} \left[A\left(t_{j1}^{*}\right) + A\left(t_{j2}^{*}\right) + A\left(t_{j3}^{*}\right)\right] + \sum_{k=1}^{r} A(t_k) \tag{4-1}$$

Applying the gradient operator to (4-1).

$$\nabla A(v_i) = \frac{1}{2}\sum_{j=1}^{m} \left[\nabla A\left(t_{j1}^{*}\right) + \nabla A\left(t_{j2}^{*}\right) + \nabla A\left(t_{j3}^{*}\right)\right] + \sum_{k=1}^{r} \nabla A(t_k) \tag{4-2}$$

According to (3-2), we have.

$$\nabla A\left(t_{j1}^{*}\right) = \frac{\cot\theta_{j3}(v_j - v_i) + \cot\theta_{j2}\left(v'_j - v_i\right)}{2}$$

$$\nabla A\left(t_{j2}^{*}\right) = \frac{\cot\theta_{j5}\left(v'_j - v_i\right) + \cot\theta_{j4}(v_{j+1} - v_i)}{2}$$

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

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

Triangle and quad configurations of the 1-ring neighborhood faces, adjacent to v_i, can be simplified to five cases, as shown in figure **4-2**.

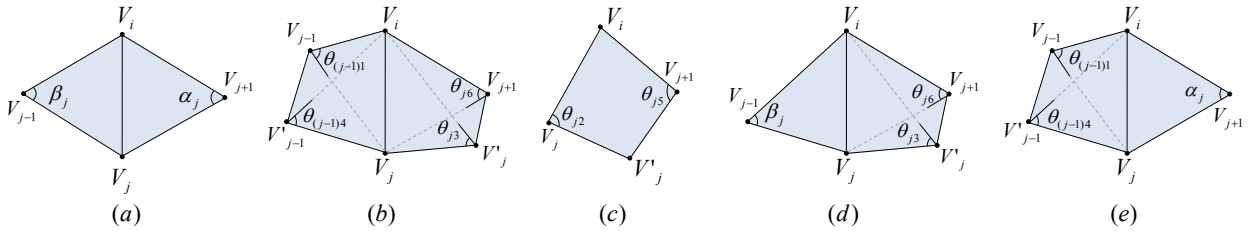

Figure **4-2**: The 5 basic triangle-quad cases with common vertex V_i and the relationship with V_j and V'_j. (a) Two triangles [Desbrun 1999]. (b) (c) Two quads and one quad [Xiong 2011]. (d) (e) Triangles and quads (TQLBO) our contribution.

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

$$\triangle_S(v_i) = 2\kappa\mathbf{n} = \frac{\nabla A}{A} = \frac{1}{2A}\sum_{v_j \in N_1(v_i)} w_{ij}(v_j - v_i) \tag{4-3}$$

where

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

We define a Laplacian operator as a matrix equation

$$L(i,j) = \begin{cases} -\frac{1}{2A_i} w_{ij} & \text{if } j \in N(v_i) \\ \frac{1}{2A_i} \sum\limits_{j \in N(v_i)} w_{ij} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \tag{4-5}$$

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 (4-4), $N(v_i)$ 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

$$L(i,j) = \begin{cases} -\frac{w_{ij}}{\sum\limits_{j \in N(v_i)} w_{ij}} & \text{if } j \in N(v_i) \\ \delta_{ij} & \text{otherwise} \end{cases} \tag{4-6}$$

Where δ_{ij} being the Kronecker delta function.

4.2 Laplacian Smooth

Computer objects, reconstructed from the real world, are usually noisy. One common way to attenueate noise in a mesh is through a diffusion process [11].

$$\frac{\partial V}{\partial t} = \lambda L(V) \tag{4-7}$$

This equation 4-7 can be linearly approximated using implicit integration with a Laplacian Operator version of TQLBO.

$$(I - \lambda \partial t L) V^{n+1} = V^n$$

Laplacian Smooth techniques over a diffusion process 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.

4.3 The Shape Inflation

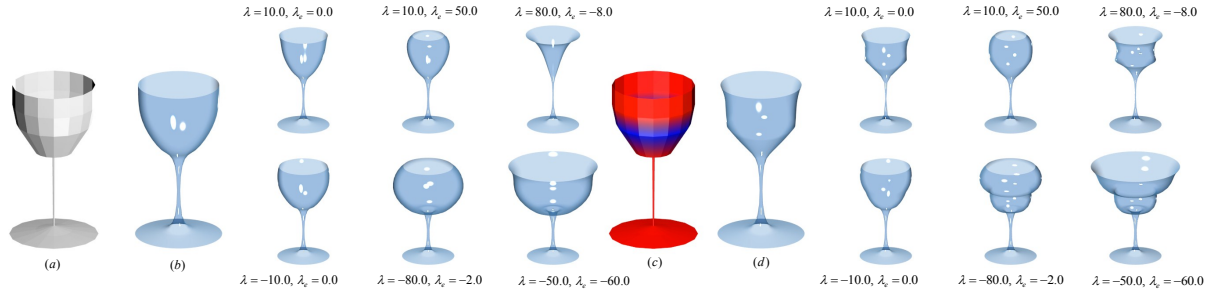

Figure **4-3**: Family of cups generated with our method, from a coarse model (a), (c): the shape, obtained from the Catmull-Clark Subdivision (b), (d), is inflated. Soft constraints, over the coarse model, is drawn in red and blue (c).

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.

The shape is inflated 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:

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

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

$$\left(I - \left|\lambda dt\right| W_p L\right) V' = V^t \tag{4-8}$$

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

The vertices V^{t+1} are inflated, along their reverse curvature direction, by solving the linear system: $Ax = b$, where $A = I - \left|\lambda dt\right| W_p L$, L is the Normalized TQLBO defined in the equation (4-6), $x = V'$ are the smoothing vertices, $b = V^t$ are the actual vertices positions,

W_p is a diagonal matrix with vertex weights, and λdt is the inflate factor that supports negative and positive values: negative for inflation and positive for smoothing.

The method was devised to use with weighted vertex groups, which specify the final shape inflation 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 4-8. 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. [11], the inflation 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:

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

$\overline{v}$ 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.

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

allows to scale the vertices to:

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

The shape inflation use a negative curvature flow that is an unstable process when performing many iterations, however, our method uses less than 3 iterations to get good results, and with 3 iterations or less the method behaves stable.

4.4 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 inflated.

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 inflation since it needs not different brushes for inflating, softening or styling. The inflated 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.

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

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

4.5 Laplacian Deform

The mesh editing is generally done with affine transformations, some 3D softwares like Blender[4] offers some tools to interpolate the transform of the vertices over connected neighbors ussing simple distance functions.

Esta tesis adapto el metodo propuesto por osikine para deformaciones de malla elminando el uso de vertices estaticos y su uso sobre mallas hibridas compuestas por triangulosy cuadranglos con el operador TQLBO propuesto.

The This project proposes to implement a method for mesh editing based on sketching lines defines by user and preserving the geometric details of the surface.

This method captures the geometric details using a differential coordinates representations. The differential coordinates captures the local geometric information (curvature and direction) of the vertex based on its neighbors. This method allows you to retrieve the best possible original model after changing the positions of some vertices using the differential coordinates of the original model.

This method also allows the user to define the desired transformation using sketch lines in the screen.

4.6 Subdivision surfaces

The Catmull-Clark subdivision transformation is used to smooth a surface, as the limit of a sequence of subdivision steps [28]. This process is governed by a B-spline curve [18], performing a recursive subdivision transformation that refines the model into a linear interpolation that approximates a smooth surface. The model smoothness is automatically guaranteed [10].

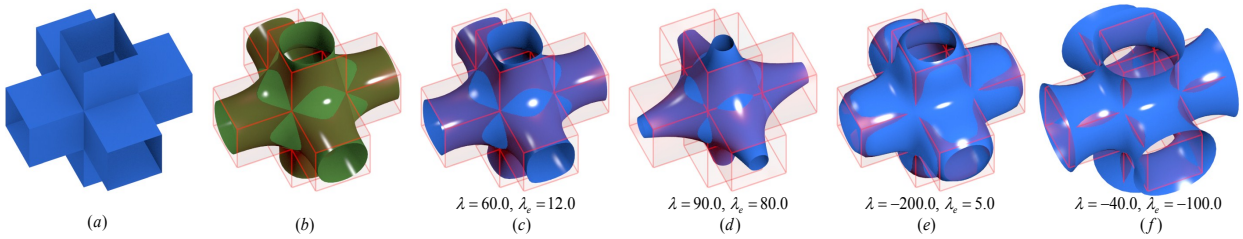

Figure **4-4**: (a) Original Model, (b) Model with Catmull-Clark Subdivision. Models with Laplacian smoothing: (c) and (d). Models with a first Laplacian filtering $\lambda = 60.0$, $\lambda_e = 12.0$ and before applying shape inflation: (e) and (f).

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 inflation 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 **4-3**.c.

In equation 4-8, 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.

4.7 Skeleton Extraction

5 Results

The results of the shape inflation method with the extension of the Laplace Beltrami operator for hybrid quad/triangle meshes with several example models (see figures **5-1**, **4-3**, **4-4**, **5-2**, **5-3**, **5-5**, **5-6**, **5-7**, **5-4**). The shape inflation was assessed with TQLBO method on a PC with AMD Quad-Core Processor @ 2.40 GHz and 8 GB RAM.

Figure **5-2** shows the results when applying the Laplace Beltrami Operator TQLBO of equation (4-5) 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 inflation allows to modify the curvature that is obtained with the process of subdivision, as shown in figure (**4-3**). This test used a coarse cup model, in which the subdivision was performed, followed by a Laplacian smoothing and inflation. In figure (**4-3**).c, (**4-3**).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 inflation obtained on the 6 cups that are at the right of the figure (**4-3**).d.

Laplacian smoothing applied with simple subdivision (see figure **4-4**.b.) may produce similar results to those obtained with Catmull-Clark (see figure **4-4**.c.), whose models are in average equal triangles. 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 **5-3** shows the generation of different versions of a camel according to the λ parameter. In the top row, it is shown the shape inflation 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 **5-1**. The larger the λ parameter the larger the model feature inflations. The bottom row of figure **5-3** shows the use of weighted vertex groups, specifying which areas will be inflated. On the left, the inflation of the camel legs produces an organic aspect, notice that the border is not distorted and smooth.

The inflation of the silhouette features is predictable and invariant under isometric transformations, as those classically used in some animations (see Figure **5-4**). In this figure, the

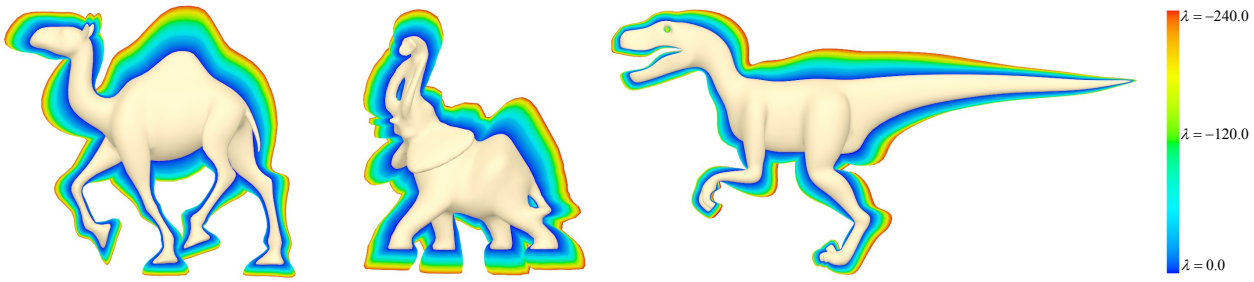

Figure **5-1**: A set of 48 successive shapes enhanced, from $\lambda = 0.0$ in blue to $\lambda = -240.0$ in red, with steps of -5.0.

animation shows some camel poses during a walk, the inflation is performed at the neck and legs, as shown in the bottom left camel in figure **5-4**. 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 inflation 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 **5-5** shows the use of a shape inflation 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 **5-5**.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 inflation is observed in figure **5-5**.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 inflation took a single step. Likewise, this new method can easily inflate organic features like muscles during the sculpting process. In figure **5-6** the shape inflation brush performance is illustrated, in this experiment three models with 12K, 40K and 164K vertices, were used. These models were sculpted with the shape inflate 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.

Tests with the Laplacian operator (equation 4-5) and its normalized version (equation 4-6), produce similar results if the triangles or quads 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 **5-7**.c bottom row. The shape inflation 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 **5-7**.c shows

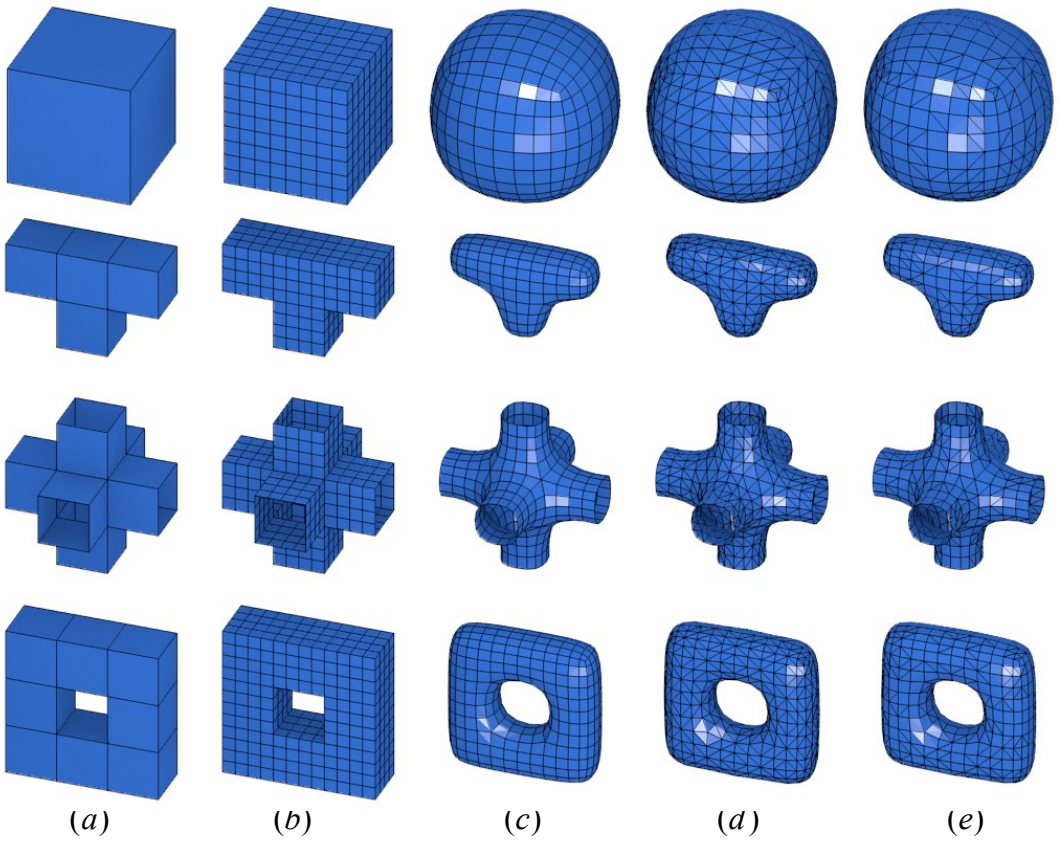

Figure **5-2**: (a) Original Model. (b) Simple subdivision. (c), (d) (e) Laplacian smoothing with $\lambda = 7$ and 2 iterations: (c) for triangles, (d) for quads, (e) for triangles and quads random chosen.

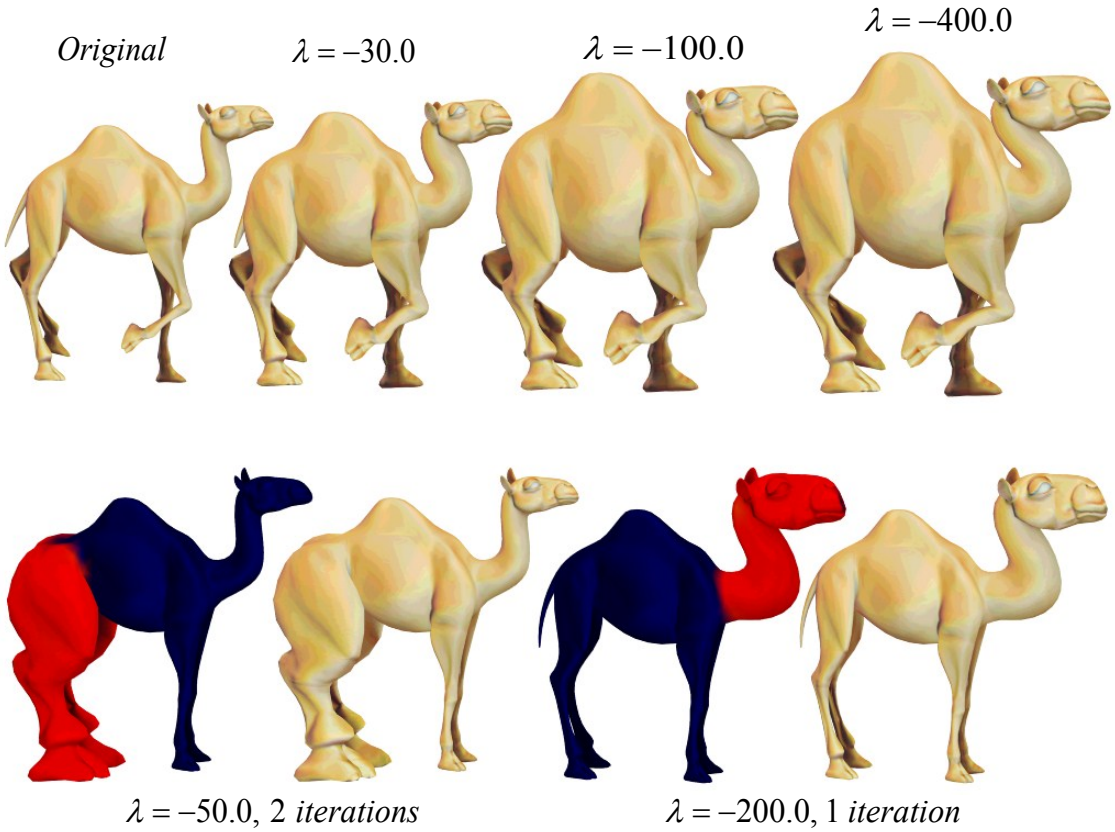

Figure **5-3**: Top row: Original camel model in left. Shape inflation with $\lambda = -30.0$, $\lambda = -100.0$, $\lambda = -400.0$. Bottom row: Shape inflation with weight vertex group, $\lambda = -50.0$ and 2 iterations for the legs, $\lambda = -200.0$ and 1 iteration for the head and neck.

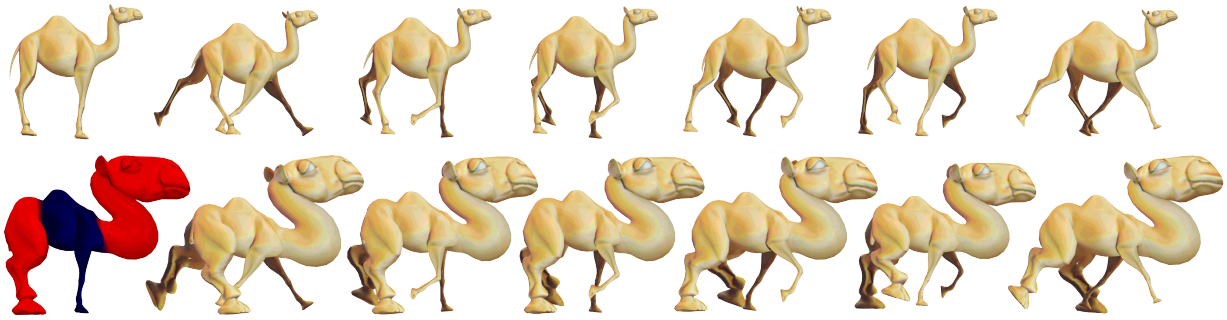

Figure **5-4**: The method is pose insensitive. The inflation for the different poses are similar in terms of shape. Top row: Original walk cycle camel model. Bottom row: Shape inflation with weight vertex group, $\lambda = -400$ and 2 iterations.

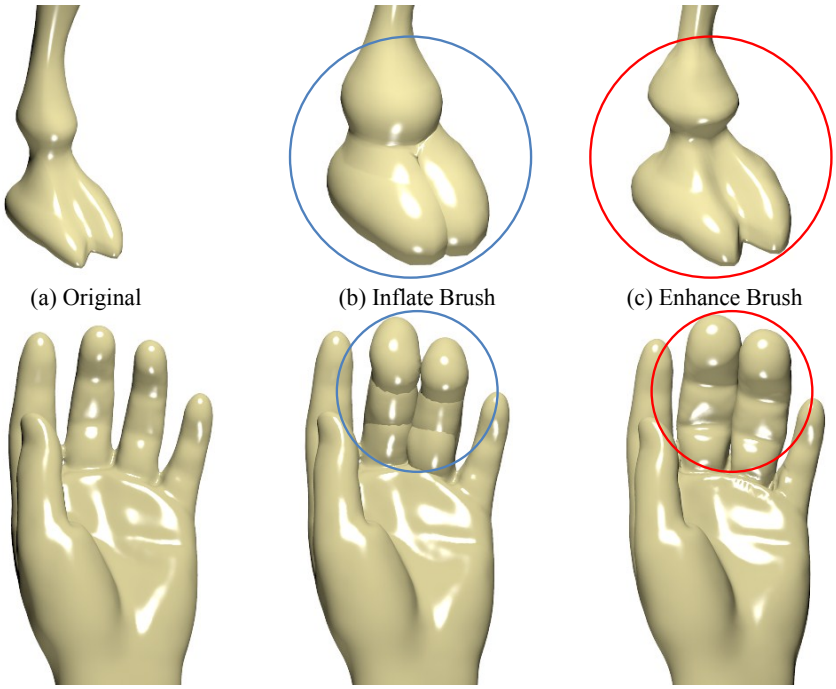

Figure **5-5**: Top row: (a) Leg Camel, (b) Inflate brush for leg into blue circle, (c) Inflate shape brush for leg into red circle. Bottom row: (a) Hand, (b) Inflate brush for fingers into blue circle, (c) Shape inflation brush for fingers in red circle.

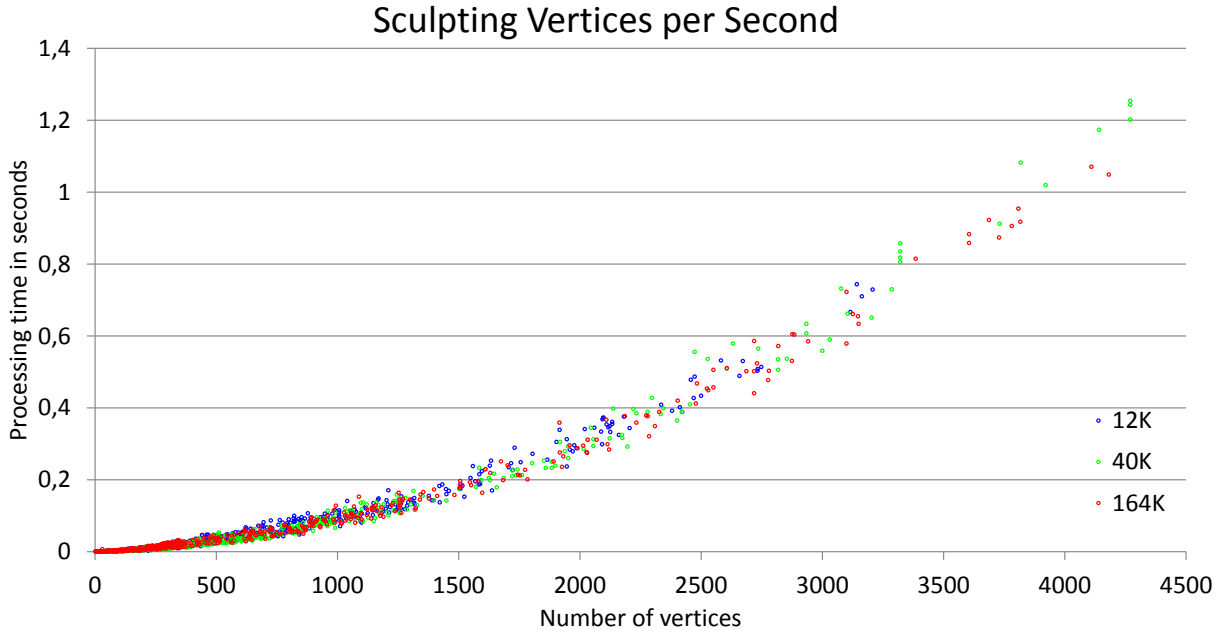

Figure **5-6**: Performance of our dynamic shape inflation brush in terms of the sculpted vertices per second. Three models with 12K, 40K, 164K vertices used for sculpting in real time.

different results due to the quads areas in the model. Quadss with larger area have smaller inflation (figure **5-7**.c skull), and smaller quads have larger inflations (figure **5-7**.c chin).

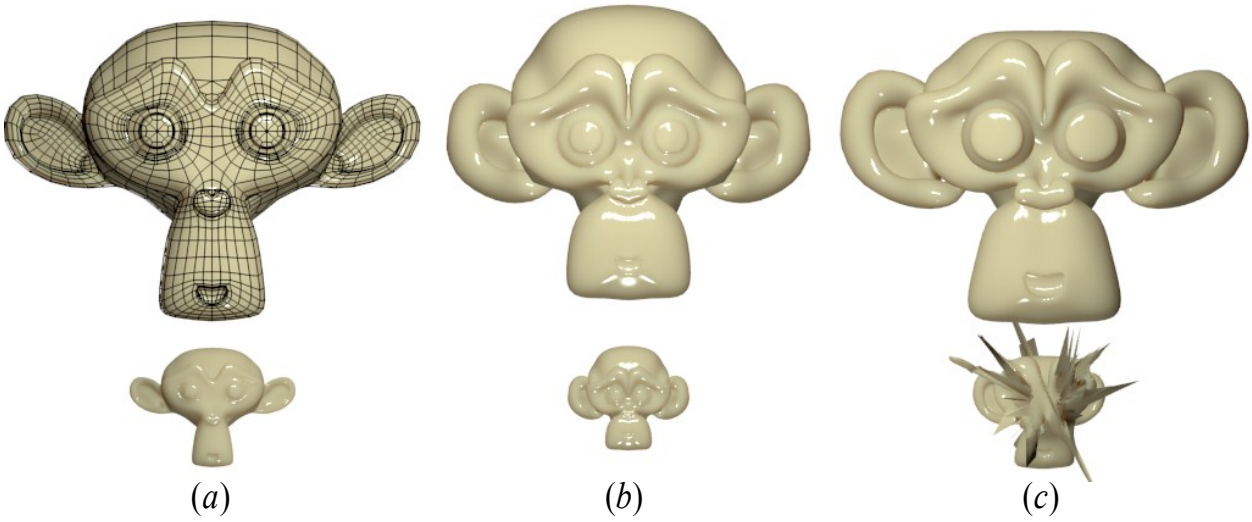

Figure **5-7**: (a) Bottom row: Original Model. Top row: Original model scaled by 4. (b) Top and bottom row: inflating with Normalized-TQLBO $\lambda = -50$ (c) Top and bottom row: inflating with TQLBO $\lambda = -50$.

5.1 Implementation

The method was implemented as a modifier for modeling and brush for sculpting, on the Blender software [4] 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 zero value that could cause spikes and bad results.

Under these conditions, the matrix at equation 4-5 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 4-8 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 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.

Bibliography

[1] Oscar Kin-Chung Au, Chiew-Lan Tai, Hung-Kuo Chu, Daniel Cohen-Or, and Tong-Yee Lee. Skeleton extraction by mesh contraction. *ACM Transactions on Graphics*, 27(3):10, 2008. Skeleton Extraction.

[2] Mikhail Belkin, Jian Sun, and Yusu Wang. Discrete laplace operator on meshed surfaces. In *Proceedings of the twenty-fourth annual symposium on Computational geometry*, SCG '08, pages 278–287, New York, NY, USA, 2008. ACM.

[3] Henning Biermann, Adi Levin, and Denis Zorin. Piecewise smooth subdivision surfaces with normal control. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '00, pages 113–120, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

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

[5] Mario Botsch, Mark Pauly, Christian Rossl, Stephan Bischoff, and Leif Kobbelt. Geometric modeling based on triangle meshes. In *ACM SIGGRAPH 2006 Courses*, SIGGRAPH '06, New York, NY, USA, 2006. ACM.

[6] E. Catmull and J. Clark. Recursively generated b-spline surfaces on arbitrary topological meshes. *Computer-Aided Design*, 10(6):350–355, November 1978.

[7] Yong Chen and Charlie C. L. Wang. Uniform offsetting of polygonal model based on layered depth-normal images. *Comput. Aided Des.*, 43(1):31–46, January 2011.

[8] Sabine Coquillart. Extended free-form deformation: a sculpturing tool for 3d geometric modeling. *SIGGRAPH Comput. Graph.*, 24(4):187–196, September 1990.

[9] Nicu D. Cornea and Patrick Min. Curve-skeleton properties, applications, and algorithms. *IEEE Transactions on Visualization and Computer Graphics*, 13(3):530–548, 2007. Skeleton Extraction Survey Member-Silver,, Deborah.

[10] Tony DeRose, Michael Kass, and Tien Truong. Subdivision surfaces in character animation. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '98, pages 85–94, New York, NY, USA, 1998. ACM.

[11] Mathieu Desbrun, Mark Meyer, Peter Schröder, and Alan H. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. In *SIGGRAPH '99: Proceedings of*

the 26th annual conference on Computer graphics and interactive techniques, pages 317–324, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co. Skeleton Extraction.

[12] Ran Gal, Olga Sorkine, Niloy J. Mitra, and Daniel Cohen-Or. iwires: An analyze-and-edit approach to shape manipulation. *ACM Transactions on Graphics (Siggraph)*, 28(3):#33, 1–10, 2009.

[13] Tinsley A. Galyean and John F. Hughes. Sculpting: an interactive volumetric modeling technique. *SIGGRAPH Comput. Graph.*, 25(4):267–274, July 1991.

[14] Ozgur Gonen and Ergun Akleman. Smi 2012: Short paper: Sketch based 3d modeling with curvature classification. *Comput. Graph.*, 36(5):521–525, August 2012.

[15] Uwe Hahne. *Weighting in Laplacian Mesh Editing*. PhD thesis, Bauhaus-Universitat Weimar, 2006.

[16] Takeo Igarashi, Satoshi Matsuoka, and Hidehiko Tanaka. Teddy: a sketching interface for 3d freeform design. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '99, pages 409–416, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[17] Liu, L., Bajaj, C., Deasy, J. O., Low, D. A., Ju, and T. Surface reconstruction from non-parallel curve networks. *Computer Graphics Forum*, 27(2):155–163, April 2008. Modelling 3D.

[18] C. Loop. Smooth subdivision surfaces based on triangles. Department of mathematics, University of Utah, Utah, USA, August 1987.

[19] Mark Meyer, Mathieu Desbrun, Peter Schröder, and Alan H. Barr. Discrete differential-geometry operators for triangulated 2-manifolds. In Hans-Christian Hege and Konrad Polthier, editors, *Visualization and Mathematics III*, pages 35–57. Springer-Verlag, Heidelberg, 2003.

[20] Tony Mullen. *Introducing character animation with Blender*. Indianapolis, Ind. Wiley Pub. cop., 2007.

[21] Yutaka Ohtake, Alexander Belyaev, and Hans-Peter Seidel. Ridge-valley lines on meshes via implicit surface fitting. *ACM Trans. Graph.*, 23(3):609–612, August 2004.

[22] Ulrich Pinkall, Strasse Des Juni, and Konrad Polthier. Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics*, 2:15–36, 1993.

[23] Pawas Ranjan. *Discrete Laplace Operator: Theory and Applications*. PhD thesis, The Ohio State University, Columbus, OH, USA, 2012. AAI3530174.

[24] Steven Rosenberg. *The Laplacian on a Riemannian manifold: an introduction to analysis on manifolds*. Number 31. Cambridge University Press, 1997.

[25] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel. Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, SGP '04, pages 175–184, New York, NY, USA, 2004. ACM.

[26] Olga Sorkine. Differential representations for mesh processing. *Comput. Graph. Forum*, 1:789–807, 2006.

[27] Michael Spivak. *A Comprehensive Introduction to Differential Geometry*, volume 1. Publish or Perish, Inc, 3rd edition, January 1999. Other.

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

[29] Lucian Stanculescu, Raphalle Chaine, and Marie-Paule Cani. Freestyle: Sculpting meshes with self-adaptive topology. *Computers & Graphics*, 35(3):614 – 622, 2011. Shape Modeling International (SMI) Conference 2011.

[30] Yunhui Xiong, Guiqing Li, and Guoqiang Han. Mean laplace-beltrami operator for quadrilateral meshes. In Zhigeng Pan, Adrian Cheok, Wolfgang Muller, and Xubo Yang, editors, *Transactions on Edutainment V*, volume 6530 of *Lecture Notes in Computer Science*, pages 189–201. Springer Berlin / Heidelberg, 2011.

[31] Kun Zhou, Jin Huang, John Snyder, Xinguo Liu, Hujun Bao, Baining Guo, and Heung-Yeung Shum. Large mesh deformation using the volumetric graph laplacian. *ACM Trans. Graph.*, 24(3):496–503, July 2005.

[32] Wei Zhuo and Jarek Rossignac. Curvature-based offset distance: Implementations and applications. *Computers & Graphics*, 36(5):445 – 454, 2012.