

An Adapted Laplacian Operator For Hybrid Quad/Triangle Meshes

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

Master in Systems Engineering and Computer Science

Alexander Pinzón Fernández

Advisor: Eduardo Romero

Bogotá-Colombia. October 2015



UNIVERSIDAD
NACIONAL



OUTLINE

- 1 Introduction
- 2 Proposed Method
- 3 Evaluation and Results
- 4 Products
- 5 Conclusions

OUTLINE

- 1 Introduction
- 2 Proposed Method
- 3 Evaluation and Results
- 4 Products
- 5 Conclusions

SPECTRAL MESH PROCESSING

Taubin in 1995 suggested that the Discrete Laplace Operator allows to do *Spectral Processing* on **Polygonal Meshes** in an analogous way to signal processing with the Fourier transform.

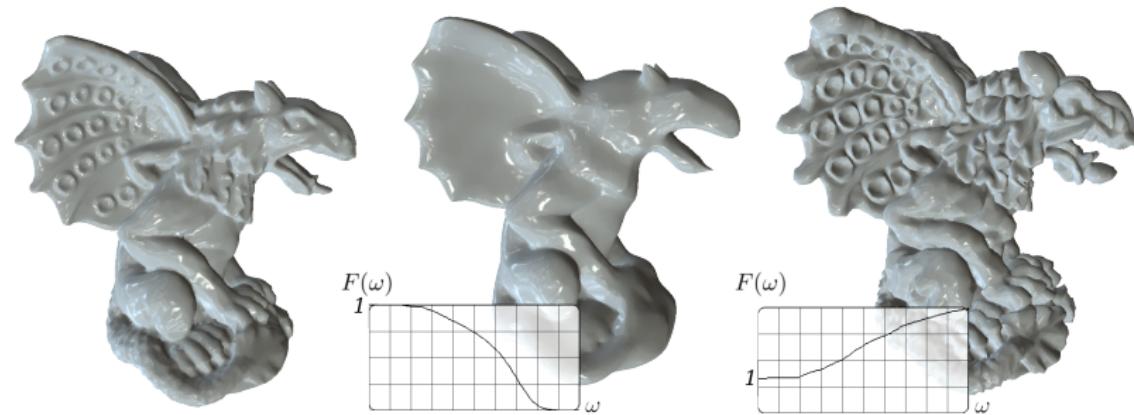


Figure: Original Mesh, Low-pass and enhancement filters [Vallet08].

FOURIER TRANSFORM

The basis functions of the Fourier transform

$$\mathbf{e}_w = e^{2\pi i w x} = \cos(2\pi w x) - i \sin(2\pi w x)$$

The Fourier transform of $f(x)$

$$F(w) = \int_{-\infty}^{\infty} f(x) \mathbf{e}_w dx$$

The Inverse Fourier transform of $F(w)$

$$f(x) = \int_{-\infty}^{\infty} F(w) \mathbf{e}_w dw = \sum_{w=-\infty}^{\infty} \langle f, \mathbf{e}_w \rangle \mathbf{e}_w$$

LAPLACIAN OF THE FOURIER TRANSFORM

if \mathbf{u} is a eigenfunction and λ is a corresponding eigenvalue of a linear differential operator \mathbf{A} then

$$\mathbf{A}\mathbf{u} = \lambda\mathbf{u}, \quad \mathbf{u} \neq 0$$

The basis \mathbf{e}_w sine and cosine functions of the Fourier transform are eigenfunctions of the Laplacian with eigenvalue λ_w .

$$\Delta(\mathbf{e}_w) = \frac{\partial^2}{\partial x^2} \mathbf{e}_w = -(2\pi w)^2 \mathbf{e}_w = \lambda_w \mathbf{e}_w$$

$$\Delta(\mathbf{e}_w) = \lambda_w \mathbf{e}_w$$

LAPLACE-BELTRAMI OPERATOR

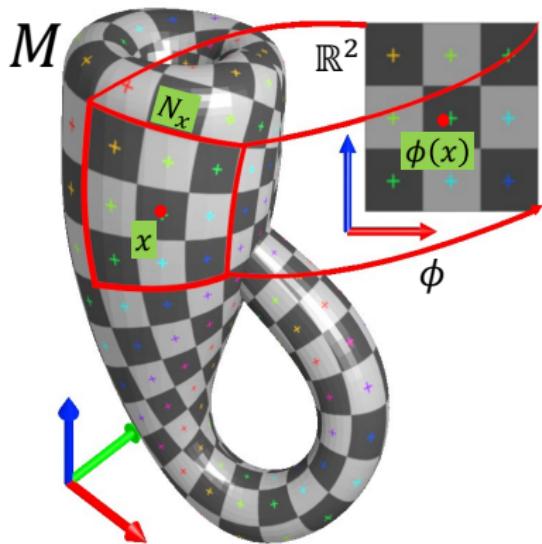
The Laplace-Beltrami operator Δ_M is a differential operator given by the divergence of a gradient field on a **surface**.

$$\begin{aligned}\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial z^2} &= 0 \\ \nabla_M^2 f &= \Delta_M = 0 \\ \Delta_M &= \operatorname{div}(\nabla_M f)\end{aligned}$$

where M is compact and smooth **Surface** (2D-Manifold)

Laplace-Beltrami is a generalization of Laplace operator for functions on surfaces.

SURFACE



A Surface M is a 2D topological manifold

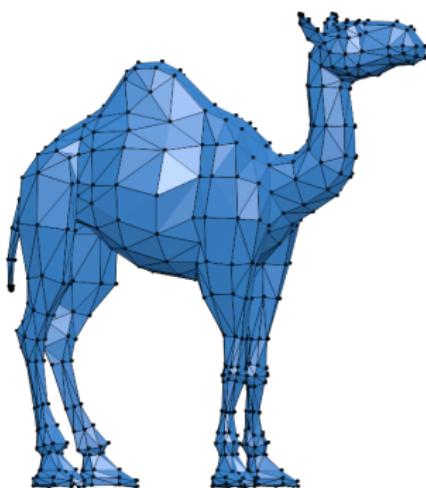
For which every point $x \in M \subseteq \mathbb{R}^3$ has a neighbourhood N_x homeomorphic to euclidean space \mathbb{R}^2 .

A homeomorphism
 $\phi : N_x \rightarrow \mathbb{R}^2$

SURFACE DISCRETIZATION WITH POLYGONAL MESHES

In computer graphics a **Surface** is often discretized into a **Polygonal Mesh**

Polygonal Mesh is a set of points that are connected by **triangles** and **quads**.



We need discrete versions of Laplacian operator to work with polygonal meshes.

DISCRETE LAPLACIAN OPERATOR AS A MATRIX EQUATION

THE DISCRETE VERSION OF LAPLACE OPERATOR FOR POLYGONAL MESHES.

Given a polygonal mesh $\mathbf{M} = [v_1, \dots, v_n]^T$ and $v_i \in \mathbb{R}^3$ we define the **Discrete Laplacian operator matrix** L with size $n \times n$

$$L(i,j) = \begin{cases} w_{ij} & \text{if } j \in N(v_i) \\ \sum w_{ik} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

w_{ij} are the weights between the vertex v_i and vertex v_j . The weights are defined depending of the polygonal structure and application.

$N(v_i)$ is the 1-ring neighborhood with shared face to vertex v_i .

DESIRED PROPERTIES FOR LAPLACIAN MATRIX

This properties ensure the construction of eigenstructure of Laplacian matrix and then those eigenvectors of L matrix are a orthogonal basis of \mathbb{R}^n

- Symmetry.
- Square.
- Locality.
- Positive Weights.
- Positive Semi-Definiteness.
- Convergence.

EIGENSTRUCTURE OF LAPLACIAN MATRIX

$$L\mathbf{e}_i = \lambda_i \mathbf{e}_i, \quad \mathbf{e}_i \neq 0$$

- The eigenvector \mathbf{e}_i of L is a *natural vibration* of the mesh [Taubin95].
- The frequency of the wave \mathbf{e}_i is the eigenvalue λ_i of L that is the *natural frequency* [Taubin95].

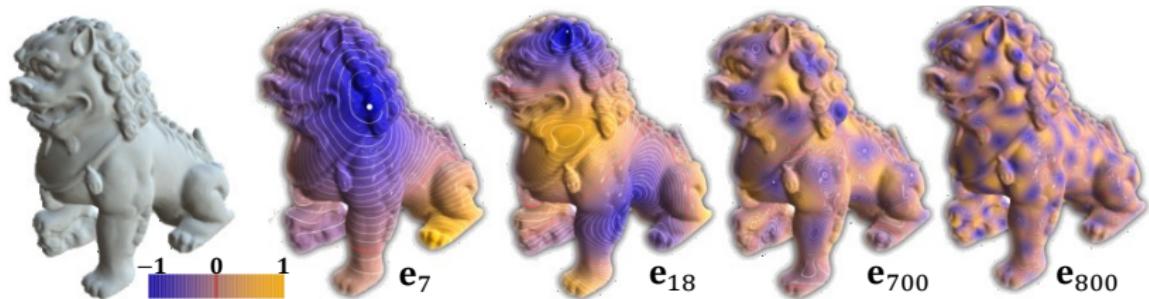


Figure: Color values are the amplitude of a wave \mathbf{e}_i projected on the mesh [Vallet08].

MESH RECONSTRUCTION

The mesh \mathbf{M} can be reconstructed from your spectral decomposition in the analogous way that Fourier transform.

$$\mathbf{M} = \sum_{i=1}^n \langle \mathbf{e}_i, \mathbf{M} \rangle \mathbf{e}_i$$

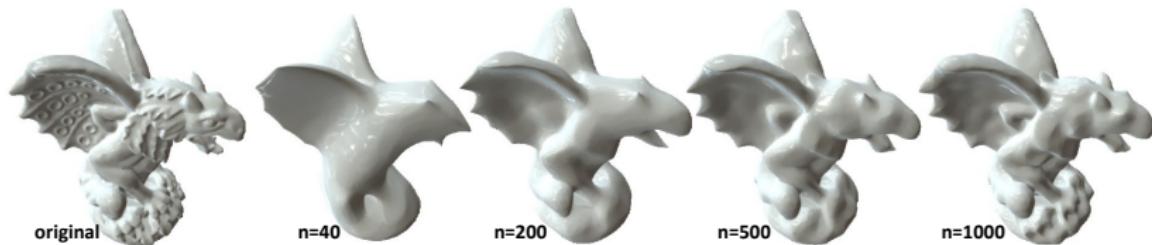


Figure: Mesh reconstruction with first n eigenvectors of the Discrete Laplacian Operator [Levy10].

WEIGHTS FOR LAPLACIAN OPERATOR

Desired property:

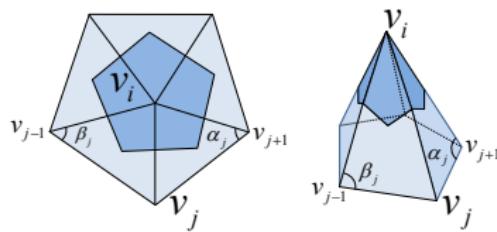
Non-negative weights ω_{ij} for $i \neq j$ ensure a positive semi-definite matrix.

- Umbrella Operator $w_{ij} = \begin{cases} 1 & \text{if } j \in N(v_i) \\ 0 & \text{else} \end{cases}$
- Fujiwara's Operator $w_{ij} = \begin{cases} \frac{1}{\|v_j - v_i\|} & \text{if } j \in N(v_i) \\ 0 & \text{else} \end{cases}$

1999 DESBRUN'S OPERATOR FOR TRIANGLE MESHES

THIS OPERATOR ONLY WORK WITH MESHES COMPOSED ONLY BY **TRIANGLES**.

Desbrun's operator is the discretized version of the Laplace-Beltrami operator.



$$w_{ij} = \frac{1}{4A_i} (\cot \alpha_j + \cot \beta_j)$$

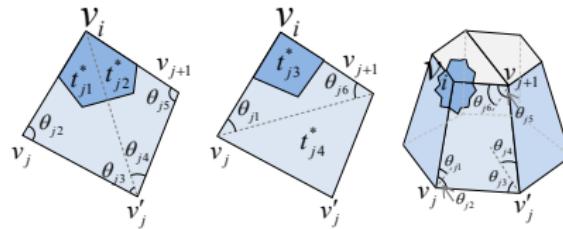
where A_i is area of the 1-ring neighborhood.

α and β are the opposite angles to edge between vertex v_i and vertex v_j .

2011 XION'S MLBO OPERATOR FOR QUAD MESHES

THIS OPERATOR ONLY WORK WITH MESHES COMPOSED ONLY BY **QUADS**.

Xion's operator is the mean of Desbrun's operator for all possible triangulations for quad meshes.



$$w_{ij} = \frac{1}{4A_i} (\cot \theta_{(j-1)1} + \cot \theta_{(j-1)4} + \cot \theta_{j3} + \cot \theta_{j6})$$

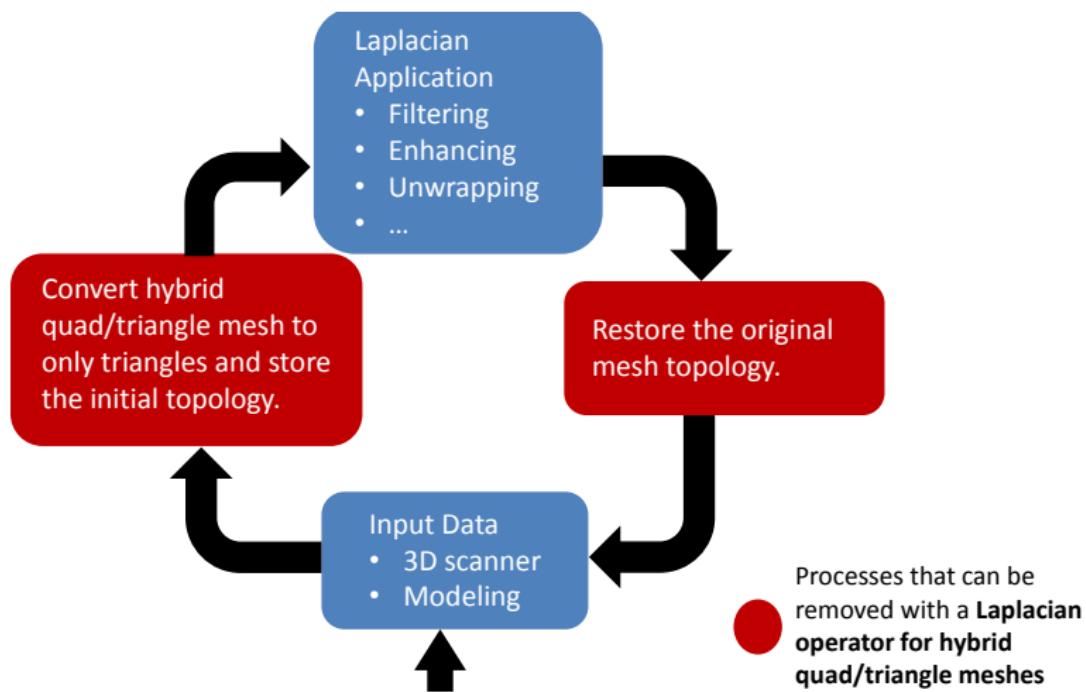
$$w_{ij'} = \frac{1}{2A_i} (\cot \theta_{j2} + \cot \theta_{j5})$$

$$A_i = \frac{1}{2} \sum_{j \in i^*} (A(t_{j2}) + A(t_{j3}) + A(t_{j4}))$$

where w_{ij} are the weights of neighbors that share an edge with v_i and $w_{ij'}$ are the weights of neighbors that share face with v_i .

LAPLACIAN APPLICATIONS ON 3D MODELING PROCESS

GENERAL 3D MODELING PROCESS ON POLYGONAL MESHES.



EDGE LOOPS FOR FACES

Hybrid quad/triangle meshes are necessary by artists to 3D modeling

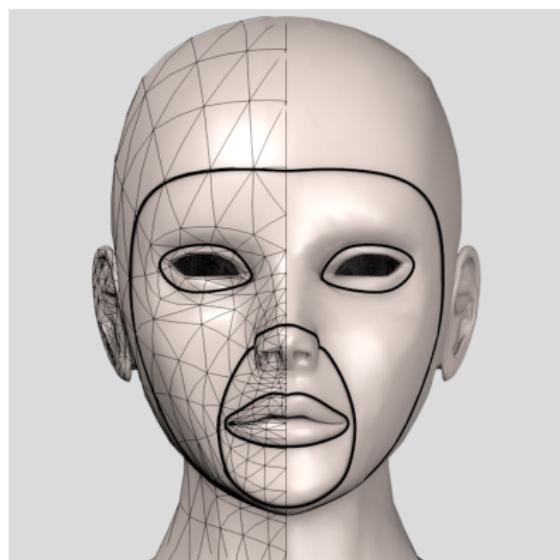


Figure: Triangle Mesh

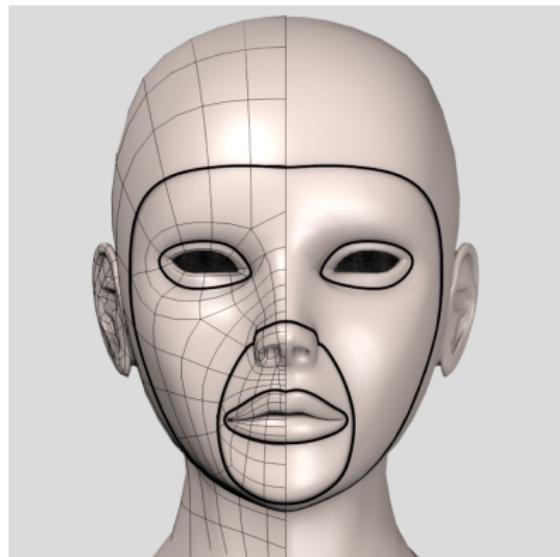


Figure: Hybrid Quad/Triangle Mesh

OUTLINE

1 Introduction

2 Proposed Method

3 Evaluation and Results

4 Products

5 Conclusions

OUR PROPOSAL

We propose an extension of the discrete Laplace-Beltrami operator to work with hybrid quad/triangle meshes and eliminates the need of triangulate the mesh and also allows the preservation of the original topology.

GRADIENT OF VORONOI AREA

The area change produced by the movement of v_i is called the gradient of *Voronoi region* [Pinkall93, Desbrun99]

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

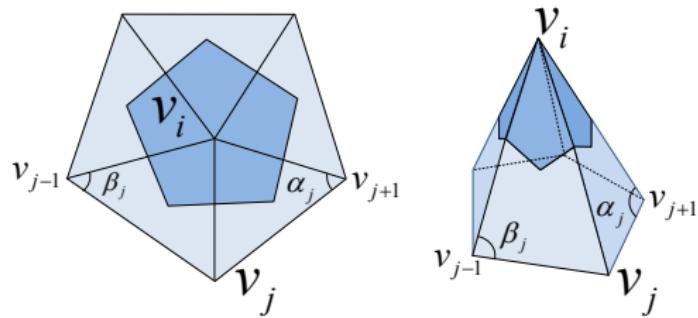


Figure: Area of the Voronoi region around v_i in dark blue. v_i belong to the first neighborhood around v_i . α_j and β_j opposite angles to edge $\overrightarrow{v_j - v_i}$.

MEAN CURVATURE OF SURFACES

In 2D, curvature κ at a given point p on a circle with radius R is defined as [Meyer01]

$$\kappa = \frac{1}{R}$$

In 3D-surface, *sectional curvature* at a given point p is the intersection of a surface with a plane parallel to a normal \mathbf{n} of p [Meyer01].

The *Mean Curvature* κ_H at a given point p is defined as

$$\kappa_H = \frac{\kappa_1 + \kappa_2}{2}$$

where κ_1 and κ_2 are the maximal and minimal sectional curvatures known as *principal curvatures* [Meyer01].

DISCRETE MEAN CURVATURE NORMAL

Gradient of voronoi region

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

If the gradient of Voronoi region is normalized by the total area of the 1-ring neighborhood around v_i , we obtained a *discrete mean curvature normal*.

$$2\kappa_H \mathbf{n} = \frac{\nabla \mathbf{A}}{\mathbf{A}}$$

The *Laplace Beltrami operator* Δ is used for measuring the mean curvature normal of the surface S [Pinkall93].

$$\Delta S = 2\kappa_H \mathbf{n}$$

MEAN AVERAGE AREA

Xiong's define the mean average area of voronoi region of quad around v_i as the average of areas of primal and dual triangulations.

$$\text{Area}(Q) = \frac{\text{Area}_1 + \text{Area}_2}{2} = \frac{A(t_{j1}^*) + A(t_{j2}^*) + A(t_{j3}^*)}{2}$$

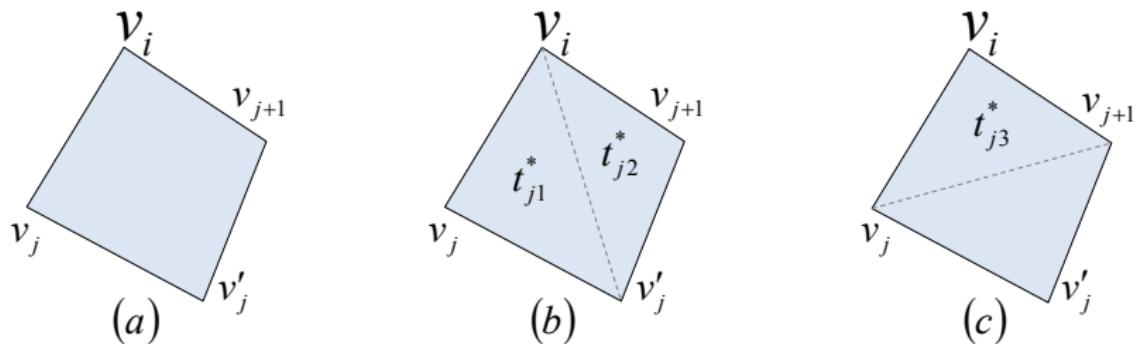


Figure: (a) Original quad. (b) Primal triangulation around v_i is $t_{j1}^* \equiv \Delta v_i v_j v'_j$, $t_{j2}^* \equiv \Delta v_i v'_j v_{j+1}$ (b) Dual triangulation around v_i is $t_{j3}^* \equiv \Delta v_i v_j v_{j+1}$.

LAPLACE BELTRAMI OPERATOR FOR HYBRID QUAD/TRIANGLE MESHES

Given a hybrid mesh $M = \{V, Q, T\}$, with vertices V , quads Q and triangles T , define the mean average area of all triangulations around v_i as

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

where $q_j \in Q_{v_i}$ and $t_k \in T_{v_i}$

Applying Xiong's mean average area to (1)

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

LAPLACE BELTRAMI OPERATOR FOR HYBRID QUAD/TRIANGLE MESHES

Applying the gradient operator to (2)

$$\nabla A(v_i) = \frac{1}{2} \sum_{j=1}^m \left[\nabla A(t_{j1}^*) + \nabla A(t_{j2}^*) + \nabla A(t_{j3}^*) \right] + \sum_{k=1}^r \nabla A(t_k) \quad (3)$$

Using Desbrun's equation to compute the gradient to (3), we have

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

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

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

$$\nabla A(t_k) = \frac{\cot \alpha_k(v_k - v_i) + \cot \beta_k(v_k - v_i)}{2}$$

LAPLACE BELTRAMI OPERATOR FOR HYBRID QUAD/TRIANGLE MESHES

Therefore (3) can be rewritten as

$$\nabla A(v_i) = \sum_{j=1}^n w_{ij} (v_j - v_i) \quad (4)$$

where v_j are the neighbors of v_i

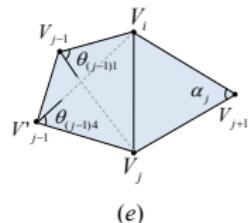
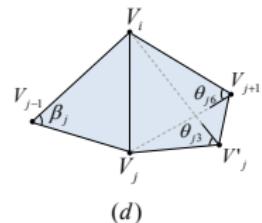
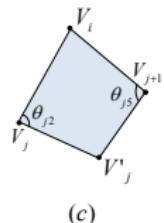
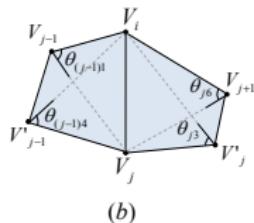
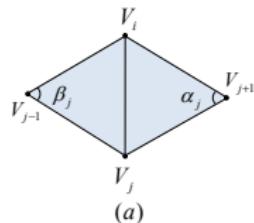
Using the relationship between Laplace-Beltrami operator Δ and Mean curvature normal in (4) we define the **Triangle Quad Laplace Beltrami Operator TQLBO** as

$$\Delta(v_i) = 2\kappa_H \mathbf{n}_i = \frac{\nabla A(v_i)}{Area_i} = \frac{1}{2Area_i} \sum_{j=1}^n w_{ij} (v_j - v_i) \quad (5)$$

where $Area_i$ is the area of the 1-ring neighborhood around v_i

WEIGHTS FOR TQLBO

We define the weights of the TQLBO based on five simple cases



The 5 basic triangle-quad cases with a vertex V_i and the relationship with V_j and V'_j .

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

LAPLACE OPERATOR AS A MATRIX EQUATION

We define a TQLBO as a matrix equation

$$L(i,j) = \begin{cases} -\frac{1}{2A_i} w_{ij} & \text{if } j \in N(v_i) \\ \frac{1}{2A_i} \sum_{k \in N(v_i)} w_{ik} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

Where L is a $n \times n$ matrix, n is the number of vertices, $N(v_i)$ is the 1-ring neighborhood with shared face to v_i , A_i is the ring area around v_i .

OUTLINE

1 Introduction

2 Proposed Method

3 Evaluation and Results

4 Products

5 Conclusions

ENHANCING

A standard diffusion process is used.

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

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

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

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

where L is the TQLBO, V' are the smoothing vertices, V^t are the actual vertices positions, W_p is a diagonal matrix with vertex weights, and λdt is the inflate factor.

SCULPTING WITH ENHANCING FILTER

INFLATE BRUSH

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.

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

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.

COMPUTE THE MINIMAL SURFACE WITH TQLBO RESULTS

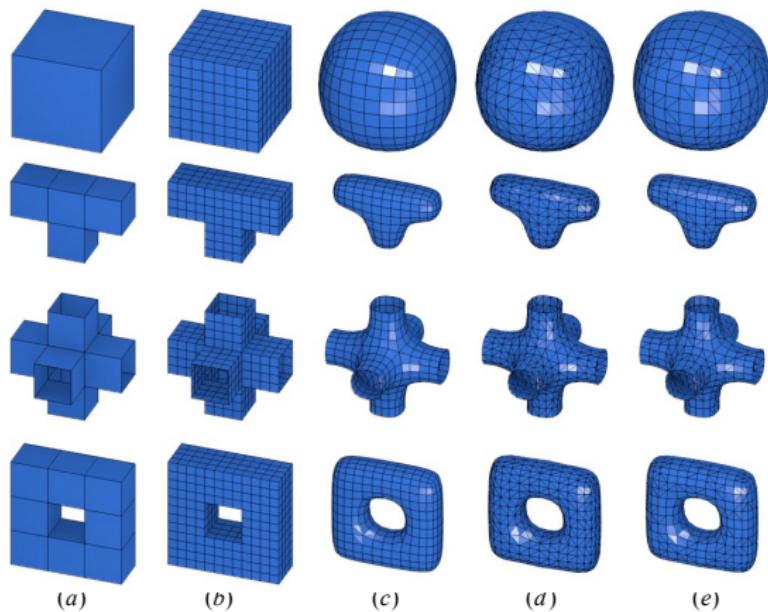


Figure: (a) Original Model. (b) Simple subdivision. (c), (d) (e) Laplacian smoothing with $\lambda = 7$ and 2 iterations: (c) for quads, (d) for triangles, (e) for triangles and quads random chosen.

SHAPE INFLATION WITH ENHANCING FILTER RESULTS

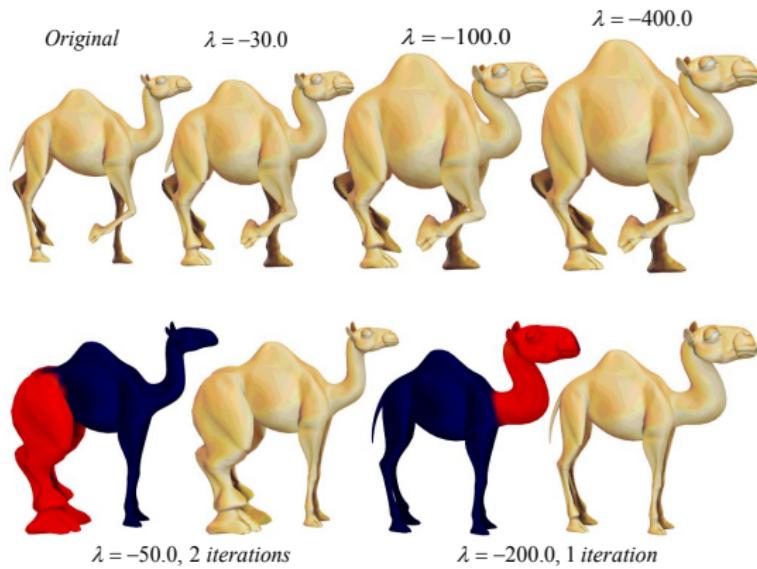


Figure: Top row: Original camel model in left. Shape inflation with $\lambda = -30.0$, $\lambda = -100.0$, $\lambda = -400.0$. Bottom row: Shape inflation with weight vertex group, $\lambda = -50.0$ and 2 iterations for the legs, $\lambda = -200.0$ and 1 iteration for the head and neck.

INFLATE BRUSH RESULTS

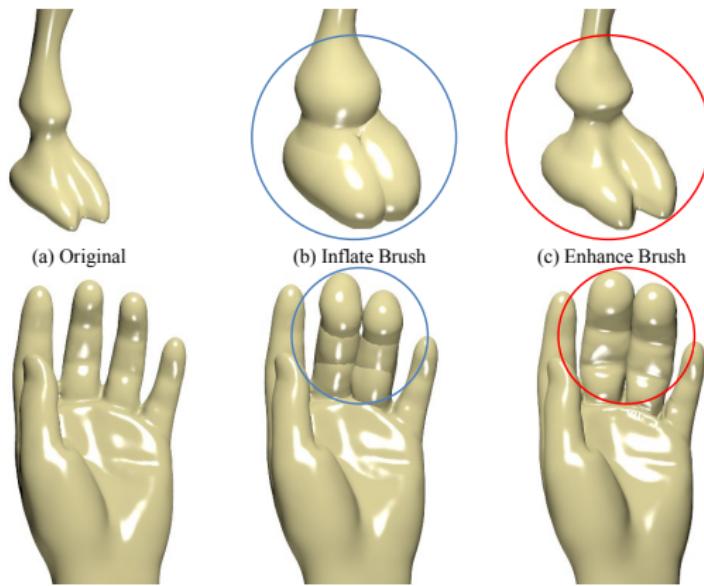


Figure: Top row: (a) Leg Camel, (b) Traditional inflate brush for leg into blue circle, (c) Shape inflation brush for leg into red circle.
Bottom row: (a) Hand, (b) Traditional inflate brush for fingers into blue circle, (c) Shape inflation brush for fingers in red circle.

INFLATE BRUSH PERFORMANCE

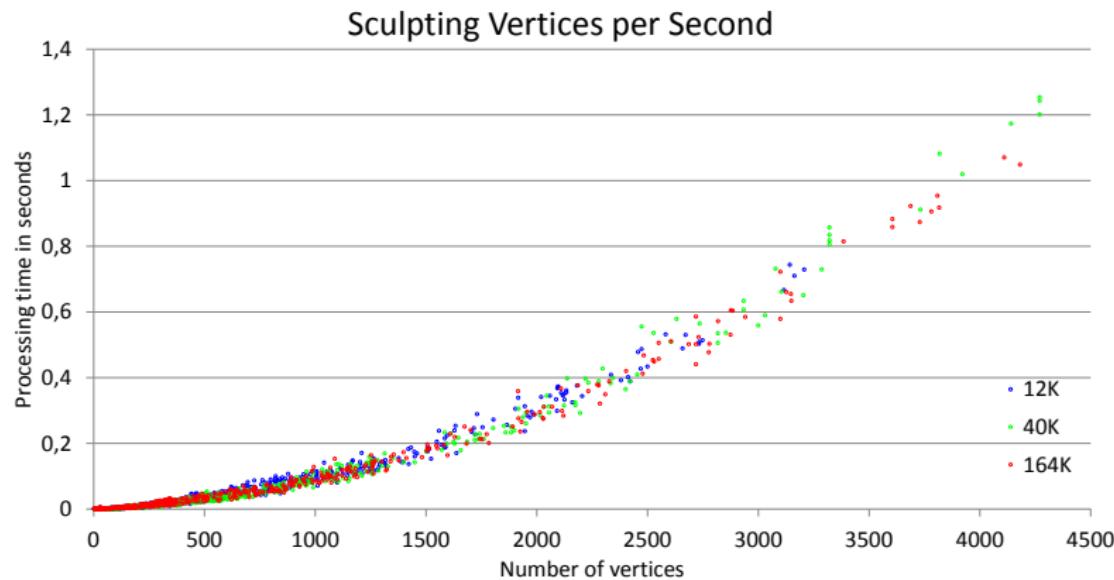


Figure: Performance of our dynamic shape inflation brush in terms of the sculpted vertices per second. Three models with 12K, 40K, 164K vertices used for sculpting in real time.

OUTLINE

1 Introduction

2 Proposed Method

3 Evaluation and Results

4 Products

5 Conclusions

1ST PRODUCT - CONFERENCE PAPER

BRAZILIAN SYMPOSIUM ON COMPUTER GRAPHICS AND IMAGE PROCESSING SIBGRAPI 2013

Shape Inflation With an Adapted Laplacian Operator For Hybrid Quad/Triangle Meshes

Alexander Pinzon, Eduardo Romero

Cimalab Research Group

Universidad Nacional de Colombia

Bogota-Colombia

Email: apinzon@unal.edu.co, edromero@unal.edu.co

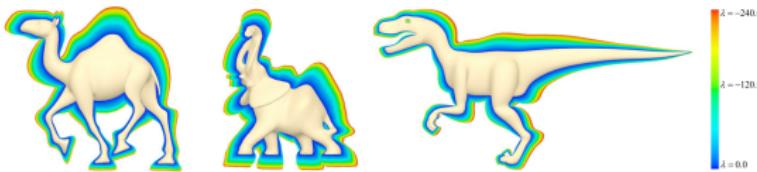


Fig. 1. A set of 48 successive shapes enhanced, from $\lambda = 0.0$ in blue to $\lambda = -240.0$ in red, with steps of -5.0 .

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 inflated 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.

Keywords—laplacian smooth; curvature; sculpting; subdivision surface

[12]. Nevertheless 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 [13]. Nowadays, many artists use a manual connection of a couple of vertices to perform animation processes and interpolation [14]. It is then of paramount

2ND PRODUCT - AWARDED INTERNSHIP

MESH SMOOTHING BASED ON CURVATURE FLOW OPERATOR IN A DIFFUSION EQUATION

Sponsor: Google Inc - Google Summer of Code 2012 program

Project: Mesh smoothing based on curvature flow operator in a diffusion equation

Synopsis: This project proposes a new and robust mesh smoothing tool that remove the noise of the surfaces of models captured with 3d scanners, zcameras among others.

Blender software is an open source 3D application for modeling, rendering, composing, video editing and game creation.

LAPLACIAN SMOOTH TOOL FOR BLENDER

MESH SMOOTHING BASED ON CURVATURE FLOW OPERATOR IN A DIFFUSION EQUATION

We define a Laplacian matrix for mesh smoothing with support for hybrid quad/triangle meshes with holes as

$$L(i,j) = \begin{cases} -\frac{1}{2A_i}w_{ij} & \text{if } j \in N(v_i) \wedge v_i \notin \text{Boundary} \\ \frac{1}{2A_i} \sum_{j \in N(v_i)} w_{ij} & \text{if } i = j \wedge v_i \notin \text{Boundary} \\ -\frac{1}{\|v_i - v_j\|} & \text{if } j \in N(v_i) \wedge \{v_i, v_j\} \in \text{Boundary} \\ \frac{2}{E_i} \sum_{j \in N(v_i)} \frac{1}{\|v_i - v_j\|} & \text{if } i = j \wedge \{v_i, v_j\} \in \text{Boundary} \\ 0 & \text{otherwise} \end{cases}$$

w_{ij} is the TQLBO defined in equation (6)

$$E_i = \sum_{j \in N(v_i)} e_{ij}.$$

USER INTERFACE

MESH SMOOTHING BASED ON CURVATURE FLOW OPERATOR IN A DIFFUSION EQUATION

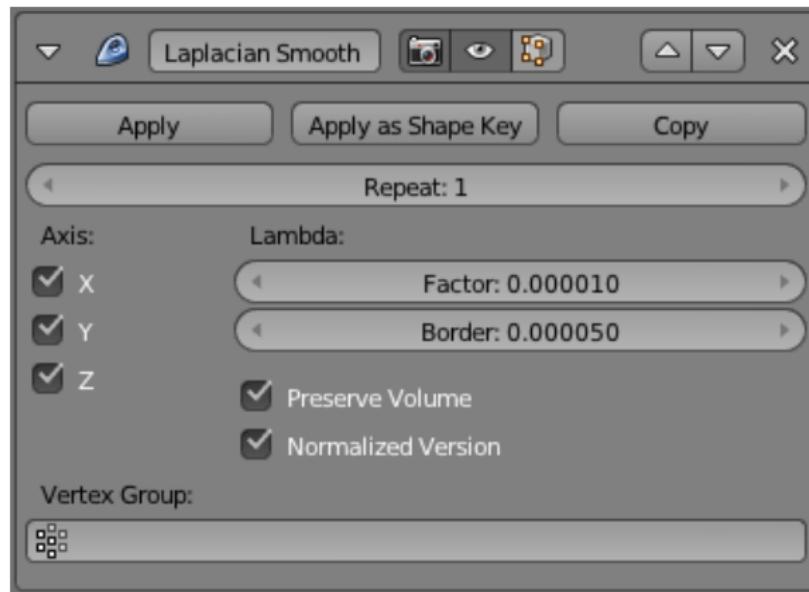


Figure: Panel inside blender user interface of the Laplacian Smooth modifier tool.

RESULTS

MESH SMOOTHING BASED ON CURVATURE FLOW OPERATOR IN A DIFFUSION EQUATION

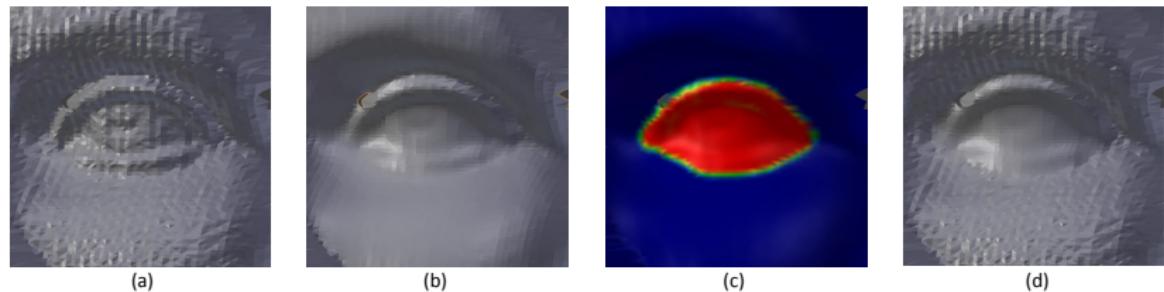


Figure: Use of weights per vertex to constrain the effect of mesh smoothing. (a) Original Model. (b) Smoothing with $\lambda = 1.5$ (c) red vertices *weight* = 1.0, blue vertices *weight* = 0.0. (d) Smoothing with $\lambda = 2.5$. The red vertices were the only vertices smoothed.

RESULTS

MESH SMOOTHING BASED ON CURVATURE FLOW OPERATOR IN A DIFFUSION EQUATION

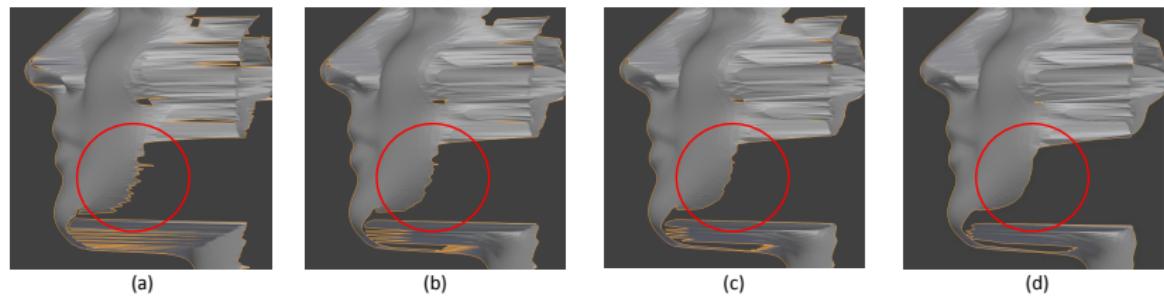


Figure: Smoothing boundary changing λ_{Border} factor. (a) Original Model. (b) Smoothing $\lambda_{Border} = 1.0$. (c) Smoothing $\lambda_{Border} = 2.5$ (d) Smoothing with $\lambda_{Border} = 10.0$.

3RD PRODUCT - AWARDED INTERNSHIP

MESH EDITING WITH LAPLACIAN DEFORM

Sponsor: Google Inc - Google Summer of Code 2013 program

Project: Mesh Editing with Laplacian Deform

Synopsis: This project proposes a new tool that allows to pose a mesh while preserving geometric details of the surface.

Blender software is an open source 3D application for modeling, rendering, composing, video editing and game creation.

DIFFERENTIAL COORDINATES

MESH EDITING WITH LAPLACIAN DEFORM

$$\delta_i = \sum_{j=1}^m w_{ij} (v_i - v_j) \quad (8)$$

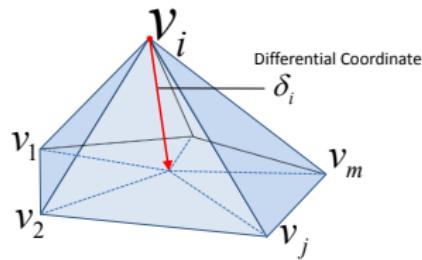


Figure: Difference between v_i and the center of mass of its neighbors v_1, \dots, v_m .

w_{ij} is the TQLBO defined in equation (6)

LAPLACIAN DEFORM

MESH EDITING WITH LAPLACIAN DEFORM

The linear system for finding the new pose of a mesh is.

$$\begin{bmatrix} L \\ W_c \end{bmatrix} V = \begin{bmatrix} \delta \\ W_c C \end{bmatrix} \quad (9)$$

L is a matrix that used our TQLBO defined in equation 7.

W_c is a matrix that has only ones in the indices of anchor vertices.

V is the vertices of mesh.

C is a vector with coordinates of anchor vertices after several manual transformations.

δ are the differential coordinates defined in equation 8.

USER INTERFACE

MESH EDITING WITH LAPLACIAN DEFORM

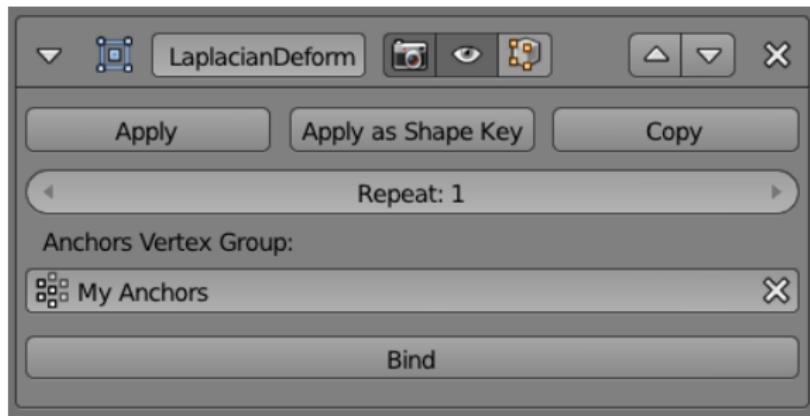


Figure: Panel inside Blender user interface of the Laplacian Deform modifier tool.

RESULTS

MESH EDITING WITH LAPLACIAN DEFORM

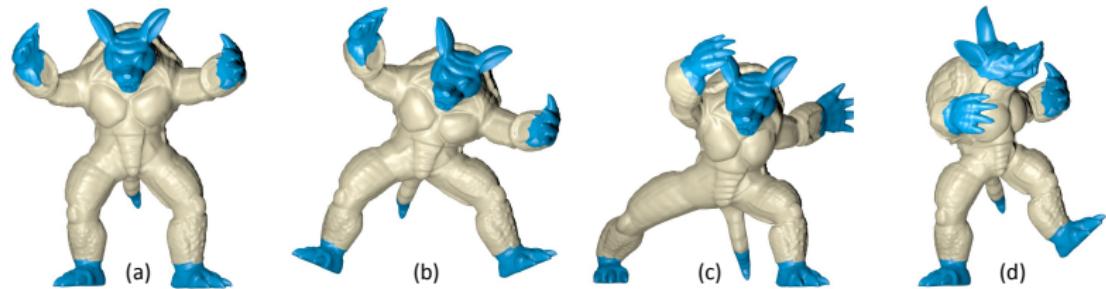


Figure: Anchor vertices in blue. (a) Original Model, (b,c,d) new poses only change the anchor-vertices, the system finds positions for vertices in yellow.

RESULTS

MESH EDITING WITH LAPLACIAN DEFORM

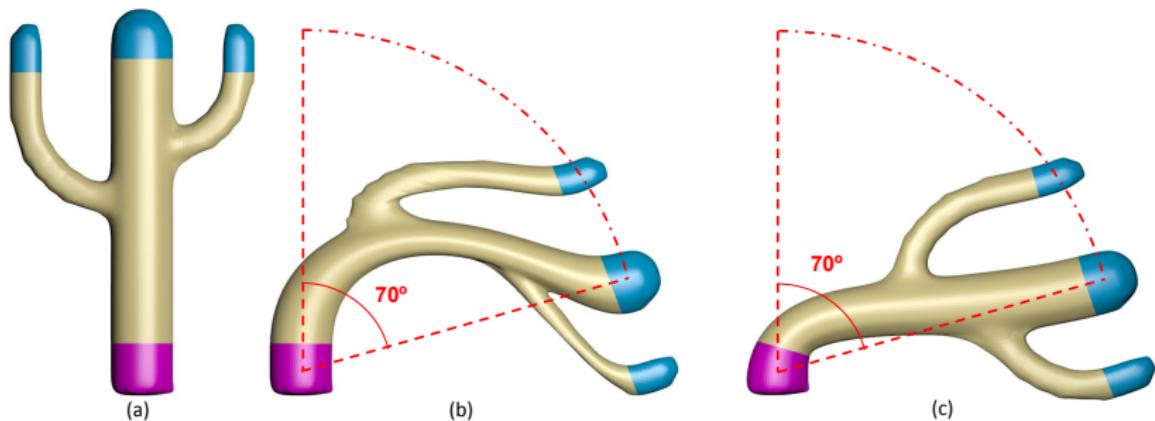


Figure: (a) Original cactus model. (b) Blue segments are rotated 70° to the right and afterwards a basic interpolation is applied to the parts in yellow (c) Blue segments are rotated 70° to the right and afterwards a Laplacian deform tool is applied to the parts in yellow.

RESULTS

MESH EDITING WITH LAPLACIAN DEFORM

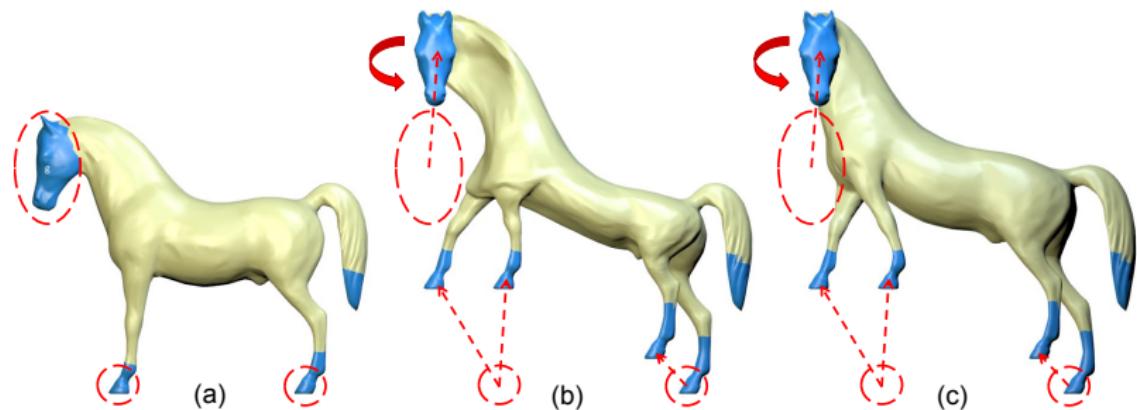


Figure: (a) Original Horse model. (b) The blue segments are translated and rotated and then basic interpolation is applied to the yellow parts (c) The blue segments are translated and rotated and then the Laplacian Deform tool is applied to the yellow parts.

4TH PRODUCT - POSTER

6TH INTERNATIONAL SEMINAR ON MEDICAL IMAGE PROCESSING AND ANALYSIS SIPAIM 2010

Análisis Experimental de la Extracción del Esqueleto por Contracción con Suavizado Laplaciano

Alexander Pineda, Raúl Martínez, Eduardo Herrera

Abstract

Este artículo presenta un análisis experimental del método de extracción del esqueleto por medio de contracción de un volumen con suavizado Laplaciano. El trabajo realiza una implementación experimental al problema de la extracción del esqueleto, para encontrar el rendimiento del método actualizado en comparación con los métodos tradicionales, y discute la forma

de evaluación cuantitativa del esqueleto que evalúa una variedad de criterios de la extracción y se comparan los diferentes en distintos niveles de la jerarquía, y distintas configuraciones del proceso de simplificación. Los resultados muestran un ligero mejoramiento del método frente a los tradicionales, y también presentan una mejor configuración de suavizado.



Contracción con Suavizado Laplaciano

Este método consiste básicamente en una sucesión de pasos por medio del suavizado laplaciano hasta tener un resultado similar a figura 1a. La implementación se realizó en MATLAB®. Se presentan los resultados de la extracción, así como los siguientes términos:

$$\|W_i \nabla V\|^2 + \sum \|W_{ij}^T \nabla V_j - V_i\|^2$$

- *+ Operador Laplaciano para remover las frecuencias altas, es decir sacar las distorsiones y ruido que existen en la imagen.
- *+ Fórmula de extracción que tiene los órbitas, para mantener información clara de los órganos.
- *+ Fórmula de contracción que hace que la forma tridimensional parezca volumétrica.



Referencias y Agradecimientos

•Guan Xin-Cheng, Xu, Chien-Ian Tsai, Yen-Hsun Chu, Samuel Gelman-Gre, and Tang-Nan Wu. 2008. Surface reconstruction of multi-view depth maps. *Comput. Vis. Image Underst.* 112, 11, 2058–2076. Springer International Publishing, and others. Project: Articulated mesh extraction from multi-view observations. *ACM Trans. Graph.* 27(3), 1–16, 2008. DOI

•Hou, J., Faraj, and Patrick M. Min. 2004. Multi-sensor registration, propagation, application, and visualization of 3D medical images. *IEEE Trans. Vis. Comput. Graph.* 10(2), 129–138, 2004. IEEE Computer Society. Survey Member: Oliver Gnilke, Germany.

Agradecimientos: The reported performance data were provided courtesy of the Computer Graphics Group of the UCI Davis Vision Research (Computer, UCI).

Especificación

Se usa la implementación hecha por Guan Xin-Cheng et al. Para los análisis, para evaluar la eficiencia del método, se usaron imágenes de la mano de los voluntarios tomadas por la cámara de video. Se realizó una configuración de los parámetros de acuerdo con el resultado deseado por el usuario. Una vez que se realizó la configuración se registró una secuencia continua de imágenes de la mano. Se realizó una evaluación cuantitativa del resultado obtenido en el resultado final y se clasificaron los criterios al ver figura 2.

De algoritmos existentes se implementaron diferentes pasos para obtener la implementación del esqueleto. Se implementó la contracción de la forma y los componentes del método frente a las transformaciones generadas de la geometría de los puntos.

Resultados

El método responde de forma rápida al input de un volumen tridimensional. La ejecución de la contracción es de 2.2 segundos para una mano. El resultado es de 28.25 segundos para una mano completa. Se realizó una evaluación cuantitativa del resultado obtenido en la figura 2. El resultado es de 1.86 de los resultados para evaluar que tienen falta para reconocer el resultado.



Genera la base de complicación en figura 2. El método es, sobre todo, en el sentido de que es más lento que el otro. El método tiene un momento temporal de 28.25 segundos para una mano completa. La ejecución es de 1.86 de los resultados que tienen falta para reconocer el resultado.

Conclusiones y Trabajo Futuro

El sentido de extracción resulta en resultados, y tienen bajo rendimiento frente a los demás. El resultado responde de forma rápida al input de un volumen tridimensional de la mano. El resultado tiene un momento temporal de 2.2 segundos para una mano. El resultado tiene un momento temporal de 28.25 segundos para una mano completa. La ejecución es de 1.86 de los resultados que tienen falta para reconocer el resultado.

Como trabajo futuro es posible mejorar la precisión informando hacia donde la información espacio temporal se presenta en la filosofía de extracción, para mejorar la precisión. Se puede implementar la extracción de la mano en el resultado de la extracción de la mano. El resultado tiene un momento temporal de 2.2 segundos para una mano completa. La ejecución es de 1.86 de los resultados que tienen falta para reconocer el resultado.

Centro de
Investigación en
Sistemas de
Imágenes y
Visualización



UNIVERSIDAD
NACIONAL
DE COLOMBIA

5TH PRODUCT - POSTER

7TH INTERNATIONAL SEMINAR ON MEDICAL IMAGE PROCESSING AND ANALYSIS SIPAIM 2011

Software para la Extracción del Esqueleto por Contracción y Suavizado

Alexander Pinto, Eduardo Romero

Abstract

Este artículo presenta un software para el procesamiento, visualización, y extracción del esqueleto desde imágenes de polígonos. El software se divide en tres secciones principales: extracción del esqueleto, visualización y aplicación. La sección de extracción contiene un filtro para la extracción del esqueleto por contracción en dirección gradiente con suavizado laplaciano. El software producido implementa una plataforma flexible para el diseño e implementación de plugins.

Métodos de Suavizado de Mallas

Los métodos para suavizar mallas reducen el ruido, la perturbación, las irregularidades y las irregularidades otras presentes en el modelo tridimensional de los modelos.

Métodos Laplacianos

La idea básica consiste en mover un vértice en la $E_0(t) = \frac{\partial X}{\partial t} = \mathcal{L}_L^t(X)$ misma dirección del Laplaciano.

La ecuación se implementa como la $E_0(t) = \mathcal{L}_L^t(X_{i+1}) = (I + \lambda L)^t X_i$, la ecuación de diferencias hacia adelante sea:

Donde I es la matriz de identidad, λ es la constante de iteración, L es el Laplaciano, y $t \in \mathbb{R}$ es la velocidad de difusión.

Y la aproximación discreta de la ecuación 2 es:

$$E_0(t) = \mathcal{L}_L^t(x_i) = \sum u_j(x_i - e_j), x_i \in \text{Vecinos}(x_i)$$

Aplicación del Laplaciano mediante la Curvatura normal



$$\text{Gra. } V_{ij} = \text{cot.}\alpha_j + \text{cot.}\beta_j \text{ para el vértice } i \text{ y sus vecinos } j.$$

Software Skeletonizer



Implementación para la extracción del esqueleto

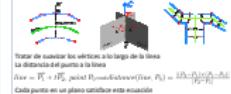
La suavización reduce la dimensionalidad y representa un cuerpo como un estructura uno-dimensional.

El esqueleto puede ser obtenido suavizando la malla pero bajo dos tipos de suavizaciones: suavizado Laplaciano y Rítmico que manejan los vértices en la localización original.

$$\text{Contracción del esqueleto: } \begin{bmatrix} W_1 L \\ W_2 L \\ \vdots \\ W_n L \end{bmatrix} X_{i+1} = \begin{bmatrix} 0 \\ W_1 V_i \\ \vdots \\ W_n V_i \end{bmatrix}$$

Donde $L = \text{suavizado Laplaciano con } \alpha = \text{cot.}\alpha_j + \text{cot.}\beta_j$

Basado en la localización de Rítmico



Si la norma de la curvatura oscila entre los vértices

$$\text{Iota} = \overline{P_1} - \overline{P_2}, \text{ donde } P_1 = \text{curvatura de } (v_1, P_1) = \frac{10(\alpha_1 + \beta_1)}{(\alpha_1 + \beta_1)}$$

Este punto en un plano certifica este resultado

$$P_1 = \alpha_1 + \beta_1 \gamma + \alpha_2 \gamma + \delta_1 = 0.$$

Resultados



- Que utilizan su planear mover a lo largo de la línea.
- El esqueleto tiene resultados claros.
- Muchas más extracciones que el algoritmo.
- La extracción dura en promedio a una regla particular de la linea.

Referencias y Agradecimientos

- Hsiao-Min Chang, Xu, Chien-Chih Lin, Yung-Huei Chu, David Cohen-Or, and Teng-Yen Lin. "Skeleton extraction by mesh contraction." ACM Transactions on Graphics (TOG) 27(5). 2008. 10
- Hsiao-Min Chang, Yung-Huei Chu, and James F. O'Brien. "Articulated mesh animation from multi-view silhouettes." ACM Trans. Graph. 27(3):1–9, 2008. 10
- Michael S. Corne and Patrick M. Flynn. "Curve skeleton properties, applications, and algorithms." Computer Vision and Image Understanding 70(1):1–24, 1998. 10
- Michael S. Corne and Patrick M. Flynn. "Curve skeleton properties, applications, and algorithms." Computer Vision and Image Understanding 70(1):1–24, 1998. 10
- Agustín Martínez, "The Laplacian performance data were provided courtesy of the Computer Graphics Group of the MIT CSAIL Vision Research [Cambridge, USA]."

Contacto

Alexander Pinto Fernández alexander.pinto@gmail.com
Grupo de Investigación Biología Computacional y Análisis de Datos
<http://www.udec.edu.cl/~apinto/>
Facultad de Medicina, Edificio AFI Póster Piso



SKELETON EXTRACTION

SOFTWARE PARA LA EXTRACCIÓN DEL ESQUELETO POR CONTRACCIÓN Y SUAVIZADO

Au et al. [Au2008] propose the next system of equations to iteratively contract the mesh until the volume is zero, next a simplification process is done and a skeleton appears.

$$\begin{bmatrix} W_L L \\ W_H \end{bmatrix} X_{t+1} = \begin{bmatrix} 0 \\ W_H X_t \end{bmatrix} \quad (10)$$

L is a matrix that used our TQLBO defined in equation 7.

W_L is a diagonal weighting matrix for the smoothing factor.

W_H is a diagonal weighting matrix for the attraction constraint factor.

A_i^t and A_i^0 are the current area and initial area of the ring surrounding x_i .

UNDESIRABLE DISPLACEMENT OF NODES FROM ROTATIONAL CENTER

SOFTWARE PARA LA EXTRACCIÓN DEL ESQUELETO POR CONTRACCIÓN Y SUAVIZADO

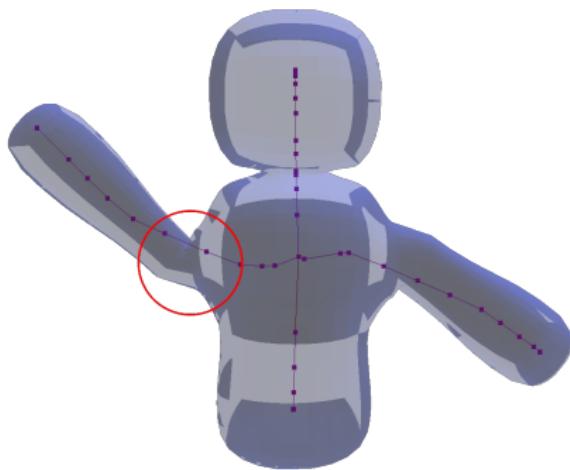


Figure: Skeleton that have a node outside of rotational center.

CONTRIBUCIÓN

SOFTWARE PARA LA EXTRACCIÓN DEL ESQUELETO POR CONTRACCIÓN Y SUAVIZADO

The basic idea is to move the vertices along a normal line, estimated at every vertex, based on the average of the normals of the faces.

This constraint eliminates the need to adjust the final skeleton.

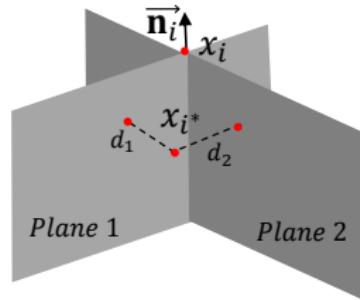
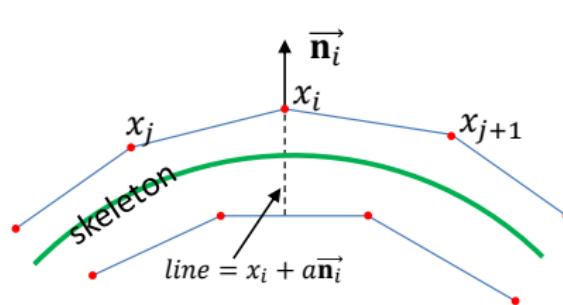


Figure: Left: The vertex x_i moves along the line constraint. Right: the distance of vertex x_i to plane 1 and plane 2 when the position in every iteration changes.

THE DISTANCE EQUATION OF POINT p TO PLANE Π

SOFTWARE PARA LA EXTRACCIÓN DEL ESQUELETO POR CONTRACCIÓN Y SUAVIZADO

The plane equation

$$a\mathbf{x} + b_0\mathbf{y} + c_0\mathbf{z} + d_0 = 0$$

The distance of point $P_0 = \{x_0, y_0, z_0\}$ to a plane

$$\Pi = a\mathbf{x} + b_0\mathbf{y} + c_0\mathbf{z} + d_0.$$

$$|\Pi - P_0| = \frac{|ax_0 + by_0 + cz_0 + d|}{\sqrt{a^2 + b^2 + c^2}} \quad (11)$$

THE NEW SYSTEM OF EQUATIONS PROPOSED

SOFTWARE PARA LA EXTRACCIÓN DEL ESQUELETO POR CONTRACCIÓN Y SUAVIZADO

$$\begin{bmatrix} W_L L \\ W_H \\ \Pi_1 \\ \Pi_2 \end{bmatrix} X_{t+1} = \begin{bmatrix} 0 \\ W_H X_t \\ -D_1 \\ -D_2 \end{bmatrix} \quad (12)$$

Π_1 and Π_2 are matrix that contain a, b, c values of the plane equation for every vertex.

D_1 and D_2 are the vectors with d values of the plane equation for every vertex.

USER INTERFACE

SOFTWARE PARA LA EXTRACCIÓN DEL ESQUELETO POR CONTRACCIÓN Y SUAVIZADO

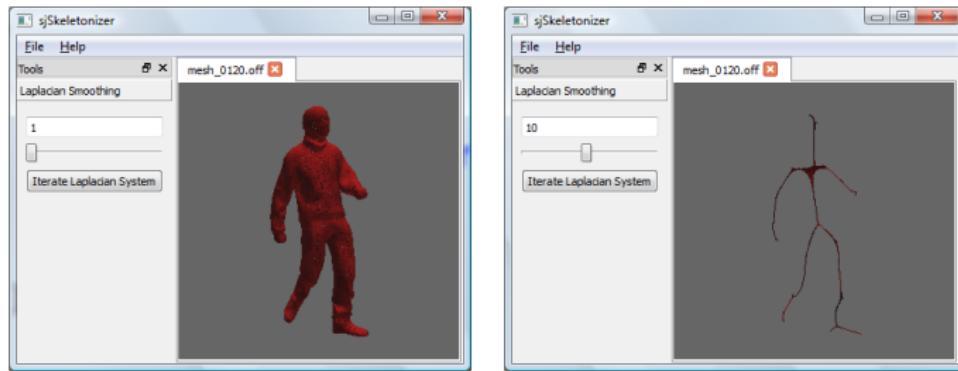


Figure: sjSkeletonizer is our prototype software application

As a result of this work, the software sjSkeletonizer permits the processing, visualization and extraction of the skeleton from polygonal hybrid meshes composed of triangles and quads.

RESULTS

SOFTWARE PARA LA EXTRACCIÓN DEL ESQUELETO POR CONTRACCIÓN Y SUAVIZADO

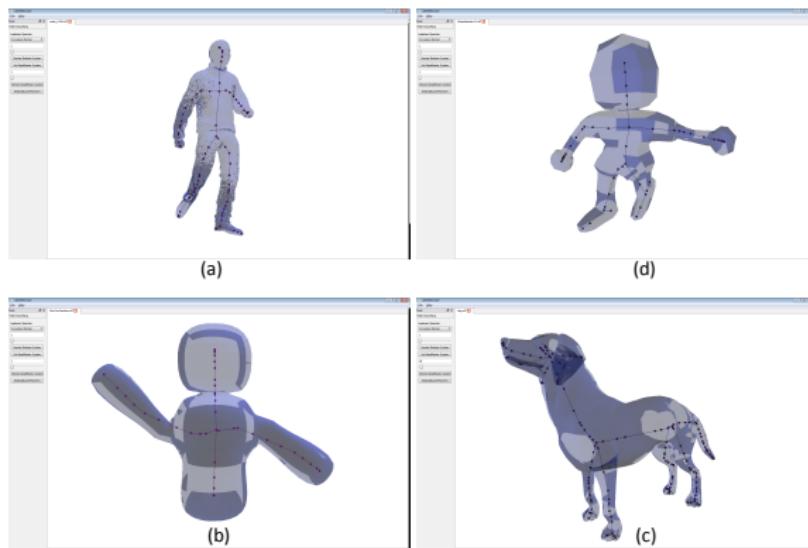


Figure: Skeleton extracted from different models. (a) Dog model (b) Character model. (c) Person model. (d) Clay model.

RESULTS

SOFTWARE PARA LA EXTRACCIÓN DEL ESQUELETO POR CONTRACCIÓN Y SUAVIZADO

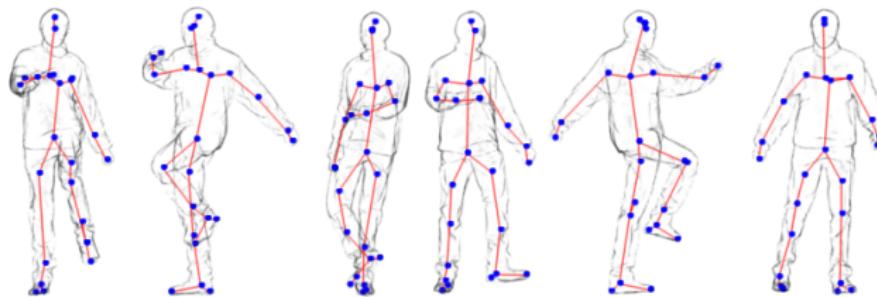


Figure: Model with different poses and skeleton obtained with our skeleton extraction software.

OUTLINE

- 1 Introduction
- 2 Proposed Method
- 3 Evaluation and Results
- 4 Products
- 5 Conclusions

CONCLUSIONS

- This work presented a novel extension of the Laplace Beltrami operator for hybrid quad/triangle meshes, and the successful application of such principles in different types of problems in computer geometric modeling like smoothing, enhancing, sculpting, deformation, reposing and skeleton extraction.
- A new sculpting brush to make a proper inflation while preserving the geometric details in real-time sessions was proposed, implemented and tested.
- We have largely demonstrated that our method has good performance, stability and robustness of the extension proposed.
- This novel extension of the Laplace Beltrami operator was introduced in the computer modeling industry inside the Blender 3D computer graphics software.

ACKNOWLEDGMENT

- We would like to thank CIM@LAB Research Group for their support of our research.
- This work was supported in part by the Blender Foundation, Google Summer of code program at 2012 and 2013.
- 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.

BIBLIOGRAPHY I

-  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.
-  Blender-Foundation.
Blender open source 3d application for modeling, animation, rendering, compositing, video editing and game creation.
<http://www.blender.org/>, 2012.
-  Mario Botsch, Leif Kobbelt, Mark Pauly, Pierre Alliez, and Bruno Lévy.
Polygon mesh processing.
CRC press, 2010.
-  Mathieu Desbrun, Mark Meyer, Peter Schroder, 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.
-  Michael S. Floater.
Mean value coordinates.
Computer Aided Geometric Design, 20(1):19 – 27, 2003.

BIBLIOGRAPHY II

-  Bruno Lévy and Hao (Richard) Zhang.
Spectral mesh processing.
In *ACM SIGGRAPH 2010 Courses*, SIGGRAPH '10, pages 8:1–8:312, New York, NY, USA, 2010. ACM.
-  Mark Meyer, Mathieu Desbrun, Peter Schroder, 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.
-  Tony Mullen.
Introducing character animation with Blender.
Indianapolis, Ind. Wiley Pub. cop., 2007.
-  Ulrich Pinkall, Strasse Des Juni, and Konrad Polthier.
Computing discrete minimal surfaces and their conjugates.
Experimental Mathematics, 2:15–36, 1993.

BIBLIOGRAPHY III



Alexander Pinzon and Eduardo Romero.

Análisis experimental de la extracción del esqueleto por contracción con suavizado laplaciano, December 2010.

Poster presented at 6th International Seminar on Medical Image Processing and Analysis (SIPAIM 2010), December 1–4, Universidad Nacional de Colombia, Bogota, Colombia.



Alexander Pinzon and Eduardo Romero.

Software para la extracción del esqueleto por contracción y suavizado, December 2011.

Poster presented at 7th International Seminar on Medical Image Processing and Analysis (SIPAIM 2011), December 6–8, Universidad Industrial de Santander, Bucaramanga, Colombia.



Alexander Pinzon and Eduardo Romero.

sjskeletonizer: Skeleton extraction software., March 2012.
<http://code.google.com/p/jeronimo/>.



Alexander Pinzon and Eduardo Romero.

Shape inflation with an adapted laplacian operator for hybrid quad/triangle meshes.

2013 26th SIBGRAPI Conference on Graphics, Patterns and Images, 0:179–186, 2013.

BIBLIOGRAPHY IV



Gabriel Taubin.

A signal processing approach to fair surface design.

In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '95, pages 351–358, New York, NY, USA, 1995. ACM.



B. Vallet and B. Lévy.

Spectral geometry processing with manifold harmonics.

Computer Graphics Forum, 27(2):251–260, 2008.



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.