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 DE COLOMBIA

cim@lab Computer Image & Medical Applications Laboratory

Outline

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.

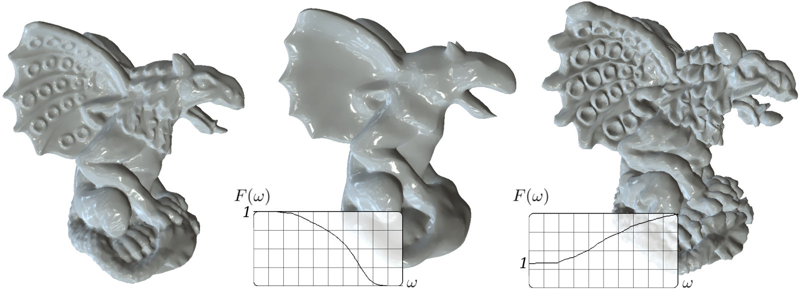


Figure: Original Mesh, Low-pass and enhancement filters [Vallet08].

The basis functions of the Fourier transform

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

The Fourier transform of $f(x)$

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

The Inverse Fourier transform of $F(w)$

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

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

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

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

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

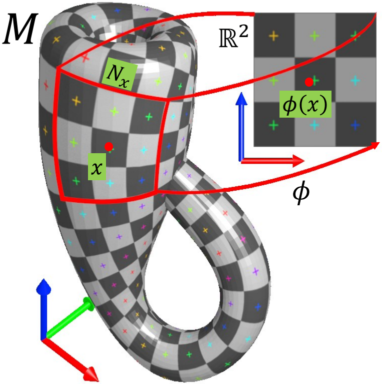
$$\Delta\left(\mathbf{e}_w\right) = \lambda_w \mathbf{e}_w$$

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

$$\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial z^2} = 0$$
$$\nabla_M^2 f = \Delta_M = 0$$
$$\Delta_M = div\,(\nabla_M f)$$

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

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

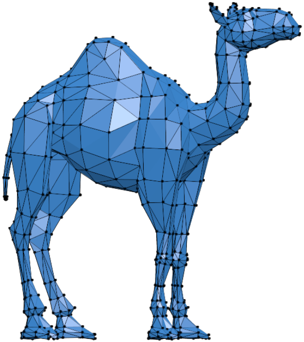


A Surface M is a 2D topological manifold
For which every point $x \in M \subseteq \mathbb{R}^3$ has a neighbourhood N_x homeomorphic to euclidean space $\mathbb{R}^2$.

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

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.

Given a polygonal mesh $\mathbf{M} = [v_1, \ldots, v_n]^T$ 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 define 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.

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

- 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].

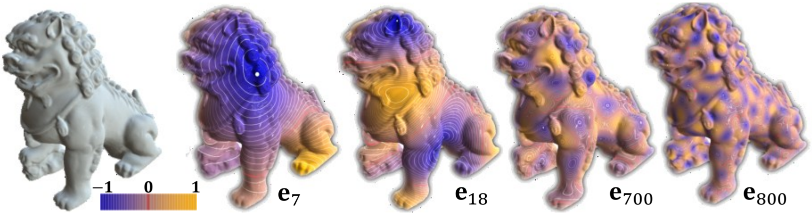


Figure: Color values are the amplitude of a wave $\mathbf{e}_i$ projected on the mesh [Vallet08].

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

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

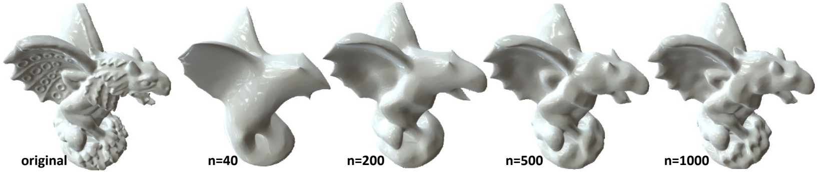


Figure: Mesh reconstruction with first n eigenvectors of the Discrete Laplacian Operator [Levy10].

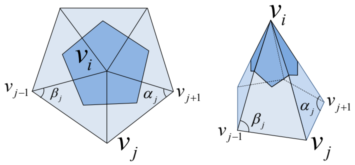
Desired property:
Non-negative weights ω_{ij} for $i \neq j$ ensure a positive semi-definite matrix.

- 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.



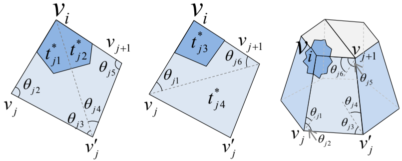
$$w_{ij} = \frac{1}{4A_i}\left(\cot\alpha_j + \cot\beta_j\right)$$

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**.

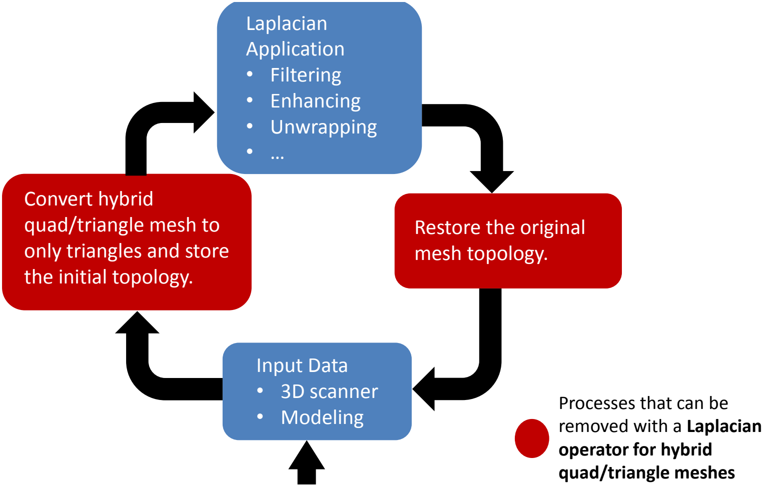
Xion's operator is the mean of Desbrun's operator for all possible triangulations for quad meshes.



$$w_{ij} = \frac{1}{4A_i}\left(\cot\theta_{(j-1)1} + \cot\theta_{(j-1)4} + \cot\theta_{j3} + \cot\theta_{j6}\right)$$
$$w_{ij'} = \frac{1}{2A_i}\left(\cot\theta_{j2} + \cot\theta_{j5}\right)$$
$$A_i = \frac{1}{2}\sum_{j\in i*}\left(A\left(t_{j2}\right) + A\left(t_{j2}\right) + A\left(t_{j3}\right)\right)$$

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.

General 3D modeling process on polygonal meshes.



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

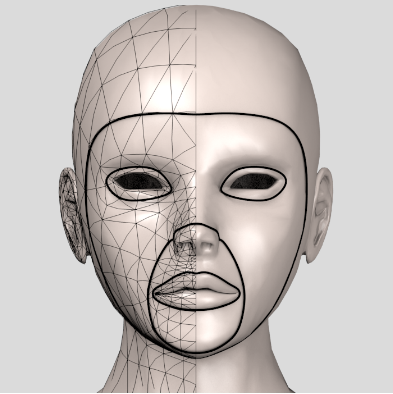


Figure: Triangle Mesh

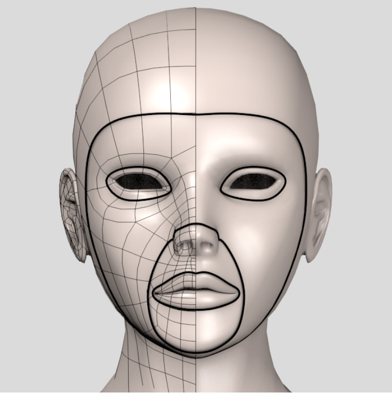


Figure: Hybrid Quad/Triangle Mesh

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.

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

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

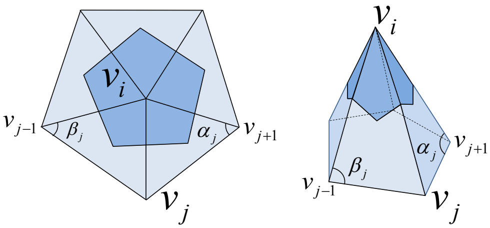


Figure: Area of the Voronoi region around v_i in dark blue. v_j belong to the first neighborhood around v_i. α_j and β_j opposite angles to edge $\overrightarrow{v_j - v_i}$.

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].

Gradient of voronoi region

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

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].

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

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.

$$Area(Q) = \frac{Area_1 + Area_2}{2} = \frac{A(t^*_{j1}) + A(t^*_{j2}) + A(t^*_{j3})}{2}$$

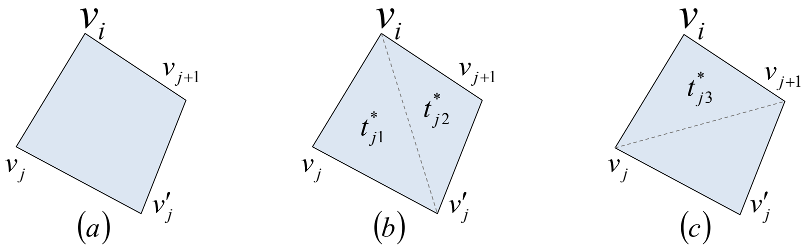


Figure: (a) Original quad. (b) Primal triangulation around v_i is $t^*_{j1} \equiv \triangle\, v_i v_j v'_j$, $t^*_{j2} \equiv \triangle\, v_i v'_j v_{j+1}$ (b) Dual triangulation around v_i is $t^*_{j3} \equiv \triangle\, v_i v_j v_{j+1}$.

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

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

where $q_j \in Q_{v_i}$ and $t_k \in T_{v_i}$
Applying Xiong's mean average area to (1)

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

Laplace Beltrami operator for hybrid quad/triangle meshes

Applying the gradient operator to (2)

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

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

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

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

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

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

Therefore (3) can be rewritten as

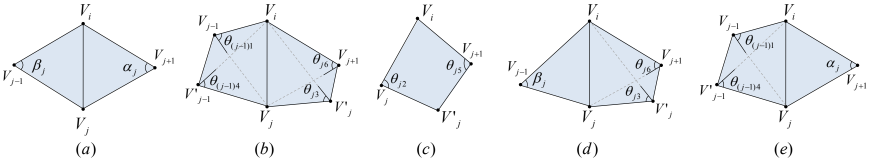
$$\nabla A(v_i) = \sum_{j=1}^{n} w_{ij}(v_j - v_i) \tag{4}$$

where v_j are the neighbors of v_i
Using the relationship between Laplace-Beltrami operator $\triangle$ and Mean curvature normal in (4) we define the **Triangle Quad Laplace Beltrami Operator TQLBO** as

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

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

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.

$$w_{ij} = \begin{cases} \left(\cot\alpha_j + \cot\beta_j\right) & \text{case } a. \\ \frac{1}{2}\left(\cot\theta_{(j-1)1} + \cot\theta_{(j-1)4} + \cot\theta_{j3} + \cot\theta_{j6}\right) & \text{case } b. \\ \left(\cot\theta_{j2} + \cot\theta_{j5}\right) & \text{case } c. \\ \frac{1}{2}\left(\cot\theta_{j3} + \cot\theta_{j6}\right) + \cot\beta_j & \text{case } d. \\ \frac{1}{2}\left(\cot\theta_{(j-1)1} + \cot\theta_{(j-1)4}\right) + \cot\alpha_j & \text{case } e. \end{cases} \tag{6}$$

We define a TQLBO as a matrix equation

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

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

A standard diffusion process is used.

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

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

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

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

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.

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

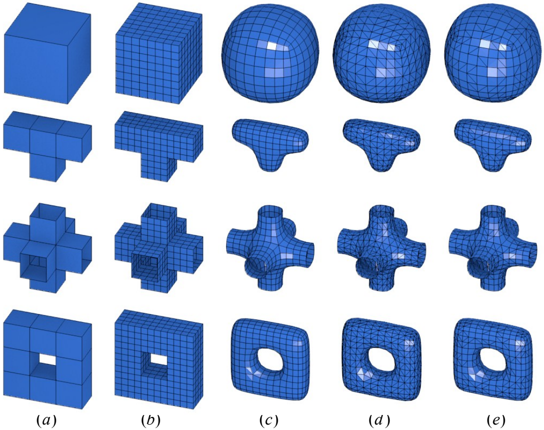


Figure: (a) Original Model. (b) Simple subdivision. (c), (d) (e) Laplacian smoothing with $\lambda = 7$ and 2 iterations: (c) for quads, (d) for triangles, (e) for triangles and quads random chosen.

Shape Inflation with enhancing filter results

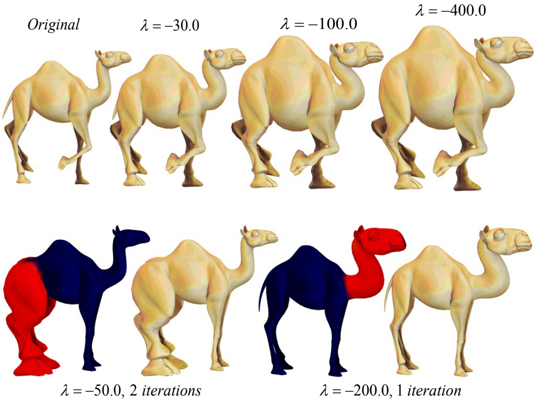


Figure: Top row: Original camel model in left. Shape inflation with $\lambda = -30.0$, $\lambda = -100.0$, $\lambda = -400.0$. Bottom row: Shape inflation with weight vertex group, $\lambda = -50.0$ and 2 iterations for the legs, $\lambda = -200.0$ and 1 iteration for the head and neck.

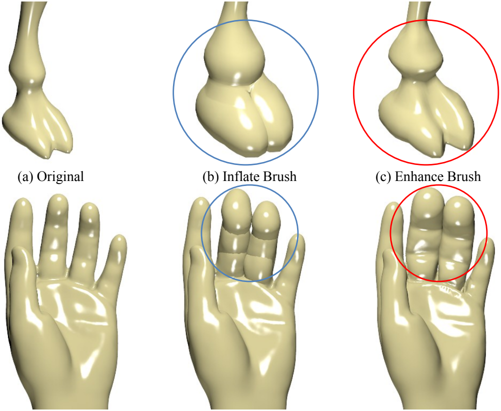


Figure: Top row: (a) Leg Camel, (b) Traditional inflate brush for leg into blue circle, (c) Shape inflation brush for leg into red circle. Bottom row: (a) Hand, (b) Traditional inflate brush for fingers into blue circle, (c) Shape inflation brush for fingers in red circle.

Inflate brush performance

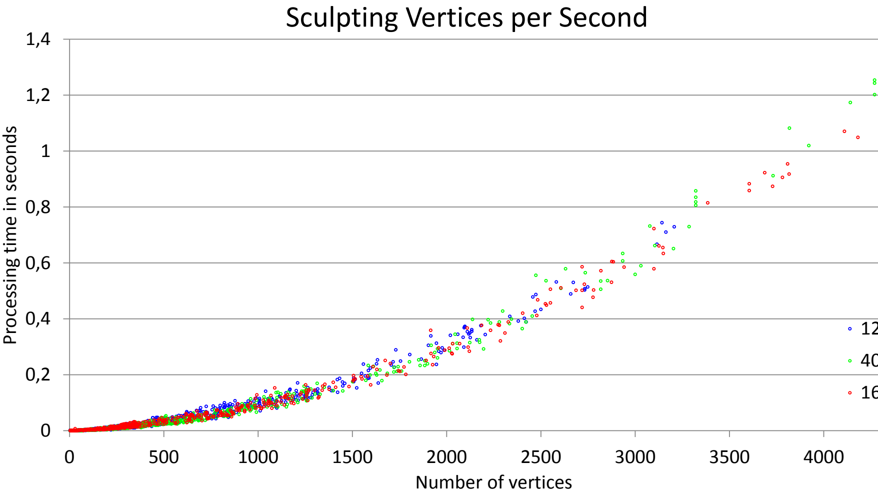


Figure: Performance of our dynamic shape inflation brush in terms of the sculpted vertices per second. Three models with 12K, 40K, 164K vertices used for sculpting in real time.

Outline

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: apinzonf@unal.edu.co, edromero@unal.edu.co

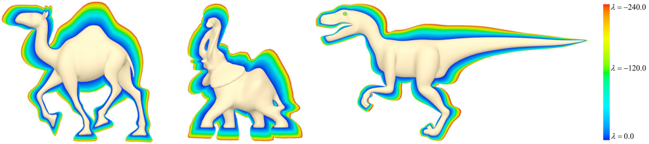


Fig. 1. A set of 48 successive shapes enhanced, from $\lambda = 0.0$ in blue to $\lambda = -240.0$ in red, with steps of -5.0.

***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

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.

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

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

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

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

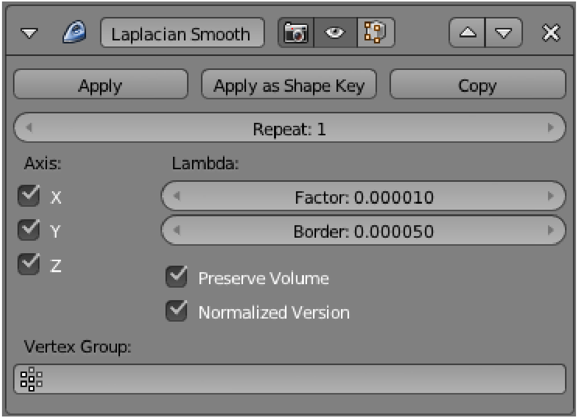


Figure: Panel inside blender user interface of the Laplacian Smooth modifier tool.

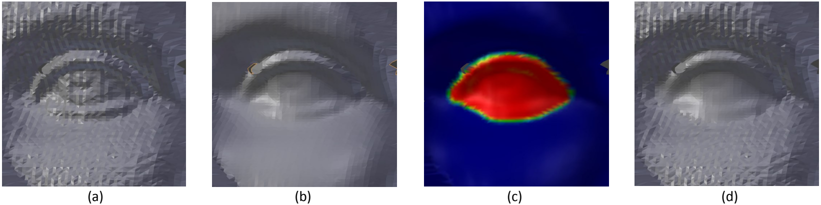


Figure: Use of weights per vertex to constrain the effect of mesh smoothing. (a) Original Model. (b) Smoothing with $\lambda = 1.5$ (c) red vertices *weight* $= 1.0$, blue vertices *weight* $= 0.0$. (d) Smoothing with $\lambda = 2.5$. The red vertices were the only vertices smoothed.

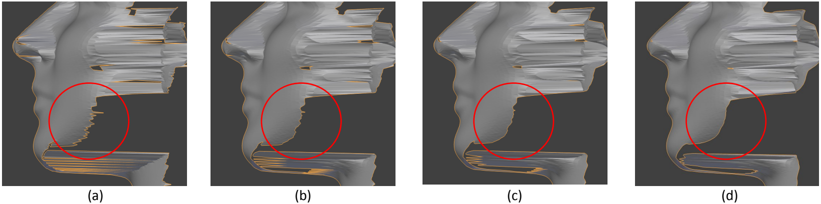


Figure: Smoothing boundary changing λ_{Border} factor. (a) Original Model. (b) Smoothing $\lambda_{Border} = 1.0$. (c) Smoothing $\lambda_{Border} = 2.5$ (d) Smoothing with $\lambda_{Border} = 10.0$.

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.

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

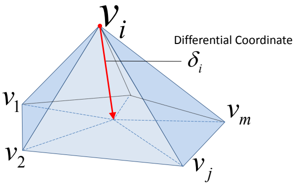


Figure: Difference between v_i and the center of mass of its neighbors $v_1, \ldots, v_m$.

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

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

$$\begin{bmatrix} L \\ W_c \end{bmatrix} V = \begin{bmatrix} \delta \\ W_c C \end{bmatrix} \tag{9}$$

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

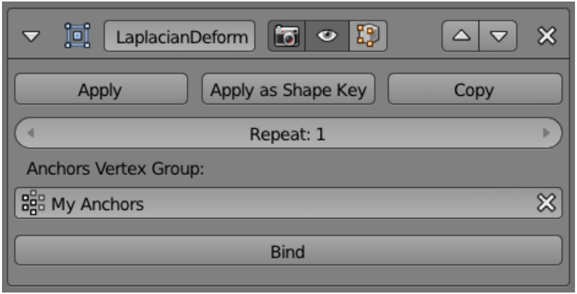


Figure: Panel inside Blender user interface of the Laplacian Deform modifier tool.

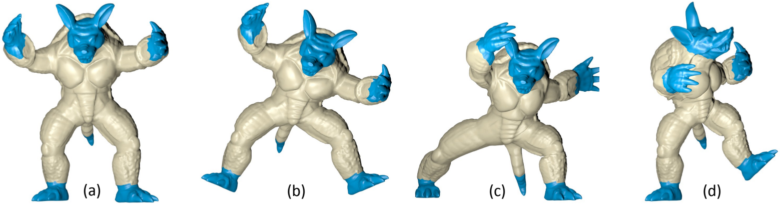


Figure: Anchor vertices in blue. (a) Original Model, (b,c,d) new poses only change the anchor-vertices, the system finds positions for vertices in yellow.

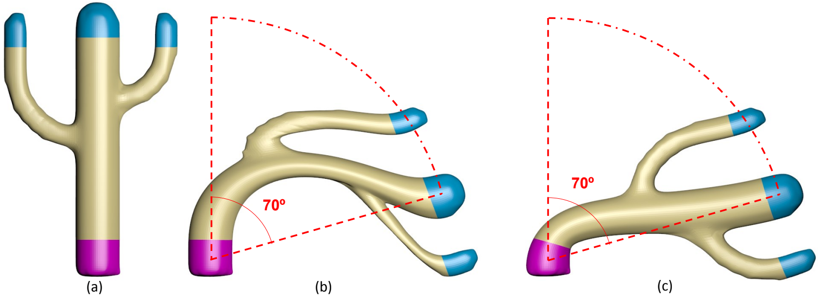


Figure: (a) Original cactus model. (b) Blue segments are rotated 70º to the right and afterwards a basic interpolation is applied to the parts in yellow (c) Blue segments are rotated 70º to the right and afterwards a Laplacian deform tool is applied to the parts in yellow.

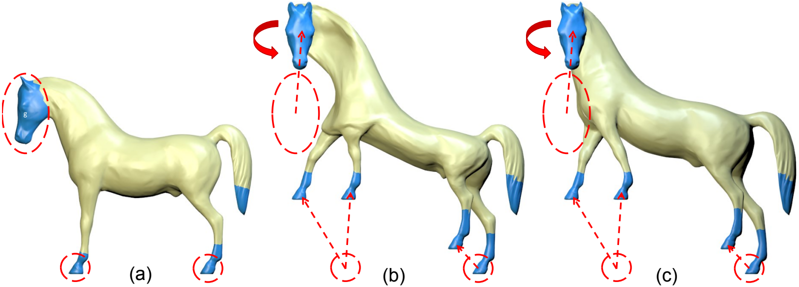


Figure: (a) Original Horse model. (b) The blue segments are translated and rotated and then basic interpolation is applied to the yellow parts (c) The blue segments are translated and rotated and then the Laplacian Deform tool is applied to the yellow parts.

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

Alexander Pinzón, Fabio Martínez, Eduardo Romero

Abstract

Este articulo presenta un análisis sistemático del método de extracción del esqueleto por medio de la contracción de un volumen con suavizado Laplaciano. El trabajo realiza una aproximación experimental al problema de la extracción del esqueleto, para encontrar el rendimiento del método evaluado frente a cambios isométricos, y durante la fase de simplificación.

Esta evaluación utilizo el modelo tridimensional animado de una persona que realizaba una caminata, a este modelo se le extrajo el esqueleto y se compararon las diferencias en distintos instantes de la animación, y distintas configuraciones del proceso de simplificación. Los resultados muestran un óptimo rendimiento del método frente a las transformaciones isométricas, y múltiples problemas en la fase de simplificación de mallas.

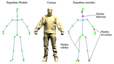


Fig. 2. Proceso de extracción del esqueleto

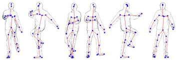


Fig. 3. Variación del numero de nodos

Contracción con suavizado Laplaciano

Este método contrae iterativamente una malla de polígonos por medio del suavizado laplaciano hasta tener un volumen de cero ver figura 1.
La contracción es tomada como un problema de minimización de energía, con los siguientes términos.

$$\|W_L L V'\|^2 + \sum_i W_{H,i}^2 \|V'_i - V_i\|^2$$

- ***L***: Operador Lapaciano para remover las frecuencias altas, es decir suavizar los detalles de la geometría
- ***W_H***: Fuerza de atracción que usa los vértices, para mantener información clave de la geometría durante la contracción.
- ***W_L***: Fuerza de contracción que hace que la forma tridimensional pierda volumen.

Fig. 1. Proceso de extracción del esqueleto

Experimentación

Se uso la implementación hecha en Oscar Kin-Chung Au et al. Por los autores, para realizar la extracción del esqueleto de un modelo tridimensional que fue obtenido mediante el método de recuperación de forma desde las siluetas realizado por Vlasic et al. De este modelo se registro una caminata durante 240 cuadros.
Para evaluar el proceso de simplificación se variaron el numero de nodos usados para describir uniones en el esqueleto (ver figura 3) y se clasificaron las uniones así ver figura 2.

En el segundo experimento se seleccionaron diferentes poses para observar la correspondencia topológica entre los esqueletos extraídos y analizar el comportamiento del método frente a transformaciones isométricas de la geometría de un cuerpo.

Resultados

El método recupera de forma óptima el esqueleto de un cuerpo bajo transformaciones isométricas, ver la tabla 1, se recuperaron en promedio 19.71 nodos de los 21 necesarios para reconstruir el esqueleto en diferentes poses.
El método no pudo recuperar 1.86 nodos de los necesarios para reconstruir totalmente el esqueleto. La línea verde en la figura 3 describe el numero de nodos que hacen falta para recuperar el modelo.

	Validos	Erróneos	Faltantes
Media	19.71	4.83	1.86
SD	1.47	1.47	1.47
Min	17	3	0
Max	21	7	4

Tabla 1. Resultados obtenidos al realizar transformaciones isométricas

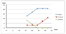


Fig. 3. Gráfico de la variación de nodos validos, erróneos y faltantes durante la fase de simplificación

Durante la fase de simplificación (ver figura 3) el método no sufre perdida ni es sensible al uso de mas de 24 nodos (línea azul). El método tiene un numero mínimo de 3 nodos recuperados erróneamente como se observa el la línea roja de la figura 3.

Referencias y Agradecimientos

- 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.
- Daniel Vlasic, Ilya Baran, Wojciech Matusik, and Jovan Popović. **Articulated mesh animation from multi-view silhouettes**. *ACM Trans. Graph.*,27(3):1–9, 2008. 3D Reconstruction.
- 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.

Agradecimientos: The captured performance data were provided courtesy of the Computer Graphics Group of the MIT CSAIL Vision Research (Cambirdge, USA).

Conclusiones y Trabajo Futuro

El método de extracción mostró ser robusto y tener baja sensibilidad frente a cambios isométricos de la geometría, el método puede trabajar de forma automática a o largo de todos los cuadros. El método recupera de forma eficiente el esqueleto sin hacer uso de un modelo, eliminando la necesidad de estimar la pose. El método permite realizar de forma sencilla y automática el seguimiento del esqueleto a lo largo del vídeo.

Como trabajo futuro es posible mejorar la recuperación de información haciendo uso de la coherencia espacio temporal no presente en la técnica de extracción, para superar la perdida de información entre cuadros de vídeo. Es posible automatizar el proceso de simplificación para encontrar el numero óptimo de nodos con lo cual puede ser representado el esqueleto, haciendo uso de algoritmos de partición de mallas.

Contacto
Alexander Pinzón Fernández apinzonf@gmail.com
Grupo de Investigación Bioingenium www.bioingenium.unal.edu.co
Universidad Nacional de Colombia www.unal.edu.co
Facultad de Medicina, Edificio 471 Primer Piso

Software para la Extracción del Esqueleto por Contracción y Suavizado

Alexander Pinzón, Eduardo Romero

Abstract

Este articulo presenta un software para el procesamiento, visualización, y extracción del esqueleto desde mallas de polígonos. El software se diseño con base en un sistema de plugins y filtros, se implemento un plugin que contenía un filtro para la extracción del esqueleto por contracción en dirección gradiente con suavizado Laplaciano. El software producido proporciona una plataforma flexible para el diseño e implementación de plugins.

Métodos de Suavizado de Mallas

Los métodos para suavizar mallas reducen el ruido, o permiten iterativamente eliminar frecuencias altas presentes en el muestreo tridimensional de los modelos.

Métodos Laplacianos

La idea básica consiste en mover un vértice en la misma dirección del Laplaciano . $Eq(1) \quad \frac{\partial X}{\partial t} = \lambda L(X)$

La ecuación 1 se implementa como la ecuación de diferencias hacia adelante así: $Eq(2) \quad X_{t+1} = (I + \lambda L) X_t$

Donde ***X*** es el conjunto de vértices, *L* es el Laplaciano, y $\lambda \in \Re$ es la velocidad de difusión.

Y la aproximación discreta de la ecuación 2 es:

$$Eq(3) \quad L(x_i) = \sum w_{ij}(x_j - x_i), \quad x_j \in Vecinos(x_i)$$

Aproximación del Laplaciano mediante la Curvatura normal

Con $w_{ij} = \cot\alpha_j + \cot\beta_j$ para el vértice x_i y sus vecinos x_j

Software Skeletonizer

sjSkeletonizer es el software desarrollado en el grupo Bioingenium para el procesamiento, visualización y extracción del esqueleto desde mallas de polígonos.

- Usa **CGAL** (Computational Geometry Algorithms Library)
- Usa **Graphite** (Software de Geometría Numérica y Computación Grafica)
- Se integraron las siguientes librerías de procesamiento numérico: ACE, AMD, ARPACK, ARPACK_UTIL, CBLAS, CCOLAMD, CHOLMOD, CLAPACK, COLAMD, F2CLIBS, METIS, MISC, NL, SUPERLU, TAUCS

Implementación para la extracción del esqueleto

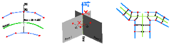
La Esqueletonización reduce la dimensionalidad y representa un cuerpo como un estructura uní-dimensional.

El esqueleto puede ser obtenido suavizando la malla pero bajo dos restricciones, W_L que da peso al Laplaciano y W_H que mantiene los vértices en su localización original.

Extracción del esqueleto. $\begin{bmatrix} W_L L \\ W_H \end{bmatrix} X_{t+1} = \begin{bmatrix} 0 \\ W_H X_t \end{bmatrix}$

Donde *L(X)* = Suavizado Laplaciano con w_{ij} basado en la curvatura de flujo $w_{ij} = \cot\alpha_j + \cot\beta_j$

Y la nueva restricción propuesta en este trabajo



Tratar de suavizar los vértices a lo largo de la línea

La distancia del punto a la linea

$$line = \overrightarrow{P_1} + t\overrightarrow{P_2},\ point\ P_0 \Longrightarrow distance(line, P_0) = \frac{|(P_2 - P_1) \times (P_1 - P_0)|}{|P_2 - P_1|}$$

Cada punto en un plano satisface esta ecuación

$$P_0 : a\mathbf{x} + b_0\mathbf{y} + c_0\mathbf{z} + d_0 = 0.$$

Resultados

- Los vértices se pueden mover a lo largo de la línea.
- El esqueleto tiene muchas ramas.
- Muchas mas ecuaciones que incógnitas.
- La solución debe ser restringida a una región particular de la línea.

Referencias y Agradecimientos

- 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.
- Daniel Vlasic, Ilya Baran, Wojciech Matusik, and Jovan Popović. **Articulated mesh animation from multi-view silhouettes**. *ACM Trans. Graph.*,27(3):1–9, 2008. 3D Reconstruction.
- 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.

Agradecimientos: The captured performance data were provided courtesy of the Computer Graphics Group of the MIT CSAIL Vision Research (Cambirdge, USA).

Contacto

Alexander Pinzón Fernández apinzonf@gmail.com
Grupo de Investigación Bioingenium www.bioingenium.unal.edu.co
Universidad Nacional de Colombia www.unal.edu.co
Facultad de Medicina, Edificio 471 Primer Piso

Bioingenium

Universidad Nacional de Colombia

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.

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

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.

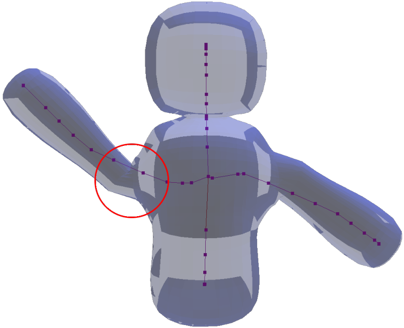


Figure: Skeleton that have a node outside of rotational center.

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.

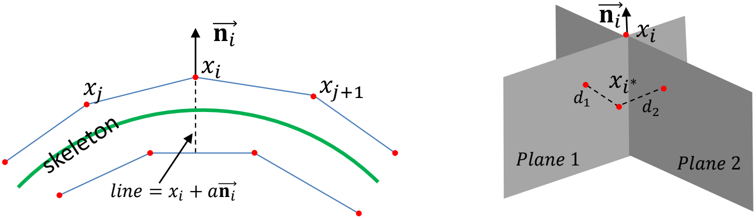


Figure: Left: The vertex x_i moves along the line constraint. Right: the distance of vertex x_i to plane 1 and plane 2 when the position in every iteration changes.

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

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

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

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

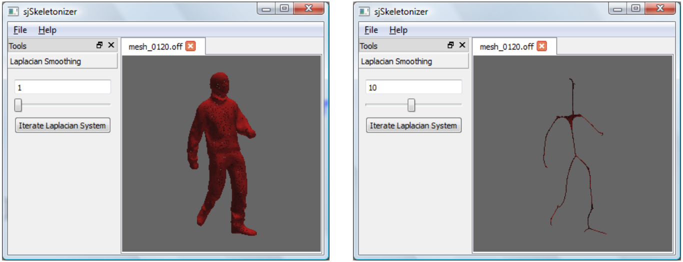


Figure: sjSkeletonizer is our prototype software application

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.

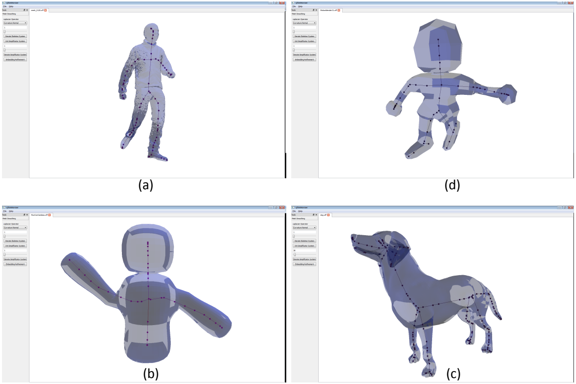


Figure: Skeleton extracted from different models. (a) Dog model (b) Character model. (c) Person model. (d) Clay model.

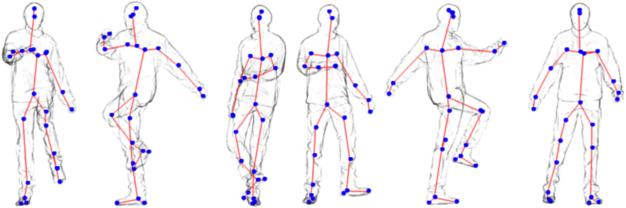


Figure: Model with different poses and skeleton obtained with our skeleton extraction software.

- 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 made 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 wold 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.

Alexander Pinzon and Eduardo Romero.
An?lisis 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.

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.