Laplacian Subdivision Surfaces

Alexander Pinzon*
Cimalab Research Group

Eduardo Romero†
Cimalab Research Group

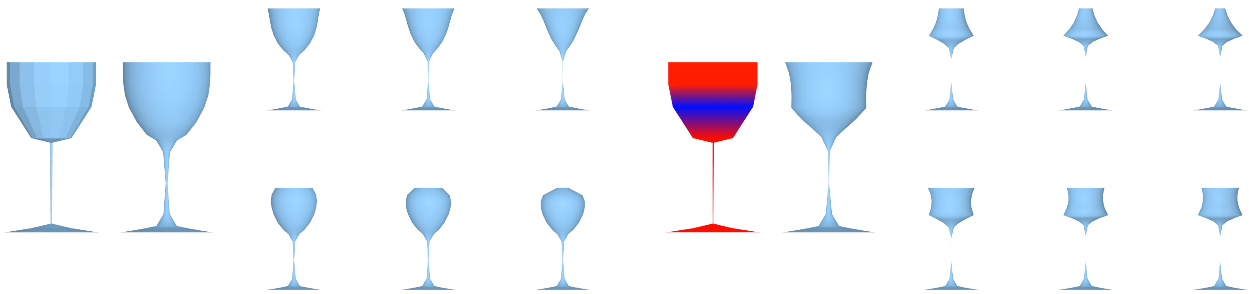

Figure 1: *Laplacian subdivision surface with weigth vertex group*

Abstract

This paper proposes a novel method for modelling poligonal mesh using subdivision surface and laplacian smoothing. This method use laplacian smooth for modelling global curvature in the model, to permit most flexible, robust and predictable results.

This method can correct traditional problems in extraordinary vertices present at catmull-clark subdivision method. The convergent rate of the laplacian smooth can be controlled by adjusting the weight in lambda parameter.

This proposal contains NN novel features

ESTE ABSTRACT NO SIRVE

(**we provide a series of examples to graphically and numerically demonstrate the quality of our results.**) debe ser cambiado copiado de desbrun 99

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling —Modeling packages

I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling —Curve, surface, solid, and object representations

Keywords: laplacian smooth, subdivision surface

1 Introduction

Mencionar la gran importancia que tiene trabajar con mallas de triangulos y poligonos, ya no se nececitara preprocesar la malla para poder aplicar alguna tecnica de procesamiento que involucre el operador laplace beltrami. en el mundo real los dise;adores crean modelos con triangulos y cuadrados.

Mesh modelling from a coarse geometry is an important step in computer graphics process such as design, art and other fields.

Nuestro metodos propone tre cosas muy novedosas:

*e-mail: apinzonf@gmail.com
†e-mail:edromero@unal.edu.co

Operador Laplace beltrami para mallas de topologia arbitraria formadas por triangulos y cuadrados, para cualquier tipo de procesamiento en geometria diferencial

Permite generar una familia de formas parametrizadas.

Controlar el nivel de suavizado y curvatura al subdividir mallas de poligonos

1.1 Previous work

1.2 Overview of our method

Nuestro metodo usa el un bosquejo de maya le aplica una subdivision cualquiera y esa subdivision la modifica a lo largo de su curvatura de flujo usando un operador laplaciano para triangulos y cuadrados

2 Theoretical Review of Related Work

2.1 Subdivision Surface

Subdivision is an iterated transformation [Warren and Weimer 2001]. Let F be a function (subdivision transformation) that maps one geometry M_i into another similar geometry with same topology M_{i+1}.

$$M_{i+1} = F\left(M_i\right) \tag{1}$$

The Catmull-Clark subdivision transformation is used to smooth a surface as the limit of sequence of subdivision steps[Stam 1998]. This method do a recursive subdivision transformation that refines the model into a linear interpolationthat is a approximate smooth surface. The process of Catmull-Clark is govern o properties of B-spline curve from multivariate spline theory[Loop 1987].

In many subdivision surfaces methods catmull clark loop so on. the smoothness of the model is autmaically guarantteed[DeRose et al. 1998].

Subdivision surfaces with catmull clark is continuos except at a extraordinary points[Loop 1987], but with our method can correct this problem

2.2 Curvature flow

La curvatura la tomamos segun desbrun 1999

2.3 Laplacian smooth

The functional used in many related work to constrain energy minimization is based on a total curvature of a surface S.

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

3 Proposed Method

Our method simplifies the design of irregular polygon meshes, generating a parameterized family of shapes using a set of vertices representing a coarse sketch of the desired model. Our method is iterative and converges towards a continuous and smooth version of the original model.

Unlike other methods, our method allows use arbitrary topologys types of representation as triangles and quads, exploiting the basic geometrical relationships facilitating and ensuring convergence of the algorithm and similar shapes consistent with the original shape against the other methods.

Our method allows the use of soft constraints weighting the effect of smoothing at each vertex based on a normalized weight, the weights are assigned to the control vertices of the original mesh or. The weights of the new vertices resulting from the iterations are calculated by interpolation, allowing to modify the behavior of the method on exact regions of the original model.

3.1 Laplace Beltrami operator over triangular and quadrilateral meshes

3.1.1 Gradient of Voronoi Area

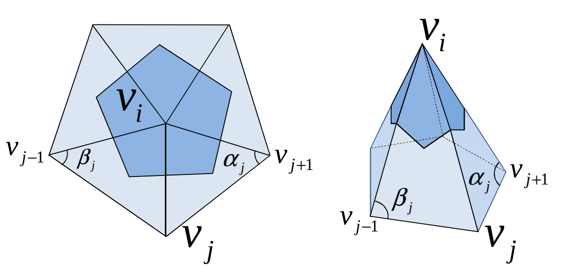

Figure 2: *Area of Voronoi region around v_i in dark blue. v_j 1-ring neighbors around v_i. α_j and β_j opposite angles to edge $\overrightarrow{v_j - v_i}$.*

Consider a surface S compound by a set of triangles around vertex v_i. We can define the *Voronoi region* of v_i as show in figure 2, The change in area produced by move v_i is named gradient of *Voronoi region,* if we normalize this gradient by the total area in ring around v_i, we have the *discrete mean curvature normal* of a surface S as shown in equation 3 [Desbrun et al. 1999].

$$\overline{\kappa}\mathbf{n} = \frac{\nabla A}{2A} = \frac{1}{4A}\sum_j (\cot\alpha_j + \cot\beta_j)(v_i - v_j) \tag{3}$$

3.1.2 Normalized LBO over triangular and quadrilateral meshes

based on [Desbrun et al. 1999] and [Xiong et al. 2011]

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

Where $N(v_i)$ is the 1-ring neighbors with shared face to v_i, and δ_{ij} being the Kronecker delta function.

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

3.1.3 Curvature normal calculation

Given a mesh $M = (V, Q, T)$, with vertices V, quads Q, triangles T.

The area of 1-ring neighborhood (N_1) with shared face to vertex v_i in M is.

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

Appling the mean average area of all posible triangulations for each quad.

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

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

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

Applying the gradient operator MLBO propose by [Xiong et al. 2011] to quads and LBO propose by [Desbrun et al. 1999] to triangles in equation (4).

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

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

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

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

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

3.1.4 Normalized curvature operator

Normalizarlo igual que los otros metodos

3.2 Boundaries scale-dependent

El manejo de los bordes con operador dependiente de escala

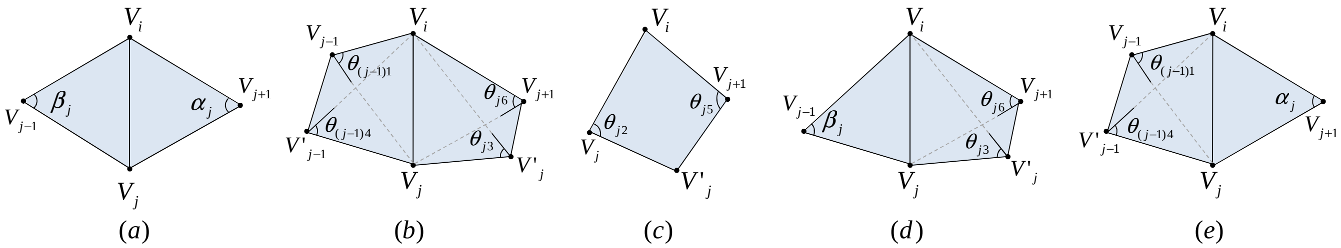

Figure 3: *The 5 basic triangle-quad cases with common vertex V_i and the relationship with V_j and V'_j. (a) Two triangles [Desbrun 1999]. (b) (c) Two quads and one quad [Xiong 2011]. (d) (e) Triangles and quads (TQLBO).*

3.3 Anti-shrinking fairing - Volume preservation

Preservacion de volumen en el centroide funciona mejor que lo propuesto por desbrum 99

The diffusion process, induce shrinkage [Desbrun et al. 1999].

3.4 Weight based somooth constraints

Las familias que se generan pueden cambiar substancialmente con el ponderamiento de puntos de control especificos

4 Experimental Results and Applications

Se trabajo con blender software blblbsd

se hicieron pruebas de rendimiento asdadasd

Pruebas de catmull clark vs nuestro metodo

pruebas de T Q, y nuestro metodos de Ty Q

Pruebas generando familias de objetos, que involucran y no CatCl.

4.1 Implementation

c y c++ blender.

4.2 Sparse linear system

superlu opennl

5 Conclusion and future work

optimizacion del metodo de solucion

aplicarlo en otras areas

Acknowledgements

CIM&LAB Computer Image & Medical Applications Laboratory.

Blender Foundation.

Google Summer of code program.

References

DEROSE, T., KASS, M., AND TRUONG, T. 1998. Subdivision surfaces in character animation. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, SIGGRAPH '98, 85–94.

DESBRUN, M., MEYER, M., SCHRÖDER, P., AND BARR, A. H. 1999. Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, ACM Press Addison-Wesley Publishing Co., New York, NY, USA, SIGGRAPH '99, 317–324.

LOOP, C. 1987. *Smooth Subdivision Surfaces Based on Triangles*. Department of mathematics, University of Utah, Utah, USA.

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

WARREN, J., AND WEIMER, H. 2001. *Subdivision Methods for Geometric Design: A Constructive Approach*, 1st ed. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

XIONG, Y., LI, G., AND HAN, G. 2011. Mean laplace-beltrami operator for quadrilateral meshes. In *Transactions on Edutainment V*, Z. Pan, A. Cheok, W. Muller, and X. Yang, Eds., vol. 6530 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 189–201.