Dent Removal: Geodesic Curve-Based Mesh Fairing

Dmitriy Pinskiy

Walt Disney Animation Studios

Abstract

This paper presents a novel mesh fairing method to remove unwanted geometric artifacts such as dents. The key element of the proposed method is our unique algorithm for the assignment of weights in the discrete Laplacian. Our weights reflect the orientation of the first ring of vertices around a given vertex with respect to the whole dented region, rather than the ring's local geometry (e.g. Gauss or mean curvature flow that needs to be re-estimated per each iteration). To establish the dent region's global behavior, we perform principal curvature analysis to determine representative geodesic curves for the surface. Our method is efficient as it only needs to compute the geodesic curves once, and then it uses them to guide subsequent iterations of anisotropic relaxation.

Categories and Subject Descriptors: G.1.1 [Numerical Analysis]: Smoothing, I.3.5 [Computer Graphics]: Geometric algorithms, languages, and systems, I.3.7 [Computer Graphics]: Animation I.4.8 [Computer Graphics]: Shape Analysis

1. Introduction

Animated deformations of complex models often result in visually unappealing artifacts, such as undesirable dents and wrinkles. Typically these problems arise when meshes are deformed into various extreme poses. For example, posing a character model with its chin up requires extra geometry around the neck region to support necessary deformations. However, when an animator poses the character with its chin down, the extra geometry produces a dent around the neck (Figure 1.a). Unfortunately, simple mesh relaxation does not fix the problem, but instead widens the dent, smoothes the slopes, and creates visually unacceptable results (Figure 1.b). Character modelers also desire a method to flatten dents and wrinkles. This becomes especially important during model prototyping when a modeler might need a simple, agile tool to try a new look on a character by removing dents or wrinkles. Moreover, dent and wrinkle removal are relevant not only to the game and feature animation industry but also to any applications that use 3d scanned data.

Motivated by adding a "dent-removal mechanism" to our artists' tool-set, we have developed a robust, surface-parameterization independent algorithm that automatically detects and removes dents without imposing significant penalties on runtime performance. Our algorithm is based on determining special geodesic curves, representative of the dent's orientation, and using these curves as reference for directional pulling iterations. Implementation of the presented algorithm proved to be straightforward and produced impressive results on real production data.

Mathematically, we define dents and wrinkles as regions that meet two requirements – first, regions contain one or more parallel soft feature curves; second, the regions exhibit high curvature across these curves. The sign of this high curvature differentiates dents from wrinkles. While our algorithm is successfully applicable to removal of both wrinkles and dents, for the purpose of example, the rest of the paper illustrates the algorithm on the case of dented geometry. The paper is structured as follows: In Section 2, we give a brief overview of related work. Then, in Section 3, we describe the main concepts behind our algorithm. In Section 4, we give a detailed description of geodesic curve selection. Finally, we provide our results in Section 5.

2. Related Work

Most fairing methods are based on Laplacian smoothing. For mesh vertex p_i, Laplacian smoothing is given by

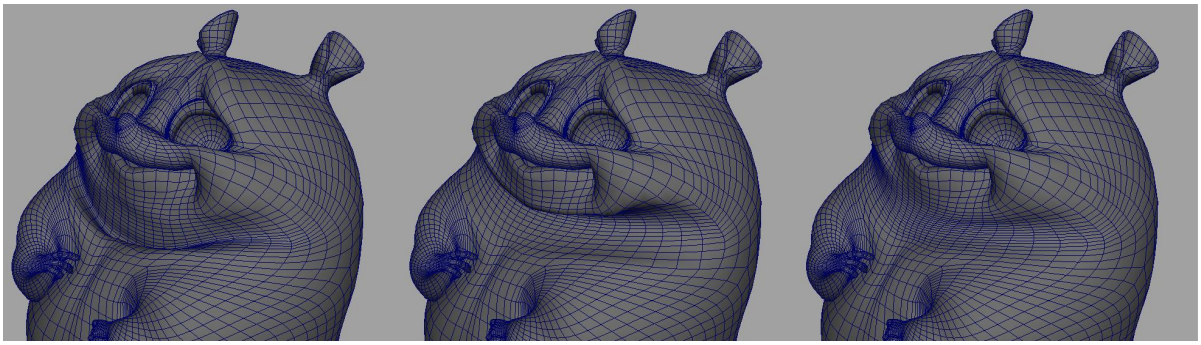
$$\frac{\partial p_i}{\partial t} = \nabla^2 p_i \tag{1}$$

where $\nabla^2 p_i$ is the Laplacian at p_i. We can approximate (1) by rewriting it using finite differences

$$p_i^{t+1} = (1-e)p_i^t + e\nabla^2 p_i^t \tag{2}$$

where e is a user defined envelope, and t is the iteration number. If we define the umbrella of p_i as the set of its edge-connected vertices, the Laplacian in (2) can be approximated using the sum of an umbrella-based function f

© The Eurographics Association 2009.



(a) *(b)* *(c)*

Figure 1: *Dented chin region on hamster model: (a) shows original shape, (b) demonstrates result of traditional (Laplacian) relaxation, and (c) is result of our relaxation scheme.*

$$p_i^{t+1} = (1-e)p_i^t + \frac{e}{\sum_j f(i,j)} \sum_j f(i,j)(p_j - p_i) \qquad (3)$$

where j is umbrella index of vertex p_j. Often $f(i,j)$ is simply the distance function between p_i and p_j. Repeatedly applying formula (3) to each vertex of the region, one can efficiently smooth the desired area of the mesh.

Unfortunately, Laplacian smoothing results in surface shrinkage. One of the first major solutions for this problem was offered by Taubin [Tau95]. He used the second order Laplacian to compensate for the shrinkage. However, the results of this relaxation were highly dependent on the choice of the coefficients for the first and second order Laplacian operators. Desbrun et al. attempted to solve the shrinkage problem by making umbrella coefficients sensitive to the mean curvature flow [DMS*99]. For each smoothing iteration, the mean curvature was calculated locally at each vertex umbrella. Pauly et al. estimated the shrinkage associated with each umbrella, and made adjustments to compensate for this artifact [PKG02]. To preserve features, Hildebrandt et al. estimated curvature around each vertex and then faired the surface by smoothing the vertex curvatures [HP04]. Mesh fairing with volume and feature preservation was further developed in [MT02, TWB*02, ZX06].

In the prior smoothing approaches, positioning of any particular vertex relies on various local geometric properties, emphasizing the vertex's local geometry (for example, vertex's mean curvature [DMS*99, HP04]). However, the local geometry of any particular vertex alone is not descriptive enough to iteratively reposition the vertex for adequate dent removal, and therefore, our algorithm adjusts the vertices based on their global positions within the dent to solve this problem.

3. Algorithm Overview

Our objective is to remove the dent by pulling the dented area upward while maintaining well-structured quads. To do this, we need to provide an appropriate function $f(i,j)$ for iterations, defined by (3). This function should be completely independent from the surface parameterization and should produce larger weights for umbrella vertices that contribute the most to pulling throughout the whole set of iterations. To achieve interactive rate, we restrict $f(i,j)$ to functions that need to be evaluated once per dented region, not every fairing iteration.

However, we cannot simply estimate curvature in each direction of the umbrella vertex and assign heavier weights to vertices in the direction of the higher curvature (e.g. estimating the tangential component of the Laplacian of the umbrella). This type of local scheme does not account for the dent's global orientation. For regions where the whole dent is bent, like the region around the character's neck, a vertex umbrella may initially exhibit maximum curvature parallel to the dent's direction. After the first few iterations, this local smoothing scheme would flatten the bent slope.

Intuitively, we want $f(i,j)$ to produce the weights reflecting the alignment of the vector, defined by the pulled vertex p_i and its umbrella vertex p_j, with the direction of the surface curve that runs across the dent. We find this direction as the tangential direction at p_i of a geodesic curve that has the maximum global curvature (i.e. total curvature) over all geodesic curves through p_j. This maximum global curvature is expressed as follows:

$$\max_{p_i \in \gamma} \mathrm{K}(\gamma) = \max_{p_i \in \gamma} \int_a^b \kappa(t)dt \qquad (4)$$

where γ is a geodesic curve that lies on the dented region and runs across p_i, $[a,b]$ is the parameter range of γ, K is total curvature of γ, and κ is a curvature at a particular point of γ. By maximizing the total curvature of γ, we ensure that the curve spans the dent's valley across the dent. Figure 2 shows an example of such a geodesic curve. Then $f(i,j)$

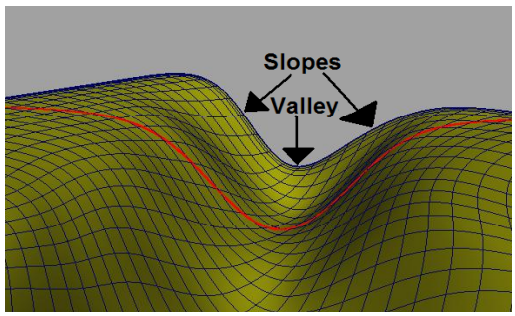


Figure 2: *The red surface curve is an example of a geodesic curve that spans the dent*

© The Eurographics Association 2009.

trivially becomes the dot product of vector p_i-p_j and the tangent direction of the geodesic curves. Given a dented region on a polygonal mesh, the main issue is then how to efficiently estimate the spanning geodesic curves for each vertex within a dented region.

4. Calculation of Spanning Geodesic Curves

Establishing the dent-spanning geodesic curves starts with locating vertices on the dent's valley. The direction of maximum curvature at a vertex on the valley describes the spanning geodesic curve that passes through this vertex. Once we have found the spanning geodesic curves in the valley, we propagate their direction to the vertices that lay on the slope. Let's consider the steps for calculating the spanning geodesic curves on the dent region more closely.

To distinguish vertices on the valley from the rest, we observe that their curvatures are much higher. Thus, performing even fast, approximate curvature estimation (for example, based on projections of the vertex normals onto the surrounding faces), we are able to efficiently locate the valley vertices. After that, we perform a more precise curvature analysis, limiting it only to the valley vertices.

There are a wide variety of techniques used to evaluate curvature on a polygonal mesh. Most of them fall into two groups – estimation based on fitting patches [e.g. CP05] and estimation based on normal curvature evaluation [e.g. Tau95*]. We found fitting patches to be unstable when the valley region contains degenerate quads or have even slightly creased slopes. On the other hand, normal curvature-based evaluation proves to be both efficient and reliable. In our algorithm, we have employed an approach presented by Taubin [Tau95*]. He specifies matrix $\boldsymbol{M}_{pi}$, defined as

$$M_{pi} = \frac{1}{2\pi}\int_{-\pi}^{\pi} \kappa_{pi}(T_\theta) T_\theta T_\theta^t d\theta \tag{5}$$

where T_θ is a unit vector, oriented by angle θ in the umbrella of the vertex p_i, and $\kappa_{pi}(T_\theta)$ is directional curvature at p_i in T_θ direction. The eigenvectors of $\boldsymbol{M}_{pi}$ are principal directions and its eigenvalues m_1 and m_2 can be used to calculate the principal curvatures k_1 and k_2:

$$\begin{aligned} k_1 &= 3m_1 - m_2 \\ k_2 &= 3m_2 - m_1 \end{aligned} \tag{6}$$

Taubin shows that one can approximate (5) by

$$M_{pi} = \sum_j w_{i,j} k_{i,j} T_{i,j} T_{i,j}^t \tag{7}$$

where $T_{i,j}$ is the projection of the edge (between p_i and its umbrella vertex p_j) onto a plane, tangent at p_i and $w_{i,j}$ is a user defined weight, proportional to the areas of faces that share p_i - p_j edge, $k_{i,j}$ is curvature in the direction of $T_{i,j}$ and can be calculated as

$$k_{i,j} = \frac{2N(p_i - p_j)}{\| p_i - p_j \|^2} \tag{8}$$

where N is surface normal at p_i. Thus, we can apply (7) to calculate $\boldsymbol{M}_{pi}$ for each valley vertex and then, through eigen-analysis, obtain the principal curvature information.

Once we have established the dent's orientation for the valley vertices, we propagate this information further to the slope vertices. This is accomplished in two steps.

First, we assign some initial directions of geodesic curves to each valley vertex, using an advancing-front technique. Let's define the vertex ring $ring_i$ as follows – $ring_0$ is a set of the valley vertices (seeds), $ring_{i+1}$ is a set of vertices from umbrellas of $ring_i$'s vertices. Note that the same vertex can belong to more than one ring. Then the front propagates from the seeds toward the dent's boundaries as represented in the pseudo code below

```
for each ring_i where i :=0...MAX
    for each vertex v_j ∈ ringi
        for each vertex v_k ∈ umbrella of v_j
            if v_k is not processed
                assigned the direction of v_k based
                on the direction of v_j
```

Second, to avoid noise in the assignment of geodesic curve directions, we run a Laplacian-like relaxation on them, locking values at the valley vertices, as we consider these values to be correct.

Temporal coherence is preserved since our algorithm is not based on discrete topology, but rather, on underlying geometry that typically deforms smoothly when animated over time. Therefore, as we advance to the next animation frame, we may select different seeds, but we should not see dramatic changes in shape of the spanning geodesic curves, which are the key elements responsible for creating the final result.

Finally, to smoothly blend the pulled area into the rest of the mesh, we perform several Laplacian smoothing iterations on vertices both inside and around the dent.

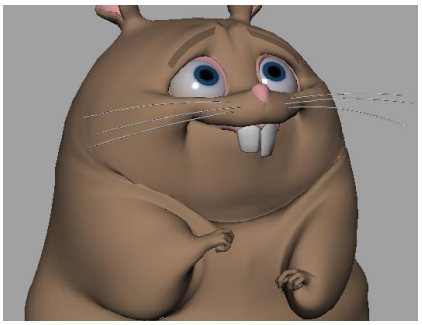


Figure 3: *Original Model*

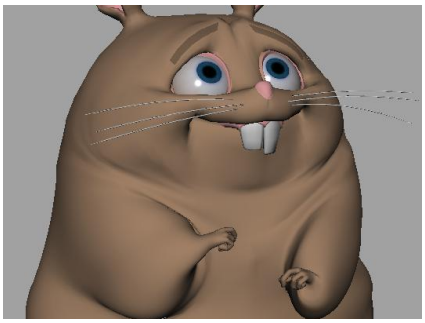


Figure 4: *Model after applying the presented method to smooth the chin and neck area.*

© The Eurographics Association 2009.

5. Results

In our implementation, we incorporated paintable weight maps (assigned envelope values per vertex) into the method. Users can designate areas where they want to partially preserve portions of the dent and/or differentiate between deformable and non-deformable parts of the mesh by painting low or zero envelope values in those areas (e.g. in Figure 4 and 6 characters heads are painted with zeros).

The presented algorithm has been deployed in a production environment and has proven to be a successful solution in a wide variety of cases. A significant practical value for animators comes from the fact that they only need to setup the dent pulling mechanism once. Then, our algorithm handles the transient migration of the dents over the course of animation by utilizing user specified areas where dents may occur. The dent migration and reorientation over time is reflected by selecting a different set of valley vertices.

Since the number of expensive numerical computations is minimized, our method yields very efficient performance. Incorporating the tool into light animation scenes causes frame rate to drop by less than 6% while adding the tool into scenes with heavy animation does not have noticeable impact on the frame rate at all.

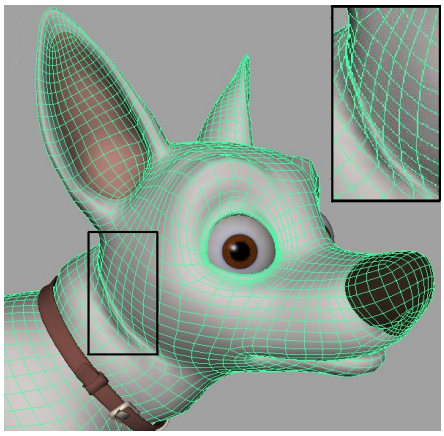


Figure 5: *Original model.*

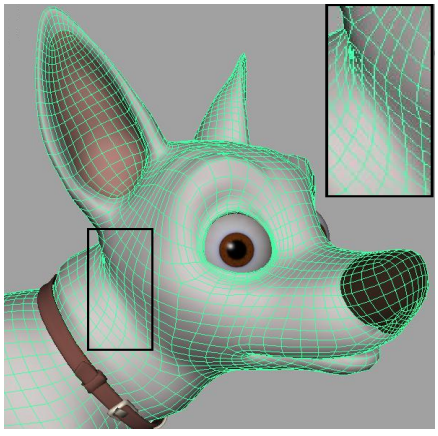


Figure 6: *Model after applying the presented method.*

6. Conclusion

We have presented a novel approach to effectively remove unwanted dent and wrinkle artifacts. Analyzing global behavior of the problematic area, our algorithm finds the most representative geodesic curves for the region. These geodesic curves guide the assignment of umbrella weights to perform the efficient pulling iterative process. Our algorithm is straightforward to implement and our results demonstrate its practical value in a production environment.

References

[CP05] F. CAZALS, M. POUGET: Estimating Differential Quantities Using Polynomial Fitting of Osculating Jets. *Computer Aided Design*, v. 22 (2005), pp. 121 –146

[DMS*99] M. DESBRUN, M.MEYER, P.SRODER, A.H. BARR: Implicit fairing of irregular meshes using diffusion and curvature flow. In Proc. *ACM SIGGRAPH '99* (1999), pp. 317–324.

[HP04] K. Hildebrandt, K. Polthier: Anisotropic Filtering of Non-Linear Surface Features. *Computer Graphics Forum*, v. 23, n. 3, (2004), pp. 391-400.

[MT02] J.M. MORVAN, B. THIBERT: Smooth surface and triangular mesh: comparison of the area, the normals and the unfolding. In Proc. *Seventh ACM Symposium on Solid Modeling and Applications* (New York, NY, 2002), ACM Press, pp. 147 –158

[PKG02] M. PAULY, L. KOBBELT, M. GROSS: Multiresolution modeling of point-sampled geometry. CS *Technical Report #378*, ETH Zurich (2002), http://graphics.stanford.edu/~mapauly/Pdfs/MultiresModeling.pdf

[Tau95] G. TAUBIN: A signal processing approach to fair surface design. In Proc. R. Cook (Ed.), Annual Conference Series, *ACM SIGGRAPH '95* (1995), pp. 351–358.

[Tau95*] G. TAUBIN: Estimating the tensor of curvature of a surface from a polyhedral approximation. In Proc. Fifth *International Conference on Computer Vision* (1995), pp, 902-907

[TWB*02] T. TASDIZEN, R. WHITAKER, P. BURCHARD, S. OSHER: Geometric surface smoothing via anisotropic diffusion of normals, In Proc. *IEEE Visualization '02* (2002), pp. 125-132

[ZX06] H. ZHAO, G. XU: Triangular surface mesh fairing via Gaussian curvature flow. *Journal of Computational and Applied Mathematics*, v. 195, n. 1, (2006), pp. 300-311. Special issue: The international symposium on computing and information

© The Eurographics Association 2009.