Geometry Processing Lab 2012
Anisotropic Filtering of Non-Linear Surface Features

Phan-Anh Nguyen
286049

August 4, 2012

1 Introduction

Nowadays, geometric data acquired through imaging or scanning devices has grown rapidly due to advances in technology, making it affordable in many aspects of our lives. However, when dealing with real data we always have to cope with measurement error which brings high frequency noise to our geometric models. Many researches have been conducted in order to remove noise from a scanned model while trying to preserve the underlying sampled surface. One of the seminal results was the work of Taubin et al. [Tau95] in which they use a signal processing approach to derive the Laplacian operator acting as a low-pass filter on the geometric signal. Even though the Laplacian operator is a powerful tool to remove high frequency noise, it isotropic behaviour makes it unable to preserve sharp features. Hildebrandt and Polthier [HP04] have developed an anisotropic method which can preserve high curvature features in a certain direction while suppressing unwanted curvature peaks in the other directions. This method makes it possible to denoise arbitrary surface meshes whereas non-linear geometric features e.g. curved surface regions and feature lines are preserved. This lab report explores and elaborates theory and practice needed to implement the prescribed mean curvature flow proposed by Hildebrandt and Polthier [HP04].

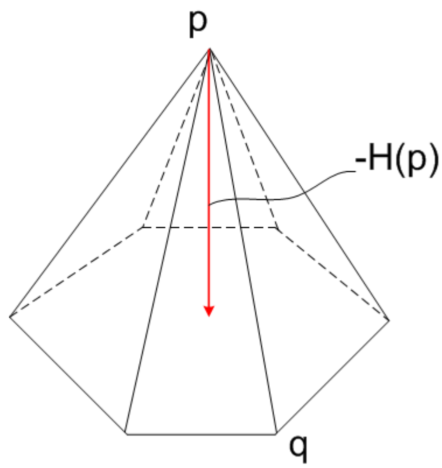

Figure 1: smooth a vertex p by moving it in the direction of the mean curvature vector $\vec{H}(p)$

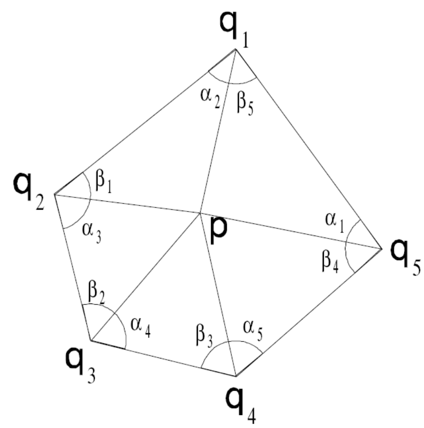

Figure 2: Cotangent weights

2 Smoothing Principle

A very intuitive smoothing operator one can think of is to move a vertex p to the center of gravity (c.o.g) of its one-ring neighbors $N_1(p)$:

$$p \leftarrow \frac{1}{\mid N_1(p) \mid} \sum_{q \in N_1(p)} q = p - \underbrace{\frac{1}{\mid N_1(p) \mid} \sum_{q \in N_1(p)} (p-q)}_{\triangle p} \tag{1}$$

Equation 1 reveals the update form of the smoothing operator in which the old vertex is moved by an amount of the update vector $\triangle p$ to the new position. The update vector $\triangle p$ can be generalized to have arbitrary weights over the 1-ring $\sum_{q \in N_1(p)} w_q(p-q)$ other than uniform weights as in the equation 1. In fact, we can choose the weights such that the update vector points in the direction normal to the mesh surface. Such an update formula was first proposed by Pinkall and Polthier [PP93] known as cotangent weights:

$$\bigtriangledown_p \; area = \vec{H}(p) = 1/2 \sum_{q \in N_1(p)} (cot\alpha_q + cot\beta_q)(p-q) \tag{2}$$

Where $\vec{H}(p)$ is the mean curvature vector at a vertex p and equal the gradient of the area functional $\bigtriangledown_p$ *area* at that vertex. Figure 1 and 2 show how the mean curvature vector is calculated.

3 Anisotropic Mean Curvature

The first step towards deriving an anisotropic mean curvature formula is to express the vertex mean curvature vector 2 in terms of an edge based mean curvature vector:

$$\vec{H}(e) = H_e \vec{N}_e \tag{3}$$

where $N_e = \frac{N_1+N_2}{\|N_1+N_2\|}$ is the edge normal vector as shown in figure 3 and $H_e = 2 \mid e \mid \cos\frac{\theta_e}{2}$ is the edge mean curvature which depends on the dihedral angle θ_e as illustrated in figure 4. Note that the smaller the dihedral angle, the sharper the edge thus resulting in the higher the mean curvature H_e. Therefore, the term H_e can be seen as the measurement of the directional curvature of the surface in the direction orthogonal to the edge.

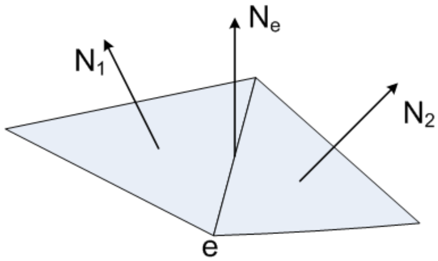

Figure 3: Edge normal vector N_e

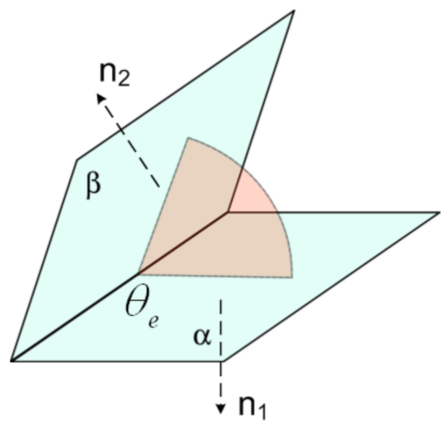

Figure 4: Dihedral angle θ_e

It can be shown, in the work of Polthier [Pol02], that the vertex mean curvature vector $\vec{H}(p)$ and the edge mean curvature vector $\vec{H}(e)$ are related by the equation:

$$\vec{H}(p) = \frac{1}{2} \sum_{e=(p,q), q \in N_1(p)} \vec{H}(e) \tag{4}$$

The anisotropic mean curvature vector $\vec{H}_A$ at a vertex p is then defined as a weighted sum over the contributions $H_e \vec{N}_e$ at the edges incident to a vertex p:

$$\vec{H}_A(p) = 1/2 \sum_{e=(p,q),q\in N_1(p)} w(H_e)H_e\vec{N}(e) \tag{5}$$

The weight function w is used to put less weight on feature vertices in order to avoid smoothing sharp features:

$$w_{\lambda,r}(a) = \begin{cases} 1 & \text{for } |a| \le \lambda \\ \dfrac{\lambda^2}{r(\lambda - |a|)^2 + \lambda^2} & \text{for } |a| > \lambda \end{cases} \tag{6}$$

The threshold λ is used to detect features and the radius r controls the width of the transition between those areas that are smoothed and those that are kept as features. In our implementation, we choose $\lambda = 2\lambda' \max|e|$ where $\lambda' \in [0,1]$ to cover the whole range of the edge mean curvature H_e. Following Hildebrandt and Polthier suggestion [HP04], we fix the radius r to 10 to ensure that $w_{\lambda,10}(2\lambda) < 0.1$.

The smoothing operation is carried out by integrating the flow of the anisotropic mean curvature vector $\vec{H}_A$ with some integration scheme. We first use the explicit Euler method because it is simpler to implement. Discussion about the semi-implicit integration scheme will be presented later on.

Given the mesh M_h with its vertices $\mathcal{P} = \{p_1, ..., p_m\}$, an explicit iteration step of the anisotropic mean curvature flow is computed as follows:

$$\mathcal{P}^{j+1} = \mathcal{P}^j - sM^{-1}\vec{H}_A(\mathcal{P}^j) \tag{7}$$

where $s \in [0,1]$ is a damping factor controlling the magnitude of the update vector thereby stabilizing the flow. M^{-1} is the inverse of the mass matrix $M \in \mathcal{R}^{m\times m}$ of the mesh M_h^j at time step j. It is used to convert the integrated mean curvature vector into a piecewise linear vector field:

$$M_{pq} = \begin{cases} \dfrac{1}{6} area(star\ p), & \text{if } p = q \\ \dfrac{1}{12} area(star\ e), & \text{if there is an edge } e = (p,q) \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

If we use a diagonalization of M called the lumped mass matrix with diagonal elements $M_{pp} = \frac{1}{3}area(star\ p)$, then the integration step for each vertex p is derived as follows:

$$p^{j+1} = p^j - \frac{3s}{area(star\ p^j)} \vec{H}_A(p^j) \tag{9}$$

One disadvantage of the lumped mass matrix is that it gives an unstable solution. To overcome this problem we have to use a very small time step leading to a slow convergence rate. In contrast, the full mass matrix gives a more stable solution, thereby giving a faster convergence rate with the cost of having to compute the inverted mass matrix.

4 Prescribed Mean Curvature

Since the mean curvature vector at a vertex is equivalent to its area gradient, applying the mean curvature flow on the mesh has the same effect as minimizing its surface area. This leads to the problem of surface shrinkage. This issue also occurs in the anisotropic case. Particularly, the anisotropic smoothing slows down the smoothing process in regions with high curvature, causing deformations of the surface as shown in figure 5.

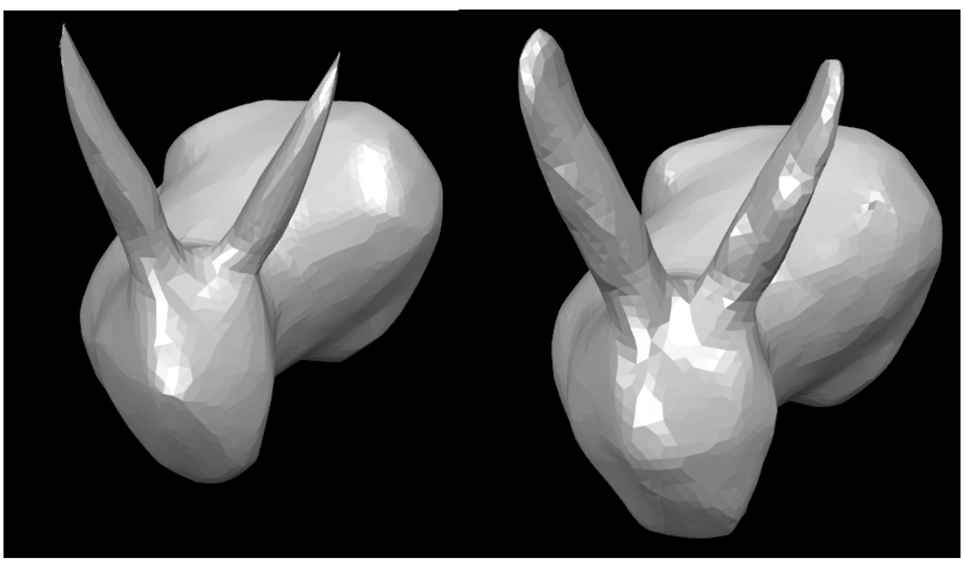

Figure 5: The Anisotropic Mean Curvature flow (left) contracts the interior of the bunny's ears making it to deform. In contrast, the Prescribed Mean Curvature flow (right) converges to a stable surface.

To circumvent this problem, Hildebrandt and Polthier [HP04] used the so-called prescribed mean curvature flow (PMC) to evolve the surface towards a surface having a prescribed mean curvature. This process includes two steps. In the first step, the surface mean curvature is computed and then this scalar field is smoothed. In the second step, the PMC flow is applied to let the surface evolve towards a surface with this precomputed mean curvature.

This idea comes from the result [PK02] given as follows:

$$\vec{H_A}(p) = \bigtriangledown_p area = H \bigtriangledown_p vol \tag{10}$$

for all interior vertices p and a constant mean curvature H. Hence instead of minimizing the anisotropic mean curvature vector $\vec{H_A}(p) \to 0$, we let $\vec{H_A}(p)$ evolve to the prescribed mean curvature according to equation 10, i.e. $\vec{H_A}(p) \to H\vec{V_A}(p)$. The anisotropic PMC flow thus is defined by:

$$\mathcal{P}^{j+1} = \mathcal{P}^j - sM^{-1}(\vec{H_A}(\mathcal{P}^j) - f(\mathcal{P}) \cdot \vec{V}_A(\mathcal{P})) \tag{11}$$

where f is a function that prescribes the anisotropic mean curvature and $\vec{V}_A$ is an anisotropic volume gradient. The volume of a surface is the oriented volume enclosed by the cone of the surface over the origin in $\mathcal{R}^3$:

$$vol\ M_h = \sum_{T=(p,q,r)\in M_h} vol(tetrahedron(o,p,q,r)) = \frac{1}{6}\sum_{T=(p,q,r)\in M_h} < p, q \times r > \tag{12}$$

Figure 6 illustrates how the volume of a surface is calculated. Note that the volume of tetrahedrons is negative in front-facing regions (where the normal pointing to the origin). This negative volume cancels out with the volume of back-facing tetrahedrons hence only the volume enclosed by the surface is computed.

Since $vol\ M_h$ linearly depends on p, the gradient of $vol\ M_h$ at p is easily derived from 12 as follows:

$$\bigtriangledown_p vol = \frac{1}{6}\sum_{T=(p,q,r)\in M_h} q \times r \tag{13}$$

Having defined the volume gradient we can now derive its anisotropic counterpart $\vec{V_A}$ as introduced in equation 11. For non-feature vertices p,

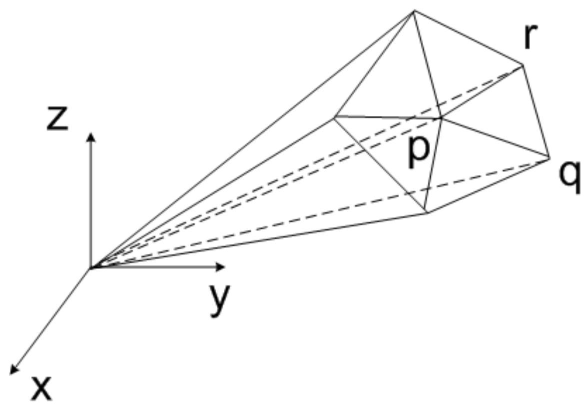

Figure 6: volume of a surface is the sum of volume of oriented tetrahedrons spanned by surface triangles with the origin.

i.e. $\vec{H}_A(p) = \vec{H}(p)$, $\vec{V}_A(p)$ is set to $\bigtriangledown_p(vol)$. For the other vertices $\vec{V}_A(p)$ is defined by:

$$\vec{V}_A(p) = sign\left(\langle \vec{e}_{H_A}(p), \bigtriangledown_p vol\rangle\right) \vec{e}_{H_A}(p) \tag{14}$$

where $\vec{e}_{H_A}(p)$ is the unit vector field of $\vec{H}^s_A$ and we get $\vec{H}^s_A$ by performing a simple smoothing step on $\vec{H}_A$:

$$\vec{H}^s_A(p) = \frac{1}{2}\left(\vec{H}_A(p) + \frac{1}{\sum\limits_{q\in N_1(p)} \omega_q} \sum_{q\in N_1(p)} \omega_q \vec{H}_A(q)\right) \tag{15}$$

$$\vec{e}_{H_A}(p) = \frac{\vec{H}^s_A(p)}{\parallel \vec{H}^s_A(p) \parallel} \tag{16}$$

where ω_q is the sum of the vertex angles at p in the triangles adjacent to the edge $\bar{pq}$ as displayed in figure .

5 Implementation and Results

use OpenFlipper, Eigen library, Cholmod suite sparse to invert mass matrix

6 Implicit Integration of the Flow

Matrix form: first attempt Ha, second attempt Taylor approximation.

7 Conclusion and Future Work

summary the main idea, pose problem to be solved in the future

References

[HP04] Klaus Hildebrandt and Konrad Polthier. Anisotropic filtering of non-linear surface features. *Computer Graphics Forum*, 23:391–400, 2004.

[PK02] ROSSMANN W. POLTHIER K. Index of discrete constant mean curvature surfaces. *J. Reine und Angew. Math*, pages 47–77, 2002.

[Pol02] Konrad Polthier. Polyhedral surfaces of constant mean curvature. Technical report, TU Berlin, 2002.

[PP93] Ulrich Pinkall and Konrad Polthier. Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics*, 2:15–36, 1993.

[Tau95] 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.