Computing Length-Preserved Free Boundary for Quasi-Developable Mesh Segmentation

Charlie C.L. Wang

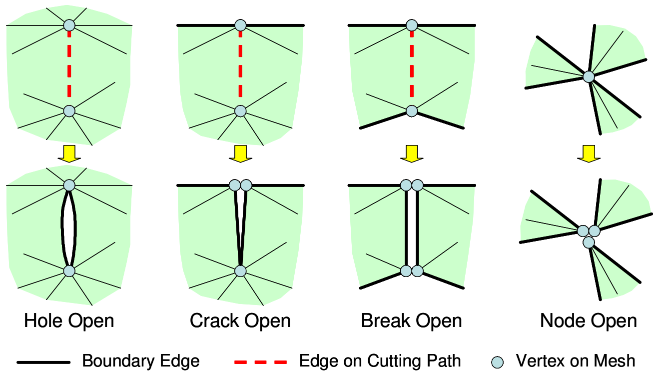

Fig. 1. A mesh surface can be cut along a path by iteratively applying four operators: *Hole Open*, *Crack Open*, *Break Open*, and *Node Open*.

1. Operators for mesh cutting

In our implementation, we iteratively introduce three operators on the edges belonging to the cutting path (see Fig. 1), which are

- *Hole open* – for a given edge e on the cutting path, when neither of the two vertices are on the boundary, this operator is applied to construct a hole by converting e into a boundary edge and adding another boundary edge coincident to e. After this, both vertices have become boundary vertices.
- *Crack open* – when one vertex v_s on e is on the boundary, this operator is applied to create a crack along e by duplicating a vertex v_{new} to v_s and a new edge e_{new} for e, where e_{new} links to v_{new} and v_d – another vertex on e.
- *Break open* – when both v_s and v_d on e are on the boundary, we fully separate the left and the right portions of e by this *break open* operator (see Fig. 1).

After opening the edges on a cutting path into boundary edges, the following *Node open* operator is applied to the boundary vertices finally.

- *Node open* – For a boundary vertex v_s linked with n (n>2) boundary edges, ((n-2)/2) new vertices coincident to v_s are constructed, and the edges and faces linking to v_s are separated so that every vertex is linked with only two boundary edges (see Fig. 1).

2. Computation for matrices

B_λ, B_θ and Λ in Eq. (12) can be efficiently evaluated.

Statement 1 *B_λ is computed by*

$$B_\lambda = \left(-\frac{\partial J}{\partial \lambda_\theta}, -\frac{\partial J}{\partial \lambda_{0x}}, -\frac{\partial J}{\partial \lambda_{0y}}, \cdots, -\frac{\partial J}{\partial \lambda_{mx}}, -\frac{\partial J}{\partial \lambda_{my}}\right)$$

where

$$-\frac{\partial J}{\partial \lambda_\theta} = -(n-2)\pi + \sum_{k=1}^{n} \theta_k \,,$$

$$-\frac{\partial J}{\partial \lambda_{px}} = -\sum_{k=\alpha(p)}^{\beta(p)-1} l_k \cos\phi_k \,, \text{ and } -\frac{\partial J}{\partial \lambda_{py}} = -\sum_{k=\alpha(p)}^{\beta(p)-1} l_k \sin\phi_k \,.$$

Statement 2 $B_\theta = \{-\partial J / \partial \theta_i\}$ can be efficient evaluated by the following recursion formulas.

$$b_{\theta_1} = -(\theta_1 - a_1) + \lambda_\theta + \sum_{p=0}^{m} \sum_{k=\alpha(p)}^{\beta(p)-1} (-\lambda_{px} \sin\phi_k + \lambda_{py} \cos\phi_k) l_k \,,$$

$$b_{\theta_{i+1}} = b_{\theta_i} + (\theta_i - a_i) - (\theta_{i+1} - a_{i+1}) + \sum_{p=0}^{m} A(p,i) \,,$$

with

$$A(p,i) = \begin{cases} (\lambda_{px} \sin\phi_i - \lambda_{py} \cos\phi_i) l_i, & \alpha(p) \le i < \beta(p) \\ 0, & \textit{otherwise} \end{cases} .$$

Proof. From $B_\theta = \{b_{\theta_i}\} = \{-\partial J / \partial \theta_i\}$, we could have

$$b_{\theta_i} = -(\theta_i - a_i) + \lambda_\theta + \sum_{p=0}^{m} \sum_{k=\alpha(p)}^{\beta(p)-1} (-\lambda_{px} \sin\phi_k + \lambda_{py} \cos\phi_k) l_k \frac{\partial \phi_k}{\partial \theta_i} .$$

Letting

$$\Gamma(p,i) \equiv \sum_{k=\alpha(p)}^{\beta(p)-1} (-\lambda_{px} \sin\phi_k + \lambda_{py} \cos\phi_k) l_k \frac{\partial \phi_k}{\partial \theta_i}$$

together with

$$\frac{\partial \phi_k}{\partial \theta_i} = \begin{cases} 0, & k < i \\ -1, & k \ge i \end{cases},$$

we could have

$$\Gamma(p,i) \equiv \begin{cases} \sum_{k=\alpha(p)}^{\beta(p)-1} (-\lambda_{px} \sin\phi_k + \lambda_{py} \cos\phi_k) l_k, & i \le \alpha(p) < \beta(p) \\ \sum_{k=i}^{\beta(p)-1} (-\lambda_{px} \sin\phi_k + \lambda_{py} \cos\phi_k) l_k, & \alpha(p) < i < \beta(p) \\ 0, & \alpha(p) < \beta(p) \le i \end{cases}$$

This can be further simplified as follows.

Case 1: When $i \ge \beta(p)$, $i+1 > \beta(p)$, we have

$$\Gamma(p,i+1) = \Gamma(p,i) = 0 \,,$$

which leads to $A(p,i) = 0$.

Case 2: For $i = \beta(p) - 1$, $\Gamma(p,i+1) = 0$,

$$\Gamma(p,i) = (-\lambda_{px} \sin\phi_k + \lambda_{py} \cos\phi_k) l_i \,,$$

thus

$$A(p,i) = (\lambda_{px} \sin\phi_i - \lambda_{py} \cos\phi_i) l_i \,.$$

Case 3: When $\alpha(p) \le i < \beta(p) - 1$,

$$\Gamma(p,i+1) = \sum_{k=i+1}^{\beta(p)-1} (-\lambda_{px} \sin\phi_k + \lambda_{py} \cos\phi_k) l_k \,,$$

$$\Gamma(p,i) = \sum_{k=i}^{\beta(p)-1} (-\lambda_{px} \sin\phi_k + \lambda_{py} \cos\phi_k) l_k \,,$$

so we can conclude that

$$A(p,i)=\Gamma(p,i+1)-\Gamma(p,i)=(\lambda_{px}\sin\phi_i-\lambda_{py}\cos\phi_i)l_i\,.$$

Case 4: $i=\alpha(p)-1$, i.e., $i+1=\alpha(p)$, which leads to

$$\Gamma(p,i+1)=\Gamma(p,i)=\sum_{k=\alpha(p)}^{\beta(p)-1}(-\lambda_{px}\sin\phi_k+\lambda_{py}\cos\phi_k)l_k\,,$$

thus $A(p,i)=0$.

Case 5: $i<\alpha(p)-1$, i.e., $i+1<\alpha(p)$, for the same reason as above case, we have $A(p,i)=0$.

By concluding all these five cases, we could have $A(p,i)$ as given in Statement 2.

Q.E.D.

Statement 3 *For* Λ, *whose dimension is* $(2m+3)\times n$, *its i-th column vector is*

$$\Lambda_i=\frac{\partial^2 J}{\partial\lambda\partial\theta_i}=\begin{pmatrix}\frac{\partial}{\partial\theta_i}((n-2)\pi-\sum_{k=1}^{n}\theta_k)\\ \frac{\partial}{\partial\theta_i}\sum_{k=\alpha(p)}^{\beta(p)-1}l_k\cos\phi_k\\ \frac{\partial}{\partial\theta_i}\sum_{k=\alpha(p)}^{\beta(p)-1}l_k\sin\phi_k\\ \cdots\end{pmatrix}_{(2m+3)\times 1},$$

with $p=0,1,\cdots,m$. *Every element of* Λ_i *can be evaluated by*

$$\Lambda_{1,i}=-1\,,$$

$$\Lambda_{2p+2,i+1}=\Lambda_{2p+2,i}+B(p,i)\,,$$

$$\Lambda_{2p+3,i+1}=\Lambda_{2p+3,i}+D(p,i)\,,$$

with

$$B(p,i)=\begin{cases}-l_i\sin\phi_i, & a(p)\le i<\beta(p)\\ 0, & otherwise\end{cases},$$

$$D(p,i)=\begin{cases}l_i\cos\phi_i, & a(p)\le i<\beta(p)\\ 0, & otherwise\end{cases}.$$

Proof. By $\Lambda_i=\partial^2 J/\partial\lambda\partial\theta_i$, we could have

$$\Lambda_{1,i}=-1\,,$$

$$\Lambda_{2p+2,i+1}=-\sum_{k=\alpha(p)}^{\beta(p)-1}l_k\sin\phi_k\frac{\partial\phi_k}{\partial\theta_i}\,,$$

$$\Lambda_{2p+3,i+1}=\sum_{k=\alpha(p)}^{\beta(p)-1}l_k\cos\phi_k\frac{\partial\phi_k}{\partial\theta_i}\,.$$

The following proof could be ignored as it is similar to the proof of statement 2.

Q.E.D.