Morphing of closed spherical curves

A. Ikemakhen, **M. Bellaihou**, T. Ahanchaou

Team Geometry, Topology Algebra and Applications

Cadi-Ayyad University, faculty of sciences and technologies, Marrakesh, Morocco

Table of contents

1. Introduction

2. Morphing of Planar Curves

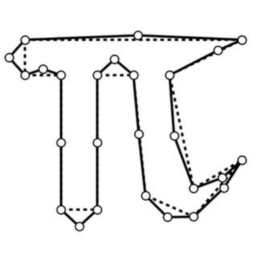
3. Curvature-based blending

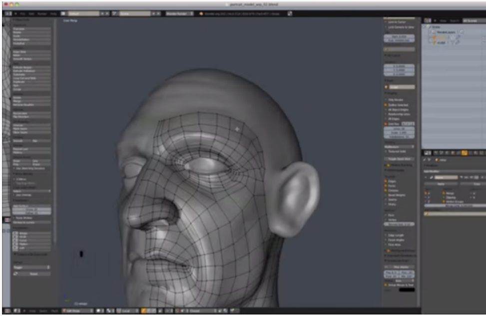
4. Morphing of spherical curves

Introduction

Overview

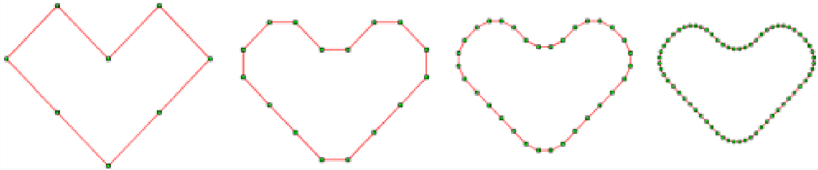
Geometry processing is an area of research that uses concepts from applied mathematics, computer science and engineering to design efficient algorithms for the reconstruction, analysis and manipulation of complex 3D models (Curves, Surfaces,...).





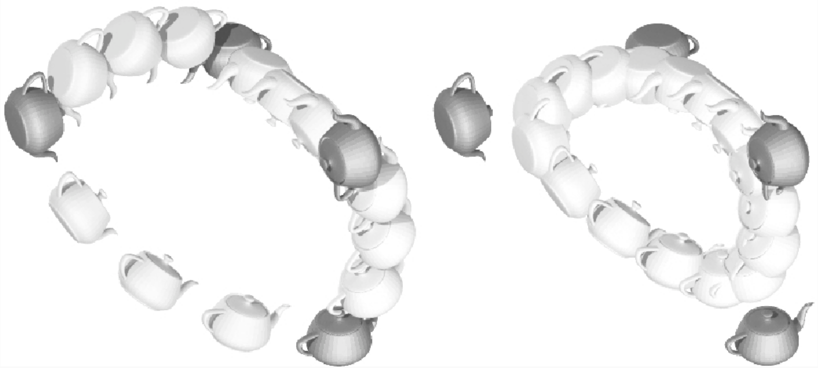
Subdivision

Subdivision is a repeated process applied to some discrete data in order to produce smooth objects.

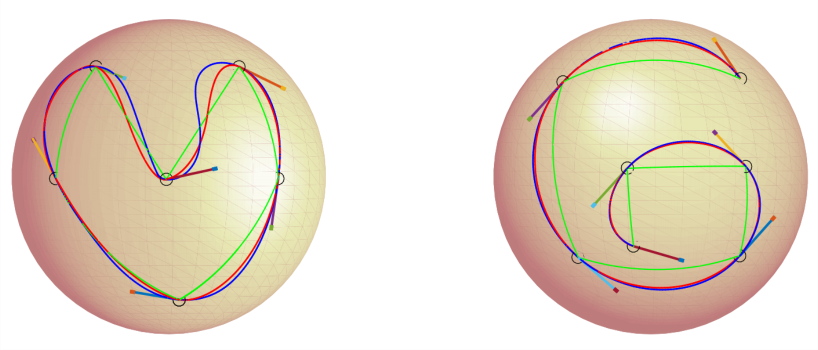


Subdivision on matrix group

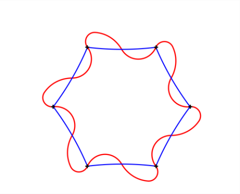
Black pot : Initial discrete data. White pot : Continuous motion passing through the fixed black pot.

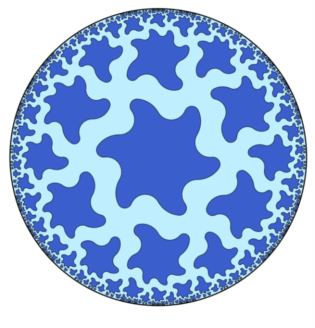


Subdivision on Spherical Geometry (Ikemakhen,Bellaihou)



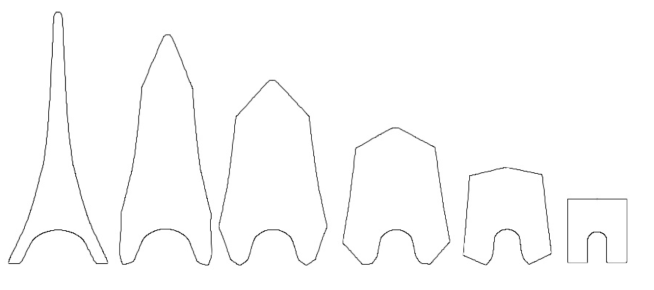
Subdivision on Hyperbolic Geometry (Ikemakhen, Ahanchaou)



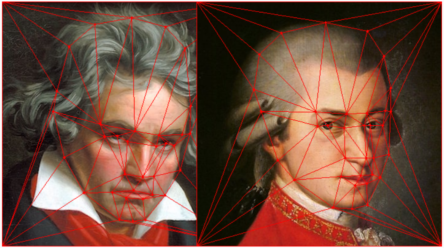


Morphing of shapes

Definition
Morphing (also called blending) algorithms determine intermediate shapes providing a continuous and gradual transition between two fixed shapes referred to as the key shapes



Morphing of Triangulations



Morphing of Surfaces

Morphing of Planar Curves

Morphing of plane curves

Given two parametric C^2-regular curves on the plane $\gamma^0 : [0, L_0] \to \mathbb{R}^2$ and $\gamma^1 : [0, L_1] \to \mathbb{R}^2$.
The **morphing** or **blending** between γ^0 and γ^1 is a continuous map $t \in [0, 1] \to f(\gamma^0, \gamma^1, t)$ such that :

- For $t = 0$ we get γ^0.
- For $t = 1$ we get γ^1.
- For each $t \in [0, 1]$, the curve $f(\gamma^0, \gamma^1, t) := \gamma^t : [0, L_t] \to \mathbb{R}^2$ is C^2-regular.

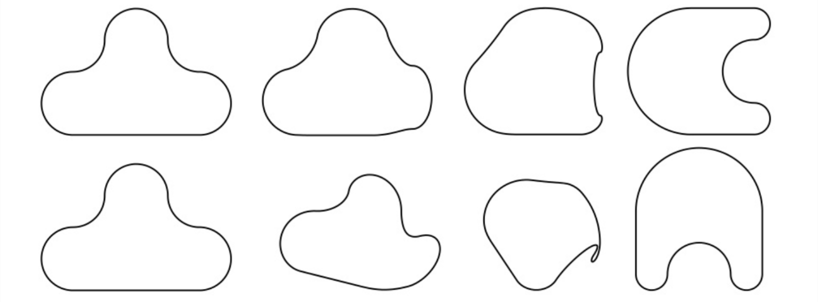
γ^0 is called the **source** curve and γ^1 the **target**.

Linear blending

We suppose that γ^0 and γ^1 are parametrized over $[0,1]$. The simplest blending is the **Linear blending** :

$$\gamma^t(s) = (1-t)\gamma^0(s) + t\gamma^1(s)$$

for $s, t \in [0,1]$.



Linear blending

- It depends on the particular parametrizations of γ^0 and γ^1.
- It can lead to **unnaturally looking** and **not visually pleasing** (undesirable) intermediate curves.
- Once the source and the target are closed, γ^t is closed.

Curvature-based blending

Curvature-based blending

Let $\kappa^0 : [0,1] \to \mathbb{R}$ and $\kappa^1 : [0,1] \to \mathbb{R}$ be the signed curvature functions of γ^0 and γ^1. The **curvature-based blending** is to define the intermediate curve γ^t such that its curvature function is given by :

$$\kappa^t(s) = (1-t)\kappa^0(s) + t\kappa^1(s), \quad s, t \in [0,1]\,.$$

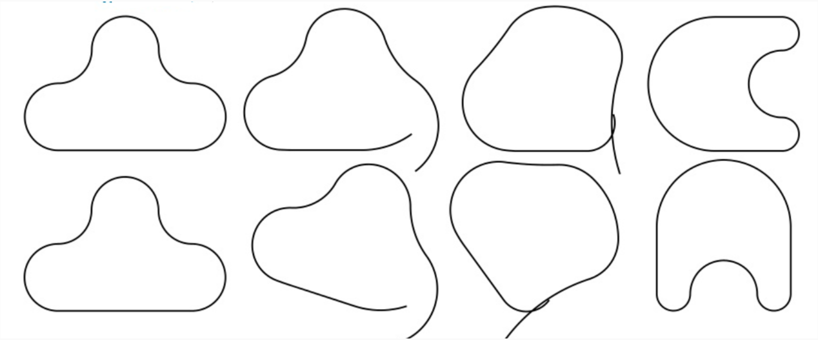
γ^t is defined from its signed curvature function by :

$$\gamma^t(s) = \int_0^s \begin{pmatrix} cos\,\theta(u) \\ sin\,\theta(u) \end{pmatrix} du + \begin{pmatrix} x_0 \\ y_0 \end{pmatrix},$$

where $\theta(s) = \displaystyle\int_0^s \kappa^t(u)du + \theta_0$.

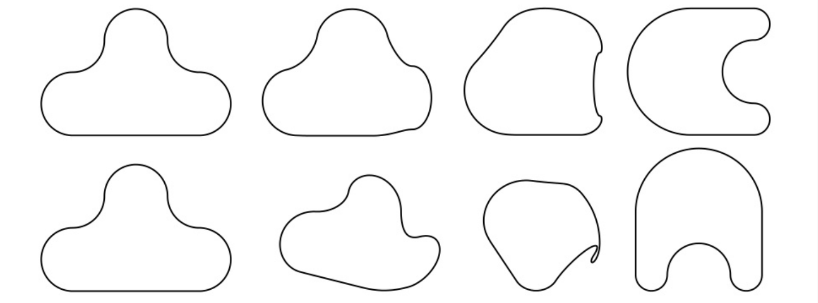
Curvature-based blending

$$\kappa^t(s) = (1-t)\kappa^0(s) + t\kappa^1(s), \quad s, t \in [0,1]\,.$$



Linear blending

$$\gamma^t(s) = (1-t)\gamma^0(s) + t\gamma^1(s)$$

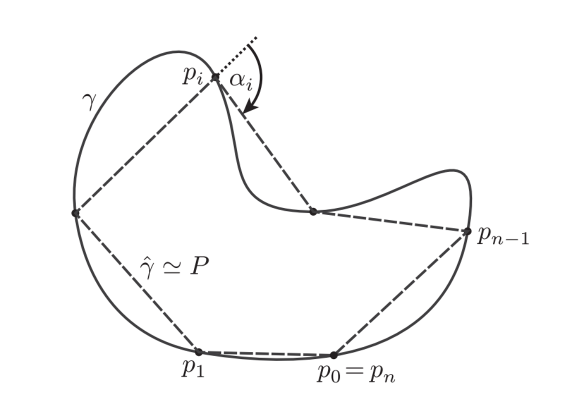


Curvature-based blending

- Gives intrinsic results (Do not depend on the parametrizations).
- Leads to open curves.

Discretization

In practice, we sample γ^0 (resp. γ^1) at $n+1$ uniformly distributed parameter values giving the polygon $P^0 = \{p_i^0\}_{0 \leqslant i \leqslant n}$ (resp. $P^1 = \{p1_i\}_{0 \leqslant i \leqslant n}$).



1. Interpolating the edge lengths by : $e_i^t = (1-t)e_i^0 + te_i^1$.
2. Compute signed discrete curvatures of each polygon P^0 and P^1, using :
$$\kappa_i = \frac{2\,\alpha_i}{e_{i-1} + e_i}. \tag{1}$$
3. Interpolating the discrete curvatures : $\kappa_i^t = (1-t)\kappa_i^0 + t\kappa_i^1$.
4. Recover the exterior angles α_i^t of the intermediate polygon P^t using Eq. (1).
5. After specifying the rigid motion i.e. : p_0^t and α_0^t, P^t is defined by :
$$p_i^t = p_{i-1}^t + e_{i-1}^t \begin{pmatrix} cos\,\theta_{i-1} \\ sin\,\theta_{i-1}, \end{pmatrix}$$
where $\theta_i = \sum_{k=0}^{i} \alpha_k$ is the angle from the x-axis to the edge $p_{i-1}^t p_i^t$.

Closing the interpolated polygon P^t

Remark

As pointed out above, if the source P^0 and the target P^1 are closed the intermediate polygon P^t is not closed.

Solution

Changing the angular defects α_i^t of P^t (Open) by :

$$\tilde{\alpha}_i^t = \alpha_i^t + \varepsilon_i$$

such that :

1. The resulting polygon $\tilde{P}^t$ is closed.
2. $\tilde{P}^t$ and P^t have the same edge lengths.
3. The curvature of $\tilde{P}^t$ is close to the curvature of P^t as possible.

Closing procedure statement

Minimize $\|\tilde{\kappa}^t - \kappa^t\| = \sum_{i=0}^{n} \left(\frac{2\,\varepsilon_i}{e_{i-1}^t + e_i^t} \right)^2$, subject to $\tilde{p}_0^t = \tilde{p}_n^t$. That is : Find

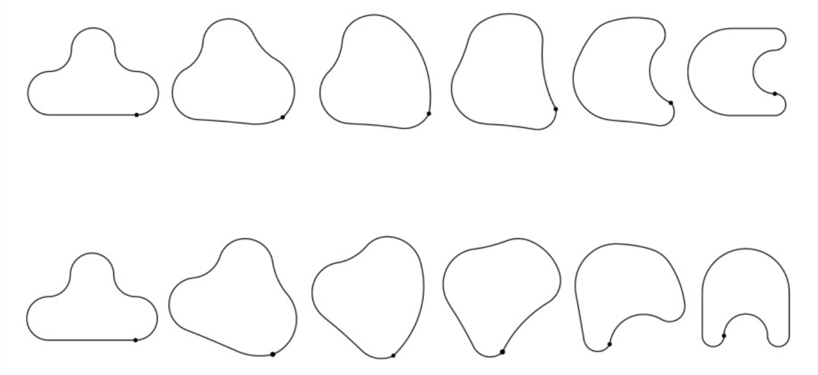
$$\min_{\varepsilon_i} \sum_{i=0}^{n} \left(\frac{2\,\varepsilon_i}{e_{i-1}^t + e_i^t} \right)^2$$

subject to the closure conditions :

$$\begin{cases} \sum_{i=0}^{n} (\tilde{x_{i+1}} - \tilde{x}_i) = \sum_{i=0}^{n} e_i^t \, sin\Big(\sum_{k=0}^{i} (\alpha_i^t + \varepsilon_i) \Big) = 0, \\ \sum_{i=0}^{n} (\tilde{y_{i+1}} - \tilde{y}_i) = \sum_{i=0}^{n} e_i^t \, cos\Big(\sum_{k=0}^{i} (\alpha_i^t + \varepsilon_i) \Big) = 0. \end{cases}$$

Curvature-based blending+Closure Process

$$\kappa^t(s) = (1-t)\kappa^0(s) + t\kappa^1(s), \quad s, t \in [0,1]$$



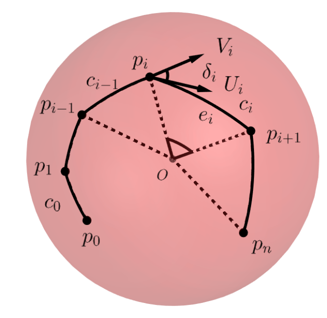
References

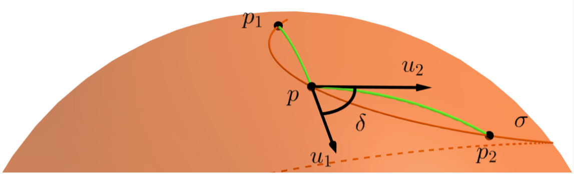
- Dym, Nadav and Shtengel, Anna and Lipman, Yaron, *Homotopic Morphing of Planar Curves*, 2015.
- Saba, Marianna et. al., *Curvature-based blending of closed planar curves*, 2014.
- Sederberg, Thomas W. et. al., *2-D shape blending : an intrinsic solution to the vertex path problem*, 1993.
- Surazhsky, Tatiana and Elber, Gershon, *Metamorphosis of planar parametric curves via curvature interpolation*, 2002.

Morphing of spherical curves

Let C^0 and C^1 be two spherical closed curves approximated by two geodesic polygons $P^0 = \{p_i^0\}_{0 \leqslant i \leqslant n}$ and $P^1 = \{p_i^1\}_{0 \leqslant i \leqslant n}$ respectively. The idea is to construct the morph P^t such that :

- Its arc length is given by : $L^t := (1-t)L^0 + tL^1$.
- Its geodesic curvature is given by : $\kappa_g^t := (1-t)\kappa_g^0 + t\kappa_g^1$.





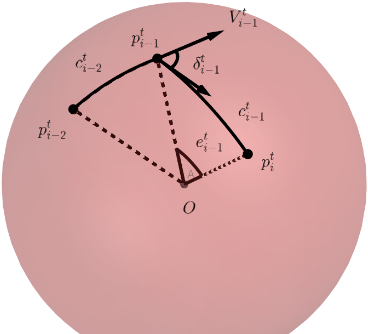
Where the geodesic curvature at p of an embedded geodesic polygon is :

$$\kappa_g(p) := \frac{2\ \delta}{e_1 + e_2},$$

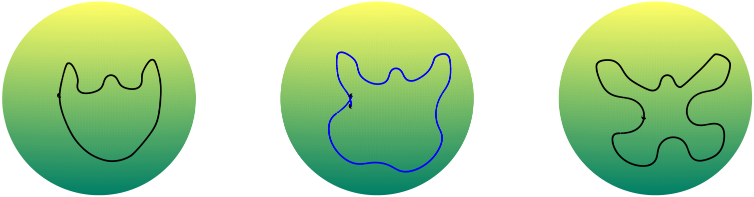
e_i is the the length of the geodesic connecting p and p_i.

The Algorithm

1. Interpolating the geodesic lengths by : $e_i^t = (1-t)e_i^0 + te_i^1$.
2. Interpolating the discrete geodesic curvatures : $\kappa_i^t = (1-t)\kappa_i^0 + t\kappa_i^1$.
3. Recover the exterior angles δ_i^t of the morph P^t.
4. We construct the geodesic c_{i-1}^t from the previously constructed c_{i-2}^t. Then using the **exponential** map and its **logarithm** to get the point p_i^t.



Result

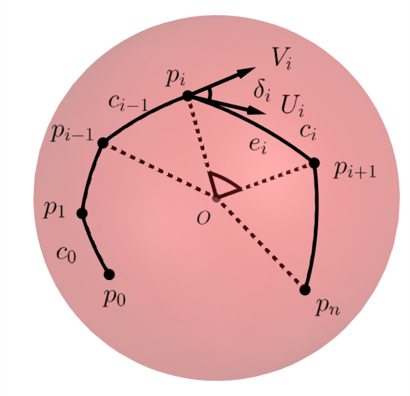


Remark
The intermediate curve (at time $t = \frac{1}{2}$) remains unclosed.

Closing process

We have seen that for a planar polygon of edge lengths e_i and exterior angles α_i, P is closed if and only if :

$$\begin{cases} \sum_{i=0}^{n} (x_{i+1} - x_i) = \sum_{i=0}^{n} e_i^t \, sin\Big(\sum_{k=0}^{i} (\alpha_i^t) \Big) = 0, \\ \sum_{i=0}^{n} (y_{i+1} - y_i) = \sum_{i=0}^{n} e_i^t \, cos\Big(\sum_{k=0}^{i} (\alpha_i^t) \Big) = 0. \end{cases}$$



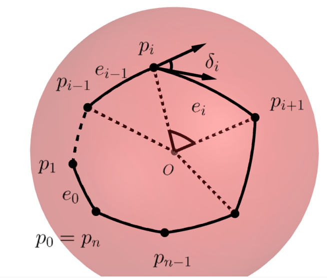
Question

What is the Closure conditions on the geodesic lengths and exterior angles of an open spherical polygon ?

Theoreme (Rotation equation of a spherical polygon)
Let $P = \{p_i\}_{0 \leqslant i \leqslant n}$ be a closed spherical polygon having e_i, $i \in \{0, .., n-1\}$ as side lengths and δ_i, $i \in \{0, .., n-1\}$ as exterior angles. Then

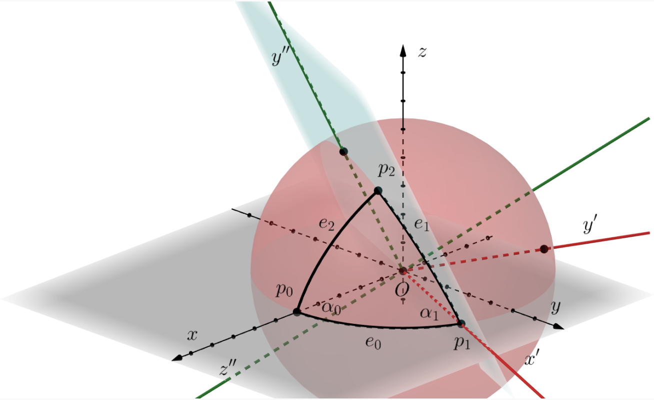
$$\prod_{i=0}^{n-1} R_z(\delta_i)\, R_x(e_i) = I. \tag{2}$$

Where R_x (resp. R_z) is the rotation matrix around x-axis (resp. z-axis).



The case of a spherical triangle

$$R_x(\delta_0)\, R_z(e_2)\, R_x(\delta_2)\, R_z(e_1)\, R_x(\delta_1)\, R_z(e_0) = I. \tag{3}$$



A spherical triangle can be **reconstructed** using only R_x and R_z rotations.

Closure condition of Open polygon

Since open spherical polygons do not have an exterior angle at p_0, we need to characterize the closure condition without involving δ_0. Let :

$$S := R_x(e_0)\, R_z(\delta_1)\, R_x(e_1) \ldots R_z(\delta_{n-1})\, R_x(e_{n-1}). \tag{4}$$

If $S = \left(S_{ij}\right)_{1 \leqslant i,j \leqslant 3}$, then the axis of rotation associated to S is given by :

$$u^S := \begin{pmatrix} S_{32} - S_{23} \\ S_{13} - S_{31} \\ S_{21} - S_{12} \end{pmatrix}.$$

Then P is closed if and only if there exists some $\delta_0 \in [0; \pi]$ such that :

$$S = R_z(-\delta_0),$$

which means $R_z(\delta_0)S = I$.

Proposition

Let $P = \{p_0, .., p_n\}$ be a spherical polygon (not necessarily closed) having e_i, $i \in \{0, .., n-1\}$ as side lengths and δ_i, $i \in \{1, .., n-1\}$ as exterior angles. Then $p_0 = p_n$ if and only if

$$\begin{cases} S_{32} - S_{23} = 0, \\ S_{13} - S_{31} = 0. \end{cases}$$

Quaternion formulation

If we use the quaternion formulation of rotations :

$$\begin{cases} Z_i := cos(\delta_i/2) + sin(\delta_i/2)\mathbf{k}, \\ X_i := cos(e_i/2) + sin(e_i/2)\mathbf{i}. \end{cases} \tag{5}$$

Then the closure conditions are :

$$\begin{cases} [\mathbf{i}, X_0\, Z_1\, X_1 ... Z_{n-1}\, X_{n-1}] = 0, \\ [\mathbf{j}, X_0\, Z_1\, X_1 ... Z_{n-1}\, X_{n-1}] = 0. \end{cases} \tag{6}$$

Where $[\mathbf{i}, q]$ (resp. $[\mathbf{j}, q]$) is the $\mathbf{i}$ (resp. $\mathbf{j}$) component of q.

Closure procedure in the spherical case

Minimize

$$\min_{(\varepsilon_1,..,\varepsilon_{n-1})\in\mathbb{R}^{n-1}} \|\tilde{\kappa}^t - \kappa^t\| = \min_{(\varepsilon_1,..,\varepsilon_{n-1})\in\mathbb{R}^{n-1}} \left| \sum_{i=1}^{n-1} \frac{4\,\varepsilon_i^2}{\left(e_{i-1}^t + e_i^t\right)^2} \right|, \qquad (7)$$

subject to the equality constraints :

$$\begin{cases} S_{32} - S_{23} = 0, \\ S_{13} - S_{31} = 0. \end{cases} \qquad (8)$$

where

$$S := R_x(e_0^t)\,R_z(\delta_1^t + \varepsilon_1)\,R_x(e_1^t)...R_z(\delta_{n-1}^t + \varepsilon_{n-1})\,R_x(e_{n-1}^t),$$

The curvature-based and the exterior-angle algorithms

1. For the curvatures-based algorithm :

$$\kappa_i^t = (1-t)\,\kappa_i^0 + t\,\kappa_i^1, \qquad i \in \{1, ..., n-1\}. \tag{9}$$

2. For the exterior angle blending method :

$$\delta_i^t = (1-t)\,\delta_i^0 + t\,\delta_i^1, \qquad i \in \{1, ..., n-1\}, \tag{10}$$

Blue : Exterior- angle method, **red** : Curvature-based method

Figure 1 – caption

Blue : Exterior- angle method, **red** : Curvature-based method

Figure 2 – caption

Panoramic images (36degree freedom images)