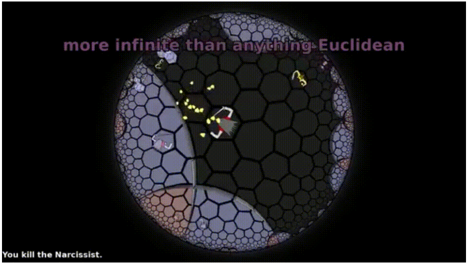
Morphing of Hyperbolic Closed Curves

T. Ahanchaou A. Ikemakhen,

Cadi-Ayyad University, Faculty of Science and Technology, Marrakesh, Morocco

24 novembre 2021

Motivation



- **Hyperrogue games :** This game is developed on a non-Euclidean space. Namely, the Poincaré disk model.

Motivation

- **Shape Morphing (or shape blending)** is a special effect in motion pictures and animations that changes (or morphs) one shape into another through a continuous transition.

Motivation

- **Shape Morphing (or shape blending)** is a special effect in motion pictures and animations that changes (or morphs) one shape into another through a continuous transition.
- Morphing has wide practical use in areas such as computer graphics, animation and modeling.

- The blending between two closed curves plays an important role in the area of generation of animation .

- The blending between two closed curves plays an important role in the area of generation of animation .
- Morphing on surfaces is another concept of morphing.

- The blending between two closed curves plays an important role in the area of generation of animation .
- Morphing on surfaces is another concept of morphing. The source and target closed curves are given on the surface and the intermediate curves must
  1. stay on the surface,

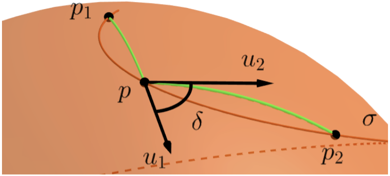
- The blending between two closed curves plays an important role in the area of generation of animation .
- Morphing on surfaces is another concept of morphing. The source and target closed curves are given on the surface and the intermediate curves must
  1. stay on the surface,
  2. be closed.

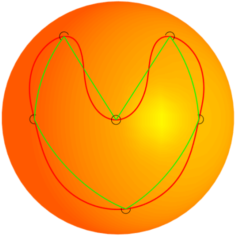
- The blending between two closed curves plays an important role in the area of generation of animation .
- Morphing on surfaces is another concept of morphing. The source and target closed curves are given on the surface and the intermediate curves must
  1. stay on the surface,
  2. be closed.

In this talk, we deal with morphing of closed curves on Poincaré disk model, and we will answer to these requirements.

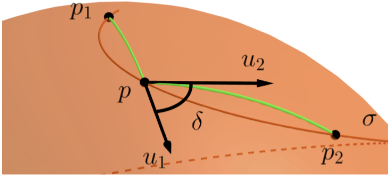
Morphing of Hyperbolic Closed Curves

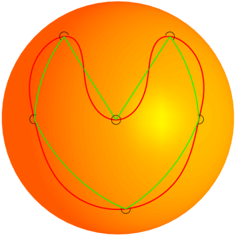
T. Ahanchaou A. Ikemakhen,





- A closed C^2-curve γ can be approximate by an inscribed polygon P .





- A closed C^2-curve γ can be approximate by an inscribed polygon P .
- and the geodesic curvature of σ at a vertex p can be approximate by the discrete geodesic curvature of P at p :

$$\kappa(p) = \lim_{\substack{z_1, z_2 \to p \\ z_1, z_2 \in C}} \frac{2\ \delta}{d(z_1, p) + d(p, z_2)}.$$

So in practice we manipulate polygons (or discrete curves) instead of curves.

Related work : planar and spherical morphing closed curves

Planar case

- Exterior angles-based blending method : [*Sederberg& Gao& Wang & Mu*; 1993] .

Related work : planar and spherical morphing closed curves

Planar case

- Exterior angles-based blending method : [*Sederberg& Gao& Wang & Mu*; 1993] .
- Curvature-based blending method :
  - [*Surazhsky & Elber*; 2002] .
  - [*Saba & Schneider & Hormann & Scateni*; 2014]

Related work : planar and spherical morphing closed curves

Planar case

- Exterior angles-based blending method : [*Sederberg& Gao& Wang & Mu*; 1993] .
- Curvature-based blending method :
  - [*Surazhsky & Elber*; 2002] .
  - [*Saba & Schneider & Hormann & Scateni*; 2014]
- Curvature flow-based blending method :
  - [*Crane & Pinkall & Schröder*; 2013]
  - [*Hirano & Watanabe & Ishikawa*; 2017].

Related work : planar and spherical morphing closed curves

Planar case

- Exterior angles-based blending method : [*Sederberg& Gao& Wang & Mu*; 1993] .
- Curvature-based blending method :
  - [*Surazhsky & Elber*; 2002] .
  - [*Saba & Schneider & Hormann & Scateni*; 2014]
- Curvature flow-based blending method :
  - [*Crane & Pinkall & Schröder*; 2013]
  - [*Hirano & Watanabe & Ishikawa*; 2017].

Spherical case

- [*Ikemakhen & Bellaihou & Ahanchaou*; 2021]

The Poincaré disc model

The Poincaré disc is the open unit disc $\mathbb{D} = \{z \in \mathbb{C} \mid \mid z \mid < 1\}$, where

- Boundary is represented by the circle at infinity.

The Poincaré disc model

The Poincaré disc is the open unit disc $\mathbb{D} = \{z \in \mathbb{C} \mid \mid z \mid < 1\}$, where

- Boundary is represented by the circle at infinity.
- Riemannian metric :

$$g = 4\frac{\mid dz \mid^2}{(1- \mid z \mid^2)^2}$$

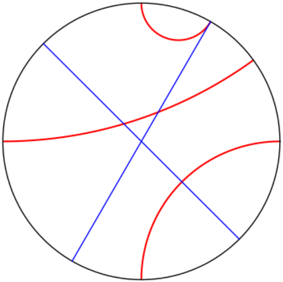
The Poincaré disc model

The Poincaré disc is the open unit disc $\mathbb{D} = \{z \in \mathbb{C} \mid \mid z \mid < 1\}$, where

- Boundary is represented by the circle at infinity.
- Riemannian metric :

$$g = 4\frac{\mid dz \mid^2}{(1- \mid z \mid^2)^2}$$

- Geodesics : line segments through the origin and the circular arcs that intersect the boundary orthogonally.



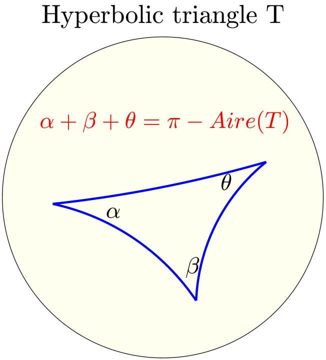
- Hyperbolic distance :

$$\cosh\left(d\left(z_1, z_2\right)\right) = 1 + \frac{\mid z_1 - z_2 \mid^2}{(1 - \mid z_1 \mid^2)(1 - \mid z_2 \mid^2)}.$$

- Hyperbolic distance :

$$\cosh\left(d\left(z_1, z_2\right)\right) = 1 + \frac{\mid z_1 - z_2 \mid^2}{(1 - \mid z_1 \mid^2)(1 - \mid z_2 \mid^2)}.$$

- $\alpha + \beta + \theta = \pi - \mathit{area}$.



Mobïus Transformations

$$SU(1,1) := \left\{ \begin{pmatrix} a & b \\ \bar{b} & \bar{a} \end{pmatrix} \mid a, b \in \mathbb{C} \mid a\bar{a} - b\bar{b} = 1 \right\}.$$

The Mobius group $PSU(1,1) := SU(1,1)/\pm I$ acts transitively on the Poincaré disc $\mathbb{D}$:

$$\begin{array}{cccc} \rho & : & PSU(1,1) \times \mathbb{D} & \to & \mathbb{D}, \\ & & \left(\begin{pmatrix} a & b \\ \bar{b} & \bar{a} \end{pmatrix}, z \right) & \mapsto & \dfrac{az + \bar{b}}{\bar{b}z + \bar{a}}. \end{array}$$

Translations and Rotations

- The rotation around the origin O by angle θ :

$$R(\theta) := \begin{pmatrix} e^{\frac{i\theta}{2}} & 0 \\ 0 & e^{\frac{-i\theta}{2}} \end{pmatrix}.$$

Translations and Rotations

- The rotation around the origin O by angle θ :

$$R(\theta) := \begin{pmatrix} e^{\frac{i\theta}{2}} & 0 \\ 0 & e^{\frac{-i\theta}{2}} \end{pmatrix}.$$

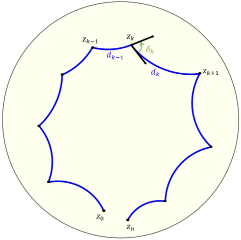
- The translation of length d along the geodesic that sends -1 to 1 is

$$L(d) := \begin{pmatrix} \cosh(\frac{d}{2}) & \sinh(\frac{d}{2}) \\ \sinh(\frac{d}{2}) & \cosh(\frac{d}{2}) \end{pmatrix}.$$

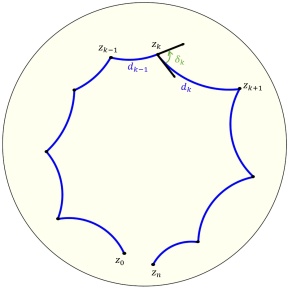
Hyperbolic Polygon

• A hyperbolic polygon $P = [z_0, ..., z_n] \in \mathbb{D}$: edges are pieces of geodesics. • The intrinsic parameters of P at any vertex z_k are :

- Geodesic edge length : $d_k := d(z_k, z_{k+1})$,
- Exterior angle : δ_k
- Discrete geodesic curvature : $\kappa_g(z_k) := \dfrac{2\ \delta_k}{d_{k-1} + d_k}$

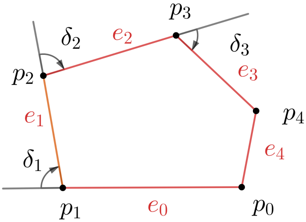


Fundamental Question : What is the Closure conditions of a hyperbolic polygon in terms of its intrinsic parameters ?



Planar Case. $P = [z_0, ..., z_n]$ is a closed polygon

$$\textit{iff} \quad \sum_{i=0}^{n} \vec{e_i} = \vec{0} \quad \textit{iff} \quad \begin{cases} \sum_{i=0}^{n} e_i \sin\left(\sum_{k=0}^{i} \delta_k\right) = 0, \\ \sum_{i=0}^{n} e_i \cos\left(\sum_{k=0}^{i} \delta_k\right) = 0. \end{cases}$$



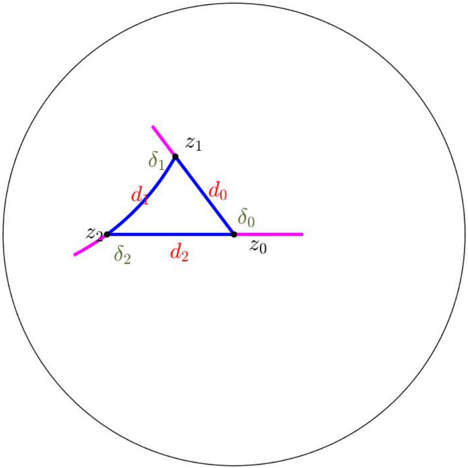
Closing condition for a Hyperbolic Triangle

Let $\Pi = [z_0 = z_3, z_1, z_2, z_3]$ be a closed hyperbolic triangle with hyperbolic sides d_0, d_1, d_2 and exterior angles δ_0, δ_1, δ_2. A hyperbolic triangle $T = [z_0, z_1, z_2]$ is closed iff

$$R(\delta_0)L(d_0)R(\delta_1)L(d_1)R(\delta_2)L(d_2) = \pm I,$$

- $R(\delta_0)$ is the rotation with angle δ_0 etc...
- $L(d_0)$ is the translation along the geodesic (z_0, z_1) etc...

$$L\left(d_2\right)$$



Closing condition for a Hyperbolic Polygon

$P = [z_0, ..., z_n]$ is a closed hyperbolic polygon iff

$$R(\delta_0)L(d_0)R(\delta_1)L(d_1)\cdots R(\delta_{n-2})L(d_{n-2})R(\delta_{n-1})L(d_{n-1}) = \pm I_d.$$

iff

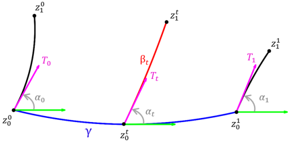
$$\begin{cases} \mid tr(S) \mid & = & 2, \\ det(S) & = & 1, \\ s_2\bar{s}_2 & = & 0. \end{cases}$$

Where

$$S := R(\delta_0)L(d_0)R(\delta_1)L(d_1)\cdots R(\delta_{n-2})L(d_{n-2})R(\delta_{n-1})L(d_{n-1}).$$

Algorithm : Exterior-angle blending

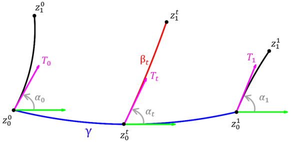
Let given $P^0 = [z_0^0, ..., z_{n-1}^0]$ and $P^1 = [z_0^1, ..., z_{n-1}^1]$ two closed hyperbolic polygons. For $t \in [0, 1]$, we compute



- the geodesic edge lengths : $d_k^t = (1-t)d_k^0 + td_k^1$.

Algorithm : Exterior-angle blending

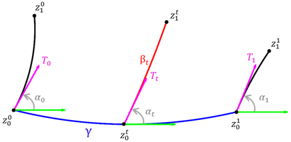
Let given $P^0 = [z_0^0, ..., z_{n-1}^0]$ and $P^1 = [z_0^1, ..., z_{n-1}^1]$ two closed hyperbolic polygons. For $t \in [0, 1]$, we compute



- the geodesic edge lengths : $d_k^t = (1-t)d_k^0 + td_k^1$.
- the exterior angles : $\delta_k^t = (1-t)\delta_k^0 + t\delta_k^1$.

Algorithm : Exterior-angle blending

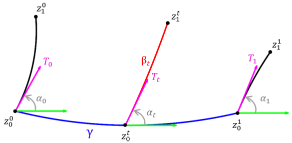
Let given $P^0 = [z_0^0, \ldots, z_{n-1}^0]$ and $P^1 = [z_0^1, \ldots, z_{n-1}^1]$ two closed hyperbolic polygons. For $t \in [0, 1]$, we compute



- the geodesic edge lengths : $d_k^t = (1-t)d_k^0 + td_k^1$.
- the exterior angles : $\delta_k^t = (1-t)\delta_k^0 + t\delta_k^1$.
- $\alpha_t := (1-t)\,\alpha_0 + t\,\alpha_1 \Rightarrow T_0^t$

Algorithm : Exterior-angle blending

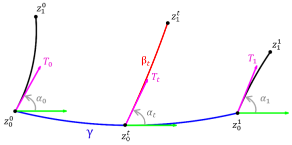
Let given $P^0 = [z_0^0, \ldots, z_{n-1}^0]$ and $P^1 = [z_0^1, \ldots, z_{n-1}^1]$ two closed hyperbolic polygons. For $t \in [0, 1]$, we compute



- the geodesic edge lengths : $d_k^t = (1-t)d_k^0 + td_k^1$.
- the exterior angles : $\delta_k^t = (1-t)\delta_k^0 + t\delta_k^1$.
- $\alpha_t := (1-t)\,\alpha_0 + t\,\alpha_1 \Rightarrow T_0^t$
- We construct the point z_0^t and the geodesic c_0^t,

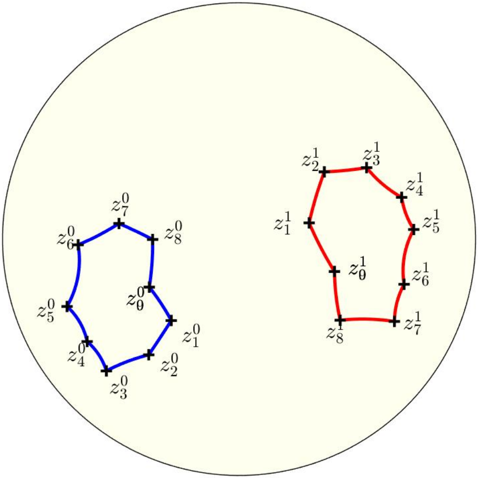
Algorithm : Exterior-angle blending

Let given $P^0 = [z_0^0, ..., z_{n-1}^0]$ and $P^1 = [z_0^1, ..., z_{n-1}^1]$ two closed hyperbolic polygons. For $t \in [0, 1]$, we compute



- the geodesic edge lengths : $d_k^t = (1 - t)d_k^0 + td_k^1$.
- the exterior angles : $\delta_k^t = (1 - t)\delta_k^0 + t\delta_k^1$.
- $\alpha_t := (1 - t)\,\alpha_0 + t\,\alpha_1 \Rightarrow T_0^t$
- We construct the point z_0^t and the geodesic c_0^t,
- By induction, we construct the other geodesic edges c_k^t.

Construction process of intermediate curve



Algorithm2 : Curvature blending

1. Interpolation of the geodesic edge lengths by : $d_k^t = (1-t)d_i^0 + td_k^1$.

Algorithm2 : Curvature blending

1. Interpolation of the geodesic edge lengths by : $d_k^t = (1-t)d_i^0 + td_k^1$.
2. Interpolation of the discrete geodesic curvatures : $\kappa_k^t = (1-t)\kappa_k^0 + t\kappa_k^1$.

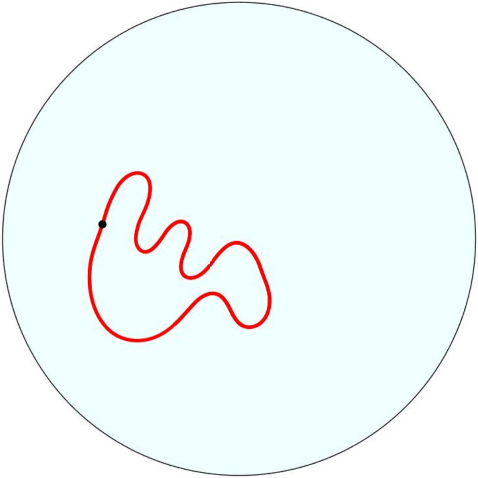
Algorithm2 : Curvature blending

1. Interpolation of the geodesic edge lengths by : $d_k^t = (1-t)d_i^0 + td_k^1$.
2. Interpolation of the discrete geodesic curvatures : $\kappa_k^t = (1-t)\kappa_k^0 + t\kappa_k^1$.
3. Recovery of exterior angles δ_k^t from the κ_k^t

Algorithm2 : Curvature blending

1. Interpolation of the geodesic edge lengths by : $d_k^t = (1-t)d_i^0 + td_k^1$.
2. Interpolation of the discrete geodesic curvatures : $\kappa_k^t = (1-t)\kappa_k^0 + t\kappa_k^1$.
3. Recovery of exterior angles δ_k^t from the κ_k^t
4. By induction, we construct the edge geodesics c_k^t.

Result without closing condition



Closing process

For that, we will change the exterior angles δ_k^t in the smallest possible way to close the intermediate polygon P^t.
This means we seek $\epsilon_0, ...\epsilon_{n-1}$ such that the polygon $\bar{P}^t$, with hyperbolic side lengths d_k^t and exterior angles $\bar{\delta}_k^t := \delta_k^t + \epsilon_k$ will be closed and the norm $\| \bar{\kappa}^t - \kappa^t \|^2$ will be minimized.
Where κ^t (resp. $\bar{\kappa}^t$) denotes the vector of components κ_k^t (resp. $\bar{\kappa}_k^t := \dfrac{2\bar{\delta}_k^t}{d_{k-1}^t + d_k^t}$), and $\| . \|$ is the Euclidean norm in $\mathbb{R}^n$. In order to solve this,

We minimize the following problem :

$$\min_{(\epsilon_0,\dots\epsilon_{n-1})\in\mathbb{R}^n} \sum_{k=0}^{n-1} \left| \frac{4\epsilon_k^2}{\left(d_{k-1}^t + d_k^t\right)^2} \right| . \qquad (1)$$

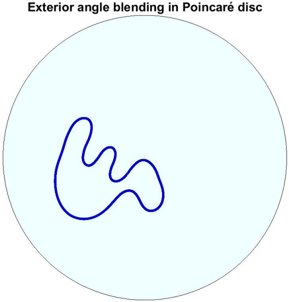
Subject to :

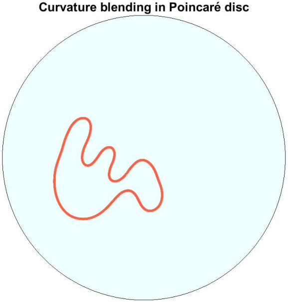
$$\begin{cases} \mid tr(S) \mid & = & 2, \\ det(S) & = & 1, \\ s_2\bar{s}_2 & = & 0. \end{cases} \qquad (2)$$

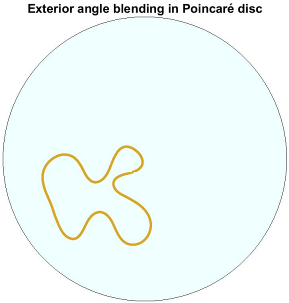
where

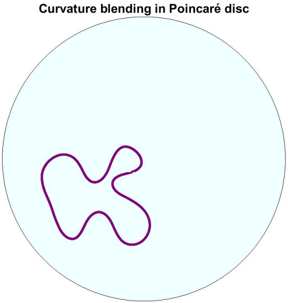
$$S := \prod_{k=0}^{n-1} R(\delta_{n-1-k} + \epsilon_{n-1-k}) L(d_{n-1-k}),$$

This will ensure the closure of the hyperbolic polygon $\bar{P}^t$.

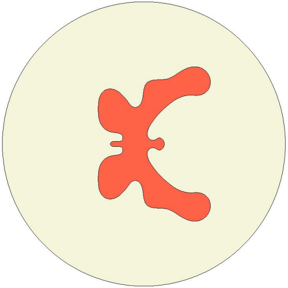

Exterior angle blending in Poincaré disc

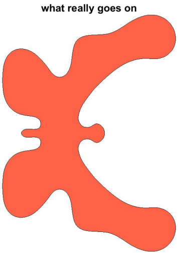

Curvature blending in Poincaré disc


Exterior angle blending in Poincaré disc

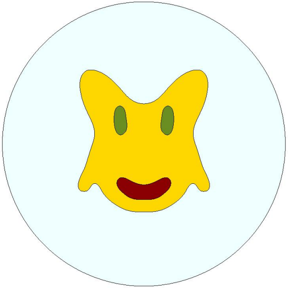

Curvature blending in Poincaré disc

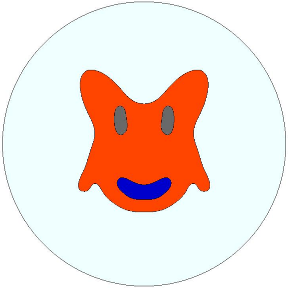
Morphing sequence between a butterfly and a bat at infinity





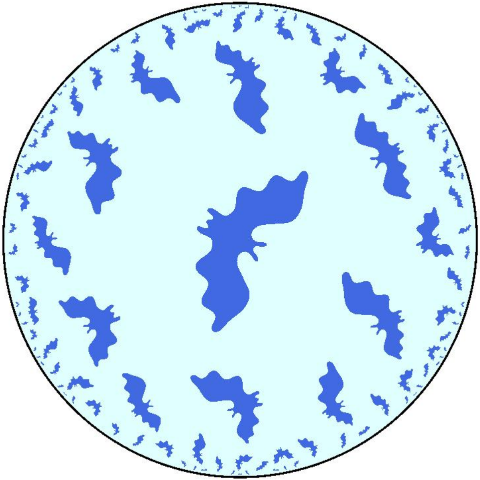
Morphing sequence between a wolf's face and a bat



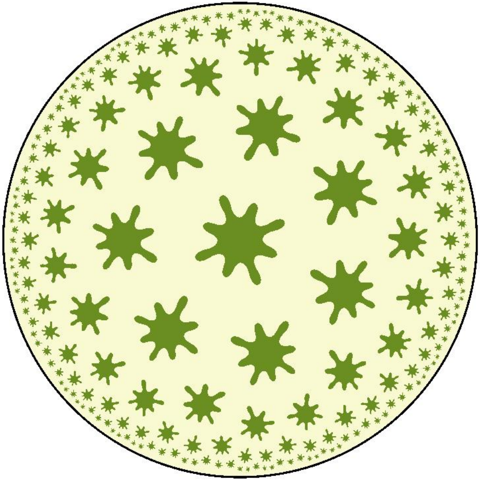


Tiling the Poincaré disc using the blending of two given motifs

Tiling the Poincaré disc using the blending of two given motifs



Tiling the Poincaré disc using the blending of two given motifs



Conclusion

- We have presented two novel algorithms for blending between two curves in the Poincaré disc, using their intrinsic variables.

Conclusion

- We have presented two novel algorithms for blending between two curves in the Poincaré disc, using their intrinsic variables.
- Both methods generate closed intermediate smooth curves by using the closure condition and by solving an optimization problem.

Conclusion

- these methods give tools to approach the geometry processing problem in the hyperbolic space and could give a contribution in this direction.

Conclusion

- these methods give tools to approach the geometry processing problem in the hyperbolic space and could give a contribution in this direction.
- Limitation. Both algorithms take a long time to generate intermediate curves.

Conclusion

- these methods give tools to approach the geometry processing problem in the hyperbolic space and could give a contribution in this direction.
- Limitation. Both algorithms take a long time to generate intermediate curves.
- Therefore, these methods can't be applied for a real-time execution.

Conclusion

- these methods give tools to approach the geometry processing problem in the hyperbolic space and could give a contribution in this direction.
- Limitation. Both algorithms take a long time to generate intermediate curves.
- Therefore, these methods can't be applied for a real-time execution.
- The goal of our future work is to give a rapid blending method which reduces the run-time.

This work was published at Eurographics Symposium on Geometry Processing 2021
https ://doi.org/10.1111/cgf.14358

References I

[1] Adrien Bernhardt et al. “Implicit blending revisited”. In : *Computer Graphics Forum*. T. 29. 2. Wiley Online Library. 2010, p. 367-375.

[2] Renjie Chen et al. “Planar shape interpolation with bounded distortion”. In : *ACM Transactions on Graphics (TOG)* 32.4 (2013), p. 1-12.

[3] Nadav Dym, Anna Shtengel et Yaron Lipman. “Homotopic morphing of planar curves”. In : *Computer Graphics Forum*. T. 34. Wiley Online Library. 2015, p. 239-251.

References II

[4] R.S. Earp et E. Toubiana. *Introduction à la géométrie hyperbolique et aux surfaces de Riemann*. Bibliothèque des sciences. Diderot Editeur Arts Sciences, 1997. ISBN : 9782841340019. URL : https://books.google.co.ma/books?id=TtbTAQAACAAJ.

[5] F Guimaraes, V Mello et Luiz Velho. “Geometry independent game encapsulation for non-euclidean geometries”. In : *Proceedings of SIBGRAPI Workshop of Works in Progress*. 2015.

[6] Masahiro Hirano, Yoshihiro Watanabe et Masatoshi Ishikawa. “Rapid blending of closed curves based on curvature flow”. In : *Computer Aided Geometric Design* 52 (2017), p. 217-230.

References III

[7] Birger Iversen et Iversen Birger. *Hyperbolic geometry*. T. 25. Cambridge University Press, 1992.

[8] Eryk Kopczynski, Dorota Celinska et Marek Ctrnáct. "Hyperrogue : Playing with hyperbolic geometry". In : *the proceedings of the Bridges Conference, July*. 2017, p. 27-31.

[9] Marianna Saba et al. "Curvature-based blending of closed planar curves". In : *Graphical models* 76.5 (2014), p. 263-272.

[10] Thomas W Sederberg et al. "2-D shape blending : an intrinsic solution to the vertex path problem". In : *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*. 1993, p. 15-18.

References IV

[11] Tatiana Surazhsky et Gershon Elber. "Metamorphosis of planar parametric curves via curvature interpolation". In : *International Journal of Shape Modeling* 8.02 (2002), p. 201-216.