Forum "Math-for-Industry" 2016 -Biology,Agriculture and Environment-

A Mathematica module for two-dimensional computer graphics -Data structure and Interpolation algorithms-

Tomomi Hirano

Graduate School of Mathematics, Kyushu University

ma215037@math.kyushu-u.ac.jp

Abstract

2D shape interpolation is widely used in Computer Graphics. We introduce a Mathematica module for drawing, transformation, interpolation of two-dimensional polygon figure using results [1] and [2]. We can analyse and investigate critical examples of interpolations using our module. Symbolic computations in Mathematica enable us to evaluate those examples using several mathematical formulas. We can use our functions to produce a pre-expanded formulas to compute an interpolation in another language such as C which does not have a facility of symbolic computations.

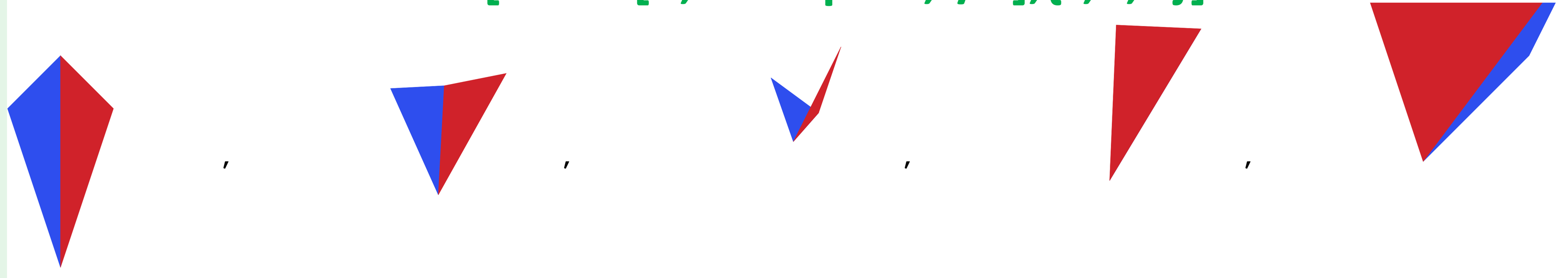
2D Shape Interpolation

Interpolating a source and target figures of 2D polygons.

Examples

Alexa + Alexa Error + C2

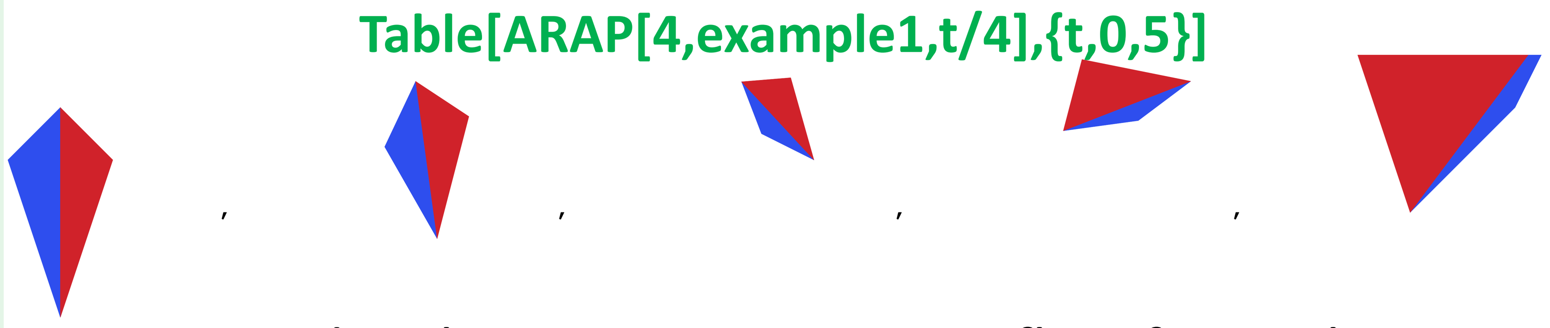
Table[ARAP[2,example1,t/4],{t,0,5}]



The triangle collapses and then turns over.

Alexa + Sim Error+ C2

Table[ARAP[4,example1,t/4],{t,0,5}]



Rotation and scale invariance prevents flip of triangles.

Local Interpolations

Interpolation of each triangle.[1][2]

- Linear Interpolation $A^L(t) := (1-t)E + tA$
  Each point move in a straight line.
- Alexa's Interpolation $A^P(t) = R_{t\theta}((1-t)E) + tS)$
  Rotation Matrix × Symmetric Matrix(Linear)
- Log-Exp Interpolation $A^E(t) = R_{t\theta}\exp(t\log S)$
  Rotate Matrix × Symmetric Matrix

Error functions

It measures how different local and global interpolation are as linear maps(Alexa)/linear maps up to rotation and scale(Sim).[1][3][4][5][6]

- Alexa's Error $E_k^F(B_k, A_k(t)) := ||B_k - A_k(t)||_F^2$
- Sim Error $E_k^R(B_k, A_k(t)) := ||B_k||_F^2 - \frac{||B_k \cdot A_k^T||_F^2 + 2\det(B_k \cdot A_k^T)}{||A_k||_F^2}$

Constraint functions

It controls the global interpolation.[2]

- $C1(v(t)) = ||(1-t)p_1 + tq_1 - v_1(t)||^2$
- $C2(v(t)) = ||(1-t)p_1 + tq_1 - v_1(t)||^2 + ||(1-t)p_2 + tq_2 - v_2(t)||^2$
- $C3(v(t)) = ||(1-t)p_m + tq_m - v_m(t)||^2$
- $C4(v(t)) = ||v_k(t) - v_l(t) - e_{kl}(t)||$ $\quad (e_{kl}(t) = e_l(t) - e_l(t))$
- $C5(v(t)) = ||v_k(t) - v_l(t) - e_{kl}'(t)||$ $\quad (e_{kl}'(t) = R(2\pi t)e_{kl}(t))$

Global Error Functions

- $E_F(t) = \sum_{k\in\Delta} ||B_k - A_k(t)||_F^2 + C(v(t))$
- $E_S(t) = \sum_{k\in\Delta} ||B_k||_F^2 - \frac{||B_k \cdot A_k^T||_F^2 + 2\det(B_k \cdot A_k^T)}{||A_k||_F^2} + C(v(t))$

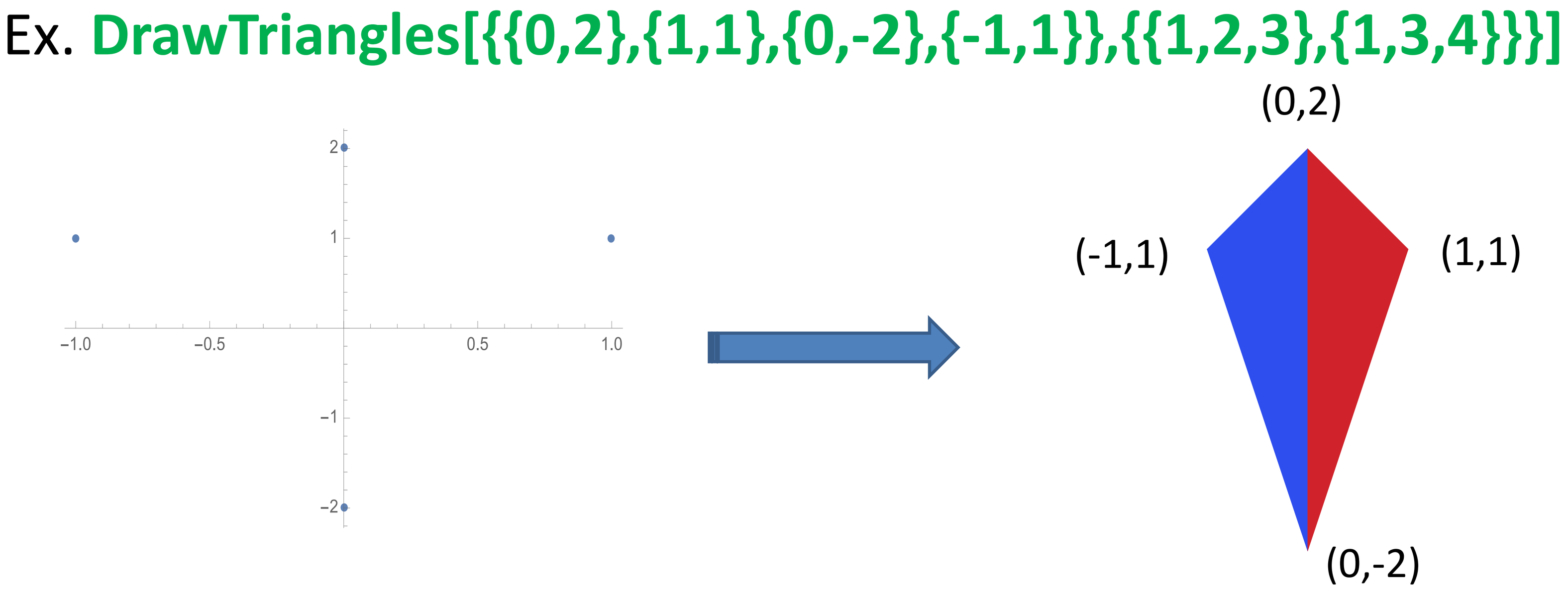
Goal

Compute v(t) that minimizes the Global Error Function.

The module can …

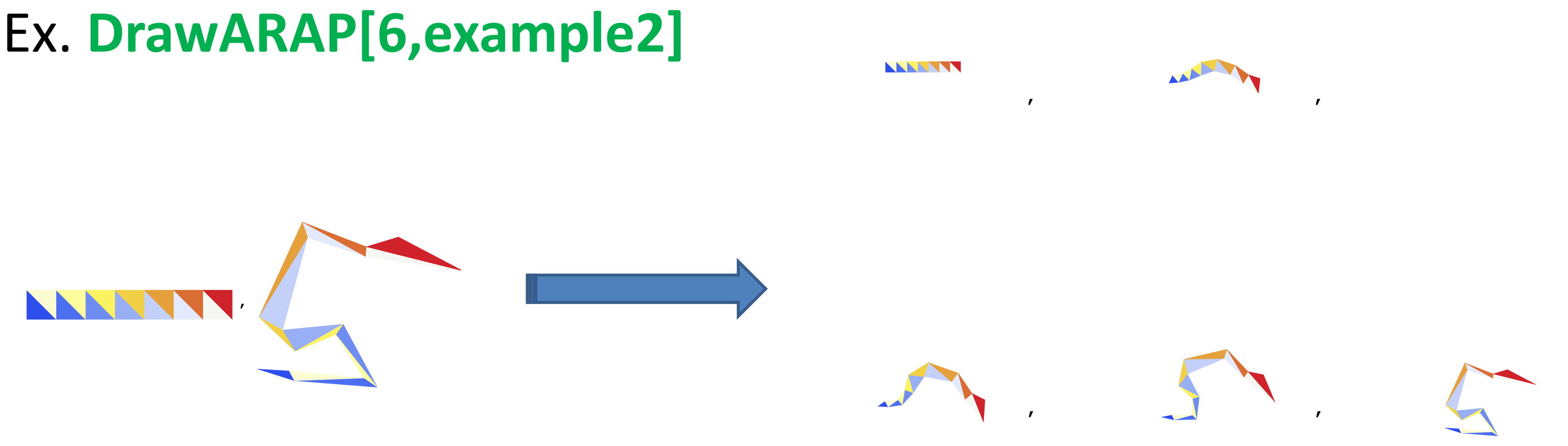
- Draw triangled polygons assigning colors.

Ex. DrawTriangles[{{0,2},{1,1},{0,-2},{-1,1}},{{1,2,3},{1,3,4}}]



- Make an interpolation from a source and a target figures.

Ex. DrawARAP[6,example2]



Speeding up using pre-expanded formulas

- We note that every error function is a positive quadratic form in elements of v(t).In order to have a unique minimizer v(t), we need some constraints. The minimization problem is solved as an inverse matrix of a quadratic coefficients matrix of an global error function. Our module can make a symbolic representation of the quadratic coefficients matrix.
- The case of using Alexa's error, the matrix is defined time-independent, so we only need to compute them just once. The case of using Sim error, the matrix is depend on time, but our module give a symbolical expansions of a function which can be used in another efficient language as a hard coding.

Conclusion

- We developed a Mathematica module for checking an effect of 2D-ARAP interpolations easily.
- We can use our functions to produce a pre-expanded formulas to compute an interpolation in C programming language.
- Future works include to extend our module for 3D CG and develop interface to other CG software such as Maya.
- This module have been published on GitHub with its manual. https://github.com/KyushuUniversityMathematics/MathematicaARAP

References

1. Marc Alexa, Daniel Cohen-Or, and David Levin. As-rigid-as-possible shape interpolation. In Proc. the 27th Annual Conference on Com- puter Graphics and Interactive Techniques, pages 157–164. ACM Press/Addison-Wesley Publishing Co., 2000.
2. Shizuo Kaji, Sampei Hirose, Shigehiro Sakata, Yoshihiro Mizoguchi, and Ken Anjyo. Mathematical analysis on affine maps for 2d shape interpolation. In Proc. the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pages 71–76. Eurographics Association, 2012.
3. Shizuo Kaji, Sampei Hirose, Hiroyuki Ochiai, and Ken Anjyo. A Lie theoretic parameterization of affine transformations. In Proc. Sympo- sium MEIS2013: Mathematical Progress in Expressive Image Synthesis, volume 50 of MI Lecture Note, pages 134–140. Kyushu University, 2013.
4. Takeo Igarashi and Yuki Igarashi. Implementing as-rigid-as- possible shape manipulation and surface flattening. J. Graphics, GPU, & Game Tools 14, 1 (2009), 17–30.
5. Takeo Igarashi, Tomer Moscovich and John F. Hughes. As- rigid-as-possible shape manipulation. ACM Trans. Graph. 24, 3 (2005), 1134–1141.
6. Michael Werman and Daphna Weinshall. Similarity and affine invariant distances between 2d point sets. IEEE Transactions on Pattern Analysis and Machine Intelligence 17 (1995), 810–814.