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

Tomomi Hirano
Graduate School of Mathematics, Kyushu University *

November 21, 2016

1 Introduction

2D shape interpolation is widely used in Computer Graphics. In 2000, Alexa and Xu suggested new algorithm preserving rigidity for algorithm of this interpolation [?] .These algorithm consists of local interpolations and a global interpolation. The local interpolation means an interpolation between one souce mesh and target mesh. Kaji gave the new parameterization method, a computation algorithm, and applications to shape deformation using Lie group and Lie algebra in [?]. In [?], they presented the algorithms to achieve global interpolation, each of which minimizes an error function with user-specified constraints. We introduce a Mathematica module for drawing,transformation,interpolation of two-dimensional polygon figure using results [?] and [?]. We can analyse and investigate critical examples of interpolations using our module. Symbolic computations in Mathmatica enable us a simple method to evaluate those examples using several mathematical formulas. In [?], they showed a closed form for the similarity invariant error function. Further, since it is a quadratic polynomial, we can computate a time-independent matrix used for getting the minimizer. In our module, it contains a function to produce a time-independent matrix by given coordinates of an source figure and a target figure. Giving those coordinates as variable symbols, our function returns a matrix which elements are polynomials of variable symbols. Those polynomials are used for hard cording in another language such as C which does not have a facility of symbolic computations. This means, we can execute faster after compiling the extracted cord.Our module will be publish in Github [1].

2 Preliminary

We consider a 2×2-matrix $A = \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix}$, and an affine matrix $\begin{pmatrix} \multicolumn{2}{c}{B} & \alpha \\ & & \beta \\ 0 & 0 & 1 \end{pmatrix}$ which transform three points (a_x, a_y), (b_x, b_y) and (c_x, c_y) to (v_{1x}, v_{1y}), (v_{2x}, v_{2y}) and (v_{3x}, v_{3y}), respectively. That is

$$\begin{pmatrix} \multicolumn{2}{c}{B} & \alpha \\ & & \beta \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} v_{1x} & v_{2x} & v_{3x} \\ v_{1y} & v_{2y} & v_{3y} \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} a_x & b_x & c_x \\ a_y & b_y & c_y \\ 1 & 1 & 1 \end{pmatrix}^{-1}.$$

Let $R_\delta = \begin{pmatrix} \cos\delta & -\sin\delta \\ \sin\delta & \cos\delta \end{pmatrix}$ be a rotation matrix. The Frobenius norm $||M||_F$ of a matrix $M = (m_{i,j})$ is defined by $||M||_F^2 = Tr(M \cdot M^T) = \sum_{i,j} m_{i,j}^2$.

* ma215037@math.kyushu-u.ac.jp
[1] `https://github.com/KyushuUniversityMathematics/MathematicaARAP`

For a given A and B, we want to find a values of (v_{1x}, v_{1y}), (v_{2x}, v_{2y}) and (v_{3x}, v_{3y}), which minimize $||B-A||_F^2$. Since $||B-A||_F^2$ is a quadratic polynomials of v_{ix} and v_{iy}, we can find them using the least square method. To solve the problem, we prepare a pre-computed quadratic form matrix defined by simple components of polynomials.

Proposition 1.

$$
\begin{aligned}
||B-A||_F^2 &= \frac{1}{q}\begin{pmatrix} v_{1x} & v_{2x} & v_{3x} & v_{1y} & v_{2y} & v_{3y} \end{pmatrix}\begin{pmatrix} p_{11} & p_{12} & p_{13} & 0 & 0 & 0 \\ p_{12} & p_{22} & p_{23} & 0 & 0 & 0 \\ p_{13} & p_{23} & p_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & p_{11} & p_{12} & p_{13} \\ 0 & 0 & 0 & p_{12} & p_{22} & p_{23} \\ 0 & 0 & 0 & p_{13} & p_{23} & p_{33} \end{pmatrix}\begin{pmatrix} v_{1x} \\ v_{2x} \\ v_{3x} \\ v_{1y} \\ v_{2y} \\ v_{3y} \end{pmatrix} \\
&\quad + \begin{pmatrix} k_1 & k_2 & k_3 & k_4 & k_5 & k_6 \end{pmatrix}\begin{pmatrix} v_{1x} \\ v_{2x} \\ v_{3x} \\ v_{1y} \\ v_{2y} \\ v_{3y} \end{pmatrix}
\end{aligned}
$$

where

$$
\begin{aligned}
q &= (a_y b_x - a_x b_y - a_y c_x + b_y c_x + a_x c_y - b_x c_y)^2, \\
p_{11} &= b_x^2 + b_y^2 - 2b_x c_x + c_x^2 - 2b_y c_y + c_y^2, \\
p_{12} &= -(a_x b_x + a_y b_y - a_x c_x - b_x c_x + c_x^2 - a_y c_y - b_y c_y + c_y^2), \\
p_{13} &= -b_x^2 + a_x(b_x - c_x) + b_x c_x + (a_y - b_y)(b_y - c_y), \\
p_{22} &= a_x^2 + a_y^2 - 2a_x c_x + c_x^2 - 2a_y c_y + c_y^2, \\
p_{23} &= -(a_x^2 + a_y^2 + b_x c_x - a_x(b_x + c_x) + b_y c_y - a_y(b_y + c_y)), \text{ and} \\
p_{33} &= a_x^2 + a_y^2 - 2a_x b_x + b_x^2 - 2a_y b_y + b_y^2.
\end{aligned}
$$

We have implemented a function $F1a$ which compute the quadratic form matrix, i.e.

$$
F1a(\left(\begin{pmatrix} a_x \\ a_y \end{pmatrix}, \begin{pmatrix} b_x \\ b_y \end{pmatrix}, \begin{pmatrix} c_x \\ c_y \end{pmatrix}\right), \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix}) = \frac{1}{q}\begin{pmatrix} p_{11} & p_{12} & p_{13} \\ p_{12} & p_{22} & p_{23} \\ p_{13} & p_{23} & p_{33} \end{pmatrix}.
$$

We can verify the fact in Proposition **??** by using a symboic computation using Mathematica.

$$
A = \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix}; B = \left(\begin{pmatrix} v_{1x} & v_{2x} & v_{3x} \\ v_{1y} & v_{2y} & v_{3y} \\ 1 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} a_x & b_x & c_x \\ a_y & b_y & c_y \\ 1 & 1 & 1 \end{pmatrix}^{-1}\right)[[1;;2, 1;;2]];
$$

$$
\mathrm{F1a}(\left(\begin{pmatrix} a_x \\ a_y \end{pmatrix}, \begin{pmatrix} b_x \\ b_y \end{pmatrix}, \begin{pmatrix} c_x \\ c_y \end{pmatrix}\right), A) ==
$$

$$
\mathrm{QuadraticFormMatrix}\left(\mathrm{NormF}(B-A), \{v_{1x}, v_{2x}, v_{3x}, v_{1y}, v_{2y}, v_{3y}\}\right)[[1;;3, 1;;3]]//\mathrm{FullSimplify}
$$

True

A similarity distance between two 2×2 matrices $\min_{s,\delta\in\mathbb{R}} ||sR_\delta A - B||_F^2$ can be represented by a closed formula using trace and determinant functions.

Proposition 2. [?]

$$
\min_{s,\delta\in\mathbb{R}} ||sR_\delta A - B||_F^2 = ||B||_F^2 - \frac{||B\cdot A^T||_F^2 + 2\det(B\cdot A^T)}{||A||_F^2}
$$

Since the similarity distance $\min_{s,\delta\in\mathbb{R}} ||sR_\delta A - B||_F^2$ is also represented by a quadratic polynomials of v_{ix} and v_{iy}, We also prepare a pre-computed quadratic form matrix defined by simple components of polynomials.

Proposition 3.

$$
\begin{aligned}
& min_{s,\delta\in\mathbf{R}}||B - sR_\delta A||_F^2 \\
= \quad & \frac{1}{f_m X^2} \begin{pmatrix} v_{1x} & v_{1y} & v_{2x} & v_{2y} & v_{3x} & v_{3y} \end{pmatrix} \begin{pmatrix} Y_5 & 0 & Y_4 & -t_2 & Y_1 & t_2 \\ 0 & Y_5 & t_2 & Y_4 & -t_2 & Y_1 \\ Y_4 & t_2 & Y_6 & 0 & -Y_2 & -t_2 \\ -t_2 & Y_4 & 0 & Y_6 & t_2 & -Y_2 \\ Y_1 & -t_2 & -Y_2 & t_2 & Y_3 & 0 \\ t_2 & Y_1 & -t_2 & -Y_2 & 0 & Y_3 \end{pmatrix} \begin{pmatrix} v_{1x} \\ v_{1y} \\ v_{2x} \\ v_{2y} \\ v_{3x} \\ v_{3y} \end{pmatrix}
\end{aligned}
$$

where

$$
\begin{aligned}
f_m &= m_{11}^2 + m_{12}^2 + m_{21}^2 + m_{22}^2, \\
f_1 &= m_{11}^2 + m_{21}^2, \\
f_2 &= m_{12}^2 + m_{22}^2, \\
t_1 &= m_{11}m_{12} + m_{21}m_{22}, \\
t_2 &= m_{12}m_{21} - m_{11}m_{22}, \\
X &= a_y(b_x - c_x) + b_y c_x - b_x c_y + a_x(-b_y + c_y) \\
Y_1 &= -b_x^2 f_1 + a_x(b_y t_1 - c_y t_1 + (b_x - c_x) f_1) \\
&\quad + b_x((a_y - 2b_y + c_y)t_1 + c_x f_1) + (a_y - b_y)(-c_x t_1 + (b_y - c_y) f_2), \\
Y_2 &= (-a_y b_x t_1 - a_y c_x t_1 + b_y c_x t_1 + b_x c_y t_1 + a_x^2 f_1 + b_x c_x f_1 \\
&\quad - a_x((-2a_y + b_y + c_y)t_1 + (b_x + c_x) f_1) + (a_y - b_y)(a_y - c_y) f_2), \\
Y_3 &= ((a_x - b_x)(2a_y t_1 - 2b_y t_1 + (a_x - b_x) f_1) + (a_y - b_y)^2 f_2), \\
Y_4 &= (-a_y b_x t_1 - a_x b_y t_1 + a_x c_y cm + b_x c_y t_1 - c_x^2 m_{11}^2 - a_x b_x f_1, \\
&\quad + a_x c_x f_1 + b_x c_x f_1 + a_y c_x m_{11} m_{12} + b_y c_x m_{11} m_{12}, \\
&\quad - 2c_x c_y m_{11} m_{12} - a_y b_y m_{12}^2 + a_y c_y m_{12}^2 + b_y c_y m_{12}^2 - c_y^2 m_{12}^2 - c_x^2 m_{21}^2, \\
&\quad + c_x(a_y + b_y - 2c_y) m_{21} m_{22} + (a_y - c_y)(-b_y + c_y) m_{22}^2), \\
Y_5 &= ((b_x - c_x)(2b_y t_1 - 2c_y t_1 + (b_x - c_x) f_1) + (b_y - c_y)^2 f_2), \ \textit{and} \\
Y_6 &= ((a_x - c_x)(2a_y t_1 - 2c_y t_1 + (a_x - c_x) f_1) + (a_y - c_y)^2 f_2).
\end{aligned}
$$

We have implemented a function $F2a$ which compute the quadratic form matrix, i.e.

$$F2a(\left(\begin{pmatrix} a_x \\ a_y \end{pmatrix}, \begin{pmatrix} b_x \\ b_y \end{pmatrix}, \begin{pmatrix} c_x \\ c_y \end{pmatrix}\right), \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix}) = \frac{1}{f_m X^2} \begin{pmatrix} Y_5 & 0 & Y_4 & -t_2 & Y_1 & t_2 \\ 0 & Y_5 & t_2 & Y_4 & -t_2 & Y_1 \\ Y_4 & t_2 & Y_6 & 0 & -Y_2 & -t_2 \\ -t_2 & Y_4 & 0 & Y_6 & t_2 & -Y_2 \\ Y_1 & -t_2 & -Y_2 & t_2 & Y_3 & 0 \\ t_2 & Y_1 & -t_2 & -Y_2 & 0 & Y_3 \end{pmatrix}.$$

We can verify the fact in Proposition **??** by using a symboic computation using Mathematica.

$$A = \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix}; B = \left(\begin{pmatrix} v_{1x} & v_{2x} & v_{3x} \\ v_{1y} & v_{2y} & v_{3y} \\ 1 & 1 & 1 \end{pmatrix} . \begin{pmatrix} a_x & b_x & c_x \\ a_y & b_y & c_y \\ 1 & 1 & 1 \end{pmatrix}^{-1}\right) [[1;;2, 1;;2]];$$

$$\mathrm{F2a}(\left(\begin{pmatrix} a_x \\ a_y \end{pmatrix}, \begin{pmatrix} b_x \\ b_y \end{pmatrix}, \begin{pmatrix} c_x \\ c_y \end{pmatrix}\right), A) ==$$

$$\mathrm{QuadraticFormMatrix}\left[\mathrm{NormF}(B) - \frac{2\left|B.A^T\right| + \mathrm{NormF}\left(B.A^T\right)}{\mathrm{NormF}(A)}, \{v_{1x}, v_{1y}, v_{2x}, v_{2y}, v_{3x}, v_{3y}\}\right]$$

//FullSimplify

True

3 ARAP

Let n,m $\in \mathbb{N}, \bar{n} = \{1, 2, \cdots, n\}$. A polygon F $= (\{P_1, P_2, \cdots, P_n\}, \Delta)$
is a pair of a point set $\{P_1, P_2, \cdots, P_n\}$ and a triangle set $\Delta = \{k_1, k_2, \cdots, k_m\}$,
where $k_j \in \bar{n} \times \bar{n} \times \bar{n}$ and $P_i \in \mathbb{R}^2$.We denote $k_j = (k_j(1), k_j(2), k_j(3))$ and $P_i = (P_{ix}, P_{iy})$.
Let $\mathbf{v}^t = (v_{1x}, \cdots, v_{nx}, v_{1y}, \cdots, v_{ny})$ and $\mathbf{v}_k^t = (v_{k(1)x}, v_{k(2)x}, v_{k(3)x}, v_{k(1)y}, v_{k(2)y}, v_{k(3}y)$.

Let $F_0 = (\{S_1, \cdots, S_n\}, \Delta)$, and $F_1 = (\{T_1, \cdots, T_n\}, \Delta)$be polygons.
For $k \in \Delta$, we define $A_k(t), B_k \in \mathbb{R}^{2\times 2}$ by

$$\begin{aligned}
\begin{pmatrix} \multicolumn{2}{c}{A_k(t)} & \begin{matrix}\alpha \\ \beta\end{matrix} \\ 0 \quad 0 & & 1 \end{pmatrix} &= \begin{pmatrix} T_{k(1)x} & T_{k(2)x} & T_{k(3)x} \\ T_{k(1)y} & T_{k(2)y} & T_{k(3)y} \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} S_{k(1)x} & S_{k(2)x} & S_{k(3)x} \\ S_{k(1)y} & S_{k(2)y} & S_{k(3)y} \\ 1 & 1 & 1 \end{pmatrix}^{-1} \\
\begin{pmatrix} \multicolumn{2}{c}{B_k(t)} & \begin{matrix}\alpha \\ \beta\end{matrix} \\ 0 \quad 0 & & 1 \end{pmatrix} &= \begin{pmatrix} v_{k(1)x} & v_{k(2)x} & v_{k(3)x} \\ v_{k(1)y} & v_{k(2)y} & v_{k(3)y} \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} S_{k(1)x} & S_{k(2)x} & S_{k(3)x} \\ S_{k(1)y} & S_{k(2)y} & S_{k(3)y} \\ 1 & 1 & 1 \end{pmatrix}^{-1}.
\end{aligned}$$

And we assume $A_k(t) = A_k^P(t)$ or $A_k^E(t)$.
$||B_k - A_k(t)||_F^2$ can be denoted by

$$\begin{aligned}
||B_k - A_k(t)||_F^2 &= Tr(B_k - A_k(t))^t (B_k - A_k(t)) \\
&= \mathbf{v}_k^t Q_k(t) \mathbf{v}_k + L_k(t) \mathbf{v}_k \\
&= \mathbf{v}^t Q'(t) \mathbf{v} + L'(t) \mathbf{v}
\end{aligned}$$

using a symmetric 4×4 matrix $Q_k(t)$ 4 vector $L_k(t)$. We also denote it using a symmetric $2n \times 2n$ matrix Q'_k and a $2n$ vector L'_k.

A constraint function $C(\mathbf{v})$ is defined by $C(\mathbf{v}) := \mathbf{v}^t C_M \mathbf{v} + C_V \mathbf{v}$ using C_M be a $2n \times 2n$ symmetric matrix, and C_V a $2n$ vector.
So we can define an energy function

$$\begin{aligned}
E_F &= \sum_{k\in\Delta} ||B_k - A_k(t)||_F^2 + C(\mathbf{v}) \\
&= \sum_{k\in\Delta} (\mathbf{v}_k^t Q_k(t)\mathbf{v}_k + L_k(t)\mathbf{v}_k) + \mathbf{v}^t C_M \mathbf{v} + C_V^t \mathbf{v} \\
&= \sum_{k\in\Delta} (\mathbf{v}^t Q'_k(t)\mathbf{v} + L'_k(t)\mathbf{v}) + \mathbf{v}^t C_M \mathbf{v} + C_V^t \mathbf{v} \\
&= \mathbf{v}^t (\sum_{k\in\Delta} Q'_k(t) + C_M)\mathbf{v} + (\sum_{k\in\Delta} L'_k(t + C_V^t)\mathbf{v} \\
&= \mathbf{v}^t (Q'(t) + C_M)\mathbf{v} + (L'(t) + C_V^t)\mathbf{v}
\end{aligned}$$

where $Q'(t) = \sum_{k\in\Delta} Q'_k(t)$ and $L'(t) = \sum_{k\in\Delta} L'_k(t)$.

Proposition 4. *For any $s,r \in \mathbb{R}$,we have $Q'(s) = Q'(t)$ and $C_M(s) = C_M(t)$.*

We note E_F is minimum if $\mathbf{v} = -\frac{1}{2}(Q' + C_M)^{-1}(L' + C_V)$.Using this $\mathbf{v}$, we can construct $B_k (k \in \Delta)$ which minimize E_F. 何か嬉しいこと.

Q' は t によらないので,$(Q' + C_M)^{-1}$ は一回だけ計算すれば良く, 高速計算が可能.

In our Mathematica library we can compute Q' using the function QuadraticFormEnergy,
(C_M by ConstMatrix,and C_V by ConstVector).

By Proposition **??**, $\min_{s,r\in\mathbb{R}} ||sR_\delta A_k(t) - B_k||_F^2$ can be also denoted by

$$\begin{aligned}
\min_{s,\delta\in\mathbb{R}} ||sR_\delta A_k(t) - B_k||_F^2 &= ||B||_F^2 - \frac{||B \cdot A^T||_F^2 + 2\det(B \cdot A^T)}{||A||_F^2} \\
&= \mathbf{v}_k^t U_k(t)\mathbf{v}_k \\
&= \mathbf{v}^t U'_k(t)\mathbf{v}
\end{aligned}$$

using a symmetric 4×4 matrix $U_k(t)$ 4 vector L_k.We also denote it using a symmetric $2n \times 2n$ matrix $U'_k(t)$.

And we have

$$\begin{aligned}
E_S &= \sum_{k\in\Delta} \min_{s,\delta\in\mathbb{R}} ||sR_\delta A_k(t) - B_k||_F^2 + C(\mathbf{v}) \\
&= \sum_{k\in\Delta} \mathbf{v}_k^t U_k(t)\mathbf{v}_k + (\mathbf{v}^t C_M \mathbf{v} + C_V^t \mathbf{v}) \\
&= \mathbf{v}^t (U'(t) + C_M)\mathbf{v} + C_V^t \mathbf{v}
\end{aligned}$$

where $U'(t) = \sum_{k\in\Delta} U_k(t)$.
For E_S, a Mathematica function for U',C_M,and C_V are

Proposition 5. *There exist $s,r \in \mathbb{R}$, such that $U'(s) \neq U'(r)$.*

SimEnergy は time-independent ではないけど、t も含めて計算することができる関数がある。

[2016.10.17 追記]
$p \in \mathbb{R}^2$ point
$p = \{p_1, \cdots, p_m\} \in (\mathbb{R}^2)^*$: points
$t \in \mathbb{N}$: Triangle Index
$T = \{t_1, \cdots, t_n\} \in (\mathbb{N}^3)^*$:Triangle Indexes
$\{p, t\} \in (\mathbb{R}^2)^* \times \mathbb{N}^3$: Polygon
Note $t = \{t_1, t_2, t_3\}, p = \{p_1, p_2, p_3\}$ のとき、(保留).
$\{p, q, r\} \in (\mathbb{R}^2)^* \times (\mathbb{R}^2)^* \times (\mathbb{N}^3)^*$: Configuration

Triangle :$(\mathbb{R}^2)^3 \to$ Triangles :$((\mathbb{R}^2)^3)^*$
Polygon $\to$ Triangles
$(p, q, T) \to (t_1, \cdots, t_n)$

Triangle $\times$ Triangle $\to$ Affine Matrix
$((p_1, p_2, p_3), (q_1, q_2, q_3)) \to \begin{pmatrix} a_{11} & a_{12} & b_1 \\ a_{21} & a_{22} & b_2 \\ 0 & 0 & 1 \end{pmatrix}$
Triangle $\to$ Triangle :重心を 0 にする.
Local Interpolation:Triangle $\times$ Triangle $\times \mathbb{R} \to$ Matrix
(Linear,Alexa,Log-Exp...)
Local Interpolations : (Local の関数名) $\times$ configuration$\to$ ($\mathbb{R} \to$ Matrices)
Grobal Interpolation : $(Local) \times (Const) \times$ configuration $\to$ ($\mathbb{R} \to$ Polygon)