

アフィン写像を用いた補間による2次元アニメーション作成ソフトウェア

松下, 昂平
九州大学大学院数理学府

<https://hdl.handle.net/2324/1434351>

出版情報 : COE Lecture Note. 48, pp.71-81, 2013-03-28. 九州大学マス・フォア・インダストリ研究所
バージョン :
権利関係 :



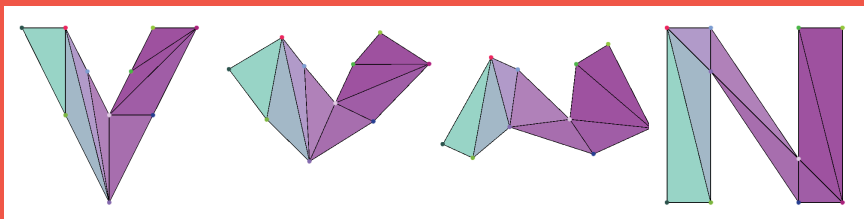
アフィン写像を用いた補間による 2次元アニメーション作成ソフトウェア

松下昂平（九州大学大学院数理学府）

「数学ソフトウェア紹介」 in Hakata Workshop 2013

概要

- モーフィングと呼ばれる2次元アニメーションの自動生成において, ARAP と呼ばれるアフィン写像を用いた補間手法が存在する.
- ARAP はオブジェクトの形状をなるべく保つことを目的とする補間手法である.
- ARAP を用いた補間手法を解析したり, 実際に生成されたアニメーション結果を確認するために, Python 言語を用いてソフトウェアを開発した.



イントロダクション

モーフィング

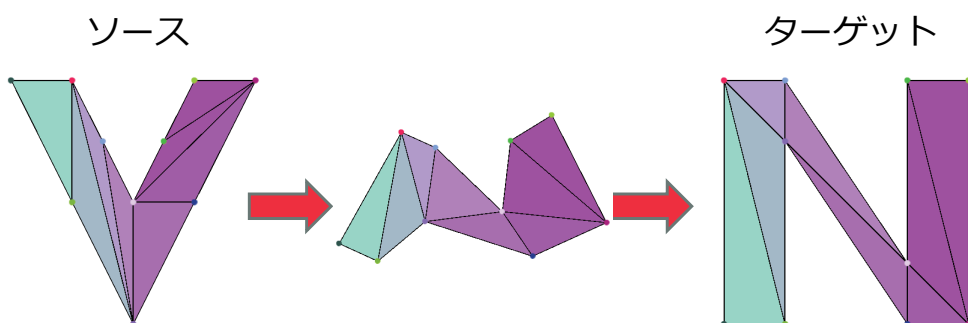
モーフィング

- キーフレームと呼ばれる少数のデータから，フレーム間を滑らかに補間してアニメーションを自動生成するCG技術.
- 特殊な映像効果を生み出したり，フレーム間の補間作業量の削減に有用.



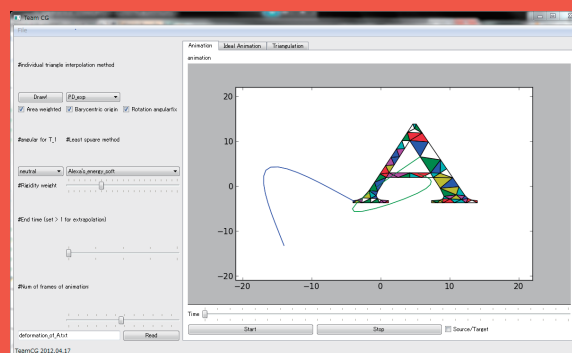
出典 : Alexa 2000 p.157

モーフィング例



ソフトウェア概要

- アルゴリズム
- 実装した機能



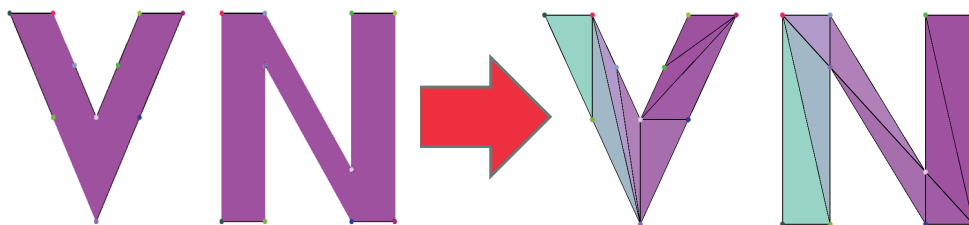
アルゴリズム

- **As-rigid-as-possible shape interpolation (ARAP)**

- 2000年に Alexa らによって始められる.
 - オブジェクトの幾何をなるべく保つような変形を構成する.
 - キャラクターアニメーションに適している.
- 具体的には, 以下の3ステップに分けたアルゴリズムで補間を実現する枠組み.
1. 両立三角形分割
 2. 局所変形
 3. 大域変形

両立三角形分割

- 入力された2つの多角形を, 組合せ同型になるように三角形分割する.
- 図形の特徴を崩すことなく補間を作成するために“良い”三角形分割を与える.



- 本ソフトウェアでは, 既存のアルゴリズムで両立三角形分割されたデータを用いる.

局所変形

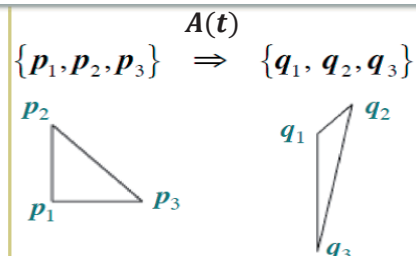
与えられた三角形の頂点の対応からアフィン変換 A がただ一つ定まる.

定義 1. 与えられた 2 つの三角形 $P = \{p_1, p_2, p_3\}$, $Q = \{q_1, q_2, q_3\}$ に対して,
 $\sum_{i=1}^3 p_i = \sum_{i=1}^3 q_i = 0$ と仮定する. このとき, P から Q への変換行列を A としたとき,

$$A(0)p_i = p_i, \quad A(1)p_i = Ap_i = q_i, \quad i = 1, 2, 3$$

を満たすような時間 $t \in [0, 1]$ でパラメータづけされた変換の族 $A(t)$ を P から Q への変形と呼ぶ.

$A(t)$ 内の t を $0 \leq t \leq 1$ で連続的に変化させることで,
 P から Q への滑らかな変形を生成できる.



$A(t)$ の構成法

1. 線形補間

$$A(t) = (1 - t)I + tA$$

2. 特異値分解を用いた補間

$$A(t) = R_{t\alpha}((1 - t)I + tD)R_{t\beta}$$

3. 極分解線形補間 [Alexa 2000]

$$A(t) = R_{t\gamma}((1 - t)I + tS)$$

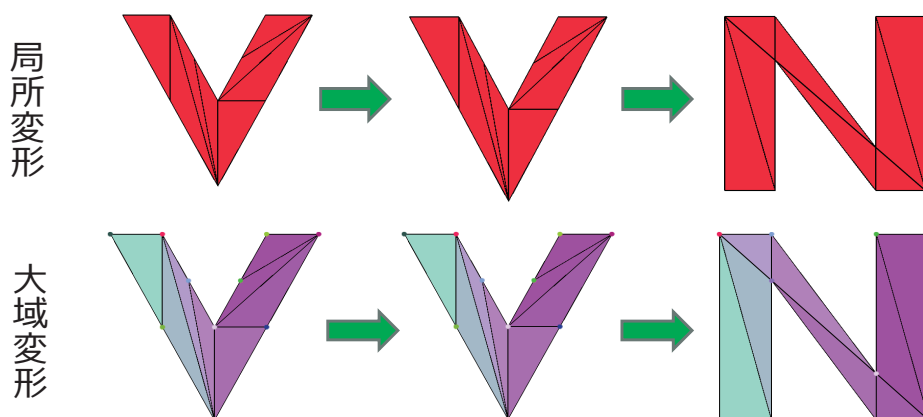
4. 極分解指数補間 [Kaji 2012]

$$A(t) = R_{t\gamma}S^t$$

今回開発したソフトウェアには上の4つの局所変形を選択できるように実装している.

大域変形

- 局所変形を元に隣接する三角形が分離したり，重なり合わないように，エネルギー関数を導入して大域変形を構成する．



エネルギー関数

エネルギー関数 $E(t)$ を次のように定義する．

$$E(t) := \sum_{T_i} \|A_{T_i}(t) - B_{T_i}(t)\|_F^2$$

ここで， $A_{T_i}(t)$ は三角形 T_i に関する局所変形， $\|\cdot\|_F^2$ はフロベニウスノルムである．

- $B_{T_i}(t)$ は三角形 T_i のアフィン変換の中で境界点を共有するアフィン変換であるもの．
- このエネルギー関数を最小にするような $B_{T_i}(t)$ を大域変形とする．
- この場合は正定値な2次関数であるので，線形方程式を解くことにより求めることができる．

エネルギー関数の構成法

1. フロベニウスノルム（変換距離） [Alexa 2000]

$$E(t) = \sum_{T_i} \|A_{T_i}(t) - B_{T_i}(t)\|_F^2$$

2. 点距離

$$E(t) = \sum_{T_i} \sum_{j=1,2,3} \|A_{T_i}(t)p_{i,j} - B_{T_i}(t)p_{i,j}\|_F^2$$

3. 相似・回転不変ノルム（変換距離） [Kaji 2012]

$$E(t) = \sum_{T_i} \left(\min_{s_i, \delta_i \in \mathbb{R}} \|s_i R_{\delta_i} A_{T_i}(t) - B_{T_i}(t)\|_F^2 \right)$$

4. 相似・回転不変ノルム（点距離） [Igarashi 2005]

$$E(t) = \sum_{T_i} \left(\min_{s_i, \delta_i \in \mathbb{R}} \sum_{j=1,2,3} \|s_i R_{\delta_i} A_{T_i}(t)p_{i,j} - B_{T_i}(t)p_{i,j}\|_F^2 \right)$$

今回開発したソフトウェアには上のエネルギー関数を選択できるように実装している。

最小となるエネルギー関数の求め方1/2

- エネルギー関数 $E(t)$ が正定値な2次関数であるとき、次のように書くことができる。

$$E(t) = v(t)^T G(t) v(t) + v(t)^T u(t) + c$$

ただし,

- $v(t)$ はオブジェクトの各頂点の x 座標と y 座標を一列に並べたサイズ $2n$ の列ベクトル。
- $G(t)$ は $2n \times 2n$ の対称行列。
- $u(t)$ は サイズ $2n$ の列ベクトル。
- c は定数項
- 目的は、 $E(t)$ が最小となる $v(t)$ を求めることである。

最小となるエネルギー関数の求め方2/2

- エネルギー関数 $E(t)$ の $v(t)$ に関する偏微分 $\frac{\partial}{\partial v(t)} E(t)$ の値が 0 になるときが最小の場合となる.
- 偏微分 $\frac{\partial}{\partial v(t)} E(t)$ は次のようになる.

$$\frac{\partial}{\partial v(t)} E(t) = 2G(t)v(t) + u(t)$$

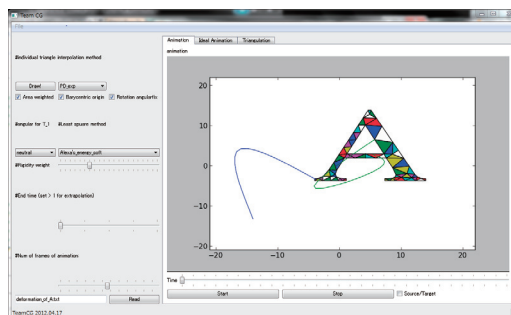
- これより, 頂点に関する列ベクトル $v(t)$ が

$$v(t) = \frac{1}{2} G(t)^{-1} u(t)$$

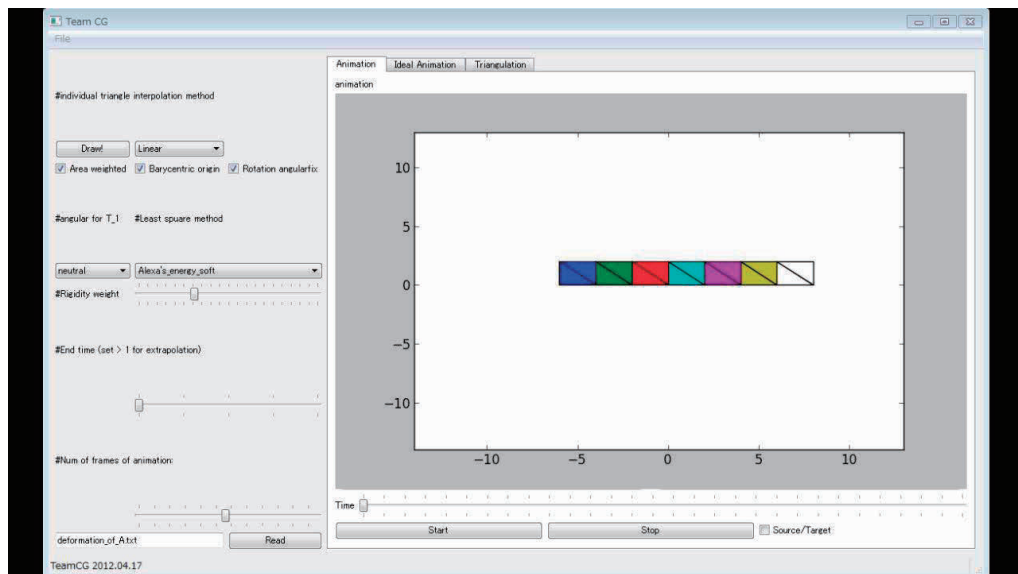
のとき, $E(T)$ は最小となる.

ソフトウェアの機能

- ファイルからデータを読み込み, アニメーションを作成する.
- 局所変形の補間方法の変更
- 大域変形におけるエネルギー関数の変更
- 回転の調整, エネルギー関数のウェイトの変更



ソフトウェアのデモ



まとめ今後の課題

- ARAP を用いたアニメーション作成ソフトウェアを Python 言語で開発した.
- 既存の補間手法の特徴を解析するために, GUI 上で各パラメータを逐次変更できるように実装した.
 - 局所変形
 - 線形補間, 特異値分解線形補間, 極分解線形補間, 極分解指数補間
 - 大域変形 (エネルギー関数)
 - フロベニウスノルム, 点距離ノルム, 相似・回転不変ノルム (変換距離, 点距離)
- 同じ入力データから異なった補間手法による結果を比較することができる.

今後の課題

- プログラムの高速化
- 3次元上のアニメーションへの拡張
- Maya ソフトウェア (3D アニメーションソフトウェア) のプラグインとして組み込むことで, 容易な補間アニメーションが作成できる環境を構築する.

共同開発者

- 井慶喜 (九州大学理学部)
- 池田有希 (九州大学大学院数理学府)
- 松田元輝 (九州大学大学院数理学府)
- 濱田裕康 (九州大学大学院数理学研究院)
- 溝口佳寛 (九州大学マス・フォア・インダストリ研究所)

参考文献

1. M. Alexa, D. Cohen-Or, and D. Levin, As-rigid-as-possible shape interpolation, Proceedings of ACM SIGGRAPH 2000, pp.157-164, 2000.
2. S. Kaji, S. Hirose, S. Sakata, Y. Mizoguchi, and K. Anjyo, Mathematical Analysis on Affine Maps for 2D Shape Interpolation, Proceedings of ACM SIGGRAPH 2012 (SCA2012) , pp.71-76, 2012.
<http://dl.acm.org/citation.cfm?id=2422368>
3. T. Igarashi, T. Moscovich, and J.F. Hughes, As-rigid-as-possible shape manipulation, ACM Transactions on Graphics 2005, pp.1134-1141, 2005.