1ZA-04

立体キャンバスを用いた線画キーフレーム間での対応線決定手法

藤田正樹 † 齋藤豪 †
† 東京工業大学 情報理工学院

1 はじめに

アニメーション自動補間には、ユーザの入力を減らしつつ意図通りのアニメーションを作り出すことが重要である。我々は先に立体キャンバスの概念を導入し、手描きによる作画で３次元的な奥行きのある動きを補間する手法を提案したが、線画間の線の対応付けは手動での入力を行う必要があった [1]。

本稿では立体キャンバスの作成に必須の線画間の対応付けを自動的に行う手法を提案する。これによりユーザの入力量を大きく減らすことができる。既存の対応付け手法 [2] と比較して、本手法では立体キャンバスを用いて３次元共通空間内での比較が可能である。また、提案手法は同じ線数のキーフレームに限定せず、一対多数の線の対応付けも行うことができる。

2 提案手法

本手法は２枚の線画 I^0, I^1 間の線の対応付け問題を解く対応付け手法である。

先に提案した手法 [1] では立体キャンバスの形状を求めるための線の対応付けが手動であったが、本手法はそれらを自動化する。立体キャンバスの形状に適当な初期値を与え、自動対応付けと立体キャンバスの形状推定を反復することで最終的な結果を得る。

2.1 距離関数

線画間の対応付けのため、まず線同士の距離関数 $d(C_p^0, C_q^1)$ を定義する。各線 $C_i^n \in I^n$ は入力と同時に立体キャンバス表面の３次元空間上に投影されており、m 点で離散化された３次元点列 $C_i^n = \{\boldsymbol{c}_{ij}^n \in \mathbb{R}^3, 0 \leq j < m\}$ である。

$$d(C_p^0, C_q^1) = d_{trans}(C_p^0, C_q^1) + \lambda d_{similar}(C_p^0, C_q^1) \quad (1)$$

λ は重みであり、距離 d は線間の移動量成分と類似度成分からなる。移動量成分 d_{trans} は動的時間伸縮法によって求める。類似度成分 $d_{similar}$ は他の曲線との位置関係による類似度を表しており、以下のように求める。

$$\begin{aligned} d_{similar}(C_p^0, C_q^1) = \min(& \\ & \sum_{\substack{C_k^0 \in I_0 \\ C_k^0 \neq C_p^0}} \min_{\substack{C_l^1 \in I_1 \\ C_l^1 \neq C_q^1}} d_{simp}(local(C_k^0, C_p^0), local(C_l^1, C_q^1)), \\ & \sum_{\substack{C_l^1 \in I_1 \\ C_l^1 \neq C_q^1}} \min_{\substack{C_k^0 \in I_0 \\ C_k^0 \neq C_p^0}} d_{simp}(local(C_k^0, C_p^0), local(C_l^1, C_q^1))) \end{aligned} \quad (2)$$

ただし、

$$\begin{aligned} local(C_a, C_b) &= \{\boldsymbol{c}_{aj} - mean(C_b) | 0 \leq j < m\} \\ d_{simp}(C_a, C_b) &= \sum_{0 \leq i < m} |\boldsymbol{c}_{ai} - \boldsymbol{c}_{bi}| \end{aligned} \quad (3)$$

である。

$local(C_a, C_b)$ は C_a を C_b の重心を基準としたローカル座標で表したものであり、$d_{simp}(C_a, C_b)$ は簡易的に求めた C_a, C_b の距離である。$d_{similar}$ は $C_{i_0}^0, C_{i_1}^1$ 以外の線について、ローカル空間内で比較を行い最小距離を類似度として足し合わせることを表している。

2.2 組合せ最適化

前項で任意の曲線間の距離が定義されたことにより、キーフレーム同士の対応付けは線を節点、距離を重みつきの枝とみることで完全２部グラフのマッチング問題に帰着することができる。しかしここでは通常のマッチングの定義とは違う、節点の重複を許したマッチングを求めたい。そのための前段階として、得られたグラフから各点にとって最も重みの小さい枝以外を削除することで小さな部分グラフの集合へ分割し、線の集合を「類似した線同士のグループの集合」にグループ分けを行う。

分割されたそれぞれの部分グラフについて、ハンガリアン法 [3] を用いて通常の重み最小となるマッチングを求める。各部分グラフについて少ない側の節点数に等しい数の対応関係が得られるので、余った節点はそれぞれ最もコストの少ない節点に接続する。これによって各部分グラフについて節点の重複を許可しつつ、高速にコストの十分少ない完全マッチングが得られる。

3 結果

提案手法は立体キャンバスを用い、共通座標空間で距離計算を行うことと重複した線の対応付けを扱うことが特徴である。それらの有効性を示すために、本手

Algorithm for Line Matching between Keyframes with Self-shaped Canvas
† MASAKI Fujita
† Suguru SAITO
School of Computing, Tokyo Institute of Technology (†)

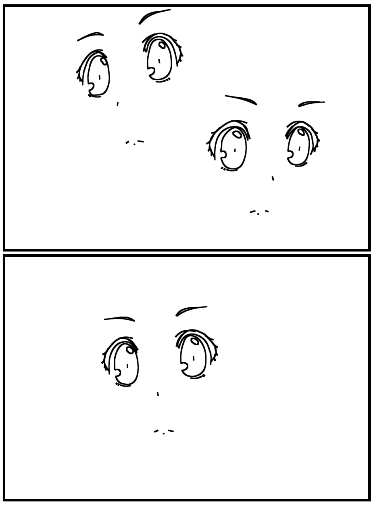

図 1: 入力画像および手動による補間結果画像

手法	再現率 (本数)	適合率 (本数)
立体キャンバスなし	0.24 (35/144)	0.21 (35/168)
対応関係の重複なし	0.72 (103/144)	0.77 (103/134)
提案手法	0.82 (118/144)	0.79 (118/150)

表 1: 手法の精度比較
再現率：推定データ中の正解数/正解データの総数
適合率：推定データ中の正解数/推定データの総数

法にて立体キャンバスを使わずに画面上の平面空間で距離計算した場合、重複を許可せずに線の対応付けを行った場合で結果に違いがみられるか比較する。

図１上の２枚の入力画像に手動で対応付けを行い作成した正解となる中間結果が図１下である。それに対して図２は立体キャンバスを使用しない場合の結果である。誤った対応付けが行われた箇所は赤線で示しており、補間結果が大きく崩れているのがわかる。図３は重複する対応関係を扱わない場合である。この場合では誤りが連鎖的に生じることによって主に目の下部に正解の場合とは多くの違いが生じている。

一方、図４に示す提案手法による場合はほぼ正解に近い結果が得られている。正解と比較するといくつか誤りは生じているが、線の対応付けが局所的に行われることによって近い範囲の誤りに抑えられている。

数値比較のため放映アニメーションを複数キャプチャしキャラクターの顔・髪・服などをトレースして、手動での対応付けを正解として再現率 (recall) と適合率 (precision) を算出、表にしたものが表１である。立体キャンバスの有無によって結果に大きな向上がみられ、重複を扱えるようにした点も再現率の上昇につながっている。

図 2: 画面上の座標を用いた補間画像

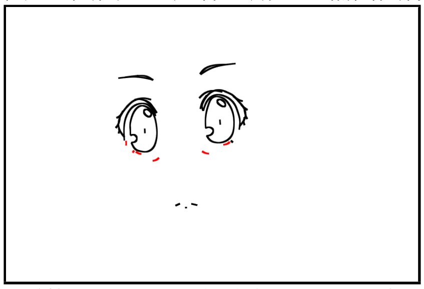

図 3: 重複を認めない対応付けによる中間画像

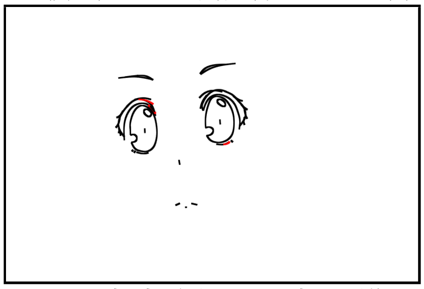

図 4: 提案手法による中間画像

4　まとめ

本稿では立体キャンバスを用いて線画間の対応付けを自動で行う手法を提案した。立体キャンバスを用いることで３次元的な動作を伴うアニメーションについての対応関係を効果的に推定することが可能となった。また、アニメーションにおける重複した対応関係も扱うことができた。これらによって手動の場合と比べて入力を効率化することが可能となった。

一方、対応付けの重複を想定しない方が良いような線画間では重複を認める対応付けは結果が悪くなる場合があり、組み合わせ手法の選択について検討の余地がある。また、本手法では対応しづらい輪郭線の扱いについても今後の課題となる。

参考文献

[1] Masaki Fujita and Suguru Saito. Hand-drawn animation with self-shaped canvas. In *ACM SIGGRAPH 2017 Posters*, p. 5. ACM, 2017.

[2] Alexander Kort. Computer aided inbetweening. In *Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering*, pp. 125–132. ACM, 2002.

[3] François Bourgeois and Jean-Claude Lassalle. An extension of the munkres algorithm for the assignment problem to rectangular matrices. *Commun. ACM*, Vol. 14, No. 12, pp. 802–804, December 1971.