輪郭形状の対応計算によるシェイプトゥイーン

高畠　綾子†　　野村　太一†　　藤吉　弘亘†

† 中部大学工学部　〒 487–8501 愛知県春日井市松本町 1200
E-mail: †{populi,nt,hf}@vision.cs.chubu.ac.jp

あらまし　本稿では異なる二つの輪郭形状の対応から中間画像を生成するシェイプトゥイーン法を提案する．連続 DP マッチングにより，二つの図形の輪郭線を変換した時系列信号同士の類似距離を計算した後に，バックトレースで得られる最適ルートが各輪郭線の対応となる．本手法は、従来のシェイプトゥイーンでは必要不可欠な図形の特徴点を選択することなく，複雑な輪郭形状を持つ図形同士の内挿形状を自動的に生成することができる．生成した中間輪郭形状の有効性をアンケートにより評価した結果，提案手法は複雑な形状においても望ましい中間形状を安定して生成出来ることを確認した．

キーワード　シェイプトゥイーン，輪郭線のフロー，連続 Dynamic Programming (DP) マッチング．

Shape Tweening by Contour Based Correspondence Computation

Ayako TAKABATAKE†, Taichi NOMURA†, and Hironobu FUJIYOSHI†

† Facurity of Engineering, Chubu University　Matsumoto-cho 1200, Kasugai city, Aichi, 487–8501 Japan
E-mail: †{populi,nt,hf}@vision.cs.chubu.ac.jp

Abstract　In this paper, we present a shape tweening technique, which is based on continuous Dynamic Programming (DP) matching. Continuous DP matching computes the similar distance between two signals calculated from contours of the shape, and then optimum contour correspondences are automatically obtained by Back Tracing. Therefore, an interpolated contour of complex curves can be obtained by the proposed method with no Shape Hint (hint of the shape). The experimental results and subjective evaluation show that our method is applicable to complex Bezier curves.

Key words　Shape Tweening, Contour Flow, Continuous Dynamic Programming (DP) matching.

1. はじめに

近年，アニメーションを作成する作業工程中最も手間のかかる原画作成を，モーフィング技法を用いて自動生成する研究が行われている [1], [2]．モーフィングによる中間画像の自動生成は，アニメーターが一枚一枚原画を描く負担の軽減に期待されている．また，アニメーション以外にメディカルビジュアライゼーション等の医療分野や CAD アニメーションにおいて，中間画像生成手法は応用されている．一般にアニメーションにおけるイラストは，ピクセルベースではなくベクトルデータを扱っており，本研究ではベクトルデータから得られる輪郭形状を対象としている．

輪郭線の中間形状を生成する手法として，Cong と Parvin は対象の輪郭と距離の変化からの内挿を提案している [3]．この手法では，対象に対して滑かな輪郭の内挿を行うことができるが，計算コスト多大であるため，アプリケーション等に用いるには実用的でない．

一方，インタラクティブな Web ページを作成する際に用いられる Flash(Macromedia,Inc.) では，輪郭のモーフィング手法としてシェイプトゥイーンと呼ばれる手法が用いられている．これは，ベクトルデータの輪郭形状に対してモーフィングを適用し，中間形状を出力することができる．しかし，シェイプトゥイーンの入力形状が人等の複雑な形状である場合，輪郭線同士の正確な対応が求められないため，ユーザが望む中間形状の生成が不可能となる場合が多い．そのため，Flash では補助機能として，ユーザが手動でシェイプヒントと呼ばれる輪郭線の特徴的な対応点を与えることで，ユーザの望む中間輪郭形状を生成することを可能としている．しかし，このような手動操作で特徴点を与えることは手間であり，また，ユーザが与える対応は必ずしも最適な補助の位置となるとは限らない．これらの問題を解決するには，全自動で正確な輪郭の対応を求め，中間形状を生成する必要がある．

本研究では，形状の特徴を表しているヒントを与えること無く輪郭形状の対応を求め，中間形状を生成する手法につい

て提案する．与えられたソース形状をターゲット形状に対して，連続 DP マッチング (Continuous Dynamic Programming matching) を適用し，バックトレースにより最適な対応を求める．得られた対応点から中間値を生成することで，全自動による輪郭のモーフィングが実現可能となることを示し，また，生成した中間形状を被験者による評価から，その有効性を示す．

2. Contour Flow

輪郭形状同士の対応を求めるために，各輪郭形状を 1 次元データへ変換し，連続 DP マッチングを用いて輪郭点の対応を求める．このとき，各輪郭線上にある点同士の対応を Contour Flow と呼び，中間形状生成の際に使用する．

2.1 輪郭の一次元化

2 次元である輪郭形状の特徴量を重心からの距離である 1 次元データに変換する [4]．対象の輪郭形状の重心点を (x_c, y_c)，I を輪郭点の数，(x_i, y_i) を i 番目の輪郭点座標とする．このとき，各輪郭点から重心までのユークリッド距離 $d(i)$ は次式より求められる．

$$d(i) = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} \tag{1}$$

図 1 に，1 次元化した二つの輪郭形状例を示す．モーフィングに使用するソースパターンの輪郭点を i，ターゲットパターンの輪郭点を j とする．

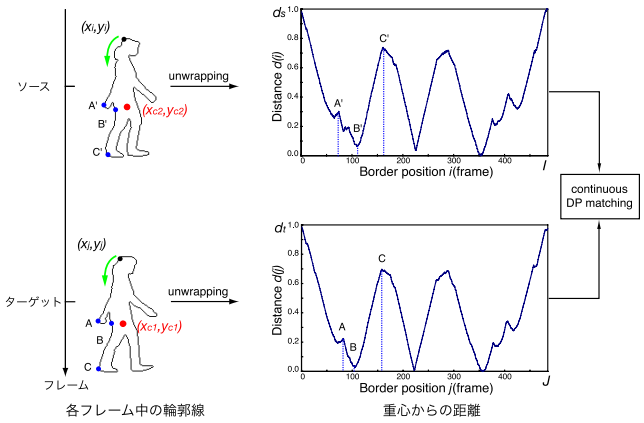

図 1　輪郭線情報の一次元化

2.2 連続 DP マッチング

DP マッチングは，異なる二つの時系列信号の類似度と最適な対応を求める手法であり，音声の認識 [5] やステレオマッチング [6]，カンターマッチング [7]- [11] など，様々な分野で利用されている．これは，非線形な時間正規化を用いたパターンマッチングである DP マッチングを利用することで，局所的な形状の違いを吸収し，かつ大局的に対応を求めることができるため多くの分野で利用されている．

端点固定の DP マッチングでは，一次元化した際の開始位置が同じでない場合，正しい対応を得ることが出来ない．また，対応させる各データをつなぎ合わせる手法を用いた場合，最も類似度が低くなる開始点を決定するための計算プロセスが必要となるため計算コストが余分にかかる等の問題がある．

そこで，本手法ではデータの始点を固定しない連続 DP マッチングを用い，さらに，任意パターン長データのみを 2 周分用意することで始点の対応位置に関する問題を解決する．輪郭特徴のデータが環状情報であり，始点 $d_s(0)$ と終点 $d_s(I)$ で繋がることを利用するのである．

以下に，連続 DP マッチングの適用法について示す．

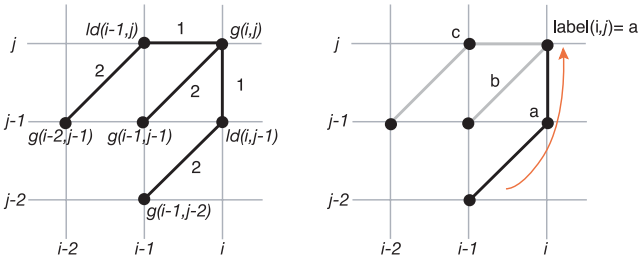

図 2　DP パス

2.3 初期設定

i を任意長パターンデータ (ソースデータ)，j を固定長パターンデータ (ターゲットデータ) としたとき，式 (2) に示すような初期値を設定する．

$$\begin{cases} g(i,0) & = 0 \quad (i = 0, 1, \ldots, 2I) \\ g(0,j) & = \infty \quad (j = 1, 2, \ldots, J) \end{cases} \tag{2}$$

ただし，I を任意パターン長データの輪郭周囲長，J を固定パターン長データの輪郭周囲長とする．$g(i,0)$ は，始端点フリーとするために 0 に初期化する．$g(0,j)$ は，固定パターン長データである d_t の周囲長未満のデータが最適な対応であると判断しないよう，予め大きな重みを与えて DP パスが選択されないように制限する．

2.4 DP 値の計算

連続 DP マッチングに用いる DP パスを図 2 に示す．各格子点 (i,j) における累積距離 $g(i,j)$ を次式より求める．

$$g(i,j) = \min \left\{ \begin{array}{ll} g(i-1,j-2) + 2 \cdot ld(i,j-1) & :(a) \\ g(i-1,j-1) + ld(i,j) & :(b) \\ g(i-2,j-1) + 2 \cdot ld(i-1,j) & :(c) \end{array} \right\} + ld(i,j). \tag{3}$$

式 (3) は，図 2(a) に示す対称的な 3 つのパスから最も輪郭特徴が似ているパスを選択する．このとき，選択されたパスに対応したラベル $(a), (b), (c)$ をバックトレースの際に用いるため保存しておく．

ローカルディスタンス $ld(i,j)$ は式 (4) より求める．輪郭特徴である $d_s(i), d_t(j)$ の差の 2 乗を算出するため，輪郭特徴が似ていないほど急激に値が大きくなり，パスとして選択されなくなる．

$$ld(i,j) = (d_s(i) - d_t(j))^2 \tag{4}$$

格子点 (i,j) までに選択された経路の長さを式 (5) より求める．

$$c(i,j) = \begin{cases} c(i-1,j-2)+3 & if(a) \\ c(i-1,j-1)+2 & if(b) \\ c(i-2,j-1)+3 & if(c) \end{cases} \quad (5)$$

以上の操作を各格子点にて行い，式 (6) より累積距離をそれまで辿った経路の長さで正規化する．

$$G(i,J) = \frac{g(i,J)}{c(i,J)} \quad (6)$$

2.5 バックトレース

DP マッチングの後，最適な対応を算出するためバックトレースを行う．式 (7) より求めた正規化距離 $G(i,J)$ の中で最も小さな値となる i' を開始点として選択する．

$$i' = argmin_{(J/2 \leqq i \leqq 2I)} G(i,J) \quad (7)$$

バックトレースの開始点は図 3 のように，傾斜制限を設けているため $(J/2 \leqq i \leqq 2I)$ 間で決定する．開始点が決定した後，最適ルートを式 (3) で記憶されたラベル a, b, c に従い各ノードをバックトレースする．このプロセスを終点 $(j=0)$ になるまで繰り返し行う．

バックトレースにより得られた最適ルートは，各輪郭線上の対応を示すことになる．

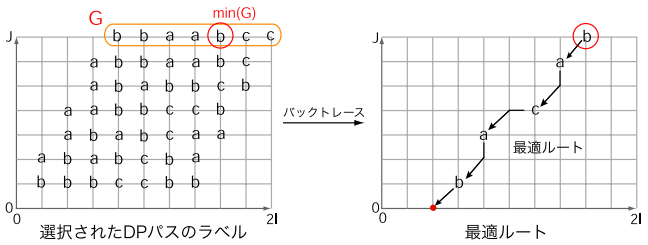

図 3　バックトレース

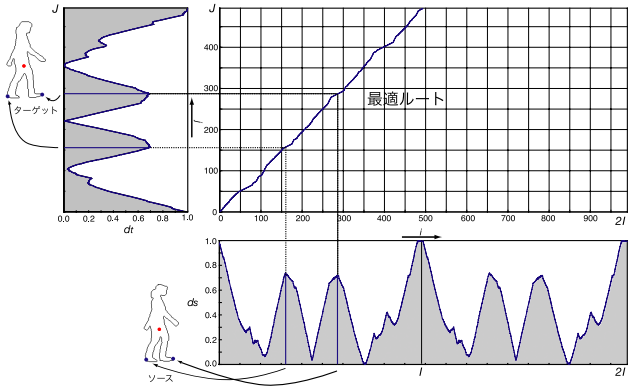

図 4　バックトレース適用例

図 4 に，バックトレースと算出された最適ルートにより対応付けされた輪郭線上のデータの一例を示す．図 4 より，最適ルートがソースとターゲットの各輪郭点の対応を表していることが分かる．

2.6 輪郭データの正規化

連続 DP マッチングで用いる DP パスの傾斜制限により，対応付けする一次元データの長さが 1/2 以下，もしくは 2 倍以上の場合，最適なルートを得ることが不可能となる．そこで，長さが 1/2 以下，もしくは 2 倍以上の場合は各データ長の最小公倍数にデータ長を正規化する．

3. Contour Flow 分析

3.1 Contour Flow の例

プリミティブな形状を平行移動，回転，拡大縮小，変形した際の対応から得られる Contour Flow を図 5 に示す．

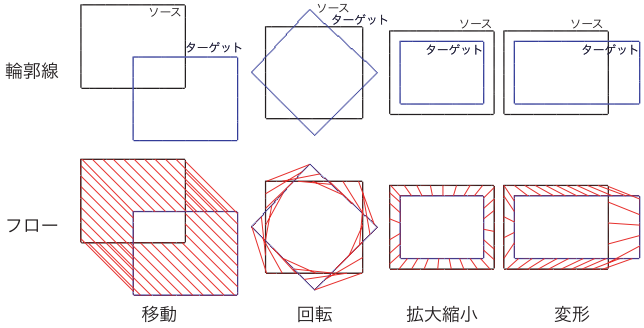

図 5　Contour Flow 表示例

図 5 より，各頂点が変形後の頂点と正しい対応を得られていることがわかる．このような，輪郭線のみからフローを求められる利点として，画像のテクスチャが必要でないこと，フローの計算に SAD 等の計算が必要でないため高速化が期待できる．

3.2 複雑な形状における Contour Flow

複雑な形状パターンとして，端点固定カメラに対して横方向に移動する歩行者と走行車のシーケンスを利用する．歩行者と走行車のビデオより選択された 20 シーケンスに対し，背景差分により移動体領域を検出し，その後得られた輪郭線より，Contour Flow を計算した．図 6 にその結果を示す．

歩行者の場合，腕や足が前後に動くため，図 6(a) に示すフローの方向ヒストグラムから，フローの方向は一定に定まらないことが分かる．一方の走行車の場合は，剛体であるゆえに一定方向のフローが観測された (図 6(b) 参照)．これらより，輪郭線上のフローは剛体 (自動車) と非剛体 (人間) の判別に用いる一つの特徴量として利用可能であるといえる．

表 1　フローの方向分布の分散.

クラス	歩行者	走行車
平均	3731.82	39.01

表 1 に，剛体・非剛体の各 10 シーケンスより求めた，フローの方向分布の分散を示す．これより，提案手法により得られるフローから剛体・非剛体を判別することが容易であることから，輪郭線上のフローが正確に求められていることがわかる．

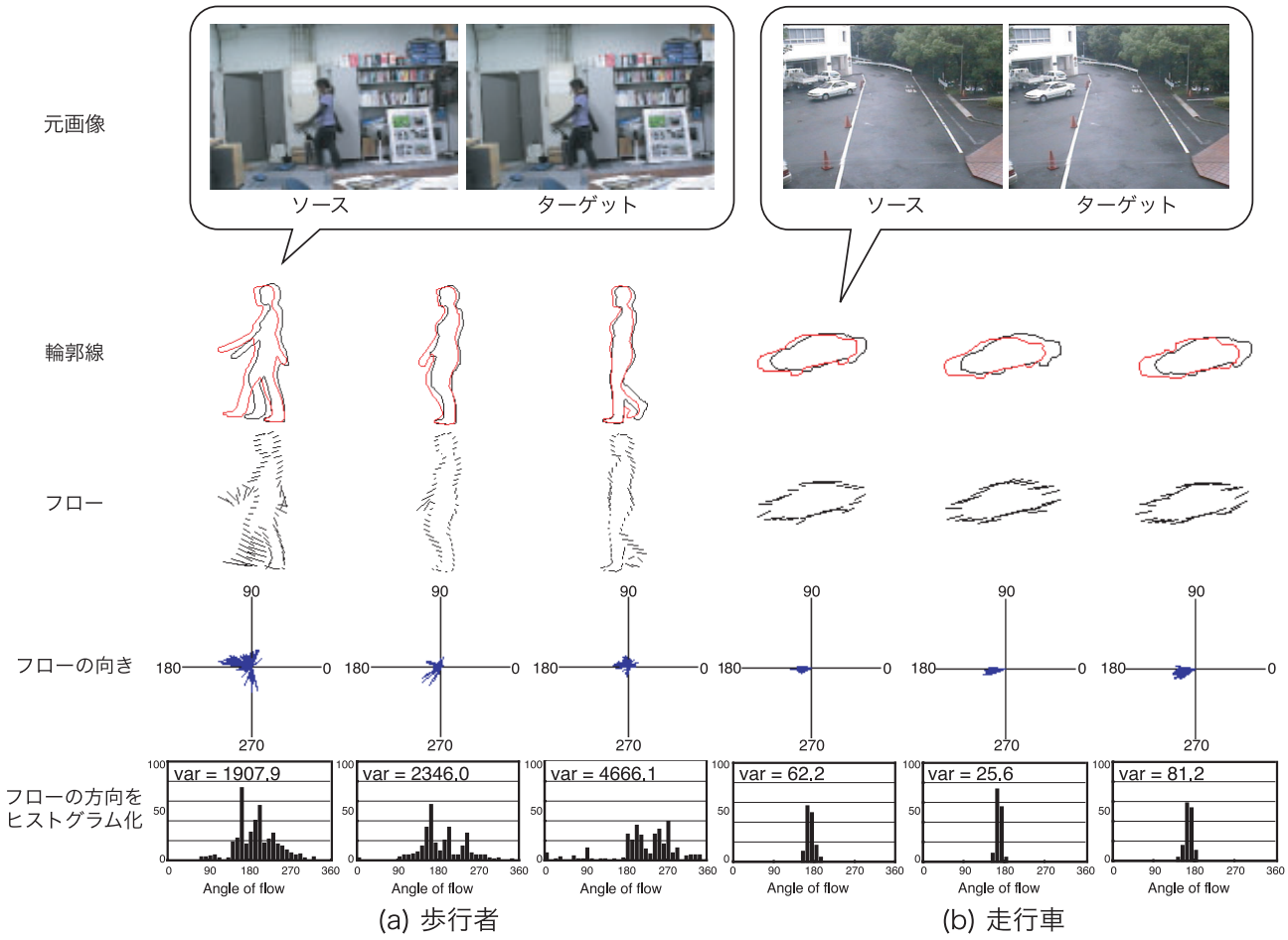

図 6　剛体・非剛体移動時のフロー表示例

4.　輪郭線の対応を基にするシェイプトゥイーン

図 7 にベクトルデータからなる輪郭線のシェイプトゥイーン手順を示す．まず初めに，ベジエ曲線で描かれたベクトルデータの輪郭線をラスタ化する．次に，形状の重心から各輪郭点までの距離である一次元信号に変換する．その後に連続 DP マッチングを用いて対応付けを行い，得られた輪郭線の対応から中間形状を生成する．最後に，中間形状データをベクトルデータに戻す．

本手法で使用した輪郭線形上のサンプルは，交差しない閉曲線を対象としている．また，アニメーションを作成することを考慮しているため，極端にフレーム数を進めた場合に相当する “変化の激しいモーフィング” は今回対象としていない．

4.1　ベクトルデータからラスタデータへの離散化

ベジエ曲線は多くの描画系アプリケーションに用いられている描画形式である．輪郭線を基にしたシェイプトゥイーンを行う初めのステップは，このベジエ曲線をラスタ化し，離散点を求める．ベジエ曲線は次式で表わされる．

$$\begin{aligned} x &= (1-t)^3x_1 + 3(1-t)^2tx_2 + 3(1-t)t^2x_3 + t^3x_4 \\ y &= (1-t)^3y_1 + 3(1-t)^2ty_2 + 3(1-t)t^2y_3 + t^3y_4 \quad (8) \\ &(0 \leqq t \leqq 1) \end{aligned}$$

(x_1, y_1) は始点，(x_2, y_2), (x_3, y_3) は制御点，そして (x_4, y_4) は終点である．ラスタ化されたベジエ曲線状の点 (x, y) は媒介変数 t により制御される．図 8 に示す様に，変数 t を変化させた場合，座標のサンプリング間隔が変化する．変数 t が 0.1 であ

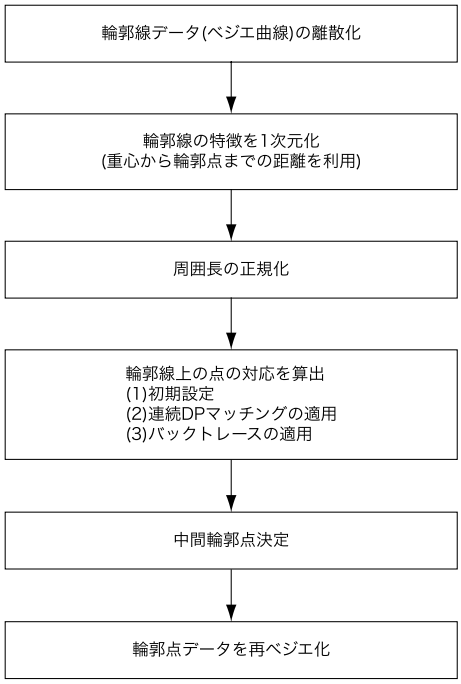

図 7　シェイプトゥイーンの流れ

る場合，ラスタ化されたデータは 10 個となる．本研究では，$t = 0.001$ を用いる．

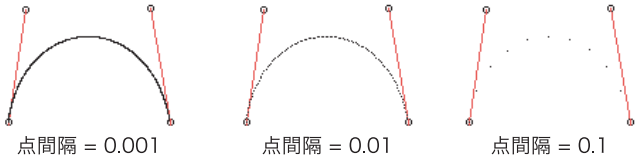

図 8　ベジェ曲線の離散化

4.2　中間形状の生成

2 章で示した提案手法を用いて，ソース形状とターゲット形状の輪郭線対応が決定された後，次式により内挿後の座標 (x', y') を求める．

$$\begin{aligned} x' &= (1-\alpha)x_s + \alpha x_t \\ y' &= (1-\alpha)y_s + \alpha y_t \\ 0 &< \alpha < 1 \end{aligned} \tag{9}$$

(x_s, y_s) はソース形状の輪郭，(x_t, y_t) はターゲット形状に対応した各座標である．α は内挿するスケールを決定するパラメータである．内挿点が決定した後に，他アプリケーションでの利用が可能となるよう，ベジエ曲線を用いてベクトル化を行う．

図 9 に α を 0.25, 0.50, 0.75 と変化させた際に，提案手法と Flash シェイプトゥイーンを用いて生成した中間形状の例を示す．Flash シェイプトゥイーンは，複雑な図形の場合ユーザからの特徴点を指定しなければユーザの望む形状に変化出来ない．このような特徴点は，適宜ユーザが試行錯誤してポイント数やその位置を決定していく必要がある．図 9 より，シェイプヒントが無い場合では足の中間形状がつま先まで正しく生成されていないことが見て取れる．また，円で示した足のつま先部分について比較すると，提案手法は人等の複雑な形状に対して，ソースとターゲット形状に含まれる特徴を残した中間形状をシェイプヒント無しに生成可能であることが分かる．

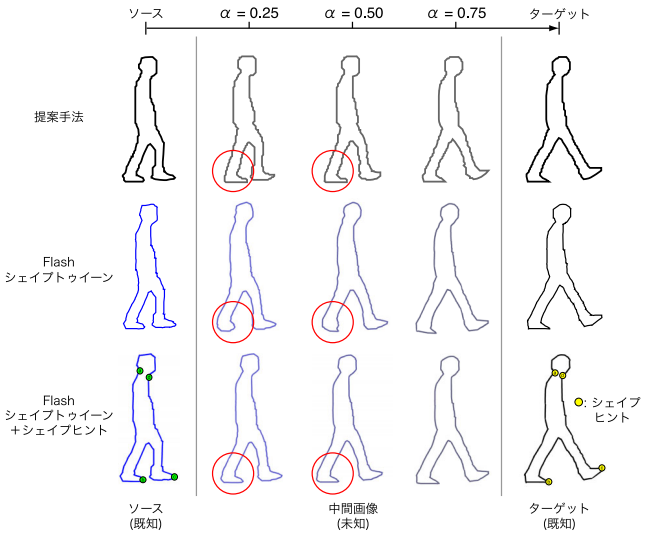

図 9　中間形状生成

5.　評 価 実 験

提案手法により自動生成された中間形状の有効性を，アンケートを利用した実験により評価する．

5.1　アンケートによる評価実験

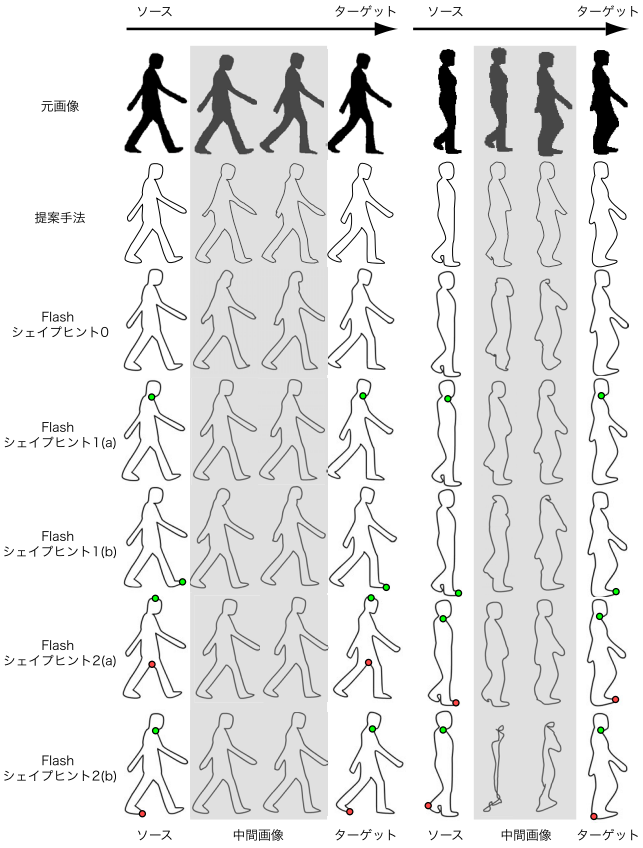

図 10　被験者へのアンケート例

歩行者の映像から連続する 4 枚のフレームを選択し，これらを用いて生成された中間形状の評価を行う．第 1 フレームをソースフレーム，第 4 フレームをターゲットフレームとして各手法により中間形状を生成する．残る第 2，第 3 フレームは，各手法により生成された中間形状と比較するための望ましい中間形状として使用する．比較実験として，Flash シェイプトゥイーンでシェイプヒントを用いない場合，用いた場合 (1 個，2 個)，そして提案手法の計 4 手法より得られた中間形状を対象とする．図 10 に実際に評価実験に用いた中間形状例を示す．これらの形状パターンを被験者に見てもらい，似ている，どちらでもない，似ていないを選択する．Flash シェイプトゥイーンでは，複雑な図形の内挿を行う際に適切な場所にシェイプヒントをおく必要がある．そこで，評価用データには予め調べた適切なポイントを用いた場合のみを対象としてデータに加えている．

5.2　評価の結果

被験者 50 人に対して評価実験を行った．被験者は各手法により生成された中間形状に対して 3 段階に評価 (類似する，どちらともいえない，類似しない) する．

アンケートより得られた 3 段階の評価結果を-1 から 1 までの数値 (-1:類似しない，0:どちらともいえない，1:類似する) に数

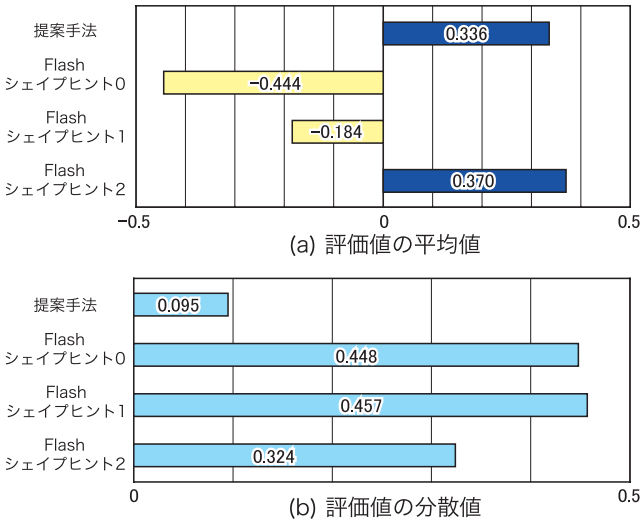

図 11　中間形状に対する評価結果

値化し，150 パターン (3 パターン ×50 人) の平均と分散を図 11 に示す．図 11(a) より，Flash シェイプトゥイーンの “シェイプヒント無し” と “シェイプヒント 1 個” の結果は負の評価であることがわかる．これは明かに似ていないと被験者が判断した事を意味する．Flash シェイプトゥイーンを複雑な形状に適応させる場合，数点のキーポイントを与える必要があるため，望ましい中間形状の生成が不可能であった．

一方，“Flash シェイプヒント 2 個” と “提案手法” では正の評価を得た．これは複数のシェイプヒントを形状の特徴となる正しいポイントに設定したため，Flash においても良い評価を得た．一方，提案手法も同様の評価を得ている．両者の評価値に対して t 検定を行った結果，有意水準を 5%としたとき，その有意差は無いと判定された．これより，提案するモーフィング手法は，既存のシェイプトゥイーンのようにシェイプヒントを与えることなく，それと同等の中間形状を生成できることがわかる．また，図 11(b) に示す評価結果の分散値を比較すると，Flash を利用した場合よりも提案手法の分散が小さいことがわかる．これは，提案手法が様々な形状に対しても，より滑らかで自然な中間形状を安定して生成可能であることを示している．

6. ま と め

本稿では，連続 DP マッチングを用いた輪郭形状のシェイプトゥイーン手法を提案した．輪郭線の対応付けは連続 DP マッチングにより自動的に算出されるため，既存のシェイプトゥイーン手法では必要とするユーザからの入力情報を，本手法では一切必要としないでユーザの望む中間形状を生成することが可能である．被験者による評価の結果，提案手法はより望ましい中間形状を出力できることを確認した．

文　献

[1] 白濱洋平，大野義夫，“多数決にもとづくアニメーションキャラクターのモーフィング手法，” 情報処理学会研究報告 Vol.2001, No.035, 2001–CG–103, Apr. 2001

[2] 爰島快行, 高橋成雄, 大渕竜太郎,“形状モーフィングにおける位相変化の制御,” 情報処理学会グラフィクスと CAD 研究会報告 2001-CG-102, pp. 61-66, Feb. 2001.

[3] G. Cong and B. parvin, “A New Regularized Approach for Contour Morphing ” *Computer Vision and Pattern Recognition*, Vlo.1,pp. 1458-1463, 2000.

[4] H. Fujiyoshi, A. J. Lipton, and T. Kanade, “Real-Time Human MotionAnalysis by Image Skeletonization” *IEICE Trans. Inf. & Syst.*, Vol.E87-D, No.1,pp. 113-120, 2004.

[5] H. Sakoe and S. Chiba, “Dynamic Programming Algorithm Optimization for Spoken Word Redognition” *IEE Trans. Acoust., Speech, and Signal Process.*, Vol.ASSP-26, pp. 43-49, 1978.

[6] Y. Ohta and T.Kanade, “Stereo by Two-Level Dynamic Programming” *Proc. of the Ninth International Joint Conference on Artificial Intelligence*, Vol. 2, pp. 1120-1126, 1985.

[7] D. Geiger, A. Gupta, L. A. Costa, and J. Vlontzos, “Dynamic Programming for Detecting, Tracking, and Matching Deformable Contours” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.17, No.3, pp. 294-295, 2004.

[8] I. Bartolini, P. Ciaccia, and M. Patella, “WARP: Accurate Retrieval of Shapes Using Phase of Fourier Descriptors and Time Warping Distance” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.27, No.1, pp. 142-147, 2005.

[9] E. G. M. Petrakis, A. Diplaros, and E. Milios, “Matching and Retrieval of Distorted and Occuluded Shape Using Dynamic Programming” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.24, No.11, pp. 1501-1516, 2002.

[10] T. W. Sederberg and E. Greenwood, “A Physically Based Approach to 2-D Shape Blending” *Computer Graphics*, Vol.26, No.2, pp. 25-34, 1992.

[11] E. Milios, and E. G. M. Petrakis, “Shape Retrieval Based on Dynamic Programming” *IEEE Transactions on Image Processing*, Vol.9, No.1, pp. 141-147, 2000.