トレーシングとデータベースを併用する 2D アニメーション作成支援システム

福里　司 *　　森島繁生 †

概要．　本研究では，ペンタブレット上で描いたイラストから中割画像を生成する処理と、既定の動作を合成する手法を組み合わせることで手描きアニメーションの作成を支援するユーザインターフェースを提案する．これまでに，3DCG モデルや煩雑な物理方程式を用いることでアニメーション制作を効率化する手法が数多く提案されているが，これらはあくまでも「現実に近い動きを再現する」ための解を導く方法に過ぎず、(1) 作者ごとの作品の特徴（手描き感）を再現することが難しく，(2)3DCG 等の専門的な知識を持たないユーザにとっては困難なタスクであるといった点が問題視されている．本研究ではこのような問題を解決するために，ユーザが描いたイラストを直接変形させることでアニメーションを生成する手法を提案する．アルゴリズムとしては,「パラパラ漫画」のように前フレームをトレースしながら描いた複数のイラストを滑らかに補間する中割り画像を生成する機能と，事前に用意したアニメーションデータをイラストに半自動的に付与する機能を導入した．

1　はじめに

近年,LINE の「動く」スタンプや Gif 画像の普及に伴い，アニメーション作品を観るだけに限らず，自ら制作するといった活動が一般的に見られる．その主な理由は，アニメーションは，絵日記やデッサンといった静止画とは異なり,「動き」によって感情や製作者の感性を直感的に伝えられることが挙げられる．しかし, パラパラ漫画の制作工程からわかるように，アニメーション制作は滑らかな動きを意識しつつ大量に絵を描く必要があるため，非常に手間のかかる作業であることが問題視されている．このような背景から，3DCG モデルや物理方程式を用いたデジタルアニメーション技術を用いたアニメーション制作の効率化を目指す研究が多数考案されているものの，これらは複雑な操作を要求する上に、作者の感性や意図 (こだわり) を自由に反映することが困難である．

そこで本研究では，ペンタブレット上で描いたイラストを直接動かすことで，効率的にアニメーションを生成するシステムを提案する (図 1)．提案手法では，ユーザはフレーム上にイラストを描くことで，フレームどうしを滑らかに補間する役割を持つ「中割り画像」を生成する．その際，従来のアニメーション制作で用いられる「トレーシング」を利用することで，従来の中割り画像生成で課題となる対応付けの手間を軽減している．さらに，中割りを生成しないように設定した箇所に対しても，事前に用意した動きのデータ (フェードアウトや振動など) を半自動的に付与する機能によって，パラパラ漫画のように前フレームを意識してイラストを描かずに，アニメーションを手軽に作成することもできる．

* 早稲田大学
† 早稲田大学理工学術院総合研究所

2　関連研究

手描きアニメーション制作を支援する手法として Nakajima ら [13] は，アナログ画材を用いて描いたイラストを簡易的な矩形で近似し、既定の動作ルール (例：平行移動，拡大縮小，回転) を付与する手法を考案した．しかし，一つのイラストを一つの単純な矩形として扱うため，イラストの各領域ごとに異なった動きやユーザが意図する結果を得ることが困難であり，様々なイラストの領域指定システムが考案されている [10]．特に LIVE2D[12] ではパーツ単位に分離されたイラスト上に手作業で二次元メッシュ構造を作成し，各メッシュ領域に動きルールを割り当てることでイラストを直接動かす機能を実現した．その一方，(1) 入力となるイラストをパーツ単位で作成する必要がある点（レイヤー構造が必要），(2) メッシュ構造を全て手作業で作成する必要がある点,(3) メッシュの変形には各頂点の線形補間を用いている点から，一つのアニメーションを制作するのに膨大な時間を要する．

一方，メッシュ構造を利用したアニメーション生成手法として，これまでに As-Rigid-As-Possible Mesh Manipulation と呼ばれる手法が多数提案されてきた [1][3][4][5][7][11]．これらの手法はメッシュを構成する最小単位である三角形に着目し，各三角形を変形させるための Affine 変換行列（ローカル変換）を基に全体的な形状変形や中割形状を生成するものである．しかし，これらの手法は三角形構造の作成の手間に加え，ユーザが描いたストロークのような線分単位（制御点 2 点間）に対する Affine 変換行列を算出することが困難である．各フレームに描かれたストロークを基に中割形状を求める手法と

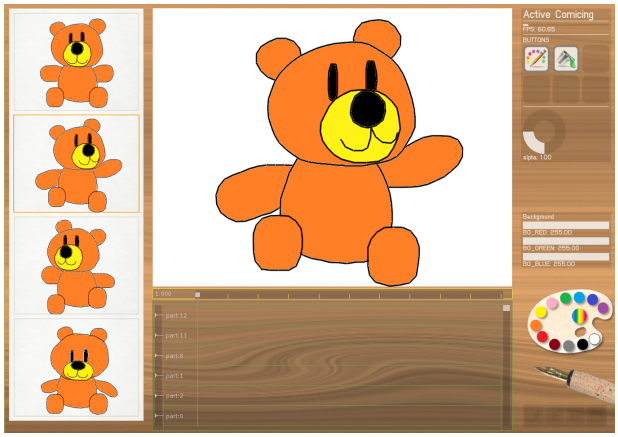

図 1. 提案システムのスクリーンショット.

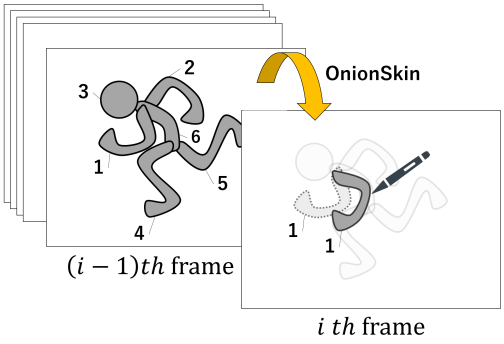

図 2. オニオンスキン機能の概要. ユーザが新たなイラストを描く際に，前フレームに描かれたイラストに付与されたラベリング番号を順次提示することで，次にどの部位を描くべきかを把握することができる.

して，Sederberg[16] はストロークを長さと角度情報と分離し，それらの情報を補間する中割生成手法を考案した．さらに，White ら [21] はストロークの制御点に基づく曲線フィッティングと接線ベースのワーピング手法を提案した．これらの手法によって，ペンタブレット上でユーザが描いたイラストをインタラクティブに中割り生成を実現した．しかし，キーフレーム間の形状（ストロークの描き方）が大きく異なる場合，生成した形状が大きく破綻する可能性がある．提案手法は, 各フレームに描いたストローク (入力) に対し，疑似的なメッシュ構造を自動生成することで，フレームとフレーム間を滑らかにつなぐ中割り画像を効率的に生成する点が大きく異なる．また, これまでにユーザが作成したアニメーションデータ (例：平行移動や回転) を活用する機能を用いることで, ユーザが新たに描いた単一フレーム上のストロークに動きを付与することができる.

3 ストロークアニメーション生成手法の概要

本インターフェースは, ユーザがイラストを描くための (1) ペンの色や太さを指定する機能，(2) 背景の色の指定, 画像の読み込みに加え, アニメーション制作で一般的に用いられる (3) オニオンスキン機能を有している．操作方法としては，ユーザは中央に表示されたメイン画面に自由にイラストを描く．左端にアニメーション生成で用いるフレームのコントロールが配置され，マウスクリックによってフレーム番号を指定できる．また, ユーザが描いたフレームを別のフレームにコピーする場合や, フレームの順番を入れ替える場合を想定し，ドラッグアンドドロップ機能を追加している．次にユーザが描いた各ストロークに対してラベリング番号を付与する（初期設定として, ストロークの描き順を基に番号を指定する）．オニオンスキン機能は, 前フレームで描いたイラストに付与されたラベリング番号と描いたストロークの形状を提示することで，ユーザが次フレームのイラストを作成する際にどの順番で描く必要があるかを確認することができる（図 2）．下記に述べるフレーム間のストローク形状に基づいた中割り形状を生成する際に必要となる対応関係の作成の手間を軽減することができる．下端にラベリング番号のコントロールが配置されており, ドラッグアンドドロップによって各ストロークに付与されたラベリング番号を変更することができる.

内部的には, ユーザによるストローク作成の結果は, 等間隔サンプリングを行うことで n 個の制御点を取得している．表示する際は，制御点間を Catmull-Rom Spline 曲線を用いることで補間を行っている．離散的な結果を編集する方法として, マウスクリックとドラッグによって，各制御点の位置やストローク全体の位置を移動させる機能も追加している.

3.1 各フレーム上のストローク形状を補間するための中割り生成手法

中割形状の生成に伴い，各フレームに描かれたイラストに付与されたラベリング番号を基にストロークどうしの対応付けを行う．ただし，対応付けを行ったストロークの制御点数が異なる場合，制御点数の正規化処理 (再サンプリング) を行った．次に, フレーム間のストローク形状の変化情報を基に Affine 変換行列 (回転，拡大・縮小情報) を算出する．従来手法として，Baxter ら [4] は入力画像の輪郭線 (ストローク) を基にドロネー三角形分割を用いた対応付け手法を考案した．しかし，ドロネー三角形分割を用いた場合，各キーフレーム上の輪郭形状が類似している必要があり，ユーザが描いた任意のストロークに対してロバストな対応関係を取得することが困難であった．そこで我々は，Sumner ら [18] や Umetani ら [20] が考案した 3DCG モデルの形状変形や制約条件として仮想頂点を配置する手法を参考に，ユー

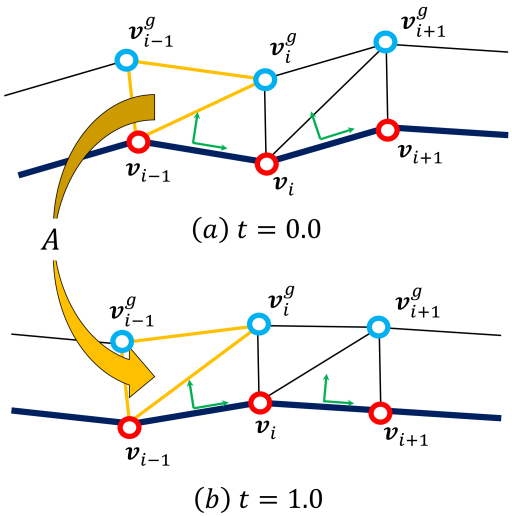

図 3. ストロークに対する疑似的な三角形構造の概要図．ストロークの制御点 (赤点) に対し仮想頂点 (青色) を配置することで，三角形メッシュ (橙色) 間の **Affine** 変換行列を計算する．

ザが描いたストロークの各制御点 $\vec{v_i}$ に対し，垂直方向に仮想的な頂点ベクトル $\vec{v'_i}$ を定義する．この頂点を用いることで，ストロークに疑似的な厚みを持たせた三角形メッシュ構造を生成する (R_{90} は 90° 回転行列を示す)．

$$\vec{v_i^g} = \vec{v_i} + R_{90}\left(\frac{\vec{v_{i+1}} - \vec{v_{i-1}}}{2}\right) \tag{1}$$

上記の手順によって対応付けられた各ストローク間の三角形構造における Affine 変換行列 A を取得することができる (図 3)．

この結果を基に特異値分解 (SVD) を用いた As-Rigid-As-Possible Mesh Interpolation 技術を利用し，各ストロークの中割り形状を実現した．具体的には，中割りパラメータ t $(0.0 \leq t \leq 1.0)$ に対し，以下の評価関数を最小化することで中割形状の最適化を行った．

$$A_i(t) = R_\theta^t \cdot \exp(t \log S) \tag{2}$$

$$E_i^R(A_i(t), B_i(t)) = \min_{s,\delta\in\mathbb{R}} \|sR_\delta A_i(t) - B_i(t)\|_F^2 \tag{3}$$

$A_i(t)$ は i 番目の三角形の Affine 行列 A に対し，中割処理を行った行列 (ローカル変換)，$B_i(t)$ は出力結果から得られる Affine 変換行列 (グローバル変換)，s はスケール値，R は回転行列，S は拡大縮小行列である．今回は，ストロークの太さ情報及び，色情報 (RGB) の補間に関しては線形補間を用いている．図 6 に Source$(t = 0.0)$ と Target$(t = 1.0)$ を基に生成した中割り形状結果を以下に示す．比較手法として本システムで対象とする「ストローク」に適用可

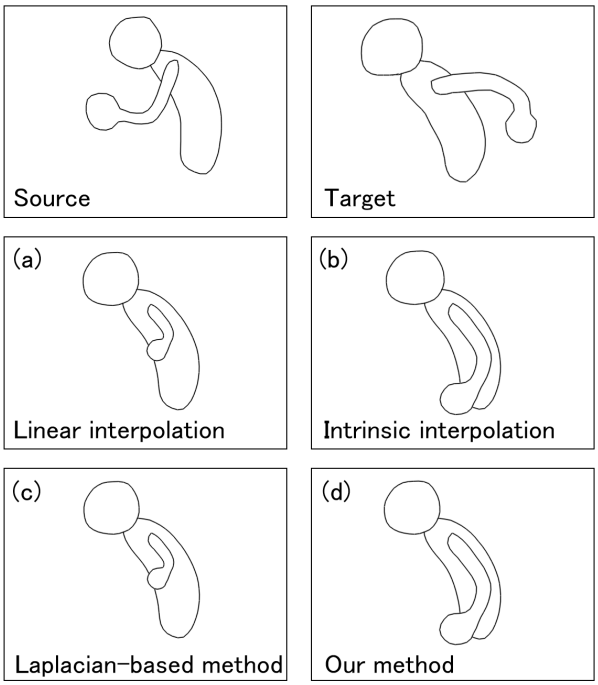

図 4. ストローク形状補間法の比較結果 ($t = 0.5$)．**(a)** 頂点単位の線形補間結果，**(b)Sederberg et al. 1993[16]**，**(c)Alexa 2003[2]**，**(d)** 提案手法．

能な手法を用いた．この結果から提案手法では，角度と長さ情報を用いた手法 (b) と同様に入力となるイラストにみられる形状特徴を破綻 (例：縮退) させることなく，自然な中割り形状を実現している．しかし，従来のストローク中割手法 (b) と異なり，本手法は角度や長さではなく、アフィン変換行列を算出し線形最小化問題として扱うことができるため，複数の制約条件（例：制御点間の位置関係や位置）を多数追加することができる．今回提案した仮想頂点を用いるアプローチは，三角形メッシュで構成された 3DCG モデルに適用することで，疑似的な四面体構造に基づく 3DCG モデルの形状変形及び中割り形状を簡単に生成できるメリットも持つ．本システムは下端に配置されたレイヤーコントロール部で，各ストロークのアニメーションのタイミングを調整し，中割パラメータ t を編集するタイムスライダ機能も追加している [1]．

3.2 データベースを用いたストロークアニメーション生成手法

次のステップは，各フレームに描かれた任意のストロークに対し既定の動きルールを付与する機能を検討する．この機能は Microsoft PowerPoint といった典型的なアニメーション機能を模したものとなっている．動きルールは平行移動，振動，回転，拡大

[1] 本稿の技術の詳細は [6] を参照されたい．

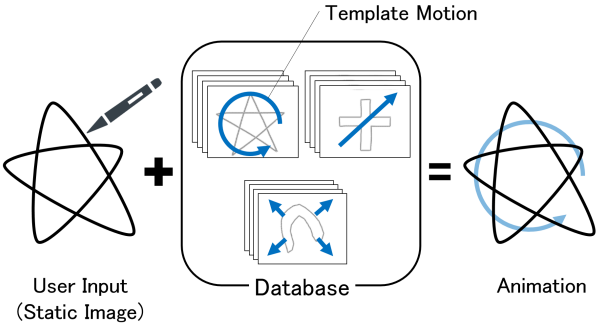

図 5. データドリブンなアニメーション生成の概要図. ユーザが描いたイラストと，データベース内のストロークとの類似度比較を行うことで，データベースのストロークにタグ付けされた動き情報を検索及び合成を行う.

縮小，フェードの 5 種類を用意し, 本システムの下端のタイムスライダに表示された各ストロークの色わけによって，どのルールが割り当てられているかを確認できる．また，タイムスライダ上でクリックすることで，各ストロークの動きルールを追加, 変更，削除を行うメニュー機能と, 付与された動きのタイミングを調整する機能を追加している．

さらに，ユーザが編集する手間を削減するために，動きルールを半自動的に決定するサポート機能にも着手した．半自動的に決定するために前提条件として,「これまでにユーザが作成したアニメーションをデータベースとして格納し, 新たに描いたイラストに適用する」ことを検討する．前提条件として「ユーザが新たに描いたイラストが，これまでに作成したイラストと類似している場合，同様のルールを適用する」と仮定し，ユーザが作成したアニメーション（イラストのシルエット情報と動きデータ）を事前にデータベースとして格納しておく．次に，ユーザ描いたイラストのシルエット情報とデータベース上のシルエットの類似度を計算し，類似度の高いイラストの動きデータを検索するものである．従来の手描きストローク間の類似度を計算する方法として，Ip ら [8] や Shin ら [17] はイラストの中心から各方向に対するベクトル情報を利用した手法を考案している．しかし、これらの手法は前章で扱ってきた手描きストロークの制御点を一切考慮しないため，描き順をはじめ複雑な形状への対応が困難であった．一方，描き順を考慮することを目的とし，ストロークの角度や曲率情報に着目した手法 [14][15] も提案されているが，ストローク全体の形状特徴を考慮できないことが課題としてあげられる．本機能は, ストロークの制御点座標を直接用いた類似度計算を検討する．手順として，入力ストロークの制御点 $\vec{v}_i,\ i \in \{0, \cdots, n\}$ に対する重心座標 $\vec{v}_{cm}$ を基準

表 1. 類似検索手法の精度比較.

	検索精度 (%)
提案手法	75.0
Ip et al. 2002 [8]	40.0

とし，データベース上のストローク座標 $\vec{d}_i$(重心座標 $\vec{d}_{cm}$) との誤差を評価関数として定義する．但し，$\vec{p}_i(=\vec{v}_i - \vec{v}_{cm})$，$\vec{q}_i(=\vec{d}_i - \vec{d}_{cm})$ とする．

$$E = \sum_i |\vec{p}_i - sR \cdot \vec{q}_i|^2 \tag{4}$$

$$R = A_{pq}S^{-1} = A_{pq}(\sqrt{A_{pq}^T A_{pq}})^{-1}$$

$$A_{pq} = \sum_i \vec{p}_i \cdot \vec{q}_i^T$$

n はストロークの頂点数,s はスケールの正規化用の値 ($s = \sum_i |\vec{p}_i|/|\vec{q}_i|$) である．$R$ は回転行列を示し，$R = A_{pq}S^{-1}$　$(S = \sqrt{A_{pq}^T A_{pq}})$ によって計算される．以上の手順によって，回転情報及び拡大縮小情報に依存しないストロークデータベースの検索及び，動きデータの割り当てを可能とする．

本手法の有用性を検証するために，類似度検索の精度評価実験を行った．実験に用いたデータベース数は 15 種類である．比較手法としては代表的な Ip らの類似度検索手法 [8] を用いた (表 1)．この結果から，従来手法に比べ，より高精度にユーザが描いたストロークと類似するデータベースを検索することが可能であることが分かる．また推定結果にユーザが満足しない場合，ユーザが手作業で動きルールを追加や変更することで，編集結果を新たな学習データとしてデータベースに格納することで，本システムを使い込むことでユーザ各自の好みにあった自動割り当てを実現する．将来的には，データベースにおいて格納するアニメーション種類の増築 (例：二次動作) に取り組む予定である．

4　生成結果

本システム上で作成した手描きアニメーションの一例を図 6 に示す．これらの作品はいずれも，5 歳から 20 歳までのユーザ 5 名が約 5 分程度で作成したものである．処理速度に関しては，ユーザが描いたストロークの本数に依存するものの，いずれの作品も 60[fps] 以上を実現した．また，本システムを使用してもらったユーザからのフィードバックとしては,「自分で描いた手描きの絵が動いたことがうれしい.」「Twitter や Facebook に気軽にアップロードしたい.」「gif 画像や LINE のモーションスタンプを自分で作りたくなる.」「授業の発表スライドに使うイラストやアニメーションとして使いたい.」と

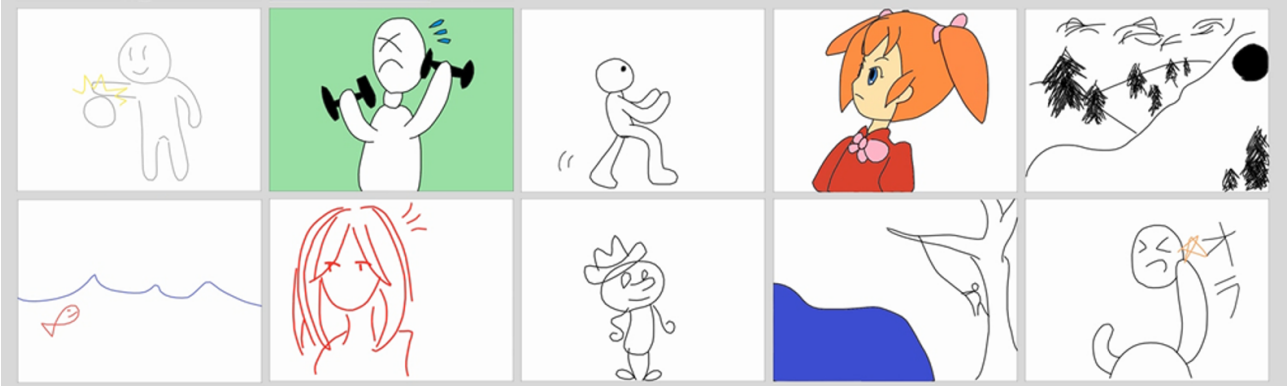

図 6. 本システムによる生成結果の一例.

いったポジティブな意見が得られた．一方で,「描いたイラストをグラデーション化する機能がほしい」「幾何学的なイラストを簡単に作成するために手描きストロークを直線状に補正する機能がほしい」といった 1 枚のイラストを描く際のサポート機能が欲しいというインターフェースの機能に関する意見も得られた.

5 まとめと今後の課題

本研究は，ユーザがペンタブレット上で描いたイラストから簡単にアニメーションを制作するために,(1) イラストを構成するストロークに特化した中割り形状の生成手法，(2) 事前に用意したデータベースを用いることで，ユーザが描いたストロークに動き情報を付与する半自動機能を開発し，その有用性を検証した．これらを併用することで，LINE のスタンプといったシンプルなアニメーション作品を簡単に作成することができるようになった．但し本システムは，イラストを構成するストロークを離散化を行っているため，離散化した制御点数や曲線補間法にイラストの形状やアニメーション結果が依存することが課題としてあげられる．そこで今後の課題として，よりユーザの描いた形状を反映し，直観的なサンプリング手法及び，曲線補間手法を開発する必要がある．またユーザにとって直観的な色味の補間 (表色系や補間方法) をはじめ，より詳細なイラスト描きこむための機能を追加する予定である．将来的には「一部の絵を書いたら残りの絵（またはアニメーション）を予想し、提示する」[22][23] などの効率的に描くための技術についても着手していきたい.

謝辞

本研究の一部は JSPS 科研費 15J07226, 情報処理推進機構 (IPA) の 2014 年度未踏人材発掘・育成事業の支援を受けて実施されたものである.

参考文献

[1] Alexa,M., Cohen-Or,D., and Levin,D. As-rigid-as-possible shape interpolation. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pp.157–164, 2000.

[2] Alexa,M. Differential coordinates for local mesh morphing and deformation. In *The Visual Computer*, Vol.19, Issue.2, pp.105-114, 2003.

[3] Baxter,W., and Barla,P., and Anjyo,K. Rigid shape interpolation using normal equations. In *Proceedings of the 6th international symposium on Non-photorealistic animation and rendering*, pp.59–64, 2008.

[4] Baxter,W., Barla,P., and Anjyo,K. N-way morphing for 2D animation. In *Computer Animation and Virtual Worlds (CAVW)*, Vol.20, Issue.2-3, pp.79–87, 2009.

[5] Chen,R., Weber,O.,Keren,D., and Ben-Chen,M. Planar shape interpolation with bounded distortion. In *ACM Transactions on Graphics (TOG)*, Vol.32, Issue.4, 108, 2013.

[6] Fukusato,T., Morishima,S. Active Comicing for Freehand Drawing Animation. In *Mathematical Progress in Expressive Image Synthesis III*, Vol.24, pp.45–56, 2016.

[7] Igarashi,T., and Moscovich,T., and Hughes,J F. As-rigid-as-possible shape manipulation. In *ACM transactions on Graphics (TOG)*, Vol.24, Issue.3, pp.1134–1141, 2005.

[8] Ip,H HS., Cheng,A KY., and Wong,W YF. Affine invariant retrieval of shapes based on hand-drawn sketches. In *Proceedings 16th International Conference on Pattern Recognition*, Vol.2, pp.794–797, 2003.

[9] Kaji,S., Hirose,S., Sakata,S., Mizoguchi,Y., and Anjyo,K. Mathematical analysis on affine maps for 2D shape interpolation. In *Proceedings of the 11th ACM SIGGRAPH/Eurographics conference on Computer Animation*, pp.71–76, 2012.

[10] Kazi,R H., Grossman,T., Umetani,N., and Fitzmaurice,G. Skuid: Sketching Dynamic Illustrations Using the Principles of 2D Animation. In

Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, pp.4599–4609, 2016.

[11] Levi,Z., and Gotsman,C. Smooth rotation enhanced as-rigid-as-possible mesh animation. In *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, Vol.21, Issue.2, pp.264–277, 2015.

[12] Live2D
http://www.live2d.com/en/

[13] Nakajima,M., Sakamoto,D., and Igarashi,T. Offline painted media for digital animation authoring. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 321–330, 2014.

[14] Namboodiri,A M., and Jain,A K. Retrieval of On-line Hand-Drawn Sketches. In *Proceedings International Conference on Pattern Recognition*, pp.642–645, 2004.

[15] Parui,S., and Mittal,A. Similarity-invariant sketch-based image retrieval in large databases. In *European Conference on Computer Vision*, pp.398–414, 2014.

[16] Sederberg,T.W., and Greenwood,E. A physically based approach to 2-D shape blending. In *ACM SIGGRAPH computer graphics*, Vol.26, Issue.2, pp.321–330, 1992.

[17] Shin,H., and Igarashi,T. Magic canvas: interactive design of a 3-D scene prototype from free-hand sketches. In *Proceedings of Graphics Interface 2007*, pp.63–70, 2007.

[18] Sumner,R W., and Popović,J. Deformation transfer for triangle meshes. In *ACM Transactions on Graphics (TOG)*, Vol.23, Issue.3, pp. 399–405, 2004.

[19] Sỳkora,D., Dingliana,J., and Collins,S. As-rigid-as-possible image registration for hand-drawn cartoon animations. In *Proceedings of the 7th International Symposium on Non-Photorealistic Animation and Rendering*, pp.25–33, 2009.

[20] Umetani,N., Schmidt,R., Stam,J. Position-based elastic rod. In *Proceedings of the 13rd ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 21–30, 2014.

[21] Whited,B., Noris,G., Simmons,M., and Sumner,R.W., Gross,M., and Rossignac,J. Betweenit: An interactive tool for tight inbetweening. In *Computer Graphics Forum (CGF)*, Vol.29, Issue.2, pp.605–614, 2010.

[22] Xing,J., Chen, H.-T., Wei, L.-Y. Autocomplete Painting Repetitions. In *ACM Transactions on Graphics (TOG)*, Vol.33, Issue.6, No.172, 2015.

[23] Xing,J., Wei, L.-Y., Shiratori,T., and Yatani,K. Autocomplete hand-drawn animations. In *ACM Transactions on Graphics (TOG)*, Vol.34, Issue.6, No.169, 2015.

WISS2016 採録判定時のコメント

採録区分：ロング採録

判断理由：

技術的な新規性（2 次元アニメーション作成の際に、手書きスケッチとデータベースを組み合わせる）に加え、システムがきっちり作られていること、既存手法に比べて優れた結果が得られていることなどを評価し、ロング採録となりました。一方で、システムの有用性や、実際にユーザを支援できたのかなどの考察が不足していますので、WISS でのディスカッションを通じて更なる研究の発展に繋げてください。

レビューサマリ：

全体の構成：
ペンタブレット上でユーザが描いた 2 次元のイラストから中割画像を生成し、データベースから既定の動作を合成することで 2 次元の手描きアニメーションの作成を支援するシステムの提案です。2 次元のアニメーションを作成する際に、手書きスケッチとデータベースをうまく組み合わせる部分に新規性があると判断します。提案手法の面白さに加え、システムがきちんと動いていること、既存手法に比べて優れた結果が得られていることなどが評価されました。

改良に向けたコメント等：

・現状では、システムの有用性や、ユーザを実際に支援できたのか等の考察が欠けています（検索精度が 75% とありますが、ユーザが満足しているかどうかの記述がありません）。

・計算時間に関しては記述がなく、リアルタイム性が保たれているかは不明です（リアルタイム性もユーザの満足度に繋がります）。

・アルゴリズム内部の動き (メッシュの動き) が不明確です。実際の図形内部のメッシュの動きを図などで示せば、よりわかりやすくなると思われます。

・ひとつの文が長すぎて意味がとりづらいことに加え、語句の間違いや図の参照ミスなどが散見されます。投稿前に十分に推敲を行うようにしてください。

本論文に対する各査読者の詳しいコメントは以下のページを参照のこと：
http://www.wiss.org/WISS2016Proceedings/oral/11.html

＊本ページは論文本体ではありません

【トレーシングとデータベースを併用する 2D アニメーション作成支援システム】