

## 4coman : 中割りアニメーション自動生成による セルアニメーション制作支援ツール

古澤 知英<sup>†1</sup> 福里 司<sup>†1</sup> 森島 繁生<sup>†2,3</sup>

**概要：**本稿では、手描きイラストをもとに作られるセルアニメーションを中割り画像の自動生成により、わずか4枚のイラストからアニメーション制作を可能とするシステム、4comanを提案する。本システムは、シーンに登場する物体をスクリーン上で絵を描くようにモデリングを行い、モデリングに関する専門的な知識を持たない人でも三次元形状を作ることが出来る。また、任意の角度から見た際の物体の形状を編集することで、モデリングを実現する。さらに、モデリング結果に中割りアニメーション生成手法を適用することで簡単にセルアニメーションを作ることが出来るインターフェースを構築する。

### 4coman : Making automatically In-between animation system from 4 hand-drawn frames

CHIE FURUSAWA<sup>†1</sup> TSUKASA FUKUSATO<sup>†1</sup> SHIGEO MORISHIMA<sup>†2,3</sup>

**Abstract:** This paper presents 4coman, a system that generates cel animations only from four hand-drawn frames. The user draws several 2D freehand strokes interactively on a screen and the system automatically generates quasi-3D models. Our system has the potential to educational use for non-professionals and children. Furthermore, this system is able to interpolate and edit freehand drawing strokes between key frames of various camera angles. With two interpolation algorithms (i.e., 2D interpolation algorithm and quasi-3D rotation interpolation algorithm), it is possible to create cel-animation easily and quickly.

### 1. はじめに

アニメーションは、絵日記や絵画のような一枚の静止画とは異なり、複数の絵を重ね合わせることで作られることから、ストーリー性を持った世界観を視聴者へ伝えることが出来る素晴らしいコンテンツといえる。特に、日本のアニメーションは世界規模で人気を博しており、その裏には原作となる多くの漫画が密接に関係していると考えられる。アニメーションはセルアニメーションとCGアニメーションの大きく二つに分けられる。日本のアニメーションは長年、セルアニメーションが大多数を占めていた。しかし、セルアニメーション制作はセル画と呼ばれる、動きのもとなる静止画をアニメータが一枚一枚描く必要があるため、時間的なコストが高いという問題を抱えている。さらに、アニメータの不足の問題も抱えており、セルアニメーション制作が危機的状態にあると言われている。一方で、Disneyをはじめ、海外のアニメーションスタジオではCGアニメーションの制作が普及している。CGアニメーションでは、キャラクタ等のモデリングを行い、モデリングした物体を各シーン中でカメラの設定、物体のポーズを指定すること

で制作される。一旦モデリングを行った物体は、他のシーンでも使用することが出来るため、作業の効率化が可能であり、アニメーション業界で大きな支持を得ている。そのような利点があるにもかかわらず、全てのアニメーションスタジオの制作工程がセルアニメーションの制作の仕方からCGアニメーションの制作の仕方に移り変わらない理由は、セルアニメーションが、物理的な矛盾表現をはじめ、アニメータの意図を十分に反映した映像が作ることが出来ることが挙げられる。さらに、CGアニメーションに比べて専門的な知識を必要とせず、人々が幼いころから慣れ親しんでいる「絵を描く」という行為でアニメーションを制作できることが、セルアニメーションを支持する大きな理由であると考えられる。よって、制作の効率化およびセルアニメーション制作の技術的難易度の敷居下げが出来る技術が登場すれば、CGアニメーションのように専門的なソフトウェアや、物理や数学的な知識を必要とせず、一般人にもアニメーション制作がより身近なものになりうるといえる。

そこで、本稿ではCGアニメーションで広く用いられるモデリング技術をセルアニメーションの制作工程に取り入れることで、セルアニメーション制作における時間的なコストの問題を解決しつつ、ユーザの意図を反映した手描きならではの表現を保持したアニメーション制作を行うことが出来るインターフェースを提案する。概念図を図1に示す。

†1 早稲田大学

Waseda University

†2 早稲田大学理工学術院総合研究所

Waseda Research Institute for Science and Engineering

†3 JST CREST

JST CREST

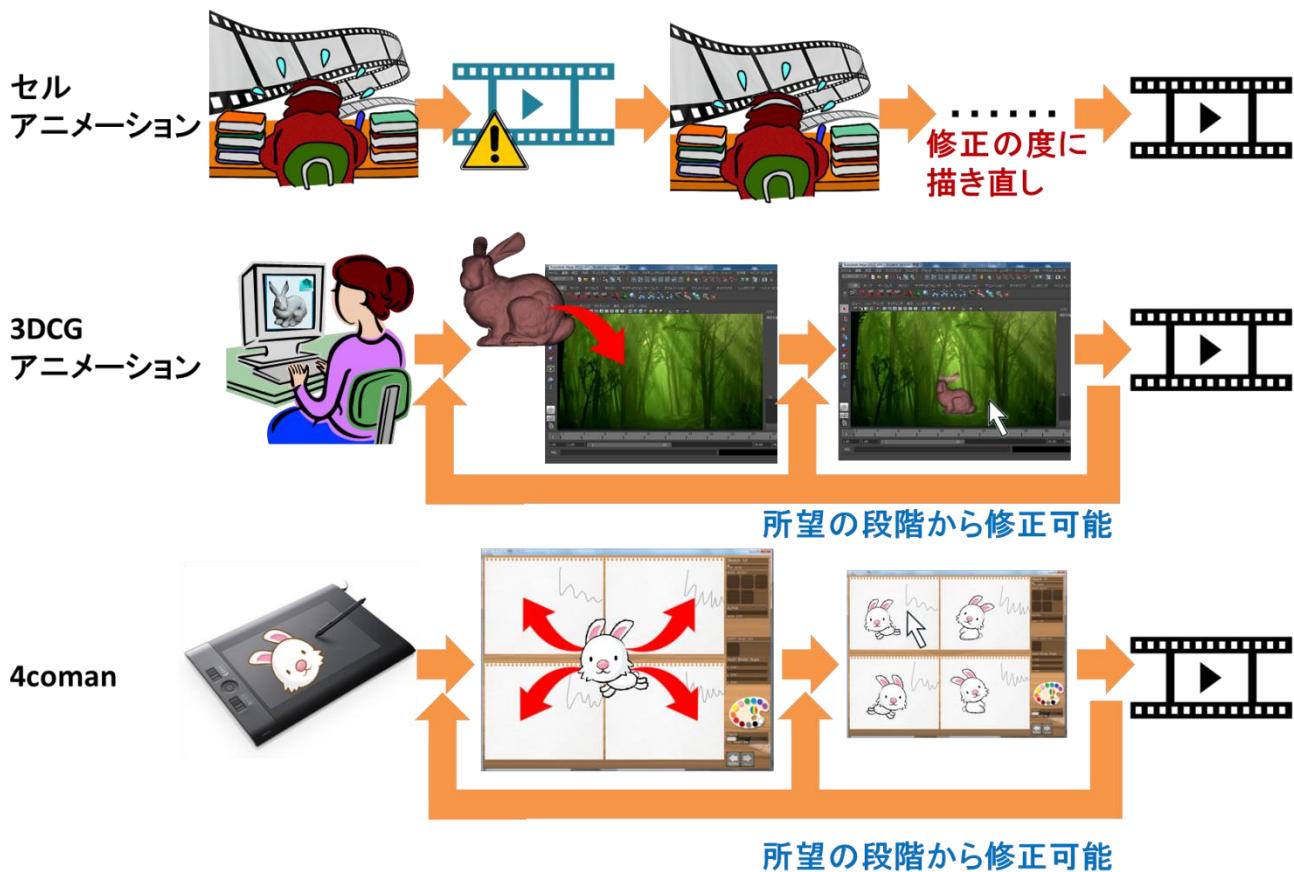


図 1. 4coman の概念図

セルアニメーション制作の課題である時間的コストを削減する方法の一つとして、数枚の絵(キーフレーム)から今の絵と次の絵の時間的経過を埋める連番画像(以下、中割り画像)を自動的に生成する手法が挙げられる。

従来の中割りアニメーションの手法では、キーフレーム間の形状はアフィン変換で近似することが多い。そのため、キーフレームで描かれた形状の特徴を保持した中割り画像を簡易的に生成することが可能となっている。しかし、キーフレーム間で回転のような三次元的な動きを含む場合、ある視点から見た物体の形状は大きく変化するため、前述の中割り生成手法を適用しても適切な中割り画像を得ることが出来ない。その結果、三次元的な回転に特化した中割り自動生成手法も提案してきた。

これらの研究背景から、中割り生成手法を用いたセルアニメーション制作技術には、二次元平面内の物体変化の表現手法と、三次元的な回転を含む物体変化の表現手法に分ける必要があることがわかる。そこで本稿では、ユーザがいくつかの視点からの物体の形状を描き、その絵をキーフレームとした三次元回転における中割り生成手法を用いたモデリングを行う。これにより、一般人でも自らの意図を反映したオブジェクトを簡単に作成することが出来る。三次元回転の動きを含む物体については、ユーザが指定した角度に対し、キーフレームを挿入し、平面的な動きで構

成される物体に関してはキーフレーム間でアフィン変形を考慮した中割り画像生成を行うことで三次元的回転を伴うセルアニメーションの自動生成が可能となる。一度ユーザがモデリングした物体は、何度もキーフレームとして挿入することができるため、CG アニメーション制作における長所を取り入れつつ、一般人でも簡単にアニメーション制作ができるインターフェースとなっている。

## 2. 関連研究

本章では、代表的なセルアニメーション制作支援インターフェースについて述べたのち、本インターフェース 4coman の機能について検討する。LIVE2D では、入力画像上にメッシュ情報を作成し、イラストを動かすシステムを提案した。この技術によって、原画の持つ魅力を生かしつつ、動き(中割り画像も含む)の生成を実現した<sup>[1]</sup>。しかし、メッシュを手作業で作成する必要があることや、メッシュの変形には各頂点間の線形補間を用いていることから、一つのシーンを制作するために多数の編集を要する。そこで、4coman ではユーザが描いたストローク上の特徴点を自動取得するアルゴリズムを加えることで、メッシュ作成の手間を省く。

Dalstein らは、複数のベクターグラフィックスを入力と

し、ベクターイメージ上のストロークの形状変化を時系列順に並べ、アニメーションを自動生成するインターフェースを提案した<sup>[2]</sup>。中割り画像中のストロークの形状は、White らの中割り生成アルゴリズムをもとに決定される<sup>[3]</sup>。手順として、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を行う。そのため、ユーザが描いたフレームを基に生成した中割りストロークは、インタラクティブかつ入力ストロークの特徴を保持することが出来る。しかし、フィッティング関数から得られる中割りストロークの形状変化量の許容幅が小さいため、キーフレーム間の変化量が大きい場合、中割りストロークの形状が大きく破綻する可能性がある。そこで、4coman では曲線フィッティングによるストローク形状ワーピングではなく、キーフレーム間でのストローク上の特徴点座標の変換による中割り生成アルゴリズムを導入する。

Xing らは、一枚目に描かれたストロークの形状と位置の情報を利用し、二枚目以降のストローク形状の予測及び提示することで、ストロークの形状特徴を保持した変形を許容し、中割り画像を簡単に作成できるインターフェースを提案した<sup>[4]</sup>。しかし、三次元的な動きを含むアニメーションを作成する場合、ストロークを考慮するだけでは、十分な予測が出来ず形状が破綻する場合も考えられる。そこで、我々のインターフェースでは、より多くのシーンに適用可能な中割り生成を目指すため、三次元的な動きにも対応した中割りアルゴリズムも検討する。

三次元的な動きを表現する方法として、Rivers らは入力ストロークごとに深度情報(離散的な z 値)を与えることで、擬似的な三次元モデリングを行うインターフェース、2.5D cartoon models を提案した<sup>[5]</sup>。ストロークの深度情報の変化および、ストロークの形状変化、位置変化をユーザがインタラクティブに設定することができる。2.5D cartoon models は、ストロークを自在に描くのではなく、頂点の位置を指定し、頂点間を曲線と直線の二種類で連結する仕様になっている。そのため、自在な形状を描くにはユーザが多数の頂点を設定する必要があり、手間がかかる。同様に、擬似的な物体のモデリングを行うインターフェースを Kitamura らが提案した<sup>[6]</sup>。Kitamura らのインターフェースでは、あらかじめ制作により自由に描かれたイラストを入力画像として用いるため、Rivers らのインターフェース上の問題を解決している。しかし、キャラクタの特徴点については、対応付けを手動で行う必要がある。そこで、4coman ではユーザが自由にストロークを描くことで入力ストローク上の特徴点の自動サンプリングを行う機能を備えたインターフェースを提案する。また、Xing らと Rivers らの研究にて、二次元平面内の動きと三次元回転を含む動きでそれぞれのアルゴリズムが必要であることが確認できるため、本インターフェースでは、これらの動きをそれぞれ作成するインターフェースの構築を行った。

### 3. ユーザインターフェース

4coman は 1) モデリングページ、2) 四コマ作成ページ、3) 中割り自動生成結果表示ページの三つの工程から構成されている。全てのページで表示されている右下部の矢印ボタンを押すことで、選択中の工程の完了および次の工程への移動をすることが出来る。各工程でのインターフェースの機能等は、各節で説明する。

#### 3.1 モデリングページ

モデリングページ図を図 2 に示す。図 2 中の左下部のモデリング視点設定部にてキーフレームとするアングルを設定した後、メインパネルにて設定したキーアングルにおける物体形状を描く。対称性のある物体を作成する場合、ユーザがストロークの一部を描き、設定した軸をもとにストローク情報を反転し、ストロークの複製を形成するモデリングのための絵描き補助機能を実装した。鏡越しにストロークを連結するか否か、モードを選択することができるため、ユーザは適宜手間を削減しつつキーフレームを作成することができる。また、ページ右上部の深度情報設定パネルにて各ストロークの深度情報を設定できる。パネルの下部に行くほどレイヤーは視点に対して手前に設定される。二枚以上のキーフレームを描いた後、右下部の『compute』ボタンを押すと、ユーザが描いたキーフレームが、ストロークの座標情報と深度情報をもとに物体形状がモデリングされる。モデリング結果がある視点からの形状が所望の形状でない場合、その視点での形状を逐次修正することができる。

#### 3.2 四コマ作成ページ

四コマ作成ページ図を図 3 に示す。モデリングページにて構築した物体以外で、アニメーション中に登場させたい他の物体のキーポーズを四コマパネル上にそれぞれ描く。モデリングページでモデリングを行った物体は、この段階で挿入することで、三次元的な動きを含ませつつ、メインアニメーションに登場させることができる。各コマで挿入したい角度を選択し、右中央部の『insert』ボタンを押すことで、指定のコマに挿入することができる。モデリングされた物体とその他のシーンに登場する物体が描かれたコマには、モーションペンシルと呼ばれる各ストロークの動きを指定するペンで物体の動きを指定することで、三次元回転の変化やストロークの形状変化だけでなく、キーフレーム間での物体の動作を付加することができる。モーションペンシルについては、4 章 3 節にて述べる。セルアニメーション制作では、同じ動作を繰り返す場合でも、どのフレームをどのタイミングで表示するのか繰り返しの回数分設定を行う必要があるが、4coman ではその手間を省きつつ、物体に動作を付加することができる。

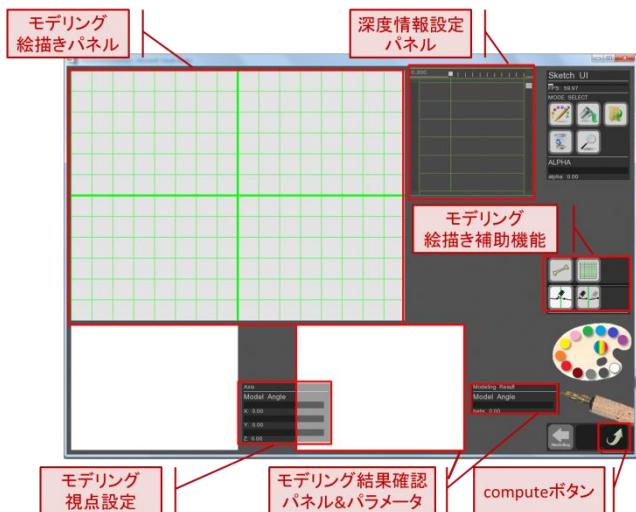


図 2. モデリングページ

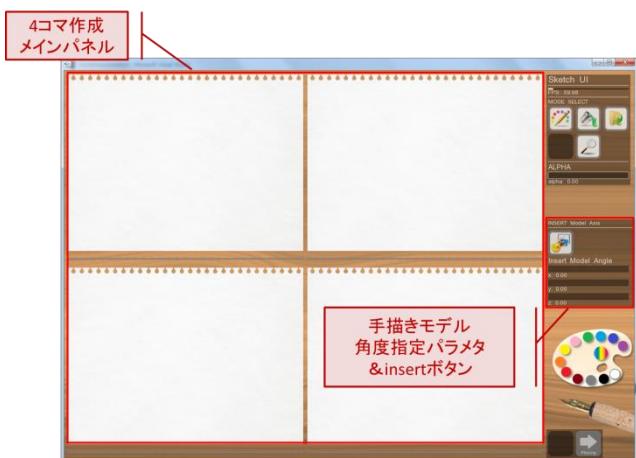


図 3. 四コマ作成ページ

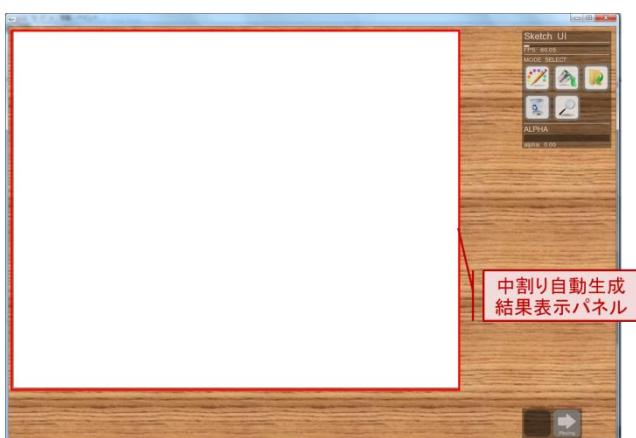


図 4. 中割り自動生成結果表示ページ

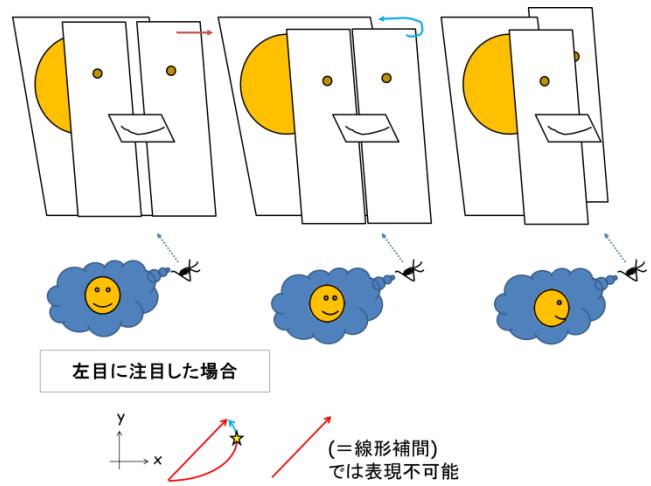


図 5. 三次元的な動きの補間について

### 3.3 中割り自動生成結果表示ページ

中割り自動生成結果表示ページを図 4 に示す。四コマページで作成した四コマの対応関係とモーションペンシルによる動きの付加情報から自動生成される中割りアニメーションをメインパネルにて確認できる。右上部の中割りパラメタを動かすことで、制作した中割りアニメーションを見ることが出来る。

## 4. アルゴリズム

本章では、モデリングページにて導入されている三次元回転を含む中割りアルゴリズムと、四コマ作成ページにて導入されている中割りアルゴリズムについて述べる。

### 4.1 モデリングアルゴリズム

第一章にて述べたように、三次元的な動きを含む中割りアニメーションは線形補間では表現することが出来ない。理由として、三次元的な動きでは、視点または物体の回転により遮蔽ペーストが生まれるため、各ストロークの形状・位置と共に深度情報の補間も行う必要があるからである。三次元的な動きを二次元平面上にて見た際の様子を図 5 に示す。図 5 中のキャラクタの左目に注目すると、左向きに回転により、左目が向かって右に移動した後に、左に折り返しつつ、左目の深度情報が輪郭の深度情報よりも後方に来ることで、左目が遮蔽されることが確認できる。よって、三次元的な動を表現するには、一方向の座標移動を示す線形補間のみでは表現が出来ないことが確認できる。

Rivers らは、深度、形状、位置の三つの情報を個別に補間を行った。我々も同様に、深度、形状、位置の三つの情報を分けて考える。ただし、位置情報の補間方法に関しては、Rivers らの手法と異なる手法を用いる。Rivers らの手法では、深度情報を設定し、設定された深度情報と各キーフレームでの角度をもとに位置を決定する。しかし、彼ら

は簡易的な三次元モデルを仮定しているため、視点を変更した後のストロークの位置が適切でない場合がある。そこで、我々は、ストローク上の各点の位置、および形状の情報の決定は、キーフレームのストローク上の点の座標値のみから算出するアルゴリズムを導入する。

Furusawa らは遮蔽ペーツの軌跡に着目し、COS 関数を用いて正面と側面の二枚のキャラクタ顔画像から三次元回転アニメーションを生成している<sup>[7]</sup>。本インターフェースも、二枚のキーフレームからモデリングを行う際は、この手法を用いる。ただし、Furusawa らの手法はキャラクタの顔の深度情報を用いていないため、領域の一部が他のペーツによって部分的に遮蔽される表現が出来なかった。そこで、深度情報をストロークごとに与え、深度情報についてのみ線形補間を行う。キーフレームが三枚以上だった場合や、一旦モデリングを行った場合、挿入されたキーフレームを全て用いて補間を行う。

#### 4.2 四コマ間補間アルゴリズム

二次元平面内での形状補間を行う際、現在、非常に有効な手法をして注目されているのが、Alexa らをはじめとした、As-Rigid-As-Possible Mesh Interpolation と呼ばれる手法である<sup>[8]-[13]</sup>。これらの手法はメッシュモデルの最小単位である三角形間のアフィン変換をもとに、物体の全体形状の変形を行うアルゴリズムとなっている。しかし、この手法はストロークのように対応づいた制御点二点から回転情報を含むアフィン変換行列を算出するのは困難であるといえる。そこで Fukusato らは、垂直方向に仮想的な頂点ベクトルを定義し、三角形メッシュを生成することで、ストロークに適用可能な擬似的な三角形構造を構築することを提案している<sup>[14]</sup>。そこで本インターフェースでも、Fukusato らの手法を用いて、ユーザが描いたストロークに対し擬似的な三角形構造を構築することで、四コマ間のストロークの中割り画像を生成する。

#### 4.3 モーションペンシルアルゴリズム

キーフレーム中のストロークに対する動きの割り当てについて記述する。インターフェース中のお絵かきモードを解除すると、モーションペンシルモードに切り替わる。モーションペンシルでスクリーン上に軌跡を描くことで、その始点と最も近いストロークに対し、移動情報を付与する機能を追加した。これにより、四コマのキーフレーム中にあるストロークに、中割り用の対応情報を与えずに動きを割り当てることが出来る。また、モーションペンシルの軌跡は、中割りアニメーション生成前に確認、および修正を行うことが出来る。

## 5. 生成結果

以上の手順で制作したアニメーション結果の一例を図 6, 7 に示す。これらの結果から、中央のキャラクタに対して三次元的な動きを導入し、周りの物体にも中割りアニメーションを付加することによって、中割りアニメーションの生成を実現したことが確認できる。また、処理速度に関しては、ストロークの本数(制御点数)やサンプリング間隔に依存するが、図 6 および 7 の場合、各フレームで 40 本以上のストロークがあり、総特徴点数が 400 点を越えていたが、いずれも 30[fps]以上を実現した。そこで、ユーザテストとして、本システムをタブレット端末上で数名のユーザに使用してもらい、簡単なフィードバックを得た。タブレット端末上でも十分な処理速度等の遅延は感じられなかった。ユーザテストの結果、「自分が手で描いた絵にあらゆる動きを付加が出来るため、面白い。」「ハイスペックな機器を必要とせずにアニメーション制作が出来るのが嬉しい。」といったポジティブな意見が得られている。一方で、モーションペンシルの動きの結果の確認やサンプリング間隔についての改善の希望の意見も得られた。

## 6. まとめと今後の課題

本研究は、ユーザが描いたストロークを対象とした中割り画像の自動生成システム、4coman を提案した。三次元的な動きを付加する物体のモデリング、およびストロークに特化した中割りアルゴリズムを検証した。ユーザが描いた手書き画像をもとに、gif アニメーション等のシンプルなアニメーションを簡易的に生成するお絵かきインターフェースの実装を行った。

今後の課題について検討を行う。まず、本インターフェースでは、ユーザが描いたストロークに等間隔サンプリングを施し、特徴点を取得している。これにより、ストロークの特徴的な形状を取得し損なう可能性がある。今後は、ストロークの曲率や勾配情報を用いた自動サンプリング手法を開発する予定である。次に、キーフレーム間のストロークの対応関係を、現在、書き順をベースに取得している。前後のフレーム間で対応付けが行われなかったストロークに関しては、フェードアウト( $\alpha$  値補間)または、効果を付加しない設定を適用させている。そこで、今後は、ストロークの書き順のみではなく、ストロークの座標や長さを考慮した対応付けを行うことで、よりユーザにやさしいキーフレーム制作を行えるように実装を加えたい。さらに、本インターフェースでは、色情報の補間として RGB 表色系の線形補間を用いている。今後は、色の変化が、よりユーザにとって自然な色の補間になるよう、その他の表色系や導入した表色系における補間方法についても今後検討を行っていきたい。

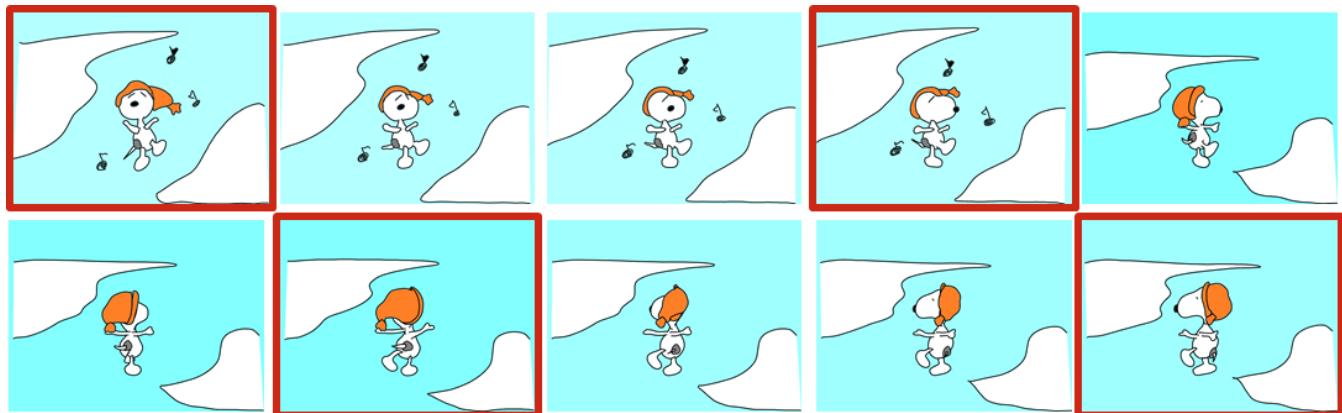


図 6. 中割り自動生成結果の例(その 1)



図 7. 中割り自動生成結果の例(その 2)

**謝辞** 本研究の一部は情報処理推進機構(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) LIVE2D. 2014. <http://www.live2d.com/en/>
- 2) B.Dalstein, R.Ronfard, and M. van de Panne. Vector graphics animation with time-varying topology. ACM TOG 34, 4, 145. 2015.
- 3) B.White, G.Noris, M.Simmons, R.Sumner, M.Gross, and J.Rossignac. BetweenIT: An Interactive Tool for Tight Inbetweening. Computer Graphics Forum, 29, 2, 605-614. 2010.
- 4) J.Xing, L.Wei, T.Shiratori, and K.Yatani. Autocomplete hand-drawn animations. ACM TOG 2015 34, 6, 169. 2015.
- 5) A.Rivers, T.Igarashi, and F. Durand. 2.5D cartoon models. ACM TOG 29, 4, 59. 2010.
- 6) M.Kitamura, and R.Tsuruno, 2.5D Modeling from Illustrations of Different Views, International Journal of Asia Digital Art and Design,8, 4, 74-79. 2015.

- 7) C.Furusawa, T.Fukusato, N.Okada, T.Hirai, and S.Morishima. Quasi 3D rotation for hand-drawn characters. ACM SIGGRAPH 2014 Posters, 12. 2014.
- 8) M.Alexa, D.Cohen-or, and D.Levin. As-Rigid-As Possible Mesh Interpolation. ACM SIGGRAPH. 157-164. 2000.
- 9) T.Igarashi, T.Mosovich, and J.F.Hughes. J.F. As-Rigid-As-Possible Shape Manipulation. ACM TOG 24, 3, 1134-1141. 2005.
- 10) W.Baxter, P.Barla, and K.Anjyo. Rigid Shape Interpolation using Normal Equations. In Proc. NPAR. 59-64. 2008.
- 11) S.Kaji, S.Hirose, S.Sakata, Y.Mizoguchi, and K.Anjyo. Mathematical Analysis on Affine Maps for 2D Shape Interpolation. In Proc. SCA. 71-76. 2012.
- 12) R.Chen, O.Weber, D.Keren, and M.B.Chen. Planar Shape Interpolation with Bounded Distortion. ACM TOG 32, 4, 108. 2013.
- 13) Z.Levi, and C.Gotsman. Smooth Rotation Enhanced As-Rigid-As-Possible Mesh Animation. IEEE TVCG, 21, 2, 264-277. 2015.
- 14) T.Fukusato, and S.Morishima. Active Comicing for Freehand Drawing Animation. Lecture Notes in Mathematics-for-Industry, Vol.64, 6-15, 2015.