

# 異なる視点から描かれたイラストからの2.5Dモデル生成

北村 真紀<sup>1,a)</sup> 金森 由博<sup>2</sup> 鶴野 玲治<sup>1</sup>

**概要:** キャラクタデザインの際、キャラクターの正面、側面や斜めなどから見たイラストが描かれる。本研究ではそのような異なる視点から描かれた2枚のイラストを用いて2.5Dモデルを生成する。2.5Dモデルは変形するビルボードを奥行き方向に重ねたモデルであり、2視点の間のキャラクターの見え方を、奥行きを考慮して表現できる。提案手法は単色の輪郭線と閉領域で構成されたカートゥン調のイラストと各イラストの視線方向を入力とする。各閉領域を2枚のイラスト間で対応付け、対応付けられた閉領域を3D空間にビルボードとして配置することで2.5Dモデルを作成する。これらは半自動的に処理される。さらに提案手法は、遮蔽によって消失・出現する閉領域の補完を行うことで、視点が変わった時に遮蔽されて見えなくなっていくような部分が存在するイラストの補間も可能にした。

## 2.5D Modeling from Illustrations of Different Views

**Abstract:** When artists design characters, they draw illustrations viewed from the front, side or slant views. In this paper, we create a 2.5D model using such illustrations. A 2.5D model is a model that is created by arranging deformable billboards in the depth direction, and it can express an appearance of the character between two viewpoints with considering depth information. Our method uses two cartoon-like illustrations and eye directions as inputs. These illustrations are consisted of contours and closed regions painted with uniform colors. Our method finds corresponding regions between two illustrations, and creates a 2.5D model by putting each matched region as a billboard in 3D space. These processes are semi-automatic. By completing regions that appear or vanish, our method can interpolate illustrations with occluded regions.

### 1. はじめに

アーティストはキャラクターをデザインする際に、キャラクターの正面、側面や斜めなどから見たイラストを描くことが多い。3DCGアニメーション制作においてはこのようなイラストをもとにキャラクターの3Dモデルが作られるが、3Dモデリングは制作に時間がかかり、また、イラストのニュアンスを反映するのが難しいという問題がある。こうした問題を解決するために近年では2.5Dモデルを使う手法も用いられている [1], [2]。2.5Dモデルとは、2Dのイラストを目や口などのパーツごとにレイヤーにし、各々のレイヤーに奥行きを与えることで、異なる視点からの見目を表現できるモデルである。既存のシステム [2] では、異なる視点から見たキャラクターのイラストを目や口などのパーツごとにベクトルデータとして作成し、補間すること

で任意の視点から見た形状を生成している。さらにこれらのパーツをビルボードとして3D空間に配置することで2.5Dモデルを作成し、正確な3Dモデルを作らずに、奥行きを考慮した表現を可能にしている。この手法ではイラストのニュアンスを反映した2.5Dモデルを3D空間で操作することを可能にした。一方で、各パーツのベクトルデータを手作業で作るため、非常に手間のかかる作業となっている。

本研究ではこの作業の手間を軽減するため、2.5Dモデルを作成する新しいワークフローを提案する。本手法において、ユーザは2.5Dモデル作成のためにパーツを一つずつ手作業で作るのではなく、すでに描かれたイラストに対応付けの操作を行うことで2.5Dモデルを作成するため、キャラクターデザインの際に描いたイラストを流用でき、パーツ作成の手間を省くことができる。複数枚の画像を補間する既存のモーフィング [3], [4], [5] と比べると、モーフィングでは遮蔽により画像間での対応が取れない部分の補間を行うことができない。一方、本手法では遮蔽により見えなくなっている部分にユーザがその形状のストロークを描くこ

<sup>1</sup> 九州大学  
Kyushu University

<sup>2</sup> 筑波大学  
University of Tsukuba

<sup>a)</sup> maki.terai@gmail.com

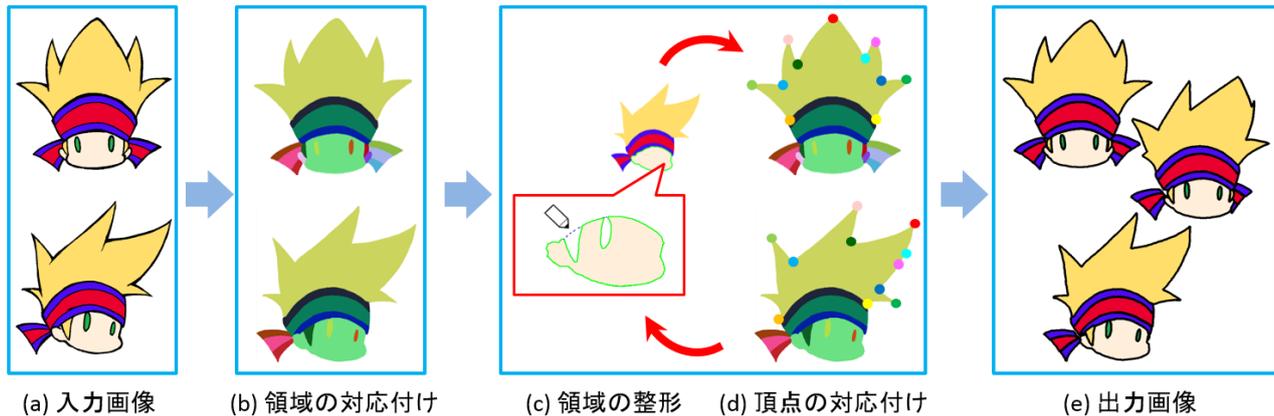


図 1 システム概要。

とで形状を補完し、画像間の補間を行う。

## 2. 関連研究

3DCG において、手描きのイラストのような表現を行うための研究は Non-photorealistic rendering の分野で発展してきた [6], [7]。しかし、手描きのキャラクターのイラストにおいては正面から見た状態と側面から見た状態に矛盾が生じることがあり、3D モデルを作ること自体が困難な場合もある。このような問題に取り組んだ研究が View-dependent geometry である [8]。この手法はキーとなる視点ごとに 3D モデルを歪め、線形補間を行うことで、静止した物体として見ると矛盾が生じる形状の表現を可能にした。一方で、この手法は視点ごとに参照用のイラストを用意し、3D モデリングを行う必要があった。本手法では 2.5D モデルにより、異なる視点から見た時に矛盾が生じるような形状の表現を可能にしている。

また、2D のキャラクターアニメーションのための研究も多く行われている。Igarashi らは 1 枚の入力画像をメッシュ分割し、形状の特徴を保ったままメッシュを変形することでキャラクターのアニメーションを可能にした [9]。また、1 枚のキャラクター画像にスケルトン構造を考慮してモーションキャプチャの動きをつける研究もある [10], [11]。複数枚の画像を用いてキャラクターに動きをつける手法としてモーフィングの手法がある [3], [4], [5]。Baxter らはカートゥーン調の手描きのイラストをモーフィングを用いて動かす研究を行った [12]。これらの手法は固定された視点でのキャラクターの見た目での表現は可能であるが、視点の変更により変形の途中で遮蔽を伴う部分が生じるような表現には対応できない。これは遮蔽された部分が欠損して補間時に対応が取れないためである。

遮蔽された領域があっても補間が可能な手法として Whited らは BetweenIT を提案した [13]。これは 2 つのキーフレームとなる線画のストロークを対応付け、対数螺

旋を用いてストロークの補間を行うものである。この手法では遮蔽が生じる部分にユーザがストロークを追加することで見えない部分の補完を行うことができるが、奥行きを考慮せずストロークの対応付けのみで補間を行うため、視点が大きく変わった時の見た目の変化に対応できない。古澤らは顔の線画の正面図と側面図に対し、特徴点を対応付けて補間を行った [14]。彼らの手法でも遮蔽は扱えるが、奥行きを考慮しておらず、我々の手法では可能な、領域の一部だけが遮蔽される様子を表現できない。

2.5D モデルを使うことで、アニメーションを生成する研究も行われてきた。Di Fiore らは複数視点から見た手描きのイラストの補間を行った [1]。この手法ではイラストの各パーツの奥行きを手動で設定する必要がある。本研究では、各視点での視線方向のベクトルを使うことで奥行きを自動で推定できる。Rivers らは 2.5D モデルを作成するインタフェースを提案した [2]。これはユーザが様々な視点から見たキャラクターの目や口などのパーツをストロークで描くことで 2.5D モデルを作成し、補間を行うことができるシステムである。この手法では複数視点からみたストロークの補間を行うことはできるが、ストロークをユーザがひとつずつ作成することは非常に時間がかかる。本手法ではアーティストがキャラクターデザインの際に描いた正面や側面から見たイラストを流用できるため、ストローク作成の手間がかからない。2.5D モデルを作成する別の研究として、Yeh らが提案した手法がある [15]。これはイラストの表と裏の面を用いて 2.5D モデリングを行うため、球や回転体のように回転時に輪郭の形状が変わらない物体であれば遮蔽を伴う表現を行うことができるが、回転時に輪郭の形状が変化するような物体の表現はできない。本研究では複数の視点から見た手描きのイラストを用いて形状を補間するため、輪郭の形状が変化するような場合にも対応できる。

### 3. アルゴリズム

#### 3.1 システム概要

本研究では入力データとして正面図や側面図など、どの方向から描かれたか既知であるような2枚のイラストとその方向を用いる。右手系の3D直交座標系を次のように設定する。キャラクタの中心を原点とし、キャラクタは $z$ 軸の正の方向を向いており、キャラクタの右および上方向をそれぞれ $x, y$ 軸の正の方向とする。キャラクタを観察する視線方向と $x$ 軸のなす角を $\theta$ 、 $y$ 軸となす角を $\phi$ とし、キャラクタを正面から観察している状態を $(\theta, \phi) = (0, 0)$ とする。イラストは閉領域と輪郭線で構成され、閉領域内の色は単色で輪郭線の色は黒色とする。

図1に本手法の概要を示す。本手法ではイラストを閉領域に分割し(図1(a))、2つのイラストの間で閉領域を対応付け(図1(b))、対応付けられた閉領域をビルボードとして3D空間に配置する。その3D位置は対応付けられた閉領域の重心位置と各イラストの視線方向から求められる。ビルボードの表示は平行投影による。ビルボードの形状は、2枚のイラストから得た閉領域の形状を線形補間することで得られる。この補間のために、ユーザは閉領域の形の修正や特徴点の対応付けを行う(図1(c),(d))。これらの処理を行い2.5Dモデルを作ることで、奥行きを考慮した見た目の変化を表現できる。

#### 3.2 領域分割と領域の形状整形

最初に入力画像に対してflood fillアルゴリズムを用いて閉領域に分割する(図1(a))。このとき、輪郭線は一定の太さを持っているため、輪郭線を含まずに閉領域をビルボードとして配置すると閉領域間に隙間ができてしまう。そこで輪郭線の領域を分割された閉領域のいずれかに統合する処理を行う。これはSýkoraらの手法[16]と同様の手法を用いる。具体的には輪郭線のピクセルデータと各閉領域との距離を求め、距離が一番近い閉領域に輪郭線ピクセルを統合する。また、小さい閉領域が大きい閉領域に内包される場合は大きい閉領域に穴が空くが、穴が空いたまま補間すると穴まで補間されてしまう。これを防ぐために、穴が空いている部分を埋める処理を自動で行う。図1(c)は目の閉領域によって顔の閉領域に空いた穴を埋めている例である。また、イラスト上では2つの閉領域が互いに隣接していても、スクリーン投影時にビルボードの奥行きの違いによって閉領域間に隙間が生じることがある。このような隙間をなくすために、ユーザが遮蔽された閉領域の形状のストロークを描くことでその形状を変形することも可能である。

#### 3.3 類似度による領域の対応付け

領域分割の後、一方のイラストの各閉領域に対し、他方のイラストの中で類似度が最も高い閉領域を対応付ける(図1(b))。閉領域の類似度の計算方法としてLiuらの手法がある[17]。彼らの手法では閉領域の色、形状、重なっている部分の面積を考慮して類似度が計算されるが、片方の閉領域が他方の閉領域に完全に内包される場合に類似度が最大となる。このことが本研究の例では問題となり得る。例えば、視点の変更によって目の位置がずれて顔の閉領域に完全に内包され、顔と目の類似度が最大となり、誤って対応付けられてしまう。そこで本研究ではLiuらの手法[17]を改良し、閉領域 $a, b$ に対する類似度 $s_{a,b}$ を次式(1)によって計算する。

$$s_{a,b} = \mathcal{J}_{a,b} e^{-w_x|x_a-x_b|} e^{-w_y|y_a-y_b|} e^{-\frac{|S_a-S_b|}{(S_a+S_b)^{1/2}}} \quad (1)$$

この式のうち $\mathcal{J}_{a,b}$ のみがLiuらのものと共通している。これは色に関する類似度であり、閉領域の色が似ているほど大きな値となる。

$$\mathcal{J}_{a,b} = H[T_c - C_{a,b}] \quad (2)$$

ここで $C_{a,b}$ はRGB色空間における閉領域 $a, b$ のベクトル $\mathbf{q}_a, \mathbf{q}_b$ のユークリッドノルムであり $C_{a,b} = \|\mathbf{q}_a - \mathbf{q}_b\|$ である。 $T_c$ はユーザ定義の定数であり、本研究では0.3を用いた。 $H[n]$ はHeavisideステップ関数であり、 $n$ が0以上のとき1でそれ以外は0である。 $e^{-w_x|x_a-x_b|} e^{-w_y|y_a-y_b|}$ は位置の類似度を表している。この類似度はキャラクタを観察する視線方向と位置の変化を考慮して求められる。閉領域 $a, b$ のバウンディングボックスの中心をそれぞれ $(x_a, y_a), (x_b, y_b)$ とする。視線方向と $x$ 軸のなす角 $\theta$ が変化しない場合、つまり視線が上下にのみ動く場合はスクリーン座標系での $x$ 成分はほとんど変化しないため、特に $|x_a - x_b|$ がより小さい場合に類似度を高くする。一方、視線方向と $y$ 軸のなす角 $\phi$ が変化しない場合は、特に $|y_a - y_b|$ がより小さい場合に類似度を高くする。これを実現するために、重み $w_x, w_y$ を角度に応じて調整する。 $w_x$ は $\theta = 0$ のとき1.0、それ以外のとき0.5とし、 $w_y$ も $\phi$ に関して同様に定める。 $e^{-\frac{|S_a-S_b|}{(S_a+S_b)^{1/2}}}$ は閉領域の大きさについての類似度である。領域 $a, b$ のそれぞれの面積を $S_a, S_b$ とすると、 $S_a, S_b$ が互いに近ければ類似度が大きくなる。正規化のために面積の平均で除算している。

上記の類似度に基づき、一方のイラストの各閉領域に対し、他方のイラストの中で最も類似度の高い閉領域のペアを対応付ける。この手法を用いて対応付けを行った結果を図2と表1に示す。図2において(a)は入力画像、(b)はLiuらの手法による対応付け、(c)は本手法による対応付けの結果である。閉領域の対応付けが正しく行われている部分はグレー、間違っているところは赤、対応が取れなかった部分は青で示している。表1は、対応が存在する閉

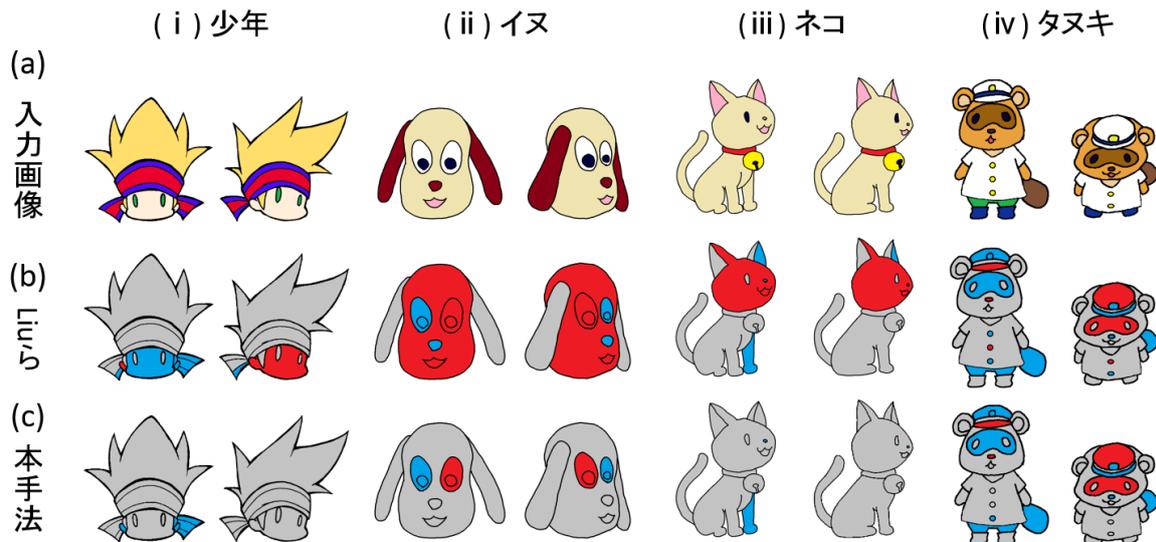


図 2 類似度による対応付けの比較。閉領域の対応付けが正しく行われている部分はグレー、間違っているところは赤、対応が取れなかった部分は青で示している。

領域のうち、正しく対応付けられた正解数を示している。少年の例では、Liu らの手法は、顔と耳の閉領域の類似度が誤って高くなってしまい対応付けに失敗している。本手法においては両耳と右側の鉢巻と髪の毛の対応が存在しないが、これらに対応する閉領域は存在しないため、対応が取れないことが正解である。イヌの例では、Liu らの手法は顔と口の閉領域の類似度が誤って高くなってしまい正しく対応付けられていない。目に関しては両手法とも間違っていて、左右が入れ替わって対応付けられている。ネコの例では、Liu らの手法は顔と口の閉領域が誤って対応付けられている。本手法ではネコの左の前足の対応が存在しないが、これは対応する閉領域がないので正解である。タヌキの画像においては、Liu らの手法ではボタンの対応が間違っているが本手法では正しく対応付けられている。対応付けが間違っている場合は、ユーザがインタラクティブに修正できる。また、遮蔽されていて対応が取れない閉領域に関しては、ユーザが遮蔽されている位置に閉領域の形状のストロークを描き、新しい閉領域を作成することで対応付けを行う。

表 1 遮蔽されておらず対応付け可能な閉領域に対する正解数。括弧内の数字は正解率。

画像	対応付け可能な閉領域数	Liu らの手法	本手法
少年	11	9 (0.82)	11 (1.00)
イヌ	9	2 (0.22)	5 (0.56)
ネコ	12	8 (0.67)	11 (0.92)
タヌキ	23	15 (0.65)	17 (0.74)

### 3.4 ビルボードの 3D 位置の推定

異なる視点から見た閉領域の対応が取れば、閉領域をビ

ルボードとして表現したときの 3D 空間での位置  $\mathbf{p}$  を推定できる。この手法は Rivers らの手法 [2] を用いる。この手法の詳細について著者に確認したところ、次の Algorithm 1 を用いているとの回答があった。

---

#### Algorithm 1 Calculate a 3D billboard position.

---

```

 $\mathbf{p}_{sum} \leftarrow (0, 0, 0), \mathbf{p}_{current} \leftarrow (0, 0, 0)$ 
 $N \leftarrow$  many times (e.g. 10,000)
for  $i = 0$  to  $N$  do
   $N_V \leftarrow$  the number of views
  for  $j = 0$  to  $N_V$  do
     $\mathbf{c}_j \leftarrow$  the center of bounding box of region  $R_j$ 
     $\mathbf{l}_j \leftarrow$  the 3D line that passes through  $\mathbf{c}_j$  and goes in the view direction of the view
     $\mathbf{p}_j \leftarrow$  the 3D position that projected  $\mathbf{p}_{current}$  onto  $\mathbf{l}_j$ 
     $\mathbf{p}_{sum} \leftarrow \mathbf{p}_{sum} + \mathbf{p}_{current}$ 
  end for
   $\mathbf{p}_{current} \leftarrow \mathbf{p}_{sum} / N_V$ 
end for
 $\mathbf{p} \leftarrow \mathbf{p}_{current}$ 

```

---

この手法を用いることで各閉領域のビルボードの 3D 位置  $\mathbf{p}$  を求めることができるが、この座標をそのまま用いるとスクリーン座標系におけるビルボードの位置と入力画像の閉領域の位置にずれが生じる。このずれをなくすために、対応付けられた 2 つの閉領域  $a, b$  の重心から視線方向に伸びる半直線に  $\mathbf{p}$  を射影することで各閉領域の最終的な 3D 位置  $\mathbf{p}_a$  と  $\mathbf{p}_b$  を決める。イラスト間の補間を行うときは、 $\mathbf{p}_a$  と  $\mathbf{p}_b$  を線形補間して求める。本システムでは、ビルボードの奥行きを修正するためにユーザがインタラクティブに任意の奥行きを設定できる。

### 3.5 領域の形状の特徴点の対応付け

対応付けと 3D 位置の決定により、各閉領域は 3 次元空間にビルボードとして配置される。ビルボードの形状は、2 枚の入力画像の各閉領域の輪郭の形状を線形補間したものである。本研究では閉領域の輪郭線上に特徴点を設定し、対応付けられた閉領域間で特徴点を対応付ける。これらの特徴点ごとに輪郭線を線分で区切り、その線分ごとに補間を行う。特徴点はユーザがインタラクティブに付与することができる。図 1(d) に特徴点の例を示す。なお、Baxterらの手法 [18] のように特徴点を自動で対応付ける手法があるが、遮蔽を伴う動きのように画像間での変化が大きい場合は自動で適切な対応付けを行うことは困難であり、対応付けの修正に手間がかかるため、本研究では特徴点の付与と対応付けはユーザ入力とした。

## 4. 結果

本手法を用いたアプリケーションを C++ と Qt Library を用いて実装した。実行環境として Intel Core i7-2760QM 2.40GHz CPU の PC を用いた。図 3 は本手法を用いて生成したアニメーションの一部である。赤枠で囲んだイラストが入力となる画像データで、青枠で囲まれているイラストは本手法により生成されたフレームである。入力したイラストは、それぞれ正面画の他に、少年、イヌ、ネコのイラストでは  $\phi = 0$  のまま  $\theta$  のみ  $45^\circ$  変化させたもの、タヌキのイラストでは  $\theta = 0$  のまま  $\phi$  のみ  $45^\circ$  変化させたものを用いた。制作にかかった時間はいずれも 10 分から 50 分程度であり、その中でも領域の整形と奥行き修正に大部分の時間を必要とした。少年のイラストでは髪や鉢巻の結び目が遮蔽されていく様子が表現できている。イヌのイラストでは左耳が顔に遮蔽される様子が表現できている。ネコのイラストでは左側の前足が遮蔽されていく様子が表現できている。タヌキのイラストでは黄色のボタンとズボンが遮蔽されていく様子が表現できている。

本手法においてキャラクターの一部が遮蔽されていく様子は表現できるが、キャラクターの輪郭線の表示に問題が生じる場合がある。本手法では閉領域の境界に黒い輪郭線を描いているが、この手法では元の輪郭線となるピクセルの座標しか保持しないため、輪郭線の太さや太さの変化の度合いを表現することができない。図 3 の入力画像では金髪の少年の髪の毛の閉領域の輪郭線の内側にも線が描かれているが、現在の実装ではこのような線を表現できない。今後はこのような線の情報の表現も必要であると考えられる。

## 5. まとめと今後の課題

本研究では異なる視点から描かれた 2 枚のイラストを用いて 2.5D モデルを作成し、イラスト間の補間を行った。まず、入力となるカートゥン調のイラストを閉領域ごとに分割し、閉領域の類似度を計算して対応付けを行う。次に各

閉領域を 3D 空間にビルボードとして配置することで 2.5D モデルを作成した。本研究の手法を実装したシステムでは対応付けを半自動で行うため、ユーザは少ない入力で異なる視点から見たキャラクターの補間を行うことができる。本手法では 3D 空間上でのビルボードの位置の計算に閉領域の座標のみを用いているため、求められた 3D 座標が意図した値にならない場合も生じる。例えば、イラストでは目の閉領域が顔の閉領域よりも手前にあるように描かれていても、求められた 3D 位置の奥行きでは目が顔より奥に配置されてしまう場合もある。こうしたエラーを削減するために、2D での重なり順を求め、3D 空間での奥行きがこの順番を反映するようにすればよいのではないかと考えている。今後はこのような重なり順を考慮した 3D 位置の計算方法および、重なり順をインタラクティブに推定し設定できるシステムを考案したい。また、閉領域ごとに補間をするシステムを拡張し、複数視点からのキャラクターのイラストを用いて任意の姿勢を作ることができるような手法を考案したい。

謝辞 本研究は日本学術振興会の特別研究員奨励費（課題番号 25・5451）の支援により実施された。

## 参考文献

- [1] Di Fiore, F., Schaeken, P., Elens, K. and Van Reeth, F.: Automatic in-betweening in computer assisted animation by exploiting 2.5D modelling techniques, *Computer Animation, 2001. The Fourteenth Conference on Computer Animation. Proceedings*, pp. 192–200 (2001).
- [2] Rivers, A., Igarashi, T. and Durand, F.: 2.5D Cartoon Models, *ACM Trans. Graph.*, Vol. 29, No. 4, pp. 59:1–59:7 (2010).
- [3] Baxter, W., Barla, P. and Anjyo, K.-i.: Rigid Shape Interpolation Using Normal Equations, *Proceedings of the 6th International Symposium on Non-photorealistic Animation and Rendering*, NPAR '08, New York, NY, USA, ACM, pp. 59–64 (2008).
- [4] Sýkora, D., Dingliana, J. and Collins, S.: As-rigid-as-possible Image Registration for Hand-drawn Cartoon Animations, *Proceedings of International Symposium on Non-photorealistic Animation and Rendering*, pp. 25–33 (2009).
- [5] Wolberg, G.: Image morphing: a survey, *The Visual Computer*, Vol. 14, No. 8-9, pp. 360–372 (1998).
- [6] Decaudin, P.: Cartoon Looking Rendering of 3D Scenes, Research Report 2919, INRIA (1996).
- [7] Rusinkiewicz, S., DeCarlo, D. and Finkelstein, A.: Line Drawings from 3D Models, *ACM SIGGRAPH 2005 Courses*, SIGGRAPH '05, New York, NY, USA, ACM (2005).
- [8] Rademacher, P.: View-dependent Geometry, *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, New York, NY, USA, ACM Press/Addison-Wesley Publishing Co., pp. 439–446 (1999).
- [9] Igarashi, T., Moscovich, T. and Hughes, J. F.: As-rigid-as-possible Shape Manipulation, *ACM Trans. Graph.*,



図 3 本手法により生成された画像の一部。赤枠で囲んだイラストが入力となる画像データで、青枠で囲まれているイラストは本手法により生成されたフレームである。

- Vol. 24, No. 3, pp. 1134–1141 (2005).
- [10] Hornung, A., Dekkers, E. and Kobbelt, L.: Character Animation from 2D Pictures and 3D Motion Data, *ACM Trans. Graph.*, Vol. 26, No. 1 (2007).
- [11] Pan, J. and Zhang, J.: Sketch-Based Skeleton-Driven 2D Animation and Motion Capture, *Transactions on Entertainment VI* (Pan, Z., Cheok, A. and Müller, W., eds.), Lecture Notes in Computer Science, Vol. 6758, Springer Berlin Heidelberg, pp. 164–181 (2011).
- [12] Baxter, W., Barla, P. and Anjyo, K.: N-way Morphing for 2D Animation, *Comput. Animat. Virtual Worlds*, Vol. 20, No. 2-3, pp. 79–87 (2009).
- [13] Whited, B., Noris, G., Simmons, M., Sumner, R., Gross, M. and Rossignac, J.: BetweenIT: An Interactive Tool for Tight Inbetweening, *Comput. Graphics Forum (Proc. Eurographics)*, Vol. 29, No. 2, pp. 605–614 (2010).
- [14] Furusawa, C., Fukusato, T., Okada, N., Hirai, T. and Morishima, S.: Quasi 3D Rotation for Hand-drawn Characters, *ACM SIGGRAPH 2014 Posters*, SIGGRAPH '14, New York, NY, USA, ACM, pp. 12:1–12:1 (2014).
- [15] Yeh, C.-K., Song, P., Lin, P.-Y., Fu, C.-W., Lin, C.-H. and Lee, T.-Y.: Double-Sided 2.5D Graphics, *IEEE Transactions on Visualization and Computer Graphics*, Vol. 19, No. 2, pp. 225–235 (2013).
- [16] Sýkora, D., Buriánek, J. and Žára, J.: Sketching Cartoons by Example, *Proceedings of Eurographics Workshop on Sketch-Based Interfaces and Modeling*, pp. 27–34 (2005).
- [17] Liu, X., Mao, X., Yang, X., Zhang, L. and Wong, T.-T.: Stereoscopizing Cel Animations, *ACM Transactions on Graphics (SIGGRAPH Asia 2013 issue)*, Vol. 32, No. 6, pp. 223:1–223:10 (2013).
- [18] Baxter, W., Barla, P. and Anjyo, K.-i.: Compatible Embedding for 2D Shape Animation, *Visualization and Computer Graphics, IEEE Transactions on*, Vol. 15, No. 5, pp. 867–879 (2009).