

# イラストの移動と変形を入力するための スポンジを用いたデバイスの検討

板垣 直生<sup>1,a)</sup> 鈴木 優<sup>1,b)</sup>

**概要:** 本研究では手で変形可能なデバイスを動かすことで、1枚のイラストを移動、変形させ、アニメーションを制作するシステムの実現可能性を検討した。既存のアニメーション制作手法には様々な方法があるが、動きの素材を大量に用意する必要があることや、既存のアニメーション制作ソフトウェアの入力方法は直感的でないなどの課題が存在し、初心者が手軽にアニメーションを制作することの妨げとなっている。そこで、アニメーション制作の知識やスキルを持たない初心者でも手軽にアニメーションを制作できるようにすることを目標とし、描画された1つのイラストを移動、変形させることでアニメーションを制作するシステムの検討を行う。直感的な移動、変形の入力を可能にするために、イラストの形に切り抜いたスポンジを手で動かすことで移動や変形を入力する方法の検討を行い、そのプロトタイプを開発した。

## 1. はじめに

アニメーションを制作するには、少しずつ位置やポーズを変えた画像を大量に用意する必要があり、時間と労力がかかる。そういった手間を省くソフトウェアも存在するが、多くの場合、動きの入力方法は直感的とはいえず、使いこなすには時間がかかる。これらは、アニメーション制作の知識やスキルを持たない初心者が、アニメーションを制作することの障壁となっている。

そこで本研究では、初心者でも簡単に自分の思いどおりのアニメーションを制作することができるシステムの開発を目指し、基礎検討を行った。

## 2. 関連研究

### 2.1 既存のアニメーション制作手法

アニメーションには様々な種類がある。制作手法に着目すると、代表的なものとして、以下のようなものがある。

**平面アニメーション** 少しずつ変化させたイラストを描き、それらを連続で表示させることでアニメーションを制作する。

**立体アニメーション** 人形や粘土で作られた立体物を少しずつ動かして1コマずつ撮影し、それを連続して表示させることでアニメーションを制作する。

**3DCG アニメーション** 3DCG ソフトウェアを用いて、3DCG で作られた 3D モデルを動かすことでアニメーションを制作する。

**モーフィングアニメーション** 2枚の画像を入力として、その間をコンピュータで補完することでアニメーションを制作する。

本研究では、初心者でも思いどおりのアニメーションを制作できるシステムを目指すため、誰でも容易に描くことのできる平面のイラストをアニメーションの素材とし、平面アニメーションを対象とする。

### 2.2 既存の制作手法の課題

初心者が平面アニメーションを制作するとき、次のような課題が存在する。

まず、動きの素材となる少しずつ変化させたイラストを大量に描く必要がある。平面のキャラクターが動くアニメーションを作ろうとしたとき、少しずつキャラクターの位置やポーズを変えたイラストを描く必要がある。アニメーションにしたときに滑らかな動きに見せるには、細かく動かし大量のイラストを描く必要があり、時間と手間がかかる。

また、Live2D[2] や、Adobe After Effects[1] のパペットツールといった、静止画にメッシュを割り当て、変形させることができるアニメーション制作ソフトウェアやツールも存在する。このようなソフトウェアを用いれば、少数の画像からアニメーションを制作できる。しかしながら、このようなアニメーション制作ソフトウェアの多くは、パラメータを操作するか、画面上の静止画をマウスでドラッグ

<sup>1</sup> 宮城大学  
Miyagi University, 1-1 Gakuen, Taiwa-cho, Kurokawa-gun,  
Miyagi 981-3298, Japan

a) p1622011@myu.ac.jp

b) suzu@myu.ac.jp

することで変形を入力するため、操作方法は直感的といえず、思ったとおりの変形を行うためにはソフトウェアの使い方を習得する必要がある。

以上のことをまとめると、アニメーション制作のスキルや知識を持たない初心者が平面のアニメーションを作ろうとしたときには、次のようなことが妨げとなる。

- a) 動きの素材となる大量のイラストを描く必要がある
- b) 多くのアニメーション制作ソフトウェアで使われている入力方法は直感的とはいえない

### 2.3 アニメーション制作支援の先行研究

アニメーションの制作支援の研究は様々なアプローチで行われている。主なものを以下に示す。

Qingkun Su らは動画から抽出した動きをスケッチに適応させることで、初心者にも容易にアニメーションを生成する手法を提案している [5]。この手法ではアニメーションを作るために大量のイラストを描かなければいけないという課題を解決しているが、あらかじめ用意した動画を使用するため、リアルタイムで自分の思ったとおりの動きをつけることはできない。

Connelly Barnes らはイラストを描いて切り抜いた紙を、カメラの前で動かす、移動と回転を読み取ることでアニメーションを制作する手法を提案している [3]。この手法ではユーザの描いたイラストを手で動かすことで直感的に移動、回転といった動きをつけることができるが、変形を伴うアニメーションを制作することはできず、制作できるアニメーションに制限がある。

箕らは手描きのキャラクターを投影したオブジェクトを手で持って移動させ、時間やオブジェクトの動きなどのルールによってイラストを切り替えることで、アニメーションを制作する手法を提案している [6]。この手法ではオブジェクトを手で持って動かすことで直感的な動きの入力を可能にしているが、キャラクターのポーズを変えたい場合は、異なるポーズのイラストをあらかじめ描いておく必要がある。

鈴木らは描画した手描きイラストを身体部位ごとに分離させた画像を生成し、それぞれの画像をユーザの身体のポーズに合わせて回転させることで、描画したイラストの姿勢をユーザの身体の動きに合わせて変える手法を提案している [7]。この手法では、1枚のイラストから異なるポーズの画像を複数生成することはできるが、イラストを自由に変形することはできない。

## 3. 本研究のアプローチ

本研究では、2.2 節に示した 2 つの課題 a) 動きの素材となる大量のイラストを描く必要がある、b) 多くのアニメーション制作ソフトウェアで使われている入力方法は直感的とはいえない、を解決するために次の 2 つのアプローチをとる。

### 3.1 1枚のイラストからのアニメーション制作

a) 動きの素材となる大量のイラストを描く必要がある、という課題に対しては、1枚のイラストを描き、それを移動、変形させることによってアニメーションを制作することで解決する。

### 3.2 スポンジを用いた入力デバイス

b) 直感的な動きの入力ができないという課題を解決するために、動きの入力は手で変形可能なものをデバイスとして用いることとする。

既存のアニメーション制作のソフトウェアのような、マウスやキーボードを用いて画面上のキャラクタを移動、変形させる方法は直感的とはいえない。そこで図 1 のように、動かしたいイラストの形に切り抜いたスポンジを入力デバイスとして用いる。このデバイスの位置と変形をイラストにリアルタイムに反映させる。これにより、ユーザがイメージする位置や変形をそのまま入力することができ、直感的にイラストの移動、変形を行うことができる。デバイスには、加工が容易であること、安価で身近にあることから、スポンジを使用することとした。

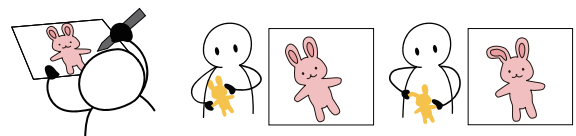


図 1 本研究のアプローチのイメージ図

## 4. スポンジアニメーションシステム

### 4.1 システム概要

本研究では入力デバイスとしてスポンジを用いたアニメーション制作システムを提案する。本システムでは、ユーザはイラストの形に切り抜いたスポンジをカメラの前で動かすことで位置や変形を入力する。システムは変形されたスポンジに合わせて1枚のイラストを移動、変形させ、その位置、変形を一定間隔で記録し、それらを連続して再生することでアニメーションを制作する。

スポンジアニメーションのユーザ側の操作は次のとおりである。まず、ユーザは動かしたいイラストを描き、システムにイラストを入力する。そのイラストの形にスポンジを切り抜く。その後、位置と変形を入力する。位置と変形の入力は、図 2 のように、スポンジを手で持って移動、変形させることで行う。ユーザはカメラの前で、切り抜いたスポンジを手を持ち、曲げる、押し縮める、移動させるなどの動作を行う。

システムの処理の流れは次のとおりである。システムはまず動かすイラストを読み込み、イラストの輪郭を検出する。ここで検出した輪郭はスポンジとイラストを対応づけ



図 2 スポンジを動かして変形を入力する様子

る際に用いる。そして、メッシュ平面を生成し、読み取ったイラストを貼り付ける。その後、ユーザがカメラの前でスポンジを動かすと、システムは web カメラから取得した画像からスポンジの位置と変形を読み取り、読み取った位置と変形を、メッシュを移動、変形させることでイラストに反映させる。なお、システムのプロトタイプは、openFrameworks と OpenCV を用いて開発した。

#### 4.2 イラストの読み込み

本システムでは手描きのイラストを web カメラで撮影することで、動かすイラストを読み込む。ユーザは画面に表示された枠に合わせて描いたイラストを置く。システムは web カメラで撮影した画像から枠線の中の画像を切り取り、動かすイラストとして保存する。また、取得したイラストの最も外側の輪郭を取得する。ここで取得した輪郭はスポンジとイラストの対応づけに用いる。

#### 4.3 スポンジの移動と変形の読み取り

ユーザが操作するスポンジの移動や変形を読み取るには、スポンジを細分化したときの各部分が、取得した画像中のどこにあるのかを判別する必要がある。そこで、リアルタイムで取得される映像の中で、スポンジの各部分のトラッキングを行う。

スポンジの各部分のトラッキングは 2 つの方法の検討を行った。まず、マーカを用いてトラッキングを行う方法を検討したところ、いくつかの問題が見つかった。マーカをスポンジに貼り付ける場合、まず、マーカを貼り付ける手間がかかるのに加え、動かしているうちにマーカが剥がれてしまうという問題がある。また、スポンジにマーカを印刷する場合、スポンジを変形させる際にマーカに歪みが生じ、正しく読み取れなくなる恐れがある。そこで、マーカを使わない方法として、スポンジ表面の特徴点を検出し、その点をトラッキングする方法を検討した。

##### 4.3.1 動きと変形の読み取りの流れ

動きと変形の読み取りは、次の 3 つのフェーズで行う。

- 特徴点の検出
- イラストと特徴点の対応づけ
- 特徴点のトラッキング

##### 4.3.2 特徴点の検出

トラッキングを行うために、まずシステムはスポンジ表

面の特徴点を検出する。特徴点を用いたトラッキングの実装を行うにあたり、まず、スポンジの表面の凹凸を特徴点として認識できるかどうか、どの特徴点検出器が適しているか、を調べるため、複数の特徴点検出器でスポンジ表面の画像から特徴点を検出した。その結果、AKAZE が最も多く特徴点を検出したため、本研究では AKAZE を用いて特徴点の検出を行うこととした。

また、実際に web カメラを用いてスポンジ表面の特徴点を検出しようとしたところ、カメラからの距離によっては特徴点を検出しづらいことがわかった。そのため、スポンジ表面にトラッキングする特徴点を満遍なく検出できるように、スポンジの表面を着色する方法を試みた。

スポンジを黒く着色したのち、特徴点を検出して結果を比較したところ、図 3 のような結果になった。スポンジ 1 が着色をしていないスポンジで、スポンジ 2、スポンジ 3、スポンジ 4 は順に濃さを濃くしながら着色したものである。

特徴点の検出結果を比較したところ、スポンジ 1 ではほとんど特徴点を検出できなかった。スポンジ 2 では着色が薄かったため、スポンジ 1 と変わらない結果となった。スポンジ 3 ではスポンジ 2 よりも特徴点を検出できているが、まだスポンジ表面から満遍なく特徴点を検出できてはいない。スポンジ 4 ではスポンジ表面から特徴点を満遍なく検出できている。そこで、本研究では、スポンジ 4 と同程度にスポンジ表面を黒く着色した上で入力デバイスとして用いることとした。

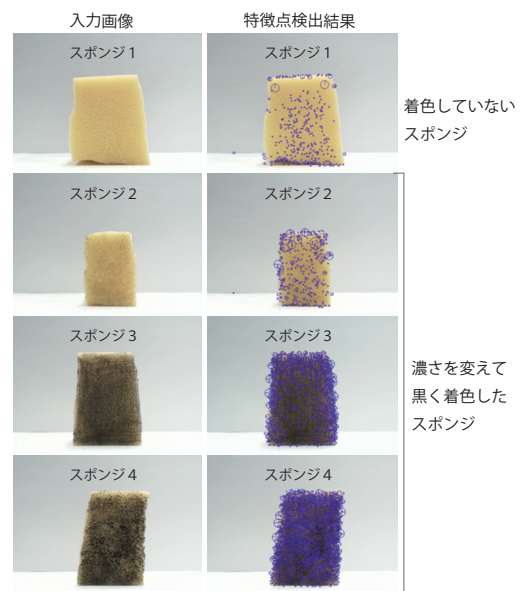


図 3 スポンジを着色して行った特徴点検出の結果

##### 4.3.3 イラストと特徴点の対応づけ

スポンジの移動と変形を動かすイラストに反映させるために、スポンジの各部分と、イラストの対応づけを行う。ユーザは表示されたイラストの輪郭に合わせてスポンジを置く。システムは最初に、変形を行っていない初期状態の

スポンジの画像を取得し、特徴点を検出する。その後、検出した特徴点と、動かしたいイラストの各部分とを紐づける。検出された特徴点のうち、イラストを貼り付けているメッシュの頂点と最も近い位置にある特徴点がメッシュの頂点と紐づけられる(図4)。このとき、イラストの輪郭の内部のメッシュ頂点のみ、特徴点と紐づける。

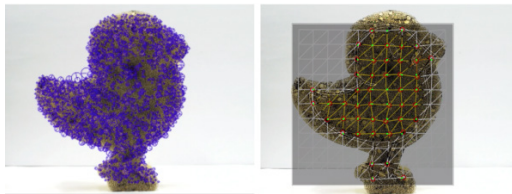


図4 検出した特徴点と紐づけたメッシュ頂点

#### 4.3.4 特徴点のトラッキング

特徴点のトラッキングの方法として、2つのやり方を試みた。1つは特徴点のマッチングを用いた方法もう1つは、オプティカルフローを用いた方法である。

まず、特徴点のマッチングを用いた方法では、毎フレーム取得した画像に対して、特徴点の検出を行い、最初に検出した特徴点とマッチングを行う。この方法では、毎フレーム特徴点の検出とマッチングを行うため、検出した特徴点が多い場合、処理が重くなる。また、スポンジを回転させた場合、特徴点のマッチングが失敗するケースが多いという問題も発生した。

そこで、オプティカルフローを用いたトラッキングを検討した。Lucas-Kanade法のアルゴリズムに従い、最初に特徴点として検出された点を毎フレームトラッキングしていく方法は、特徴点のマッチングを用いた方法よりも処理が高速で、リアルタイムな動きの入力に適している。また、スポンジの回転や変形に対してもトラッキングが外れにくいという利点もあった。

#### 4.4 画像の変形

画像を変形させるために、openFrameworksでメッシュ平面を生成し、そこに動かしたい画像をテクスチャとしてマッピングした。次に、このメッシュの変形のアルゴリズムに従って基準点の情報を基に変形を行う。画像変形の方法はFFD(Free Form Deformation)とIgarashiらの提案したアルゴリズム[4]の2つを試みた。本研究ではイラストの形に切り抜いたスポンジで動きを入力するため、様々な形に対応する必要がある。そのため、FFDは四隅の基準点の位置を指定しなければならないという性質上不適当であった。そのため、任意の2点以上の点を指定することでメッシュ全体を動かすことのできるIgarashiらのアルゴリズムを用いて変形を行うこととした。

#### 4.5 動きの記録と再生

動きを記録するため、フレームごとに基準点の位置を記録し、csv形式で書き出す。動きを再生するときはここで記録した一連の基準点の位置を読み込み、画像に順に適用させることで、記録した動きを再生することができる。

### 5. まとめと今後の課題

本研究では、アニメーションの制作の知識やスキルを持たない人でも手軽にアニメーションを制作できるようにすることを目標とし、手で動かすことのできる物理的なデバイスを用いて、1枚のイラストを移動、変形させることでアニメーションを制作するシステムの実現可能性を検討した。そのなかで、直感的な操作による、イラストへの位置や変形の入力を可能にするために、スポンジを用いた物理的な入力デバイスの検討を行い、プロトタイプの実装を行った。

本手法には以下のような制限が存在する。まず、スポンジの位置や変形をイラストにそのまま反映させているため、イラストに入力できる位置や変形は、物理的に操作可能なものに限られる。また、トラッキングの特性上、スポンジを裏返してしまうとトラッキングができなくなるため、入力できる位置や変形は、反転なしの移動、変形に限られる。今後はこれらの解決を目指す。

#### 参考文献

- [1] After Effects. <https://www.adobe.com/jp/products/aftereffects.html>.
- [2] Live2D. <https://www.live2d.com>.
- [3] Connelly Barnes, David E. Jacobs, Jason Sanders, Dan B Goldman, Szymon Rusinkiewicz, Adam Finkelstein, and Maneesh Agrawala. Video Puppetry: A Performative Interface for Cutout Animation. *ACM Trans. Graph.*, Vol. 27, No. 5, pp. 124:1–124:9, 2008.
- [4] Takeo Igarashi, Tomer Moscovich, and John F. Hughes. As-Rigid-As-Possible Shape Manipulation. *ACM Trans. Graph.*, Vol. 24, No. 3, pp. 1134–1141, 2005.
- [5] Qingkun Su, Xue Bai, Hongbo Fu, Chiew-Lan Tai, and Jue Wang. Live Sketch: Video-Driven Dynamic Deformation of Static Drawings. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pp. 662:1–662:12, 2018.
- [6] 笈康明, 山岡潤一, 赤塚大典, 苗村健. Tablescape animation:手描きキャラクターと桌上オブジェクトを用いた即興的アニメーション制作環境. 日本バーチャルリアリティ学会論文誌, Vol. 15, No. 3, pp. 389–396, 2010.
- [7] 鈴木優, 林凌平. 身体部位の分離画像の生成と関節点の自動設定による手描きイラストの姿勢制御. 情報処理学会論文誌, Vol. 60, No. 11, pp. 1961–1969, 2019.