

直感的なメッシュ操作によるアニメーション作成ツールの提案

加藤公久†、米倉達広††、鎌田 賢††、岡本 秀輔†††

†茨城大学理工学研究科 ††茨城大学工学部情報工学科 †††成蹊大学理工学部情報科学科

概要 Web ページやプレゼンテーション用の簡易なアニメーションを作成するためのツールの要求は以前から存在し、様々なツールが提案されてきたが、生成結果の予想のしやすさについては十分に議論されていなかった。我々は $r-\theta$ 座標上で動的にメッシュを構築しメッシュの存在をユーザに意識させずにワーピングさせる技法を考案した。この手法はアニメーションの生成結果がユーザにとって予想しやすく、尚且つある程度複雑な変形にも対応することができる。これを SPHINX と名づけ提案する。

Proposal of animation making tool by intuitive mesh operation

Takahisa KATO†, Tatsuhiro YONEKURA††, Masaru KAMADA††, Shuusuke OKAMOTO†††

†Graduate School of Science and Engineering, Ibaraki University

††Department of Computer Information Sciences, Faculty of Engineering, Ibaraki Univ.

†††Faculty of Science and Technology Seikei University

Abstract.

The demand of the tool to make simple animation for the Web page and presentation exists since before, and various tools have been proposed. However, the easiness of expectation of the quality of generated animation was not discussed enough.

We developed a new method of creating animation. Our method is designed for ease of use and allows user to expect animation generated. We call this proposing tool Schematic morPHINg eXpression (SPHINX). SPHINX dynamically places meshes on $r-\theta$ coordinate system. And do not need user to consider about how to place meshes.

1.はじめに

Web ページやゲームなどに用いる簡易な二次元のアニメーションを作成する手法は過去様々な検討され、Macromedia Flash[1]などがよく用いられている。Macromedia Flash のモーショントゥイーン機能を用いればコンテンツ製作者は絵や文字を回転、平行移動、拡大縮小させることができ、シェイプトゥイーン機能を用いて複雑に変形させることもできる。しかし、これらの UI は不慣れなユーザにとっては直感性に欠け、満足のいくものを作成できるようには長時間の練習を要する。

中京大の加藤、高橋、柴山らは絵をユーザの描いた効果線に沿って回転、平行移動、拡大縮小させる手法を提案した[2]。この手法を用いればユーザは直感的にアニメーションを作成できる一方、動かす絵そのものを複雑に変形させることは難しい。絵を複雑に変形させる手法として代表的なのがモーフィングと呼ばれる技法である。モーフィングは入力された 2 つの画像の中間画像を生成し、なめらかなアニメーションにするための技法の総称であり、様々な研究が報告されている。輪郭抽出などを行わずに画像を

そのままモーフィングさせるための技法としてメッシュワーピングがある[3]。メッシュワーピングでは画像をメッシュと呼ばれる領域に分割し、ユーザは1つのメッシュが画像の1つの部位を収めるよう設定する(図.1)。開始フレームと終了フレームで対応する部位を収めたメッシュがあることから、開始フレームと終了フレームで対応する画素がわかる。これを基に線形変換させることにより中間画像を得る。この方法は容易に1画像からアニメーションを生成することにも利用でき、比較的高品位なものが作成できる。しかし、従来の手法ではメッシュの設定は熟練を要する。本研究では、この点を踏まえてユーザフレンドリーなアニメーション作成ツールの提案をしていく。

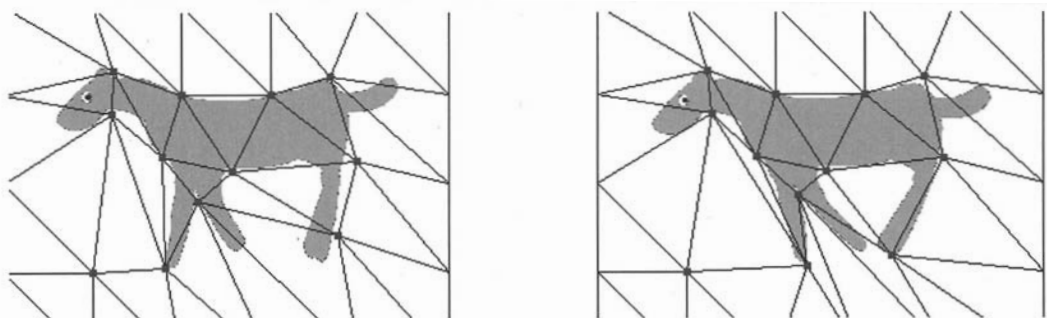


図.1 メッシュワーピングでのメッシュ設定

2.SPHINX

通常、メッシュワーピングは $x-y$ 領域で行われるが、この方法は前述のように操作性が悪い。そこで我々はメッシュワーピングの考え方を通常の $x-y$ 領域で用いず、 $r-\theta$ 領域で用いたツールを開発した。以下これを SPHINX(Schematic morPHing eXpression)と呼ぶ。SPHINX ではユーザにメッシュの存在を意識させることなくメッシュワーピングの原理を用いてアニメーションを作成させることができる。

図.2 は飛行機を平行移動させるシーンである。このシーンでは、ユーザが入力した飛行機を右から左に移動させるために、入力画像と移動先の中心点の指定をすることで飛行機を平行移動させるアニメーションを作成することができる。白抜き矢印の根本が入力画像の、指し示され先が移動先の中心点である。画面に薄く表示されているのはこの中心点移動を行った場合生成される最終フレームとなっている。

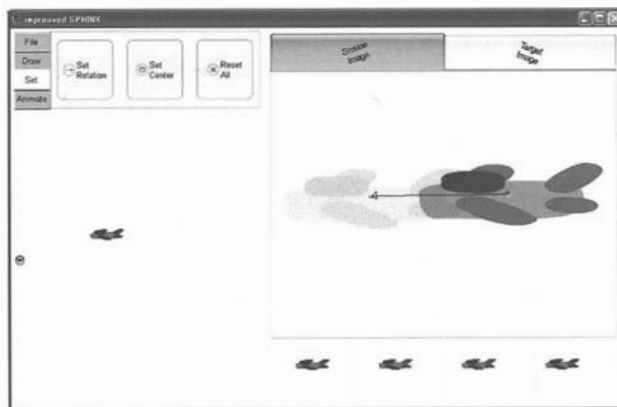


図.2 飛行機が飛ぶシーン

図.3 は人がお辞儀をするシーンで、今度は中心の移動先を指定せずに、頭部と右側の足の移動先を指定している。黒塗りの矢印の根本が頭部にあり、先端がその右側下方に向いており、この矢印の指定により頭を下げる変形の指示を与えている。また、右側の足にも矢印があり、始点と終点と同じ場所であり、これは右側の足を動かさないという指示になる。これによって作成されたアニメーションが図.4 である。

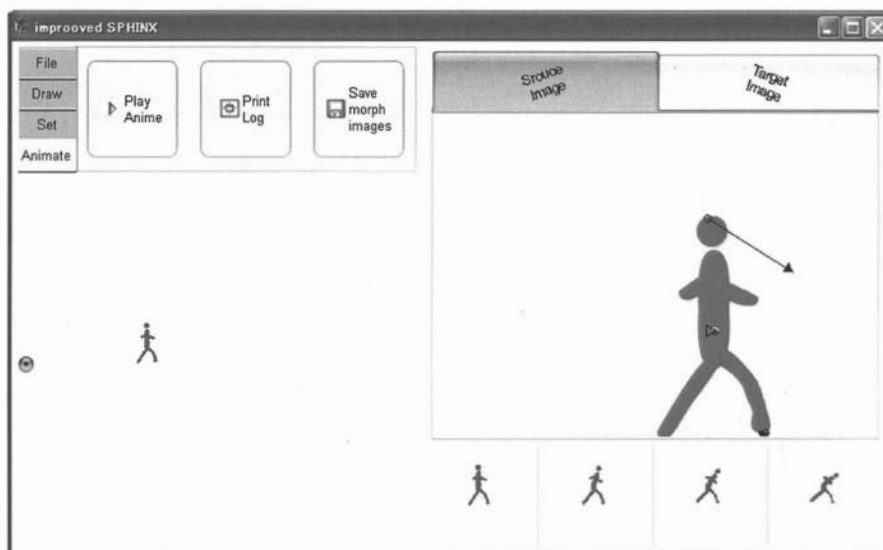


図.3 人がお辞儀をするシーンの作成場面

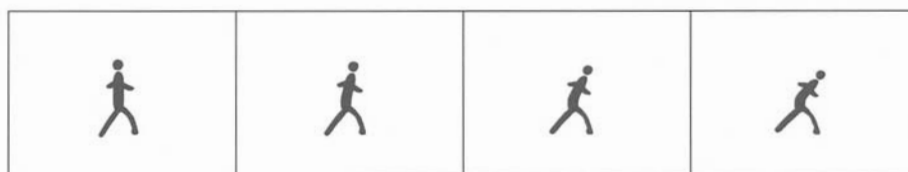


図.4 人がお辞儀するアニメーション

3.SPHINX のアルゴリズム

通常のメッシュワーピングでは開始フレームで各部位をメッシュとして分割し、終了フレーム上でも対応するメッシュが対応する部位に設置する。これにより、メッシュ内の位置を基に各画素の RGB 値を一意に定めることができる。しかし、この手法ではメッシュの設置に時間がかかると共に、慣れも必要となってくる。G.Cong らは予め与えられた輪郭情報をその重心からの距離と角度に変換して(図.5 参照)、それを基に対応点を自動的に抽出してモーフィングを行う手法を提案している[4]。また、高畑らは同様の手法を人物などのキャラクターに用いた例を報告している[5]。図.6 に示すように、長方形の画像は $r-\theta$ 座標系では二つの山となる。右下のように2つの山を近づけていくと $x-y$ 座標系では左下のように長方形が折れ曲がりくの字のような形になる。

我々の提案する SPHINX はこの性質を用いて、輪郭情報を用いずに $r-\theta$ 座標系で画像をメッシュ分割してワーピングを行い、自在に人型の画像などをアニメーションさせることを可能とした。この方式は平行

移動、回転、拡大縮小ならびに簡易な変形を4~6クリックで作成することを可能にする。

この手法は ユーザにとってはアニメーションがどう生成されるか予想が容易であり、この特性は作業時間の短縮に大いに役立つものと考えられる。

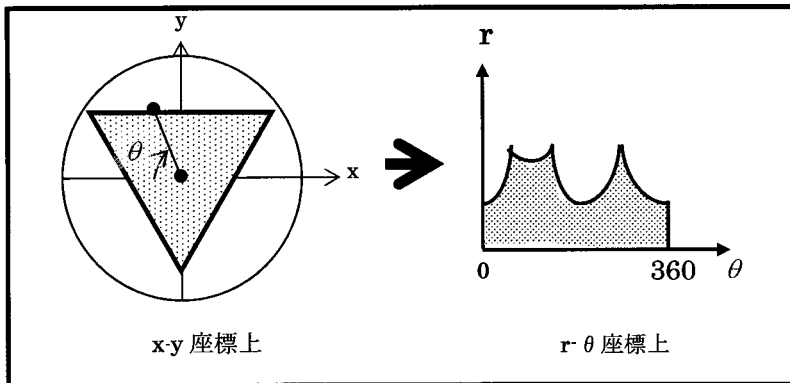


図.5 x-y 座標上の画像の r-θ 座標上の表現

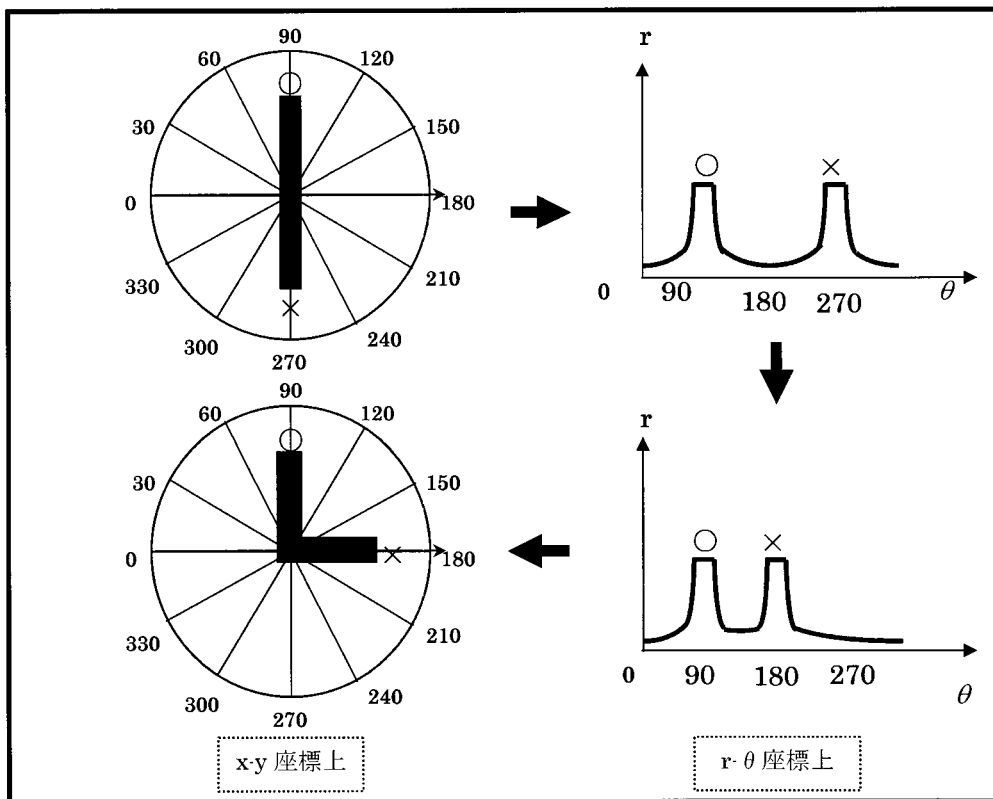


図.6 r-θ 領域での画像の変形

3.1.動的メッシュ分割

SPHINX は画像が入力されると、背景をとり除いた画像の重心を設定するとともに画像を重心からの距離 r と角度 θ に変換した $r-\theta$ 座標系に変換する。そして、ユーザが入力した 3-4 組の対応点の入力画像の側の座標を基に入力画像をメッシュに分割する (図.7)。下図.の例では、点 1 と点 2 が入力され、点 1 と点 2 の $r-\theta$ 座標をそれぞれ (r_1, θ_1) (r_2, θ_2) とすると以下の二つのメッシュが生成される。
 $[(r_1, \theta_1), (0, \theta_1), (0, \theta_2), (r_2, \theta_2)]$ (mesh 1), $[(0, \theta_2), (r_2, \theta_2), (r_1, \theta_1 + 360^\circ), (0, \theta_1 + 360^\circ)]$ (mesh2)

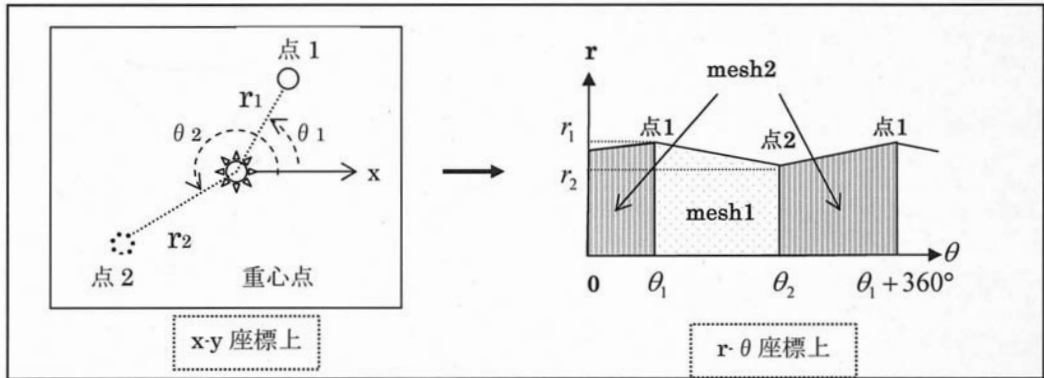


図.7 点 1、点 2 が入力されて生成されるメッシュ

3.2.メッシュ内部の画素の移動

同様に移動先にもメッシュを生成するために、対応点の移動先の点もメッシュの端点として利用される。矢印の対として入力されているために入力画像のメッシュと移動先のメッシュの数と対応関係は自明であるために(図.8)、入力画像のメッシュはアニメーションの途中で徐々に変形していき、最終的に移動先のメッシュの形になって行くと共に内部の画素も同様の変形をしていく。

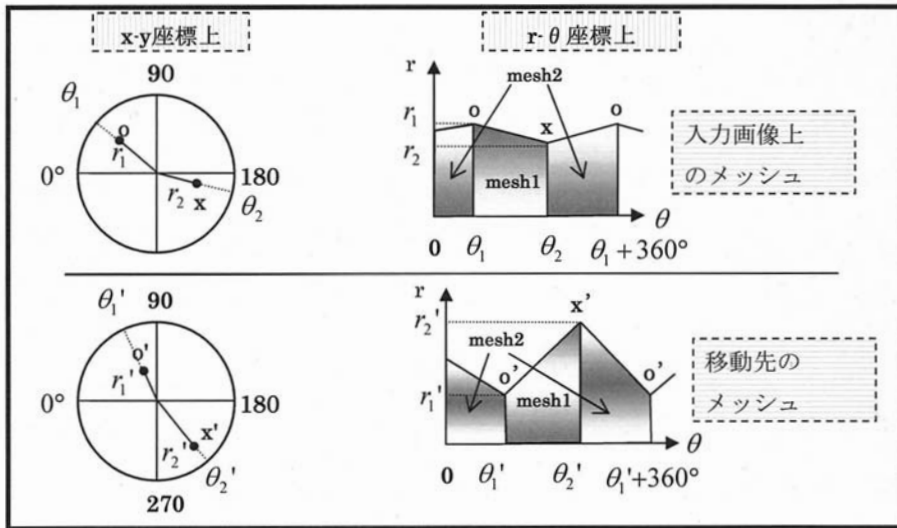


図.8 入力画像、ターゲット上のメッシュ

このとき、入力画像で mesh1 内の $(r, \theta) = (r_a, \theta_a)$ にある画素は最終フレームでの位置 (r'_a, θ'_a) は下式で与えられる。

$$m_\theta = \frac{\theta_a - \theta_1}{\theta_2 - \theta_1}, m_r = \frac{r_a - r_1}{r_2 - r_1},$$

$$r'_a = r_1 + (r_2 - r_1) \times m_r,$$

$$\theta'_a = \theta_1 + (\theta_2 - \theta_1) \times m_\theta$$

これにより、入力画像のメッシュ内の移動先が一意に求まる。中間フレームの生成も同様である。

4.まとめ

本研究では、簡易な変形を伴うアニメーションを単純な操作で作成できるツールを提案した。ユーザが意識しないようにメッシュを動的に生成し、入力画像が平行移動、回転、拡大縮小、変形するアニメーションを作成することができる。本手法ではユーザが操作によってどのようにアニメーションが生成されるか予想することが容易であり、単純なアニメーションを手早く作成するのに向いていると考えられる。

参考文献

- [1] Adobe Corporation, "Macromedia Flash 8", <http://www.adobe.com/support/flash/>
- [2] Yoshikazu Kato, Etsuya Shibayama, Shin Takahashi, "Effect lines for specifying animation effects.", Proceedings of The IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC'04), pp. 27-34, September 2004, Rome, Italy.
- [3] George Wolberg, "Recent Advances in Image Morphing", Computer Graphics International, 1996. Proceedings, Jun 1996
- [4] G.Cong, B.Parvin, "A New Regularized Approach for Contour Morphing", Computer Vision and Pattern Recognition, Vol. 1, pp. 1458-1463, 2000
- [5] Ayako TAKABATAKE, Taichi NOMURA, and Hironobu FUJIYOSHI, "Shape Tweening by Contour Based Correspondence Computation", MIRU2005 (in Japan), pp. 715-720, 2005.7