

スクリーン座標における変形を考慮したアニメ用視点依存モデル

佐熊 敬介[†] 斎藤 隆文[‡][†]東京農工大学 工学部情報工学科[‡]東京農工大学 大学院生物システム応用科学府

1. はじめに

手書きアニメーションに 3DCG を利用する例が増えてきている。その主な理由として、作業の簡略化と品質の均一化が挙げられる。しかし 3DCG に用いられる一般的な 3D モデルでは、手書きアニメーションを忠実に再現するのは難しい。なぜなら手書きのイラストは多くの場合、視点に依存する歪みや誇張を含んでいるからである。一般的な 3D モデルは視点が変わっても形状は不変であるため、イラストの持つ視点依存の歪みや誇張を表現することはできない。このような理由により、手書きアニメーションに一般的な 3D モデルを併用した場合、不自然な結果になってしまう。

この問題に対し、Paul Rademacher は 3D モデルの形状を視点に合わせて変形する手法(View-dependent geometry)[1]を提案した。この手法はまず通常の 3D モデルを作成し、それから特定の視点において変形させた特定の形状を設定する。レンダリング時に現在の視点によって特定の形状を補間することによって実現している。Alec Rivers らはその手法を 2D ベクタ画像に適用する 2.5D Cartoon Models[2]という手法を提案した。これにより、3D モデルを作成せずとも、視点依存の歪みや誇張を含んだイラストをあらゆる角度から見るできるようになった。しかしこの手法は視点の変化に対してスクリーン座標における変形の動きを考慮しておらず、表現に制限を持っている。

本研究では、これらの手法を参考に、スクリーン座標における変形を考慮したアニメ用視点依存モデルを提案する。スクリーン座標においてどのように変形するかについて着眼点を置き、手法を改善する。手書きアニメーションと併用することを目的に、違和感のない結果を得られるようにする。

2. 提案手法

本研究の目的は視点依存の歪みや誇張を含み、あらゆる視点から見るができるモデルを提案することである。これを視点依存モデルと呼ぶ。本研究では横方向のみに問題を限定して視点依存モデルを構築する。また、本研究では 2.5D Cartoon Models のように 2D ベクタ画像を使用して視点依存モデルを表現する。3D モデルを用いる手法では、ユーザに作成する手間がかかり、また 3DCG に関する深い知識を要するからである。2D ベクタ画像であればイラストを描くのと同一ような操作で作成することが可能であると考えられる。

まず、異なる視点から見た場合における複数の 2D ベクタ画像を入力に取る。このそれぞれの視点(キー視点)と 2D ベクタ画像の対はユーザに輸入してもらう。1つの 2D ベクタ画像は複数のストロークで構成される。システムとユーザの入力の簡単化のため、それぞれのストロークは他の 2D ベクタ画像においても同じ構造を持つものとする。それぞれ決められた視点の間において、ストロークの形状及び位置は単純に二次元的な補間が行われる。2.5D Cartoon Models では形状と位置は別個に扱っていた。本研究ではストロークの形状と位置はストロークの制御点で統一的に扱うことにする。単純化することで、スクリーン空間上でどのようにストロークが動くのかユーザに想像しやすくする。

二次元的な位置の補間だけでは 3DCG のようにあらゆる角度から見るということはできない。視点の位置が回転するという事は、あるストロークは前から後ろへ、別なストロークは後ろから前へ回転するという事である。それぞれのストロークは前後関係によってお互いに遮蔽されることになる。この前後関係を表すためにそれぞれのストロークについて Z 値(奥行き)を制御しなければならない。本研究ではストロークの Z 値はストロークにそれぞれ別な値を持たせて視点間で補間する。これは通常の 2D ベクタ画像にはない情報である。これもまたユーザに輸入してもらう。

View-Dependent Model for Cartoon Animation Considering Deformation in Screen Space

[†]Tokyo University of Agriculture and Technology

[‡]Graduate School of Bio-Applications and Systems Engineering, Tokyo University of Agriculture and Technology.

2.1 ストローク

本研究で使用するストロークは 3 次ベジェ曲線とする。これはベクタ画像に広く用いられており、滑らかな曲線を描くことができる。3 次ベジェ曲線は 4 つの制御点から成る。このうち両端以外の制御点は曲線が通らない。これを 1 つのセグメントとし、複数のセグメントを接続することで 1 つのストロークとする。各セグメントの端点は、前後のセグメントの端点と一致する。ストロークは開くか閉じるかを選択することができる。

簡単のため、曲線の通過する点のみをユーザに指定してもらう。各セグメントの曲線の通過しない 2 つの制御点の位置は、セグメントが滑らかに接続されるようにその端点における速度が一致するという条件の下、一意に求められる。ただし開いた曲線の端点においては速度が 0 になるという条件を設定する。この通過点の座標のみを補間する。これにより、ユーザは直感的に形状を設定できる。

ストロークは太さと色、さらに塗りつぶしの色の情報を持つ。これにより、ストロークで囲まれた領域を表現することができる。

2.2 補間

補間の際は現在の視点に最も近いキー視点を選択する。本研究では正面から横方向に限定する。そのため最も近いキー視点は、現在の視点を挟む 2 つの視点となる。この 2 つの視点と対応する 2D ベクタ画像を用いて補間する。2D ベクタ画像同士の補間はそれぞれの構造が同一であるため、ストロークの各通過点の座標と Z 値の補間のみで実現できる。生成されるストロークの各通過点の座標を P 、2 つのキー視点におけるストロークの各通過点の座標を P_1 、 P_2 とすると式(1)で書き表すことができる。

$$P = aP_1 + (1-a)P_2 \quad (1)$$

ここで a は補間係数とする。これは 2 つのキー視点間における現在の視点の線形補間である。同様に Z 値も求められる。

実際に作成した例を図 1 に示す。(a)、(c)及び(e)がユーザによる入力である。その間の視点において補間により得られた結果が(b)及び(d)である。(c)と(d)の間で右目と顔の Z 値の大小が逆転し、右目が遮蔽されている。

3. おわりに

本研究では、2D ベクタ画像においてストロークの通過点をそのまま補間することにより、直感的な視点依存モデルを提案した。今後の課題として、横方向のみならずあらゆる視点から見られるように拡張することが挙げられる。ま

た、実際の手書きアニメーションと比較し、評価と改善を行いたい。

参考文献

- [1] Paul Rademacher, View-Dependent Geometry, *Proc. SIGGRAPH '99*, pp.439 - 446 (1999).
- [2] Alec Rivers, Takeo Igarashi, and Frédo Durand, 2.5D Cartoon Models, *Proc. SIGGRAPH '10*, Article No. 59 (2010).

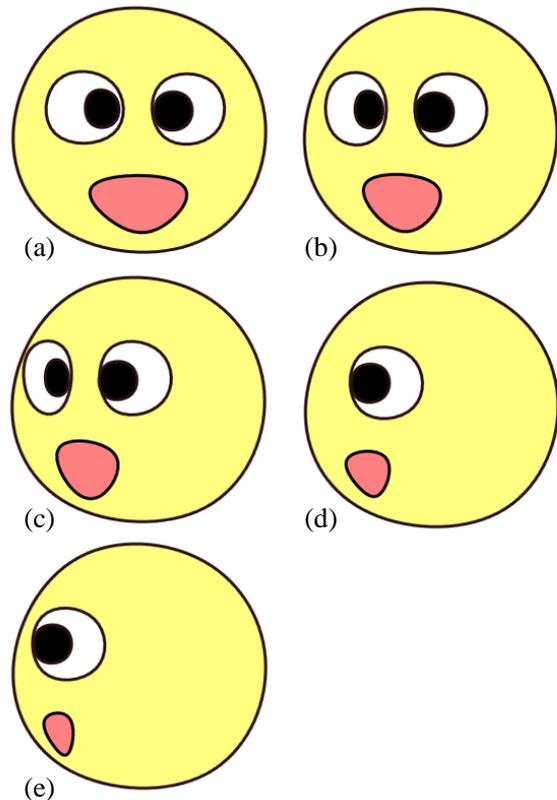


図 1. 作例