

アニメーションの人体動作の3次元復元

幸村琢

村井将彦

品川嘉久

概要

近年、テレビ番組、映画、ゲームにおいて、3D キャラクターアニメーションに対する要求は高まりつつある。現在、セル・アニメーションを描ける人口は多い一方で、3次元漫画アニメーションを扱える人口はまだ少ない。本論文では、いまある2次元アニメーションのキャラクターから3次元的な動きを作り出す手法を提案する。本手法を用いることにより、これまでの2Dセルアニメーション再利用することができる。得られた3次元動作を再び2次元平面に投影することにより、様々な角度からのセルアニメーションを生成することができる。

1 はじめに

近年、3Dアニメキャラクターの需要はテレビ番組、映画、ゲーム等において高まってきている。しかしながら、その場合、キャラクターモデルは2次元平面上ではなく、3次元空間内で制御されなければならない。3次元空間内で人や動物の姿を制御することは2次元平面で動きを描くよりもはるかに難しい。人体の自由度が増えるにつれ、姿勢の制御はより複雑になり、結果としてアニメーターはキャラクターの動きを生成するのにトライ・アンド・エラーを繰り返す必要が出てくる。他方で、漫画とセルアニメーションの世界でコンピュータを使って、アニメーターをサポートし、人、建物、背景を描く需要が高まってきている。しかしながら、人物等のキャラクターの姿を描くのを支援するシステムはまだあまりない。本論文では、2次元セルアニメーションにおけるキャラクターの動きを、3次元的にキャプチャする方法を提案する。

キャラクターの姿勢を得るため、キャラクターが木構造で構成された剛体の体であると想定する。また、

キャラクターの3次元モデルとそのスケルトンも用意しておく。ユーザはそれぞれのフレームで体の関節にあたるピクセルをマウスでクリックする。これらの2次元座標の情報と、スケール値を用いることによって、体の3次元座標が計算される。異なるスケール値を異なるセグメントに用いることにより、セルアニメーション等でよく用いられる誇張表現の三次元化も可能となる。

結果として、オリジナルのセルアニメーションに対応したキャラクターの3次元モデルを最小限の努力で生成することができ、3Dゲーム、インターネットコンテンツ、3Dメディアなどの様々なアプリケーションで利用できる。

2 関連研究

人体の姿勢を画像から判定するシステムは古く空多数の研究が行なわれてきている [1, 4, 2]。これらの研究は本当の人間の動きを実写映像から抽出することが目的であるため、誇張的な表現として変形を多く用いているセルアニメーションのような人工的な映像には直接適用することが出来ない。

手書きの2次元からユーザが3次元のシーンを創りだすことができるようになる研究は多々あるが、五十嵐 [3] はすばやく簡単にスケッチインターフェイス「Teddy」システムを提案した。しかし、「Teddy」はぬいぐるみのような形状を生成するのに特化したシステムであり、人体の姿勢を判定することには適していない。

ディズニーなどのたくさんのセルアニメーション映画が、アニメーターの負担を減らすために3次元グラフィクスを用いているが、アニメーターがキャラクターの動きをつくるのを支援する技術はほとんどない。

3 アルゴリズム

3.1 体のモデル

はじめにキャラクターのスケルトンと 3D ポリゴンモデルが定義される。スケルトンは体の位相的な構造を表現する。図 3.1 に二種類のスケルトンとポリゴンモデルを示す。それぞれの関節の 3 次元位置を決定す

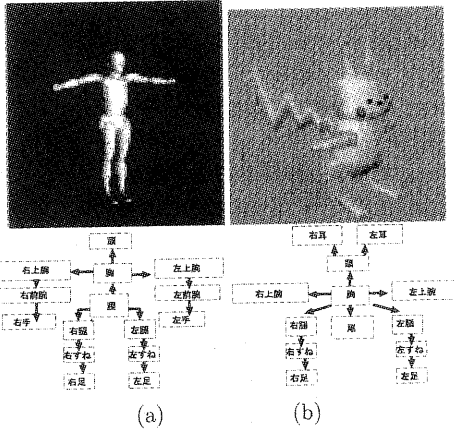


図 1: 位相的構造を伴った体のモデル: (a) 普通の人体モデルと (b) 漫画キャラクター

るために、それぞれの関節間の距離が一定であるという仮定を用いる。はじめに、ユーザは画面上の体の関節があるところのピクセルをマウスでクリックする。一般的な人型のキャラクターの場合、ユーザがクリックする関節を図 2 に示す。このクリックする過程は体が木構造であると仮定した場合、その木の根から葉の関節まで行われる。ひとつの関節をクリックする際にユーザは親関節と比較してその関節がカメラに近いか遠いかを指定しなくてはならない。(図 3) 結果として、体の 3 次元姿勢が最小限の努力で得られる。定義されたこれら関節位置は $(u_1, v_1), \dots, (u_n, v_n)$ と表される。ここで n は体の関節の数である。クリックする過程のウィンドウが図 4 である。この研究において、カメラのシーンは 3 次元世界の平行投影と仮定する。そのため、ユーザはカメラの焦点距離を指定する代わりにスケール値を指定する。スケール値 s を使って、3 次元座標系 (x, y, z) での位置とスクリーン (u, v) に投

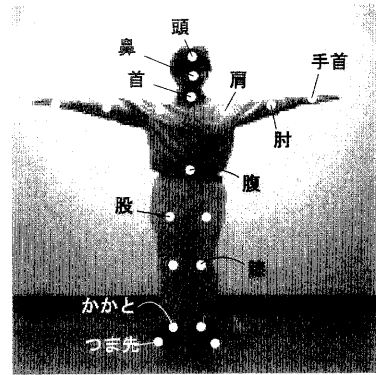


図 2: 普通の人間モデルにおけるクリックする関節の位置

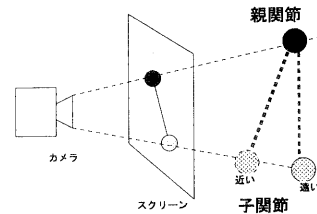


図 3: 3 次元世界における 2 つの関節位置の候補

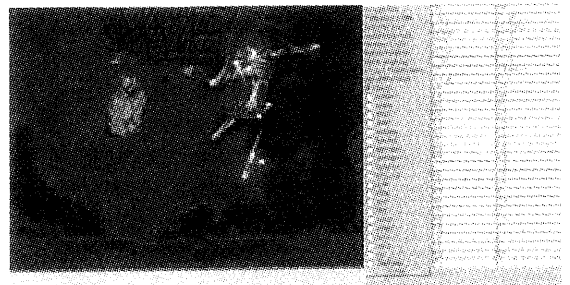


図 4: 関節の 3 次元座標を得るために今回実装されたクリックするプログラム

影された点との関係が以下の式で表される。

$$x = su, y = sv, z = \text{arbitrary value} \quad (1)$$

それゆえ、世界座標系の関節の3次元位置を得るため、 z 座標とスケール値 s を指定することが必要である。

ひとつの関節の3次元位置が既知で関節間の距離が一定であるという仮定をした場合、次の関節の位置が世界座標系の2つの点に限定される。(図3)。関節 i の3次元座標が (x_i, y_i, z_i) 、この関節の子ノードである次の関節 $i+1$ のスクリーン上の座標を (u_{i+1}, y_{i+1}) と仮定する。そのとき関節 $i+1$ の3次元座標 $(x_{i+1}, y_{i+1}, z_{i+1})$ は次のように求める。カメラは平行投影としているので、 x, y は座標は次のように計算される。

$$x_{i+1} = su_{i+1}, y_{i+1} = sv_{i+1}. \quad (2)$$

そのため未知の値は z_{i+1} だけである。もし、関節 i と $i+1$ の間の距離が一定だと仮定すれば、次の拘束条件を用いることができる：

$$(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2 = r_i^2 \quad (3)$$

ここで r_i は関節 i と $i+1$ との距離である。結果として、 z_{i+1} は

$$z_{i+1} = z_i \pm \sqrt{r_i^2 - ((x_{i+1} - x_i)^2 - (y_{i+1} - y_i)^2)}.$$

によって求めることができる。もしカメラに関節 $i+1$ が関節 i より近ければ、 z_{i+1} は

$$z_{i+1} = z_i - \sqrt{r_i^2 - ((x_{i+1} - x_i)^2 - (y_{i+1} - y_i)^2)}.$$

となり、また遠い場合は

$$z_{i+1} = z_i + \sqrt{r_i^2 - ((x_{i+1} - x_i)^2 - (y_{i+1} - y_i)^2)}.$$

となる。こうして、すべての関節の3次元座標が体の根の z 座標 z_0 を使って計算できる。ここで j 番目のキーフレームにおける関節 i の z 座標を $z_i^j = f_i^j(z_0)$ という関数によって定義する。

次に体の根である z_0 の z 座標 z_0 を得る方法であるが、以下の手法が考えられる。

- 足か手の座標が動きの中で固定されている
- 速度情報を用いる
- スケール値を見積もる

- ユーザが指定する

もし、一連の動きで足が同じ点に固定されていれば、 z_0 は足の座標が同じ場所にくるように指定すればよい。つまり j 番目のフレームで、右足の位置が z_r^j であった場合、 $j+1$ 番目のフレームで z_r^{j+1} を計算するのに $z_r^{j+1} = z_r^j$ という拘束条件を用いればよい。この拘束条件を繰り返すことにより、ユーザはどちらかの足が常に地面についている歩行運動などの体の位置を計算できる。

次に速度を用いる手法であるが、もし前の2つのキーフレームにおける z_0 の値、 (z_0^j, z_0^{j+1}) がわかっているれば、 z_0^{j+2} を

$$z_0^{j+2} = z_0^{j+1} + \frac{z_0^{j+1} - z_0^j}{t_{j+1} - t_j} (t_{j+2} - t_{j+1}) \quad (4)$$

によって得ることができる。ここで、 t_j と t_{j+1} がキーフレーム j と $j+1$ の時間である。

次にスケール値から z_0 を得る方法であるが、もし、 j 番目のフレームで体の平均的なスケーリングパラメータが s^j で、次のキーフレームで s^{j+1} だった場合、 z_0^{j+1} は

$$z_0^{j+1} = \frac{s_1}{s_0} z_0^j. \quad (5)$$

とすることができる。

最後の問題はスケール値 s をどうやって決定するかである。 s はクリックされた剛体上の3つの点から計算することができる。例えば、もし頭が剛体であると仮定した場合、首、鼻、頭のてっぺんの画面上での2次元座標が $(u_{neck}, v_{neck}), (u_{nose}, v_{nose}), (u_{top}, v_{top})$ であるとすると、これらの点の3次元座標 $P_{neck}, P_{nose}, P_{top}$ は次のようになる。

$$P_{neck} = (su_{neck}, sv_{neck}, z_{neck}) \quad (6)$$

$$P_{nose} = (su_{nose}, sv_{nose}, z_{nose}) \quad (7)$$

$$P_{top} = (su_{top}, sv_{top}, z_{top}) \quad (8)$$

これらの点の間の距離はあらかじめわかっているものとして、次の3つの方程式を解いて、未知の値 $s, z_{neck}, z_{nose}, z_{top}$ を求める。

$$\|P_{neck} - P_{nose}\| = r_1^2 \quad (9)$$

$$\|P_{nose} - P_{top}\| = r_2^2 \quad (10)$$

$$\|P_{top} - P_{neck}\| = r_3^2 \quad (11)$$

$z_{neck}, z_{nose}, z_{top}$ がこれらの式で $z_{neck} - z_{nose}(= z_1), z_{nose} - z_{top}(= z_2), z_{neck} - z_{top}(= z_1 + z_2)$ という形式でのみ現れるので、未知の値の数は実際は3つで、それゆえ、3つの方程式から s が計算される。

キャラクターの強い感情や大きな動きの勢いを強調するために体を変形することはよく漫画アニメーションに使われる。そういう場合は部分的にサイズを変えるために体の異なる部分に異なるスケール値を使ってやればよい。

前述したように、各セグメントのスケール値を決定するためにセグメントは少なくとも3つのクリックする点をもたなければならない。普通の人体モデルでは、足、頭、胸、腰は3つ以上の点を含む。

しかしながら、クリックする過程でのあいまいさとセグメントが剛体であるという仮定の不正確さから、この方法で得られるスケール値はいつもアニメーターの要求を満たしているわけではない。それゆえに、本システムではこれらの値は初期値として取り敢えず用いられ、アニメーターが適切な値を選べるようにしている。

4 実験結果

本論文で提案された手法を様々な漫画キャラクターに適用し、有効性を示した。以下のテレビや映画のカットが使われた。

- ルパン三世 (Japanese TV program)
- トムとジェリー (US Movie)
- ポケットモンスター (Japanese Movie)

「ルパン三世」において、キャラクターは本物の人間に近く描かれており、関節位置は簡単に指定できる。一方、「トムとジェリー」と「ポケットモンスター」はキャラクターの体が「ルパン三世」より変形している。

4.1 ルパン三世

テレビ番組の「ルパン三世」のフィルムの3つのカットがキャラクターの動きをキャプチャするのに使われた。車の上を転げ落ちるはじめのひとつは、図5のように適切にキャプチャされた。次に、図6のように、

銃で撃った後逃げる動きが得られた。最後の図7はドアに突入して、落下する動きである。

4.2 トムとジェリー

「トムとジェリー」からここではひとつのシーンが使われた。トムが驚いて跳び上がり走り出す。キャプチャされた動きは図8である。ここでは人体モデルがフィッティングの過程で使われている。

4.3 ピカチュウ

「ピカチュウ」の体は人間とは異なる位相的な構造なので、ここで使われる体のモデルは「ピカチュウ」(図9のひとつ)に特化した。このシーンではピカチュウは敵から逃げている。結果は図9である。

5 ディスカッション

本手法では1つのセグメントにおけるスケール値を計算するのに少なくとも3つの点をクリックしなければいけない。この状態はセグメントのスケール値を計算できる可能性を極端に限定し、結果として、体の形がよく変わったり、クリックする点が少なかったりする漫画キャラクターに本手法を適用するとあまりよい結果が得られない。例えば、変形の多い「ピカチュウ」は「ルパン三世」より動きが滑らかではない。この問題を解決するためにはクリックされた点からでなく、スクリーン上のセグメントのおよその面積からセグメントのスケール値を計算する方法が考えられる。

6 結論

本論文では、2Dセルアニメーションキャラクターから3Dモーションキャプチャーをする方法を提案した。我々の手法により、古いセルアニメーションを再利用して、3Dキャラクターアニメーションを最小限の努力と訓練で創り出すことができる。将来課題としては、アニメーターによる入力を増やすことなく、すべてのセグメントで異なるスケール値を自動的に計算するようにすることがある。セルアニメーションに特化した動き追跡もまた興味深いトピックである。

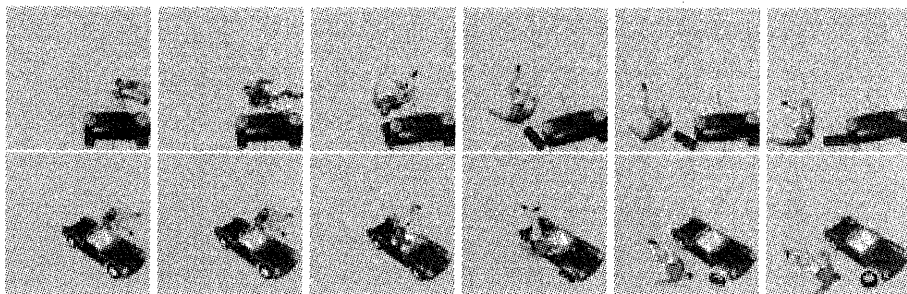


図 5: 車に飛び乗り転げ落ちるルパン (ルパン三世より)

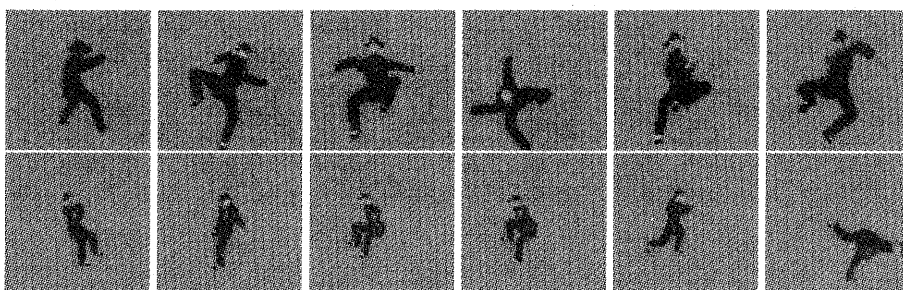


図 6: 撃って逃げる次元大輔 (ルパン三世より)

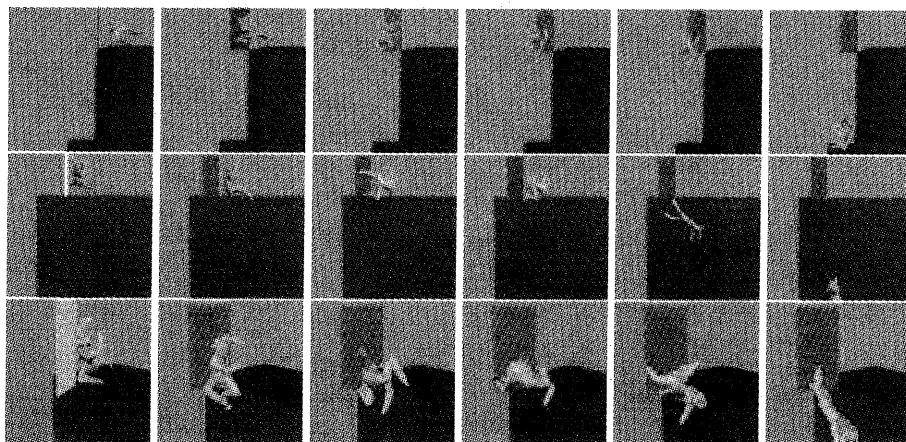


図 7: ドアから飛び出して落下する銭型警部 (ルパン三世より)

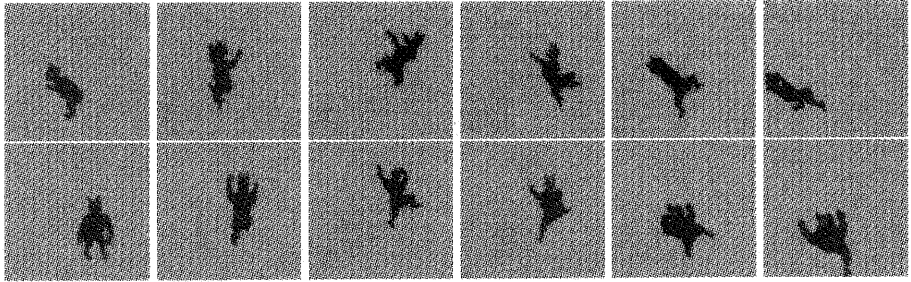


図 8: 驚いて、飛び跳ねて走り出すトム (映画「トムとジェリー」より)

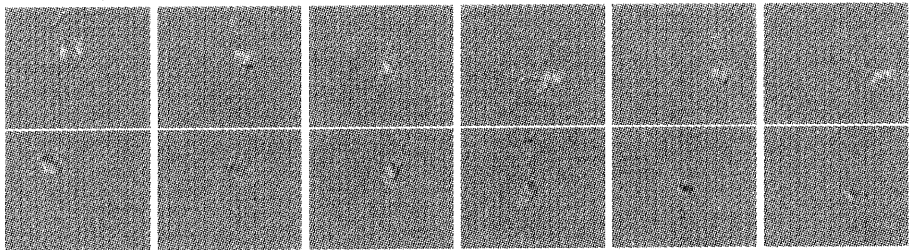


図 9: 『逃げるんだ、ピカチュウ!!!』 (「ポケットモンスター」より)

参考文献

- [1] J.O'Rourke and N.Badler. Model-based image analysis of human motion using constraint propagation. *IEEE Trans. Pattern Anal. Mach. intell.*, pages 522–536, 1980.
- [2] Zhiqiang Lao and Ling Li. Video-approach to human animation. *Computer Graphics forum*, pages 401–409.
- [3] Satoshi Matsuoka Takeo Igarashi and Hidehiko Tanaka. Teddy: A sketching interface for 3d freeform design. *Computer Graphics (Proceedings of SIGGRAPH 99)*, pages 409–416.
- [4] Z.Chen and H.Lee. Knowledge-guided visual perception of 3-d human gait from a single image sequence. *IEEE Trans. Systems Man Cybernet*, pages 336–342, 1992.