セルアニメーションからの三次元キャラクタモデルの生成とその応用

小野 豊[†] 陳 炳宇[†,☆] 西田友是[†]

セルアニメーションの制作において、キャラクタへの様々な効果の付与やキーフレーム間の中間画像の生成等を行うために、三次元モデルが用いられることがある。しかし、三次元モデルの作成は非常に手間と時間がかかる作業である。本稿では対応が定義されたストロークよりなる二次元の画像列から、それらに対応する三次元モデルの組を容易に生成する手法を提案する。生成された三次元モデルを用いることで、キャラクタへの影付けやテクスチャマッピング等を行うことができる。さらに、各三次元モデルの間には頂点の一対一対応があり三次元モーフィングを行うことができるので、中間画像の生成の補助としても本手法を利用することができる。

3D Character Model Creation from Cel Animation and its Applications

Yutaka Ono,[†] Bing-Yu Chen[†,☆] and Tomoyuki Nishita[†]

For creating a cel animation, the animators often use 3D character models to add some effects on the character or to generate intermediate images between the key frames. However, it is troublesome and time-consuming task to create a 3D model. In this paper, we present an easy-to-use approach for creating a set of 3D models from the user-specified strokes on a 2D image sequence. The created 3D models can be used in cel animation editing systems for adding shadowing effects, textures, etc. Moreover, since the vertices of the 3D models have one-to-one mapping among frames, by using 3D morphing techniques, the approach can also be used to generate intermediate images from key images.

1. はじめに

近年、セルアニメーションの作成を支援するために多くの三次元 CG 技術が使われるようになってきている。例えばトゥーンレンダリングは、三次元モデルを二次元のセルアニメーションと違和感無く合成することを可能にし、それを用いることで三次元モデルと手書きによる二次元の絵の両者の利点を活かしたアニメーションを作ることができる。三次元モデルの作成は非常に手間のかかる作業ではあるが、機械等の動きの少ないオブジェクトを表現する場合、少数のモデルのみを用いて様々なシーンに利用することができるため非常に有効である。しかし、人物等のキャラクタはシーンによってその形状が非常に大きく変化して描かれるため、完全な三次元モデルの構築はせず、シーンの影付け等の補助的な目的に用いるためのシンプルなモデルのみ使用されることが多い。

影付けに用いられるシンプルなモデルの作成には Igarashi ら[5] や Karpenko ら[6] 等が提案したスケッチシステムのような、アニメータの入力の手間が小さいモデリングシステムが広く用いられるが、これらは各アニメーションのフレームの輪郭のみを入力とするためフレーム間の対応関係を考慮しておらず、テクスチャマッピング等のアプリケーションに用いることは困難である。

そこで本稿ではユーザが入力したフレーム間の対応と輪郭を保ち、影付けやテクスチャマッピング、さらには中間画像の生成の補助として使用することができる、キーフレーム間で変形可能な三次元モデルを生成する手法を提案する。対応関係は各フレーム間で対応している特徴的ないくつかの曲線で定義され、それらを三次元化することで三次元モデルを生成する。また、近年のセルアニメーションはビットマップ画像ではなくベクトル形式の画像を用いて制作されることが増えてきている。提案法はベクトル間の対応を定義するだけで簡単に利用することができるため、アニメーションの製作工程に自然に組み込むことができると思われる。

† 東京大学
The University of Tokyo
☆ 現在，台湾大学
Presently with National Taiwan University

2. 関連研究

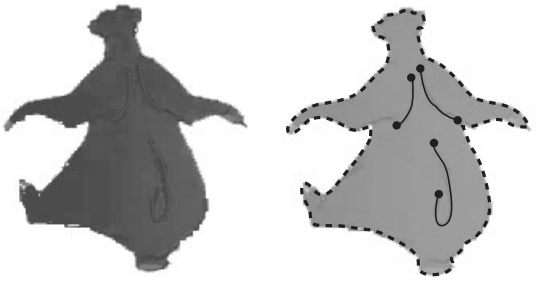
Rademacher[10] は熟練したアニメータが作成した三次元のキャラクタモデルを変形させてアニメーションを作る手法を提案した。この手法はいくつかの画像を参照に、アニメータが作成したモデルを手入力で変形し、それらを補間しレンダリングすることでアニメーションを生成する。しかし、手入力による三次元モデルの生成や画像に合わせた変形は非常に手間のかかる作業である。Corrêa ら[3] は三次元モデルをユーザのわずかな介入により画像にあわせるように変形する手法を提案したが、三次元モデルはアニメータが事前に作成する必要がある。

Bregler ら[2] は既存のアニメーションの動きを追跡することで、三次元キャラクタモデルにそのアニメーションと同様な動きをさせる手法を提案した。この手法もアニメータによる三次元モデルの作成が必要であり、キャラクタの二次元的な動きのみが追跡されるため、三次元的な動きを作るにはユーザの介入が必要となる。

三次元モデルの生成に不慣れなアニメータでも簡単にモデルを生成できる手法として、Igarashi ら[5] や Karpenko ら[6] は二次元の輪郭を入力するだけで三次元モデルを生成するスケッチシステムを提案した。Petrović ら[8] はこのようなスケッチシステムを利用してセルアニメーションの各フレームの輪郭から三次元モデルを作り、簡単な三次元シーンを構築して元のセルアニメーションに影を付ける手法を提案した。この手法では、各フレームで独立に三次元モデルを生成するためモデル間に対応関係は無く、他のセルアニメーション編集法への適用は困難である。

コンピュータビジョンの分野では、対応が与えられた二次元のフレーム列から対応関係を保った三次元モデルを生成する手法が数多く提案されているが、これらの多くは対象の物体が剛体の場合に限られている。Bregler ら[1] は剛体ではない物体の三次元モデルを生成する手法を提案したが、セルアニメーションのようにキャラクタの形状ゆがみや変形の度合いが非常に大きなものを対象とした場合はうまくいかないことが多い。

本稿では Petrović ら[8] の手法と同様、二次元セルアニメーションを入力としてシンプルなインターフェイスで三次元キャラクタモデルを生成する手法を提案する。提案法は Petrović ら[8] の手法とは異なり、ユーザが定義したフレーム間の対応関係を保った三次元モデルを生成するので、テクスチャマッピング等や中間画像生成の補助等のより幅広い応用が見込まれる。ユーザの入力の手間は従来法より多少面倒ではあるが、不慣れなアニメータでも簡単に操作できる程度のものである。



(a) (b)

図 1 輪郭と二次元特徴点、特徴ストロークの例。(a): 入力画像。(b): 輪郭と特徴点、特徴ストローク。それぞれ破線、実線、実線の端点で表される。©Disney

3. 提案法の概要と定義

ここでは提案法の流れを簡単に説明する。まず入力として二次元のセルアニメーションを与える。ユーザは入力アニメーションの各フレームに対し、キャラクタの輪郭と特徴的な曲線（特徴ストローク）を指定する。この際、キャラクタが関節モデル等、複数のコンポーネントよりなる複雑な形状をしている場合には各コンポーネントごとに入力を与える。次に、システムが各特徴ストロークを三次元化する。最後に、システムが各フレームについて輪郭から三次元モデルを生成し、三次元化された特徴ストロークをそれに埋め込み変形する。生成された三次元モデルは頂点間の対応関係を持ち、各フレーム間を補間するように変形可能である。

今後の議論を明確にするために、以後使用される言葉を定義する。輪郭とは対象としているキャラクタのコンポーネントの境界で定義される、画像空間（xy の二次元空間）上の閉じた曲線である。特徴点とはフレーム間で対応が指定されている点である。特徴ストロークとは特徴点同士を結んだ長さが 0 以上の閉じていない曲線である。画像空間上の特徴点、特徴ストロークを二次元特徴点、二次元特徴ストローク、それらに z 座標を付加した三次元空間 (画像空間の xy 座標系に奥行方向 z を加えた空間) 上の特徴点、特徴ストロークを三次元特徴点、三次元特徴ストロークと呼ぶ。輪郭や特徴ストロークの座標は全特徴点の座標値の平均との差を取ることで、原点をキャラクタ中心とする座標に正規化する。図 1 に輪郭と二次元特徴点、

特徴ストロークの例を示す。提案法はそれぞれの特徴ストロークが画像空間上で交差しないもののみを対象とする。以下では各プロセスについて詳しく説明する。

4. 特徴ストロークの入力とその三次元化

4.1 ユーザによる特徴ストロークの入力

ユーザにより m フレームよりなるセルアニメーションが与えられたとする。ユーザが介入することで、各フレーム $j(1,2,\cdots,m)$ で輪郭と二次元特徴点、二次元特徴ストロークを指定する。その際、インデックス i を付与することでフレーム間の対応関係を指定する。以下ではユーザにより与えられたインデックスの最大値を n とする。全ての特徴点は必ずしも全フレームで指定される必要は無く、可視のもののみ指定すれば良い。同様に、特徴ストロークは可視の領域のみ指定する。

4.2 各フレームの位置あわせ

三次元モデルを生成する対象となるキャラクタはアニメーション中で様々な動きをする。ここでは各フレームの画像に平行投影を仮定し、二次元特徴点をもとに各フレームの三次元空間での位置あわせを行う。すなわち、各フレーム j での最初のフレームに対する相対的な画像のスケール s_j とキャラクタの回転 R_j のパラメータを求める。提案法ではまず Tomasi ら[11] の手法を利用する。この手法はスケール、回転のパラメータに加え特徴点の三次元座標を推定することができ、全ての特徴点が全てのフレームで定義されていなくても使用することができる。しかし、この手法は対象が剛体である場合に限られているので、ダイナミックな動きをするアニメーションのキャラクタに適用した場合は期待するような位置あわせをできない場合が多い。後に述べる三次元特徴ストロークの生成をできるだけ正確に行うために、我々のシステムではユーザが各フレームでパラメータ s_j, R_j を明示的に指定することも許しているが、ユーザの負担を減らすため以下のような Tomasi ら[11] の手法の改良を用いる。

ユーザにより m フレームよりなるセルアニメーションの各フレームで、n 個の二次元特徴点が指定されたとする。まず、Tomasi ら[11] の手法を適用し、n 個の三次元特徴点の初期座標 $p_i(i=1,2,\cdots,n)$ を求める。これらは全フレームで共通であることに注意されたい。次に、各フレーム $j(1,2\cdots,m)$ について、オイラー角[4] で表される回転 $R_j(\alpha_j,\beta_j,\gamma_j)$ とスケール s_j を変数として評価関数 $\sum_{i,j} F_{ij}(q_{ij})$ を最大化する。ここで、q_{ij} は p_i を R_j, s_j を用いて画像空間に投影した二次元座標で、$F_{ij}(x)$ は以下のように定義される密度関数である。

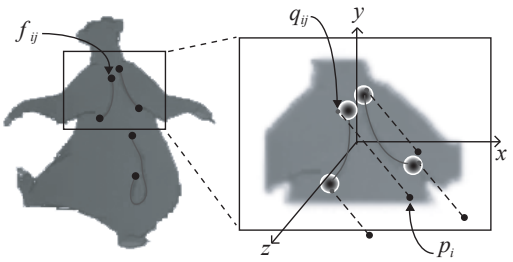


図 2 評価関数の模式図。各三次元特徴点 p_i を R_j, s_j により投影した二次元座標 q_{ij} と、対応する二次元特徴点 f_{ij} 間の距離を用いて評価関数の値を決定する。各円の半径は閾値 r_j を表す。©Disney

$$F_{ij}(x) = \begin{cases} (\|x - f_{ij}\| - r)^2, & \text{if } \|x - f_{ij}\| \le r_j \\ 0, & \text{otherwise} \end{cases}, \quad (1)$$

ここで、f_{ij} はフレーム j での i 番目の二次元特徴点の座標、r_j はユーザ指定の適当な閾値を表す。r_j は通常全てのフレームで同じ値を用いればよい。三次元特徴点の投影が二次元特徴点の座標に近いほどこの評価関数は大きくなる。 また、ユーザは必要であれば $\alpha_j, \beta_j, \gamma_j$ を任意に固定することで位置あわせの精度をより良くすることができる。この手法では適当な r_j を使うことで、いくつかの特徴点の座標がフレーム間で大きく変化するような場合でも比較的ユーザの期待するものに近い位置あわせを行うことができる。

4.3 特徴ストロークの三次元化

前節で得られたスケール s_j と回転のパラメータ R_j をもとに、各特徴ストロークをそれぞれのフレームについて三次元化する。以下のアルゴリズムでは異なるインデックスを持つ特徴ストロークをそれぞれ独立に三次元化することができるので、以下の説明では全てのフレームからある特定のインデックスを持つ、対応した二次元特徴ストロークのセットのみに着目し、それらを三次元化する方法を述べる。

まず、全フレームから最長の二次元特徴ストロークを探し、それを基本ストロークと定義する。基本ストロークをユーザ定義の定数長、基本セグメント長でサンプリングしたポリラインで表し、その他のフレームの二次元ストロークは基本ストロークと同じセグメント数からなるポリラインで表す。ポリラインの各フレームでの各頂点の z 座標を未知数とし、評価関数の最小化によりそれらを求める。z 座標の初期値には、前節での Tomasi ら[11] の手法により得られた特徴点の z 座標を線型補間した値を用いる。評価関数 G を、セグメント数 l よりなる、フレーム j での三次元ストローク $S_j = (v_1^j, v_2^j, \ldots, v_{l+1}^j)$ の集合 $S = \{S_j | 1 \le j \le m\}$

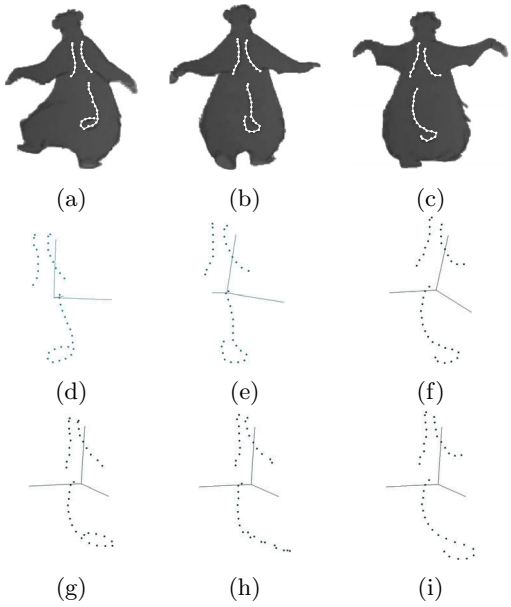


図 3 特徴ストロークの三次元化。(a)〜(c): 二次元特徴ストローク。(d)〜(f): (a)〜(c) を三次元化した三次元特徴ストローク。各軸はそのフレームでの回転を表す。(g)〜(i): (d)〜(f) を右方向の同一視点からみたもの。©Disney

に対し、以下のように定義する。

$$G(S) = \sum_{1 \le j \le m} (V(S_j) + \varepsilon K(S_j)) \tag{2}$$

$$V(S_j) = \| L(S_j) - \frac{1}{m} \sum_{1 \le j' \le m} L(S_{j'}) \|^2 \tag{3}$$

$$L(S_j) = \sum_{1 \le k \le l} \| s_j^{-1} R_j^{-1} (v_{k+1}^j - v_k^j) \| \tag{4}$$

$$K(S_j) = \sum_{1 \le k \le l+1} \| M(v_k^j) - \frac{1}{m} \sum_{1 \le j' \le m} M(v_k^{j'}) \|^2 \tag{5}$$

$$M(v_k^j) = s_j^{-1} R_j^{-1} v_k^j \tag{6}$$

すなわち、対応しているストロークはフレーム間で三次元的な長さは変化しないという仮定をおき、平均からの長さのずれの和を最小にする。$K(S_j)$ は三次元ストロークが同一平面上にのってしまうを防ぐ項であり、ε は小さな正の係数である。このアルゴリズムは他の多くの非線形関数の最小化問題と同じように最適解に収束する保証は無いが、評価関数は常に正であるので極小解には収束する。

なお、あるフレームで二次元特徴ストロークの一部のみが可視である場合には、可視の領域のみを基本セグメント長で区切り、不可視の領域は単純な線型補間により x, y 座標を割り当て、全体として他の特徴ストロークと同じセグメント数よりなるポリラインで表し、上のアルゴリズムを適用する。 図 3 は三次元ストロークの例である。フレーム間で形状の細部には差があるが、$K(S_j)$ の項により全体的には似通った形状が得られている。

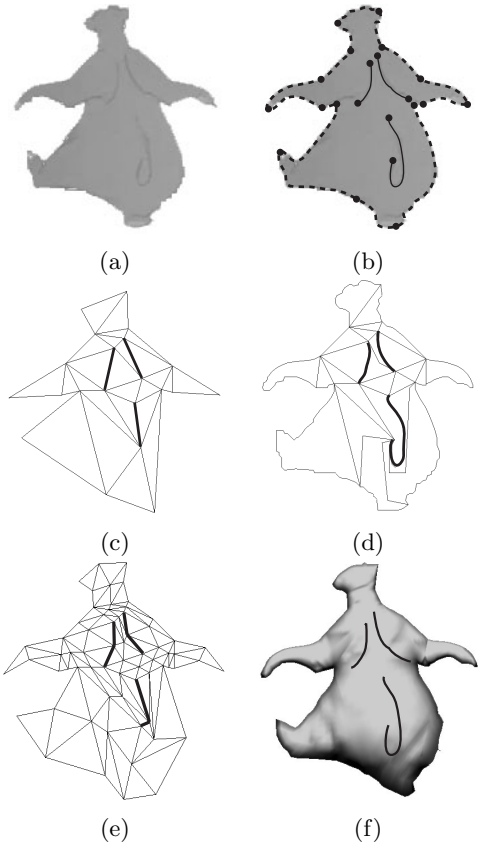


図 4 各フレームごとの三次元モデルの作成。(a): 入力フレーム。(b): (a) での輪郭と二次元特徴点、特徴ストローク。この例では印をつけた輪郭上の点も特徴点、それらを結ぶ輪郭の一部は特徴ストロークとして扱っている。(c): 輪郭と特徴ストロークで定義されるドメイン。(d): (c) に対してパラメータ化されるパッチ。(e): ドメインを一度細分割して得られる三角形。(f): 四度の細分割の後、三角形の頂点を ±z 方向に膨らませて得られた三次元モデル。(b)〜(f) の太い実線は特徴ストロークの対応を明示的に示したものである。©Disney

5. 三次元モデルの生成

ここでは各フレームごとに三次元モデルを生成する手法について述べる。基本的なアイディアは Igarashi ら[5] のアルゴリズムと同じように、三角化した輪郭の頂点を画像平面に垂直な $\pm z$ 方向に膨らませることで三次元モデルを作るというものであるが、我々の手法では前節までに得られた三次元特徴ストロークを用いて z 座標値を決定する。以下では全てのフレームで全ての特徴点、特徴ストロークが見えるシンプルな例を用いて説明する。

フレーム間で対応が取れた三次元モデルを生成する

ために、まずフレーム間で共通するドメイン[9]と呼ばれる三角形集合を定義する (図 4(c))。これは特徴点の間に二次元特徴ストロークの接続性を制約としておいた制約つき Delaunay 三角化のアルゴリズムにより得られる。それと同時に、ドメインと同じ接続性を持ち、特徴ストローク同士を結ぶパスで定義されるパッチを定義する (図 4(d))。もしパッチのパス同士が交差するようであれば、パスの間に新しい頂点を挿入し経路を調整することで、パス同士の交差がおこらないようにする。次に、パッチをドメインに対してパラメータ化する。これは、対応するそれぞれの三角形同士の間での弦長パラメタライゼーションにより行う。得られたパラメタライゼーションを用いて、ドメインの各三角形を再帰的に 1 対 4 の細分割を行うことで、輪郭の内部を三角形で埋める (図 4(e))。最後に、各頂点を ±z 方向に変位させることで三次元モデルが得られる (図 4(f))。Igarashi ら[5]のアルゴリズムでは内部の頂点の z 座標に輪郭からの距離に応じた値を与えるが、我々の手法では三次元特徴点、特徴ストロークの座標を利用して z 座標を設定する。すなわち、対応するストロークを持つ頂点については三次元特徴ストローク上の点での z 座標を、そうでない頂点については、その頂点に最も近い三次元特徴ストローク上の点での z 座標と Igarashi ら[5]の手法によって定まるそれとの、ストロークからの距離に比例した加重和により z 座標を定める。

全てのフレームにおいて全ての特徴点、特徴ストロークが可視で無い場合は、可視の特徴点と特徴ストロークを表の領域、不可視の特徴点と特徴ストロークを裏の領域というように、それぞれ別々の二次元領域に配置し、+z 方向と-z 方向への膨らましを独立に行うことで三次元モデルを生成する。ドメインの接続性は、最初のフレームのストロークから順に制約を追加していく制約つき Delaunay 三角化の増分アルゴリズムにより定める。

なお、上記のアルゴリズムは特徴ストロークが画像空間上で複雑に交差する場合等には適用できないことがあり改良の余地がある。

6. 実験結果

15 フレームよりなるアニメーションに対して提案法を用いて実験を試みた。図 4 はそのうちの 1 フレームでの三次元モデルの生成過程を示している。ユーザによる輪郭や特徴ストロークの入力は各フレームで 2〜4 分程度かかった。三次元ストロークの生成は MATLAB で実装し、Intel Pentium 4 1.7GHz の PC を用いて 2 分程度の実行時間、三次元モデルの組の生成は C++で実装し、1 分程度の実行時間を要した。

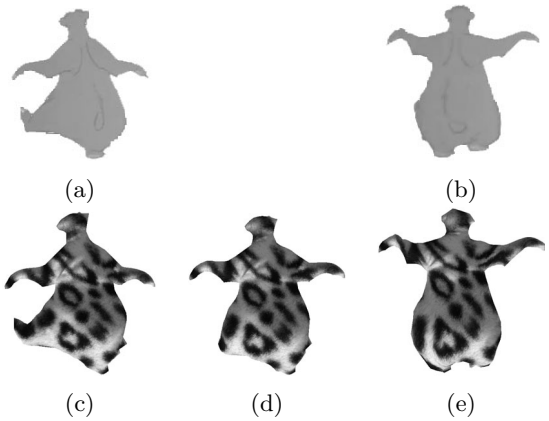


図 5 テクスチャマッピングを施したモデル間の三次元モーフィング。(a), (b): 入力フレームの一部。(c), (e): (a), (b) に対応する三次元モデル。(d): (c), (e) を補間して得られた三次元モデル。(d), (e) のテクスチャ座標は (c) のものを用いた。©Disney

図 6 入力アニメーションへの陰影付け。(a): 入力の 1 フレーム。(b): (a) に対して影をつけたもの。©Disney

生成された三次元モデルを用いたアプリケーションを以下に示す。 図 5 は生成したモデルにテクスチャマッピングを施し、中間形状を生成した例である。各フレームのモデル間に明示的な対応関係があるため、手足等においてテクスチャの模様の対応がとれていることがわかる。

図 6 は提案法を用いて生成したモデルを 2 つ利用して、元の画像に陰影を付け加えた例である。図 6(b) は、地面が設定された三次元空間上で計算した陰影の画像と、元の画像との間で積演算を行うことで得られた画像である。光源は画像の右から左への平行光源である。右のキャラクタの影が左のキャラクタに映りこんでいるのがわかる。この例に関しては Petrović ら[8]の手法とほぼ同様の出力が得られるが、特徴ストロークの本数が多い例に対しては、モデルの凹凸による複雑な影ができることが期待される。

7. おわりに

本稿では、セルアニメーションに対してユーザが与えた輪郭と、フレーム間で対応がとれた曲線から三次元モデルの組を生成する手法を提案した。対応づけられた曲線はフレーム間での長さと位置の矛盾が少ないように三次元化され、生成された三次元モデルはキーフレーム間の補間に用いることができる。生成されたモデルは図 5 や図 6 等のように、アニメーション作成の様々な支援を目的に使用することができると思われる。

将来の研究の課題として、より広範な三次元モデルの生成を可能にすることや、特徴要素の自動的な抽出、追跡、特徴ストロークの長さを考慮したモデル間の補間等があげられる。また、ドメインの定義には Kraevoy ら[7] の手法を用いるのも有効であると思われる。

謝辞 本稿で用いた入力アニメーションは Bregler らの論文[2] の Web サイト http://mrl.nyu.edu/~ bregler/tooncap/masters02.mov から頂いたものである。彼らおよび彼らの論文にアニメーションを提供した Disney 社に謹んで感謝の意を表する。

参考文献

1) Bregler, C., Hertzmann, A., and Biermann, H.: Recovering Non-Rigid 3D Shape from Image Streams, *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, 2000*, pp.690–696 (2000).
2) Bregler, C., Loeb, L., Chuang, E., and Deshpande, H.: Turning to the Masters: Motion Capturing Cartoons, *ACM SIGGRAPH 2002 Proceedings*, pp. 399–407 (2002).
3) Corrêa, W. T., Jensen, R. J., Thayer, C. E., Finkelstein, A.: Texture Mapping for Cel Animation, *ACM SIGGRAPH '98 Proceedings*, pp. 435–446 (1998).
4) Heckbert, P. S.: *Graphics Gems IV*, Academic Press Professional (1994).
5) Igarashi, T., Matsuoka, S., Tanaka, H.: Teddy: A Sketching Interface for 3D Freeform Design, *ACM SIGGRAPH '99 Proceedings*, pp.409–416 (1999).
6) Karpenko, O., Hughes, J. F., and Raskar, R.: Free-form Sketching with Variational Implicit Surfaces, *Eurographics 2002 Proceedings*, (2002).
7) Kraevoy, V., Sheffer, A., Gotsman, C.: Matchmaker: Constructing Constrained Texture Maps, *ACM SIGGARAPH 2003 Proceedings*, (2003).
8) Petrović, L., Fujito, B., Williams, L., and Finkelstein, A.: Shadows for Cel Animation, *ACM SIGGRAPH 2000 Proceedings*, pp. 511–516 (2000).
9) Praun, E., Sweldens, W., Schroder, P.: Consistent Mesh Parameterizations, *ACM SIGGRAPH 2001 Proceedings*, pp. 179–184 (2001).
10) Rademacher, P.: View-Dependent Geometry, *ACM SIGGRAPH '99 Proceedings*, pp.439–446 (1999).
11) Tomasi, C. and Kanade, T.: Shape and Motion from Image Streams under Orthography: a Factorization Method, *Int. J. of Computer Vision*, Vol. 9, pp. 137–154 (1992).