Pillow: ぬいぐるみ作成のためのインタラクティブな型紙デザイン

Pillow: Interactive Pattern Design for Stuffed Animals

森 悠紀　　五十嵐 健夫*

概要.　本稿ではぬいぐるみ作成のためのインタラクティブな型紙デザインの紹介をする．自分だけのオリジナルなぬいぐるみをデザインするためには，2 次元の型紙をデザインする必要があるが，その過程は素人には難しく専門家によって行われている．そこで我々はぬいぐるみ作りのための型紙生成を支援するシステムを作成した．ぬいぐるみをデザインする手法は 2 つ提案する．1 つ目は，既存の 3 次元サーフェスモデルを入力としてユーザがモデルの表面形状に沿ってインタラクティブに縫い目を描くと，描かれた縫い目からシステムが 2 次元の型紙へと展開する手法である．生成された型紙を 3 次元に再構築することで，ユーザはできあがり形状を確認することが可能になる．2 つ目は手書きスケッチに基づくモデリング操作によってぬいぐるみの立体形状を対話的にデザインする手法である．本システムを使うことで素人でも簡単にオリジナルのぬいぐるみが作成可能となる．

1　はじめに

現在，我々の身の回りにあるぬいぐるみは，大きくわけて既製品と手作り品の 2 種類に分類することができる．ところが，手作り品であったとしても，元々のぬいぐるみの形は，本からの型紙であったり，手作りキットの型紙であったりすることがほとんどである．オリジナルのぬいぐるみを作成するためには，対応した型紙を作成する必要があるが，できあがりの 3 次元形状を推測しながら 2 次元の型紙を作成することは素人には難しい．

そこで我々はぬいぐるみ作りのための型紙生成を支援するシステムを作成した．ぬいぐるみをデザインする手法は 2 つ提案する．1 つ目は，既存の 3 次元サーフェスモデルを入力としてユーザがモデルの表面形状に沿ってインタラクティブに縫い目を描くと，描かれた縫い目からシステムが 2 次元の型紙へと展開する手法である．2 つ目は手書きスケッチに基づくモデリング操作によってぬいぐるみの立体形状を対話的にデザインする手法である．

3 次元メッシュから 2 次元座標を割り振る研究はいくつか存在する [1, 2, 3, 4] が，これらはいずれも基本的に Texture Map 用であり布でできた形状であるという考慮は一切ない．

Julius らはぬいぐるみのための自動領域分割アルゴリズムを提案し，実際に自動生成された型紙を使用してぬいぐるみを作成した [5]．しかし，彼らの手法では，物体の特徴を捉えることができず，対称性や顔，胴体，手足，などの重要な情報を逃してしまっている．そのため，彼らの手法を用いると顔に複雑な縫い目が入ってしまったり，非対称な縫い目になったりしている．我々はユーザのセンスを活かしたパターンデザインをするためのツールを作成し，ぬいぐるみ作成プロセスを手助けしたいと考える．

本論文は以下のように構成されている．まず，次章で既存モデルからの型紙生成について紹介し，手書きスケッチに基づくモデリング操作からの型紙生成を 3 章で述べる．最後に 4 章で本論文をまとめ，今後の課題にふれる．

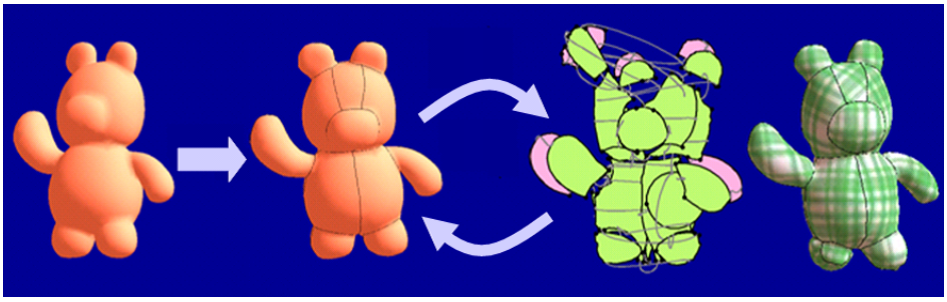
2　既存モデルへの縫い目デザインと型紙生成

既存モデルを用いての縫い目デザインと型紙生成の流れを図 1 に示す．

入力となる 3 次元モデルは任意の閉じた多面体 (メッシュ) であり，通常の CG モデラーを用いて作成するか，既存のものを利用することを仮定している (図 1(a))．この入力モデルに対してユーザは自分のセンスを活かした縫い目を入力していく (図 1(b))．システムは入力された縫い目を元に，入力モデルの形状を構築するための型紙を自動で提示する (図 1(c))．入力モデルは布で作られているということを考慮していない形状であるため，出力された型紙を実際に縫い合わせたぬいぐるみ形状は入力モデルと比較すると必ず歪みが生じる．そのため，この型紙を使って縫い合わせた結果どのようなぬいぐるみ形状になるかをシミュレーションする (図 1(d))．これによりユーザは，実際に作る前に出来上がりのぬいぐるみ形状を検証することが可能となり，本システムの中でユーザは試行錯誤をし，満足する縫い目になったら実際にぬいぐるみを作成するということが可能となる．

Copyright is held by the author(s).

* Yuki Mori, 東京大学大学院情報理工学系研究科コンピュータ科学専攻, Takeo Igarashi, 東京大学大学院情報理工学系研究科コンピュータ科学専攻，科学技術振興機構さきがけ



(a) 入力 3 次元モデル　(b) ユーザが入力した縫い目　(c) 型紙生成　(d) シミュレーション結果

図 1. 既存モデルへの縫い目デザインと型紙生成，シミュレーションの流れ

2.1　実装

本システムは Java 言語を用いて実装した．展開アルゴリズム，シミュレーションともに家庭などで広く使われている一般のノート PC 上でリアルタイムに稼動する．

3 次元モデルから 2 次元パターンへは ABF++[4] を利用して，角度ベースで展開を行った．3 次元上の三角形パッチと対応する 2 次元上の三角形パッチの面積がなるべく同じになるような調整を加えて大きさの調整をしている．

シミュレーションには現在は単純なバネモデルを用いている．まず，それぞれの頂点に対して，物体の内側から外側へ (法線方向へ) 膨らます力をかけ，頂点を移動させる．ある程度ふくらんだら，膨らまし方向の力をなくし，バネモデルを用いてそれぞれのエッジの長さを調整する．この 2 つのステップを 3 次元形状が収束するまで繰り返す．

2.2　結果

実際にいくつか実験を行った．本システムは一般的なポリゴンメッシュモデルを入力としているため，ネット上にあるモデルをダウンロードして利用することが可能である．また，広く使われている既存の 3D モデラーを利用して作成することも可能である．今回は初心者でも簡単に 3 次元サーフェスモデルをモデリングできる Teddy [6] を用いて実験を行った．

図 2. 図 1 から実際に作成したぬいぐるみ

図 1 のように縫い目をデザインし，実際に作成したぬいぐるみが図 2 である．

次に図 3 は同じ入力モデルに対して異なる縫い目をデザインした例である．上段のくまは平たい座布団のような形状をしているが，下段のくまは顔の真ん中に縫い目が入っているため，鼻が飛び出た形状となる．

このようにできあがりの形状が提示できるため，ユーザが縫い目を満足するまでデザインし，納得してから実際に縫うことが可能となる．

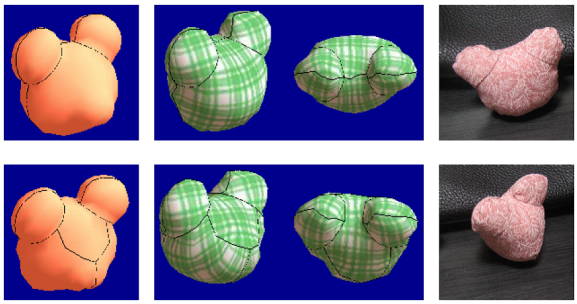


図 3. 異なる縫い目デザインの例：ユーザが入力した縫い目 (左)；シミュレーション結果 (中央)；システムが生成した 2 次元パターンを利用して実際に縫ったぬいぐるみ (右)；

3　ぬいぐるみモデリングと型紙生成

もう一方は，手書きスケッチに基づくモデリング操作によってぬいぐるみの立体形状を対話的にデザインする手法である．前章で紹介した手法は，すでに存在する 3 次元モデルのぬいぐるみを作成したいときには便利であるが，まったく新しく好みのぬいぐるみをデザインすることはできない．

そこで，ユーザが作りたいぬいぐるみの形状の外形を描くと，入力されたストロークを元に型紙を生成し，型紙の対応する縫い目を縫い合わせた結果の 3 次元シミュレーションを提示するシステムを作成した．これにより，ユーザはぬいぐるみを簡単にモ

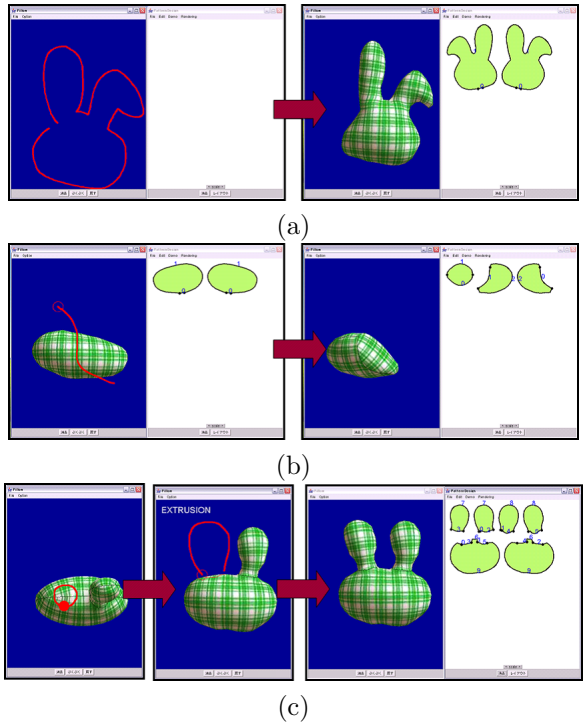


図 4. ユーザができる操作一覧：新規生成 (a)；切断 (b)；突起生成 (c)．

デリングするだけで，対応する型紙を得ることが可能となる．

3.1 インタフェース

本システムは型紙生成やモデリングの初心者をターゲットユーザとしているため，モデリングの知識や経験がなくとも簡単に 3 次元をモデリングできる Teddy [6] のユーザインタフェースを利用した．

ユーザが利用できる操作は図 4 に示した 3 つである．これにより，常にぬいぐるみになるようなモデルしかモデリングできないモデラーとなり，さらにモデリングをしているそばから，対応する型紙を生成することが可能となる．

3.2 アルゴリズム

システムはユーザが入力した 2 次元ストローク情報を元に型紙を生成する．モデルの新規生成 (図 4(a)) では，ユーザが描いたスケッチをそのまま 2 次元の型紙とし，同じものを 2 枚縫い合わせたものを構築する．モデルの切断 (図 4(b)) では，すでにある型紙の中でユーザが描いたストロークが横断されているものに関しては，型紙をそのストロークで切断する．新規生成された断面に関しては ABF++[4] を利用して平面展開をする．モデルが切断された際に捨てられる側にしか使用されない型紙は削除する．出力された型紙を 3 次元に再構築すると断面は平面ではなくなる．モデルの突起生成 (図 4(c)) では，ま

図 5. 実験会場の様子

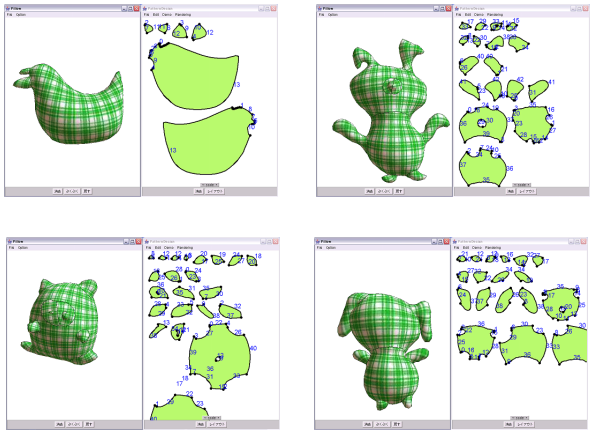


図 6. 本システムを利用してぬいぐるみをデザインした例

ずユーザが 1 番目に描くストロークでモデルの切断の際と同様に型紙を切断する．次にユーザが 2 番目に描くストロークをそのまま型紙として，突起部分に対応する型紙を生成する．

型紙を生成したら，三角形分割をしたのち，前章と同様の 3 次元シミュレーションを用いた．

3.3 実験

実際にユーザビリティの実験を行った．対象は日本科学未来館に来館した一般男女 38 人である．図 5 は実験会場の様子である．

本システムを利用して，モデリングをしていただいた．図 6 に実験結果のモデルを提示する．アンケートによると，「面白かった」「楽しかった」という意見がほとんどで，「自分だけのオリジナルのぬいぐるみが作れると思うとわくわくした」「描いた部分がしだいに“ぷくぷく”ともりあがってくる様子がリアルでよかった」という声もあった．また，マウスで絵を描くことや，突起生成の操作に関しては小学生程度の子供にはとても難しく，「丸を描くだけでも難しかった」「自分の思い通りにならなかった」「基

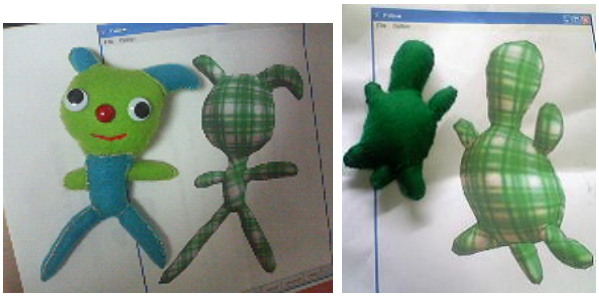


図 7. 実際にぬいぐるみを作成してくれた人の例

本図形やパーツをテンプレートとして用意してそこから選べると良い」などの意見があった．

さらに持ち帰った型紙を利用して実際にぬいぐるみを作成し, 報告してくれた人の例を図 7 に紹介する．このように素人でも簡単にオリジナルのぬいぐるみを作成することができる．

4 まとめと今後の課題

オリジナルのぬいぐるみを作成するためには，それに対応する型紙の生成が必要不可欠である．通常３次元形状から型紙を作成する工程は，ぬいぐるみ設計の専門知識をもつ者によって行われており，素人には困難である．

そこで我々はぬいぐるみ作りのための型紙生成を支援するシステムを作成し，ぬいぐるみをデザインする手法を 2 つ提案した．1 つ目は，既存の 3 次元サーフェスモデルを入力としてユーザがモデルの表面形状に沿ってインタラクティブに縫い目を描くと，描かれた縫い目からシステムが 2 次元の型紙へと展開する手法である．2 つ目は手書きスケッチに基づくモデリング操作によってぬいぐるみの立体形状を対話的にデザインする手法である．これらにより，素人には困難であったぬいぐるみデザインと型紙生成が手軽に行えるようになった．

今後は実証実験の結果を元にぬいぐるみモデラーをより簡単に使えるようにユーザインタフェースを検討するほか，一度作成したモデルに対して変形を加えると型紙も変形するアルゴリズムを検討する．また，現在は 3 次元の構築に単純なバネモデルを用いているが，布のシミュレーションには多数の研究が行われている [7, 8] ため，これらをぬいぐるみに拡張することに取り組んでいく予定である．

謝辞

OpenNL を提供してくださった Bruno Levy 氏，本研究を進めるにあたってコメントをいただいている Dan Julius 氏，実証実験を行うにあたって協力してくださった常盤拓司氏をはじめとするみなさま，実験場所を提供してくださった日本科学未来館に深く感謝いたします．本研究の一部は，情報処理推進機構 (IPA) の 2006 年度未踏ソフトウェア創造事業の支援を受けて実施されたものである．

参考文献

[1] O. Sorkine, D. Cohen-Or, R. Goldenthal, and D. Lischinski. Bounded-Distortion Piecewise Mesh Parameterization. In *Proceedings of IEEE Visualization '02*, pp. 355–362, 2002.

[2] B. Levy, S. Petitjean, and N. Ray. Least Squares Conformal Maps for Automatic Texture Atlas Generation. *ACM Transactions on Graphics*,21(3), pp.362–371, 2002.

[3] A. Sheffer, and E. de Sturler. Smoothing an Overlay Grid to Minimize Linear Distortion in Texture Mapping. *ACM Transactions on Graphics*, 21(4), pp.874–890, 2002.

[4] A. Sheffer, B. Levy, M. Mogilnitsky, and A. Bogomyakov. ABF++: Fast and Robust Angle Based Flattening. *ACM Transactions on Graphics*, 24(2), pp. 311–330, 2005.

[5] D. Julius, V. Kraevoy, and A. Sheffer. D-Charts: Quasi-Developable Mesh Segmentation, Computer Graphics Forum(In *Proceedings of Eurographics 2005*), 24(3), pp.981–990, 2005.

[6] T. Igarashi, S. Matsuoka, and H. Tanaka. Teddy: A Sketching Interface for 3D Freeform Design. *ACM Transactions on Graphics* (In *Proceedings of SIGGRAPH'99*), 21(3), pp.409–416, 1999.

[7] K.-J. Choi, and H.-S. Ko. Stable but Responsive Cloth. *ACM Transactions on Graphics* (In *Proceedings of SIGGRAPH 2002*), 21(3), pp.81–97, 2002.

[8] E. Grinspun, P. Krisl, and P. Schroder. CHARMS: A Simple Framework for Adptive Simulation. *ACM Transactions on Graphics*, 21(3), pp.281–290, 2002.