

# ステンシルデザインのためのドローエディタ

A Drawing Editor for Stencil Design

五十嵐 悠紀 五十嵐 健夫\*

概要． 素人のユーザがステンシルを楽しむためのインタラクティブなドローソフトを提案する．ステンシル型版は常に1枚につながっているという制約を満たすようにデザインしていかなくてはならないため、オリジナルなステンシル型版をデザインすることは大変難しい．そこで我々はステンシル型版を生成するための手法を提案する．ユーザは通常のドローソフトのように自由に図柄をデザインしていく．システムはユーザの描いた線を元に制約に基づいてインタラクティブにステンシル型版を生成する．システムの内部ではベクター形式とラスター形式のハイブリッドの実装になっている．ステンシル型版はベクターデータとして出力できるため、カッタープリンタなどを用いて実世界のステンシル型版を作成することができる．ユーザはそれを用いて自由に布や手紙、ポストカードなどにステンシルをすることができる．我々は素人のユーザに使用してもらい、本システムの有効性を示す．

## 1 はじめに

ステンシルは穴の開いたステンシル型版を用いて上からインクなどを使って図柄をデザインするアートである．ステンシル型版は positive と negative の2種類の領域からなる．negative な領域は穴の開いた領域であり、インクで描くことができる．一方、positive な領域はすべてがつながっているという制約がある．ステンシルはクリスマスカード、年賀状、誕生日カードなどのオリジナルなカードをデザインする際にもよく使われるアートであり、ロゴやシンボルなどをデザインする際にも良く使われる．しかし、ステンシル型版には1枚につながっていないとなければならないという制約があるため、素人がオリジナルなステンシル型版をデザインすることは非常に難しい(図1)．そのため、ステンシル型版をデザインするためには経験と知識が必要となり、ほとんどの人々はデザイナーがデザインしたステンシル型版を購入してきてステンシルを楽しんでいる．自分でオリジナルなステンシルデザインをする人は限られているため、我々は素人のユーザでも簡単にステンシル型版をデザインできるツールを提案する．

本システムは素人がステンシル型版をデザインするものである．図2にシステムの概要を示す．ユーザはキャンバス上に自由にストロークを対話的に描いていくことができる．システムはユーザの入力した線を元に自動的にステンシル型版を生成する．ユーザはシステムが出力したステンシル型版を実際の紙に写し取る．カッタープリンタなどを用いて切ることも可能である．最後に得られたステンシル型版を



図 1. ステンシル型版と実際のステンシルの例．ステンシル型版は1枚のつながった板からなる．  
<http://www.mintclothes.com/>．

用いて上からインクでデザインすることで図柄をステンシルすることができる．我々は本システムを用いてユーザスタディを行い、素人がステンシルをデザインできることを確かめた．

内部では、システムはベクターデータとラスターデータを保持しているユーザのストロークはベクターデータのプリミティブとして保持しており、システムはラスターイメージに変換して boolean operation でラスターイメージに変換している．システムはストロークのまわりに図1のように positive な領域を作り、negative な領域を描く．システムは negative な領域の周囲をトレースしてベクターデータとして出力する．本システムでは SVG 形式と DXF 形式をサポートしている．

## 2 関連研究

本システムは対話的なドローイングエディタである．2次元画像はコンピュータグラフィックスの分野ではベクターイメージ形式(パスごとの構成)とラスターイメージ形式(ピクセルごとの構成)の大きく2つの種類に分類される．ベクターグラフィックス形式では点列、線、カーブなどの幾何学的なプリ

Copyright is held by the author(s).

\* Yuki Igarashi, 東京大学大学院 工学系研究科 / 学術振興会 特別研究員 DC2, Takeo Igarashi, 東京大学大学院 情報理工学系研究科 / JST ERATO

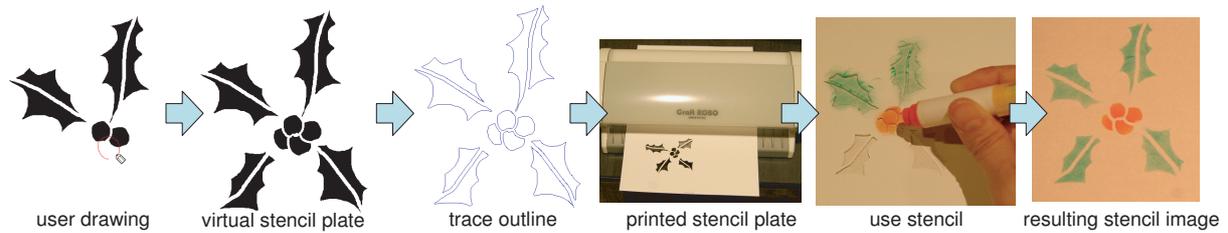


図 2. システムの概要．ユーザがデザインした図柄からシステムは対話的にステンシル型版を生成する．カッタープリンタなどを用いて実際のステンシル型版を生成することができ、紙や布などにステンシルをする．

ミティブを保持しており、Adobe Flash や Adobe Illustrator などが本形式の代表例である．2次元アニメーションスタジオの専門家のためのシステムである TicTacToon [1] は、従来の紙ベースでの製作過程に代わるシステムであり、ベクターベースのスケッチ&ペインティングソフトとしては最初のアニメーションシステムである．Gangnet らは TicTacToon のセルの色づけのギャップを解決したシステム [2] を提案した．Asente らは planar-map-based ですべての元のパスを変えることなく色づけしていくメタファを提案した [3] ．

一方、ラスタイメージはそれぞれの独立したピクセルに対して色を決定していく形式である．3次元グラフィックスのボクセル表現と対応する形式で、Adobe Photoshop や Microsoft PaintBrush などがラスタイメージ形式の代表ソフトウェアである．Qu らは強い連続性のある領域に同じ色を色づけしていく手法を提案し、白黒マンガに適用した [4] ．ストローク、ハッチング、ハーフトーニング、スクリーニングなどを含むマンガの色づけに有効な手法である．McCann らはアーティストがメガピクセルサイズの絵をリアルタイムに描くためのシステムを提案した [5] ．

Bronson らは3次元ポリゴンモデルや2次元イメージデータからステンシル型版を生成する手法を提案した [6] ．このシステムではユーザは3次元モデルを入力して、作りたい図になるように角度、ライト環境、線幅などを与えるとシステムがステンシル型版を生成する．ステンシル型版生成のアルゴリズムは複数の測定値をもとに適切な領域同士を結んで生成している．橋を安定させるために歪みを最小化する local feature を見つけてくる手法である．このシステムでは3次元モデルや2次元の絵からステンシル型版を生成する．我々の提案システムではユーザのスケッチ入力から対話的にステンシル型版をデザインしていく．

### 3 ユーザインタフェース

ユーザは通常のドローエディタのようにキャンバス上に対話的にストロークを描く．本システムでは

基本的なプリミティブ(ブラシツール、塗りツール、消しゴムツール)を用意した．ユーザはこれらのプリミティブを動かしたり、消したり編集することができる．システムは自動的にすべての positive 領域がつながったステンシル型版のイメージを生成する(図2)．本システムでは negative 領域の周りには必ず positive 領域が存在するようにステンシル型版を生成する．ユーザが描いていて positive 領域が切断されるような状況になるとシステムは自動的に橋を生成して positive 領域をつなげる処理を行う．詳細は後述する．

ユーザはブラシツール(図3)か塗りツール(図4)を選択する．ブラシツールの場合にはユーザの入力したストロークをそのまま negative 領域としてシステムはステンシル型版を生成する．塗りツールの場合には、システムは入力ストロークの始点と終点をつなげ、ストロークの内部の領域すべてを negative 領域としてステンシル型版を生成する．ユーザはブラシツールと塗りツールを用いてステンシルの図柄をデザインしていく．本システムでは常に negative 領域の周りに positive を生成することで、図1にある専門家がデザインしたようなスタイルを得ることができる．

システムはオーバーライティングモードとアンダーライティングモードの2つのモードを備えている．オーバーライティングモードでは最後に描いたストロークが一番上に描かれていく．アンダーライティングモードでは最後に描いたストロークが一番下になるように描かれていく．ユーザは図柄をデザインしている最中にモードを切り替えることが可能である．初期設定はアンダーライティングモードになっている．

ユーザは消しゴムツール(図5 a, b) や領域消去ツール(図5 c, d) を用いてすでに描いた図柄を消すことができる．これらの消しゴムツールは positive なプリミティブとしてブラシストロークや塗りストロークの上に描かれる．消しゴムツールを使った際には、システムは positive 領域をユーザの描いたストロークに沿って作る．領域消去ツールの際にはシステムは入力されたストロークの始点と終点をつなげ、内部の領域全体を positive 領域とする．

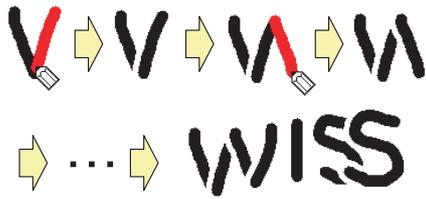


図 3. ブラシツールを用いてデザインした例 .

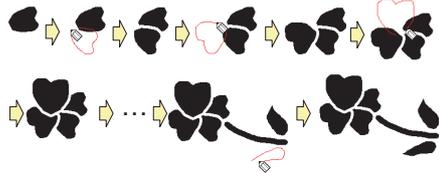


図 4. 塗りツールを用いてデザインした例 .

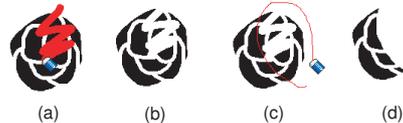


図 5. 消しゴムツール (a, b) や、領域消去ツール (c, d) を用いてデザインした図柄を消すことができる .

内部的には、システムはすべてのブラシストロークと消しゴムストロークをプリミティブとして保持している . そのため、描かれたプリミティブを簡単に編集することができる . ユーザは通常のドラッグ操作でストロークを動かすこともできる (図 6) . ストロークの上で右クリックをすることでポップアップメニューが開く . メニューからは消去 (図 7) , 順序変更 (図 8) などのさまざまなコマンドを選ぶことが可能である .

すべてのストロークは始点と終点を区別しており、自己交差が起きたときにも図 9 のように適切に処理をする . ユーザはポップアップメニューから「リバース」を選ぶことで、始点と終点を逆にすることができる . 初期設定では自動的にストロークの周囲にマージンが生成されるが、ユーザが必要ないと判断した場合にはマージンをなくすこともできる (図 10) . このつなげる処理をした場合には、ストローク同士がグループとして記録される . ユーザはプリミティブを矩形選択で複数選択して、コピー・ペーストすることもできる (図 11) .

ステンシルの図柄をデザインしていくうちにステンシル型版と接続していない、孤立した positive 領域 (島) ができてしまうことがある . システムはこの島を自動的に検出して、ステンシル型版と接続する positive 領域 (橋) を生成する . たとえば、図 12(a) のようにユーザがアンダーライティングモードにおいて塗りストロークを描くと、システムは自動的に島をステンシル型版に接続する橋を生成する



図 6. ストロークを動かした例 .



図 7. ポップアップメニューから選択することでストロークを消去できる .

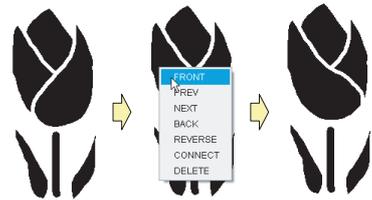


図 8. 描かれたストロークの順序を変更することができる .



図 9. 自己交差の例 .

(図 12b) . ユーザがストロークを動かした際には、システムは自動で橋の位置を修正したり (図 12d) , 橋を生成する必要がなくなった場合には橋を消去する (図 12e) . 自動で生成された橋をユーザが気に入らない場合には、ユーザは自分がデザインしたい位置に消しゴムツールで橋を描くことで、システムは自動で生成した橋は必要なくなるので消去する (図 12f) . ユーザは橋やマージンの幅を変更することができ、これらは最終的に作るステンシル型版の素材に応じて決定すると良い .

ユーザは自動橋生成モードを切することもできる . この場合にはシステムは警告モードになり、ステンシル型版に接続しない positive 領域は緑色にハイライトしてユーザに提示する (図 13b) . この領域があ

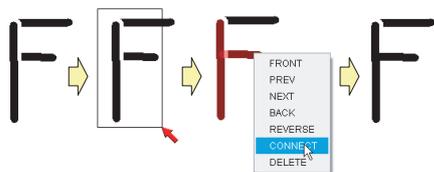


図 10. ストロークをつなげることもできる .



図 11. コピーとペーストの例 .

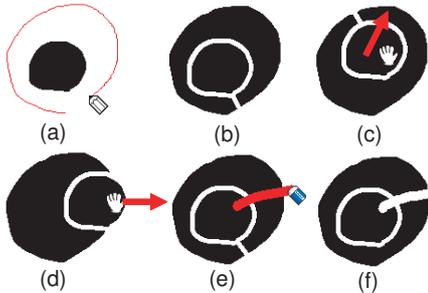


図 12. システムは自動的に島を検出し，橋を生成する . 必要のない橋は消去される .

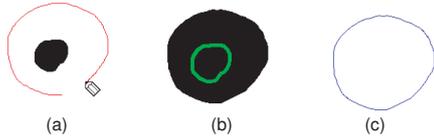


図 13. 警告モードの例 . ユーザが描いているときに島ができるとその領域をハイライトしてユーザに示す .

る状態で最終的なベクターデータファイルに出力した場合には，島となった領域は消される (図 13c) .

システムは最後にステンシル型版をベクター形式の画像として出力する . 本システムでは SVG 形式と DXF 形式をサポートしている . ユーザは描かれた線に沿って切ることによって実際のステンシル型版を得ることができる . カuttingプロッターを使うと便利である . ユーザは得られたステンシル型版を用いてステンシルを楽しむことができる .

#### 4 実装

プロトタイプシステムは Java で実装しており，内部的にはベクター形式とラスター形式の両方を組み合わせた形式になっている . プラシストロークはポリラインで表現されており，塗りストロークはポリゴンで表現されている . 通常のストロークは negative プリミティブとして，消しゴムストロークは positive プリミティブとして表されており，これらは入力順に保持されている . ステンシル型版イメージは図 14 のように positive と negative 領域の boolean ラスタイメージである . システムは保持されているストロークのリストをステンシル型版イメージ上に 1 つ 1 つ描画していく . 詳細は以下に述べる .

図 15 に提案システムのデータ構造を示す . システムはプリミティブのリスト (stroke と group) と

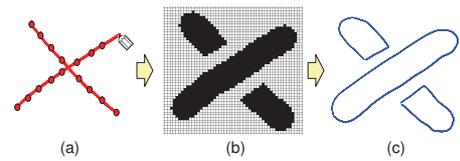


図 14. ステンシル型版イメージは Boolean ラスタ形式で保持されている .

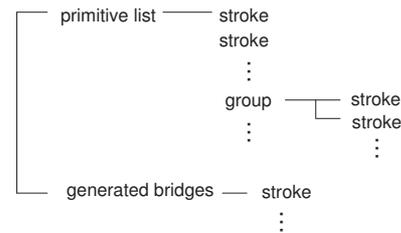


図 15. 提案システムのデータ構造 .

自動で生成された橋 (stroke) のリストを保持する . ユーザがプリミティブの順序を変更したり消去したり動かしたりした際にはシステムはすべてのプリミティブを再描画する . プリミティブは頂点列とストロークのタイプ (ブラシ，塗り，消しゴム，領域消去) の情報を持つ . システムがプリミティブをロードした際にも，すべてのプリミティブを順に再描画する . 一方，オーバーレイモードの場合にはシステムは最後に描かれたプリミティブのバウンディングボックス内だけを描画する .

まず，ステンシル型版の全画素を positive に初期化する . ブラシツールの際にはユーザのストロークにマージンをつけた positive なストロークをステンシルラスタイメージに描く . システムは次に通常の幅で negative ストロークを描く (図 16) . このようすることでストロークの周囲にマージンを生成することができる . 塗りツールの場合には，システムは最初にストロークの始点と終点を結ぶ . マージンをつけた positive 領域をまず描き，その後通常の領域を negative として描く .

システムはプリミティブのリストを 1 つずつ順に描画していく . ユーザが新しいストロークをキャンバスに描くと，ドローイングモードに応じてプリミティブリストの先頭か末尾に追加する . 初期設定のアンダーレイティングモードではリストの先頭に追



図 16. 描かれたストロークのアルゴリズム . この図では positive な領域を説明のためにグレーで描いている . (システムでの実装は白色) .

加され、ステンシルイメージ全体を描画し直す。オーバーライティングモードでは、描いたストロークはリストの末尾に追加され、この場合にはすでにあるステンシルイメージは新しく追加されたストロークのバウンディングボックス内だけを描画し直す。

自己交差を持つストロークは島領域ができてしまう(図9)。この問題を解決するために、我々は複雑な処理をしている。ユーザが通常のドローツールを選んだ際には、システムは内部でマウスドラッグイベントが呼ばれるたびに、positive な幅の広いストロークと negative なストロークを少しずつ描画していく。まず、最初に  $p_{n-1}$  から  $p_n$  の positive の幅の広いストロークを描き、次に  $p_{n-2}$  から  $p_n$  まで普通の negative ストロークを描く。このようにすることで図9のように自己交差をもつストロークを描いたときに島ができないようにすることができる。交差している部分の重なりを逆にしたいとしたときには、システムはそのストロークの頂点列を逆向きにして再描画する。

消しゴムツールでは、ユーザが入力したストロークをシステムは positive な通常の幅のストロークとして描画する。領域消去ツールではシステムはまずストロークの始点と終点をつなぎ、ストロークの内部を positive な領域として描く。消しストロークに関してもプリミティブの1つとして保持される。

ユーザがストロークをつなげたいとしたときには、システムは選択されたストロークをグループプリミティブとして登録する。プリミティブをレンダリングする際にはまずグループ内のすべてのストロークに関して positive な幅の広いストロークを描き、次に negative な通常の幅のストロークを描く。

描いたり消したりしている間にステンシルイメージはステンシル型版では生成できない、孤立した positive 領域が作られてしまう。システムは自動的にこの領域(島)を検出して橋を生成する。まず、flood fill アルゴリズムを適用して positive 画素を抽出する。システムは次に2つの領域の最短距離を計算し、距離をコストとする完全グラフを生成する。システムは次に最小木を抽出し、残った枝に対応するエッジに橋を渡す。この橋は橋リストとしてプリミティブに保存される。これらはユーザが画像を編集するたびに再計算される。

システムはカッタープロッターなどで出力するために、ベクター形式のファイルとして出力することができる。本システムは DXF 形式と SVG 形式をサポートしている。まず警告モードの際に現れる島領域を消して、マーチングスクエアアルゴリズム [7] を用いる。得られたエッジをトレースしてスムージングをする。最後にベクター形式として保存する。

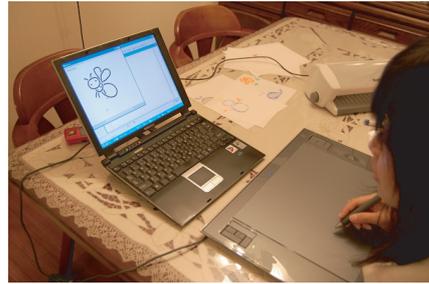


図 18. ユーザテストの様子。



図 19. ユーザテストで作ってもらったステンシル型版と実際のステンシル。

## 5 結果

システムは 1.1 GHz Pentium M PC 上でリアルタイムに稼動する。我々は本システムを利用していくつかのステンシルの図柄をデザインした(図17)。

3人のテストユーザに使ってもらい、それぞれの独自の図柄をデザインしてもらった(図18,19)。テストユーザはシステムの使い方を5-10分ほどで習得し、それぞれのデザインを5-10分ほど行った。得られた図柄をカッタープリンタで切って、実際にステンシルするのにだいたい10分くらいであった。大変使いやすく楽しい経験ができたと述べた。特に自動で橋ができる機能が使いやすいと述べた。

## 6 まとめと今後の課題

我々はユーザの入力ストロークから対話的にステンシル型版をデザインするシステムを提案した。内部的には入力ストロークはベクター形式で保存されているので編集作業も簡単にでき、ステンシル型版はラスタ形式で Boolean オペレーションを用いて処理される。システムは最後にベクター形式(SVGやDXF形式など)として出力可能である。ユーザは実際のステンシル型版をカッタープリンタなどを用い

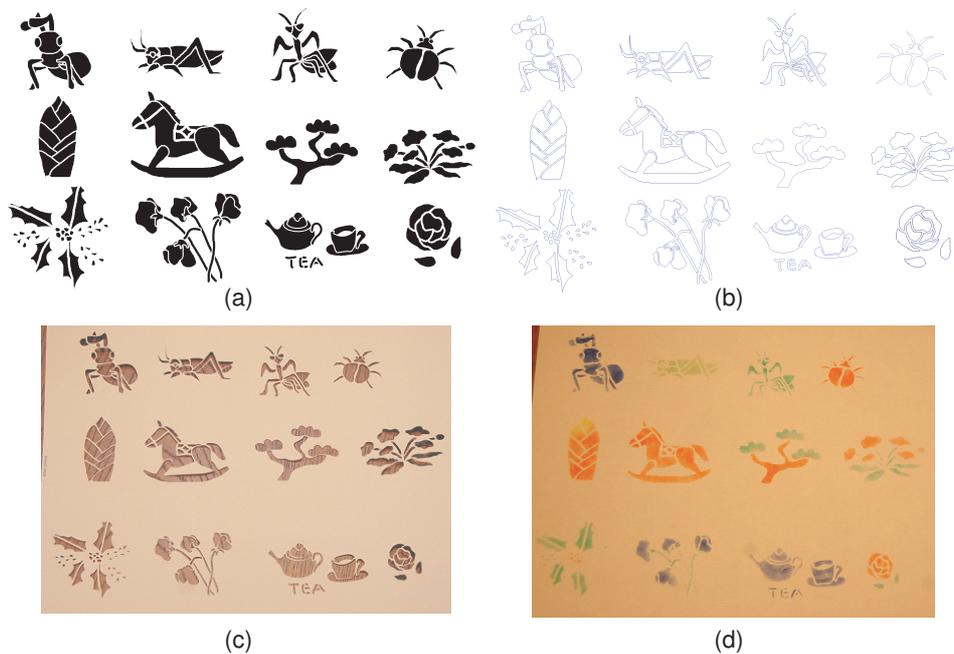


図 17. 本システムを利用してデザインした図柄 .

て作ることができ、実際のステンシルを楽しむことができる。我々は素人のユーザでも自由にオリジナルな図柄がデザインできることを示した。10月に本システムを利用したワークショップの実施<sup>1</sup>を行うため、より多くのユーザに使ってもらい検証をする。

現在の実装では最終的にステンシル型版を作る素材などを考慮していない。たとえば木などの硬い素材を使うのであれば橋は細くてよいが、紙などの薄い柔らかい素材を使うのであれば太い橋でなければならない。加えて、大きな島をつなぐ橋は太くないといけない。現在のシステムでは白黒で表示しているが、実際のステンシルでは穴の部分に何色のインクを乗せても良いため、カラーをサポートしたい。

本研究は実世界の物体をコンピュータを用いて効率的にデザインするシステムである。このアプローチを用いてほかの物質のデザインなどにも応用していきたい。

## 謝辞

ワークショップ開催に関してお世話になっている友の会担当鶴ヶ崎勝氏、科学コミュニケーター松島淳一氏、本研究で用いている DXF ファイル出力に関するソースコードを提供して下さった筑波大学准教授 三谷純氏、ユーザスタディの被験者になって下さったみなさまに感謝いたします。本研究は日本学術振興会特別研究員奨励費 (20-10073) の助成を受けたものである。

<sup>1</sup> 日本科学未来館友の会イベント『ステンシルでオリジナルエコバッグをデザイン』2009年10月25日開催予定。

## 参考文献

- [1] J. D. Fekete, E. Bizouarn, E. Cournaire, T. Galas, and F. Taillefer. TicTacToon: a paperless system for professional 2D animation. *Computer Graphics and Interactive Techniques*, 79-90,1995.
- [2] M. Gangnet, J. V. Thong, and J. Fekete. Automatic gap closing for freehand drawing, *Technical Sketch*, SIGGRAPH 94, ACM,1994.
- [3] P. Asente, M. Schuster, and T. Pettit. Dynamic planar map illustration, *ACM Transactions on Graphics (SIGGRAPH 2007)*,26(3), 30, 2007.
- [4] Y. Qu, T. -T. Wong, and P. -A. Heng. Manga Colorization, *ACM Transactions on Graphics (SIGGRAPH 2006)*, 25(3), 1214-1220, 2006.
- [5] J. McCann, and N. S. Pollard. Real-time Gradient-domain Painting, *ACM Transactions on Graphics (SIGGRAPH 2008)*,27(3),93, 2008.
- [6] J. Bronson, P. Rheingans, and M. Olano. Semi-Automatic Stencil Creation through Error Minimization. *Proc. of Non-Photorealistic Animation and rendering*,31-37,2008.
- [7] W. E. Lorensen, and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. *ACM SIGGRAPH Computer Graphics*,21(4), 163-169, 1987.