

Holly：ステンシルデザインのためのドローエディタ

五十嵐 悠紀^{1,2,a)} 五十嵐 健夫^{3,4}

受付日 2011年4月17日, 採録日 2011年12月16日

概要：初心者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“Holly”を提案する。ステンシルと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表現手法である。このステンシル型版はつねに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自らデザインすることは子どもや初心者には難しく、既存デザインのステンシル型版を利用して楽しむか、自らデザインするにしてもサンプルパターンの組合せでのデザインにとどまっている。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もと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形式とラスタ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データとして出力できるため、カッタ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。我々は実際に子どもや初心者のユーザに使用してもらうことで本システムの有効性を示したので報告する。

キーワード：デザイン, インタラクティブシステム, ステンシル, 初心者

Holly: A Drawing Editor for Stencil Design

YUKI IGARASHI^{1,2,a)} TAKEO IGARASHI^{3,4}

Received: April 17, 2011, Accepted: December 16, 2011

Abstract: We introduce an interactive system that allows nonprofessional users to design their own stencil plates. The design of an original stencil is very difficult because of the specific physical constraint requiring that all positive regions are connected. We present a method for generating expressive stencils in which the user simply uses standard drawing operations and the system automatically generates the appropriate stencil plate that satisfies the constraint. The user obtains the physical stencil plate by sending the result to a cutting plotter when available. Otherwise the user can print out the final tracing outlines using a standard printer and cut these lines using a cutter knife. Finally the user decorates the target object (e.g., fabric, letter, or postcard) using the stencil plate. In a workshop for novices, we observed that even children can design their own stencil using our system.

Keywords: design, interactive system, stencil, nonprofessional users

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表現手法である。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用いられており、ロゴやシンボルなどをデザインする際にもよく使われる（図1）。ステンシル型版は positive と negative

¹ 筑波大学
University of Tsukuba, Tsukuba, Ibaraki 305-8573, Japan
² 日本学術振興会
JSPS, Chiyoda, Tokyo 102-8472, Japan
³ 東京大学
The University of Tokyo, Bunkyo, Tokyo 113-0033, Japan
⁴ JST ERATO
JST ERATO, Bunkyo, Tokyo 112-0002, Japan
a) yukim@acm.org

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．ステンシルにおいてはこの「positiveな領域はすべてつながっている」という制約が一番大事であるが，子どもや初心者がこの制約を満たすようにオリジナルなステンシル型版をデザインすることは難しい．そのため，子どもや初心者はサンプルパターン [1] の組合せや簡単にアウトラインをトレースすればよいような図柄をデザインするにとどまっている．また，ステンシルを手作業で作成しようとするとき，実際に紙に絵を描いてどこをつなげるか決定した後にかッターで切り取っていく作業になるが，切り取る作業はやり直しがきかないため，何度も紙の上で試行錯誤を繰り返すことになり，これは非常に面倒で煩雑な作業である．そこで，我々は子どもや初心者のユーザでも手軽にステンシル型版をデザインできるツールを提案する．

本システムは初心者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描いていくことができる．システムは自動でユーザの入力した線をもとに対話的にステンシル型版を生成していく．本システムを用いることでステンシルの制約を満たしながら，インタラクティブに自由



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/>

Fig. 1 Example of the user of a stencil plate and a real stencil. In a valid stencil such as this, the template is a single connected piece of material. All islands are connected by a bridge to the main sheet.

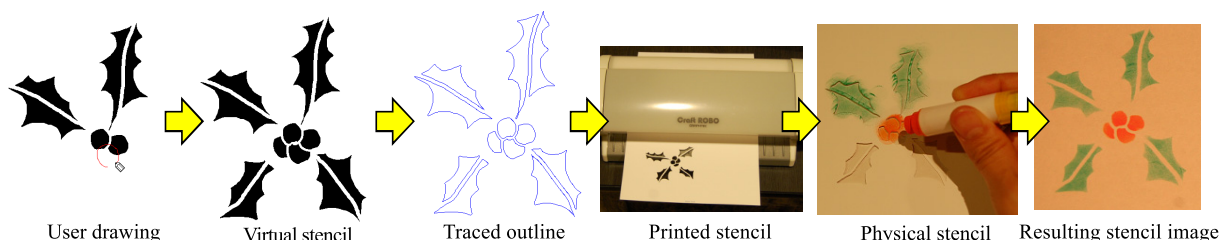


図2 システムの概要．ユーザがデザインした図柄からシステムは対話的にステンシル型版を生成する．カッタブリッタなどを用いて実際のステンシル型版を生成することができ，紙や布などにステンシルをする

Fig. 2 Overview of our system. A user draws freeform strokes. The system automatically generates a virtual stencil plate. The user designs a physical stencil plate using a cutter knife or a cutting plotter. Finally, the user adds pigment using the stencil plate.

な形状がデザインでき，サンプルパターンの組合せではない本当に独自のデザインを創ることができる．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．カッタブリッ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，子どもや初心者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，システムはベクタデータとラスタデータを保持している．ユーザのストロークはベクタデータのプリミティブとして保持しており，システムは boolean operation を適用してラスタ画像に変換している．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データとして出力する．

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画像形式（パスごとの構成）とラスタ画像形式（ピクセルごとの構成）の大きく2つの種類に分類される．ベクタ画像形式では点列，線，カーブなどの幾何学的なプリ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 [2] 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 [3] を提案した．Asente らは planar-map-based ですべてのものと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 [4]．

一方，ラスタ画像形式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．Adobe Photoshop や Microsoft PaintBrush などがラスタ画像形式の代表ソ

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[5]。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[6]。

Bronsonらは3次元ポリゴンモデルや2次元画像データからステンシル型版を生成する手法を提案した[7]。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どうし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。また、2次元の画像から1枚の紙につながった papercutting へ変換する技術も提案されている[8]。一方、我々の提案システムではユーザのスケッチ入力からインタラクティブにステンシル型版をデザインしていく。機械的バッチ処理とは異なり、“ステンシルをデザインをする”といった側面から考えると、読み込んだ画像を一括変換でステンシルに変換してしまうよりも自然であり、かつ描いている最中にユーザにフィードバックするインタラクティブシステムであるため、ユーザの意図に沿ったものがデザインできると考える。

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(negative ポリラインオブジェクト, negative ポリゴンオブジェクト, positive ポリラインオブジェクト, positive ポリゴンオブジェクト)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 positive 領域をつなげる処理を行う。詳細は後述する。

ユーザはまず、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。本システムではつねに

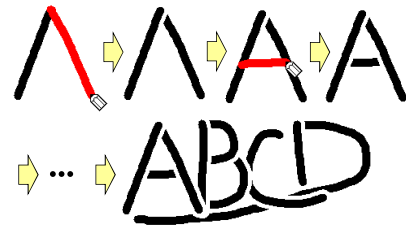


図3 ブラシツールを用いてデザインした例

Fig. 3 Example of the brush tool.

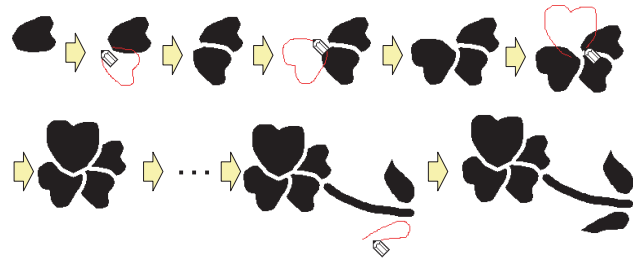


図4 塗りツールを用いてデザインした例

Fig. 4 Example of the fill tool.



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

Fig. 5 The user can erase strokes using (a), (b) the brush-eraser tool; and (c), (d) the fill-eraser tool.

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。

システムはオーバライティングモードとアンダ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。

内部的には、システムはすべてのブラシストロークと消



図 6 ストロークを動かした例
Fig. 6 Example of moving a stroke.



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
Fig. 7 The user can erase stroke using pop-up menu.

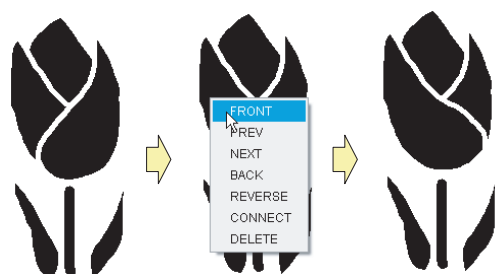


図 8 描かれたストロークの順序を変更することができる
Fig. 8 The user can change the order.



図 9 自己交差の例
Fig. 9 Example of self-intersection.

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「リバーシ」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、ストロークどうしがグループとして記録される。ユーザは矩形選択でプリミティブを複数選択して、コピー・ペース

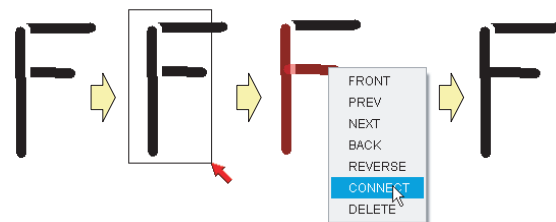


図 10 ストロークをつなげることもできる
Fig. 10 The user can connect strokes without a margin when desired.

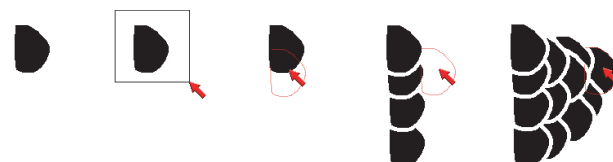


図 11 コピーとペーストの例
Fig. 11 Example of copy and paste.

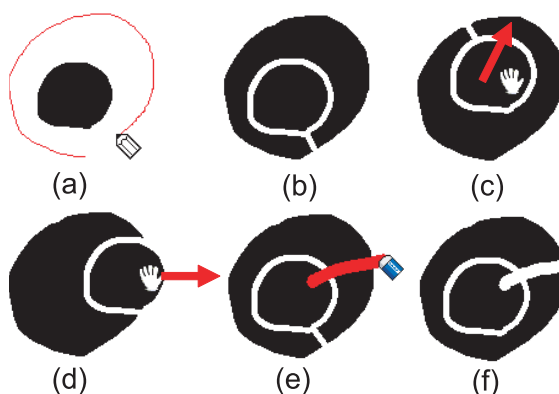


図 12 システムは自動的に島を検出し、橋を生成する。必要のない橋は消去される
Fig. 12 Example of automatic bridge creation mode. The system automatically detects an island and adds a bridge to connect it to the main sheet.

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 領域ができてしまうことがある。この領域を本稿では「島」とよぶ。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 領域を生成する。この領域を本稿では「橋」とよぶ。たとえば、図 12 (a) のようにユーザがアンダ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 12 (b))。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 12 (d))、橋を生成する必要がなくなった場合には橋を消去したりする (図 12 (e))。この自動で挿入される橋の位置の決め方は複数考えられる。審美的観点から考えると、たとえば、左右対称の図形の場合には対象性を保つような位置に橋は挿入されると良いと考えられる。また、曲線的なデザインの場合

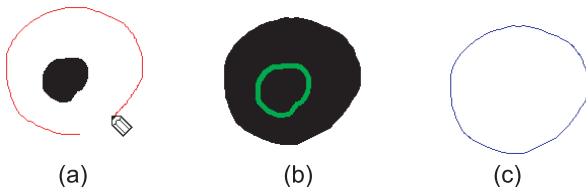


図 13 警告モードの例. ユーザが描いているときに島ができるとその領域をハイライトしてユーザに示す

Fig. 13 Example of warning mode. In warning mode, the system automatically warns the user by highlighting the isolated region with a green boundary.

には橋も曲線的に挿入されると良いと考えられる. 本システムではユーザの入力はフリーハンドのデザインであるため, 自動生成される橋はユーザの描いたデザインに対する変更を最小限にするため, 生成される橋の長さが一番短く生成できる位置に直線で挿入されることとした. この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 positive 領域がつながるようなデザインを描くことができる. その場合, システムが自動で生成した橋は必要なくなるので自動的に消去される (図 12(f)). ユーザは橋やマージンの幅を Pop-up ウィンドウで変更することができる. これらは最終的に作るステンシル型版の素材に応じて強度が異なるため, それをもとに決定するとよい.

ユーザは自動橋生成モードを切ることでもできる. 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 13(b)). この領域がある状態で最終的なベクタ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 13(c)).

システムは最後にステンシル型版をベクタ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で実際のステンシル型版を得ることができる. 描かれた線に沿って切る作業ではカットプリンタ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 JavaTM で実装しており, 内部的にはベクタ形式とラスタ形式の両方を組み合わせたハイブリッドな実装になっている. ブ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. ステンシル型版画像は図 14 のように positive と negative 領域の boolean ラスタ画像である. システムは保持されているス

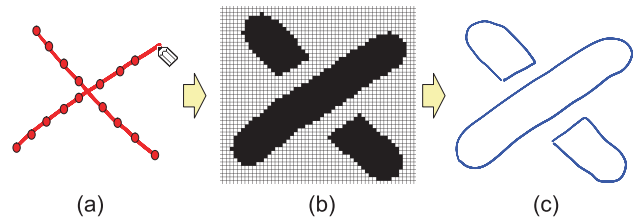


図 14 ステンシル型版画像は Boolean ラスタ形式で保持されている

Fig. 14 The stencil image is represented as a Boolean image.

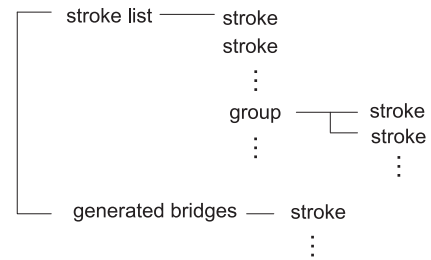


図 15 提案システムのデータ構造

Fig. 15 Data structure of our system.



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)

Fig. 16 Algorithm for drawing a stroke on a raster image.

トロークのリストをステンシル型版画像の上に 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, オーバライティング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画像に描く. システムは次に通常の幅で negative ストロークを描く (図 16). このように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。初期設定のアンダライティングモードではリストの先頭に追加され、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、我々は以下のような処理をしている。ユーザが通常のプロ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よばれるたびに、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持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ぎ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。プリミティブを描画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。

ユーザがデザインしている間に、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ことがある。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(図17(a), (b))。まず、flood fill アルゴリズムを適用してpositive画素を抽出し、画面内にpositive領域が複数存在しないかチェックする。具体的には左上のpositive画素をシードとしてflood fill アルゴリズムでpositive領域を抽出する。次にまだ抽出されていないpositive画素があれば、その画素をシードとしてflood fill アルゴリズムでpositive領域を抽出する。これをpositive画素がすべて抽出されるまで繰り返す。positive領域が1つの場合には島は存在しないため処理は何も行わない。positive領域が複数存在する場合(図17(c))には、システムは2つの領域の最短距離をすべての領域の組合せにおいて計算し、その距離を辺のコストとし、領域を頂点とする完全グラフを生

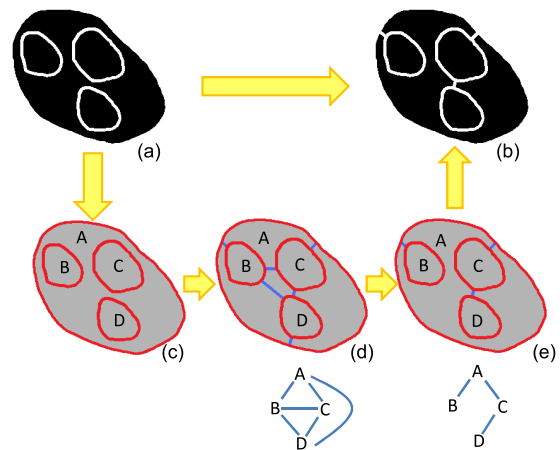


図 17 自動橋生成アルゴリズム

Fig. 17 Algorithm for automatic bridge creation.

成する(図17(d))。次に、最小木に含まれるエッジに橋を生成してつなげる(図17(e))。この橋は橋リストとしてプリミティブに保存する。これをpositive領域が1つになるまで繰り返す。これらはユーザが画像を編集するたびに再計算される。

システムはカットプリンタなどで出力するために、ベクタ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[9]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。最後にベクタ形式として保存する。

実際のステンシル型版を印刷した後、黒い領域を切り抜く時間はこのトレースした線の合計に比例する。ストロークの長さを L_{stroke} 、単位あたりの L_{stroke} を切るのにかかるコストを W_{cut} とすると、切り抜くのにかかる時間の合計(T_{total})は、

$$T_{total} = W_{cut} L_{stroke}, \quad (1)$$

で計算される。本システムでは切り抜く時間を手動と自動(カットプリンタを用いた場合)の両方で提示した。手動の場合は文献[10]より、 $W_{cut} = 1.1 \text{ s/cm}$ とし、自動の場合はCraft ROBO[11]の取扱説明書より、 $W_{cut} = 0.5 \text{ s/cm}$ とした。

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。図18は本システムを利用して著者らがデザインしたステンシルの図柄の例である。また、本システムの有効性を確かめるため、予備実験を行ったうえで、ユーザスタディを兼ねた子ども向けワークショップを行った。また、インタラクティブシステムに関する国内ワークショップにおいて成人を対象としたデモ展示も行った。

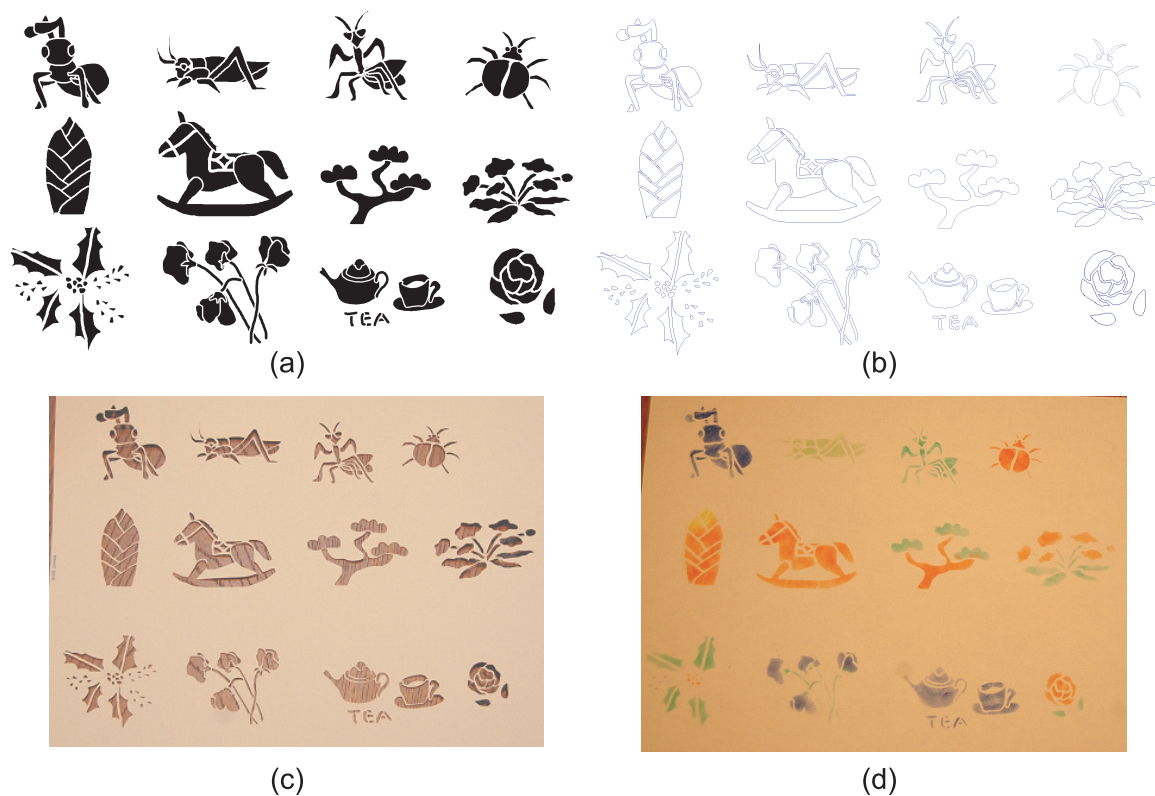


図 18 Holly システムでデザインしたステンシルデザインと紙にステンシルをした結果

Fig. 18 Stencil plate images created using our system, and real results.

5.1 予備実験

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。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トプリンタ [11] 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言った。特に自動で橋ができる機能が使いやすいと言った。

5.2 ワークショップ（小学生対象）

本システムを用いて、一般の子どもたち（8～11 歳）とその親 10 組を対象にユーザスタディを行った。会場は日本科学未来館のワークショップ用の部屋を利用し、対象者は同館主催のワークショップとして友の会会員からの公募で募集した*1。図 19 にワークショップの様子と子どもたちがデザインしたステンシルの例を示す。

最初に 5～10 分ほど全員向けにレクチャを行い、タブレット PC を用いて操作方法を学んでもらった後、実際にデザインをしてもらった。30 分～1 時間程度でたくさんのデザインを楽しんでもらい、保存したものの中から自分の気に入ったデザインをカットイングプロッタを用いて出力した。出力した型版を用いて、布エコバッグに布用イン

クでステンシルを行った。およそ 30 分～1 時間弱で完成した。

子どもたちはシステムが使いやすかったと述べており、タブレットペンも特に問題なく使いこなしていた。「普段使うペイントソフトと同じようにお絵かきが楽しめた」「ステンシルをデザインしたのは初めてだったのでうれしかった」というコメントももらった。

子どもたちはレクチャをする前からタブレットペンを片手にシステムを使って自由に絵をデザインしていた。特に説明がなくても使いこなすことができ、楽しむことができていたようだ。また、近年のタブレットペンのゲーム機普及によりタブレットペンも子どもたちに身近になっており、違和感なく使いこなせたようだった。

ステンシルのツールではブラシツールは細かいところのデザインに用いてあり、主に塗りツールを用いてデザインしたものが多かったようである。子どもたちは独自のデザインを構築したり、未来館のエコバッグを見て、「この図柄もステンシルだね」などと発見したりして、楽しんでいった。また、1 つのデザインを用いて実際にステンシルする際に異なった配色で複数ステンシルを施して楽しむなどの工夫をしている子もいた。

また、一から自由にデザインがしたいと思う人がいる一方、定型のテンプレートを用いて組み合わせたり変形させたりすることで図柄をデザインしたいという人もいた。細

*1 日本科学未来館オリエンテーションルーム 1、友の会イベント、新開発ソフトの体験教室「ステンシルでオリジナルエコバッグをデザイン」（2009/10/25）。



図 19 Holly システムを用いたワークショップの様子

Fig. 19 We ran a small workshop for novices to try our system. (a) Design using the system. (b) Decorate real fabric. (c) Example of original stencils designed by children.

かい絵をデザインしすぎるとカッタプリンタで裁断した後、紙をはがす過程が大変だったという人もいた。これらは今後の課題としたい。

5.3 ワークショップ（成人対象）

次に、成人向けの評価実験として、第 17 回インタラクティブシステムとソフトウェアに関するワークショップ (WISS2009) の参加者を対象にデモ会場において 90 分間にわたり、20 代～50 代の男女に本ソフトウェアを実際に体験していただき、コメントをもらった。1 人あたりの使用時間はおよそ 5～10 分程度であったが、すぐにシステムに慣れ、自由にデザインしていた。

「楽しい」「自分でデザインできておもしろい」「このようなシステムは通常作成した本人しかデザインできないことが多いが、これは簡単にステンシルがデザインできて楽しい」「スマートフォンのアプリに欲しい」など好評なコメントばかりであった。実際にその場でカッティングプロッタを用いてデザインしたステンシル型版を出力して、自らが着ていた T シャツにオリジナルのステンシルデザインを施

した人もいた。

体験者の 1 人からは、“ステンシルは「面を用いて図柄をデザインしていく」という普段の線画とは異なった体験ができることが利点であり、ステンシルの特徴であるため、子どもたちに向けてワークショップを開催するなど創作支援学習を目標にした場合には、面を使ったデザインを意識して作業させるよう促すのが良い”とのコメントもいただいた。

一方、塗りツールではなく、ドローツールを用いてデザインしようとした場合には、文字や数字、細かい部分はデザインしやすいが、動物や既存のキャラクタなどをデザインするには島が多く発生してしまうため向いていないことも分かった。

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 “Holly” を提案した。内部的には入力ストロークはベクタ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形式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プリンタなどを用いて作ることができ、実際のステンシルを楽しむことができる。我々は子どもや初心者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。

現在の自動橋生成アルゴリズムは橋の本数は1本、および幅はユーザが定めることとしている（橋の幅のデフォルトはマージンと同じ幅を使用している）が、橋は島の面積や外周の長さ、材質の厚みなどによって幅を変化させる必要がある。たとえば、円をブラシツールで描いた場合には、円を大きくすると、橋が1つであれば、島の保持ができなくなる限界というものがある。そのようなときに専門家であれば強度を増すために橋を追加することを行う。本システムでもユーザが橋の強度が保てないと思った際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 positive 領域がつながるようなデザインを描くことができるがこれを判断するには知識や経験が必要である。また、橋の本数を固定し、橋の太さを変化させてもよいなど、複数の解が考えられ、答えは一意には決まらない。さらに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。今後ユーザスタディなどを行うことでこれらのパラメータを自動推定するなど、より高機能な自動橋生成技術へと発展させたい。シミュレーションを適用してステンシル型版の強度を確かめ、強度が足りない部分に橋を適宜自動挿入するといった方法も考えられる。

本システムは初心者を対象として、設計デザインを行い開発したものである。実際にステンシルデザインを施している専門家に使用していただくためには、ベジエ曲線など綺麗な線や図の入力および編集できるようツールの改善が必要であると考え。また対称な図柄を描く際の支援、実際にカラーインクを載せた際の雰囲気に応じて図柄を編集し直すためのカラーサポートなどが必須になると考える。これらの機能を追加して専門家にとっても有用なツールへと発展させていきたい。

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。

参考文献

- [1] Thomas, P.: *Protest Stencil Toolkit*, Laurence King Publishing (2011).
- [2] Fekete, J.D., Bizouarn, E., Cournarie, E., Galas, T. and Taillefer, F.: TicTacToon: A paperless system for professional 2D animation, *Computer Graphics and Interactive Techniques*, pp.79–90 (1995).
- [3] Gangnet, M., Thong, J.V. and Fekete, J.: Automatic gap closing for freehand drawing, *Technical Sketch, SIGGRAPH 94*, ACM (1994).
- [4] Asente, P., Schuster, M. and Pettit, T.: Dynamic planar map illustration, *ACM Trans. Graphics (SIGGRAPH 2007)*, Vol.26, No.3, p.30 (2007).
- [5] Qu, Y., Wong, T.-T. and Heng, P.-A.: Manga Colorization, *ACM Trans. Graphics (SIGGRAPH 2006)*, Vol.25, No.3, pp.1214–1220 (2006).
- [6] McCann, J. and Pollard, N.S.: Real-time Gradient-domain Painting, *ACM Trans. Graphics (SIGGRAPH 2008)*, Vol.27, No.3, p.93 (2008).
- [7] Bronson, J., Rheingans, P. and Olano, M.: Semi-Automatic Stencil Creation through Error Minimization. *Proc. Non-Photorealistic Animation and Rendering*, pp.31–37 (2008).
- [8] Xu, J., Kaplan, C.S. and Mi, X.: Computer-generated papercutting, *Pacific Graphics 2007*, pp.343–350 (2007).
- [9] Lorensen, W.E. and Cline, H.E.: Marching cubes: A high resolution 3D surface construction algorithm, *ACM SIGGRAPH Computer Graphics*, Vol.21, No.4, pp.163–169 (1987).
- [10] 三谷 純：計算機による立体紙模型の設計支援に関する研究, 2003 年度東京大学大学院工学系研究科精密機械工学専攻博士論文 (2004).
- [11] Graphtec, Craft ROBO, available from <http://craftrobo.jp/index.html>.



五十嵐 悠紀 (正会員)

2005 年お茶の水女子大学理学部情報科学科卒業。2007 年東京大学大学院情報理工学系研究科修士課程修了。2010 年同大学院工学系研究科博士課程修了。博士 (工学)。2010 年より日本学術振興会特別研究員 PD (筑波大学)。ACM Student Research Competition Grand Finalist, 医療情報学会若手研究奨励賞, 先端技術大賞最優秀文部科学大臣賞等受賞。コンピュータグラフィックス, ユーザインタフェースに関する研究に従事。ACM 会員。



五十嵐 健夫 (正会員)

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教授。独立行政法人科学技術振興機構 ERATO 五十嵐デザインインタフェースプロジェクト総括。2000 年東京大学大学院工学系研究科博士課程修了。博士 (工学)。同年より、米ブラウン大学において博士研究員として研究活動に従事し、2002 年 3 月に東京大学大学院情報理工学研究科講師就任、2005 年 8 月より同助教授、2011 年 5 月より同教授。IBM 科学賞, 文部省若手科学者賞, ACM SIGGRAPH Significant New Researcher Award 等受賞。ユーザインタフェース, 特に, インタラクティブコンピュータグラフィックスに関する研究に従事。ACM 会員。