Vol.2018-CG-169 No.8
2018/3/3

漢字フォント用機械刺繍データの自動変換

山谷 佳祐[1,a)]　栗山 繁[1,b)]

概要：機械刺繍で用いられるデータは専用のソフトウェアを用いて製作されるが，専門的な知識が必要とされ，編集にも時間を要する．特に，文字を刺繍として縫い込む機会は多いが，漢字用の刺繍データはその構造やセリフ表現が複雑で画数が多く，アルファベット文字と比較して部位の交差箇所も頻発するので，一般的に縫い順の決定が難しい．ゆえに，本研究では漢字の刺繍データを手軽に製作できる環境の構築を目指す．本稿では，入力となる漢字画像に対する輪郭抽出やステッチ抽出に基づき，ステッチデータをフィッティングさせることで，異なる漢字フォントへ刺繍データを自動的に転写する手法を提案する．複数の異なる漢字フォントのデータを用いて，トポロジーが同一および異なる漢字フォントに対して刺繍データを変換した結果を示す．

Automatic conversion of machine embroidery data for kanji fonts

Keisuke Yamaya[1,a)]　Shigeru Kuriyama[1,b)]

1. はじめに

刺繍とは生地に刺繍針と糸で施す装飾であり，イラストや図 1 に示す漢字のような文字を題材としてデザインする．刺繍には手作業で生地に糸を通して作成する手刺繍と，大型のミシンなどの機械を使用する機械刺繍の大きく 2 つに分類される．

機械刺繍における刺繍では，刺繍データ作成ソフトウェアを用いてそのデザインを作成し，針の移動量や糸切などの命令を記述したバイナリデータ（刺繍データ）を刺繍ミシンへ出力するという方法が一般的である．近年では，手書きのイラストや自然画像から刺繍データを自動生成し縫製できるなど，家庭用刺繍ミシンの高性能化が進んでいる．

刺繍に関する研究として，ステッチの輪郭を抽出するアルゴリズム [1] やイラスト画像を縫い分けるための領域分割手法 [2] が提案されている．刺繍のモデリングやレンダリングに関する研究として，線画を入力として刺繍モデリングした後 3D モデルへとテクスチャマッピングする手法 [3] や，自然画像を入力として刺繍ステッチのレンダリングをシミュレーションする手法 [4][5] が提案されている．また入力画像を Superpixel に分割し各要素でスタイル転写させて刺繍レンダリングする手法 [6][7] やユーザー入力を得ながらステッチのレイアウトをシミュレーションする手法 [8]，輝度を保存したノンフォトリアリスティックレンダリングによる刺繍シミュレーション手法 [9] が提案されている．しかしながら，いずれも刺繍の結果をレンダリングする手法であり，縫い順のデータを生成するものでは無い．

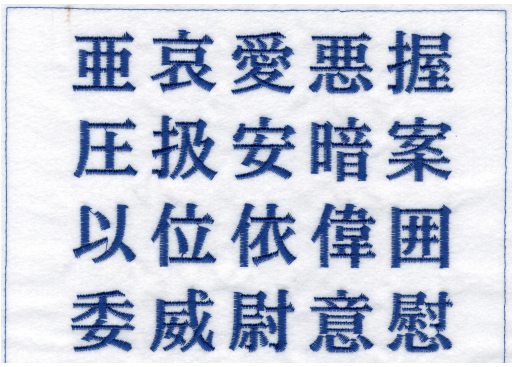


図 1: 漢字フォント刺繍

関連する製品として，既存のオーサリング製品では入力したイラストやフォント（文字）に対する刺繍データ作成が可能である．しかし，ソフトウェアによる刺繍データ作

[1] 豊橋技術科学大学
TUT, Toyohashi, Aichi 441–8580, Japan
[a)] yamaya@val.cs.tut.ac.jp
[b)] sk@tut.jp

(a) 手作業で製作された見本　(b) 既存手法を用いた生成結果

図 2: 縫い経路と刺繍

成は専門的な知識が必要とされ，編集にも時間を要する．特に漢字は交差箇所や画数が多く，縫い上がりを考慮した縫い順は一般的な漢字の書き順とは異なるためその決定は難しい．

図 2 に市販のソフトウェア（以後，既存手法と呼ぶ）を用いて縫い経路を生成した結果と実際に刺繍した結果を示す．図 2(a) は，機械刺繍機の製造会社が手作業で製作したデータを用いた結果である．アルファベットとは異なりその種類が多い漢字における刺繍データ製作の作業は，フォントごとに細かなセリフやエレメントなどの装飾を刺繍として反映できるように施す必要があり，熟練者にとっても労力を要する．既存手法ではフォントのアウトラインをスキャンして縫い順を生成するが，図 2(b) の生成結果を見ると見本データと異なる縫い順が生成されており，生地の裏側の縫い目が絡んでしまうなどの原因となっている．

そこで本研究では，見本データの縫い順から熟練の技を転写するために，見本の変形によって縫い順のデータを生成する手法を提案する．本稿ではユーザーが入力する漢字画像に対する輪郭抽出や骨格形状抽出に基づき，熟練者によって製作されたステッチと呼ばれる縫い目をフィッティングさせることで，異なる漢字フォントへ刺繍データを自動的に転写した結果を報告する．

また，提案手法によって変換した刺繍データを用いて，実際に刺繍ミシンでフェルト生地へ刺繍し，既存のオーサリング製品による刺繍結果との再現性や縫い上がりを比較・考察する．

2. 刺繍データとシステムフロー

まず最初に，機械刺繍データを扱う上で必要となる用語や縫い方の種類，および刺繍データについて説明する．

ステッチとは縫い方や縫い目を表す用語で，提案手法ではひと針ごとの始点座標と終点座標の組をステッチとして扱う．始点座標を $P_s = (x_s, y_s)$，終点座標を $P_e = (x_e, y_e)$，ステッチ数を n とすると各ステッチ $\boldsymbol{S}_i$ は以下で表される．

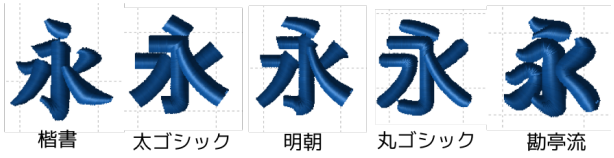


図 3: 刺繍対象フォント

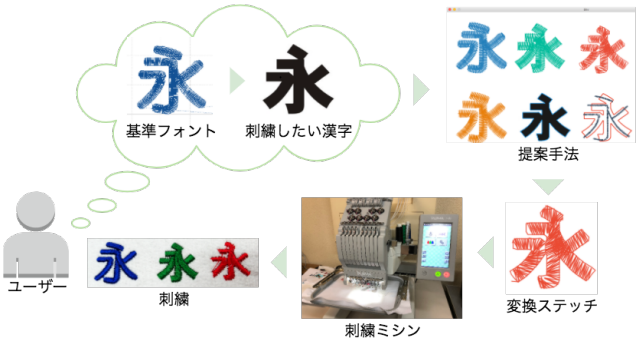


図 4: システムフロー

$$\boldsymbol{S}_i = \begin{bmatrix} x_e - x_s \\ y_e - y_s \end{bmatrix}, \ 0 \leq i \leq n \tag{1}$$

機械刺繍で用いられる縫い方の技法として様々な種類が存在するが，本研究で扱う文字のステッチにはサテンステッチと呼ばれる縫い方が用いられる．これは領域のアウトラインにのみ針を落とす方法 [10] である．イラストに比べまとまって縫い込む領域やレイヤーが少ないため，漢字の刺繍ではサテンステッチでデザインを形成していく場合がほとんどである．

本研究では DST データと呼ばれるフォーマットに限定し，以下の条件下で刺繍データ変換する．

- 移動量の最大値は x，y 軸方向に± 12.1mm
- 縫い目の最大登録数は約 485,700 針
- 色替，糸切処理はバイトデータ内に記述

刺繍の対象とするフォントは変換元の刺繍データが存在する図 3 の 5 つのフォントを用いる．

本研究のシステムフローを図 4 に示す．まずユーザーが変換したい漢字画像を入力として，変換の基となるフォントを選択する．次に実装したアプリケーションが変換の基となるフォントのステッチデータをベースに変換したい漢字に合わせてフィッティングし変換した刺繍データを生成する．この変換した刺繍データを刺繍ミシンへ送信し刺繍生地へ刺繍する．

3. 提案手法

本章では，刺繍データを任意のフォントへ自動変換するアルゴリズムについて述べる．

3.1 提案手法のフロー

提案手法の流れを図 5 に示す．大きく 3 つのステップで

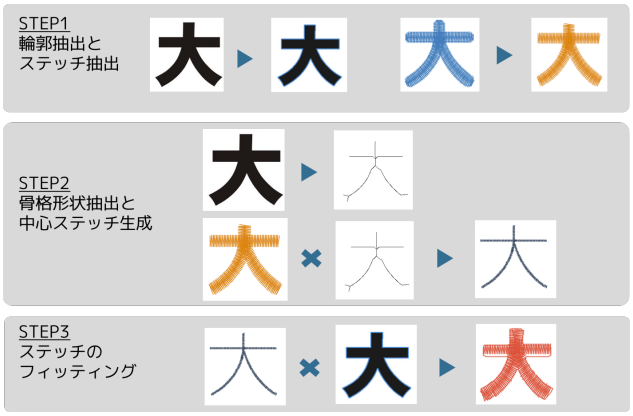


図 5: 提案手法のフロー

刺繍データを変換する．はじめにステッチのフィッティングに用いるため入力画像の輪郭を抽出する．もう一つの入力である基準フォントのステッチから輪郭を縫うステッチを取り除いて輪郭の中を埋める縫い目を抽出する．次に後述する中心ステッチ生成のため漢字画像から骨格形状を抽出する．STEP1 で抽出した輪郭中のステッチと漢字の骨格を基に中心ステッチを生成する．そして生成した中心ステッチに基づき漢字内部の内側（中心）から外側へ向かってステッチ座標の始点と終点を漢字画像の輪郭形状に対し平行移動することでフィッティングする．

以上のステップによって基準ステッチから変換したいフォントへの自動変換を実現する．

3.2 輪郭抽出とステッチ抽出

入力となる漢字画像に対して鈴木らのアルゴリズム [11] により輪郭抽出する．漢字画像は単漢字の画像であり，これをグレースケール化し 250×250 画素にスケール変換する．漢字によっては輪郭が複数得られる場合があり，輪郭形状を C_i，得られた輪郭の数を m とするとその集合を $\boldsymbol{C} = [C_1, C_2, \cdots, C_m]$ として保持し後述するフィッティングのステップにおいてステッチ座標の内外判定に用いる．

次に入力となる基準フォントのステッチ（図 5 中の青色で示したステッチ）から輪郭を縫うステッチを除いた，輪郭の内側を埋めるステッチを抽出する．これは後述するフィッティングにおいてその対象となるのが中を埋めるステッチであるため，あらかじめ基準ステッチから除外しておく．図 6 に縫い目の拡大図のように，輪郭中を埋めるステッチの基本的な縫い方は骨格に対し垂直方向と斜め方向のステッチの繰り返しになっている．この縫い方の特徴から，2 組のステッチをそれぞれベクトル $\boldsymbol{a}, \boldsymbol{b}$ とおき，ベクトルのなす角を θ とすると角度 θ は次式で得られる．

$$\theta = \cos^{-1} \frac{\boldsymbol{a} \cdot \boldsymbol{b}}{||\boldsymbol{a}|| \, ||\boldsymbol{b}||} \tag{2}$$

すべてのステッチがなす角 θ に対し，$0 \leq \theta \leq \pi/9$ を満たせば輪郭中を埋めるステッチとして抽出する．

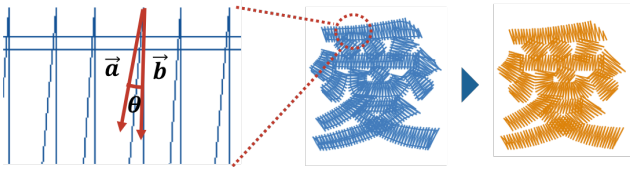


図 6: 縫い目の拡大図と輪郭中のステッチ抽出

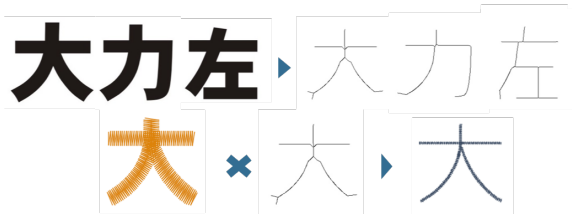


図 7: 骨格形状抽出と中心ステッチ生成

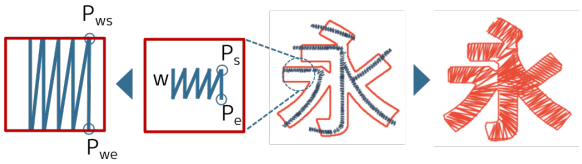


図 8: ステッチのフィッティング

3.3 骨格形状抽出と中心ステッチの生成

漢字画像に対し Zhang-Suen らのアルゴリズム [12] を用いて細線化処理し，文字本来の姿を示す線である「骨格」（文字の中心線）を抽出する．この骨格と輪郭中のステッチを重ね合わせ，骨格を通るステッチについてステッチ幅を短縮した中心ステッチを生成する．中心ステッチの幅 w は経験的に $w = L/7$ としている．以上のアルゴリズムによって太ゴシックの骨格形状を抽出した例を図 7 に示す．

3.4 ステッチのフィッティング

3 つ目のステップとして，骨格形状と漢字の内側に存在するステッチ抽出から生成した中心ステッチを輪郭の外側方向へ伸ばして輪郭に対しフィッティングさせる．各ステッチに対して以下の手順をすべての中心ステッチにおいて繰り返す．

(1) 着目ステッチ S_i の始点 P_s と終点 P_e を取得
(2) 輪郭集合 $\boldsymbol{C}$ から $\boldsymbol{S}_i$ の中点 $P_{ct} = \{(x_e - x_s)/2, (y_e - y_s)/2\}$ が含まれている輪郭 C_i を探索する
(3) P_s と P_e それぞれについて，輪郭に含まれている間 w ずつ輪郭の外側方向へ平行移動する
(4) 輪郭の外側へ外れたら，P_s と P_e をそれぞれ新しい座標 P_{ws}，P_{we} として更新する

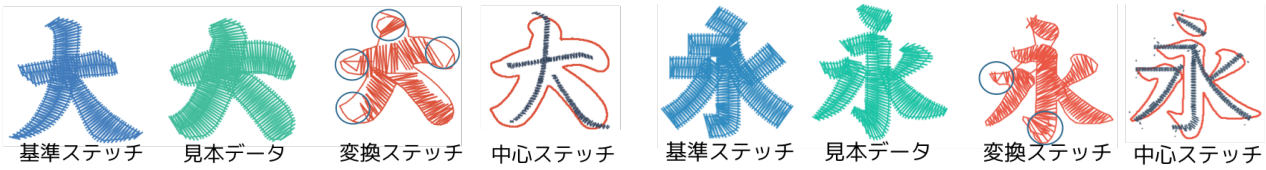
こうして得られた P_{ws}，P_{we} への組を新しいステッチとし，フィッティングした例とその拡大図を図 8 に示す．

4. 刺繍データの変換実験

複数の漢字とフォントを用いて，前章で提案した手法に

(a) 明朝体から楷書体への変換　(b) 丸ゴシック体から太ゴシック体への変換

図 9: 類似フォント間の変換例



(a) 楷書体から勘亭流体への変換　(b) 太ゴシック体から楷書体への変換

図 10: 異種フォント間の変換例

より縫い経路を変換し既存手法で生成した刺繍データとの比較によりその変換精度を検証する．また刺繍ミシンを用いて変換した刺繍データを刺繍生地へ刺繍し，縫い上がりやフォントの再現性の観点からその結果を考察する．

4.1 検証用データセット

刺繍対象の漢字として，「大」「力」「左」「永」「東」「国」「愛」の 7 種類を採用した．書体制作の流れとして，比較的画数が少なく，1 文字のみの単独で構成されている文字から書体制作での流れ・展開が始まること，また「永」「東」「国」「愛」の 4 種類は書体制作の軸となる漢字であること [13] より採用した．これら 7 つの漢字を図 3 で示した 5 つの刺繍対象フォントで見本データ（基となるフォントのステッチから変換して生成したいステッチ）として，計 35 種類の刺繍データを用意した．

次に 7 種類の漢字それぞれについて，変換に用いるため 5 つの刺繍対象フォント画像を用意した．これにより各漢字について刺繍データと漢字画像あわせて 14 ずつあり，全漢字の合計 70 の刺繍データと漢字画像のデータセットを構築した．

このデータセットを用いて，提案手法を実装したアプリケーション上で 7 つの漢字それぞれについて基準となるフォントを固定し，異なるフォントへ変換し刺繍データを生成する．変換した刺繍データを刺繍ミシンへ送信し，刺繍糸を色分けしてフェルト生地へと縫い込む．

4.2 変換結果の検証

提案手法により自動変換した結果として，漢字「大」を明朝体から楷書体へ変換した例を図 9(a) に示す．フィッティング前の重ね合わせを見ると，中心ステッチを文字の中心線である骨格形状に近い形で重ね合わせることでフィッティングの結果も向上し，理想的な変換ステッチが得られている．

再現性の観点から見ると，丸印で示した箇所ではステッチが輪郭に対して埋めきれていないことがわかる．基準ステッチと見本データを見比べると，基準ステッチではステッチが右方向になっているのに対し見本データでは左方向にステッチが伸びている．こうしたフォントに特有な字面，ふところ，エレメントといった要素の再現が変換において本質的に難しいことを示唆している．

次に漢字「永」を丸ゴシック体から太ゴシック体に変換した例を図 9(b) に示す．ステッチ形状に比較的類似性が見られるゴシック体同士の変換例であり，フィッティング前の重ね合わせを見ると骨格形状が変換対象の太ゴシック体においてもほぼ変化していない．丸印で示した箇所では輪郭形状に対してステッチ間隔が空いているまたは埋められておらず，このような部分は刺繍ミシンで刺繍した際に生地の部分が見えてしまうことや縫い込みが緩くなってしまうことが予想される．しかしながら全体的には基準ステッチをベースに太ゴシック体の輪郭に対してステッチ間の隙間を少なく縫い込んでおり，丸ゴシック体から太ゴシック体を再現できる理想的な結果が得られている．

続いて異種フォント間の変換例を図 10 に示す．図 10(a) の楷書体から勘亭流体への変換では骨格形状こそ類似性があるもののセリフ部分において輪郭に対し埋めきれていないステッチが多い．縫い込みがきれいな変換結果を得るには、ある程度変換先のフォントに類似した特徴を持つフォント同士である必要があると考えられる．

図 10(b) ではゴシック体から楷書体というセリフ部分やエレメントが異なる変換であるが，中心ステッチを合わせることでセリフ部分まで再現できる結果を得た．丸印で示した明朝体のセリフ部分は再現が難しく，ステッチの補間

(a) 楷書体から明朝体　(b) 太ゴシック体から丸ゴシック体　(c) 太ゴシック体から丸ゴシック体

(d) 太ゴシック体から楷書体　(e) 丸ゴシック体から太ゴシック体　(f) 太ゴシック体から丸ゴシック体

図 11: 提案手法による縫い順生成と刺繍ミシンを用いた刺繍．左から青の刺繍が基準ステッチ，緑の刺繍が見本データ，赤の刺繍が提案手法により変換したステッチ．

が必要であることがわかる．

4.3 刺繍ミシンを用いた縫い順生成結果の検証

図 11 に，提案手法によって変換した刺繍データを刺繍ミシンでフェルト生地へ刺繍した結果をその縫い経路と共に示す．図 11(d)(e)(f) においては大きく飛び越えるような縫い経路や輪郭の外側を縫うステッチが見られる．また特に図 11(c) のように文字そのものの空間であるふところが密な場合には，使用する糸の太さや縫い込む領域の面積比などの考慮が必要となってくる．このように実際に刺繍として縫い込んで始めてわかる発見も多く，縫い上がりの向上には輪郭外を経由するあるいは大きく飛び越えるようなステッチを観測した場合は糸切り命令を刺繍データに埋め込むなどの処理が必要と考えられる．

図 11(a)(b)(d)(e) では漢字の交差箇所やセリフ部分においてステッチが欠落している．このような補間が必要な箇所の位置情報の獲得，またどのような補間が必要かは交差箇所かセリフ部分かで異なるアプローチが必要であろう．しかしながらそのためには変換元と変換先のフォントを比較し部首や辺の対応関係を求める必要があり，補間の方法によってはステッチ数が増加することや均等な縫い目でなくなることが予想される．したがってこうした状況では対話的な操作で補間が必要な箇所を選択する機構も検討の余地がある．この場合，文字の骨格形状に合わせる操作をユーザーに入力してもらうことで，ユーザーが変換結果を逐次確認しながら中心ステッチの位置を合わせることにより，変換精度の向上が期待できる．

4.4 TrueType フォントへの変換

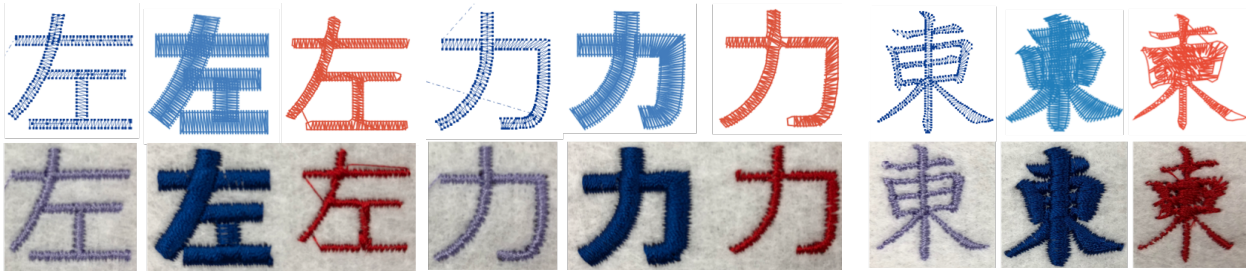
提案手法の応用として，刺繍対象フォントからデータセット以外の TrueType フォントへ変換し刺繍した．図 12 に既存手法で生成した TrueType フォントの縫い経路とその刺繍，また提案手法によって変換した例を示す．既存手法ではフォント形状に対し一定の間隔で隙間なく縫い込めるように縫い経路を生成するが，図 12(c)(e) の縫い経路のように，見本データとは異なり骨格形状に対して垂直に縫えていない部分が確認される．

図 12(b) に既存手法での生成に使用したフォントへ提案手法により変換した例を示す．基準フォントはいずれも変換対象のフォントに形状が類似したフォントを基にしているが，メイリオや創英角ゴシック体のように再現性高く縫える例もあれば，ステッチが欠落する場合やステッチが詰まり過ぎて品質を損ねている例も確認された．また漢字のある部位から他の部位へ大きく飛び越えるようなステッチや縫い込みが緩くなっている箇所が存在するため，縫い目を整える工程も必要であろう．

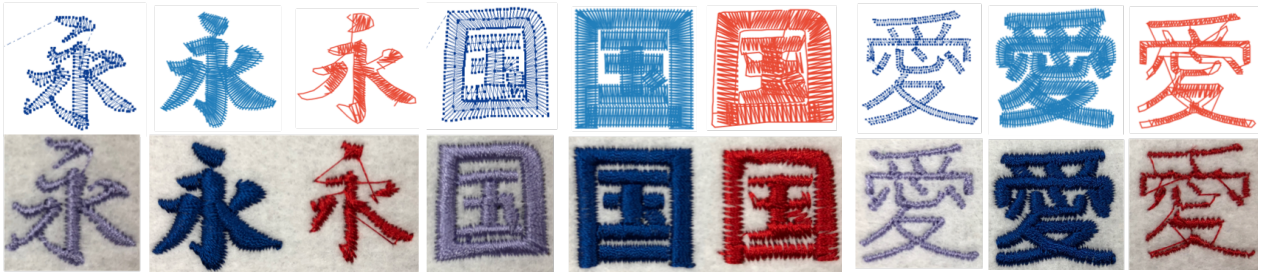
5. おわりに

本稿では，既存の刺繍フォントのステッチに基づき，任意のフォント画像に対して縫い経路の変換により漢字フォントの刺繍データを生成する手法を提案した．また刺繍ミシンにより生地へ縫い込むための刺繍データ変換環境を構築した．複数の漢字フォント同士へ変換する実験により，トポロジーが同一および異なる漢字フォントに対して刺繍データを生成でき任意の TrueType フォントへも変換とその刺繍ができる結果を示した．現時点で発見された欠点を解消できれば，将来的には既存手法よりも高品質な刺繍データが生成できる可能性がある．

今後の課題として，ステッチ補間によるセリフ部分の再現性向上や，補間が必要な箇所や漢字のパーツ指定ができ



(a) 太ゴシック体からメイリオ　(b) 太ゴシック体からメイリオ　(c) 楷書体から HGP 教科書体



(d) 楷書体から行書体　(e) 太ゴシック体から HG 創英角ポップ体　(f) 丸ゴシック体から游ゴシック体

図 12: TrueType フォントへの変換例．紫色が既存手法による生成データの刺繍，青色が基準フォントの刺繍，赤色が提案手法によって変換したステッチの刺繍．

るユーザーインタラクションの導入を挙げる．また刺繍データ数が増加できれば，ステッチの前後依存関係の学習によって漢字の骨格形状に対する最適なステッチ幅を学習させるアプローチも考えられる．

本稿では漢字を対象としたが，より複雑な構造を有する繁体字やハングル文字，さらにはアラビア文字といった言語の刺繍データ生成への応用も今後の課題である．

参考文献

[1] W. Dongqing, Y. Fengjian, Z. Chaolong, and H. Xiaojian. Research in embroidery stitch contour restoring algorithm. In *2009 Second International Symposium on Electronic Commerce and Security*, Vol. 2, pp. 52–55, May 2009.

[2] D. Wu, F. Yang, X. Hu, C. Zhang, and J. Yang. A stitch partition algorithm based on bp neural network. In *2009 Second International Symposium on Knowledge Acquisition and Modeling*, Vol. 3, pp. 144–146, Nov 2009.

[3] Xinling Chen, Michael McCool, Asanobu Kitamoto, and Stephen Mann. Embroidery modeling and rendering. In *Proceedings of Graphics Interface 2012*, GI '12, pp. 131–139, Toronto, Ont., Canada, Canada, 2012. Canadian Information Processing Society.

[4] Dele Cui, Yun Sheng, and Guixu Zhang. Image-based embroidery modeling and rendering. *Computer Animation and Virtual Worlds*, Vol. 28, No. 2, pp. e1725–n/a, 2017. e1725 cav.1725.

[5] Kewei Yang, Jie Zhou, Zhengxing Sun, and Yi Li. Image-based irregular needling embroidery rendering. In *Proceedings of the 5th International Symposium on Visual Information Communication and Interaction*, VINCI '12, pp. 87–94, New York, NY, USA, 2012. ACM.

[6] Kewei Yang, Zhengxing Sun, Chen Ma, and Wei Yang. Paint with stitches: A random-needle embroidery rendering method. In *Proceedings of the 33rd Computer Graphics International*, CGI '16, pp. 9–12, New York, NY, USA, 2016. ACM.

[7] Kewei Yang and Zhengxing Sun. Paint with stitches: a style definition and image-based rendering method for random-needle embroidery. *Multimedia Tools and Applications*, Jun 2017.

[8] Jie Zhou, Zheng-xing Sun, and Ke-wei Yang. A controllable stitch layout strategy for random needle embroidery. *Journal of Zhejiang University SCIENCE C*, Vol. 15, No. 9, pp. 729–743, Sep 2014.

[9] Qiqi Shen, Dele Cui, Yun Sheng, and Guixu Zhang. *Illumination-Preserving Embroidery Simulation for Non-photorealistic Rendering*, pp. 233–244. Springer International Publishing, Cham, 2017.

[10] これならできる! みんなの教科書 刺しゅうきほんの基本. 高橋書店, 9 2016.

[11] Satoshi Suzuki and KeiichiA be. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*, Vol. 30, No. 1, pp. 32 – 46, 1985.

[12] T. Y. Zhang and C. Y. Suen. A fast parallel algorithm for thinning digital patterns. *Commun. ACM*, Vol. 27, No. 3, pp. 236–239, March 1984.

[13] タイポグラフィの基本ルール [デザインラボ] -プロに学ぶ，一生枯れない永久不滅テクニック-. ソフトバンククリエイティブ株式会社, 5 2013.