

# 小袖屏風画像を利用した 模様画像の合成

Pattern Image Synthesis Using Kosode Byobu Images

白川真一・井田有香・大原剛三・豊田哲也

SHIRAKAWA Shinichi, IDA Yuuka, OHARA Kouzou and TOYOTA Tetsuya

①はじめに

②本研究で使用する画像合成技術

③小袖屏風画像を用いた新しい模様画像の生成

④提案手法の評価実験

⑤おわりに

## 【論文要旨】

歴史資料のデジタルアーカイブ化によって、貴重な歴史資料を計算機で扱うことが容易になり、高度な知的利用や新たな展示技術に関する研究に注目が集まっている。デジタル化された歴史資料にコンピュータグラフィックスやバーチャルリアリティなどの技術を応用することで、インタラクティブな展示やバーチャルな展示といった従来とは異なる新しい展示形式の実現が可能になる。本研究では、小袖屏風と呼ばれる実物の小袖裂を屏風に貼装した歴史資料に着目する。一般の人々が小袖屏風を鑑賞する際に注目する要素のひとつとして、屏風に貼装された小袖の様々な模様が挙げられる。この小袖中の模様を使って自分自身のオリジナルの模様を簡単に作り出す方法が実現できれば、デジタルデータを活用した新たな小袖屏風の展示システムにつながると考えられる。このような背景のもと、本研究では小袖屏風に存在する模様をユーザが選択するだけで、自動的に新しい模様を生成する手法を提案する。博物館の来館者などの一般ユーザを対象として考えた場合、模様生成の過程で画像編集に関する知識や煩雑な操作を要求することは好ましくない。そこで、画像合成技術である Poisson Image Editing やテクスチャ合成技術を組合せ、ユーザが選択した複数の小袖屏風の模様画像から各模様の特徴をもった新しい模様画像を自動生成する手法を開発した。提案手法は、(1) 模様部分を取り除いた背景画像の作成、(2) Poisson Image Editing による背景画像への模様の貼り付け、(3) テクスチャ合成による複数の模様を組合せた新しい模様合成画像の生成、の3ステップから構成される。さらに、提案手法で生成した模様画像と類似の模様画像を被験者に画像編集ソフトで作成してもらって評価実験を実施した。実験結果から、被験者が画像を作成するのと比べて、提案手法が安定して短い時間で画像を合成できることを示した。

【キーワード】 小袖屏風画像、テクスチャ合成、画像合成

## ①……………はじめに

歴史資料のデジタルアーカイブ化によって、貴重な歴史資料を計算機で扱うことが容易になり、高度な知的利用や新たな展示技術に関する研究に注目が集まっている。通常、歴史資料に直接触れ資料の細部を見ることは、歴史資料の劣化を防ぐために禁じられているが、歴史資料をデジタル化し、高精細画像などを活用したデジタルアーカイブを構築することで歴史資料に直接触れることなく細部を間近で見ることが可能になる。このように歴史資料のデジタルアーカイブ化は、貴重な歴史資料の永続的な記録や再発見の手段として期待されている。また、2003年に定められた「e-Japan戦略II」ではデジタルコンテンツの充実が目標として掲げられ、博物館や美術館の所蔵資料に関するデジタルアーカイブが必要なコンテンツとして位置づけられている。その結果として、高精細画像を活用したデジタルアーカイブを誰でも手軽に閲覧できる環境が整いつつある。

デジタル化された歴史資料にコンピュータグラフィックスやバーチャルリアリティ、拡張現実などの技術を応用することで、インタラクティブな展示やバーチャルな展示といった従来とは異なる新しい展示形式の実現が可能になる [1]。このような新しい形式の展示は、来館者の興味を引くことが期待できるため、博物館にとっても重要であると考えられる。

本研究では、小袖屏風と呼ばれる16世紀から19世紀に製作された実物の小袖裂を屏風に貼装した歴史資料に着目する。小袖屏風は、状態の芳しくない小袖裂の保存と鑑賞の両立を目的として、昭和初期に野村正治郎によって考案されたものであり、貴重な裂類が多数貼装されているため、染織研究において重要な資料であるとともに美術品としての価値も高い [2]。一般の人々が小袖屏風を鑑賞する際に注目する要素のひとつとして、屏風に貼装された小袖の様々な模様が挙げられる。この小袖中の模様を使って自分自身のオリジナルの模様を簡単に作り出すような方法が実現できれば、デジタルデータを活用した新たな小袖屏風の展示システムにつながると考えられる。

このような背景のもと、本研究では小袖屏風に存在する模様をユーザが選択するだけで、自動的に新しい模様を生成する手法を提案する。博物館の来館者などの一般ユーザを対象として考えた場合、模様生成の過程で画像編集に関する知識や煩雑な操作を要求することは好ましくない。そこで、提案手法では小袖屏風の特徴である模様の画像をいくつか選択するだけで、各模様の特徴をもった新しい模様画像を生成することを目標にする。さらに、提案手法で生成した模様画像と類似の模様画像を被験者に画像編集ソフトで作成してもらい、所要時間を比較することで、提案手法の模様生成効率の評価を行った。

本報告の構成は次の通りである。次章で提案手法で使用する画像合成技術の概要を説明する。第3章では、提案手法のアルゴリズムの説明を行い、第4章で評価実験について述べる。第5章では本研究のまとめと今後の課題を述べる。

## ②……………本研究で使用する画像合成技術

ここでは、提案の模様生成手法で使用する画像合成技術の Poisson Image Editing [3, 4] とテク

スタチャ合成 [5, 6], 顕著性マップ [7] について簡単に紹介する。

## 2.1 Poisson Image Editing

Poisson Image Editing [3, 4] は画像の勾配情報を利用した画像合成技術であり, 厳密にオブジェクトを切り出さなくても, オブジェクトと背景の境界が自然な合成画像を生成することができる。Poisson Image Editing では, オブジェクトが貼り付けられる領域の周辺の画素値と貼り付けオブジェクトの勾配情報から, 合成画像でのオブジェクト内の画素値を推定することで滑らかな合成を実現している。

図1に示すように, 貼り付けオブジェクトの領域を $\Omega$ , その境界を $\partial\Omega$ とし, 貼り付けオブジェクトの画素値を与える関数を $g$ , その勾配を $\mathbf{v}$ , 貼り付け先の画像の画素値を与える関数を $f^*$ , 合成画像の $\Omega$ 上の画素値を与える関数を $f$ とする。合成画像中のオブジェクト領域 $\Omega$ の境界 $\partial\Omega$ では貼り付け先画像の画素 $f^*$ を保存しつつ, オブジェクト内の勾配 $\nabla f$ は元のオブジェクトの勾配 $\mathbf{v}$ に近くなるように合成画像上の画素値を求めることがPoisson Image Editingの目的となる。これを実現するためには, 次の目的関数を最小化するように合成画像中のオブジェクト内の画素値 $f$ を求めればよい。

$$\min_f \int \int_{\Omega} |\nabla f - \mathbf{v}|^2 \quad \text{with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

ただし,  $\nabla = [\frac{\partial}{\partial x}, \frac{\partial}{\partial y}]$ は勾配オペレータである。境界条件を無視すれば, すべての $p \in \Omega$ に対して次の方程式を解くことで, 上記の目的関数を最小化するような解を求めることができる。

$$|N_p|f_p - \sum_{q \in N_p} f_q = \sum_{q \in N_p} v_{pq} \tag{1}$$

ここで,  $f_p$ は合成画像の画素 $p$ の画素値,  $N_p$ は $p$ の上下左右の隣接する画素集合,  $|N_p|$ は $N_p$ の画素数を表す。また,  $v_{pq}$ は貼り付けオブジェクトの画素値の勾配を表現しており, 貼り付けオブジェクトにおける注目画素 $p$ の画素値 $g_p$ と隣接画素の値 $g_q$ の差 ( $v_{pq} = g_p - g_q$ ) で求めることができる。式(1)で定義される多次元一次元方程式は, 一次連立方程式を繰り返すことで解を求めるガウスザイデル法によって解くことができる。

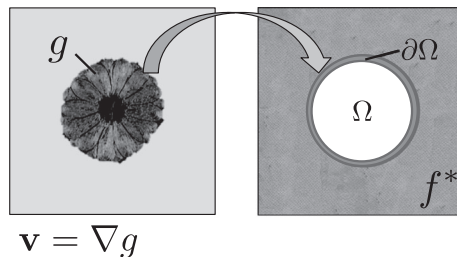


図1 Poisson Image Editingでの貼り付けオブジェクトと貼り付け先画像の例と記号の定義

## 2.2 テクスチャ合成

### 2.2.1 概要

テクスチャ合成技術は与えられた見本となるテクスチャ画像を利用し、その画像に類似した任意の形・サイズの別の新しいテクスチャ画像を合成する技術である [5, 6]。テクスチャ合成を応用することで、CG (Computer Graphics) における物体表面上のテクスチャを覆う際に、自動合成されたテクスチャを利用することで、テクスチャ作成にかかるコストを下げるができるといった利点がある。また、画像中の欠損領域や消したい物体の領域をテクスチャ合成の技術を応用して補完することで、自然な画像編集を実現することができる。

テクスチャ合成の手法は、ピクセルベースの手法 [8]、パッチベースの手法 [9]、最適化ベースの手法 [10]、座標ベースの手法 [11] など様々なものが提案されている。さらに、これらの基本的なテクスチャ合成のアルゴリズムを発展させ、より高度な画像合成を実現するような手法も開発されている。文献 [12] では、類似した構造をもつ画像サンプル集合を入力として、それらに類似しているが微妙に異なる画像のバリエーションを合成する手法を提案している。この手法では、例えば複数のジーンズの画像サンプル集合から、どれとも少し異なる（合成画像中のある部分はサンプル画像のどれかと似ている）合成画像を無数に生成することが可能となる。Park らは異なる複数のテクスチャを自然なかたちでつなぎ合わせ、複数のテクスチャの特徴を併せ持つ画像を合成する方法を提案している [13]。

本研究では、将来的にリアルタイムなテクスチャ合成を実現することを見据えて、並列計算に向けたテクスチャ合成アルゴリズムである座標ベースの手法 [11] を用いる。座標ベースの手法では合成画像の画素値を直接操作するのではなく、合成画像中の各画素値がサンプル画像のどの座標からコピーされたかという座標情報を操作していく。

### 2.2.2 本研究で用いるテクスチャ合成アルゴリズム

本研究では、Lefebvre と Hoppe による座標ベースのテクスチャ合成アルゴリズム [11] を一部改変したアルゴリズムを採用する。このテクスチャ合成アルゴリズムでは、粗い解像度から鮮明な解像度のテクスチャへ合成を進めていき、各解像度での処理は Upsampling ステップ、Jitter ステップ、Correction ステップの3ステップから構成される。Upsampling ステップでは、前ステップの結果画像をアップサンプリングし、Jitter ステップでは合成画像に摂動を加えて合成画像が単純なテクスチャのタイリングにならないようにする効果を導入する。最後に、Correction ステップでは見本画像から近傍画素が類似する画素を探索し合成画像を修正する。

ここでは、 $m \times m$  の見本画像 (exemplar image)  $E$  から、合成画像  $S$  を合成することを考える。合成画像  $S$  の各画素  $S[p]$  は、色情報ではなく見本画像の座標を値としてもつ ( $S[p] \in \mathbb{Z}^2$ )。ここでは簡単のため、見本画像が1枚の場合について説明を行うが、複数枚の見本画像を用いる場合は、合成画像の各画素  $S[p]$  の値に何枚目の画像の座標かという情報を加えれば良い (つまり、 $S[p] \in \mathbb{Z}^3$  となる)。この表記を用いると、合成画像  $S$  上の画素の RGB 値は  $E[S[p]]$  で得られる。

テクスチャ画像の合成では、階層的に画像ピラミッド  $S_0, S_1, \dots, (S_L = S), L = \log_2 m$  を作成し、

各ピラミッドレベルで、Upsampling, Jitter, Correction の3ステップからなる処理を行うことで、徐々に鮮明な画像を合成していく。これによって、最終的に、見本画像に類似のテクスチャ画像が得られる。

各レベルで画像を合成するために、見本画像も各レベルで  $E_0, E_1, \dots, (E_L = E)$  と用意する。文献 [11] では一般的な Gaussian ピラミッドを利用するかわりに、元の見本画像を Gaussian フィルタでぼかしていったもの (Gaussian stack) を利用する。Gaussian stack では画像の縮小は行わず、すべてのレベルの見本画像が  $m \times m$  のサイズをもつ。

以降、各ステップでの詳細な手続きについて説明を行う。

**Upsampling** Upsampling ステップでは次のレベルの合成画像を作成する。具体的には次のレベル  $l$  の合成画像  $S_l$  の各値は次のように決定される<sup>(1)</sup>。

$$S_l[2p + \Delta] := \left( S_{l-1}[p] + \left[ h_l \left( \Delta - \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix} \right) \right] \right) \bmod m,$$

$$\Delta \in \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right\}$$

ここで、 $h_l = 2^{l-1}$  である。

**Jitter** Jitter ステップでは合成画像に摂動を加える。具体的には次のように合成画像がもつ座標情報を変更する。

$$S_l := (S_l[p] + \mathcal{J}_l(p)) \bmod m,$$

$$\text{where } \mathcal{J}_l(p) = \left[ h_l \mathcal{H}(p) r_l + \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix} \right]$$

ここで、 $\mathcal{H}$  は決定的なランダム性を加えるためのハッシュ関数  $\mathcal{H} : \mathbb{Z}^2 \rightarrow [-1, 1]^2$  であり、 $r_l$  ( $0 \leq r_l \leq 1$ ) はユーザが指定するランダム性を制御するパラメータである。この Jitter ステップの摂動によって、見本画像とは異なるテクスチャ画像を生成することができる。

Jitter ステップによって、見本画像とは異なるテクスチャ画像を生成することが可能になるが、今回対象とするような小袖の模様に対しては模様が崩れてしまうという問題がある。そこで、画像中の模様領域を指定するマスク画像を利用し、模様領域の画素については Jitter ステップで変更を加えない工夫 [11] を行う。

**Correction** Correction ステップでは、前の Jitter ステップで摂動を加えられた合成画像を画素近傍の類似性に基づいて修正していく。合成画像中の画素  $p$  の  $5 \times 5$  近傍の RGB 値のベクトルを  $N_S(p)$  と表記する。この近傍と見本画像の近傍  $N_E(u)$  をユークリッド距離で比較し、最も距離の小さい画素を選択する。ただし実際には、メモリと計算効率の観点から前処理の段階で、見本画像の各レベル ( $E_0, E_1, \dots, E_L$ ) の  $N_{E_l}(u)$  を使って主成分分析による、次元削減を行い、削減された次元数の

ベクトルで距離計算を行う。実験では  $5 \times 5 \times 3 = 75$  次元のベクトルを 12 次元に削減することとした。さらに、近傍マッチングの処理を高速化するために、 $k$ -coherence search [14] を用いる。 $k$ -coherence search を適用するにあたって、前処理として見本画像の各画素近傍について類似画素近傍を求めておく必要があるが、この類似探索を高速化するために PatchMatch アルゴリズム [15] を採用する。Correction ステップでは、すべての画素に対して  $c$  回の画素値の更新が行われる。実験では  $c = 8$  としている。

## 2.3 顕著性マップ

提案手法で新しい模様画像を合成する際には、画像中のどの領域が模様であるのかを特定する必要がある。本研究では、小袖中の模様は人間の注視を引くと考えられることと、模様領域を自動で抽出することが望ましいことから、顕著性マップ [7] を利用することとした。顕著性マップとは、画像中で視覚的注意を引き起こす注目箇所（顕著点）を分布図として表現したものである。顕著性マップにも様々な方法があるが、本研究では、近年提案され精度と計算コストの面で優れた Cheng らの Region Based Contrast (RC) マップ [16] を利用する。

人間の注視には、カラーコントラストやオブジェクトの空間的な関係性が重要な役割を果たすと考えられる。RC マップでは、計算コスト低減のため、領域分割された領域間のカラーコントラストや位置関係、領域の大きさなどを考慮して顕著値の計算を行う。

RC マップでは、まず Felzenszwalb らが提案したグラフベースの領域分割手法 [17] によって入力画像を領域分割する。次に、領域  $r_k$  に対する顕著値を次式で定義する。

$$S(r_k) = \sum_{r_i \neq r_k} w(r_i) D_r(r_k, r_i) \quad (2)$$

ここで、重み  $w(r_i)$  は領域  $r_i$  中の画素の数であり、 $D_r(r_k, r_i)$  は 2 領域間の色の距離である。重み  $w(r_i)$  によって、より大きな領域ほど顕著値への影響が大きくなる。これは、小さな領域よりも大きな領域とのコントラストの方が注視への影響が大きいと考えられるからである。また、ある 2 つの領域  $r_1$  と  $r_2$  間の色の距離  $D_r(r_1, r_2)$  は次式で定義される。

$$D_r(r_1, r_2) = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} f(c_{1,i}) f(c_{2,j}) D(c_{1,i}, c_{2,j})$$

ここで、 $f(c_{k,i})$  は領域  $r_k$  に含まれる  $n_k$  個の色のうち  $i$  番目の色  $c_{k,i}$  の割合を表し、 $D(c_{1,i}, c_{2,j})$  は色  $c_{1,i}$  と  $c_{2,j}$  の CIE L\*a\*b\* 色空間内での距離を表す。

さらに、ある領域に注視が向くかどうかは、その周りの領域と比べてコントラストが高いかが重要であり、離れた領域の情報は重要視されないと考えられる。これを実現するために、ある領域の顕著値を計算する際に、そこに近い領域の影響が大きくなるようにし、逆に遠い領域の影響は小さくなるような重みを導入する。これに加えて、人間が画像を見る際は中央の領域ほど注視が向くことを考慮すると、画像中央の領域ほど顕著値が大きくなるような重みを導入することが考えられる。式 (2) にこれらの重みを導入したものが式 (3) である。

$$S(r_k) = w_s(r_k) \sum_{r_k \neq r_i} e^{\frac{D_s(r_k, r_i)}{-\sigma_s^2}} w(r_i) D_r(r_k, r_i) \quad (3)$$

ここで、 $D_s(r_k, r_i)$  は領域  $r_k$  と領域  $r_i$  の空間的な距離を表し、 $\sigma_s$  をパラメータとし、2領域が離れているほど影響が小さくなるような重みを与える。また、 $w_s(r_k) = \exp(-9d_k^2)$  は領域  $r_k$  が画像の中心に近いほど大きな値となる重みである。ただし、 $d_k$  は領域内の画素と画像の中心との平均距離である。式 (3) によって、領域の大きさ、領域間のカラーコントラストと空間的な位置関係を考慮した顕著値が算出できる。また、実際には式 (3) に加えて減色処理などの計算コストを低減する工夫が施される。

### ③……………小袖屏風画像を用いた新しい模様画像の生成

図2に3枚の異なる画像を入力とし、1枚の模様合成画像を出力する場合の提案手法の処理の流れを示す。以下の説明文中の画像A～FおよびB'は図2中の対応する画像を指すものとする。入力画像Aは背景を指定する画像であり、入力画像画像BとB'は合成に使いたい模様を含んだ画像である。

提案手法の処理の手順は次の通りである。

Step I (背景画像の作成) 画像Aから模様部分を取り除いた背景画像Cを作成する。

Step II (模様合成画像の作成) Poisson Image Editing によって画像BとB'の模様部分をそれぞれ背景画像Cと合成し、同じ背景に異なる模様が存在する画像DとEを得る。

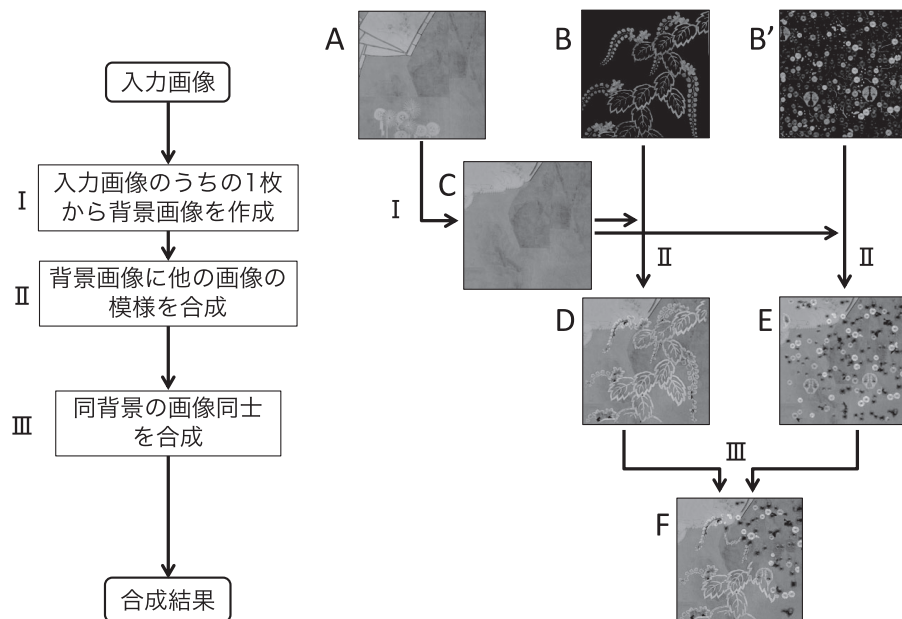


図2 異なる入力画像3枚から、1枚の模様合成画像を出力する場合の提案手法の処理の流れ

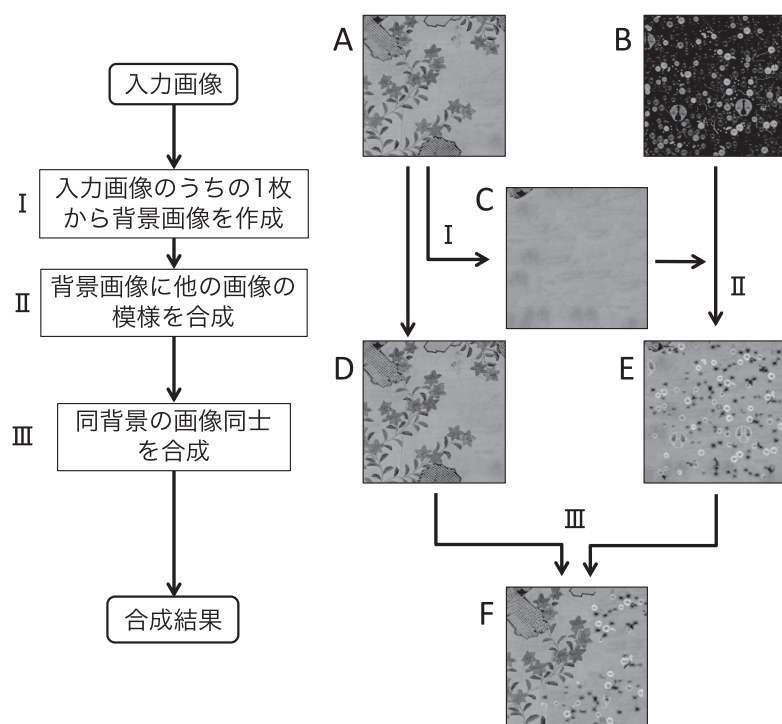


図3 異なる2枚の入力画像から、1枚の模様合成画像を出力する場合の提案手法の処理の流れ

Step III (新しい模様画像の合成) 画像 *D* と *E* を見本画像としたテクスチャ合成を行い、複数の模様を組合せた新しい模様画像 *F* を生成する。

図2では背景となる画像1枚(画像 *A*)と、模様となる画像2枚(画像 *B* と *B'*)を使用した例を示しているが、模様となる画像の一方を画像 *A* で代用することも可能である。この場合、図3に示すように、入力画像は2枚となり、画像 *A* と *D* が同一のものとなる。

次に、各ステップでの処理の詳細について説明を行っていく。

### 3.1 背景画像の作成 (Step I)

背景を指定する入力画像 *A* から、テクスチャ合成によって模様のない背景画像を作成する。ここでは、顕著性マップによって模様領域を二値画像で表現したマスク画像を利用する。顕著性マップには2.3節で説明した Cheng らの RC マップを利用し、得られた顕著値を判別分析法で二値化した画像をマスク画像として用いる。図4 (a) と (b) にそれぞれ背景画像作成の元となる入力画像と顕著性マップによって作成した対応するマスク画像の例を示す。このマスク画像で模様領域と指定された領域以外の画素値だけを使用して合成画像を生成することで、図4 (c) のような背景だけの画像を得ることができる。ここでの画像合成では、均一な背景の画像を作成することが目的であるため、テクスチャ合成の Jitter ステップで使用するランダム性を制御するパラメータ  $r_l$  は、すべての  $l$  に対して  $r_l = 0.0$  とする。



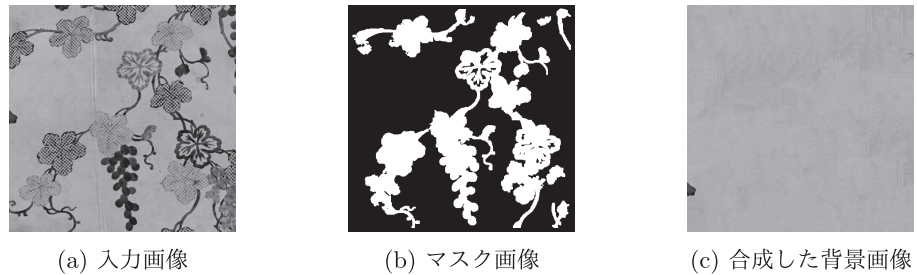


図4 テクスチャ合成によって背景画像を作成する際に使用する入力画像とそのマスク画像, 合成した背景画像の例

### 3.2 模様合成画像の作成 (Step II)

前ステップで作成した背景画像  $C$  に, 模様画像  $B$  と  $B'$  の模様を Poisson Image Editing によって合成することで, 同じ背景に異なる模様が存在する画像  $D$  と  $E$  を作成する。Poisson Image Editing において画像  $B, B'$  中の貼り付け模様の領域を指定するために, 前ステップと同様に Cheng らの RC マップを利用して得られる二値マスク画像を使用する。

単純な Poisson Image Editing では, 模様を合成した際に, 貼り付けた画像の色味が全体的に変化してしまい, 模様本来の色味が失われてしまう問題が生じることを予備実験から確認している。そこで本研究では, Poisson Image Editing で作成した画像と, 模様領域を背景画像に単純に貼り付けた画像をさらに合成することで, 模様本来の色味が残る合成を行う。

Poisson Image Editing によって作成された画像を  $K$ , 単純な貼り付けによって作成された画像を  $J$  とし, それらの座標  $c \in \mathbb{Z}^2$  での画素値を,  $P[c], J[c]$  のように表す。模様領域の境界では画像  $K$  の画素値に, 模様内部の領域では画像  $J$  の画素値になるような画像を得るために, 式 (4) で定義されるグラデーションマスク画像  $G$  を使用する。グラデーションマスク画像は, 二値マスク画像とは異なり模様の境界に近づくほど画素値が徐々に黒へ近づくマスク画像である。グラデーションマスク画像  $G$  の座標  $c$  での画素値  $G[c]$  は, 次式で算出する。

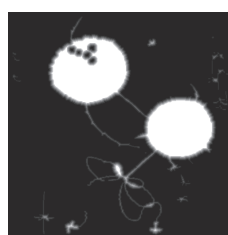
$$G[c] = \begin{cases} 255 & (\text{if } d(c) > t) \\ 255 \times \frac{d(c)}{t+1} & (\text{otherwise}) \end{cases} \quad (4)$$

ここで,  $d(c)$  は二値マスク画像上の画素  $c$  から最も近い黒い画素までのユークリッド距離である。しきい値  $t$  はグラデーションの範囲を変化させるパラメータであり,  $t$  が大きくなると  $0 \sim 255$  の中間の値をもつ領域が大きくなる。本研究では  $t = 10$  として実験を行っている。二値マスク画像とグラデーションマスク画像の例を図5に示す。このグラデーションマスク画像  $G$  を使って, 模様領域の境界では画像  $K$  の画素値に, 模様内部の領域では画像  $J$  の画素値になるような画像  $I$  を次のように作成する。

$$I[c] = J[c] \times \frac{G[c]}{255} + K[c] \times \frac{255 - G[c]}{255}$$

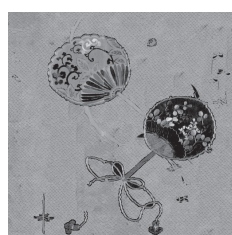


(a) 二値マスク画像

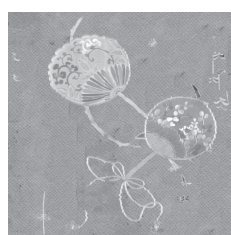


(b) グラデーションマスク画像

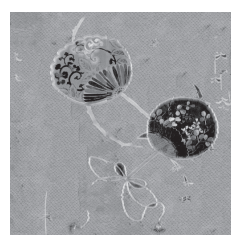
図5 二値マスク画像とグラデーションマスク画像の例



(a) 単純な貼り付けによって作成された画像 *J*



(b) Poisson Image Editing によって作成された画像 *K*



(c) グラデーションマスク画像を使った合成画像 *I*

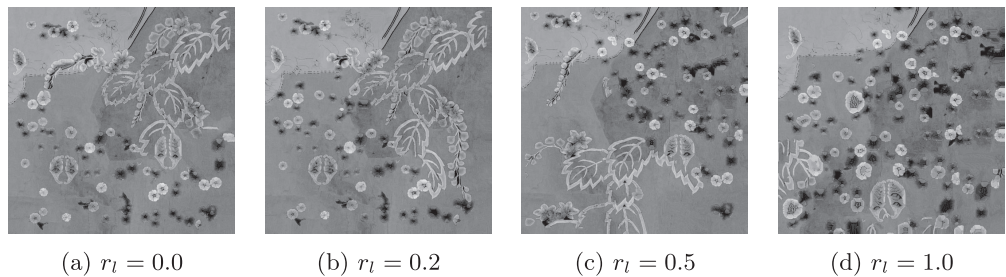
図6 模様画像と背景画像の合成結果の比較

図6 (a)に単純な貼り付けによって作成された画像 *J*, (b)に Poisson Image Editing によって作成された画像 *K*, (c)に画像 *J* と *K* をグラデーションマスク画像を使って合成した画像 *I* の例を示す。図6 (c)は、図6 (b)と比較すると、Poisson Image Editing による合成結果よりも模様本来の色味が残り、かつ境界が連続的に変化していることがわかる。このように Poisson Image Editing を行った上でさらに単純な貼り付け画像と合成することで、模様本来の色味を残した合成画像を作成することが可能となる。

### 3.3 新しい模様画像の作成 (Step III)

前ステップで作成した2枚の同背景の画像を見本画像としてテクスチャ合成を行い、2つの模様を持つ新しい模様画像 *F* を生成する。本ステップのテクスチャ合成では、2.2.2節で説明したように、画像 *D* と *E* の二値マスク画像を利用して模様領域の画素には Jitter ステップで変更を加えないという工夫を加えることで、模様の形状を保持し、模様が合成によって崩れることを防ぐ。

テクスチャ合成の Jitter ステップでのランダム性を制御するパラメータ  $r_1$  を 0.0, 0.2, 0.5, 1.0 と変更した場合の合成画像の生成例を図7に示す。パラメータ  $r_1 = 0.0$  の場合は、Jitter ステップでのランダム性がないため、模様の形状やレイアウトが崩れていない。一方、 $r_1 = 0.5$  や  $r_1 = 1.0$  では、合成した模様の形状がやや崩れてしまっているのが確認できる。これらのことから、本研究ではパラメータ  $r_1$  は低めの値に設定することとし、 $r_1 = 0.2$  として実験を行うこととした。

図7 テクスチャ合成によって新しい模様画像を合成する際のパラメータ $r_l$ の値を変化させた結果

#### 4……………提案手法の評価実験

本章では、提案手法の模様生成効率を検証するために行った評価実験について述べる。評価実験では、提案手法で新しい模様画像を作成する方が被験者が画像編集ソフトで作成するよりも時間効率が良いことの確認と、合成された画像に対する主観的評価を行うことを目的としている。

##### 4.1 実験の手順

提案手法の入力画像と同じ画像と、提案手法で生成した合成画像を基に作成した教示画像を被験者に与え、画像編集ソフト Adobe Photoshop Elements 13 を使用して、入力画像から教示画像にできるだけ近い画像を作成してもらう。入力画像数は提案手法の入力に2枚の画像を使う場合と3枚の画像を使う場合を考え、模様の細かさは粗く大きい模様と細かい詳細な模様の2通りを用意した。ただし、画像編集ソフトに対する慣れの問題を回避するために、10名2グループ、合計20名の被験者により実験を行った。

各被験者グループに対する入力画像数と模様画像の細かさの組合せを表1にまとめる。以下、表1における4通りの設定をそれぞれ設定I～IVと参照する。また、図8～11にそれぞれの設定における入力画像と提案手法で生成した合成画像を示す。入力画像と合成画像のサイズはすべて512×512であり、提案手法はDELL Precision T3500, CPU: Intel Xeon W3520 (2コア, 2.67GHz, 8MB キャッシュ), メモリ: 4GB の計算機上で実行した。図12に、図8～11中に示した提案手法で生成した合成画像を基に筆者が作成した被験者実験のための教示画像を示す。

被験者実験の手順は次の通りであり、各グループにつき2枚の合成画像を作成してもらう。

表1 被験者実験の設定と被験者グループ

設定	入力画像枚数	模様の細かさ	被験者グループ
I	2	粗い	1
II	3	細かい	
III	2	細かい	2
IV	3	粗い	

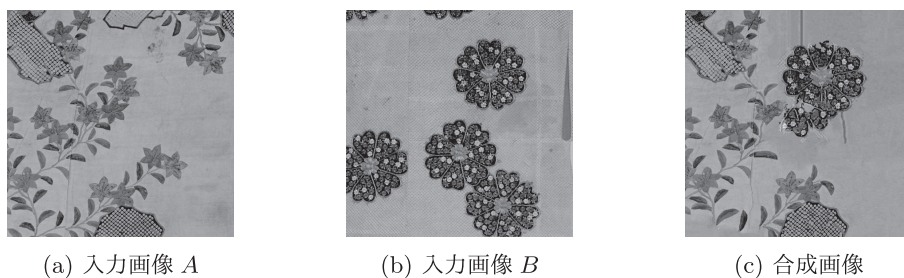


図8 設定Iの入力画像と提案手法による合成画像

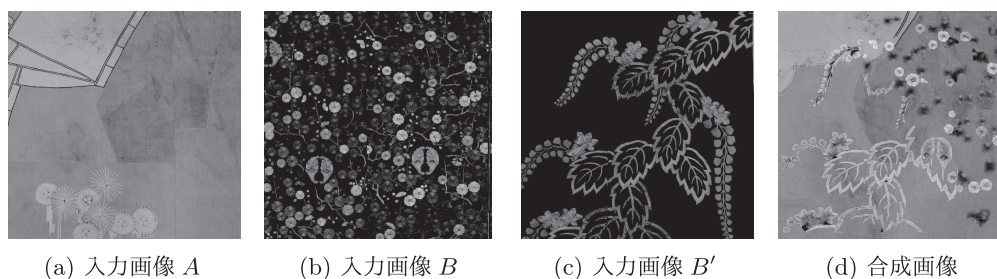


図9 設定IIの入力画像と提案手法による合成画像

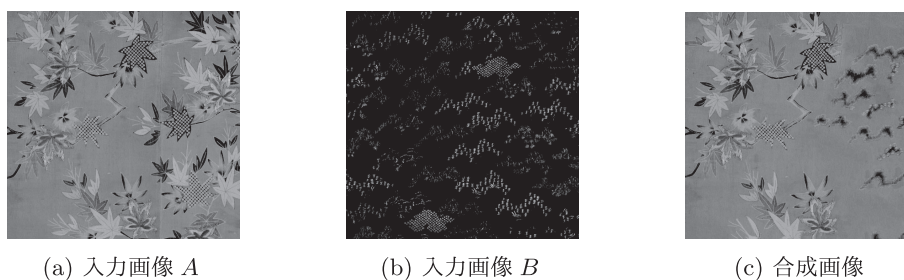


図10 設定IIIの入力画像と提案手法による合成画像

1. 被験者に入力画像から新しい画像を作成することを説明し、各作業には制限時間を設けることを事前に被験者に通知する。
2. Adobe Photoshop Elements 13の使用方法と画像の作成方法を説明書とサンプル画像を用いて6分程度で説明する。
3. 1枚目の合成画像を被験者に作成してもらう。グループ1では図8(a)~(b)の入力画像と図12(a)の教示画像を、グループ2では図10(a)~(b)の入力画像と図12(c)の教示画像を1つ目の入力画像と教示画像として被験者に与え、作業時間を計測しながら入力画像から教示画像に近い画像を作成してもらう。なお、作業中は被験者は説明書を見ることや監督者にソフトの使用法を聞くことができる。
4. 被験者が申告した段階で作業を終了し、そこまでの作業時間を記録する。ただし、被験者は作業中に残り時間を知ることができ、グループ1では10分、グループ2では30分経過した段階で作業を終了する。

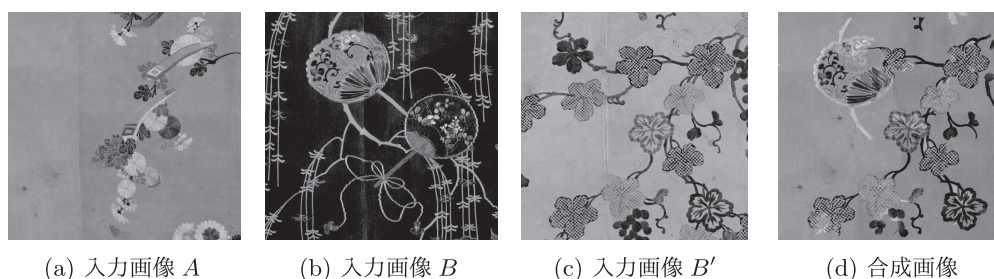


図11 設定Ⅳの入力画像と提案手法による合成画像

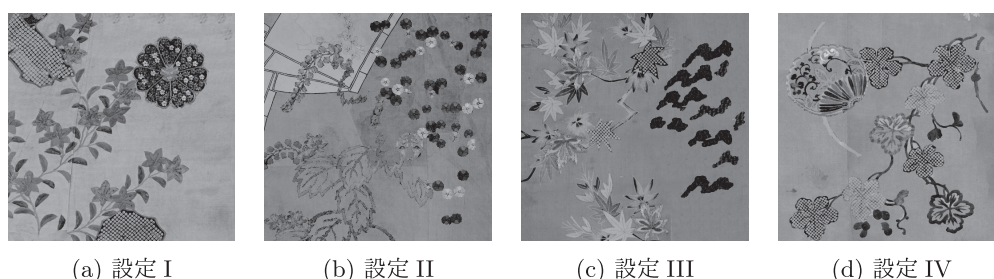


図12 提案手法による合成画像を基に作成した被験者への教示画像

表2 評価実験で用いたアンケート内容

質問番号	質問内容
Q1	一枚目の画像で自分で作成した画像と提案システムでの画像ではどちらが良いと感じますか
Q2	二枚目の画像で自分で作成した画像と提案システムでの画像ではどちらが良いと感じますか
Q3	画像の編集を行ったことはありますか
Q4	この実験を行った感想を記入してください

- 2枚目の合成画像を被験者に作成してもらう。グループ1では図9(a)～(c)の入力画像と図12(b)の教示画像を、グループ2では図11(a)～(c)の入力画像と図12(d)の教示画像を2つ目の入力画像と教示画像として被験者に与え、作業時間を計測しながら入力画像から教示画像に近い画像を作成してもらう。なお、1枚目の作業と同様に作業中は被験者は説明書を見ることや監督者にソフトの使用方法を聞くことができる。
- 被験者が完成と申告した段階で作業を終了し、そこまでの作業時間を記録する。ただし、被験者は作業中に残り時間を知ることができ、グループ1では40分、グループ2では30分経過した段階で作業を終了する。
- 提案手法で合成した画像と被験者が作成した模様画像を並べて表示し、表2に示したアンケートに回答してもらう。

## 4.2 実験結果と考察

ここでは、画像作成時間による定量的な評価と、アンケートによるユーザの主観的評価に基づき実験結果を考察する。なお、グループ1と2のどちらにおいても、9名の被験者は画像編集を行ったことがない初心者であり、1名は画像編集を行ったことのある経験者であった。

表3に、提案手法を用いて合成画像を作成するのに要した実計算時間と被験者が作成に要した平均時間、最長時間、最短時間を示す。被験者の最短作成時間と提案手法の計算時間を比べると、設定I以外で提案手法の方が短時間で合成画像を作成することができている。また制限時間内に合成画像の作成を完了できなかった被験者は、設定Iで1名、IIで2名、IIIで0名、IVで1名であった。

表3より、入力画像数がともに2枚である設定IとIIIでは、提案手法は粗い模様の方に多くの時間を要していることがわかる。入力画像数が3枚である設定IIとIVの場合でも同じ傾向がみられる。その理由としては、Poisson Image Editingの処理時間が粗く大きな模様に関与するほど長くなるためであると考えられる。それに対し、いずれの設定でも、被験者は入力画像数が同じであれば模様が粗い場合に早く画像を作成することができている。これは、模様が粗い場合、画像編集ソフトでの切り取り範囲の選択と貼り付けが初心者でも容易なため、被験者の作成時間が短くなったと考えられる。

一方、異なる入力枚数で模様の細かさが同じである設定IIとIIIの場合、提案手法は、入力枚数に関係なく同程度の時間で合成ができているのに対し、被験者は入力画像数が3枚の場合に多くの時間を要しており、設定IとIVでも同様の傾向が見られる。このことから、人手では入力画像枚数の増加に応じて作業時間が大きく増加する一方、提案手法では画像の枚数に関わらずある程度一定した時間で画像合成を行うことが可能であることがわかる。

アンケートの回答から、提案手法が生成した合成画像よりも自身が作成した合成画像の方が良いと回答した被験者数は、設定Iで10名、設定IIで3名、設定IIIで2名、設定IVで6名であった。合成画像の良し悪しの評価は主観的なアンケートではあるが、実験II～IVにおいては、画像編集ソフトを使って人手で画像を合成した場合に相当する程度のクオリティが提案手法でも得られていると示唆される。

被験者の実験の感想として「画像作成に時間がかかって大変であった」、「画像編集ソフトを使いこなすのが困難であった」というものがあった。画像編集ソフトを使用して新しい模様画像を作成するのは、初心者だけでなく経験者であっても入力画像の枚数が多かったり模様が細かったりすると時間がかかってしまう。それに対して、提案手法は模様の種類に関わらず入力画像を選択するだ

表3 画像作成時間の比較 (分:秒)

設定	I	II	III	IV
グループ	1		2	
提案手法	07:12	05:14	05:30	08:10
被験者平均	05:17	35:26	19:47	17:19
被験者最長	10:00	40:00	29:50	30:00
被験者最短	03:43	20:08	10:34	08:19

けで全自動で新しい模様画像を生成できることが利点である。そのため提案手法は、生成したい模様の組合せをたくさん試したり、試行錯誤しながらデザインのレイアウトを決めるといった用途にも応用の可能性があると考えられる。

## ⑤……………おわりに

本研究では小袖屏風画像中の模様を組合せ、新しい模様画像を自動生成する手法を提案し、被験者実験を通して提案手法の時間的効率性を確認した。評価実験から、提案手法は被験者と比べて、入力画像の枚数に依らず安定して短い時間で画像を合成することが可能であることが示せた。提案手法は、気になる模様の組合せを試す場合や、試行錯誤しながらデザインの方向性を決めるといった用途にも応用の可能性がある。

今後の課題としては、模様が崩れてしまうことを防止するために良質なマスク画像を作成することや、GPUで提案手法を実装することで模様画像の合成に必要な時間を短縮することなどが挙げられる。本研究では模様本来の色味を保持するために Poisson Image Editing の結果をグラデーションマスクにより補正した。この効果を評価するために、色味の再現が主観評価に与える影響についても詳細に調査する必要がある。さらに、合成画像を実際の小袖の形に生成することで、新しい模様の組合せを持つ小袖屏風を作成し、ユーザーに提示することも検討していきたい。

### 註

(1)——ただし、mod や床関数はすべて座標毎に計算されることとする。

### 参考文献

- [1] 廣瀬通孝：デジタルミュージアム，情報処理，Vol. 52, No. 4, pp. 492-497 (2011).
- [2] 国立歴史民俗博物館編：国立歴史民俗博物館資料図録2 野村コレクション小袖屏風 (2002).
- [3] P. Peréz, M. Gangnet, and A. Blake: Poisson Image Editing, *ACM SIGGRAPH '03*, pp. 313-318 (2003).
- [4] 山崎俊彦：勾配ベースの画像編集：Poisson Image Editing, 映像情報メディア学会誌，Vol. 64, No. 5, pp. 726-737 (2010).
- [5] 高山健志，五十嵐健夫：テクスチャ合成技術の新たな応用と展開，情報処理，Vol. 53, No. 6, pp. 582-588 (2012).
- [6] L. Wei, S. Lefebvre, V. Kwatra, and G. Turk: State of the Art in Example-based Texture Synthesis, *Eurographics '09 State of the Art Reports (STARs)* (2009).
- [7] L. Itti, C. Koch, and E. Niebur: A Model of Saliency-based Visual Attention for Rapid Scene Analysis, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 11, pp. 1254-1259 (1998).
- [8] L. Wei and M. Levoy: Fast Texture Synthesis Using Tree-structured Vector Quantization, *SIGGRAPH '00*, pp. 479-488 (2000).
- [9] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick: Graphcut Textures: Image and Video Synthesis Using Graph Cuts, *SIGGRAPH '03*, pp. 277-286 (2003).
- [10] V. Kwatra, I. Essa, A. Bobick, and N. Kwatra: Texture Optimization for Example-based Synthesis, *ACM Transactions on Graphics (TOG) - Proceedings of ACM SIG-GRAPH '05*, Vol. 24, No. 3, pp. 795-802 (2005).
- [11] S. Lefebvre and H. Hoppe: Parallel Controllable Texture Synthesis, *ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH '05*, Vol. 24, No. 3, pp. 777-786 (2005).
- [12] E. Rissler, C. Han, R. Dahyot, and E. Grinspun: Synthesizing Structured Image Hybrids, *ACM Transactions on*

- 
- Graphics (TOG) - Proceedings of ACM SIGGRAPH '10*, Vol. 29, No. 4, pp. 85:1-85:6 (2010).
- [13] H. Park, H. Byun, and C. Kim: Multi-exemplar inhomogeneous texture synthesis, *Computers & Graphics, Vol. 37*, No. 1-2, pp. 54-64 (2013).
- [14] X. Tong, J. Zhang, L. Liu, X. Wang, B. Guo, and H. Shum: Synthesis of Bidirectional Texture Functions on Arbitrary Surfaces, *ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH '02*, Vol. 21, No. 3, pp. 665-672 (2002).
- [15] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman: PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing, *ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH '09*, Vol. 28, No. 3 (2009).
- [16] M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S. Hu: Salient Object Detection and Segmentation, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 37, No. 3, pp. 569-582 (2015).
- [17] P. Felzenszwalb and D. Huttenlocher: Efficient graph-based image segmentation, *International Journal of Computer Vision*, Vol. 59, No. 2, pp. 167-181 (2004).

白川真一（横浜国立大学大学院環境情報研究院，国立歴史民俗博物館共同研究員）

井田有香（青山学院大学理工学研究科理工学専攻，国立歴史民俗博物館共同研究研究協力者）

大原剛三（青山学院大学理工学部，国立歴史民俗博物館共同研究研究協力者）

豊田哲也（青山学院大学理工学部，国立歴史民俗博物館共同研究研究協力者）

（2019年3月14日受付，2019年8月5日審査終了）



---

## Pattern Image Synthesis Using Kosode Byobu Images

SHIRAKAWA Shinichi, IDA Yuuka, OHARA Kouzou and TOYOTA Tetsuya

Recently, the digital archive technology of historical materials enables us to use precious historical materials on computers easily. With this situation, advanced studies such as the highly-intelligent applications using historical materials and unique exhibits with the latest technologies, such as computer graphics and virtual reality, are actively conducted. This research focuses on the digital data of Kosode Byobu. Kosode Byobu is a historical material in which the real Kosode (Japanese clothes) is pasted up on Byobu. The various patterns in Kosode are one of the reasons for visitors' interests. Therefore, if we realize a method for easily generating the user's original pattern images based on the existing patterns in Kosode Byobu, it would lead to a new Kosode Byobu display system using digital data. Because of this situation, this research attempts to construct a method that automatically generates a new pattern image from existing ones in Kosode Byobu selected by a user. Since our target users are the public visitors to the museum, a complicated operation, such as image editing, should not be included in the pattern generation procedure. In this research, based on image editing and texture synthesis technologies, we develop a method that automatically generates a new pattern image with pattern image features in Kosode Byobu as selected by a user. In the proposed method, the user is only required to select several images to be synthesized. The proposed method consists of following three steps: (1) make the background fabric image, (2) paste the patterns into the background fabric image using Poisson image editing, and (3) generate the synthetic pattern image by texture synthesis. Furthermore, we conduct an experiment in which the subjects create a similar pattern image with the one generated by the proposed method using image-editing software. The experimental result showed that the proposed method could generate the pattern images in a shorter time without depending on the number of input images, compared to the case in which the subjects manually create the synthetic image.

Key words: Kosode Byobu Image, Texture Synthesis, Image Blending