

リジッドルームのための織物組織図の近似最適化

吉田 哲也^{1,a)}

受付日 2021年1月28日, 再受付日 2021年3月9日,
採録日 2021年3月18日

概要: 経糸（たて糸）と緯糸（よこ糸）を絡み合わせて制作する布は織物と呼ばれ、糸の交差は織物組織図として表現される。綜統の数が少ない卓上織機であるリジッドルームを用いて複雑な模様を制作するためには緯糸ごとに経糸を手作業で拾う操作が必要となるが、拾う糸の数が規則的でない場合は間違いやすく負担が大きいという課題がある。リジッドルームで複雑な模様を制作する際の負担の軽減に向けて、本稿では経糸の拾い方の割当てに基づく織物組織図の近似を提案する。織物の模様における組織図の敷き詰めと画像特徴量に基づいて組織図の差異を表現し、経糸の拾い方の逐次的な再割当てを行う組織図生成アルゴリズムを提案する。提案法を実装し、アルゴリズムの評価と織物の制作を通じて有効性を検証した。

キーワード: リジッドルーム, 手織り, 織物組織図, 近似

Approximate Optimization of Weave Diagram for Rigid Loom

TETSUYA YOSHIDA^{1,a)}

Received: January 28, 2021, Revised: March 9, 2021,
Accepted: March 18, 2021

Abstract: A textile is woven using warp and weft, and their relationship is represented as a weave diagram. Since the number of healds is small in a rigid loom, it is necessary to pick up warp for each weft by hand. This operation is rather cumbersome when the number of warp to be picked up is irregular. This paper proposes a method for approximating the specified weave diagram based on the assignment of warp pick up pattern. By representing the difference of weave diagrams based on their tiling and image features, a sequential re-assignment algorithm is proposed to generate an approximated weave diagram. The proposed approach is validated in terms of the performance of the algorithm and several woven fabric.

Keywords: rigid loom, hand weaving, weave diagram, approximation

1. はじめに

経糸（たて糸）と緯糸（よこ糸）を絡み合わせて作られた布は織物と呼ばれ、布を織るための装置は織機と呼ばれる [9], [11]。製品として販売される織物は自動織機で製造されることが多いが、工房での制作や家庭で手芸を楽しむための手織り機も普及している。卓上織機のうちリジッドルームは比較的安価で家庭でも使いやすいが、綜統の数が少ないため制作可能な模様の種類に限られる。リジッドルームを用いて複雑な模様を制作することもできるが、緯

糸ごとに経糸を手作業で拾う操作が必要となる [10]。しかし、拾う糸の数が規則的でない場合は間違いやすく負担が大きいという課題がある。

リジッドルームを用いて複雑な模様を制作する際の負担の軽減に向けて、本稿では経糸の拾い方の割当てに基づく織物組織図の近似を提案する。織物の模様における組織図の敷き詰めと画像特徴量に基づいて組織図の差異を表現し、経糸の拾い方の逐次的な再割当てを行う組織図生成アルゴリズムを提案する。本稿のアプローチを実装し、アルゴリズムの評価と織物の制作を通じて有効性を検証した。

2 章で織りと織物組織図を紹介し、3 章でリジッドルームのための織物組織図の近似を提案する。4 章で実装と検証を述べ、5 章でまとめと今後の展望を述べる。

¹ 奈良女子大学
Nara Women's University, Nara 630-8506, Japan
^{a)} tyoshida@cc.nara-wu.ac.jp

2. 織物の制作

2.1 織り

織物は経糸に緯糸を通すことで作られ、基本の織り方である平織、綾織、縹子織は三原組織と呼ばれる [9], [11]. 平織の例を図 1, 綾織の例を図 2 に示す. 織りの構造を表す構造図で経糸と緯糸の交差点ごとにどちらの糸が上かを指定する必要があるため、織物の組織は織物組織図 (以下、組織図と呼ぶ) として表される. 意匠紙のマス (方眼) の一目を交差点とし、経糸が上に現れる点を黒、その逆を白として表すことが多く、黒い点は組織点と呼ばれる. 組織図には一般に規則的な繰り返しがあり、繰り返しの最小単位は完全組織と呼ばれる. 以下では、組織図は織物の完全組織に対応することとする.

布を織る装置は織機と呼ばれ、製品として販売される織物を製造する織機にはタベット織機、ドビー織機、ジャガード織機などがあるが、本稿では工房や手芸などでの手織りに用いられる織機を扱う. 図 3 にその模式図を示す. 経糸を通す綜統を装着した綜統枠が踏み木とつながり、踏み木を踏むと綜統の上下運動で経糸が 2 層に分離して開口部が形成され、その間に緯糸を通して織物を制作する.

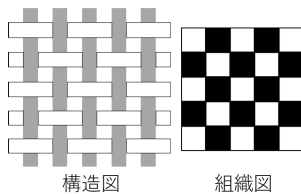


図 1 平織

Fig. 1 Plain weave.

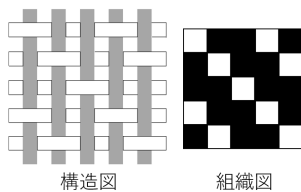


図 2 綾織

Fig. 2 Twill weave.

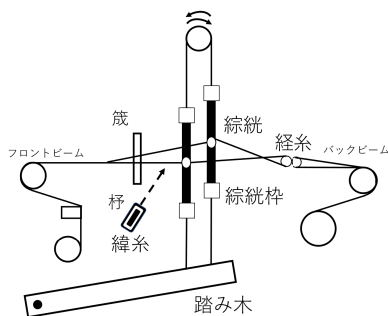


図 3 手織り織機の模式図

Fig. 3 Loom for hand weaving.

図 3 に示すように、織物の制作では 1) 踏み木の踏み方, 2) 綜統枠と踏み木のつなげ方, 3) 綜統への経糸の通し方, を指定する必要がある. これらを表す図の呼び方は文献によっても異なるが、本稿ではそれぞれ踏み方図, タイアップ図, 綜統図と呼ぶ. また、組織図, 踏み方図, タイアップ図, 綜統図を図 4 のように配置して表すものを織方図と呼ぶ [2].

2.2 織物組織図の行列表現

組織図の交差ごとに、経糸が上である組織点を 1 で表し、それ以外を 0 で表すことにより、組織図を行列を用いて表現できる [2]. 同様に、織方図の他の図でもマス目ごとに黒と白が指定されるため、組織図と同様に黒の箇所を 1、それ以外を 0 とすることで行列として表現できる.

以下では、行列は太字の大文字、ベクトルは太字のイタリック小文字で表し、行列 $\mathbf{A}$ の第 ij 要素を a_{ij} 、ベクトル $\mathbf{a}$ の第 i 要素を a_i で表す. 行列 $\mathbf{A}$ の転置を $\mathbf{A}^T$ 、ベクトル $\mathbf{a}$ の転置を $\mathbf{a}^T$ で表し、 $\mathbf{A}$ が正則行列のときはその逆行列を $\mathbf{A}^{-1}$ で表す. また、行列 $\mathbf{A}$ の第 i 行ベクトルを $\mathbf{a}_{i+}$ 、第 j 列ベクトルを $\mathbf{a}_{+j}$ で表す.

本稿では、組織図を行列 $\mathbf{W} \in \{0, 1\}^{m \times n}$ 、踏み方図を行列 $\mathbf{A} \in \{0, 1\}^{m \times p}$ 、タイアップ図を行列 $\mathbf{B} \in \{0, 1\}^{h \times p}$ 、綜統図を行列 $\mathbf{C} \in \{0, 1\}^{h \times n}$ と表す. このとき、組織図 $\mathbf{W}$ は行列演算を用いて

$$\mathbf{W} = \mathbf{A}\mathbf{B}^T\mathbf{C} \quad (1)$$

と表すことができる [2]. ただし、経糸 1 本がそれぞれ 1 枚の綜統に通る、同時に踏める踏み木は 1 本であるため、 $\mathbf{A}$ の行ごと、 $\mathbf{C}$ の列ごとに、それぞれただ 1 つの 1 を持つという制約を満たす必要がある.

2.3 関連研究

織物の手織りに対する支援として、カードを用いた手織りに対するインタラクティブなデザイン支援と制作支援を行う Weavy [5] がある. このシステムはデザインした織りを表示するとともに、制作時にカードに経糸を通すことを支援する. また、展示やワークショップでの検証も報告さ

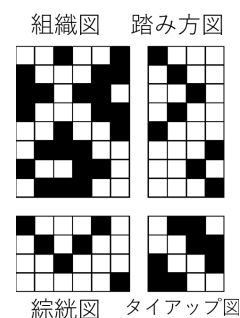


図 4 織方図

Fig. 4 Weaving draft.

れている。

行列に基づくアプローチとしては、ドビー織機に対して 1 本の経糸を複数の綜統に通せる長目綜統を導入した際の綜統数最小化問題の定式化と近似アルゴリズムの提案がある [6], [7]. 組織図などを 0 と 1 を要素とする行列（文献 [6], [7] ではブール行列と呼ばれる）で表現し、綜統数最小化問題をブール行列の階数を求める問題として定式化するとともに、グラフの彩色問題や集合被覆問題への還元に基づくアルゴリズムが示されている。

画像処理に基づくアプローチとしては、任意の交差点で経糸が上にくることを指定できるジャガード織機を対象として、縐子織に対するジャガード組織図の生成する手法がある [14]. この手法では組織図を二値画像と見なし、一定範囲内の間隔で経糸と緯糸を上下させるという制約を考慮した組織的ディザ法を用いて組織図を生成する。

異なる組織図を繰り返しても同じ模様の織物を制作できる場合があり、それらは等価組織と呼ばれる。同じ模様を生成する等価組織を除外した組織図の数え上げを扱う研究もある [8].

3. 織物組織図の近似学習

3.1 リジッドルームを用いた手織り制作

リジッドルームは比較的安価で家庭でも使いやすいが、綜統の数が少ないため制作可能な模様の種類が限られる。たとえばクロバー社の「咲きおり」(図 5 参照) [16] は綜統数が 2 枚の織機に対応するため、経糸と緯糸が 1 本ずつ交互に上となる平織りしか制作できない。リジッドルームを用いて複雑な模様の織物を制作することもできるが、綜統を用いて経糸を開口するのではなく、ピックアップスティックを用いて手作業で緯糸ごとに経糸を拾うことが必要となる (図 6 参照)。しかし、拾う糸の数が規則的でない場合

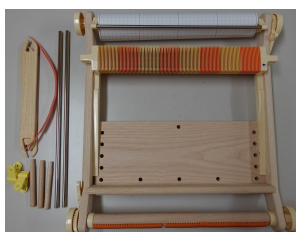


図 5 リジッドルーム「咲きおり」
Fig. 5 Loom “Sakiori”.

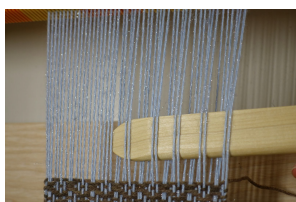


図 6 経糸を拾う操作
Fig. 6 Warp pick up.

は間違いが生じやすいという課題がある。そこで、リジッドルームを用いて複雑な模様の織物を制作する際の負担の軽減に向けて、経糸の拾い方も考慮した組織図の近似を考える。

3.2 組織図の近似モデル

3.2.1 織りのデザインの枠組み

本稿の枠組みを図 7 に示す。まず、織物の模様をデザインする。その際、経糸と緯糸に対して異なる色を割り当て、経糸と緯糸の交差点ごとに織物で上に浮く色を指定する。経糸と緯糸に使用した色の情報をもとに、デザインした模様を組織図 $\mathbf{W}_0$ に変換し、3.3 節で述べる経糸のすくい方のバリエーションに基づいて組織図 $\mathbf{W}$ を生成する。生成した組織図に対応する模様とその織方図を確認し、デザインした模様と比較する。デザイン上の必要性があれば模様を修正し、このプロセスを繰り返す。

3.2.2 組織図の差異の表現

組織図を 0 と 1 を要素とする行列として表現するため、 $\mathbf{W}_0$ と $\mathbf{W}$ の差異 $\Delta\mathbf{W}$ を以下で定義する。

$$\Delta\mathbf{W} = (\mathbf{W}_0 + \mathbf{W}) \bmod 2 \quad (2)$$

ただし、式 (2) で和 $+$ と $\bmod$ の演算は要素ごとに行う。式 (2) の $\Delta\mathbf{W}$ は、 $\mathbf{W}_0$ と $\mathbf{W}$ で経糸が上であるかが異なる交差点を 1、一致する交差点を 0 と表す。

式 (2) の $\Delta\mathbf{W}$ は組織図の差異を表すが、織物の模様は組織図を繰り返したものであるため、 $\mathbf{W}_0$ と $\mathbf{W}$ の違いだけではなく、組織図を規則的に繰り返して敷き詰めた模様としての差異を考慮する必要がある。組織図の敷き詰めはタイリングと見なすことができるため、1 つの組織図を取り囲む範囲だけを考慮すれば十分である [13]. そこで、 $\Delta\mathbf{W}$ を上下左右に 3 つずつ敷き詰めた分割行列 $\Delta\mathbf{W}_t$ を、要素がすべて 1 である 3×3 行列 $\mathbf{H}$ とのクロネッカー積 $\otimes$ を

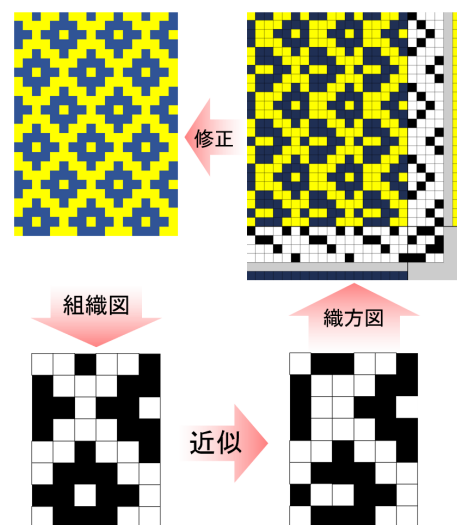


図 7 織りのデザインの枠組み
Fig. 7 Weave design framework.

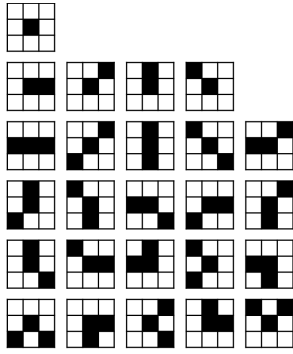


図 8 HLAC のマスクパターン
Fig. 8 HLAC mask patterns.

用いて以下で生成する.

$$\Delta W_t = H \otimes \Delta W \quad (3)$$

$$= \begin{pmatrix} \Delta W & \Delta W & \Delta W \\ \Delta W & \Delta W & \Delta W \\ \Delta W & \Delta W & \Delta W \end{pmatrix} \quad (4)$$

式 (4) の ΔW_t は繰り返しも考慮した組織図の差異を表すが、織物の模様は 2 次元の拡がりを持つため、交差点に対応する個々の要素だけではなく、要素ごとにその近傍の情報も考慮して模様としての差異を表すことが重要となる。その際、等価組織と同様に平行移動に不変な表現であることが望ましい。

本稿では、組織図を 2 値画像と見なし、画像における近傍情報を考慮できる高次局所自己相関特徴 [12] (以下では HLAC と呼ぶ) を用いて組織図の差異を表現する。HLAC は画像の各要素 (ピクセル) に対するマスクパターン (図 8 参照) に基づく特徴量であり、マスクパターンの形状から要素ごとにその近傍の情報を活用できる。また、位置不変性や部分画像に対する加法性が成り立つことも知られている。

近似精度の向上に向けて局所的な勾配に基づく画像特徴量を活用することも考えられるが [15], 一般画像認識などで扱う画像と比べて組織図のサイズは小さいことが多いため、マスクパターンのサイズが 3×3 と小さく簡便な HLAC を用いている。他の画像特徴量を用いて組織図の差異を表現することは今後の課題である。

マスクパターンとマッチングする際、組織図の周辺部にある要素でも 8 近傍の情報が得られるようにする必要がある。そこで、 ΔW_t の中央にある ΔW の要素ごとに、HLAC で定義される 3×3 のサイズの 25 個のマスクパターン (図 8 参照) とのマッチングを行い、パターンの生起頻度に基づいて組織図の差異を表現する (図 9 参照)。図 9 で赤の点線で囲む範囲の要素に図 8 のマスクパターンを適用し、組織図の差異を行列として表現する。

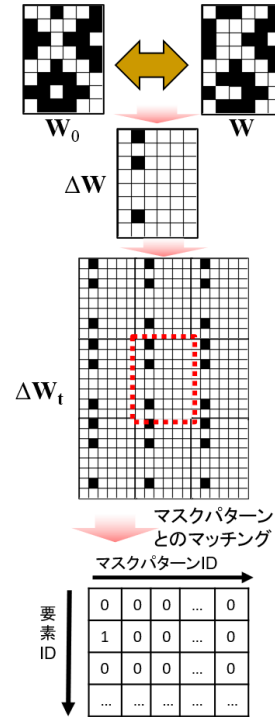


図 9 組織図の差異の表現

Fig. 9 Representation of weave diagram difference.

3.3 リジッドルームのための組織図の近似

3.3.1 経糸の拾い方の割当てを通じた組織図の生成

組織図は式 (1) のように行列演算を用いて表現できるが、これを以下のように書き換える。

$$\begin{aligned} W &= AB^T C \\ &= A(B^T C) \end{aligned} \quad (5)$$

式 (5) は、タイアップ図 B と綜統図 C を用いて行列 $(B^T C)$ を求め、2.2 節で述べた踏み方図 A における制約に基づき、組織図の各行ごとに $(B^T C)$ の行ベクトルを割り当てることを表す、と見なすことができる。

しかし、リジッドルームには踏み木はなく、また、複雑な模様を制作する際には綜統も用いない。そこで、式 (5) の考え方を一歩進めて、リジッドルームでの制作で用いる経糸の拾い方のバリエーションを考え、それを行列 $S \in \{0, 1\}^{s \times n}$ として表現し、組織図の各行に対して S における経糸の拾い方を割り当てることで組織図を生成することを考える。行列 S の行番号を経糸の拾い方の ID と見なし、組織図の各行への経糸の拾い方の割当てを並べたベクトルを $a \in N^m$ とすると、割当てに従って S の行ベクトルをコピーすることで組織図 W を生成できる。この操作を

$$W \leftarrow weave(a, S) \quad (6)$$

と表す。

行列 S の各行は経糸の拾い方に対応する 0 と 1 を並べたものとなるが、その並び方が不規則な場合には経糸の拾い間違いが生じやすくなる恐れがある。他方、規則的な並

```

Require:  $\mathbf{W}_0$ ; //組織図
Require:  $\mathbf{S}$ ; //経糸の拾い方

1: initialize  $\mathbf{a}$     //{1, ..., s} から復元抽出
2: while not yet terminated do
3:   for each  $i \in \{1, \dots, m\}$  do
4:     式 (8) で  $a_i$  への割当てを更新
5:   end for
6: end while
7: return  $\text{weave}(\mathbf{a}, \mathbf{S})$ 

```

図 10 経糸の拾い方の割当て

Fig. 10 Warp pick up assignment.

び方の場合には拾い間違いがおきにくいと期待される。どのような経糸の拾い方が望ましいかは制作者に依存するが、たとえばランレングス符号化のように同じ値の連続 (11 など) を単位と考え、これを交互に反転させた拾い方 (1100110011... など) は経糸の拾い方が規則的で間違いににくいと考えられる。

本稿では、指定された経糸の拾い方の集合のもとで組織図を近似する経糸の拾い方の割当てを求めることを提案する。3.2.2 項で述べた組織図の差異の表現に対し、生起頻度の総和を用いて近似誤差を表す目的関数を以下で定義する。

$$J = \text{diff}_{\text{HLAC}}(\mathbf{W}_0, \mathbf{W}) \quad (7)$$

このため、式 (6) を用いて組織図 $\mathbf{W}$ を生成する際に式 (7) を最小化する割当て $\mathbf{a}$ を求める。

3.3.2 アルゴリズム

組織図 $\mathbf{W} \in \{0, 1\}^{m \times n}$ の各行をデータ、 $\mathbf{S} \in \{0, 1\}^{s \times n}$ の各行をクラスタと見なし、組織図の m 個の行を s 個の経糸の拾い方のどれかに逐次的に再割当てすることを通じて式 (7) の最小化を実現する。具体的には、組織図の各行 $i \in \{1, \dots, m\}$ に対して

$$a_i = \arg \min_{q \in \{1, \dots, s\}} \text{diff}_{\text{HLAC}}(\mathbf{W}_0, \mathbf{W}) \quad (8)$$

$$s.t. \quad a'_j = \begin{cases} q & j = i \\ a_j & \text{otherwise} \end{cases}$$

と式 (7) を最小化する経糸の拾い方を選ぶことで割当て $\mathbf{a}$ を更新する。提案するアルゴリズムを図 10 に示す。1 行目で割当て $\mathbf{a}$ を初期化し、4 行目で式 (8) に基づく再割当てを行う。2 行目の終了条件は $\mathbf{a}$ の変更がなくなった場合とする。なお、kmeans 法 [3] などのクラスタリング手法と異なり、すべてのクラスタ (経糸の拾い方) に必ず 1 つ以上のデータ (組織図の行番号) が割り当てられるとは限らず、生成する組織図で行列 $\mathbf{S}$ として表す経糸の拾い方がすべて使用されるとは限らない。

式 (8) に示す割当てを逐次的に繰り返すことで、EM 法 [1] や kmeans 法と同様に目的関数の単調減少性を保証できる。

さらに、HLAC の 3×3 のマスクパターンの形状から、式 (8) に示す割当ての影響は図 10 の 4 行目での更新対象に対応する行とその前後の行を含めた範囲に限定できる。前後の行におけるパターンの生起頻度にも再割当ての影響が及ぶため、動的計画法のように図 10 の 4 行目で着目する行とそれ以外の部分とに問題を完全に分割できるわけではないが、HLAC の性質と生起頻度の和に基づく $\text{diff}_{\text{HLAC}}(\cdot, \cdot)$ の加法性により目的関数の大域的最小化が実現できると予想される。しかし、アルゴリズムの性質を厳密に示すことは今後の課題である。

組織図 $\mathbf{W} \in \{0, 1\}^{m \times n}$ では各要素が 0 と 1 の 2 値を取り得るため、等価組織などを考慮せず素朴に扱えば $O(2^{mn})$ と指数的な組合せが考えられる。本稿のアプローチでは経糸の拾い方を表す行列 $\mathbf{S} \in \{0, 1\}^{s \times n}$ における各行の織り方に限定するものの、やはり $O(s^m)$ と指数的な組合せとなる。しかし、上記で述べたように HLAC のマスクパターンとのマッチングは更新対象の行とその前後の行だけに局所化できる。パターン数は定数であるため、式 (8) の $\text{diff}_{\text{HLAC}}(\cdot, \cdot)$ は組織図の行数 m 、列数 n に対して $O(n)$ で計算でき、これを $\ell \in \{1, \dots, s\}$ の s 回行う。このため、入力 ($\mathbf{W}_0$ と $\mathbf{S}$) と初期値に依存する while の繰り返し数を無視すれば、図 10 のアルゴリズムの計算量は $O(mns)$ と多項式のオーダーとなる。

3.4 組織図に対する織方図の作成

織りは織方図として表されることが多いため、生成した組織図に対して織方図を作成する必要がある。一般に行列分解は一意ではないが、文献 [9], [11] などでの手順を 2.2 節と同様に行列を用いて示す。与えられた組織図 $\mathbf{W}$ に対し、織方図の踏み方図 $\mathbf{A}$ 、タイアップ図 $\mathbf{B}$ 、綜統図 $\mathbf{C}$ を以下のように定める。

まず、 $\mathbf{W}$ で相異なる行ベクトルを列挙し、それらを $\{\mathbf{r}_1^T, \dots, \mathbf{r}_p^T\}$ とする。次に、要素がすべて 0 の $m \times p$ 行列 $\mathbf{A}$ を準備する。最後に、 $\mathbf{W}$ の各行ベクトル $\mathbf{w}_{i+}$ ごとに、列挙した $\{\mathbf{r}_1^T, \dots, \mathbf{r}_p^T\}$ と一致するベクトルの番号を k とするとき、 $\mathbf{A}_{ik} = 1$ とする。

同様に、 $\mathbf{W}$ で相異なる列ベクトルを列挙し、それらを $\{\mathbf{c}_1, \dots, \mathbf{c}_h\}$ とする。次に、要素がすべて 0 の $h \times n$ 行列 $\mathbf{C}$ を準備する。最後に、 $\mathbf{W}$ の各列ベクトル $\mathbf{w}_{+j}$ ごとに、列挙した $\{\mathbf{c}_1, \dots, \mathbf{c}_h\}$ と一致するベクトルの番号を ℓ とするとき、 $\mathbf{C}_{\ell j} = 1$ とする。

最後に、文献 [9], [11] などでの手順とは異なるが、行列 $\mathbf{B}$ を行列演算を用いて以下で定める。

$$\mathbf{B} = \{(\mathbf{A}^T \mathbf{A})^{-1}(\mathbf{A}^T \mathbf{W} \mathbf{C}^T)(\mathbf{C}^T \mathbf{C})^{-1}\}^T \quad (9)$$

式 (9) の証明は付録 A.1 に示す。

4. 実装と検証

4.1 実装

4.1.1 織りのデザインシステム

図 7 に示した枠組みを以下のように実装した。

織物の模様デザインのデザインはスプレッドシートを用いて行い、まず経糸と緯糸の色を決め、次に交差点に対応するセルごとに経糸と緯糸のどちらかの色を設定する。デザインした模様の交差点の色をスプレッドシートの機能を用いてそれぞれ数値に変換し、ファイルに保存する。

3 章で述べた手法は R 言語で実装した。模様の変換点を数値として表現するファイルと色の数値情報に基づいて組織図 $\mathbf{W}_0$ を生成し、指定した $\mathbf{S}$ のもとで図 10 のアルゴリズムで組織図 $\mathbf{W}$ を生成する。また、3.4 節の手順で $\mathbf{W}$ に対する踏み方図 $\mathbf{A}$ 、タイアップ図 $\mathbf{B}$ 、綜統図 $\mathbf{C}$ も生成し、生成した行列をそれぞれ保存する。

織方図を提示するインタフェースは **Processing** 言語で実装した [4]。織方図に対する行列をそれぞれ読み込み、指定した経糸と緯糸の色に従って模様を提示する。実装した画面例を図 11 に示す。図 11 では、ネイビーの経糸と黄色の緯糸を用いた際の織方図を示す。さらに、図 11 の画面上で白と黒の行列として提示される踏み方図 **A**、タイアップ図 **B**、綜統図 **C** のセルをクリックして反転することで、式 (1) に示した行列演算を通じてインタラクティブに模様を変更できる。図 11 に示す織方図をインタフェース上で操作し変更した例を図 12 に示す。

4.1.2 経糸の拾い方の生成

本稿のアプローチでは経糸の拾い方を行列 $\mathbf{S}$ として与え

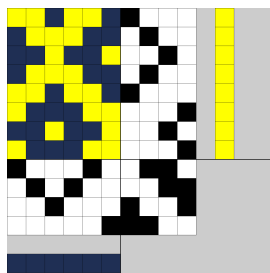


図 11 織方図の画面例

Fig. 11 Interface for weaving draft.

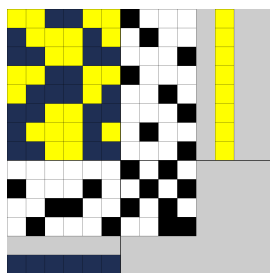


図 12 模様の変更例

Fig. 12 Example of modified weave.

必要がある。負担の軽減につながる経糸の拾い方は制作者に依存するが、本稿では連続する数字（11 など）を単位と考え、交互に反転を繰り返す規則的な拾い方を生成して用いることとした。それぞれの拾い方は 0 と 1 で表される文字列となるため、連続する数字の最大長 r を指定して文字列を生成した。また、制作経験者の意見に基づき、最初だけは r 未満の連続も許すこととした。上記の手順により、最大長 r に対し $r(r+1)$ 個の文字列が生成される。

最大長 r を 3 とした場合に生成した文字列を図 13 に示す．図 13 でピンクの部分が続く数字の並びであり、紫の部分が最初の r 未満の連続に相当する．たとえば図 13 の 8 行目は 000111000111... のように 3 本ずつ経糸を拾わない・拾うを規則的に繰り返すことに対応する．なお、 $r = 1$ は図 1 の平織りに相当するが、 r は連続して経糸を拾う・拾わないを規則的に繰り返す際の最大長を表すものの、経糸の拾い方の簡潔さを表すわけではない．経糸の拾い方の簡潔さの扱いは今後の課題として残されている．また、 r を大きくすると経糸の拾い方のバリエーションは増えるが、経糸と緯糸が織物の表に出る間隔が長くなると織った布が破損や切断しやすくなり [14]，また、手織りで綺麗な織物の制作がしにくくなるという課題もある．

図 13 に示す文字列に対し，左端から組織図の列数 n と同じ長さで打ち切り，最初に右側から緯糸を通して経糸を拾うことが多いために文字列を反転させて行列 $\mathbf{S}$ を生成した。

4.2 結果

4.1 節で述べたシステムを用いて模様をデザインし、それらに対する近似組織図を生成して検証した。なお、織物の制作では経糸と緯糸が一定間隔以内ごとに上に表れるようにする必要があるため [14]，経糸の拾い方の候補を生成する際には，連続する数字の最大長 r は図 13 のように $r = 3$ とした。

4.2.1 アルゴリズムの検証

図 10 に示したアルゴリズムについて、1) 正しさ、2) 目的関数の単調減少性、3) 最適性、を実験を通じて検証した。

まず、平織りのように、図 13 に示した規則的な経糸の

$$r=3 \left[\begin{array}{l} r=1 \\ r=2 \end{array} \right. \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & \cdots \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & \cdots \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & \cdots \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & \cdots \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & \cdots \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & \cdots \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & \cdots \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & \cdots \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & \cdots \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & \cdots \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & \cdots \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & \cdots \end{bmatrix}$$

図 13 規則的な経糸の拾い方

Fig. 13 Regular warp pick up.

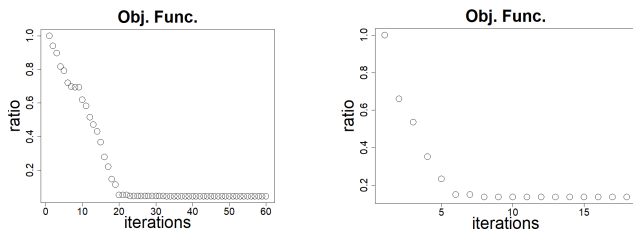


図 14 目的関数の推移 (比率)

Fig. 14 Transition of objective function (ratio).

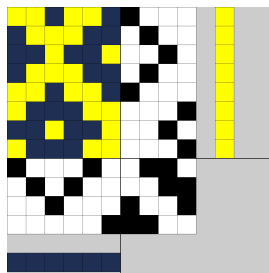


図 15 デザインした模様と織方図

Fig. 15 Weaving draft for designed weave.

拾い方だけを用いて織ることができる組織図を再現できることを確認した。これは、組織図では用いられていない拾い方が行列 S に登録されていた場合でも、近似誤差の最小化に必要な経糸の拾い方だけを選択できることを示す。これにより、アルゴリズムの正しさを確認した。

次に、式 (7) の目的関数が単調に減少して収束することを確認した。図 14 に例を示す。図 14 では図 10 の 1 行目の初期割当てに対する目的関数の値との比を示している。図 10 の 2 行目の while の繰り返し数は組織図に依存するが、実験で用いた行数と列数がそれぞれ 10 行程程度の組織図では数回の繰り返しで終了し、また、経糸の拾い方の再割当てでも数十回程度であった。

最後に、組織図の行数 m と最大長 r を指定して生成した行列 S の行数 s に対し、 s^m 個の経糸の拾い方をすべて列挙し、それらに対して式 (6) を用いて生成した組織図に対する式 (7) の最小値と提案法で生成する組織図に対する値を比較して有効性を検証した。4.1.2 項で述べたように連続する数値の最大長 r を 2 とすると $s = 2 \times 3 = 6$ なるため、たとえば $m = 6$ に対して $6^6 = 46,656$ の可能な割当てが存在するが、検証したすべての例^{*1}に対して図 10 のアルゴリズムが大域的最小化を実現していることを確認した。

4.2.2 織物の制作を通じた検証

提案法で生成した組織図に対する織物を制作した例を図 18 に示す。咲きおり [16] を使用し、手作業で緯糸ごとに経糸を拾って織物を制作した。デザインした模様の織方図を図 15 に、その組織図をもとに生成した組織図を同じ糸の色で表示した織方図を図 16 に示す。図 17 はデザインした模様 (図 15) を制作した織物である。

^{*1} 非常に計算時間がかかるため、6 つの例に対して検証した。

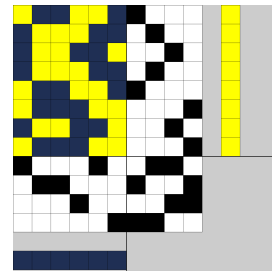


図 16 生成した組織図と織方図

Fig. 16 Weaving draft for generated weave.



図 17 デザインした模様の織物

Fig. 17 Woven fabric for designed weave.

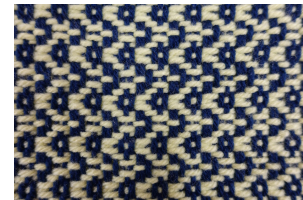


図 18 生成した組織図の織物

Fig. 18 Woven fabric for generated weave.

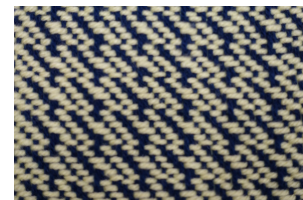


図 19 デザインした模様の織物

Fig. 19 Woven fabric for designed weave.

図 15 と図 16 で踏み方図 A は同じであるが、タイアップ図 B と綜統図 C は異なる。このため、式 (5) で述べたように経糸の拾い方のバリエーションに対応する行列 $B^T C$ は異なるため、異なる組織図となる。提案法で生成した組織図に対する織物 (図 18) を制作した際、糸の拾い方が規則的なため、図 17 の織物と比べて制作しやすくなったことが観察された。また、図 16 の組織図と図 15 の組織図は異なるため織物の模様が一致するわけではないが、図 18 のように綺麗な模様の織物を制作することができた。

別の織物の制作例を図 19、図 20 に示す。主観的な評価とは異なるが、類似した模様の織物を制作できたと考えられる。しかし、それぞれの模様を生成するための組織図は大きく異なる (図 21、図 22 参照)。図 21 では必要な綜統数 (行列 C の行数) は 8、踏み木の数 (行列 A の列数)

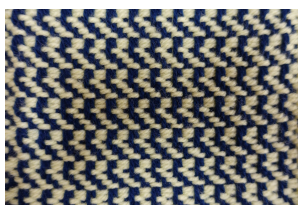


図 20 生成した組織図の織物

Fig. 20 Woven fabric for generated weave.



図 21 デザインした模様と織方図

Fig. 21 Weaving draft for designed weave.



図 22 生成した組織図と織方図

Fig. 22 Weaving draft for generated weave.



図 23 デザインした模様の織物

Fig. 23 Woven fabric for designed weave.



図 24 生成した組織図の織物

Fig. 24 Woven fabric for generated weave.

は 6 であるが、生成した組織図では綜統数は 4、踏み木の数も 4 である。本稿では文献 [6], [7] のように綜統数の最小化を考慮しているわけではないが、組織図の近似を通じて必要な綜統数や踏み木の数を少なくできる可能性もある。

他方、本稿では審美的なことを考慮してはいないため、与えられた模様や経糸の拾い方のバリエーションによってはあまり模様が類似しない場合もある。たとえば図 23 に

示す模様では斜めに交差する模様が見られるが、これを行ごとの経糸の拾い方から考えると、途中に 1 カ所だけ 0 と 1 が反転する拾い方をするのに対応する。4.1.2 項で述べた手順ではこのような経糸の拾い方を生成できないため、生成した組織図の模様（図 24）は似ているとはいいがたい。デザインに応じた経糸の拾い方の候補の生成は今後の課題である。

5. まとめ

リジッドルームを用いて複雑な模様を制作する際の負担の軽減に向けて、経糸の拾い方の割当てに基づいて組織図を近似することを提案した。組織図の敷き詰めにより織物の模様が造られることをタイリングの観点からとらえ、高次局所自己相関特徴を用いて組織図の差異を表現した。また、経糸の拾い方の割当てに基づく組織図の生成をクラスタリングの観点からとらえ、経糸の拾い方の逐次的な再割当てを行うアルゴリズムを示した。

本稿のアプローチを実装し、デザインした模様に対する近似組織図を生成して検証した。アルゴリズムの正当性と目的関数の単調減少性を確認するとともに、大域的最小化を実現することを実験的に確認したが、性質を厳密に示すことは今後の課題である。また、生成した組織図に対する織物を制作して有効性を検証した。生成する組織図は指定する経糸の拾い方にも依存するため、今後は模様に応じた経糸の拾い方の扱いにも取り組む予定である。

謝辞 本研究の一部は文部科学省科研費 (No.21K12542) の補助による。有益なご指摘を賜りました査読者の方々に深く謝意を表します。また、織物を制作した前田佳央理氏に深く謝意を表します。

参考文献

- [1] Dempster, A., Laird, N. and Rubin, D.: Maximum Likelihood from Incomplete Data via the EM Algorithm, *Journal of the Royal Statistical Society*, Vol.39, No.2, pp.1–38 (1977).
- [2] 藤田 正：行列による織方図の計算，繊維機械學會誌，Vol.14, No.7, pp.513–517 (1961).
- [3] Hartigan, J. and Wong, M.: Algorithm AS136: A k-means clustering algorithm, *Journal of Applied Statistics*, Vol.28, pp.100–108 (1979).
- [4] 巴山竜来：数学から創るジェネラティブアート，技術評論社 (2019).
- [5] 五十嵐悠紀，三谷 純：カード織りのためのテキスタイルデザインおよび制作支援，芸術科学会論文誌，Vol.13, No.2, pp.97–106 (2014).
- [6] 松浦 勇，柳浦睦憲，平田富夫：ドビー織機の綜統枠数最小化問題に対する集合被覆アプローチ，情報処理学会論文誌，Vol.50, No.6, pp.1539–1548 (2009).
- [7] 松浦 勇，平田富夫：張力均一条件下におけるドビー織機の綜統枠数最小化，情報処理学会論文誌，Vol.54, No.7, pp.1913–1920 (2013).
- [8] 松浦 勇：等価組織を除外した織物組織図の数え上げ——たて糸本数とよこ糸本数がともに素数の場合，情報処理

- 学会論文誌, Vol.60, No.4, pp.1088–1095 (2019).
- [9] 箕輪直子：手織り大全，誠文堂新光社 (2011).
 - [10] 箕輪直子：はじめての裂き織りレッスン，誠文堂新光社 (2017).
 - [11] 彦根 愛：手織りの組織図辞典，グラフィック社 (2020).
 - [12] Otsu, N. and Kurita, T.: A new scheme for practical flexible and intelligent vision systems, *Proc. IAPR Workshop on Computer Vision (MVA1988)*, pp.431–435 (1988).
 - [13] 杉原厚吉：エッシャー・マジック，東京大学出版会 (2011).
 - [14] 豊浦正広，五十嵐哲也，庄司麻由，茅 暁陽：ジャカード織物作製のための制約付き画像二値化，芸術科学会論文誌, Vol.13, No.3, pp.124–133 (2014).
 - [15] 柳井啓司：一般物体認識の現状と今後，情報処理学会論文誌：コンピュータビジョンとイメージメディア, Vol.48, No.CVIM19, pp.1–24 (2007).
 - [16] クロバー：咲きおり，クロバー株式会社（オンライン），入手先 <https://www.clover.co.jp/seihin/sakiori/>（参照 2021-01-23）.

付 録

A.1 式 (9) の証明

3.4 節の手順で求める $\mathbf{A} \in \{0, 1\}^{m \times p}$ と $\mathbf{C} \in \{0, 1\}^{h \times n}$ は 2.2 節で述べた制約を満たし，また， $\mathbf{A}$ と $\mathbf{C}$ はそれぞれ行ベクトルや列ベクトルとして零ベクトルを含むことはない．このため， $\mathbf{A}^T \mathbf{A}$ と $\mathbf{C} \mathbf{C}^T$ はそれぞれ正の整数を要素とする対角行列となる．このため， $\mathbf{A}^T \mathbf{A}$ と $\mathbf{C} \mathbf{C}^T$ は正則行列となりそれぞれ逆行列が存在する．この性質に基づき，式 (1) を以下のように書き換える．

$$\begin{aligned} \mathbf{W} &= \mathbf{A} \mathbf{B}^T \mathbf{C} \\ \mathbf{A}^T \mathbf{W} \mathbf{C}^T &= (\mathbf{A}^T \mathbf{A}) \mathbf{B}^T (\mathbf{C} \mathbf{C}^T) \end{aligned} \quad (\text{A.1})$$

式 (A.1) に左から $(\mathbf{A}^T \mathbf{A})^{-1}$ ，右から $(\mathbf{C} \mathbf{C}^T)^{-1}$ を掛け，転置をとることで，式 (9) が成り立つ．



吉田 哲也 （正会員）

1968 年生．1991 年東京大学工学部航空工学科卒業．同年大阪大学大学院基礎工学研究科助手．2001 年大阪大学産業科学研究所助手．2004 年北海道大学大学院情報科学研究科助教授．

2014 年奈良女子大学研究院教授．主に機械学習，データマイニング等に興味を持つ．人工知能学会，芸術科学会，日本建築学会各会員．