

複数の正答を持つ単位正方形の組合せパズルに関する研究

山本 陽平¹ 金森 由博² 三谷 純^{2,a)}

受付日 2014年10月1日, 採録日 2015年3月4日

概要: 本論文では, 両面に色の付いた単位正方形をピースとする組合せパズルについて議論する. ピースは両面に色を付けることができ, すべてのピースを過不足なく使用してあらかじめ指定された色模様(「正答」と呼ぶ)を復元することを目的とする. 正答が2つ以下であれば, ピースの各面に適切に色を割り当てることで, 必ずすべての正答を再構成できる. 正答が3つ以上ある場合には, 正答間で共通する色をうまく再利用する必要がある, 場合によってはすべての正答を実現するピースの集合が存在しないこともあるため, パズルの問題を作成するときには注意が必要となる. そこで, 複数の正答が入力として与えられたときに, その正答を実現できるピースの集合が存在するか否かを判定するための条件式と, 存在する場合にはピースの生成を行う手法を提案する. そうでない場合には, 正答の色合いを調整して, ピースを生成可能にする手法の提案も行う. いくつかの例題で提案手法の実験を行い, 実際にパズルを試作したので報告する.

キーワード: ゲーム・パズルの数理的基礎, モノミノ, パズル設計支援

A Study on Square-tile Puzzle with Multiple Answers

YOUHEI YAMAMOTO¹ YOSHIHIRO KANAMORI² JUN MITANI^{2,a)}

Received: October 1, 2014, Accepted: March 4, 2015

Abstract: We discuss about a new tile puzzle whose pieces are single-unit squares with colored front and back faces. The goal is to reconstruct given colored grid patterns with the pieces with nothing left. It is obvious that any two answer patterns are able to be reconstructed by simply assigning an appropriate color to each face of a piece. However, we have to reuse faces of pieces between different answer patterns if the number of answer patterns is more than two. Because this reuse may fail if too many colors are used in the answers, we have to pay careful attention to the design of answer patterns. We found the condition of answer patterns for the existence of a set of pieces which is able to reconstruct all of them. We propose a method for generating pieces from input answer patterns. Our method reduces the number of colors used in the answer patterns if there is no set of pieces that satisfies the answer patterns. We tested our method with several examples, and we made a prototype puzzle.

Keywords: mathematical foundations of games and puzzles, monomino, puzzle design support

1. はじめに

ゲームやパズルを対象とした研究は, 情報科学の分野における探索技術の発展などに大きく貢献している. 図形パ

ズルの一種に, ポリオミノをピースに用いるものがこれまでに数多く考案されている. ポリオミノとは, 単位正方形の辺と辺を平面上でつなぎ合わせて作られる多角形の総称であり, ポリオミノをピースに用いたパズル(以降ではポリオミノパズルと呼ぶ)は, 1953年にGolombにより考案された[6], [7]. その後, ポリオミノパズルには多くのバリエーションが作り出されている. たとえばピースを5つの正方形をつなげたものに限定したパズルはペントミノと呼ばれ世界中で広く親しまれている. また単位正方形に色を割当て, 敷き詰めたときに市松模様が再現されるよう配置

¹ 筑波大学大学院システム情報工学研究科
Graduate School of Systems and Information Engineering,
University of Tsukuba, Tsukuba, Ibaraki 305-8573, Japan

² 筑波大学大学院システム情報系
Information and Systems, University of Tsukuba, Tsukuba,
Ibaraki 305-8573, Japan

^{a)} mitani@cs.tsukuba.ac.jp

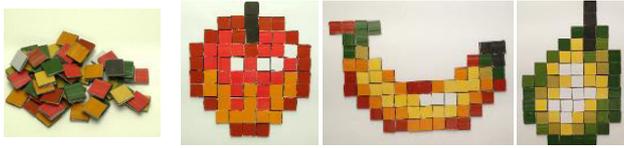


図 1 本論文で提案するパズルの例。両面に色の付いた単位正方形のピース 64 個 (左) から構成される 3 つの正答を持つ

Fig. 1 An example of the puzzle we propose in this paper. This example has three answers reconstructed with 64 unit squares (left) with colored front and back faces.

に制限を与えるなど、正答と判断する条件に制約を与えることで、パズルとしての面白味や難易度に変化を与える試みがなされている。

本論文で対象とするパズルでは、各ピースは 1 つの正方形からなる (モノミノと呼ぶ)。各ピースの裏と表の両面に色を付け、正答のための制約を格子に色を付けた「色模様」で指定する (図 1)。正答を構成するセルの数はピースの数と同一であり、ピースは過不足なく使用する。見えている面で正答を構成することが目的であり、完成時に見えない面の状態は問わない。このようなパズルでは、正答が 1 つであれば、正答となる色模様を分割するだけでピースを作れる。また正答が 2 つであれば、ピースの裏面に第 2 の正答を構成する色を割り当てればよい。しかしながら、正答が 3 つ以上である場合の話は簡単でない。正答間で共通する色を使いまわさないと、すべての正答を実現するピース集合を作れないことがあり、またそもそも、そのようなピース集合が存在し得ないこともある。ここで、興味深い問題が提起できる。複数の正答が与えられたときに、それらを実現できるピース集合が存在するかどうか、どのように判定できるだろうか。また、存在すると判定された場合には、各ピースの表面と裏面に割り当てる色はどのように決めるべきだろうか。さらに、存在しないと判定された場合には、正答の方を修正する必要があるが、どのように修正すればよいだろうか。我々が調べた限り、このような問いの解、および、問いそのものに言及した文献は見あたなかった。本論文では、正答を実現できるピースの集合が存在するか否かを判定するための条件式と、存在する場合にはピースの生成を行う手法を提案する。そうでない場合には、正答の色合いを調整して、ピースを生成可能にする手法の提案も行う。また、いくつかの例題で提案手法の実験を行い、実際にパズルを試作したので報告する。

2. 関連研究

小谷によって行われたパズルの分類 [10] によると、ポリオミノパズルとは、「実世界で与えられた物体を用いる物体のパズル」のうち、与えられた物体から 1 つの物体を復元する「合わせるパズル」に分類される。このパズルを解くためのソフトウェアには、Gerard によって開発さ

れた Polyomino Solver [5], Andreas によって開発された BurrTools [15] などがあり、一般に公開されている。

ポリオミノパズルに関する研究には、西岡によるタイリング可能なポリオミノの列挙を行う研究 [14] や、是川らのポリオミノパズルを作る研究 [11], Erickson による NP 完全なドミノの敷き詰め問題に関する研究 [3] があげられる。是川らは、ポリオミノのセルごとに付与された模様を、ラテン方陣を描くように配置する問題を扱っている。ポリオミノパズルを解くことは、村井らの研究 [13] が集積回路の設計に用いることを前提としているように、あるものを敷き詰める分野に役立つことがある。安倍らは計算ブロックパズルを対象とした研究 [2] を行っている。計算ブロックパズルは見方を変えると、 $1, 2, \dots, n$ までの数値が割り振られた n 枚のモノミノを扱うパズルと見ることができる。色の付いたピースを用いて、複数の異なる絵柄を復元するパズルには、株式会社テンヨーの商品である JIGAZO PUZZLE [17] がある。このパズルは、各ピースの色の濃淡の違いを利用し、並び順を変えることで様々な絵柄を作り出している。

近年の 3D プリンタの普及とともに、3 次元物体の表面を、歪みを許容したポリオミノの集合に分割にする研究 [12] や、モデルの内部で互いに固定し合うピース (lock 構造を持つピース) に分解する研究 [16], [18] など、立体構造を持つ組合せパズルに関する研究も行われている。

異なる複数の正答を共通のピース集合から復元するパズルとして、「裁ち合わせパズル」があげられる。本研究で対象とするパズルは色に注目するが、裁ち合わせパズルは幾何的な形に注目する。2 種類以上の面積の等しい図形が与えられたときに、一方の図形を複数の部品に分割して並べ替え、他方の図形を復元することを目的とする。一方を有限個の部品に分割して他方の図形を復元できることは、ボヤイの定理として知られているため、パズルとしては部品数が最小のものを考える。このパズルは、小谷によるパズルの分類では、「紙面上で行う抽象的なパズル」のうちの図形を対象とする「図形パズル」に属する。図形パズルを対象にした研究は、小谷らによる 1 枚のポリオミノ図形を、合同な 2 枚の図形に分割する研究 [9] などがあげられる。裁ち合わせパズルを対象とした研究には、Czyzowicz らによる長方形を正方形に裁ち合わせるシステム [4] や、Abbott らによる多角形をヒンジでつながるピースで裁ち合わせる研究 [1] があげられる。また、ポリオミノ図形を対象にした研究として、五十嵐らによる全解探索システム [8], Zhou による複数の図形を裁ち合わせる解を求める研究 [19] などあげられる。近年では、3 次元の物体を同体積の立方体の形に裁ち合わせる研究も行われている [20]。

ここであげたように、ピースを組み合わせて形を作るパズルに関して様々な研究が行われているが、本論文で扱うものと同様のパズルを対象にしたものは、我々が調べた限りでは存在しなかった。

3. 複数の正答を構成可能なピース集合

3.1 用語の定義

以降での議論を進めるために、まず用語の定義を行う。本研究で対象とするパズルは N 個の単位正方形をしたピースで構成される。ピースをセルと呼ぶこととする。セルには表面と裏面があり、これをセル面と呼ぶ。セル面にはそれぞれに任意の色を割り当てることができる。すべてのセルを過不足なく使用し、手前を向いたセル面が正答として与えられた色模様と一致している状態を作ることがパズルの目的である。

すべての正答の中で使用されている色の数を L とする。異なる正答の数を M とし、 $i \in \{1, 2, \dots, M\}$ 番目の正答に含まれるセル面の集合を g_i で表す。 g_i の要素となるセル面は、配置される位置および色の情報を持つ。正答はすべてのセルを過不足なく用いるため $|g_i| = N$ である。

g_i に含まれるセル面の色を抽出した集合を、重複集合 g_i^c と表記し、ある色 c_j が何枚含まれているかを重複度関数 $m_{g_i^c}(c_j)$ で示す。たとえば、 g_1 に含まれるセル面の数が4で、各セル面の色が $\{\alpha, \alpha, \beta, \gamma\}$ である場合、 $g_1^c = \{\alpha, \alpha, \beta, \gamma\}$ 、 $m_{g_1^c}(\alpha) = 2$ 、 $m_{g_1^c}(\beta) = 1$ 、 $m_{g_1^c}(\gamma) = 1$ である。また、 g_1^c と g_2^c の積集合を $A_{1,2}$ と表記する。たとえば、 g_1^c が先ほどのとおりであり、 g_2^c が $\{\alpha, \alpha, \alpha, \gamma\}$ である場合、 $A_{1,2}$ は $\{\alpha, \alpha, \gamma\}$ である。さらに、 g_1^c と g_2^c および g_3^c の積集合は $A_{1,2,3}$ のように記す。このようにした場合、集合 A の添え字は自然数 $\{1, 2, \dots, M\}$ に対するべき集合 $\mathfrak{P}(\{1, 2, \dots, M\})$ から空集合を除いたものと等しく、全部で $2^M - 1$ 通り存在する。この集合を U_M で表す。つまり、 U_M の要素 $u \in U_M$ を用いて、 $A_u = \prod_{i \in u} g_i^c$ と定義できる (演算子 \prod は重複集合に対する直積演算を表す)。

さらに、自然数 i を含む集合のみを要素に持つ、 U_M の部分集合を $U_M^{(i)}$ 、 U_M から要素 $\{1, 2, \dots, M\}$ を取り除いたものを U_M' と記し、 $\{1, 2, \dots, M\} \setminus u$ を \bar{u} と記すものとする。これらの例を表 1 に示す。

定義 M 集合のベン図へのセル面振り分け問題

M 個の集合のベン図の各領域に対して、次の2つの条件を満たすようにセル面を割り振る問題を「 M 集合のベン図へのセル面振り分け問題」(以降では、単に「セル面振り分け問題」と記すこともある)と呼ぶこととする。ただし、ベン図の領域 u に含まれる要素の集合を a_u とする。

$$\forall i \in \{1, 2, 3, \dots, M\}, \prod_{u \in U_M^{(i)}} a_u = g_i^c \quad \text{条件 1}$$

$$\forall u \in U_M', |a_u| = |a_{\bar{u}}| \quad \text{条件 2}$$

演算子 \prod は重複集合に対する直和演算を表す。図 2 では、 a_u とベン図の各領域の対応を示す。図から読み取れるように a_u は A_u の部分集合である。条件 1 は、添え字 i を含む a_u の集合から g_i^c を復元できることを意味し、条件

表 1 $M = 3$ のときの表記の例

Table 1 Example of notations when $M = 3$.

$U_3 = \{\{1\}, \{2\}, \{3\}, \{1,2\}, \{2,3\}, \{1,3\}, \{1,2,3\}\}$
$U_3' = U_3 \setminus \{1,2,3\} = \{\{1\}, \{2\}, \{3\}, \{1,2\}, \{2,3\}, \{1,3\}\}$
$U_3^{(1)} = \{\{1\}, \{1,2\}, \{1,3\}, \{1,2,3\}\}$
(u, \bar{u}) の例: $(\{1\}, \{2,3\}), (\{2\}, \{1,3\}), (\{3\}, \{1,2\})$

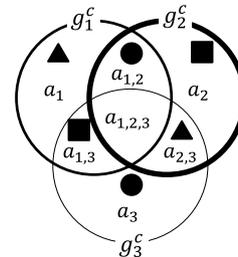


図 2 $M = 3$ におけるベン図と記号による領域の対応付け

Fig. 2 The Venn diagram of $M = 3$. The correspondences between areas are shown by the symbols.

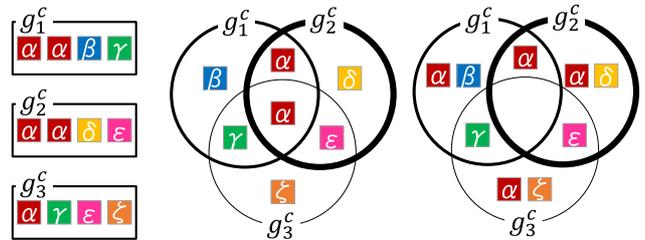


図 3 3つの正答の色集合の例 (左)。セル面振り分け問題の解となる例 (中)。条件 2 を満たさないためにセル面振り分け問題の解とならない例 (右)。

Fig. 3 An example of the sets of colors for three answers (left). An example of a valid answer of the Venn diagramming problem (center). An example of an invalid answer (right).

2 は、図 2 に示すベン図で、同じ記号を付けた領域 (たとえば a_1 と $a_{2,3}$) に含まれるセル面の数が等しいことを意味する。より具体的な例を図 3 に示す。図 3 に示す3つの正答 g_1^c, g_2^c, g_3^c は、そのセル面を3集合のベン図の領域に割り振る方法が図中央や図右のように複数存在する。セル面の割り振りとして a_1 を例にあげると、図中央は $a_1 = \{\beta\}$ 、図右は $a_1 = \{\alpha, \beta\}$ だが、どちらも $A_1 = \{\alpha, \alpha, \beta, \gamma\}$ の部分集合である。そのなかに、セル面振り分け問題の条件を満たす割り振り方 (図中央) が存在するため、 $\{g_i^c\}$ はセル面振り分け問題の解を持つといえる。

3.2 本節の構成

前項では用語の定義を行った。以降では、まず次の定理 1 を証明する (3.3 項)。

定理 1 M 種類の正答を構成可能なピース集合の存在の有無は「 M 集合のベン図へのセル面振り分け問題」の解の有無と同値である。

続いて、 $M = 3$ におけるセル面振り分け問題が解を持つための条件式を示す (3.4 項). さらに、この条件式を判定し、解が存在する場合には、その解の生成を $O(N)$ で実現するアルゴリズムを示す (3.5 項). 最後に、セル面振り分け問題の結果から、実際の色付き図形パズルを作成する方法について述べる (3.6 項).

3.3 定理 1 の証明

証明 (十分条件)

M 種類の正答が与えられたときに、そのすべての正答を構成可能な N 枚のセルの集合が存在するならば、セル面振り分け問題は解を持つことを示す.

与えられた正答 g_i に含まれるすべてのセル面に、数字 i を記録する操作を行うものとする. たとえば、あるセルの表面が g_1 と g_2 で使用され、裏面が g_3 で使用される場合には、表面には数字の 1 と 2 が記録され、裏面には 3 が記録される. 各セル面に記録される数字の集まりは自然数 $\{1, 2, \dots, M\}$ のべき集合の要素である. たとえば $M = 3$ であれば、セル面に記録される内容の種類は最大で 2^3 通りで、具体的には以下の 8 通りである.

$$\phi, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}$$

セル面に記録された数字の集合 (べき集合の要素) を u とし、そのセル面の色をベン図の a_u で表される領域に割り振る. セル面振り分け問題の条件 1 は、数字 i が記録されたセル面が含まれる a_u の直和集合は g_i^c と等しい、ということ を主張している ので、これは当然成り立つ.

1 枚のセルに着目すると、両面合わせて 1 から M までの番号が必ず 1 つずつ割り当てられる. 裏面と表面の両方に同じ数字が記録されることはないため、一方の面に割り当てられた数字の集合を u とすると、他方の面に割り当てられた数字の集合は $\{1, 2, \dots, M\} \setminus u$ である. 正答数が 3 のときを例にあげると、各セルの両面に割り当てられる数字には、表と裏を区別せずに次の 4 通りの組みが存在する.

$$(\{1\}, \{2, 3\}), (\{2\}, \{1, 3\}), (\{3\}, \{1, 2\}), (\phi, \{1, 2, 3\})$$

定義より、 $a_{\{1, 2, 3, \dots, M\} \setminus u} = a_{\bar{u}}$ であるから、つねに $|a_u| = |a_{\bar{u}}|$ を満たす (たとえば、 $\{2\}$ の出現回数と $\{1, 3\}$ の出現回数は同じであることを意味する). よって、セル面振り分け問題の条件 2 が成り立つ. (証明終)

証明 (必要条件)

セル面振り分け問題の解となるようなセル面の割り振りを表す $\{a_u\}$ が存在するならば、すべての正答を構成可能な N 枚のセルの集合が存在することを示す.

ベン図の各領域を、図 2 に示すような数字の集合によって区別する. 各領域に含まれるすべてのセル面に、この数字の集合 u を記録した場合、セル面振り分け問題の条件 2 より、 $u = 1, 2, \dots, M$ 以外の場合は、 \bar{u} を記述したセル面

がちょうど同じ数だけ存在する. ここで、 u を記述したセル面と \bar{u} を記述したセル面を 1 対 1 で対応させて 1 枚のセルを作成すると、作成された各セルは、表面と裏面に、同じ正答番号を割り当てずに済ませることができる. 条件 1 より、正答 g_i を、数字 i を含むセル面から構成できる.

(証明終)

以上より、 M 種類の正答を構成可能なピース集合が存在することと、セル面振り分け問題が解を持つことが同値であることが示せたので、定理 1 を証明できた.

3.4 $M = 3$ のセル面振り分け問題が解を持つための条件式

定理 1 より、 M 種類の正答を構成できるピース集合が存在するか否かは、セル面振り分け問題の解となる $\{a_u\}$ が存在するか否かである. この判定を行うための単純なアプローチとして、取りうるすべての $\{a_u\}$ について、それがセル面振り分け問題の解となっているかどうかを調べる方法が考えられる. では、 $\{a_u\}$ の場合の数は何通りあるだろうか.

a_u は重複集合 A_u の部分集合であるから、 a_u のとりうる場合の数は最大で $2^{|A_u|}$ だけ存在する. その全組合せは最大で $\prod_{u \in U_M} 2^{|A_u|}$ だけある (ここでの演算子 \prod は総乗を意味する). $|U_M|$ は 2^M である. $|A_u|$ の値を評価するのは難しいが、セルの数 N に比例すると仮定すると、 $\{a_u\}$ の場合の数は $O(2^{2^M N})$ となり、 N または M の値が大きくなると、現実的な時間で判定することが困難になることが予想される.

そこで、これよりも遥かに小さいオーダー $O(N)$ で、セル面振り分け問題が解を持つかを簡単に判定できる条件式を示す. なお、ここで示すのは $M = 3$ についてのみである. $M = 1, 2$ の場合は解を持つことが自明であり、 $M > 3$ については本論文では未解決である.

さて、一般性を失うことなく与えられた 3 つの正答のうち、任意に選択した 2 正答を g_1, g_2 、残りの正答を g_3 とする. ここで、3 つの正答に対するセル面振り分け問題が解を持つ条件は、次式で与えられる.

$$N - |(a_1^{(2)} \sqcup a_2^{(2)}) \cap g_3^c| \leq |a_{1,2}^{(2)}| \quad (1)$$

右肩の添え字の (2) は、2 つの正答を対象としたセル面振り分け問題で用いた集合であることを明示的に示すためのものであり、以降では正答数が 3 の場合の議論を進める. 左辺の $|(a_1^{(2)} \sqcup a_2^{(2)}) \cap g_3^c|$ は、 $a_1^{(2)}$ または $a_2^{(2)}$ に含まれるセル面のうち、 g_3^c で使用できる数 (重複集合の演算であることに注意されたい.) であり、 $(a_1^{(2)} \sqcup a_2^{(2)}) \cap g_3^c = a_{1,3} \sqcup a_{2,3}$ を満たすよう $a_{1,3}$ と $a_{2,3}$ を割り当てられる. 左辺全体では、 $a_1^{(2)}$ または $a_2^{(2)}$ に含まれない g_3^c のセル面の枚数を示し、そのセル面を a_3 に割り当てる. 一方、右辺は $a_{1,2}$ に割り当てられる最大のセル面の枚数、すなわち g_3^c のため

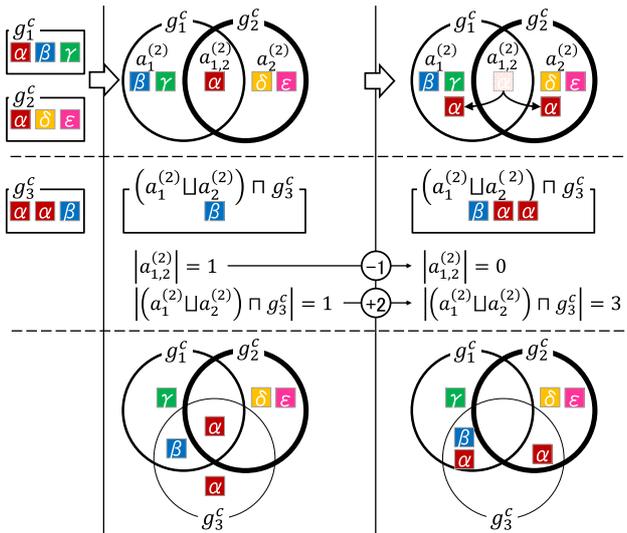


図 4 ある 3 つの正答 (左) に対する色の集合と式 (2) 左辺の各項の関係の色 α の割当て方に着目して見た例 (中, 右)

Fig. 4 A three answers set (left) and the illustration of the relation between the left-hand side of Eq. (2) and the Venn diagramming of color α (center, right).

に自由に割当てを調整できるセル面の枚数を示す. 割り当てられていない $a_1^{(2)}$ と $a_2^{(2)}$ のセル面をそれぞれ a_1 , a_2 に割り当てると, 式 (1) の右辺が左辺と等しければ $a_{1,2}^{(2)}$ を直接 $a_{1,2}$ に, 右辺が左辺より大きければ a_3 と同じ大きさだけ $a_{1,2}^{(2)}$ の要素を $a_{1,2}$ に割り当て, その余りを a_1 と a_2 に分離して割り当てることによって, $M = 3$ におけるセル面振り分け問題は解を持つことができる.

3.5 $M = 3$ における, セル面振り分け問題が解を持つための条件式の判定と, 解の導出

式 (1) の項を移動させて, 次のように変形する.

$$|a_{1,2}^{(2)}| + |(a_1^{(2)} \sqcup a_2^{(2)}) \cap g_3^c| - N \geq 0 \quad (2)$$

図 4 に示すように, $a_{1,2}^{(2)}$ の 1 枚のセル面を $a_1^{(2)}$ および $a_2^{(2)}$ に割り振ると, $|a_{1,2}^{(2)}|$ の値が 1 減少するが, 割り振ったあとの 2 つのセル面が g_3^c に含まれるのであれば, 第 2 項 $|(a_1^{(2)} \sqcup a_2^{(2)}) \cap g_3^c|$ の値が 2 増える. つまり, 左辺の値を 1 増やすことができる. 左辺の値が最大になるのは, $a_1^{(2)}$ および $a_2^{(2)}$ に割り振った 2 つのセル面が g_3^c に含まれる限り, この割り振り操作を最大回数行ったときである. このようにして左辺の値を最大にしても 0 未満である場合は, セル面振り分け問題は解を持つことができずと判定できる. なお, 上記の操作は色ごとに独立して行い, すべての色に対して条件式を満たす場合のみ, セル面振り分け問題は解を持つということができる.

図 4 の例では, 左側のベン図では, 左辺が -1 となり, 式 (2) の関係を満たさない. 一方で, 右側のベン図では左辺が 0 となり式 (2) の関係を満たす. 右側のベン図に示すようなセル面の割り振りを行った場合は, たとえば各セル

表 2 色 α の割り振り例

Table 2 Examples of allocations of color α .

	Case1	Case2	Case3	
$m_{g_1^c}(\alpha)$	1	1	2	1
$m_{g_2^c}(\alpha)$	1	3	2	2
$m_{g_3^c}(\alpha)$	3	1	2	2
分類				

に対して, (表, 裏) = (γ, α) , (β, δ) , (ϵ, α) のような色を割り当てることによって, 3 つの正答を構築できる.

式 (2) の左辺が最大となるように各正答の色集合をベン図の領域に割り振るには, 色ごとに次に示す方法を適用することで実現できる. 具体例を表 2 に示す.

Case1 「 g_3^c に含まれる色 α のセル面の数 $\geq g_1^c$ と g_2^c に含まれる色 α のセル面の数の和」である場合, すべて $a_1^{(2)}$ と $a_2^{(2)}$ に配分し, できるだけ $a_{1,3}$ と $a_{2,3}$ に割り当てられる枚数を増やす.

Case2 「 g_3^c に含まれる色 α のセル面の数 $\leq g_1^c$ と g_2^c に含まれる色 α のセル面の数の差」である場合, できるだけ $a_{1,2}^{(2)}$ に統合して自由に割当てを調整できるセル面の枚数を増やし, $a_1^{(2)}$ もしくは $a_2^{(2)}$ に溢れたセル面を $a_{1,3}$ もしくは $a_{2,3}$ に割り当てる.

Case3 「 g_1^c と g_2^c に含まれる色 α のセル面の数の差 $< g_3^c$ に含まれる色 α のセル面の数 $< g_1^c$ と g_2^c に含まれる色 α のセル面の数の和」である場合, 上記の中間で, ちょうど $a_{1,3}$ と $a_{2,3}$ に割り当てられるような統合・分離を行う. 具体的には $m_{g_3^c}(\alpha) - |m_{g_1^c}(\alpha) - m_{g_2^c}(\alpha)|$ が不足分なので, この値を超えない範囲で最大の分離を行う.

以上をまとめると, g_1 , g_2 に対するセル面振り分け問題の解となる $\{a_u^{(2)}\}$ の $a_{1,2}^{(2)}$ が含むべき色の枚数は, 色ごとに次式で記述できる. なお, g_1^c , g_2^c , g_3^c に含まれる各色 c_j の枚数を $g_{i,j}$ ($= m_{g_i^c}(c_j)$) として記述する.

$$m_{a_{1,2}^{(2)}}(c_j) = \begin{cases} 0 & (g_{1,j} + g_{2,j} \leq g_{3,j}) \\ \min\{g_{1,j}, g_{2,j}\} & (g_{3,j} \leq |g_{1,j} - g_{2,j}|) \\ \min\{g_{1,j}, g_{2,j}\} - \left\lfloor \frac{(g_{3,i} - |g_{1,j} - g_{2,j}|)}{2} \right\rfloor & (|g_{1,j} - g_{2,j}| < g_{3,j} < g_{1,j} + g_{2,j}) \end{cases} \quad (3)$$

この方法で得られる $a_{1,2}^{(2)}$ を用いて, $a_1^{(2)} = g_1^c \setminus a_{1,2}^{(2)}$, $a_2^{(2)} = g_2^c \setminus a_{1,2}^{(2)}$ を定義し, 式 (1) が満たされるならば, セル面振り分け問題が解を持つ.

この判定は前処理として, 各正答に含まれる色の枚数をカウントする. これには $O(N)$ の計算量を必要とする. 実際の判定は, 色の数 L だけ繰り返されるため $O(L)$ である. 色の数の最大値は $3N$ であるため, 全体の計算量は $O(N)$ である.

3.6 セル面振り分け問題の解からセルを作成する方法

正答 $\{g_i\}$ の色の集合 $\{g_i^c\}$ から、セル面振り分け問題の解である $\{a_u\}$ が与えられたときに、 N 個のセル (パズルのピース) は次の方法で作成する。まず、 $u \in U'_M$ の各 a_u と $a_{\bar{u}}$ 間で要素の 1 対 1 の対応を任意にとり、この 2 つの色をセルの表と裏に割り当てる。 $a_{1,2,\dots,M}$ に含まれるセル面は、反対側の面を使用しないセルであるため、使用しない方の面には任意の色を適当に割り当てる。このようにして、計 N 個のセルを得る。

4. 色の調整

本節では、セル面振り分け問題が解を持たないような正答が与えられた場合に、正答に含まれるセル面の色数を減らすことで、解が存在するように調整する手法を提案する。次の定理 2 より、色数を減らすことで必ずセル面振り分け問題が解を持つようにできる。

定理 2 セル面振り分け問題が解を持たない場合には、正答に含まれる色を減らすことで、必ずセル面振り分け問題が解を持つようにできる。

定理 2 が真であることは、正答に含まれる色の数がわずか 2 色 $\{\alpha, \beta\}$ であれば、各セルの片面に α 、反対の面に β を割り当てることで、すべての正答を構成できることから自明である。

与えられた $\{g_i^c\}$ がセル面振り分け問題の解を持たない場合は、できるだけ元の色を減らすことなく、セル面振り分け問題が解を持つ色の集合に調整することが望ましい。そのため、1 色ずつ減色を行う方法として、以下のアルゴリズム 1 を提案する。

アルゴリズム 1 減色の方法

WHILE セル面振り分け問題が解を持たない

$\{g_i^c\}$ に含まれる色から、距離 $d(\mathbf{c}_1, \mathbf{c}_2)$ を最小にする 2 色 $\mathbf{c}_1, \mathbf{c}_2$ を見つける

$\{g_i^c\}$ に含まれる \mathbf{c}_1 あるいは \mathbf{c}_2 の色を持つセル面集合を、それぞれ C_1, C_2 とする

C_1 と C_2 に含まれる要素数を重みとして、新しい中間色 \mathbf{c}_3 を決定する

C_1 と C_2 に含まれるセル面の色を \mathbf{c}_3 とする

END WHILE

色を RGB 色空間内のベクトル $\mathbf{c}_i \in [0, 1] \times [0, 1] \times [0, 1]$ と見なし、アルゴリズム 1 で用いられている距離関数 d を以下のように定義する。

$$d(\mathbf{c}_1, \mathbf{c}_2) = \begin{cases} \text{infinity} & \mathbf{c}_1, \mathbf{c}_2 \text{ が共通する正答ただ 1 つだけに含まれる場合} \\ \|\mathbf{c}_1 - \mathbf{c}_2\| & \text{それ以外} \end{cases} \quad (4)$$

$\mathbf{c}_1, \mathbf{c}_2$ が共通する正答ただ 1 つのみ含まれる場合、距

離は無限大とする。これはその 2 色を別の色に置き換えても、セル面振り分け問題の結果が変わらないためである。それ以外のときは 3 次元ベクトル間のユークリッド距離とする。

中間色 \mathbf{c}_3 は以下の式で定める。

$$\mathbf{c}_3 = \frac{|C_1|}{|C_1| + |C_2|} \mathbf{c}_1 + \frac{|C_2|}{|C_1| + |C_2|} \mathbf{c}_2 \quad (5)$$

このアルゴリズムの前処理として、色の値を類似度で並び替えるのに $O(L \log L)$ かかるが、距離の近い 2 色を見つけるのに $O(L)$ 、ループの実行数を $O(L)$ と仮定すると、全体で $O(L^2)$ の計算量である。データの持たせ方を工夫することでさらに計算量を減らすことが可能かもしれないが、ここではこれ以上は議論しない。

5. 実験

5.1 セル面振り分け問題の全探索と提案手法の比較

入力として与えられた正答集合がセル面振り分け問題の解を持つか否かの判定を全探索で行うプログラムと、本論文で提案する手法で行うプログラムを作成し、計算時間を比較する実験をした。どちらのプログラムも、解を持たないと判定した場合は、4 章で述べた方法で色の調整を行う。実験はメモリ 8 GB、Intel(R) Core(TM) i7-4700MQ CPU @ 2.40 GHz の PC 上で行った。OS は Windows 8.1 でプログラムは Java 言語で開発した。

まず、入力とする正答として、次のようにセル数が 4 だけの簡単なものを 3 つ用いた。

$$g_1^c = \{\alpha, \alpha, \beta, \gamma\}, g_2^c = \{\delta, \delta, \epsilon, \zeta\}, g_3^c = \{\alpha, \beta, \delta, \eta\}$$

その結果、このままでは解を持たないと判定され、減色処理が 1 回行われた。この処理を全探索では 48 ミリ秒、提案手法では 8 ミリ秒で終えた。提案手法の方が速いが、その差は僅かであったため、上記の 3 つの正答に対して、色の構成比率を保ったままでセル数を 3 倍の 12 とし、同じ実験を行った。その結果、全探索では 96,622 ミリ秒、提案手法では 8 ミリ秒を要した。全探索の時間が大幅に増えたのは、セルの数が 3 倍に増えただけでも、組合せの数が大きく増えたためと見られる。セル数をこれ以上に増やすと、全探索では現実的な時間で判定するのが困難になると予想される。提案手法の処理時間がセル数を増やしても変わらなかったのは、判定に要する計算量が色数のみに依存し、セル数には依存しないためである。

続いて、前述の 3 つの正答に対して、新しく $g_4^c = \{\alpha, \zeta, \zeta, \iota\}$ を加えた 4 つの正答に対しての実験を行った。提案手法は正答数が 3 の場合のみを対象としているため、4 つの正答の判定は全探索のみで行った。セル数はわずか 4 であったが、処理に要した時間は 621,712 ミリ秒であった。その中で減色処理は 3 回行われ、その度に組合せ数が

5.24×10^5 , 8.39×10^6 , 2.83×10^7 と増加することを確認した。これは、共通する色が多いほど、 $A_{1,2}$ などの、複数の正答の共通領域が大きくなり、組合せの数が大幅に増えたものと考えられる。この共通領域の大きさは、正答の状態によって大きく異なるため、一概にいうことはできないが、正答を4つ持つ問題を扱うには、セルの数が十分に少なく、かつ共通する色の種類が少ない必要があると考えられる。

5.2 提案手法によるセル面振り分け問題の解法

前項による実験で、全探索ではセル数が12であっても、必要な処理に要する時間が大きいことを確認した。ここでは、提案手法を用いることで、セル数が64, 430, 2,400と十分大きな例を用いて実験を行い、計算時間および減色による影響について検証した。実験の環境は前項と同じである。この結果を表4に示す。表中の「減色数」は入力と出力における色数の差を示し、「色距離」は減色処理によって、セル面あたり、どれだけ異なる色に変更されたかを示す、次式で定める値(0から1の範囲を取る)である。値が小さいほど色の変化が小さいことを表す。

$$\frac{1}{\sqrt{3NM}} \times \sum_{\text{(すべての正答)}} \sum_{\text{(正答に含まれるセル面)}} |\text{減色前の色} - \text{減色後の色}| \quad (6)$$

「色の割り振り例」ではセル面振り分け問題の解の例について、ベン図の各領域に割り振られるセル面の色の種類と枚数を示す。

実験1の正答1を例にあげると、 $\{a_u\}$, $u \in U_M^{(1)}$ の集合に含まれる色とその枚数を数えあげ、出力された正答に含まれるセル面と比較することで、間違いなく正答1を復元できることを確認できる(セル面振り分け問題が解を持つための条件1)。また、 a_u と $a_{\bar{u}}$ に含まれるセル面の数が一致することから、ある正答に含まれる2つのセル面が、同じセルの表と裏に割り当てられることを回避できることを確認できる(セル面振り分け問題が解を持つための条件2)。

実験1, 2は、類似する色を持つ2つの正答に対して異なる3つ目の正答を与えたものである。実験2は実験1よりも色の変化が大きい(減色の影響が大きい)ことを確認できる。実験1は第3の正答で使われている色が第2の正答のものと類似しているが、実験2の第3の正答は第1, 2の正答と大きく異なる色を持っている。そのため実験2では、正答1, 2の間で色を統合する作用が働いたものと考えられる。このことから、3つの正答を用意する際には、なるべく共通の色を使った方が色の変化を抑えられるといえる。

実験3, 4は、セル数が430の正答の塗り方を変えて、その違いを確認した例である。実験3は少ない色数で塗り、

表3 ユーザテストでの解答時間

Table 3 Time taken to complete a puzzle in our user test.

	1人目	2人目	3人目	4人目	平均
正答1	6m27s	4m02s	5m33s	3m34s	4m54s
正答2	6m11s	11m47s	4m56s	7m51s	7m41s
正答3	5m44s	4m02s	4m38s	2m58s	4m20s

実験4はグラデーションをつけることで色数を増やした。その結果、色数を増やした方が減色の影響を小さく抑えられることが確認できた。これは、減色処理は同じ色のセルの色を一度にまとめて変更してしまうため、少ない色数だと減色の影響が広範に及ぶためと考えられる。対策として、一度に複数のセルの色を変更するのではなく、1つずつセルの色を変更しながらセル面振り分け問題の解の有無について調べる方法が考えられる。

実験5は計算時間の評価のために、2,400セルという、パズルにするには現実的でない大型の正答を用いたものである。この例でも計算時間は1秒未満であり、提案手法は十分高速であることを確認した。

5.3 パズルとしての利用と考察

表4に示した例題の中で、実験1のパズルを実際に制作し評価を行った(図1)。解き手が正答1の復元を試みる時、まず正答1のみに含まれる色および、正答1に含まれない色を持つセルは、使用する面をただちに決定できる。それ以外のセルは、どちらの面も正答1に含まれるために、どちらを表にするべきかで悩むことになる。どちらかを仮定して復元を進めていくと、たとえば、最後に残った1ピースが期待する場所に配置できない(欲しい色が表にも裏にもない)というケースが発生する。その場合は、すでに配置済みのピースの裏側をそれぞれ確認し、残った1ピースとどれかを交換する必要が生まれる。解き手は、一度に片方の面しか見られないため、裏側の色を把握していなければ、欲しい裏側の面を探すのに悩むことになる。制作したパズルを4名の被験者が完成させるのに要した時間を表3に示す。パズルを解くのにある程度の試行錯誤と時間を要したことから、十分に遊べるものであったと評価できる。解答時間の平均は正答2が最も長く、これは共通して利用する色($a_{1,2}$, $a_{2,3}$ の要素)が多いため、どちらの面を利用するのか分からないピースが増えることが原因であると推測できる。この点をあらかじめふまえて使用する色を決めることで、パズルの難易度の調整が可能になると考えられる。

6. 結論と今後の課題

本論文では、両面に色の付いた正方形をピースとし、複数の正答を持つ組合せパズルについて議論し、提案した手法を実装したシステムの開発を行った。システムの入力は、

表 4 正答数が3の例

Table 4 Results of puzzles that have three answers.

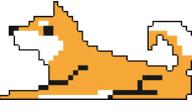
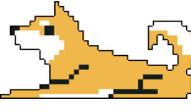
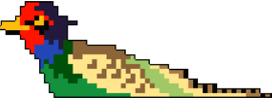
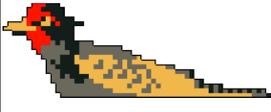
実験 1

セル数	64	色数	入力 17, 出力 11	減色数	6	色距離	0.03	計算時間	23ms	
正答 1			正答 2			正答 3				
入力 (9 色)		出力 (6 色)		入力 (10 色)		出力 (8 色)		入力 (6 色)		出力 (6 色)
										
色の割り振り例										
$a_1 = \{\text{red } 17, \text{orange } 5, \text{yellow } 4\}, a_2 = \{\text{yellow } 2, \text{green } 2, \text{orange } 1\}, a_3 = \{\text{green } 12, \text{green } 10, \text{yellow } 6, \text{white } 3\},$ $a_{2,3} = \{\text{yellow } 15, \text{green } 5, \text{yellow } 2, \text{white } 3, \text{black } 1\}, a_{1,3} = \{\text{black } 1, \text{white } 4\}, a_{1,2} = \{\text{orange } 16, \text{yellow } 15\}, a_{1,2,3} = \{\text{black } 2\}$										

実験 2

セル数	64	色数	入力 21, 出力 4	減色数	17	色距離	0.16	計算時間	41ms	
正答 1			正答 2			正答 3				
入力 (9 色)		出力 (2 色)		入力 (10 色)		出力 (3 色)		入力 (7 色)		出力 (4 色)
										
色の割り振り例										
$a_1 = \{\text{orange } 6\}, a_2 = \{\text{black } 1\}, a_3 = \{\text{black } 35, \text{green } 10, \text{green } 6\}, a_{2,3} = \{\text{black } 4, \text{green } 2\}, a_{1,3} = \{\text{orange } 1\}, a_{1,2} = \{\text{orange } 51\}, a_{1,2,3} = \{\text{black } 3, \text{orange } 3\}$										

実験 3

セル数	430	色数	入力 16, 出力 5	減色数	11	色距離	0.07	計算時間	58ms	
正答 1			正答 2			正答 3				
入力 (3 色)		出力 (3 色)		入力 (4 色)		出力 (4 色)		入力 (11 色)		出力 (4 色)
										
色の割り振り例										
$a_1 = \{\text{yellow } 143, \text{white } 94\}, a_2 = \{\text{red } 64, \text{grey } 109\}, a_3 = \{\text{red } 2, \text{black } 1, \text{grey } 5\},$ $a_{2,3} = \{\text{red } 32, \text{black } 57, \text{grey } 148\}, a_{1,3} = \{\text{black } 25, \text{yellow } 148\}, a_{1,2} = \{\text{yellow } 8\}, a_{1,2,3} = \{\text{black } 8, \text{yellow } 4\}$										

実験 4

セル数	430	色数	入力 33, 出力 8	減色数	25	色距離	0.06	計算時間	112ms	
正答 1			正答 2			正答 3				
入力 (7 色)		出力 (3 色)		入力 (9 色)		出力 (4 色)		入力 (22 色)		出力 (8 色)
										
色の割り振り例										
$a_1 = \{\text{yellow } 148, \text{white } 11\}, a_2 = \{\text{grey } 170, \text{red } 40\}, a_3 = \{\text{blue } 13, \text{grey } 2, \text{green } 15, \text{red } 1, \text{blue } 4, \text{white } 1\},$ $a_{2,3} = \{\text{grey } 61, \text{black } 78, \text{red } 20\}, a_{1,3} = \{\text{yellow } 66, \text{black } 88, \text{white } 56\}, a_{1,2} = \{\text{yellow } 1, \text{black } 35\}, a_{1,2,3} = \{\text{yellow } 5, \text{black } 20\}$										

実験 5

セル数	2400	色数	入力 52, 出力 10	減色数	42	色距離	0.07	計算時間	485ms	
正答 1			正答 2			正答 3				
入力 (34 色)		出力 (9 色)		入力 (25 色)		出力 (8 色)		入力 (11 色)		出力 (5 色)
										
色の割り振り例										
$a_1 = \{\text{white } 637, \text{yellow } 77, \text{yellow } 155, \text{yellow } 287\}, a_2 = \{\text{red } 196, \text{red } 364\}, a_3 = \{\text{black } 68, \text{blue } 574\},$ $a_{2,3} = \{\text{black } 684, \text{blue } 144, \text{blue } 328\}, a_{1,3} = \{\text{white } 112, \text{black } 256, \text{blue } 60, \text{blue } 56, \text{blue } 81\}, a_{1,2} = \{\text{white } 405, \text{yellow } 13, \text{red } 41, \text{blue } 56, \text{yellow } 31, \text{blue } 96\}$ $a_{1,2,3} = \{\text{white } 3, \text{black } 16, \text{blue } 13, \text{blue } 10\}$										

セル数が等しい3または4種類の正答であり，出力は，その正答のセル面振り分け問題の解と，必要があれば色合いを調整した後の正答である．今回実装したシステムには，次の特徴がある．

- 正答数が3であれば， $O(N)$ の計算量の判定式を用いて，セル面振り分け問題の解を持つか判定し，解を持つ場合は解を出力する．
- 4正答以上であれば， $O(2^{MN})$ の計算量の全探索を用いて，セル面振り分け問題の解を得る．
- 色の調整には，類似する2色を中間色に置き換える．

このシステムを用いて，実際に複数のパズルを作成できること，それが十分に考えながら解くことができるものであることを確認した．しかしながらセル面振り分け問題が解を持たない場合の色の調整方法に関しては，その評価方法も含めたより厳密な議論が必要と考えられる．また，4正答以上の場合には，セル面振り分け問題がNP完全であるかもしれないが，本論文では示せていない．

パズルとしての面白さを向上させるためには，次のような課題があげられる．

- ペントミノのように，複数のセルが連結したピースの導入．
- パズルの難易度の定量化と難易度の調整方法の開発．

本研究を通して，新しいタイプの組合せパズルの提案を行うことができた．またそれが興味深い問題を内包していることを確認できた．

参考文献

- [1] Abbott, T.G., Abel, Z., Charlton, D., Demaine, E.D., Demaine, M.L. and Kominers, S.D.: Hinged dissections exist, *Proc. 24th Annual Symposium on Computational Geometry*, pp.110–119 (2008).
- [2] 安倍泰孝, 原口和也, 丸岡 章: 計算ブロックパズルの生成アルゴリズム, 情報処理学会研究報告, GI, [ゲーム情報学], Vol.2011-GI-25, No.6, pp.1–7 (2011).
- [3] Alejandro, E. and Frank, R.: Domino Tatami Covering Is NP-Complete, *24th International Workshop, IWOCA 2013*, No.10, pp.140–149 (2013).
- [4] Czyzowicz, J., Kranakis, E. and Urrutia, J.: Rectilinear glass-cut dissections of rectangles to squares, *Applied Mathematical Sciences*, Vol.1, No.52, pp.2593–2600 (2007).
- [5] Gerard: Polyomino Solver, available from (<http://gp.home.xs4all.nl/Site/Polyomino.Solver.html>) (accessed 2014-10-01).
- [6] Golomb, S.W.: *Checkerboards and Polyominoes*, *The American Mathematical Monthly* 61, 10 (Dec), pp.287–294 (1954).
- [7] Golomb, S.W.: *Polyominoes: Puzzles, Patterns, Problems, and Packings*, Princeton University Press, Revised edition (1994).
- [8] 五十嵐力, 但馬康宏, 小谷善行: 裁ち合わせパズルの重複解なし全探索システム, ゲームプログラミングワークショップ 2006 論文集, pp.40–47 (2006).
- [9] 小谷義行: 図形合同分割パズルの自動生成, 情報処理学会研究報告, GI, [ゲーム情報学], Vol.2005-GI-014, No.87,

- pp.47–54 (2005).
- [10] 小谷義行: ゲーム情報学: 2. ゲーム情報学におけるパズル研究, 情報処理, Vol.53, No.2, pp.101–111 (2012).
- [11] 是川 空, 小谷善行, 柴原一友, 五十嵐力: BitsPuzzleの解答作成と問題作成, ゲームプログラミングワークショップ 2006 論文集, pp.187–190 (2006).
- [12] Lo, K.-Y., Fu, C.-W. and Li, H.: 3D polyomino puzzle, *ACM Trans. Graph.*, Vol.28, No.5, pp.157:1–157:8 (2009).
- [13] 村井保之, 巽 久行, 徳増真司: 位相的特徴量に基づく平面ポリオミノ箱詰め問題の解法, 情報処理学会論文誌, Vol.43, No.12, pp.4009–4022 (2002).
- [14] 西岡 潤, 堀山貴史: pmg タイリング可能なポリオミノの列挙, 研究報告アルゴリズム (AL), Vol.2014-AL-148, No.18, pp.1–8 (2014).
- [15] Rover, A.: Burr Tools, available from (<http://burrtools.sourceforge.net/>) (accessed 2014-10-01).
- [16] Song, P., Fu, C.-W. and Cohen-Or, D.: Recursive interlocking puzzles, *ACM Trans. Graph.*, Vol.31, No.6, pp.128:1–128:10 (2012).
- [17] 株式会社テンヨー: ジガゾーパズル, 入手先 (<http://www.tenyo.co.jp/jigazo/>) (参照 2014-10-01).
- [18] Xin, S.-Q., Lai, C.-F., Fu, C.-W., Wong, T.-T., He, Y. and Cohen-Or, D.: Making burr puzzles from 3D models, *ACM Trans. Graph.*, Vol.30, No.4, pp.97:1–97:8 (2011).
- [19] Zhou, Y. and Wang, R.: An algorithm for creating geometric dissection puzzles, *Proc. Bridges 2012: Mathematics, Music, Art, Architecture, Culture*, pp.49–56 (2012).
- [20] Zhou, Y., Sueda, S., Matusik, W. and Shamir, A.: Box-elization: Folding 3D Objects Into Boxes, *ACM Trans. Graph.*, Vol.33, No.4 (2014).



山本 陽平

2011年筑波大学情報学群情報科学類卒業，2015年，同大学大学院システム情報工学研究科コンピュータサイエンス専攻修了．修士（工学）．同年株式会社技研製作所入社．パズルや折り紙の理論に関する研究を行っている．



金森 由博 (正会員)

2009年3月東京大学情報理工学系研究科コンピュータ科学専攻博士課程修了．博士（情報理工学）．同年4月より筑波大学に勤務し，現職は筑波大学システム情報系・助教．コンピュータグラフィクス，特にレンダリング技術に興味を持つ．現実世界の現象を再現する画像編集技術や，イラストやアニメの制作支援技術に取り組んでいる．ACM，画像電子学会，芸術科学会各会員．



三谷 純 (正会員)

2004年東京大学工学部精密機械工学科博士課程修了。博士(工学)。理化学研究所研究員を経て、2006年より筑波大学に勤務し、現職は筑波大学システム情報系教授。コンピュータグラフィックスにおける形状モデリングの

研究に従事。最近は折り紙で作れる幾何形状の設計技法に関する研究等を行っている。