

# カラーダイスパズルを利用した $n$ -モード行列展開の 理解支援アプリの試作

山本 直樹<sup>1,\*</sup> 石田 明男<sup>2</sup> 村上 純<sup>1</sup> 大石 信弘<sup>3</sup>

## Prototype of Application to Support Understanding of $N$ -mode Matrix Unfolding by Using Color Dice Puzzle

Naoki Yamamoto<sup>1,\*</sup>, Akio Ishida<sup>2</sup>, Jun Murakami<sup>1</sup>, Nobuhiro Oishi<sup>3</sup>

Up to now we have engaged in research on the development of a system to support the understanding of multi-dimensional data processing using 3-d puzzles. In this paper, we picked up a color dice puzzle as the 3-d puzzle and developed an application for use as teaching materials. Furthermore, we tried this application on open-campus event day.

キーワード：立体パズル、行列展開、HOSVD、理解支援システム

Keywords : 3-d puzzle, Matrix unfolding, HOSVD, Understanding support system

### 1. はじめに

我々は、これまで多次元データ処理のアルゴリズムの開発や多次元データ処理を応用したデータ分析に関する研究に携わってきた。

多次元データ処理はアルゴリズムが複雑なことから、近年我々は、その視覚化や理解支援にも興味を持って取り組んでいる。主な取り組みとしては、特にテンソル分解手法の1つとしてよく利用される高次特異値分解 (HOSVD)<sup>(1)</sup>を取り上げ、そのアルゴリズムのCGによる視覚化の研究<sup>(2)</sup>、立体パズルを利用したアルゴリズムの理解支援の研究<sup>(3),(4)</sup>などを行っている。後者の研究については、現在教材となるパズル収集が課題の1つであり、今回、立体パズルの1種であるカラーダイスパズル<sup>(5)</sup>を新たに収集し、これが多次元データ処理の理解支援に適用できるのではないかと考えた。

そこで、本研究では、カラーダイスパズルを利用して、HOSVD アルゴリズムの理解支援に役立つアプリを試作し、その利用方法について検討したので報告する。

### 2. カラーダイスパズル

#### 2.1 カラーダイスパズルの作成

本論文では、図1に示すカラーダイスパズルを利用する。このパズルは、Instant Insanity、キューブ4などの名称で市販されているが、ここでは、木製の立方体と色シールを使って自作した。パズルは、4個の木製の立方体の各6面に赤、黄、緑、青の4色のシールを貼り、配色は文献(5)を参考にし、図2の展開図に示す配色のパズルピースを作成した。



図1 カラーダイスパズル

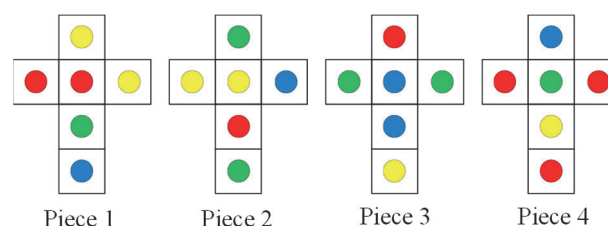


図2 パズルピースの配色

<sup>1</sup> 人間情報システム工学科  
〒861-1102 熊本県合志市須屋 2659-2  
Dept. of Human-Oriented Information Systems Engineering,  
2659-2 Suya, Koshi-shi, Kumamoto, Japan 861-1102

<sup>2</sup> 共通教育科  
〒861-1102 熊本県合志市須屋 2659-2  
Faculty of Liberal Studies,  
2659-2 Suya, Koshi-shi, Kumamoto, Japan 861-1102

<sup>3</sup> 情報通信エレクトロニクス工学科  
〒861-1102 熊本県合志市須屋 2659-2  
Dept. of Information, Communication and Electric Engineering,  
2659-2 Suya, Koshi-shi, Kumamoto, Japan 861-1102

\* Corresponding author:  
E-mail address: naoki@kumamoto-nct.ac.jp (N. Yamamoto).

## 2.2 パズルの揃え方とパズルの解

このパズルは、各ピースを自由に入れ替えて、図1のように横1列の柱状に並び、4つの側面がいずれも、4色の異なる色の並びとなるように揃えるものである。

ここで、図1 (および図2) の配置は、カラーダイスパズルの解となっている。図1で見える上面と前面について説明すると、図2の展開図において、各ピースの十字の交差部分にある色が図1の各ピースの上面の色に対応し、十字の交差部分から1つ下の色が図1の前面の色に対応している。また、図1では隠れて見えないが、図2の1番上と1番下部分の色は、図1の後面と下面の色にそれぞれ対応している。これより、図2に示すピース1から4を順番に横1列に並べると、側面の色がどれも4色になるように揃えられ、パズルの解となっていることが分かる。

このパズルの解については、各ピースの配置により複数の誘導解が存在するが、得られた誘導解からピースの入れ替え (柱全体の回転、各ピースを縦軸回りに180度回転、各ピースの位置を移動など) をすることにより、結局は、図2の展開図に示される1つの基本解に集約される<sup>(6)</sup>。

なお、本論文では、パズルを揃える際に、図2に示すピース1~4を左から右に順に並べるものとする。本来は、各ピースを並べる位置を固定する必要はないが、このようにするのは、各ピースと後述する2次元的なパズルのマップとの空間的な配色の関係を把握しやすくさせるためである。

## 3. HOSVD と $n$ -モード行列展開

### 3.1 HOSVD の概要とアルゴリズム

高次特異値分解 (HOSVD) は、行列の特異値分解 (SVD) を3階以上の高階テンソルに拡張したもので、画像処理、パターン認識、データ解析などに応用される分解手法である。高階テンソルとは、多次元配列のことを指し、例を挙げると、1階テンソルはベクトル、2階テンソルは行列、3階テンソルは3次元配列に対応する。

HOSVD の例として、図3に3階テンソルの場合における分解のイメージを示す。元の3階テンソル  $\underline{A}$  は、1つのコアテンソル  $\underline{B}$  (SVD の対角行列に対応) と3つの直交行列  $\underline{U}^{(n)}, (n = 1, 2, 3)$  の  $n$ -モード積 (テンソルと行列の積で、1-モード積は  $\times_1$  と表す) に分解される。次に、 $M$  階テンソルに一般化された HOSVD のアルゴリズムを示す。

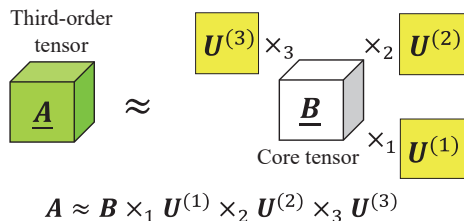


図3 3階テンソルの HOSVD

### [(アルゴリズム 1) $M$ 階テンソルの HOSVD]

入力 :  $M$  階テンソル  $\underline{A}$

出力 :  $M$  個の直交行列  $\underline{U}^{(n)}, (n = 1, 2, \dots, M)$  およびコアテンソル  $\underline{B}$

(ステップ 1)  $M$  階テンソル  $\underline{A}$  を  $n$ -モード行列展開して、 $M$  通りの行列  $\underline{A}_{(n)}, (n = 1, 2, \dots, M)$  を求める。

(ステップ 2) ステップ 1 で得られた  $\underline{A}_{(n)}$  に SVD をそれぞれ適用して、 $M$  個の左特異行列 (直交行列)  $\underline{U}^{(n)}, (n = 1, 2, \dots, M)$  を計算する。

(ステップ 3) コアテンソル  $\underline{B}$  を次式により計算する。

$$\underline{B} \leftarrow \underline{A} \times_1 \underline{U}^{(1)T} \times_2 \underline{U}^{(2)T} \dots \times_M \underline{U}^{(M)T}$$

(ステップ 4)  $\underline{U}^{(n)}, (n = 1, 2, \dots, M)$  および  $\underline{B}$  を返す。

(アルゴリズム終わり)

### 3.2 $n$ -モード行列展開の定義とアルゴリズム

前節で述べた HOSVD のアルゴリズムにおけるステップ 1 の  $n$ -モード行列展開の定義<sup>(1),(4)</sup> を次に示す。

#### [(定義) $n$ -モード行列展開]

サイズ  $I_1 \times I_2 \times \dots \times I_M$  の  $M$  階テンソル  $\underline{A}$  の  $(i_1, i_2, \dots, i_M)$  成分  $a_{i_1 i_2 \dots i_M}$  を  $(i_n, j_n)$  成分とする  $I_n \times (I_{n+1} I_{n+2} \dots I_M I_1 I_2 \dots I_{n-1})$  行列  $\underline{A}_{(n)}$  を求めることを  $n$ -モード行列展開という。ただし、 $j_n$  は次式で与えられる。

$$j_n = (i_{n+1} - 1) I_{n+2} I_{n+3} \dots I_M I_1 I_2 \dots I_{n-1} + (i_{n+2} - 1) I_{n+3} I_{n+4} \dots I_M I_1 I_2 \dots I_{n-1} + \dots + (i_M - 1) I_1 I_2 \dots I_{n-1} + (i_1 - 1) I_2 I_3 \dots I_{n-1} + (i_2 - 1) I_3 I_4 \dots I_{n-1} + \dots + i_{n-1}$$

(定義終わり)

$n$ -モード行列展開は、簡単に言えば、 $M$  階テンソルを行列に変換する操作のことである。ここで、モードとは高階テンソルの方向を指し、図4上のサイズ  $I_1 \times I_2 \times I_3$  の3階テンソル  $\underline{A}$  では、 $\underline{A}$  の縦方向が1-モード、横方向が2-モード、奥方向が3-モードに対応する。いま、 $\underline{A}$  の  $(i_1, i_2, i_3)$  成分を  $a_{i_1 i_2 i_3}$  とすると、これらの添字とモードの対応は、添字  $i_1$  が1-モード

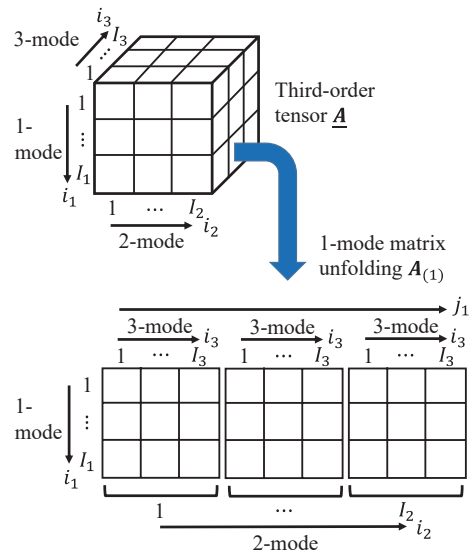


図4 3階テンソルの1-モード行列展開

を、 $i_2$ および $i_3$ が2-モードおよび3-モードをそれぞれ表す。

図4の3階テンソル $\underline{A}$ の1-モード行列展開は、上述の定義より、 $\underline{A}$ の $(i_1, i_2, i_3)$ 成分 $a_{i_1 i_2 i_3}$ が行列 $\mathbf{A}_{(1)}$ の $i_1$ 行 $j_1 = (i_2 - 1)I_3 + i_3$ 列に配置されるので、これを $i_1 = 1, \dots, I_1, i_2 = 1, \dots, I_2, i_3 = 1, \dots, I_3$ のすべての場合について配置することで図4下のように求められる。他のモード行列展開についても、これと同様にして求めることができる。

行列展開を求めるには、前述の定義に従い、高階テンソルの各成分を行列展開の各成分に1つずつ移せばよいが、実装上は、より効率的な方法がある。それは、図4の場合で述べると、 $\underline{A}$ の2-モードに沿って、 $\underline{A}$ を左から右に行列に輪切りにして、それぞれ得られる行列を左から右に横に並べることで、1-モード行列展開を求めることができる。以下に、その効率的な行列展開のアルゴリズムを示す。

[(アルゴリズム2) 3階テンソルの1-モード行列展開]

入力： サイズ $I_1 \times I_2 \times I_3$ の3階テンソル $\underline{A}$

出力： サイズ $I_1 \times I_2 I_3$ の1-モード行列展開 $\mathbf{A}_{(1)}$

(ステップ1)  $\underline{A}$ から部分行列 $\mathbf{A}_{i_2} = (a_{*i_2*})$ , ( $i_2 = 1, \dots, I_2$ )を取り出し、横に並べて $\mathbf{A}_{(1)} = (\mathbf{A}_{i_2} | \mathbf{A}_{i_2} | \dots | \mathbf{A}_{i_2})$ とする。

ただし、 $(a_{*i_2*})$ は、 $i_2$ を固定した $i_1 = 1, \dots, I_1, i_3 = 1, \dots, I_3$ の行列を表す。

(ステップ2)  $\mathbf{A}_{(1)}$ を返す。

(アルゴリズム終わり)

4. カラーダイスパズルの高階テンソル表現と行列展開

4.1 パズルの高階テンソル表現

図1および図2のパズルを高階テンソルで表現するために図5のように考えることにした。いま、図5上に示すサイズ $1 \times 1 \times 1$ の各ピースを、図5下のようなサイズ $3 \times 3 \times 3$ の3階テンソルにそれぞれ拡大する。そして、図2に示される各ピースの配色が、拡大された3階テンソルの各側面の中央に塗られるようにする。したがって、各側面の中央の成分には元のピースの色の値が格納される。さらに、パズル全体は、図5上に示すようにピース1から4が左から順に4つ横に並べられるため、図5下に示すような拡大された3階テンソルとして構成する。

4.2 パズルの行列展開による表現

本研究では、高階テンソルの $n$ -モード行列展開について学生に理解を深めさせるために、図1のカラーダイスパズルと、図5下に示した4階テンソルを1-モード行列展開して得られる2次元的なパズルのマップを利用する。ここで、そのマップを作成するアルゴリズムを次に示す。

[(アルゴリズム3) パズルのマップ作成]

入力： 図5下に示すようなパズルの配色情報を持つサイズ $3 \times 3 \times 3 \times 4$ の4階テンソル $\underline{C}$

出力：  $\underline{C}$ の1-モード行列展開 $\mathbf{C}_{(1)}$

(ステップ1)  $\underline{C}$ から各ピースの3階テンソル $\underline{C}_{i_4} = (c_{***i_4})$ , ( $i_4 = 1, 2, 3, 4$ )を取り出す。ただし、 $(c_{***i_4})$ は、 $i_4$ を固

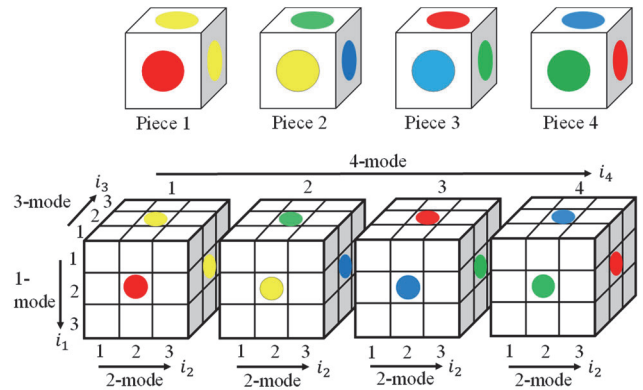


図5 パズルの高階テンソル表現

定した $i_1, i_2, i_3 = 1, 2, 3$ の3階テンソルを表す。

(ステップ2)  $\underline{C}_{i_4}$ , ( $i_4 = 1, 2, 3, 4$ )にそれぞれアルゴリズム2を適用して、1-モード行列展開 $\mathbf{C}_{i_4(1)}$ , ( $i_4 = 1, 2, 3, 4$ )を求める。

(ステップ3)  $\mathbf{C}_{i_4(1)}$ , ( $i_4 = 1, 2, 3, 4$ )を横に並べて、 $\mathbf{C}_{(1)} = (\mathbf{C}_{1(1)} | \mathbf{C}_{2(1)} | \mathbf{C}_{3(1)} | \mathbf{C}_{4(1)})$ とする。

(ステップ4)  $\mathbf{C}_{(1)}$ を返す。

(アルゴリズム終わり)

上述のアルゴリズム3を適用して得られた $\mathbf{C}_{(1)}$ におけるピース1の部分のマップ例(イメージ図)を図6に示す。図6は、図5下の $i_4 = 1$ (ピース1)の3階テンソルを2-モードに沿って、左から右に行列に輪切りにして、得られた行列を横に並べたものになっている。したがって、一番左の行列には、元のピース1の左面の色(赤)が現れ、左から2番目の行列には、上・下・前・後面の色(それぞれ、黄・緑・赤・青)が、一番右の行列には右面の色(黄)が現れる。パズル全体のマップは、図6に示されるようなマップがピース順に並べられる。

図6はピース1の1-モード行列展開になっているが、我々は、図6のようなマップを確認させながら、実際のパズルピースを配置させることで、モード行列展開の理解支援に繋がるのではないかと考え、これをアプリの実装に利用した。

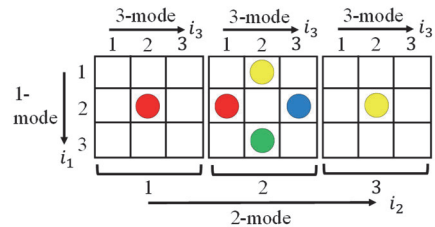


図6 ピース1のマップ例 (4-mode:  $i_4 = 1$ )

5.  $n$ -モード行列展開の理解支援アプリ

本論文のアプリの開発は $\mathbb{R}^7$ を利用して実装した。 $\mathbb{R}$ で高階テンソルを扱うためにはパッケージrTensorを導入し、パズルのマップ表示に関数 `unfold` および関数 `image` を利用した。このアプリは平成30年8月に本校で開催されたオープ

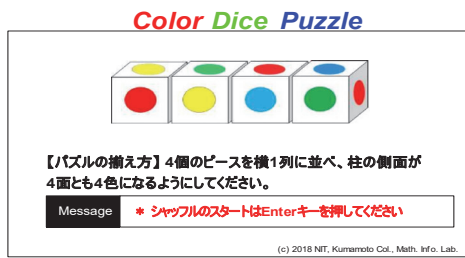


図7 パズルの解き方の説明

ンキャンパスの研究室展示で試用してみたので、このときの利用方法を以下に説明する。

まず、アプリを起動すると、図7に示すパズルの解き方の説明がパソコンの画面上に表示される。ここで、説明者がパズルの揃え方の説明を利用者に対して行う。その際に実際のパズルも併用して、ピースを横に並べながら解説する。

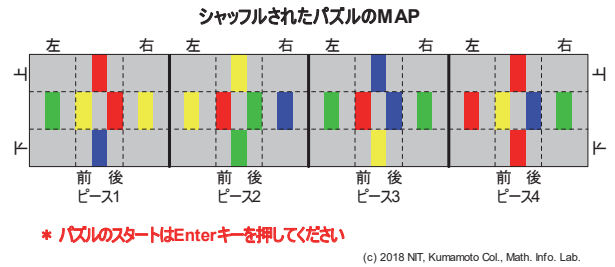
次に、Enterキーを入力すると、パズルがシャッフルされ、シャッフルされたパズルのマップが図8(a)のように表示される。説明者は、利用者に対してマップの見方を説明するために、ピース1のマップの見方を、実際のパズルを使い、配置してみせながら説明する。その後、利用者に残りのピース2から4のマップを確認させながら、実際のパズルピースを机の上に配置させる。

さらにEnterキーを入力すると、図8(b)に示すように、コンピュータはパズルを解き始め、マップの色がリアルタイムで変わっていくので、キーを入力する前に、説明者は、利用者にも実際のパズルを解くよう促して、コンピュータと対戦させる。コンピュータが最長2万手までに解を求められれば、図8(c)のように答えのマップとして表示するが、解を求められなかった場合でも、解の1例を図8(c)と同様にして表示する。もし、利用者がタイムアップ時に実際のパズルを解けていないなら、このマップを見ながら実際のパズルを配置させ、答えを確認させる。

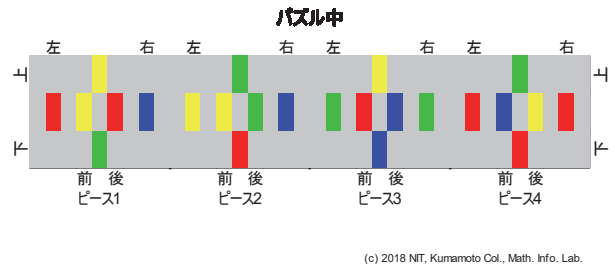
以上のように、アプリを実行して、マップを繰り返し参照させながら、実際のピースを配置することで、3階テンソルの1-モード行列展開を反復して学習する効果があるものと考えている。他のモード行列展開についても、1-モードの場合と同様にアプリに実装可能である。

## 6. まとめ

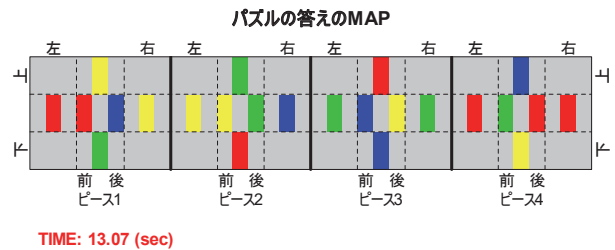
本研究では、カラーダイスパズルを用いて、図2のような従来の展開図とは異なった行列展開を利用した展開図表現を考案し、HOSVDの $n$ -モード行列展開の理解支援のためのアプリの実装を行った。開発されたアプリは、上述のとおり、今年度のオープンキャンパスで試用し、研究室展示に訪れた43名(小学生:1名、中学生:31名、教員・高専生・一般:11名)の利用者に図8に示したマップが理解できたか質問したところ、「理解できた」が29名(67%)、「理解しづらかった・できなかった」が14名(33%)であった。また、マップの表現の仕方に関する複数の改善意見もあった。



(a) シャッフルされたパズルのマップ



(b) 求解途中のパズルのマップ



(c) パズルの解のマップ

図8 パズルのマップ表示

今後の課題として、マップ表示やアプリの利用方法を改善し、より学習しやすいアプリにしていきたいと考えている。

(平成30年9月25日受付)

(平成30年12月5日受理)

## 参考文献

- (1) L. De Lathauwer et al.: "A Multilinear Singular Value Decomposition", SIAM Journal on Matrix Analysis and Applications, Vol.21, No.4, pp.1253-1278 (2000).
- (2) 大隈千春, 長岡翔, 村上純, 山本直樹, 石田明男: 「計算過程の可視化による高次特異値分解の理解支援システムの開発」, 高専教育, Vol.38, pp.129-134 (2015).
- (3) 山本直樹, 石田明男, 平田将大, 村上純: 「立体パズルを利用した多次元データ分解手法の理解支援の試み」, 熊本高専研究紀要, Vol.9, pp.67-74 (2018).
- (4) 石田明男, 山本直樹, 大石信弘, 村上純: 「多次元データ分解の手法を用いた立体パズルの解法」, 初等数学, Vol.83, pp.18-22 (2018).
- (5) 高木茂男: 「PLAY PUZZLE パズルの百科」, pp.41-42, 平凡社 (1981).
- (6) パズルショップ・トリト: 「3-17. インスタント・インサニティ」, パズル遊びへの招待・オンライン版, <http://www.torito.jp/puzzles/317.shtml>, 2018年8月22日アクセス.
- (7) R Core Team: "R: A language and environment for statistical computing", R Foundation for Statistical Computing, Vienna, Austria.