

# 自己相似図形を利用した次元変換による多次元データの多解像度可視化手法

96725 藤原 孝紀  
指導教員 岡本 孝司 教授

Multi-dimensional and large scale data are difficult to understand because of their spatial complexity and large data size. Visualization is useful tool for supporting the understanding those data, but there is no effective method for visualizing those data, particularly multi-dimensional data. In this study, we developed visualization method for multi-dimensional and large scale data. Our method can transform the dimensions of the data and visualize the data as a 1D, 2D and 3D image by using self-similar object and recursive mapping. Using self-similar object and recursive mapping can also give multi-resolutional and abstractive result. With these features, our method supports understanding of complicated multi-dimensional and large scale data.

Key words: Multiple dimensional data, Self-similar object, Multi-resolution, Dimensional transformation

## 1 緒言

近年、測定技術の向上、計算機の処理速度の向上や研究対象の多様化により、測定やコンピュータシミュレーションなどで得られるデータが多次元化・大規模化してきている。こうした多次元、大規模なデータはその形状の複雑さやデータ数の多さのために、データが示す形状や傾向の理解が困難となっており、理解を補助する可視化手法が必要とされている。

岩丸は、ボリュームデータの理解を補助する手段として、自己相似図形を用いたボリュームデータの2次元化多解像度可視化手法を開発した<sup>1)</sup>。本研究では、この手法を発展・一般化させ、より多様な次元のデータを高次元化・低次元化、さらに多解像度化できる手法を開発する<sup>2) 3)</sup>。提案手法により高次元のデータが低次元化により表示可能になる、大規模なデータを高次元化により画面使用率の高い状態で表示できる、多解像度化により複雑な現象の要約的な可視化が可能になる、といったことが実現される。また、実際に提案手法を用いた可視化事例も示す。

## 2 多解像度化可能な低次元化可視化手法

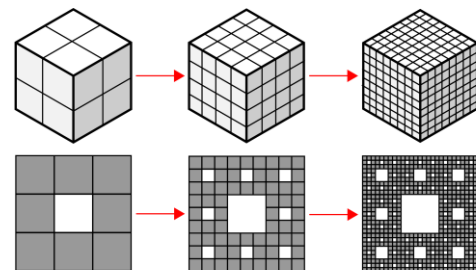
### 2.1 データの次元・解像度と要素数

次元数  $n$  のデータが解像度  $R$  を持つとき、データを構成する要素数は  $2^{nR}$  個となる。ここでは、解像度  $R$  とは各次元軸方向にデータが  $2^R$  個分割されていることを示すこととする。 $n=3$  のデータの例を Fig.1 上部に示す。

### 2.2 再帰的分割によってできる自己相似図形

シェルピンスキーのカーペット等に代表される再帰的な分割ルールにより作成される自己相似な図形は、再帰的なルールの適用回数により図形の詳細さが変わる。ここで、再帰的な分割ルールを  $R$  回適用した自己相似な図形の解像度を  $R$  と定めることとする。例えば、シェルピンスキーのカーペットは解像度 1 のとき要素数 8 を持ち、解像度  $R$  のとき要素数は  $8^R$  となる (Fig.1 下)。一般に解像度 1 のとき要素数  $a$  の自己相似図形は、解像度  $R$  のとき要素数  $a^R$  となる。また、前節 2.1 で挙げた各次元軸方向に  $2^R$  分割された図形も、 $2^{R-1}$  分割された図形の各要素を各次元軸方向に 2 分割することで得られることから、再帰的な分割によってできる自己相似な図形であると考えられる。

さらに、自己相似図形は局所的に再帰的な分割ルールを適用することが可能であり、Fig.3 や Fig.5 のように一部分だけ高解像度化することや低解像度のまま残して置くことが可能である。そのため、自己相似図形は多解像度表示が可能な図形である。



Resolution: 1                      2                      3  
Elements: 8                      64                      512

Fig. 1 Top: 3D data and Bottom: Sierpinski carpets with resolutions 1, 2 and 3.

### 2.3 高次元データの1次元化可視化手法

解像度が  $R$  のとき、次元数が  $n_1 (n_1 \geq 1)$  の高次元データは要素数  $2^{n_1 R}$  であるため、解像度が 1 のとき要素数が  $2^{n_1}$  であるような 1 次元自己相似図形を作成し、再帰的な分割ルールとマッピングルールによって高次元データを 1 次元図形上にマッピングすればよい。1 次元図形は、単純に 2 分割を再帰的に行うと、解像度  $n_1$  を持つときに要素数が  $2^{n_1}$  となる。そこで、この要素数  $2^{n_1}$  の 1 次元図形と次元数  $n_1$  のデータの間解像度 1 のマッピングルールを決めて再帰的に分割することによって、高次元データの 1 次元化を行うことができる。Fig.2 に 3 次元データを 1 次元化する際のマッピングルールの例を挙げる。

### 2.4 高次元データの2次元化可視化手法

解像度  $R$ 、次元数  $n_2 (n_2 \geq 2)$  の高次元データは要素数  $2^{n_2 R}$  であるため、解像度 1 のとき要素数が  $2^{n_2}$  であるような 2 次元自己相似図形を作成し、再帰的に分割・マッピングを行えばよい。正方形を単純に縦横 2 分割にしていると、解像度があがるたびに要素数が 4 倍になっていくため、要素数  $2^{n_2}$  の 2 次元図形を作成するには、場合分けが生じる。要素数  $2^{n_2 R}$  は、次元数が奇数・偶数の場合で、

式(1)のように変形できる(ただし,  $k$  は自然数). 式(1)より, 偶数次元の場合は, 正方形を縦横方向に繰り返し 2 分割を行ってできる自己相似図形の解像度  $k$  のもの, 奇数次元の場合は, 解像度 1 のとき 8 要素を持つ 2 次元自己相似図形の各要素を縦横ともに  $(k-1)$  回分割したものを利用すればよい. 8 要素を持つ自己相似図形として, 解像度 1 のシェルピンスキーのカーペットが挙げられる. Fig.3 に 2 から 5 次元に利用する 2 次元図形の一部だけ解像度を上げた例を示し, Fig.4 にマッピングルール例を載せる.

$$2^{n_2 R} = \begin{cases} 4^{kR} & (n_2 = 2k) \\ (8 \cdot 4^{k-1})^R & (n_2 = 2k + 1) \end{cases} \quad (1)$$

### 2.5 高次元データの 3 次元化可視化手法

解像度  $R$ , 次元数  $n_3 (n_3 \geq 3)$  の高次元データは要素数

$2^{n_3 R}$  であり, 解像度 1 のとき要素数が  $2^{n_3}$  であるような 3 次元図形を作成し, 再帰的に分割・マッピングを行えばよいが, 2 次元化のときと同様に場合分けが必要となってくる.  $n_3 = 3k, 3k+1, 3k+2$  ( $k$  は自然数)のそれぞれの場合に

よって, 式(2)のように表すことができる. 式(2)より,  $n = 3k$  のときは, 解像度  $k$  の単純に縦横高さに 2 分割を繰り返してできる立方体,  $n = 3k + 1$  のときは解像度 1 のとき 16 要素を持つ 3 次元自己相似図形の各要素を縦横高さ共に  $(k-1)$  回分割した 3 次元図形,  $n = 3k + 2$  のときは解像度 1 のとき 32 要素を持つ 3 次元自己相似図形の各要素を縦横高さ共に  $(k-1)$  回分割した 3 次元図形を利用すればよい. 解像度 1 のとき 8 要素, 16 要素, 32 要素を持つ 3 次元自己相似図形の一部だけ解像度を挙げた例を Fig.5 に示す. 要素数の一致する図形が得られたので, 解像度 1 のときの高次元データから 3 次元図形へのマッピングルールを定めれば, 3 次元化が可能となる.

$$2^{n_3 R} = \begin{cases} 8^{kR} & (n_3 = 3k) \\ (16 \cdot 8^{k-1})^R & (n_3 = 3k + 1) \\ (32 \cdot 8^{k-1})^R & (n_3 = 3k + 2) \end{cases} \quad (2)$$

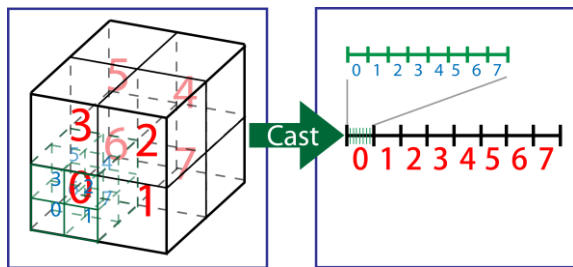


Fig. 2 A mapping rule example of dimensional reduction for high-dimensional data (3D to 1D).

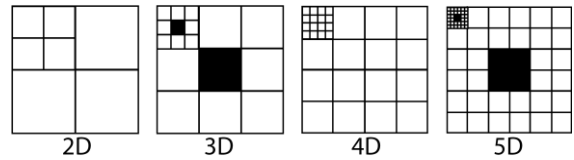


Fig. 3 Self-similar objects for 2D, 3D, 4D and 5D. Each object has 4, 8, 16, 32 elements for resolution 1.

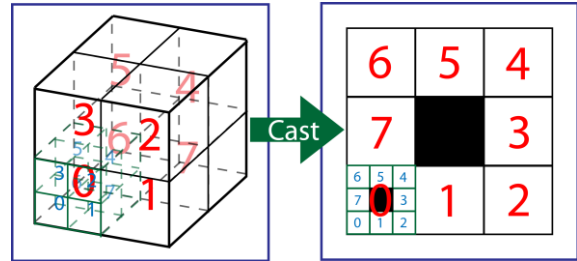


Fig.4 A mapping rule example of dimensional reduction for high-dimensional data (3D to 2D).



Elements:  $8^R$   $16^R$   $32^R$

Fig. 5 3D self-similar objects.

## 3 多解像度化可能な高次元化可視化手法

### 3.1 1次元データの 2次元化可視化手法

解像度が  $R$  のとき, 1 次元データの要素数は  $2^R$  である. 一方, 2 次元図形のうち解像度 1 のときの要素数が最小な自己相似図形は要素数 4 のものである. そのため, 1 次元データの 4 要素を最小単位として, 2 次元化する必要がある. Fig.6 のような Morton の並びに 4 要素を並べれば, Morton の並び自身が自己相似な形となっているため, 1 次元データの多解像度化可能な 2 次元化可視化が実現する(Fig.7). ここでは Morton の並びを例としてあげたが, 他のマッピングルールであっても高次元化は可能である.

### 3.2 1次元データの 3次元化可視化手法

3 次元図形のうち立方体を各要素の形として持ち, 解像度 1 のときの要素数が最小な自己相似図形は要素数 8 であるため, 1 次元データの 8 要素を最小単位として, 3 次元化すればよい. 3 次元 Morton の並び上に 8 要素を並べるマッピングルールを定めれば, 1 次元データの多解像度化可能な 3 次元可視化が実現する. ここでも 3 次元 Morton の並び以外のマッピング方法も考えられる.

### 3.3 2次元データの 3次元化可視化手法

解像度が  $R$  のとき, 2 次元データは要素数  $4^R$  である. そのため, 2 次元データの要素数 64 を最小単位として, 解像度 2 の単純分割立方体上に写像する(Fig.8 上)か, 2 次元データの要素数 16 を最小単位として, Fig.5 に挙げ

た 16 要素を基本要素数とする 3 次元フラクタル図形上に写像する(Fig.8 下)ことで 3 次元化が行える。

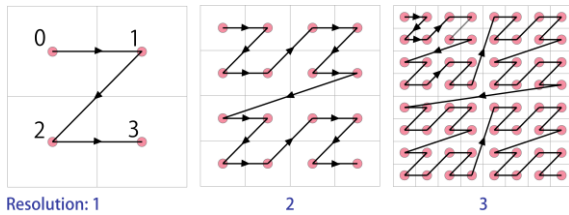


Fig. 6 Morton orders with resolutions 1, 2 and 3.

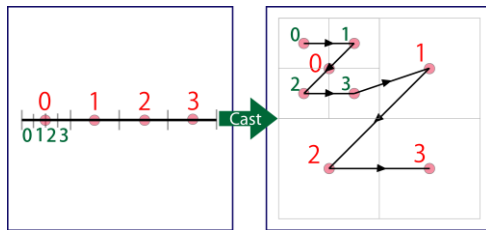


Fig. 7 A mapping rule example of dimensional increase (1D to 2D).

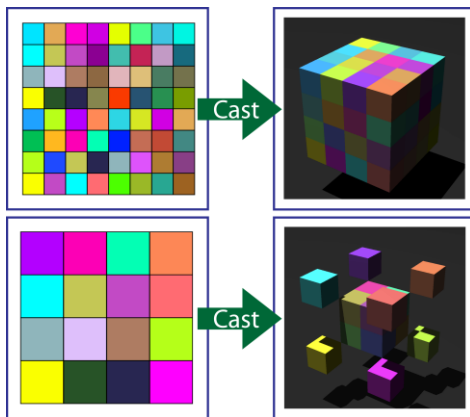


Fig. 8 Mapping rule examples (2D to 3D). 2D data with resolutions 3 (Top) and 2 (Bottom) to 3D self-similar object. The color areas in 2D are mapped onto the same color areas in 3D

#### 4 次元別の次元変換可視化手法

ある複数の次元を持つデータを次元の種類別に低次元化・高次元化したいような場合がある。例えば、時系列ボリュームデータは、空間の次元(3次元)と時間の次元(1次元)を併せ持つ 4 次元データである。4 次元すべてを同様に扱い低次元化を行うと、時間変化の把握が困難である、空間と時間が同等の解像度を持つ必要があるなどといった短所がある。そのため、このようなデータの場合、次元の種類別に次元変換を行うことが有効となる。

4 次元データの場合の例を挙げて、次元別の次元変換について説明する。はじめに、扱う次元の種類とその種類に所属する次元数を表示する方法として、次元種類が A, 次元数が  $n$  のとき、 $A(n)$  と表す方法を導入する。4 次元で構成されるデータの場合、 $A(4)$ ,  $A(3)+B(1)$  (例えば A: 空

間次元, B: 時間次元),  $A(2)+B(2)$ ,  $A(2)+B(1)+C(1)$ ,  $A(1)+B(1)+C(1)+D(1)$  といった 5 つの扱う方法がある。これらの場合について、次元変換を行い、3 次元以下の可視化像を得なければならない。例えば、 $A(3)+B(1)$  の場合、A 次元群を 2 次元化し、B 次元群方向に並べて 3 次元化、A 次元群を 1 次元化し、B 次元群方向に並べて 2 次元化、A 次元群を 1 次元化し、B 次元群を 2 次元化し 3 次元化するなどといったことが考えられる。

一般的な  $n$  次元データの場合、 $A(n)$ ,  $A(a)+B(n-a)$ ,  $A(a)+B(b)+C(n-a-b)$  の 3 種類の次元までであれば、各次元の次元変換を行い、各次元群の次元変換された結果の次元数の合計が 3 以下に収まるようにすることができ、次元別の次元変換が可能である。

#### 5 適用事例

本手法の適用事例として、時系列ボリュームデータの次元別次元変換を行った例を載せる。対象データは、自動車周りの流れの数値シミュレーションから得られた車体後部の圧力データである。8×8×8 の領域の空間次元が 128 ステップ時間変化したもの(合計 65536 要素)を可視化した。次元別の次元変換の方法として、空間次元を 1 次元化したものを時間方向に並べて 2 次元化することを選択した。Fig.9 に空間次元と時間次元の解像度 1 のときのマッピングルールを載せる。解像度 1 のときは、各次元方向に 2 分割ずつされているので、座標値が小さいほうを 0, 大きいほうを 1 として表している。このマッピングルールを用いて可視化を行った結果が Fig.10 である。空間次元の変化を表すのに横軸方向(S 方向), 時間次元の変化を表すのに縦軸方向(T 方向)を利用している。この図では、右に行くほど  $z$  の値が大きい領域になり、さらに  $z$  値で分けられる各領域の中において右に行くほど  $y$  の値が大きくなる。さらに、 $y$  値で分けられる各領域の中でも右に行くほど  $x$  の値が大きくなっている。各位置における  $x, y, z$  の値を把握しやすくするため、可視化像の上部にその S 方向の位置における  $x, y, z$  の値の大きさを青から赤の 8 段階の色で表示している(それぞれ  $x, y, z$  値 0 から 7 に対応する)。また、空間次元と時間次元の値の分散を考慮して多解像度化を行い、表示矩形数が多解像度化を行わない場合の約 4 分の 1 の 16542 領域に減少されている。この可視化結果より、1)  $z$  座標が 4 以上の領域では、データの値が小さなおおが多く、分散が少ない、2)  $z$  座標が 4 以上の領域でも大きな値を取るところがあり、そのような領域では常に大きな値を取り続ける、3)  $x$  座標が 3 以下かつ  $z$  座標が 3 以下の領域では値の分散が全体的に大きく、値の大小も時間変化に合わせて大きく変化している。さらに値の振動に似た挙動も見て取れる、4)  $z$  値が 2 から 3,  $y$  値が 2 から 3 を持つ領域で、 $x$  値が 0 から 1 の領域から 2 から 3 の領域へ変化しているところなどでは、同じ圧力値であると考えられる箇所が右上がりに移動していることが見て取れる。この右上がりの傾きによって  $x$  方向の圧力移動の速度も把握できる(傾きが急であるほど

速度は小さい). 同様に  $y$  値が変化している領域や  $z$  値が変化している領域を比較しても, 右上がりの変化が見られるところがあり, その傾きによって速度が把握できるため, 速度の値もある程度確認することができる, などのデータの複数の特徴や値を一画像から把握できる. このようにして, 次元別に次元変換を行い, 2次元可視化像を得ることで, 空間的・時間的遮蔽が存在せず, なおかつ時間変化を把握しやすい形で全体俯瞰可能な可視化像を得ることができる.

S	(0,0,0)	(1,0,0)	(0,1,0)	(1,1,0)	(0,0,1)	(1,0,1)	(0,1,1)	(1,1,1)
T	0				1			

Fig. 9 1D mapping rule for Top: spatial dimensions, Bottom: temporal dimension.

また, 本例以外にも 4次元以上の空間における現象のとして, 5次元拡散方程式に基づく拡散現象(時系列 5次元データ)の 3次元可視化, 4次元量子ウォークシミュレーション結果の 2次元可視化などを行い, 高次元空間の現象の可視化に本手法が有効であることを示した.

### 6 本手法の有する特徴

本手法は自己相似図形および再帰的なマッピングルールを用いた次元変換可視化手法であり, その手法の性質と可視化事例の結果から次のような長所を有すると言える.

1) あらゆる次元データを低次元化・高次元化できる. そのため, 低次元化によって, 高次元データの表示を可能にする. また, 低次元データの高次元化により, 画面使用率が高い可視化を可能にする. 2) 対象データの持つ各要素の空間的な階層構造を保持した可視化ができる. そのため, 単なるマッピングとは違い, 階層構造を把握しながらのデータの理解ができる. これは, データのおおまかな位置や傾向を把握することに適しているため, 複雑なデータを理解する際に役立つ. 3) 解像度 1 のときのマッピングルールを決めれば, 全解像度のマッピングができ, なおかつ結果像は解像度 1 のルールが持つ性質の傾向を持つ. そのため, マッピングルールの決定が容易であることや, 解像度 1 のときのルールを適切に決めておくことで, 効果的

な可視化結果を得ることができる. 4) 多解像度化が可能であり, 要約的な可視化ができる. この特徴により, 認知負荷を下げた可視化が可能となり, 3)と同様に複雑なデータの理解を容易にする. 5) 2次元以下の次元で可視化を行った場合, オクルージョンのない可視化結果を得ることができる. このことにより, ボリュームデータのようなオクルージョンが問題になるデータの理解を補助することができる. 6) 上記のような長所を併せ持つため, 任意の次元データを要約化された形で 2次元化でき, 対象データの全体俯瞰を可能にする. そのため, ボリュームレンダリングなどの他の可視化手法や解析手法と連携させることで, 高次元・大規模なデータの理解を効率的に行うことができる. 一方で, 次元変換を行うため, もとのデータが持つ空間的形狀が崩れて可視化され, その形状が把握しにくいといった短所がある. この問題は形状の表示に強みを持つ可視化手法と本手法を併用することで解決することができる.

### 7 結論

本研究では, 自己相似図形を利用した次元変換によって, 多様な次元のデータの多解像度可視化を行える手法を開発し, 更に適用事例を示した. 提案手法により任意次元のデータを 1, 2, 3次元表示できるようになり, 高次元データの可視化, オクルージョンのない可視化, 大規模データの効率的な画面配置, 要約的な可視化が行えるようになった. 提案手法を用いた可視化によって, 高次元・大規模データの全体俯瞰ができるため, 他の可視化・解析手法と本手法を併せて用いることでこれらのデータの効率的な理解ができるようになる.

### 文献

- 1) 岩丸雅紀:「フラクタル図形を利用した 3次元構造データのマルチレゾリューション可視化に関する研究」, 東京大学新領域創成科学研究科修士論文, 2009.
- 2) 藤原孝紀, 岩丸雅紀, 岡本孝司:「高次元データ可視化装置および方法ならびにプログラム」, 特願 2009-237504, 2009.
- 3) T. Fujiwara, et al.: "Fractal Map: Fractal-Based 2D Expansion Method for Multi-scale High-dimensional Data Visualization," *Advances in Visual Computing*, pp.306-315, 2010.

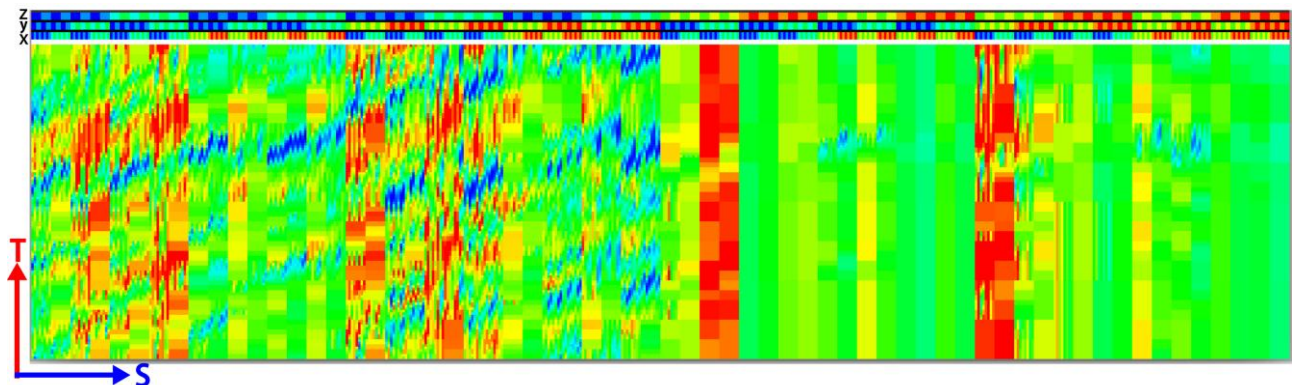


Fig. 10 Dimensional transfer visualization result of time-varying pressure volume data from numerical simulation. High pressure parts are colored by red and low pressure parts are colored by blue.