Java Applet による立体パズルの再現とその解法

和島　茂

要旨　カラーラベルのついたピースを三次元的に移動・回転させて元の状態に戻す立体パズルとしてはルービックキューブが有名である。本研究ではこれを初めとしたいくつかのパズルを Excel VBA, Java Applet, Visual C#で再現し、プログラムによる自動解法を開発した。

1.　序論

現在市販されている立体パズルには様々なものがあるが、パズル全体の形が正多面体のもので特定の条件を満たすものはピースの回転に対する形の対称性を持っており、形は異なるが等価なものも存在する。このことからある種のパズルの解法が別の対応するパズルの解法に応用できる可能性が考えられる。そこで、まずは多面体の持つ回転対称性を考えてみる。

正六面体を自分自身に重ね合わせる回転は正六面体群の位数 24 だけ、すなわち

- 恒等変換 → 1
- 六面体の中心と面の中心を通る軸のまわりの回転(3 軸, 90°, 180°, 270°) → 9
- 六面体の中心と頂点を通る軸のまわりの回転(4 軸, 120°, 240°) → 8
- 六面体の中心と辺の中点を通る軸のまわりの回転(6 軸, 180°) → 6

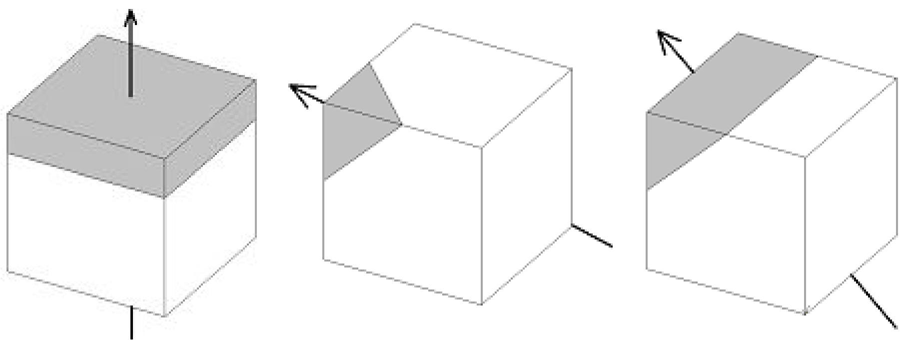


Fig.1.1 立方体の形を保つ分割と回転

がある [1]。正六面体をこれらの軸のまわりに全体回転させて元の形に重ねられるということは、正六面体をこれらの軸に垂直な任意の面で分割し、一方をその軸を中心に単位回転角だけ回転しても完全に元の形になるということである(Fig.1.1)。他の多面体についても同様な回転軸が考えられる(Table 1.1)。正六面体と正八面体、正十二面体と正二十面体は互いに双対であるため、それらの面の中心を結んだ軸と頂点を結んだ軸は互いに同じものであり、辺の中点を結んだ軸はそのまま一致する。Fig.1.1 の様に 1 つの回転軸のまわりに回転するだけではパズルにならないので、

条件 1. 全体の形が正多面体

条件 2. 任意の Layer を規定の角度だけ回転しても全体の形が元の多面体と同じ

条件 3. 回転後に元のピースの位置を同型のピースが占める

条件 4. 内部で交わる分割面を持つ

条件 5. Table1.1 の同種の軸の交換に対してピースの構造が不変

の条件を満たすパズルについて考察する。

条件 1 は考察を限定するため、条件 2, 3 はある軸のまわりの回転が次の回転を妨げることのないようにするためのものである(実在のパズルには故意にそのような作りにすることで難易度を上げているものもあるが)。条件 4 を満たさないものは単純すぎてパズルとしての意味をなさない。条件 5 は対称性を失わないためのものである。

Table 1.1 正多面体を自分自身に重ねる回転の回転軸と単位回転角

	頂点と面の中心を結んだ軸	2 つの面の中心を結んだ軸	2 つの頂点を結んだ軸	2 つの辺の中点を結んだ軸
正四面体	4 軸(120°)	―	―	3 軸(180°)
正六面体	―	3 軸(90°)	4 軸(120°)	6 軸(180°)
正八面体	―	4 軸(120°)	3 軸(90°)	6 軸(180°)
正十二面体	―	6 軸(72°)	10 軸(120°)	15 軸(180°)
正二十面体	―	10 軸(120°)	6 軸(72°)	15 軸(180°)

2. 可能な組合せ

Fig 2.1 の様に条件 3 を満たさない非対称な分割では、面の中心を結んだ一方の軸のまわりの回転に対してはピースが互いに重なる相手を持つが、残り 2 つの軸のまわりの回転で他のピースに重なることができない。

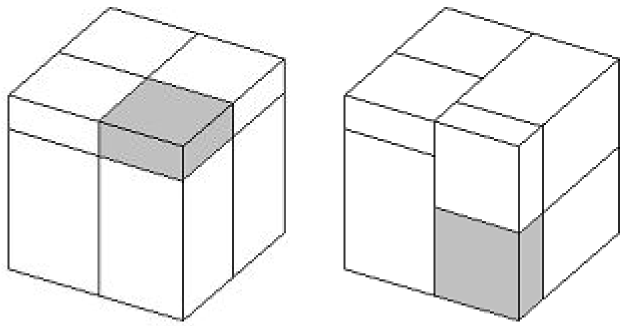


Fig.2.1 非対称な分割による分割面の不整合

多面体の頂点以外のピースの頂点は、パズルの面ー分割面ー分割面の交点かパズルの辺ー分割面の交点である。ピースが回転で重なる相手を持つためにはこれらの頂点も回転で別な頂点に重ならなければならない。

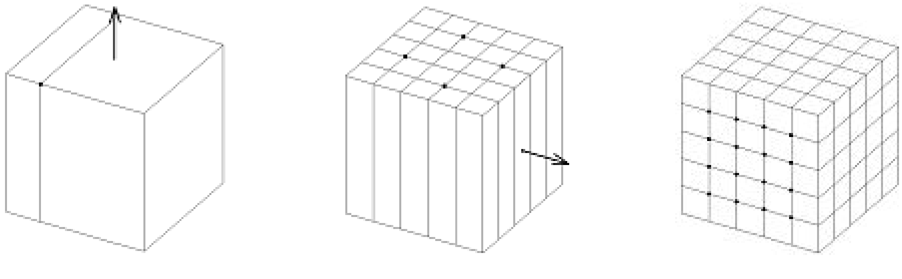


Fig.2.2 ピースが対称な相手をもつための最小限の分割

たとえば Fig2.2 左のような交点を持つ 2 つの面でピースが分割されていた場合、条件 3 を満たすためには、左上に示す軸に対して 90°ずつ回転させた点もピースの頂点になっていなければなら

ない(中央)。すると元々の頂点を 90°回転させたものだけではなく、新しくできた面同士の交点も生じるので、それらも含めて中央上に示す軸に対して 90°の回転に対して対称な点もピースの頂点である必要がある(右)。このことから軸方向の鏡映反転に対してもピースの構造が対称でなければならないことがわかる。これは頂点を結んだ 4 軸同士でも同様。

辺の中点を結んだ 6 軸から同一平面上の 2 軸を選べば閉じた組み合わせができ、条件 4 までは満たすが(Fig.2.3 左)、条件 5 を満たすためには残り 4 軸との交換に対しても対称でなければならない(右)。

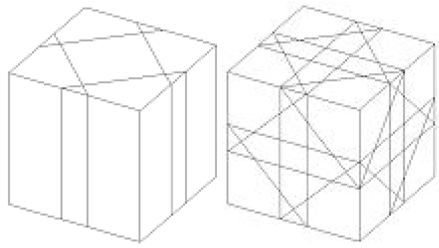


Fig.2.3 同種の軸に対して対称な分割

Fig.2.4 の様に頂点を結ぶ軸と面の中心を結ぶ軸をもつ分割の仕方を組み合わせた場合も考えられるが、異なるタイプの軸同士については分割の幅は無関係にとることができる。実物を作れるかどうかはさておき、この種の分割・回転の組合せで序論の条件を満たす立体パズルを作ることができる。

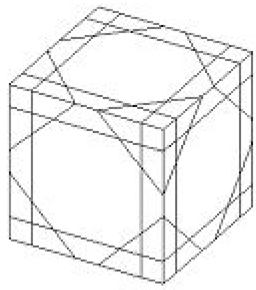


Fig.2.4 複数の種の軸に対する分割の組合わせ

本研究では Table1.1 の網掛けをした軸に垂直な面による分割とその軸まわりの回転でできるパズルをプログラム上で再現した。以後 Table 1.1 の第 1～4 列の軸のまわりに回転するタイプのパズルを頂点－面型、面型、頂点型、辺型と呼ぶことにする。正四面体・頂点－面型、正六面体・面型のパズルについては自動解法を開発した。以下の章でそれぞれについて詳しく述べる。

3. 正六面体・面型パズル(ルービックキューブ型)

ここで扱うパズルを

- 全体の形が正六面体
- 面の中心とパズルの中心を結ぶ 3 本の軸に垂直な面がそれぞれ全体を N 等分する
- 単位回転角は 90°
- ピースは表面の層にのみ存在する

と定義する。$N=2$ のものが *Pocket Cube*、$N=3$ のものが *Rubik Cube*、$N=4$ のものが *Rubik Cube Revenge*、$N=5$ のものが *Professor Cube* として販売されている [2]。プログラムでは $2 \le N \le 10$ の範囲でピース数をメニューから選択できる様にした。

3.1. ピース情報

プログラム上では、フィールドとして固有番号と方向の情報をもったクラス Piece の配列インスタンスとしてパズル全体を扱う。

3.1.1. 座標系

パズル全体を総数N^3のピースの 1 次元配列で表す。表面上のピースに対応するものだけを考えるので、該当するピースを持たない配列要素も存在する。ピースの位置はX,Y,Zの方向にそれぞれ 0〜$N-1$の値をとる座標値で表されるが、位置(x,y,z)にあるピースの index を$x+N(y+Nz)$とすることで index と座標を 1 対 1 に対応させる。一方、パズルを完成させた時点で収まるべき座標の index に一致する量$Number$を固有番号としてピースに割り当てておき、自動解法ルーチンではこれを手がかりに目的のピースを捜す。

3.1.2. ピースの方向

各ピースは 1〜3 色のラベルの面を露出しているが、初期状態で全てのピースが同じ方向を向いているものとし、ピースの方向を表す量を

$$\begin{pmatrix} D_{xX} & D_{xY} & D_{xZ} \\ D_{yX} & D_{yY} & D_{yZ} \\ D_{zX} & D_{zY} & D_{zZ} \end{pmatrix} = \begin{pmatrix} \mathbf{D}_x \\ \mathbf{D}_y \\ \mathbf{D}_z \end{pmatrix} \tag{1}$$

の様に定義する。一つ一つのピースがこれらの情報を独立に持つ。$\mathbf{D}_x$，$\mathbf{D}_y$，$\mathbf{D}_z$はピースの方向ベクトルで、回転してもこれらは互いに垂直で、鏡映の状態にはならないので取り得る方向は$6\times 4=24$通りだけになる(正六面体群の位数 = 4 次の対称群の位数 = $S_4=4!=24$)。これらは初期状態でグローバルなX,Y,Zの方向を向いているので、行列は単位行列になる。シャッフル後にはこの行列から任意のピースの任意の面側の色を知ることができる。X_+面の色が知りたければ､そのピースの$\mathbf{D}_x$，$\mathbf{D}_y$，$\mathbf{D}_z$のうちX軸に平行か反平行のものを探す｡たとえば$\mathbf{D}_y$が(−1, 0, 0)なら、完成状態でいうところのY_-側の面を(すなわち色を)X_+面側に見せていることになる。

3.1.3. 層の回転

任意の最小単位の回転は 1 層のみの 90 °の回転であり

- 回転軸
- 回転方向
- 回転層

の 3 つの量で一意に決定される。回転軸はX,Y,Zのいずれか、回転方向は回転軸に対して右ネジの方向を+1, 逆方向を−1 とし、回転層の番号は 0〜$N-1$の値をとる。

回転軸方向の座標が回転層番号と等しいピースが移動・回転の変換をうける。ピースは表面にしか存在しないので両端の層とそれ以外でピース数が異なる(Table 3.1 参照)。

ここでいうセンターピースとは回転軸上にある面中央のピース(Nが奇数の時だけ存在する)、インナーピースとはセンターピース以外の 1 つの面だけに属するピース、エッジピース・コーナーピースはそれぞれ 2 つ・3 つの面に属するピースをいう。一般の解説書や解説サイトなどではセンターピース→センターキューブなどと呼ばれているが、本稿ではパズル全体との混乱を避けるためと、他の種類のパズルの場合と呼称を統一するためこの様に呼ぶことにする。

Table 3.1 回転層上のピース数

Layer	センター＋インナーピース	エッジピース	コーナーピース	全ピース
0，$N-1$	$(N-2)^2$	$4(N-2)$	4	N^2
1～$N-2$	$4(N-2)$	4	0	$4(N-1)$

3.1.4. 回転による位置の変化

X 軸まわりの第 L 層の回転では $x=L$ を満たすものが移動する。Y 軸，Z 軸まわりの回転についても同様で、Table3.2 の位置にあるピースが (x,y,z) の位置に移動する。N が奇数の時にだけ存在する軸上のセンターピースをのぞけば、ピースの移動は 4 つずつの巡回置換になる。置換されるのは必ず同種のピース同士になる。

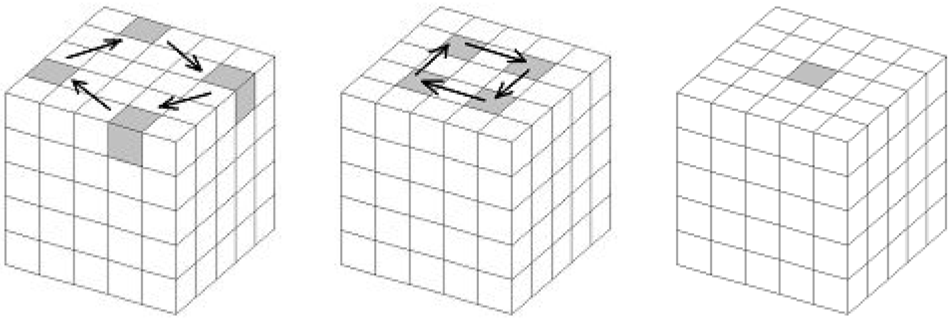


Fig.3.1 Y 軸まわり第 $N-1$ 層逆方向回転によるピースの移動

Table 3.2 回転によって (x,y,z) に移動するピース

	X 軸	Y 軸	Z 軸
正方向	$(x,z,N-1-y)$	$(N-1-z,y,x)$	$(y,N-1-x,z)$
逆方向	$(x,N-1-z,y)$	$(z,y,N-1-x)$	$(N-1-y,x,z)$

3.1.5. 回転による方向の変化

$$X_+=\begin{pmatrix}1 & & \\ & & 1\\ & -1 & \end{pmatrix}\quad Y_+=\begin{pmatrix} & & -1\\ & 1 & \\ 1 & & \end{pmatrix}\quad Z_+=\begin{pmatrix} & 1 & \\ -1 & & \\ & & 1\end{pmatrix} \tag{2}$$

X ,Y ,Z 軸のまわりに正方向に回転した後の方向行列はもとの行列に(2)の回転行列を右からかけることで得られる。また、逆方向の回転に対応する行列 X_-,Y_-,Z_- はこれらの転置行列で表される。回転層上の全てのピースがこの方向変換をうける。

3.2. 表示

3.2.1. VBA での表示・操作

$N=3,4$ の場合についてのみ VBA による再現も行った。オートシェイプの平行四辺形をピースの面として用いてパズルの表面のうち X_+,Y_+,Z_+ の 3 面に充て、ピースの情報に応じてその色を変えることでパズルの情報を表示した。また、Excel シートのセルを正方形に整形して展開図とし、

本体と同時にセルの色を変えて裏側の情報も見られる様にした。回転の操作はユーザーフォームに用意した単層回転用の 18 または 24 個のボタン、全体回転用の 6 個のボタンで行い、シャッフル、自動解法実行もそこから行う様にした。

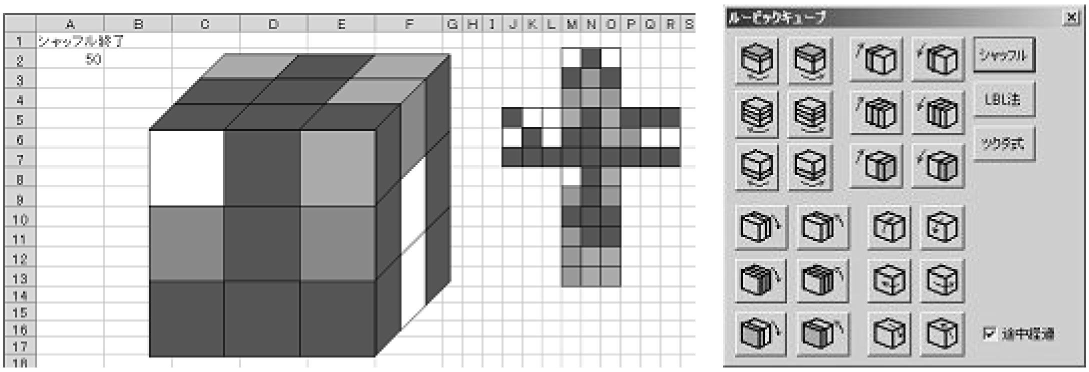


Fig.3.2　$N = 3$ のパズル VBA の表示と操作メニュー

VBA では表示､操作系に限界があるため再現したのはこの 2 つだけであり､以降は全て Visual C#, Java Applet によるものである。

3.2.2.　座標系

一つのアプリケーションで複数のパズルを選択できる様にするため、種類選択の際にパズルの表示に必要な 3 次元座標点の配列 $Grid$ を生成し、ユーザーがマウスのドラッグ、キーボードのカーソルキーで全体を回転したときにそれを書き換える様にした。

今後 (x, y, z) は整数の値をもつ座標、(X, Y, Z) は 3 次元空間の点に対応する実数の座標を表すものとし、以後のパズルにおいてもこの意味で小文字と大文字を使い分ける。

$Grid$ は $(N+1)^3$ 個の要素をもつ 1 次元配列で定義し、0～ N の値をとる (x, y, z) に対応する $x + (N+1)(y + (N+1)z)$ の index を与えた。これに対して、3 次元空間上の実数座標を(3) の様に割り当てることで、パズル全体を半径 1 の外接球を持つ立方体にした。

$$X = \frac{2x - N}{N}\frac{1}{\sqrt{3}} \qquad Y = \frac{2y - N}{N}\frac{1}{\sqrt{3}} \qquad Z = \frac{2z - N}{N}\frac{1}{\sqrt{3}} \tag{3}$$

表示の初期状態では見た目をわかりやすくするためこれを X 軸まわりに－30°、Y 軸まわりに 30°回転させる。

3.2.3.　全体の回転

画面上をマウスでドラッグしたときにパズルが回転する機能を加えた。すなわち、マウスポインタのある点の位置を連続で取得し、X 方向の変位に応じて Y 軸まわりに回転、Y 方向の変位に応じて X 軸まわりに回転させた。また、キーボードの上下左右を 1 回押す毎にそちら側へ 15°ずつ回転する様させた。これ以降の別のパズルについても同様。これだけでパズルの裏表の回転は容易にできるが、人間が立体パズルを解くときは手前に来る面自体は変えないまま視線に垂直な軸、すなわち Z 軸まわりの回転ができると感覚的に理解しやすくなることがある。この回転はマウスの変位を $(\Delta x, \Delta y)$,画面の中央を (C_x, C_y), マウスの位置を (P_x, P_y) として、マウスの変位のベクトルと画面中央からマウスポインタのある点を指す単位ベクトルとの外積の大きさ(4)を求め、そ

れに比例した角度だけZ軸まわりの回転をさせることで実現した(Shift キーを押しながらドラッグしたときにこのタイプの回転をし、そうでないときはX ,Y軸まわりの回転をする様にした)。市販のCGや3-D CADのアプリケーションでも視線に垂直な方向の軸まわりの回転の機能が使えず不便に感じていたが、この手法で解決できた。

$$\frac{\Delta x(C_y - P_y) - \Delta y(C_x - P_x)}{\sqrt{(C_x - P_x)^2 + (C_y - P_y)^2}} \qquad (4)$$

どちらの種類の回転でも、計算の負荷を最小限にするために、点の座標を受け取り表示に必要な表面上の点であるかどうかを返すboolメソッド *IsPointOnSurface* を作成・利用した。

3.2.4. 面の表示

全体回転をすると表示すべき面が変わるので、手前側すなわちZの正方向にある面だけを表示するための判定が必要になる。原点から(N, N, N), $(N,0,0)$の 2 つの点へのベクトルを合成すると X 面の正方向の法線になることから、これらの 3 次元座標のZ成分だけを加えてその正負で表示面の判定をした。同様に(N, N, N)と $(0, N,0)$, $(0,0, N)$を組み合わせることでY ,Z面の表示側を決めた。

X の正方向の面を表示する場合には、その面上にあるピースについてX の正側を向いている面を方向行列から調べて色を決定し、ピース座標からX の正側の面の頂点に対応する 4 つの*Grid*座標を調べて一つのピースのX 面を表示した。実際には、たとえばピースの座標$(N-1, y, z)$を渡すと、その頂点にあたる*Grid*の座標$(N, y, z),(N, y+1, z),(N, y+1, z+1),(N, y, z+1)$の 4 点の 3 次元実数座標を返し、$Y$,Z面上のピースについても同様な動作をするメソッドを作り、それぞれのピースについてこれを適用することでパズル全体を表示する。

3.2.5. 回転情報とプレビュー

表示されたパズルの上でマウスのボタンをクリックするとそのピースのある層が回転する様にした。ただし、予想外の方向に回転してしまっては困るので、ピース上にマウスポインタを置いた時点で回転方向を矢印で表示し、回転するピースの枠を太線で強調表示する様にした。回転方向は 3 軸の正負で 6 方向あるが、ある特定の面上にポインタがあるときはその面に垂直な軸まわりの回転はないものとし、残り 4 方向のいずれかを決定する。回転する方向を決めるため、マウスの置かれているピースの面の頂点の座標とポインタのある点をもとに、

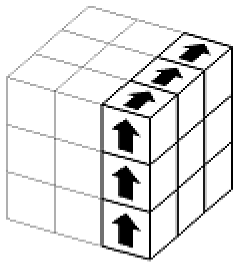


Fig.3.3 回転方向のプレビュー

(1) ピース面の頂点の一つが原点に来る様に平行移動

(2) x 座標に比例した値を y 座標に加えて一辺が x 軸に重なる様にする(その辺が y 軸上にあったときのみ回転移動)

(3) y 座標に比例した値を x 座標に加えてもう一辺が y 軸に重なる様にする

(4) 原点が中心に来る様に平行移動し、スケール変換で縦横の長さを揃える

という一連の座標変換でピース面を原点を中心にもつ正方形に変形し、同時にその変換をマウス

から取得した点にも施し、対角線で分割した 4 つの領域のどこにその点があるかを調べる。

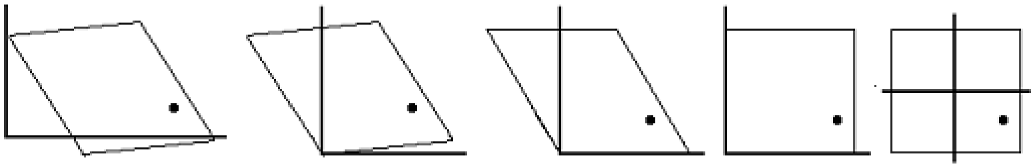


Fig.3.4 ピース面の原点を通る正方形への座標変換

矢印の表示は、ピースの頂点の座標と中心からの 2 つの方向への変位(－0.5～0.5)から二次元的な内分点を返すメソッド $InnerPoint$ を作り、それに矢印の頂点に対応するパラメータをいくつか渡すことで実現している。

3.2.6. アニメーション

ある層が実際に回転している様に見せるために、その層だけを軸まわりに少しずつ回転させて表示した。すなわち、回転軸に垂直な面の頂点から回転軸方向のベクトルを取得し、先に述べた基準の $Grid$ を(1)～(5)の様に変換したものを一時的な $Grid$ として回転する層の表示に使用した。

(1) 回転軸方向のベクトルが XY 平面上に来る様に全体回転

(2) そのベクトルが X 軸上に来る様に全体回転

(3) X 軸まわりに指定角度(0～90°)全体回転

(4) (2)の逆回転

(5) (1)の逆回転

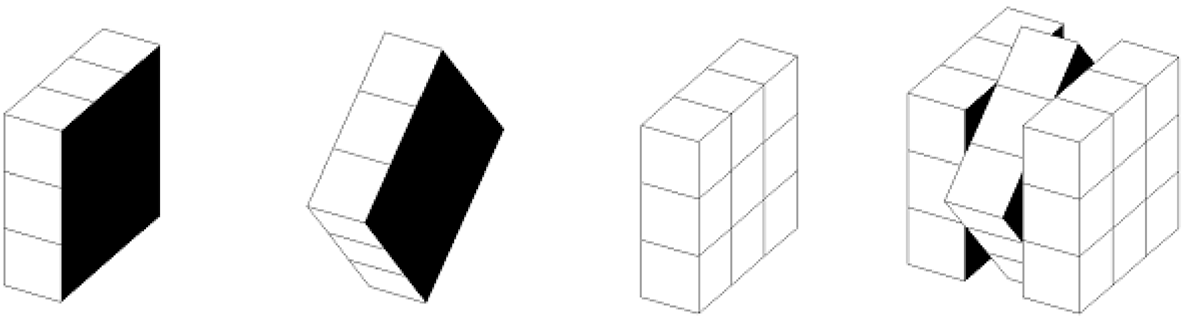


Fig.3.5 基準 $Grid$ と一時 $Grid$ の組合せによるアニメーション表示

その際高速化のため回転する層のピースの頂点に対応する座標のみを計算した。もっとも、回転層では表示面の裏表が途中で変わることもあるため、たとえ回転層上になくても面の前後判定に使用する (N,N,N), $(N,0,0)$, $(0,N,0)$, $(0,0,N)$ の 4 点は常に計算に含めた。

奥のブロック、回転する層、手前のブロックの 3 ブロックを順に仮イメージに表示し、その後画面を更新することで前後の矛盾無く表示できた。アニメーション中に見える内部の黒の面の表示には $Grid$ から 4 つの点を取得して使用した。

ピースの色情報をそのままにしてこの様にアニメーション表示をし、アニメーションが終わった瞬間に情報を書き換えて通常表示をすることでシームレスに回転後の表示につながる。

3.2.7. マーク表示

表面のラベルの色だけを頼りにパズルを解いた場合、見た目が元の状態と同じになっていても面の内側のピースの向きや位置($N \geq 4$)が初期状態とは異なっている場合がある。向きを知るために Fig.3.6 の様に表面にピースの方向を示すマークを表示する機能を加え、メニューで ON, OFF できる様にした。マークの方向はピースの方向データから決定し、表示には回転プレビューで矢印表示に使用した *InnerPoint* を流用した。

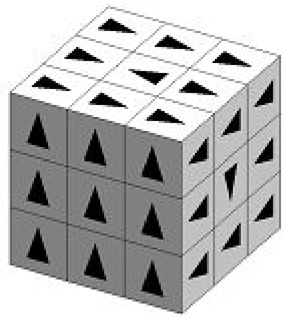


Fig.3.6 マーク表示

3.2.8. ミラー表示

マウスのドラッグなどで全体を回転させなくてもパズルの裏側を見られる様に、右側に裏側の状態を表示する機能を加えた。通常の表示の際に使用する表示面判定を逆に適用し、表示位置を右側にオフセットすることで裏面を表示している。これもメニューから ON, OFF する様にした。この機能は全てのパズルで使用できる様にしてある。

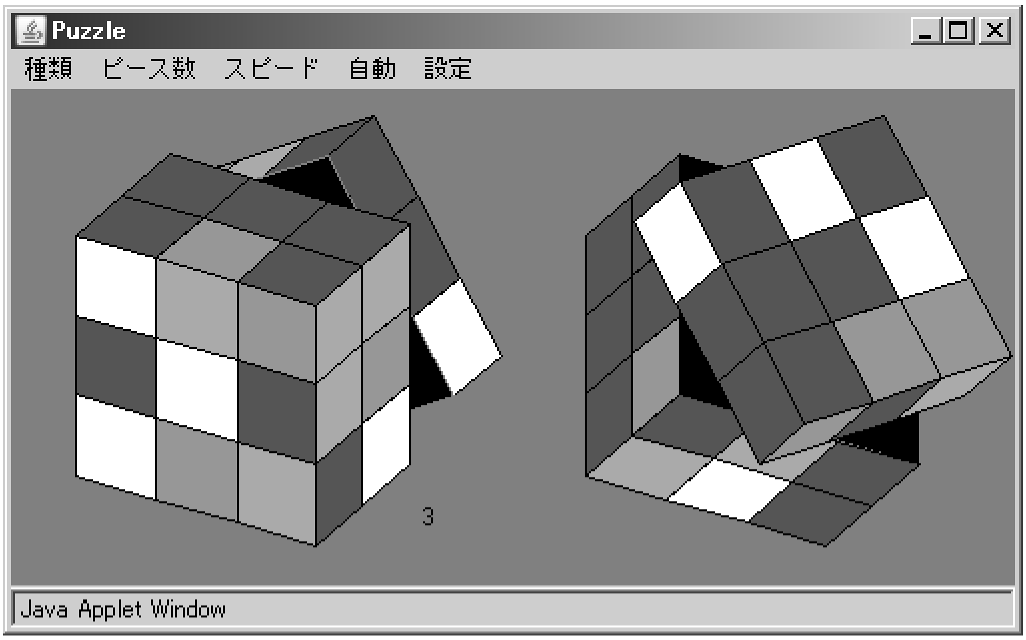


Fig.3.7 ミラー表示

4. 正六面体・頂点型パズルⅠ

ここで扱うパズルを

- 全体の形が正六面体
- 頂点とパズルの中心を結ぶ 4 本の軸に垂直な面が全体を $3N$ 個に等間隔に分割する
- 単位回転角は 120°
- ピースは表面の層にのみ存在する

と定義する。 $N=1$ のものが *Dinosaur*, $N=2$ のものが *Lattice Cube* として販売されている[3)]。

プログラムでは$1 \le N \le 10$の範囲でピース数をメニューから選択できる様にした。

4.1.　ピース情報

4.1.1.　座標系

このパズルの回転軸は 4 本あるが、ピースは正六面体の表面にのみ存在するので、ピースの位置を表す座標系はx, y, zの 3 次元のものを使う。

Fig.4.1 に示す様にピースの正方形の面の中心をピースの基準点とし、原点からのX ,Y ,Z 方向の変位に比例する様に座標値を振る。一辺のピース数が N であるときピースの座標値x ,y ,z はそれぞれ0～$2N$ の値を持ち、$x=0$か$x=2N$ の面ではy ,z は必ず偶数と奇数の組合せになる。Y ,Z の裏表の面についても同様。(x, y, z) の座標値に対して$x+(2N+1)(y+(2N+1)z)$の index を対応させ、

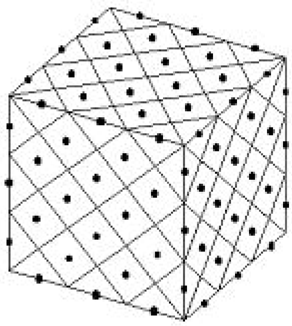


Fig.4.1 ピースの基準点

初期状態でそれに等しい$Number$を与える。露出している面はX ,Y ,Z の正負方向のみなので、ピースの方向の情報は正六面体面型パズルと同じ 3×3 の行列で表される。

4.1.2.　層の回転

回転軸はI 軸((1, 1, 1)方向)，J 軸((1,－1，－1)方向)，K 軸((－1, 1，－1)方向)，L 軸((－1，－1, 1)方向)の 4 本あり、回転方向は回転軸に対して右ネジの方向を$+1$，逆方向を-1とし、回転層は奇数の値をとり、回転軸の先に向かって 2 ずつ増加する様に数える(1～$6N-1$)。回転層上のピース数を Table 4.1 に示す。ここで「インナーピース」は 1 面だけを露出した正方形のピース、「エッジピース」は 2 面にまたがるピースを示す。インナーピースは正八面体を 4 つの頂点を通る面で二等分した形、エッジピースはそれをさらに二等分した形をしている。

Table 4.1 回転層上のピース数

$Layer$	インナーピース	エッジピース	全ピース
1～$2N-1$	$3(Layer-1)$	3	$3Layer$
$2N+1$～$4N-1$	$6N$	6	$6(N+1)$
$4N+1$～$6N-1$	$3(6N-1-Layer)$	3	$3(6N-Layer)$

4.1.3.　回転による位置・方向の変化

$N_{max}=2N$ として、第 $Layer$ 層を回転する場合に移動するのはI 軸まわりでは$x+y+z=Layer$を満たすピース、J 軸まわりでは$-x+y+z=N_{max}-4Layer$、K 軸まわりでは$x-y+z=N_{max}-4Layer$、L 軸まわりでは$x+y-z=N_{max}-4Layer$をを満たす座標値(x, y, z)をもつピースである。これらが Table 4.2 のような移動をする。どの場合も 3 つずつのピースの巡回置換となる。

Table 4.2 回転によって(x, y, z)の位置に移動するピース

	I 軸	J 軸	K 軸	L 軸
正方向	(y, z, x)	$(N_{\max} - y, z, N_{\max} - x)$	$(N_{\max} - y, N_{\max} - z, x)$	$(y, N_{\max} - z, N_{\max} - x)$
逆方向	(z, x, y)	$(N_{\max} - z, N_{\max} - x, y)$	$(z, N_{\max} - x, N_{\max} - y)$	$(N_{\max} - z, x, N_{\max} - y)$

I , J , K , L 軸のまわりの正方向の回転行列は(5)の様になる(逆方向の回転行列はこれらの転置行列)。回転する層にある全てのピースがこの変換を受ける。

$$I_+ = \begin{pmatrix} & 1 & \\ & & 1 \\ 1 & & \end{pmatrix} \quad J_+ = \begin{pmatrix} & -1 & \\ & & 1 \\ -1 & & \end{pmatrix} \quad K_+ = \begin{pmatrix} & -1 & \\ & & -1 \\ 1 & & \end{pmatrix} \quad L_+ = \begin{pmatrix} & 1 & \\ & & -1 \\ -1 & & \end{pmatrix} \tag{5}$$

4.2. 表示

4.2.1. 座標系

$Grid$ の座標系はピースの座標系と同じく 3 次元でとる。それぞれの座標が 0～$2N$ の値をとるので、要素数$(2N+1)^3$で$Grid$ を定義するが、ピースの頂点に当たる、表示に必要な点の座標はXの正側の面 $x = 2N$ では y, z が共に偶数か共に奇数の点のみとなる。

4.2.2. 面の表示

正六面体･面型のパズルの場合と同様にピースの座標から$Grid$ の座標をもとめるメソッドをつくるが､変則的な形のピースがあるので場合分けを行う｡インナーピースの座標を渡されたときは､X 面ならピース座標から y, z 座標が(0, 1), (1, 0), (0, －1), (－1, 0)だけずれた座標をもつ 4 つの $Grid$ 点を返し、エッジピースの場合はパズル外の 1 点を落とした 3 点を返す様にする。

4.2.3. 回転情報とプレビュー

4 本の軸のまわりに正負の方向に回転するので、あるピースが回転しうる方向は 8 通りある。面に並行な回転はないので 8 通りすべてについて考える。そのため、正方形のピース面を 8 つの領域に分けて回転情報の判断に使用した。エッジの三角形のピース面は 4 つに分割して、回転候補はエッジ側へ回るもののみとした。

パズル上の全ての点で回転候補の取得はするが、実際に矢印を三角形上に表示するとプレビューの矢印が小さくなってしまうので、コーナーの部分でその頂点を含む軸まわりの回転が候補であるときのみ 2 つのピースにまたがる矢印を表示することとし、それ以外は表示を省略した。

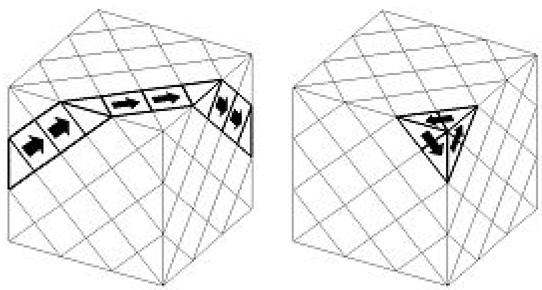


Fig.4.2 エッジ部分とコーナー部分の例外処理

正方形の面の矢印表示には面型パズルで使用したメソッド $InnerPoint$ を流用するが、コーナー部の矢印の表示には、そのメソッドを直角二等辺三角

形用に書き換えたものを使用する。

4.2.4. アニメーション

表示の手法は面型パズルの場合と同じだが、回転軸の方向ベクトルはそれぞれ(N,N,N), $(N,0,0)$, $(0,N,0)$, $(0,0,N)$の実数座標そのものを利用した。これらのベクトルは奥のブロック、回転ブロック、手前ブロックの順に表示する際の前後判定をするのにも利用した。

また、このパズルの軸の取り方では、軸に含まれる頂点からもう一方の頂点との間を 3 等分した区間のうち、両端の区間での分割では断面は三角形(実際は正三角形)になり、中央の区間では六角形になるため、分割面の黒表示の頂点の座標を取得する際は層番号による場合分けを行った。

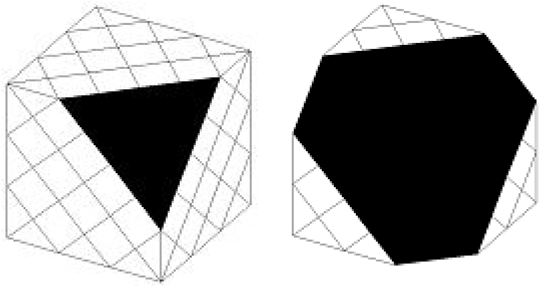


Fig.4.3 正六面体·頂点型パズルⅠの分割面

5. 正六面体・頂点型パズルⅡ

ここで扱うパズルを

- 全体の形が正六面体
- 頂点とパズルの中心を結ぶ 4 本の軸に垂直な面がそれぞれ全体を$3N-4$個の厚さの等しい層と、厚さがその半分で軸上にある頂点を含む 2 つの層に分割する
- 単位回転角は 120 °
- ピースは表面の層にのみ存在する

と定義する。$N=2$のものが $Skewb$, $N=3$のものが $Master\ Skewb$ として販売されている[3)]。プログラムでは$2 \le N \le 10$の範囲でピース数をメニューから選択できる様にした。

5.1. ピース情報

5.1.1. 座標系

4 本の軸まわりに回転すること、ピースを 3 次元座標系x, y, zで表すことはタイプⅠの場合と同じだが、層の切り方が異なるので、とることのできる座標の条件が違う。タイプⅠと同様に、正方形の中心をピースの基準点とする。すると、X の正負の面上にあるピースy,zの座標値はともに偶数かともに奇数かのいずれかになる。また、それぞれの方向の座標の最大値は$2N-2$になるので、(x,y,z) の位置に $x+(2N-1)(y+(2N-1)z)$の index を対応させる。ピースの方向は 3×3 の行列で表される。

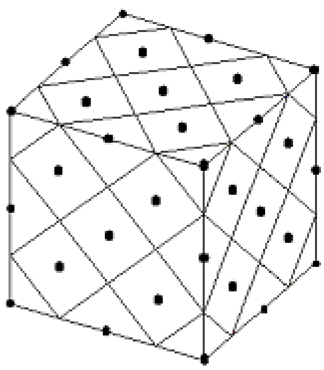


Fig.5.1 ピースの基準点

5.1.2.　層の回転

回転軸と回転の方向はタイプⅠと同じだが、回転層の層番号は 0～$6N-6$ の偶数になる。回転層上のピース数を Table 5.1 に示す。インナーピース、エッジピースはタイプⅠと同様で、コーナーピースは 3 面を露出したコーナーに位置するピース(正六面体のコーナーの形)を表す。

Table 5.1 回転層上のピース数

Layer	インナーピース	エッジピース	コーナーピース	全ピース
0	0	0	1	1
$2 \sim 2(N-2)$	$3(Layer-1)$	3	0	$3Layer$
$2(N-1)$	$3(2N-3)$	0	3	$6(N-1)$
$2N \sim 4(N-2)$	$6(N-2)$	6	0	$6(N-1)$
$4(N-1)$	$3(2N-3)$	0	3	$6(N-1)$
$4N \sim 6(N-2)$	$3(6N-7-Layer)$	3	0	$3(6N-6-Layer)$
$6(N-1)$	0	0	1	1

5.1.3.　回転による位置・方向の変化

回転層が両端のどちらかの場合は、層上にあるのが 1 個のコーナーピースだけになり、位置は回転後も変わらない。それ以外の場合では同種のピースの 3 つずつの組み合わせが巡回置換を受ける。方向の変化の変換行列はタイプⅠと全く同じになる。

5.2.　表示

5.2.1.　座標系

Grid の座標系の取り方はほぼタイプⅠと同じだが、それぞれの座標の最大値が $2N-2$ なので、要素数 $(2N-1)^3$ の *Grid* を定義する。ピースの頂点に当たる、表示に必要な点の座標は X の正側の面だと $x=2N-2$，y,z が偶数と奇数の組合せになる点のみ。

5.2.2.　表示・操作系

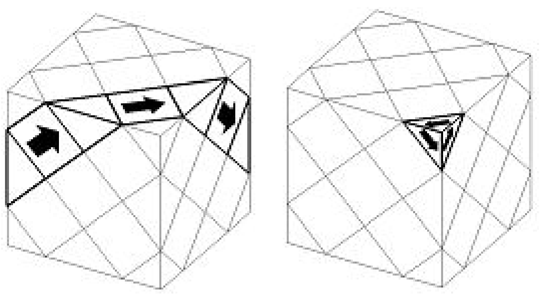


Fig.5.2 エッジ部分とコーナー部分の例外処理

表示のためのインナー・エッジピースの *Grid* の座標の取得方法はタイプⅠと同じ。コーナーピースでは正方形の面の頂点に当たる 2 つの点とパズル全体の頂点を組み合わせて直角二等辺三角形を作る。マウスによる回転ではインナー・エッジピース上ではタイプⅠと同じ様に回転候補を取得する。ただし、コーナーピース上で 8 または 4 方向の回転候補を取得しようとすると、分割された領域が狭すぎて操作性が悪くなるので、2 分割でコーナーを含む軸まわりの回転の正負だけを与える様にした。プレビュー表示はインナーピースとコーナーピース上のみとした。

6.　正六面体・辺型パズル

ここで扱うパズルを

- 全体の形が正六面体
- 対向する辺の中点同士を結ぶ 6 本の軸に垂直な面がそれぞれ全体を$2N$個の厚さの等しい層に分割する
- 単位回転角は 180 °
- ピースは表面の層にのみ存在する

と定義する。この定義に従う実物は確認できていないが、$2N=3$にあたるものが $Bevel\ Cube$ (これは 90 ° 単位で回転するので序論の条件 2 を満たさず、回転させると全体の形が変化していく)として販売されている [3]。プログラムでは$1 \le N \le 10$の範囲でピース数をメニューから選択できる様にした。

6.1. ピース情報

6.1.1.　座標系

6 本の軸まわりに回転するが、ピースは 3 次元座標系x, y, zで表す。それぞれの辺の方向に 0～$2N$の値をとる均等な座標系を考えると、Fig.6.1 に示す格子点がピースの面内に 1 つずつ収まるので、これをピースの座標とする。X_+ 面内のピースの座標は $x=2N$, (y,z) = $(4m+2,4n+1)$, $(4m+2,4n+3)$, $(4m+1,4n+2)$, $(4m+3,4n+2)$のどれかになる(m,nは 0～$N-1$の整数)。

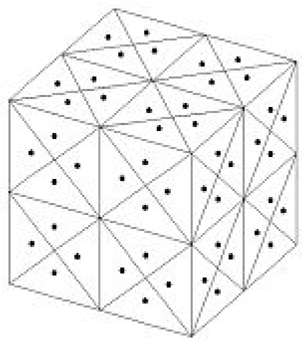


Fig.6.1 ピースの座標点

ピースはどれも立方体のコーナーの形をしている。回転軸I, J, K, L, M, Nは$a=1/\sqrt{2}$として方向ベクトル$\mathbf{e}_i, \mathbf{e}_j, \mathbf{e}_k, \mathbf{e}_l, \mathbf{e}_m, \mathbf{e}_n$を持つものとする。ピースの露出する面はどれも 1 面だけで、X, Y, Zの正負のいずれかを向くので、方向は正六面体の他のパズル同様 3×3 の行列で表す。

Table 6.1 回転軸の方向ベクトル

$\mathbf{e}_i$	$\mathbf{e}_j$	$\mathbf{e}_k$	$\mathbf{e}_l$	$\mathbf{e}_m$	$\mathbf{e}_n$
$(0,a,a)$	$(a,0,a)$	$(a,a,0)$	$(0,a,-a)$	$(-a,0,a)$	$(a,-a,0)$

6.1.2.　層の回転

6 本の回転軸のまわりに$2N$個の層のいずれかが回転する。回転角は 180 ° なので正負どちらに回転しても結果は同じになるが、アニメーションのために方向の区別もしておく。

層上にあるピースの判別のため、回転層の番号は 2 から始まり 4 ずつ増加して最大で$8N-2$になるものとする。隣の面に移動するピースを「側面ピース」、対面に移動するピースを「底面ピース」と呼ぶことにすると回転層上にあるピース数は Table 6.2 の様になる。

Table 6.2. 回転層上のピース数

$Layer$	側面ピース数	底面ピース数	全ピース数
$2 \sim 4N-2$	$8N$	$2Layer$	$2(4N+Layer)$
$4N+2 \sim 8N-2$	$8N$	$2(8N-Layer)$	$2(16N-Layer)$

6.1.3. 回転による位置・方向の変化

層上のどのピースも対になる相手を持ち、回転は単純なそれらの互換になる。それぞれの軸のまわりの回転では Table 6.3 にあるピースと(x,y,z)のピースが入れ替わる。回転行列は(6)の様になる。回転方向に正負はない。

Table 6.3 回転によって(x,y,z)に移動するピース

I 軸	J 軸	K 軸
$(4N-x,z,y)$	$(z,4N-y,x)$	$(y,x,4N-z)$
L 軸	M 軸	N 軸
$(4N-x,4N-z,4N-y)$	$(4N-z,4N-y,4N-x)$	$(4N-y,4N-x,4N-z)$

$$I=\begin{pmatrix}-1&&\\&&1\\&1&\end{pmatrix}\quad J=\begin{pmatrix}&&1\\&-1&\\1&&\end{pmatrix}\quad K=\begin{pmatrix}&1&\\1&&\\&&-1\end{pmatrix}$$
$$L=\begin{pmatrix}-1&&\\&&-1\\&-1&\end{pmatrix}\quad M=\begin{pmatrix}&&-1\\&-1&\\-1&&\end{pmatrix}\quad N=\begin{pmatrix}&-1&\\-1&&\\&&-1\end{pmatrix} \tag{6}$$

6.2. 表示

6.2.1. 座標系

点の位置を表す座標系はピースの座標系に準ずる。ピースの頂点に当たる点はX_+面内であれば$x=4N$，(y,z)= $(4m,4n)$，$(4m+2,4n+2)$のどれかになる(m,nは$0\sim N$または$0\sim N-1$の整数)。

6.2.2. 表示・操作系

描画の際には、ピースの座標からピース面の斜辺の方向を判断し、ピース面の直角二等辺三角形の頂点の座標を取得する。回転層のプレビューでは底面側の領域の幅が狭いので側面側のみ矢印を表示する様にした。また、マウスポインタから取得する回転の回転軸はポインタのある面と交差する 4 軸まわりの正負 8 方向とした。

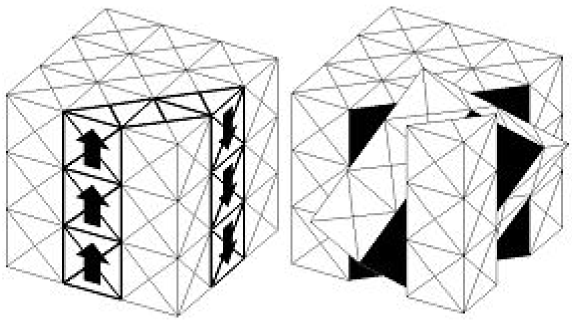


Fig.6.2 回転プレビューとアニメーション

7.　正四面体・頂点一面型パズル

ここで扱うパズルを

- 全体の形が正四面体
- 頂点と対向する面の中心を結ぶ 4 本の軸に垂直な面が全体を N 個に等間隔に分割する
- 単位回転角は 120 °
- ピースは表面の層にのみ存在する

と定義する。$N=3$ のものが $Pyraminx$, $N=4$ のものが $Master\ Pyraminx$ として販売されている [3]。プログラムでは $2 \le N \le 10$ の範囲でピース数をメニューから選択できる様にした($N=2$ のものはパズル内部で分割面が交差しないのでパズルとして意味をもたず序論の条件を満たさないが、便宜上ここに分類する)。

7.1.　ピース情報

7.1.1.　座標系

パズル全体の形を原点を中心にもつ半径 1 の球に内接し、I 軸 ：(1, 1, 1)方向、J 軸 ：(1, −1, −1)方向、K 軸 ：(−1, 1, −1)方向、L 軸 ：(−1, −1, 1)方向の 4 本の軸上の正側に頂点を持つ正四面体とし、それらに垂直なパズルの面を I , J , K , L 面とする。ピースには 4 次元座標 (i, j, k, l) を振り、i 座標はパズルの I 面(I 軸に垂直な面)に接するもので 0 となり、I 面に平行な分割面を I 軸上の頂点側に向かってひとつ越えるたびに 1 増加するものとする。

パズルを構成するピースには 2 種類あり、一つはパズル全体と相似なピース(順方向ピース)、もう一つは順方向ピースと一辺の長さの等しい正八面体の逆方向ピースである。$N \ge 4$ ではどちらの種類のピースも 1 つの面だけに属するものから 3 つの面にまたがるものまである。

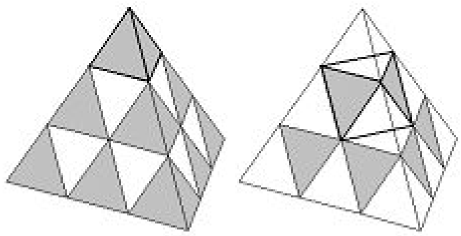


Fig.7.1 順方向ピースと逆方向ピース

定義より I 軸上の頂点を含むピース(Fig.7.1. 左太枠)の座標が $(N-1,0,0,0)$ であり、隣り合う 3 つの順方向ピースは i 座標がこれより 1 小さく、 j,k,l のいずれかの座標が 1, その他が 0 であることから、全ての順方向ピースの座標について

$$i + j + k + l = N - 1$$

が成り立つことがわかる。また、$(N-1,0,0,0)$ に隣り合う逆方向ピース(Fig 7.1 右太枠)の座標は $(N-2,0,0,0)$ であり、これも隣接する逆方向ピースとの座標の増減を考えれば、すべての逆方向ピースが

$$i + j + k + l = N - 2$$

を満たすことがわかる。すなわち N^4 通りの座標値のうち実際のピースに対応するのは $N-2 \le i+j+k+l \le N-1$ を満たすものだけとなる。

座標が 4 次元なので (i, j, k, l) の位置に $i + N(j + N(k + Nl))$ の index を対応させる。

順方向ピースと逆方向ピースで形は異なるが、露出する面は平行であり、回転に対して同じ変換

を受けるので、初期状態で全てのピースが同じ方向を向いているものとして扱える。個々のピースの方向は(7) で表される。初期状態で $\mathbf{D}_i, \mathbf{D}_j, \mathbf{D}_k, \mathbf{D}_l$ はそれぞれ I 軸, J 軸, K 軸, L 軸の方向を向いている。回転後のこれら 4 つのベクトルの位置関係が変わらないことから、取り得る方向は $4 \times 3 = 12$ のみとなる(正四面体群の位数 4 次の交代群の位数 = $4!/2 = 12$)。正多面体の中で正四面体だけは「裏側」を持たないので、シャッフルした後もそれぞれのベクトルは必ずこれら 4 軸のどれかの方向を向く。

$$\begin{pmatrix} D_{iI} & D_{iJ} & D_{iK} & D_{iL} \\ D_{jI} & D_{jJ} & D_{jK} & D_{jL} \\ D_{kI} & D_{kJ} & D_{kK} & D_{kL} \\ D_{lI} & D_{lJ} & D_{lK} & D_{lL} \end{pmatrix} = \begin{pmatrix} \mathbf{D}_i \\ \mathbf{D}_j \\ \mathbf{D}_k \\ \mathbf{D}_l \end{pmatrix} \tag{7}$$

7.1.2. 層の回転

回転軸は I, J, K, L の 4 軸、回転方向は正負の 2 通り、回転層は 0～ $N-1$ のいずれかになる。Table 7.1 に回転層上のピース数を示す。

Table 7.1 回転層上のピース数

$Layer$	順方向ピース	逆方向ピース	全ピース
0	$N(N+1)/2$	$N(N-1)/2$	N^2
1～ $N-2$	$3(N-1-Layer)$	$3(N-1-Layer)$	$6(N-1-Layer)$
$N-1$	1	0	1

7.1.3. 回転による位置・方向の変化

I 軸まわり第 $Layer$ 層の回転では $i = Layer$ を満たすピースが移動する。他の軸まわりの回転についても同様。それらが Table 7.2 の様に移動する。

$Layer$ が $N-1$ のときは回転層上にあるのが 1 個の順方向ピースだけなので回転後も位置は変わらない。また、N が 3 の倍数でない場合は、第 0 層に回転軸上のピースが存在し、これも回転によって移動しない。それ以外の場合は 3 つずつのピースが巡回置換をうけ、順方向ピース同士、逆方向ピース同士がそれぞれ置き換わる。

I, J, K, L 軸のまわりの正方向の回転行列は(8)の様になる(逆方向の回転の行列はこの転置行列)。回転する層にある全てのピースがこの方向変換をうける。このタイプのパズルではピースの方向ベクトルが I, J, K, L の 4 軸のいずれかを向くのみで、六面体パズルの様に負の方向を向くことがないため、回転の行列要素には負の数は現れない。

Table 7.2 回転によって (i, j, k, l) に移動するピース

	I 軸	J 軸	K 軸	L 軸
正方向	(i,k,l,j)	(l,j,i,k)	(j,l,k,i)	(k,i,j,l)
逆方向	(i,l,j,k)	(k,j,l,i)	(l,i,k,j)	(j,k,i,l)

$$I_+=\begin{pmatrix}1&&&\\&&1&\\&&&1\\&1&&\end{pmatrix}\quad J_+=\begin{pmatrix}&&&1\\&1&&\\1&&&\\&&1&\end{pmatrix}\quad K_+=\begin{pmatrix}&1&&\\&&&1\\&&1&\\1&&&\end{pmatrix}\quad L_+=\begin{pmatrix}&&1&\\1&&&\\&1&&\\&&&1\end{pmatrix}\tag{8}$$

7.2. 表示

7.2.1.　座標系

点の座標系はピースの座標系に準ずる。i 座標はパズルの I 面上で 0 となり I 面に平行な分割面を一つ越える毎に 1 増加するものとする。一般の三次元空間にこのような座標系をとると 1 点に割り当てられる座標が一意に決められないが、正四面体の表面に限定すれば点と座標は 1 対 1 に対応する。

四面体の表面上の隣の格子点への移動では、4 つの座標のうち 1 つが増加, 1 つが減少, 残り 2 つは変化しないので $i+j+k+l=N$ が常に成り立つ。移動方向は正四面体の辺の方向の数 6 だけあり、パズル全体は先に定義した方向を向いているので、整数座標のそれぞれの方向の単位移動に対して 3 次元の実数座標は Table 7.3 に示す方向に同じ距離だけ変化する。

Table 7.3　4 次元座標値と実数座標の変化

座標値	i 増 j 減	i 増 k 減	i 増 l 減	k 増 l 減	l 増 j 減	j 増 k 減
実数座標の変化方向	0, 1, 1	1, 1, 0	1, 0, 1	0, 1, －1	－1, 0, 1	1, －1, 0

この条件から $\Delta X\propto\Delta(2i+2j-k-l)$, $\Delta Y\propto\Delta(2i+2k-l-j)$, $\Delta Z\propto\Delta(2i+2l-j-k)$ が言えるので、パズルの外接球の半径が 1 で原点を中心にもつことから実数座標は $X=\dfrac{2i+2j-k-l-N/2}{3\sqrt{3}/2}$, $Y=\dfrac{2i+2k-l-j-N/2}{3\sqrt{3}/2}$, $Z=\dfrac{2i+2l-j-k-N/2}{3\sqrt{3}/2}$ となる。

7.2.2.　表示・操作系

ピース座標と面の 3 つの頂点の座標の関係が、順方向ピースと逆方向ピースでは異なるので場合分けをする。また、表示の際には 4 つの頂点の Z 座標を調べ、それが負であるものに対向する面のみを表示する。

回転方向は 4 軸の正負で 8 方向あるが、ピース面に垂直な軸まわりの回転を除外して 6 方向のいずれになるかを決定する。そのため、正六面体のパズルの場合と似た手法で三角形の頂点とマウスから取得した点に座標変換を施して原点を中心とした正三角形にし、取得した点が

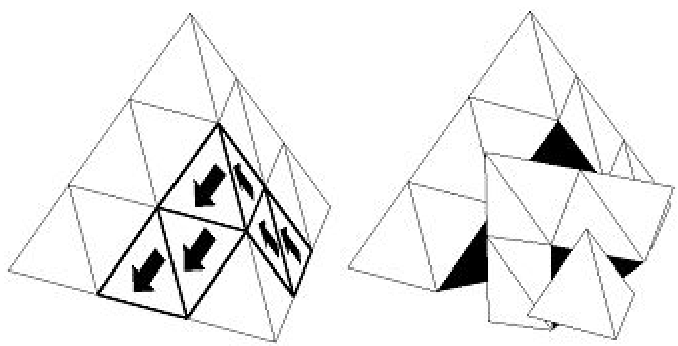


Fig.7.2 回転プレビューとアニメーション

その正三角形を 6 分割したどの領域にあるかを調べる。矢印の表示にはメソッド $InnerPoint$ を正

三角形用に書き換えたもの、すなわち底辺方向の変位と垂線方向の変位を受け取り正三角形内部の点を返すメソッドを使用する。表示するブロックの前後判定は面の前後判定と同じものを使う。断面は常に三角形(実際は正三角形)になるので、頂点にあたる 3 点を標準 $Grid$, 仮 $Grid$ から取得して表示する。

8. 正四面体・辺型パズル

ここで扱うパズルを

- 全体の形が正四面体
- 対向する辺の中心を結ぶ 3 本の軸に垂直な面が全体を N 個に等間隔に分割する
- 単位回転角は 180 °
- ピースは表面の層にのみ存在する

と定義する。 $N=2$ で、単位回転角が 90 ° のものが $Pyramorphix$(これも序論の条件 2 を満たさず、回転させると全体の形が変化していく。構造的には $N=2$ の正六面体・面型パズルと等価である。 $N \geq 3$ になるとピースがばらばらになってしまうので、少なくとも実物を作ることはできない。ここで考えるパズルは単位回転角が 180 ° のものなので正四面体の形を保つ)として販売されている [4]。プログラムでは $2 \leq N \leq 10$ の範囲でピース数をメニューから選択できる様にした。

8.1. ピース情報

8.1.1. 座標系

原点を中心に持ち、頂点 $P_i(p,p,p)$, $P_j(p,-p,-p)$, $P_k(-p,p,-p)$, $P_l(-p,-p,p)$ をもつ正四面体($p>0$)で、P_iP_j の中点と P_kP_l の中点、 P_iP_k の中点と P_jP_l の中点、 P_iP_l の中点と P_jP_k の中点を結ぶ直線はそれぞれ X,Y,Z 軸になる。この 3 軸まわりに回転させるため、YZ 平面, ZX 平面, XY 平面に平行な面で全体を等間隔に分割する。 X,Y,Z 軸方向に分割するので、座標も (x,y,z) で表される。

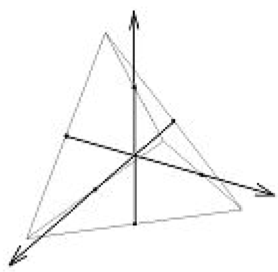


Fig.8.1 辺の中点を結んだ軸

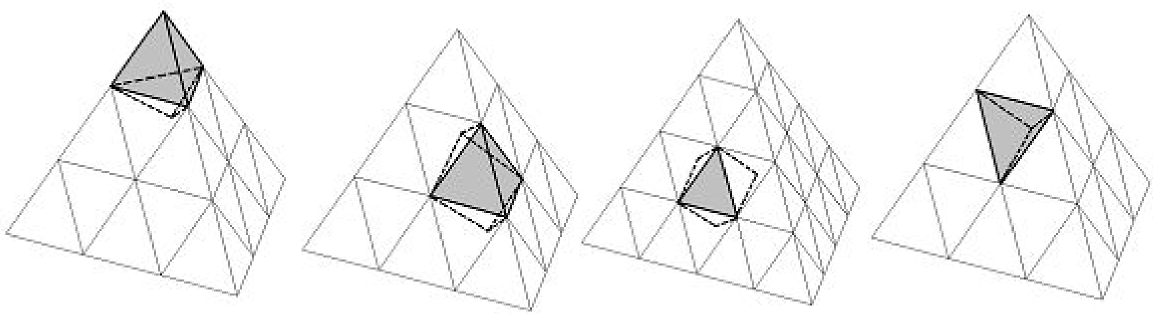


Fig.8.2 順方向ピース(左 1,2,3 番目)と逆方向ピース(右)

順方向ピースは全体に相似な正四面体に、六面体のコーナーの形をした四面体を 1〜3 個貼り付けた形になっていて、1 つの面だけに属するもの($N \geq 4$)から 3 つの面にまたがるものまである。逆

方向ピースは全て六面体のコーナーの形をしていて、属する面は常に 1 つになる。P_i，P_j，P_k，P_l に対向する面を I 面, J 面, K 面, L 面とすると、意味のあるピースの座標 (x, y, z) は I 面では $x+y+z=N-1$(順方向ピース), $x+y+z=N-2$(逆方向ピース)を満たす。同様に J 面では $-x+y+z=N-1, N$、K 面では $x-y+z=N-1, N$、 L 面では $x+y-z=N-1, N$ (順方向、逆方向)を満たす。また、ピースの方向は頂点－面型パズルと全く同じ様にして表される。

8.1.2. 層の回転

回転軸は X ,Y ,Z のいずれか、回転方向はアニメーション表示のために正負の 2 つを考えるが、どちら向きに回転しても結果は同じなのでピース情報の扱いとしては 1 方向のみ。回転層の番号は 0～ $N-1$ の値をとる。回転層上にあるピース数は Table 8.1 の様になる。

Table 8.1 移動層上のピース数

$Layer$	順方向ピース	逆方向ピース	全移動ピース
$0, N-1$	N	$2(N-1)$	$3N-2$
1～$N-2$	$2(N-1)$	$2(N-1)$	$4(N-1)$

8.1.3. 回転による位置・方向の変化

X 軸まわりの回転では x 座標が回転層番号に等しいピースが移動する。Y, Z 軸まわりの回転でも同様。方向の正負はなく Table 8.2 の位置にあるピースが (x, y, z) に移動する。

Table 8.2 回転によって (x, y, z) に移動するピース

X 軸まわり	Y 軸まわり	Z 軸まわり
$(x, N-1-y, N-1-z)$	$(N-1-x, y, N-1-z)$	$(N-1-x, N-1-y, z)$

回転層が 0 か $N-1$ のとき以外は必ずどのピースも対になる相手を持っていて回転はそれらの互換になるが、両端の層の回転では、軸上にあるものだけは位置を変えない。

ピースの方向の変化の行列にも正負はなく(9)の様になる。どれも単純な 2 つの面の入れ替えに相当する。$YZ = ZY = X, ZX = XZ = Y, XY = YX = Z$ が成り立ち、取り得る方向は基本状態の向きとこれら 3 つを単独でかけたものしかなく全部で 4 通り。

$$X = \begin{pmatrix} & 1 & & \\ 1 & & & \\ & & & 1 \\ & & 1 & \end{pmatrix} \quad Y = \begin{pmatrix} & & 1 & \\ & & & 1 \\ 1 & & & \\ & 1 & & \end{pmatrix} \quad Z = \begin{pmatrix} & & & 1 \\ & & 1 & \\ & 1 & & \\ 1 & & & \end{pmatrix} \qquad (9)$$

8.2. 表示

8.2.1. 座標系

正六面体・面型パズルと同様にとる。4 つの頂点は正六面体の頂点と一致するので、外接球の半径を 1 にするためのスケーリングの係数まで全く同じになる。ただし、表面上の点は I 面: $x+y+z=N$, J 面: $-x+y+z=N$, K 面: $x-y+z=N$, L 面: $x+y-z=N$ を満たす。

8.2.2.　表示・操作系

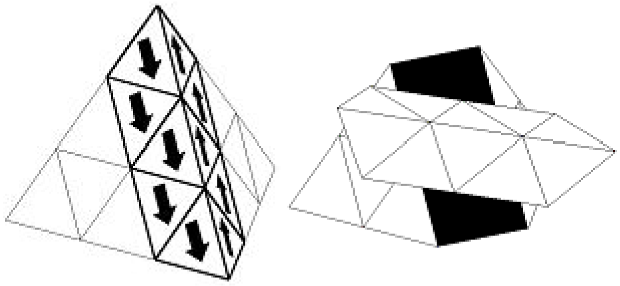


Fig.8.3 回転プレビューとアニメーション

マウスポインタがどの面上にある場合も 3 軸の正負 6 通りすべての方向に回転しうるので、ピース面を 6 等分してマウスから取得した位置をもとに回転方向を判断する。回転軸の方向ベクトルは中心から 2 つの頂点へのベクトルの合成ベクトルから取得する。そのため、回転層になくてもパズル全体の 4 頂点はつねに仮$Grid$ の計算に含める。断面の黒表示は常に平行四辺形(実際は長方形)になるので対応する 4 点を標準$Grid$, 仮$Grid$ から取得する。

9.　正八面体・面型パズル

ここで扱うパズルを

- 全体の形が正八面体
- 面の中心とパズルの中心を結ぶ 4 本の軸に垂直な面が全体を等間隔にN 個に分割する
- 単位回転角は 120 °
- ピースは表面の層にのみ存在する

と定義する。$N=2$ のものが $Skewb\ Diamond$、$N=3$ のものが $Dino\ Octa$ (ただし中央の層が両端より厚いため､台形のピース面や､本来はないセンターピースがある)として販売されている [5)]。プログラムでは$2 \le N \le 10$ の範囲でピース数をメニューから選択できる様にした。

9.1. ピース情報

9.1.1.　座標系

原点を中心にもつ正八面体を考え、その面を垂直に貫く軸I , J , K , L がそれぞれ(1, 1, 1), (1, －1, －1),(－1, 1, －1), (－1, －1, 1)方向を向いているものとする。これらに垂直で軸の正側にある面をI_+, J_+, K_+, L_+、負側の面をI_-, J_-, K_-, L_- と定義する。これらに平行な面がそれぞれ全体をN 等分する。

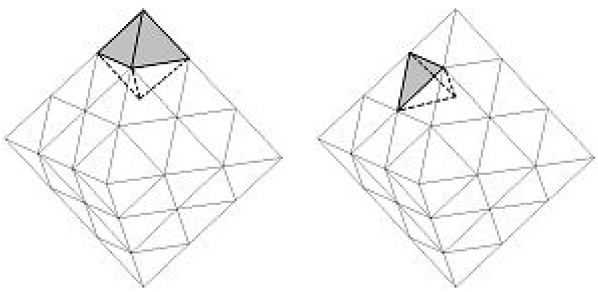


Fig.9.1 順方向ピースと逆方向ピース

このパズルにも順方向ピースと逆方向ピースがある。順方向ピースは全体に相似な正八面体であり、1 つの面だけに属するものと 2 つまたは 4 つの面にまたがるものが存在する。逆方向ピースは一辺の長さが順方向ピースと等しい正四面体で、常に 1 つの面だけに属する。

ピースにi, j, k, l の座標を振り、I_- 面に接しているピースのi 座標が 0 で、I_- 面に平行な分割面を越えるたびにi 座標が 1 増えるものとする。j, k, l 座標についても同様。4 つの座標は 0～$N-1$

の値をとるので、ピースは要素数 N^4 の配列であらわすが、実際のピースに対応するのは $i+j+k+l=2N-2$ (順方向ピース), $i+j+k+l=2N-3$ (逆方向ピース)を満たすものだけである。ピースの方向を表す行列は正四面体・頂点ー面型パズルと同じく 4×4 の行列になる。ピースの取り得る方向も同じく 12 通りのみになる(正八面体群の位数は正六面体群の位数と同じく 24 なので、本来はそれだけの方向をとりうるはずだが、回転の方向が制限されているため)。

9.1.2. 層の回転

回転軸は I, J, K, L の 4 軸で、正負の方向に回転する。回転層は 0〜 $N-1$ の N 層ずつある。Table 9.1 に回転層上のピース数を示す。

Table 9.1 回転層上のピース数

Layer	順方向ピース	逆方向ピース	全ピース
$0, N-1$	$N(N+1)/2$	$N(N-1)/2$	N^2
$1〜N-2$	$3(N-1)$	$3(N-1)$	$6(N-1)$

9.1.3. 回転による位置・方向の変化

正四面体・頂点ー面型パズルと同様に Table 7.2 に示されるピースが (i, j, k, l) に移動する。N が 3 の倍数でない場合、両端の層の回転では軸上のピースは移動しないが、それ以外は層上にある 3 個ずつのピースが巡回置換を受ける。

ピースの方向も正四面体・頂点ー面型パズルと同様(6)に従い変化する。行列要素が全て 0 か 1 なので、どのような回転を組み合わせても I_+, J_+, K_+, L_+ の面、I_-, J_-, K_-, L_- の面の間でそれぞれ置換がおこるだけで、ある面が正負の異なる面を向くことはない。

9.2. 表示

9.2.1. 座標系

4 つの座標軸を頂点ー面型回転パズルと同様にとると、整数座標の変位と実数座標の変位の比例関係もその場合と同じになる。違いは外接球の半径を 1 にするための係数と中心を原点に合わせるためのオフセットの定数のみである。2 つの条件を満たす実数座標値は $X=(2i+2j-k-l-N)/3N$, $Y=(2i+2k-l-j-N)/3N$, $Z=(2i+2l-j-k-N)/3N$ となる。

9.2.2. 表示・操作系

4 つの回転軸のうちマウスポインタの置かれている面に垂直な軸まわりの回転は除外し、残り 3 軸の正負 6 方向を、分割した三角形の領域から判断する。

断面は常に六角形なので、6 つの点を *Grid* から取得して表示する。ブロックの前後を判断するための回転軸のベクトルは面表示の判断に使うものを流用する。

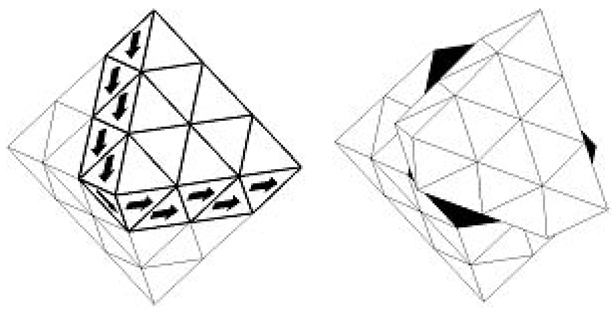


Fig.9.2 回転プレビューとアニメーション

正六面体と同様にI_+, J_+, K_+, L_+の面とI_-, J_-, K_-, L_-の対向する面の表示は排他的なので、それぞれの法線ベクトルを貫通面の 3 頂点から取得して、手前を向いている側だけを表示する。

10. 正八面体・頂点型パズル

ここで扱うパズルを

- 全体の形が正八面体
- 頂点とパズルの中心を結ぶ 3 本の軸に垂直な面が全体を$2N$個に等間隔に分割する
- 単位回転角は 90 °
- ピースは表面の層にのみ存在する

と定義する。$N=3$で第 2, 3 層(辺を挟んだ 2 層)がくっついたものが Magic Octahedron として販売されている [6)]。プログラムでは$2 \le N \le 10$の範囲でピース数をメニューから選択できる様にした。

10.1. ピース情報

10.1.1. 座標系

3 本の回転軸をX,Y,Z軸になる様にとり、それぞれに垂直な分割面が八面体を$2N$個に等間隔に分割する。Xの負の側の頂点を含むピースのx座標を 0 とし、分割面でx座標が 1 増えるものとする。

このパズルにも順方向ピースと逆方向ピースがあり、順方向ピースは立方体のコーナーの形をしており、逆方向ピースは立方体からそれを取り除いた形をしている。どちらも常に 1 つの面だけに属する。ピースの方向を表す行列は面型パズルの場合と同じ。こちらは 24 通り全ての方向を向く。それぞれの面上にあるピースはI_+面で$x+y+z=4N-1$, $4N-2$、J_+面で$x-y-z=1$, 0、K_+面で$-x+y-z=1$, 0、L_+面で$-x-y+z=1$, 0、I_-面で$x+y+z=2N-2$, $2N-1$、J_-面で$-x+y+z=2N$, $2N-1$、K_-面で$x-y+z=2N$, $2N-1$、L_-面で$x+y-z=2N$, $2N-1$ (順方向ピース、逆方向ピース)を満たす。

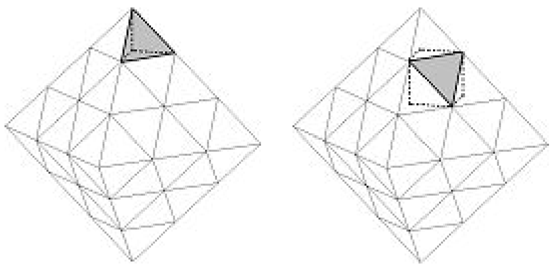


Fig.10.1 順方向ピースと逆方向ピース

10.1.2. 層の回転

Table 10.1 回転層上のピース数

$Layer<N$	順方向ピース	逆方向ピース	全ピース
0～$N-1$	$4(Layer+1)$	$4Layer$	$4(2Layer+1)$
$N-1$～$2N-1$	$4(2N-Layer)$	$4(2N-1-Layer)$	$4(4N-1-2Layer)$

回転軸はX,Y,Zの 3 軸で、正負の方向に回転する。回転層は 0～$2N-1$の$2N$層ずつある。X

軸まわり第 $Layer$ 層の回転では $x = Layer$ のピースが変換を受ける。Table 10.1 に回転層上のピース数を示す。

10.1.3. 回転による位置・方向の変化

回転の仕方は正六面体・面型パズルと同じなので、4 個ずつのピースの巡回置換がおこる。このパズルには軸に貫通されるピースはないので例外はない。

X, Y, Z 軸まわりの回転で方向行列にかかる変換は(10)の様になる(逆方向回転はこの転置行列)。どの回転でも特定の面は隣り合った面に移動するが、接する面の符号は必ず異なっているので、非 0 の行列要素は全て−1 になる。また、それぞれの変換の 2 乗は正四面体・辺型パズル(180°回転)の変換行列(9)になる。

$$X_+ = \begin{pmatrix} & & -1 & \\ & & & -1 \\ & -1 & & \\ -1 & & & \end{pmatrix} \quad Y_+ = \begin{pmatrix} & & & -1 \\ -1 & & & \\ & -1 & & \\ & & -1 & \end{pmatrix} \quad Z_+ = \begin{pmatrix} & -1 & & \\ & & & \\ -1 & & & -1 \\ & & -1 & \end{pmatrix} \qquad (10)$$

10.2. 表示

10.2.1. 座標系

ピースと同様に 3 本の回転軸が X, Y, Z 軸になる様にとる。X 軸上の−側の頂点の x 座標を 0 として、分割面 1 つ毎に x 座標を 1 増加させる。x 座標は YZ 平面では N, +側の頂点では $2N$ となる。この定義にから各面上の点が満たす条件は I_+, I_- 面上にある点で $x + y + z = 4N, 2N$ 、J_+, J_- 面上にある点で $-x + y + z = 0, 2N$ 、K_+, K_- 面上にある点で $x - y + z = 0, 2N$ 、L_+, L_- 面上にある点で $x + y - z = 0, 2N$ となる。

頂点で整数座標 x, y, z が最大、最小になることと外接球の半径が 1 であることから、実数座標 X, Y, Z は(11)の様になる。

$$X = \frac{x - N}{N} \qquad Y = \frac{y - N}{N} \qquad Z = \frac{z - N}{N} \qquad (11)$$

10.2.2. 表示・操作系

回転する方向は 3 軸の正負 6 方向あるのでピース面の三角形を 6 分割して方向を取得する。

ブロックの前後は回転軸のベクトルの向きによるので軸上の頂点の z 座標から判断する。

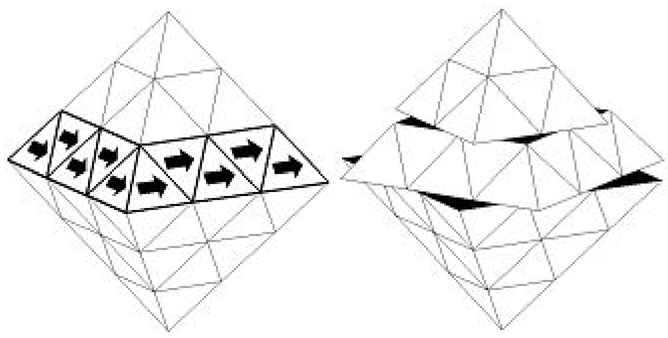


Fig.10.2 回転プレビューとアニメーション

11. 正十二面体・面型パズルⅠ

ここで扱うパズルを

- 全体の形が正十二面体

- 面の中心とパズルの中心を結ぶ 6 本の軸に垂直な面が全体を 2 等分する
- 単位回転角は 72 °

と定義する。該当する実在のパズルは未確認。

正十二面体は 1 つの面が正五角形であるため、今までのパズルとは異なり全体を等間隔で分割すると非相似な面が沢山できてしまう。比較的扱いやすい切り方の一つとしてパズルの中心を含み軸に垂直な 6 つの面でそれぞれ 2 等分するものを選択した [7]。

11.1. ピース情報

11.1.1. 座標系

回転軸を I, J, K, L, M, N とし、ピースには i, j, k, l, m, n の 6 つの座標を振る。軸の方向の単位ベクトル $\mathbf{e}_i, \mathbf{e}_j, \mathbf{e}_k, \mathbf{e}_l, \mathbf{e}_m, \mathbf{e}_n$ が $a = \left(1+\sqrt{5}\right) \big/ \left(2\sqrt{5+2\sqrt{5}}\right)$, $b = \left(3+\sqrt{5}\right) \big/ \left(2\sqrt{5+2\sqrt{5}}\right)$ として Table 11.1 の様になる様に軸をとる。

Table 11.1 軸の方向の単位ベクトル

$\mathbf{e}_i$	$\mathbf{e}_j$	$\mathbf{e}_k$	$\mathbf{e}_l$	$\mathbf{e}_m$	$\mathbf{e}_n$
$(0, a, b)$	$(b, 0, a)$	$(a, b, 0)$	$(0, a, -b)$	$(-b, 0, a)$	$(a, -b, 0)$

ピースの座標は分割面に対して正側にあれば 1, 負側にあれば 0 とする。

ピースには 1 つの面に属する 12 個のセンターピースと、3 つの面にまたがる 20 個のコーナーピースがある。センターピースは正五角錐であり、コーナーピースは 2 つの正三角錐を底面で貼り合わせた形をしている。

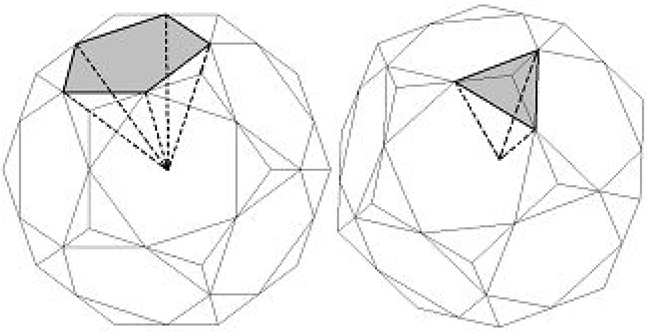


Fig.11.1 センターピースとコーナーピース

ピースの方向は 6×6 の行列で表され、取り得る方向は $12 \times 5 = 60$ 通りある(正十二面体群の位数 = 5 次の交代群の位数 = 5!/2 = 60)。

$$\begin{pmatrix} D_{iI} & D_{iJ} & D_{iK} & D_{iL} & D_{iM} & D_{iN} \\ D_{jI} & D_{jJ} & D_{jK} & D_{jL} & D_{jM} & D_{jN} \\ D_{kI} & D_{kJ} & D_{kK} & D_{kL} & D_{kM} & D_{kN} \\ D_{lI} & D_{lJ} & D_{lK} & D_{lL} & D_{lM} & D_{lN} \\ D_{mI} & D_{mJ} & D_{mK} & D_{mL} & D_{mM} & D_{mN} \\ D_{nI} & D_{nJ} & D_{nK} & D_{nL} & D_{nM} & D_{nN} \end{pmatrix} = \begin{pmatrix} \mathbf{D}_i \\ \mathbf{D}_j \\ \mathbf{D}_k \\ \mathbf{D}_l \\ \mathbf{D}_m \\ \mathbf{D}_n \end{pmatrix} \qquad (12)$$

11.1.2. 層の回転

6 本の軸に対してそれぞれ 2 つの回転層があり、正負の方向に回転する。回転層上にあるピース数は常にセンターピース 6 個、コーナーピース 10 個になる。

11.1.3. 回転による位置・方向の変化

I, J, K, L, M, N 軸まわりの正負の回転で方向行列にかかる変換は(13)の様になる(逆方向の回転はこの転置行列)。回転軸上のセンターピースは移動せず、接するコーナーピース、それに接するセンターピース、さらにそれに接するコーナーピース 5 個ずつが巡回置換を受ける。

$$
I_+ = \begin{pmatrix} 1 & & & & & \\ & & 1 & & & \\ & & & 1 & & \\ & & & & & -1 \\ & 1 & & & & \\ & & & & -1 & \end{pmatrix}
\quad
J_+ = \begin{pmatrix} & & & & 1 & \\ & 1 & & & & \\ 1 & & & & & \\ & & & & & -1 \\ & & & -1 & & \\ & & 1 & & & \end{pmatrix}
\quad
K_+ = \begin{pmatrix} & 1 & & & & \\ & & & & & 1 \\ & & 1 & & & \\ 1 & & & & & \\ & & & -1 & & \\ & & & & -1 & \end{pmatrix}
\tag{13}
$$

$$
L_+ = \begin{pmatrix} & & 1 & & & \\ & & & & & 1 \\ & & & & -1 & \\ & & & 1 & & \\ & 1 & & & & \\ -1 & & & & & \end{pmatrix}
\quad
M_+ = \begin{pmatrix} & & & & & -1 \\ 1 & & & & & \\ & & & 1 & & \\ & -1 & & & & \\ & & & & 1 & \\ & & 1 & & & \end{pmatrix}
\quad
N_+ = \begin{pmatrix} & & & & 1 & \\ & & & -1 & & \\ & 1 & & & & \\ 1 & & & & & \\ & & -1 & & & \\ & & & & & 1 \end{pmatrix}
$$

11.2. 表示

11.2.1. 座標系

表示に必要な点は正十二面体の頂点 20 個と辺の中点 30 個で計 50 個ある。Fig 11.2 に示す様に、中心とその点を結ぶベクトル $\mathbf{P}$ の I 軸への射影 P_i は 7 通りのいずれかになる(t は中心から面に降ろした垂線の長さ)。そこで全ての点に 0〜6 の値をとる整数座標 i, j, k, l, m, n を振る。

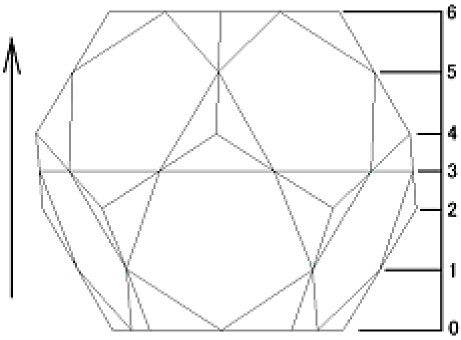


Fig.11.2 I 軸と i 座標

Table 11.2 座標値とその軸への射影

i	0	1	2	3	4	5	6
$P_i(i)$	$-t$	$-\dfrac{\sqrt{5}-1}{2}t$	$-\left(\sqrt{5}-2\right)t$	0	$\left(\sqrt{5}-2\right)t$	$\dfrac{\sqrt{5}-1}{2}t$	t

6 軸方向のベクトルは一次独立ではないため、この射影で表される量(14)と中心から座標点までのベクトルは等しくはないが比例関係にあるので、(14)に $s = \sqrt{5+2\sqrt{5}} \big/ \left(2\sqrt{15}t\right)$ を乗じることで外接球の半径が 1 場合の $\mathbf{P}$ が得られる。

$$
P_i(i)\mathbf{e}_i + P_j(k)\mathbf{e}_j + P_k(k)\mathbf{e}_k + P_l(l)\mathbf{e}_l + P_m(m)\mathbf{e}_m + P_n(n)\mathbf{e}_n \tag{14}
$$

これで整数座標に実数座標が割り当てられたことになる。

11.2.2. 表示・操作系

1 つの面を表示するにはセンターピース 1 つとコーナーピース 5 個を表示しなければならないので、面と整数を受け取り、面全体にあたる五角形の頂点か中点を結んだ五角形の頂点のいずれかを返すメソッドをつくり、受け取った点を組み合わせて五角形と三角形を描画する。

回転候補の取得では、マウスポインタのある面に垂直な軸のまわりの回転を除外し、5 軸の正負 10 方向を決める。小さなピースを分割して判断の基準にすると狭すぎて操作性が悪くなるので、面全体にあたる大きな五角形を 10 分割して方向を決める。回転する層はマウスポインタのある面上のセンターピースを含む 5 つの層の中から選択される。

プレビューの矢印表示には $InnerPoint$ を正五角形用に書き換えたものを使用する。アニメーションでは 1 単位の回転角が 72°になり、ブロックがつねに 2 つで、断面の黒表示の 10 角形が 0, 1 の層で共通になる。回転軸ベクトルは軸に垂直な正側の面の 5 点の座標を加えて取得した。

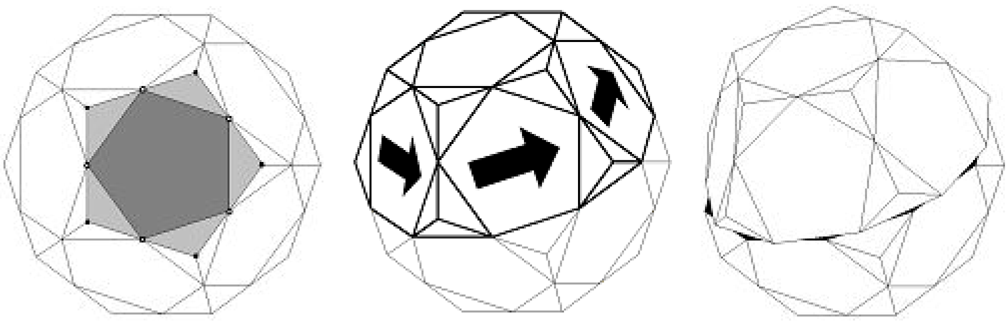


Fig.11.3 面の表示と回転プレビュー･アニメーション

12. 正十二面体・面型パズルⅡ

ここで扱うパズルを

- 全体の形が正十二面体
- 面の中心とパズルの中心を結ぶ 6 本の軸に垂直な 5 つの辺の中点を通る面が全体を 3 つに分割する
- 単位回転角は 72°
- ピースは表面の層にのみ存在する

と定義する。これは $Megaminx$ として販売されている [6]。

12.1. ピース情報

12.1.1. 座標系

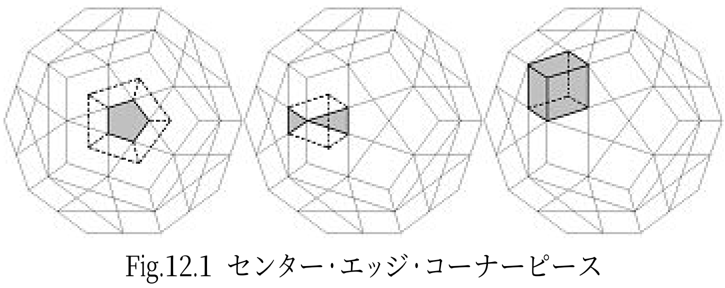


Fig.12.1 センター･エッジ･コーナーピース

座標軸の取り方はタイプⅠと同じだが、1 つの軸に平行な 2 つの面で 3 分割するのでピースの座標は 0～2 の値をとる。この分割の仕方では面の中央に位置し 1 つの面だけに属する 12 個のセンターピース、2 つの面にまたがる 30 個のエッジピース、3 つの面にまたがる 20 個のコーナーピースの 3 種類

のピースができる。センターピースは平行で大きさの違う 1 対の正五角形と 5 つの合同な台形でできている。エッジピースは 2 対の平行な二等辺三角形と台形、2 つの合同な菱形でできている。コーナーピースは 3 対とも平行なので平行六面体である。ピースの方向が 6×6 行列で表されるのはタイプ I と同じ。

12.1.2. 層の回転

6 軸、正負の方向、0～3 層の回転がある。両端の層と中央の層が回転するときでは層上のピース数が異なる。

Table 12.1 回転層上のピース数

Layer	センターピース	エッジピース	コーナーピース
0, 2	1	5	5
1	10	20	10

12.1.3. 回転による位置と方向の変化

どの層が回転する場合でも、ピースが移動する場合は 5 個ずつの巡回置換になる。

第 0 層、第 2 層の回転では回転軸上のセンターピースは移動せず、1 組ずつのエッジピース、コーナーピースが巡回置換をうける(Fig.12.2 左)。第 1 層の回転では 2 組のセンターピース、4 組のエッジピース、2 組のコーナーピースが同様に巡回置換を受ける(左から 2～5 番目)。

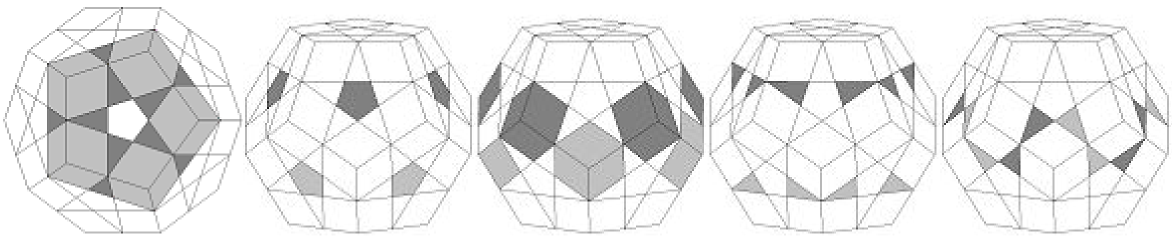


Fig.12.2 層の回転で移動するピース

12.2. 表示

12.2.1. 座標系

Fig.12.3 に示す様に、面内部の五角形の頂点が加わったため、i の取り得る値は 0～8 となる。それぞれの i 座標に対して原点とその点を結ぶベクトル $\mathbf{P}$ の I 軸への射影 P_i は Table 12.2 の様になる。

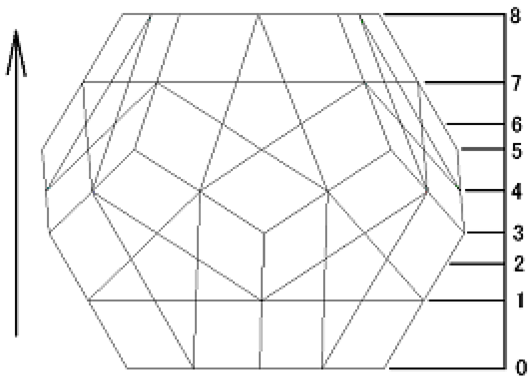


Fig.12.3 I 軸と i 座標

Table 12.2 座標値とその軸への射影

i	0	1	2	3	4	5	6	7	8
$P_i(i)$	$-t$	$-\frac{\sqrt{5}-1}{2}t$	$-\frac{3-\sqrt{5}}{2}t$	$-\left(\sqrt{5}-2\right)t$	0	$\left(\sqrt{5}-2\right)t$	$\frac{3-\sqrt{5}}{2}t$	$\frac{\sqrt{5}-1}{2}t$	t

12.2.2. 表示・操作系

1 つの面を表示するにはセンターピース 1 つとエッジピース、コーナーピース 5 個ずつを表示しなければならないので、面と整数を受け取り、面全体にあたる五角形の頂点、中点を結んだ五角形の頂点、センターピースの頂点のいずれかを返すメソッドをつくり、受け取った点を組み合わせて五角形、菱形、三角形を描画する。

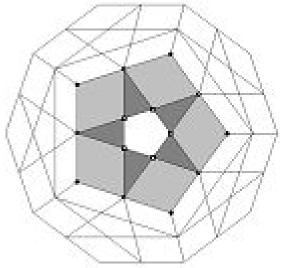


Fig.12.4 面の表示

操作が煩雑になるためと、第 0 層、第 2 層の回転の組合せと等価であることから、第 1 層の回転はマウスによる操作の候補から除外した。判断にはマウスポインタのある面全体を 10 分割したものを使い、その面のセンターピースを含まず、エッジピース 1 つ、コーナーピース 2 つを含む 10 通りの回転から選択する。

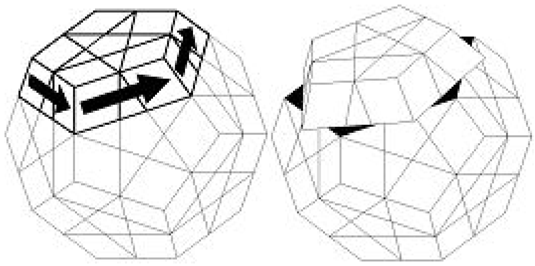


Fig.12.5 回転プレビューとアニメーション

矢印表示の際は $InnerPoint$ に回転面側へのオフセットの値も入れてやり、表示位置を面の中央から回転層側にずらして表示する様にした。

13. その他のパズルの可能性

正多面体を同種の軸に垂直な面で分割してできるパズル 14 種のうち 8 種をプログラム上で再現することができた。

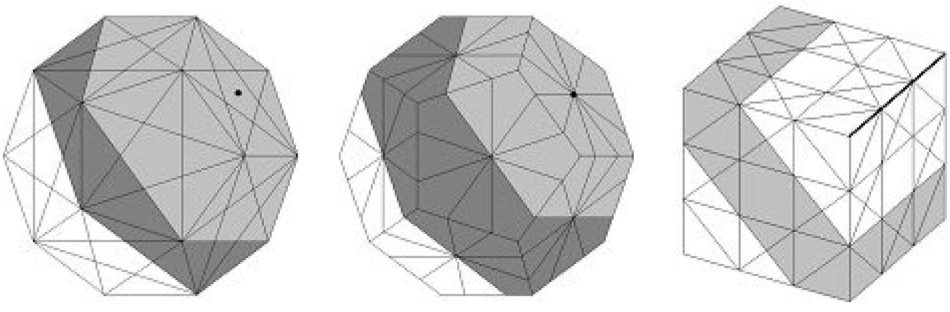


Fig.13.1 正十二面体･面型パズルの他の分割方法と 3 分割した六面体･辺型パズル

正十二面体・面型は 2 種類のものができたが、これ以外にも Fig.13.1 の様に頂点や面の中心を通る面で 3 分割すれば 2～3 種類のピースで構成されたパズルができる [5)]。点の 6 次元座標は左を 0

～7, 中央を 0～9 とすれば全ての点を扱うことができる。また、正六面体・辺型のパズルは 1 つの軸に垂直な面で $2N$ 個に分割したが、奇数個に分けると Fig.13.1.右の様になり(太線の辺の中点と対向する辺の中点を結んだ軸が回転軸)、3 種類のピース面ができる 3)。

正八面体・辺型パズルは Fig.13.2 に示す様に軸に垂直な面でパズルを等間隔に N 個に分割すると規則性をもったピースができる。N が奇数の時は正三角形、二等辺三角形、正六角形のピース面が現れ、N が偶数の時は全ての全てのピース面が合同な直角三角形になる。

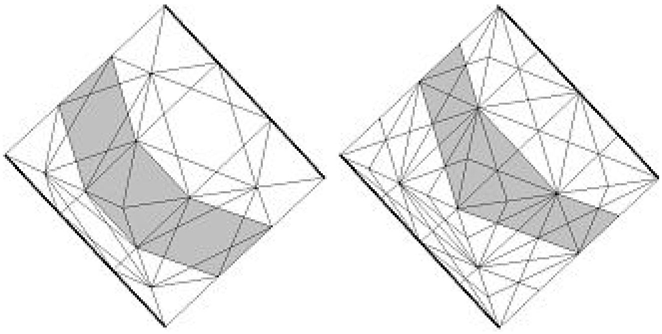


Fig.13.2 正八面体($N = 3,4$)の辺型パズルとその回転層

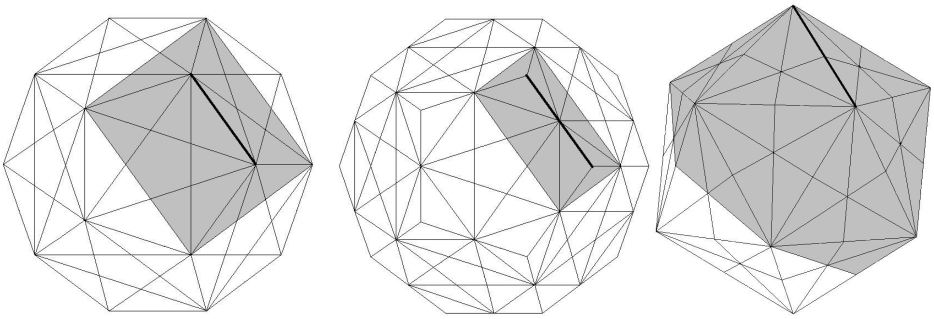


Fig.13.3 正十二面体･二十面体の辺型パズルとその回転層

正十二面体･正二十面体の辺型では分割面を慎重に選ばないと非相似なピース面が多くできるが、Fig13.3 の様に分割面が頂点や辺の中点を通る様にすれば比較的扱いやすいピースだけになる。

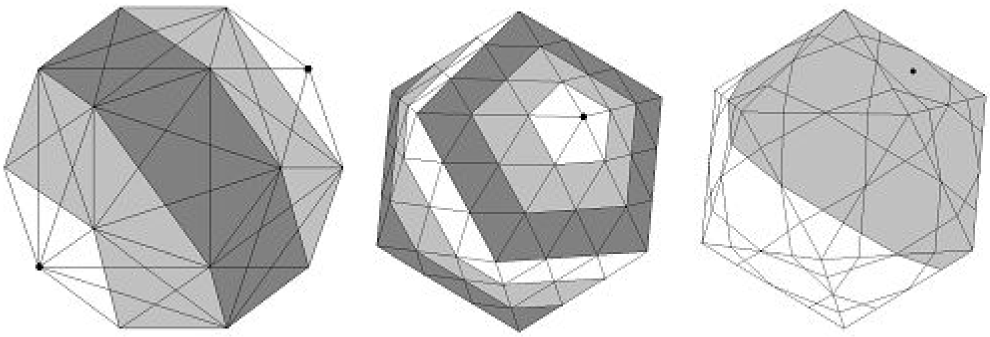


Fig.13.4 正十二面体と二十面体･頂点型、二十面体面型パズルとその回転層

正十二面体・頂点型は 1 つの軸に垂直な分割面を頂点を通るものだけにすると Fig.13.4.左の様に全体が 5 層に分割され、3 種類のピースができる。この 3 つはいずれも 1 つの面にだけ属する。正二十面体・頂点型は全体を等間隔に分割せずに、軸に含まれる 2 つの頂点に接する 10 個の面と中間の 10 個の面の部分でそれぞれ等間隔に分割すればピースの面が全て正三角形になるので、4～8 面体の場合の様にどのような N に対してもパズルを作ることができる(一辺のピース数を N として層数が $3N$)これはどのピースも 1 面だけに属する。正二十面体・面型は全体の中心を含む面で分割すれば Fig.13.4 右の様になり、5 つの面に属するコーナーピース、2 つの面に属し直角三角形 2 つの面を持つエッジピース、正六角形のセンターピースができる。

14. 正六面体・面型パズルの自動解法

パズルの任意の状態Sは基本状態S_0に有限回の回転$R_0, R_1, R_2, \ldots$を施したものなので

$$S = S_0 R_0 R_1 R_2 \cdots \qquad (15)$$

の様に書ける。最短の解を求めることは、これを満たす等価なRの積のうちRの数が最小のものを求めることにあたる。Optimal Solver[8] などの最短手順を求めるアルゴリズムでは、全ての可能なR_iを1回施す度に生成される状態を調べ、それまでに出現したものと等価な状態は除外して、与えられた状態と一致するかどうかをチェックしていくという方法をとる。$N = 3$の場合では可能な回転は3軸×3層×2方向 = 18通りあるので、ターン数が多くなると状態の数は膨大になる。この方法は計算機の能力に大きく依存するので、ここでは人間がパズルを解くときに使われる手法を応用してパズルを解く。

14.1. ツクダ式解法

解法名は 1985 年頃のブーム時にルービックキューブを販売していたツクダオリジナルの社名に基づく[2]。覚えるべき手数とパターンが少ないのでよく初心者に推奨される。ただし、後述の LBL 法に較べてステップ数が多いので完成までに必要なターン数は多くなる。現在ルービックキューブを購入すると$N = 3$のものの解法のマニュアルが同梱されている。

1. ある一面のピースを正しい位置に移動させる。同時に方向もそろえる(こちらをY_-面とする)。
2. Y_+面のコーナーピースを正しい位置に移動させる
3. Y_+面のコーナーピースの方向を変えて正しい向きにする
4. Y_+面のエッジピースを位置・方向とも正しく移動させる
5. Y軸まわりの回転で第 0,2 層エッジ、コーナーピースと第 1 層のセンターピースの方向を揃える
6. Y軸第 1 層のエッジピースを正しい位置に移動させる
7. Y軸第 1 層のエッジピースを正しい方向に回転させる

開発した解法のルーチンではこれを一般のNに適用できる様にするため、1〜7 以外のステップも導入した。

人間がパズルを解く場合は、それぞれのステップで揃えたいピースがどこにあるかを目で探し、その位置と方向に応じて決められた手順(回転操作の組合せ)通りに層を回転させる。この手順はそれ以前のステップで既に揃えたピースの位置や方向が狂わない様に工夫されたもので、一般に後のステップになるほどピース 1 つあたりの手数は多くなっていく。
プログラムの自動ルーチンでもこれに倣い、ピースの位置情報と方向の情報をもとにそのステップで実行すべき手順を決める。この際、どのような状態からでも完成できる様に、それぞれのステップで移動すべきピースが取り得る状態を数え落とすことなく考えておく必要がある。
それぞれのステップでは、

- 移動すべき 1 つまたは複数のピースの位置と方向を調べること
- 手順の実行

の 2 つを行わなくてはならない。3.1.1 で述べた様にピースの現在位置が配列の index、フィールド *Number* が基本状態のピースの位置に対応するので、目的とする *Number* をもつピースを捜してその位置を調べ、方向の行列 D からピースの方向を調べる。手順の実行は文字列のデータを翻訳して実行するメソッドを作成・利用して行う。

```
private void ReadDataTsukuda(){
        // 青面コーナーピース合わせ用
        String[] dataBCorner ={
        // 右下前に正しいピースを-y 面が-Y 方向を向く様に入れる
        "No.0,n,0,n,Y,",                        // 右下前(OK), -y 面が-Y 方向(OK)
        "No.1,n,0,n,-X,xn' yn' xn ynd zn yn' zn' ",   // 右下前(OK), -y 面が X 方向
                ・・・・・・・・・・・
        "No.23,0,n,n,-Z,ynd zn yn' zn'"        // 左上前, -y 面が Z 方向
        };
                ・・・・・・・・・・・
}
```

Fig.14.1 自動解法用データ

すなわち条件と手順を対応させてリストアップしたデータベースをつくり、条件をチェックするメソッドと手順を実行するメソッドを組み合わせることで必要な操作を行わせる。
例えば Fig.14.1 の No.1 の n,0,n は目的のピースの座標が $(N-1,0,N-1)$ であることを表し、−X はピースの方向ベクトル $\mathbf{D}_y$ が X の負の方向を向いていることを表している。その次に続く文字列が一連の回転の操作にあたるコマンドで、空白で区切られた文字列一つ一つが回転に対応している(xn' → X 軸まわり負の方向に第 $N-1$ 層を回転)。ここには出ていないが、全体回転に対応するコマンドでは Xn の様に回転軸を大文字で表記する。ルーチンの性能を比較するため、回転毎にターン数をカウントするが、全体回転は人間でいう「持ち替え」とみなしカウントしない。
No.1 は「目的とするピースが右下前の位置にあり、そのピースの下側色のラベルがパズルの右側の面にあった場合は、最右層を逆回転、最上層を逆回転、最右層を正回転、最上層を 180 ° 回転、最前層を正回転、最上層を逆回転、最前層を逆回転の順に回せ」に対応する。

Table 14.1 文字と座標･層(m は N が奇数の時のみ使用)

文字	0	l	m	r	n
座標・層	0	L	$(N-1)/2$	$N-1-L$	$N-1$

条件指定の座標、回転指定の層などに使われる文字はこの 5 通りで、0, n が両端、m が中央、l,r がそれ以外のピース、層に対応する。$N \geq 6$ では L を $0 < L < N/2$ の範囲で変化させることで全てのピース、層を移動・チェックする。
この様にしてそれぞれの条件に応じて手順を実行するが、コマンドの間違いや条件の見落としが残ることを防ぐため、ステップ毎に「そのステップの全ての場合分け条件が現れるまでシャッフ

ルと解法実行を繰り返し、実行後にそのステップの目的が満たされているか、またそれ以前のステップで揃えた部分が狂っていないかをチェックする」メソッドを作り、どのような状態にも対応できることをチェックしながらルーチンを開発した。

14.1.1. Y_{-} 面センターピース移動

センターピースがない N が偶数のパズルではこのステップは実行しない。
以降のステップの基準となる Y_{-} 面中央にあるべきピースを全体回転で Y_{-} 面に移動させる。あり得るパターンは 6 通りで、目的のセンターピースが元からそこにあった場合は何もしない。

14.1.2. Y_{-} 面インナーピース合わせ

インナーピースがない $N \le 3$ のパズルではこのステップは実行しない。Y_{-} 面にあるべきインナーピースが Z_{+} 面にあり、Y_{-} 面の対応する位置が異色ピースである場合(図左)、既に Y_{-} 面にあるピースの配置を崩すことなくそのピースを Y_{-} 面に移動させる(4 手)。後述の Y_{-} 面などのインナーピースを合わせるステップでは、既に確定したピースの位置を崩さない様にするために 1 つのピースを移動させるのに多くの手数を必要とするが、ここでは比較的自由に動かすことができる。移動先の位置に同色のピースがある場合(図右)は Y_{-} 面を回転させてその位置に異色ピースを移動させ、Z_{+} 面からそこに移動させる(5～6 手)。これを Z_{+} 面にある Y_{-} 側の色のピースがなくなるまで繰り返す。次に Y 軸まわりの全体回転で X_{+} 面であった側を Z_{+} 側に移動させて同様のことを行う。この様にして $X_{\pm}, Z_{\pm}$ 面から Y_{-} 色のピースを全て移動させる。最後に Y_{+} 面に残った Y_{-} 色のピースを Z_{+} 面に移動(3 手)させ、これまでと同様の手順で Y_{-} 面に移動させる。なお、このステップでは全てのインナーピースに対して位置のチェックを行う必要があるので、前述の場合分けとコマンドをセットにしたデータは使用しない。

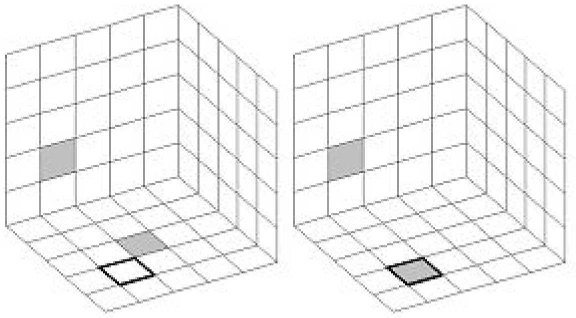


Fig.14.2 インナーピースの移動

14.1.3. Y_{-} 面コーナーピース合わせ

$(N-1, 0, N-1)$ の位置(Fig14.3 の太枠)にあるべきピースを、正しい方向で入れる。このピースの取り得る位置は Fig14.3 に示す 8 ヶ所で方向は 3 通りある(ピースが自由に動ければ方向は 24 通りだが、特定のコーナー上では 3 通りしかとれない)ので、24 通りのパターンについて手順(最大 8 手)を決めておき、合致した条件に従って実行する。この手順では位置と方向を同時に合わせる。残り 3 つの Y_{-} 面上のコーナーピースについても同じ手順を使う。

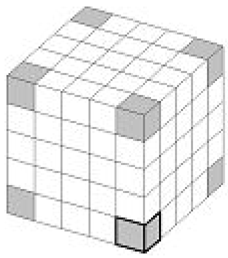


Fig.14.3 コーナーピースの移動

14.1.4. Y_- 面非中央エッジピース合わせ

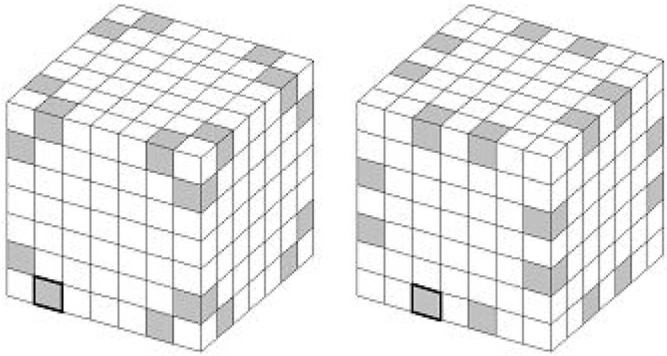


Fig.14.4 非中央エッジピースの移動

非中央エッジピースがない $N \le 3$ のパズルではこのステップは実行しない。$(1,0,N-1)$ の位置(図左の太枠)にあるべきピースを正しい方向で入れる。ピースの取り得る位置は図の 24 ヶ所で、位置に対して方向が決まっているのでチェックすべき条件は位置のみ(逆向きに入っている様に見える場合はこれと対をなす $(N-2,0,N-1)$ にあるべきピースがそこにある)。
同様にして $(N-2,0,N-1)$ を合わせる。1 個目のために 24 通り、2 個目のために 23 通りの場合分けをし、それぞれの条件に合った手順(最大 13 手)を実行する。同様の手順で $(2,0,N-1)$, $(N-3,0,N-1)$ を合わせる(図右)。条件や手順は Table 14.1 の様に文字列をもとに生成しているので L の数値を 1, 2 と変えるだけで図の左右どちらを実行するかを切り替えられる。
N が偶数のときは 1～$N/2-1$, 奇数の時は 1～$(N-3)/2$ の範囲の L についてこれを実行することで Y_- 面と Z_+ 面に接する非中央エッジピースを全て合わせることができる。
さらに Y 軸のまわりに全体回転してこれを他の 3 つの側面に対して適用する。

14.1.5. Y_- 面中央エッジピース合わせ

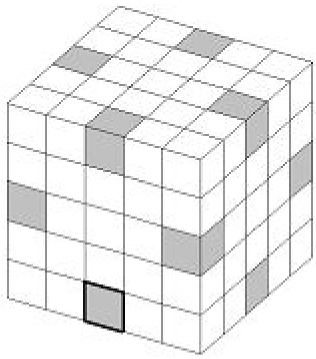


Fig.14.5 中央エッジピースの移動

中央エッジピースのない N が偶数のパズルではこのステップは実行しない。$((N-1)/2,0,N-1)$ の位置(Fig.14.5 太枠)にあるべきピースを正しい方向で入れる。このピースの取り得る位置は図の 12 ヶ所で、それぞれ向きが 2 通りある。この 24 通りについて手順(最大 8 手)を記述し、位置と方向を合わせる。Y 軸のまわりに全体回転してこれを他の 3 つのピースにも適用する。

14.1.6. Y_+ 面インナーピース合わせ

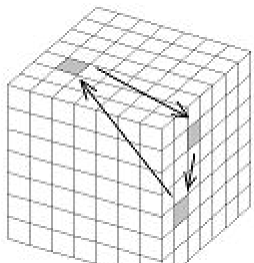


Fig.14.6 面間のインナーピースの置換

インナーピースのない $N \le 3$ のパズルではこのステップは実行しない。$(N-1,j,k) \to (N-1-k,N-1,N-1-j) \to (N-1,k,N-1-j)$ のピースを置換し、他のピースの方向・位置を変えない手順(10 手)を使い、X_+ 面から Y_+ 面へ全ての Y_+ 色のインナーピースを移動させる。Y_+ 面側の対応する位置が既に Y_+ 色ピースであった場合は、手順実行前に

Y_+ 面を回転させることでそのピースが X_+ 面に移動しない様にする。この際、画像処理から得られたデータを元にしていると考えて、人間が解く場合の様にピースの色しか手がかりはないものとし、ピースの *Number* から Y_+ 面のインナーピースであることが判別できるものは全て同類として扱い、Y_+ 面に移動させる。Z_+，Z_-，X_- 面からも同様に移動させて Y_+ 面のインナーピースを全て Y_+ 色にする。

14.1.7.　Y_+ 面コーナーピース移動

Y_- 面側のコーナーピースはすでに位置が確定しているので、Y_+ 面上には既に 4 つの必要なコーナーピースが揃っている。Y_+ 面上にはこの時点でこれらのコーナーピース以外に Y_+ 面を回転させて位置の狂う確定済みのピースはないので、4 つの並びは円順列 $3! = 6$ 通りだけ考えればよい。そのパターンに応じた手順(最大 14 手)を実行してピースを正しい位置に移動させる。

14.1.8.　Y_+ 面コーナーピース方向合わせ

Y_+ 面上のコーナーピースの y ベクトルが Y 方向を向く様に回転させれば良いが、今までに揃えたピースをくずさずに 1 つのピースだけの向きを変えることはできない。可能なのは Fig14.7 の軸のまわりに「1 つのピースがそのままの状態を保ち、3 つのピースが全て正方向または負方向に 120°回転する」という手順(8 手)だけである。y ベクトルが Y_+ 方向を向いている状態を 0, その状態から 120°正方向に回転した状態を 1, 負方向に 120°回転した状態を 2 と表すと、

可能な組合せは Table 14.2 の 8 通りになり、正負の方向への回転は 4 つのうちどれか 3 つに 1 を加えるか 2 を加えることに相当する。これで 4 つの数全てが 3 の倍数になる様にすればよい。

B,C は 1 回、C～H は 2 回で A の状態にできる。解くことのできるパズルである限りこれ以外の組合せは現れない(ルービックキューブをバラバラにしてランダムにピースを入れ直すと多くの場合解けないものができあがる。ここでは基本状態に通常の回転を複数回繰り返して作った状態から解き始めているのでそのようなものは考慮しない)。

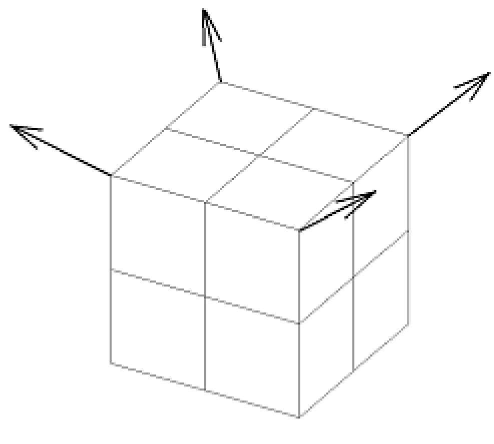


Fig.14.7 コーナーピースの回転軸

Table 14.2 コーナーピースの方向

A		B		C		D		E		F		G		H	
0	0	0	1	0	2	0	1	0	1	0	2	1	2	1	2
0	0	1	1	2	2	2	0	0	2	0	1	1	2	2	1

14.1.9.　Y_+ 面非中央エッジピース合わせ

非中央エッジピースがない $N \leq 3$ のパズルではこのステップは実行しない。

$(1, N-1, N-1)$の位置(図左の太枠)にあるべきピースを正しい方向で入れる。ピースの取り得る位置は図の 16 ヶ所(すでに確定している$Y_$面側にはない)で、Y_+面非中央エッジピースのときと同様に位置だけをチェックする。同様にしてペアのピース$(N-2, N-1, N-1)$を合わせる。

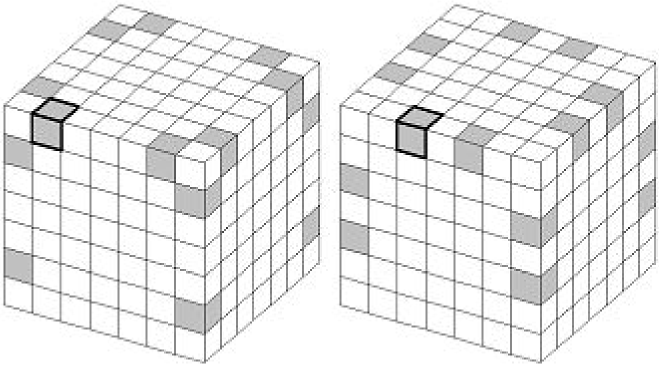


Fig.14.8 非中央エッジピースの移動

1 個目のために 16 通り、2 個目のために 15 通りの場合分けをし、それぞれの条件に合った手順(最大 15 手)を実行する。同様の手順で内側のピースのペア$(2, N-1, N-1)$, $(N-3, N-1, N-1)$を合わせる(Fig.14.8 右)。Nが偶数のときは 1～$N/2-1$, 奇数の時は 1～$(N-3)/2$の範囲のLについて$(L, N-1, N-1)$, $(N-1-L, N-1, N-1)$のペアを順に入れていき、Y_+面とZ_+面に接する非中央エッジピースを全て合わせる。さらにY軸のまわりに全体回転してこれを他の 3 つの側面に対して適用する。

14.1.10. Y_+面中央エッジピース合わせ

中央エッジピースのないNが偶数のパズルではこのステップは実行しない。$((N-1)/2, 0, N-1)$の位置(図左の太枠)にあるべきピースを正しい方向で入れる。このピースの取り得る位置は Fig.14.9 の 12 ヶ所で、それぞれ向きが 2 通りある。この 24 通りについて手順(最大 8 手)を実行し、位置と方向を合わせる。Y軸のまわりに全体回転してこれを他の 3 つのピースにも適用する。

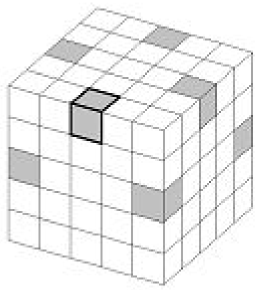


Fig.14.9 中央エッジピースの移動

14.1.11. 非中央Y中間層エッジピースペアリング

非中央エッジピースがない$N \leq 3$のパズルではこのステップは実行しない。非中央のエッジピースは自分と対をなす同色の相手(以後「同種」と呼ぶ)を持つ。このステップでは同種のピースが同じx, z座標を持つ様に並び替える。同種のエッジピースはYの第L層と第$N-1-L$層に 4 対存在するが、同種のピースは基本状態で同じx, z座標を持つので、$Number$の定義(基本状態の座標(x, y, z)から$x + N(y + Nz)$)から、$Number$の差が$N(N-1-2L)$であるものが同種のエッジピースであることが判定できる。このステップでの目的は同種がY方向に並ぶことだけであり、個々のピースが本来のY層にあるかどうかは問わない。つまり、ここでは同種のピースは同じものとして考えてよい。そこで、8 個のピースをa, a, b, b, c, c, d, dの文字で表すことにすると、層の間のピースの分かれ方は Table 14.3 の様になる(この時点で順列は考えない)。また、ここでは同種か異種かだけを問題としているので$a \leftrightarrow b$などの様に文字を交換したものも等価なものとする。

Table 14.3 ピースの 2 層への分かれ方

4 つとも異なる	同じ層に同種が 1 組	同じ層に同種が 2 組
abcd	*abcc*	*aabb*
abcd	*abdd*	*ccdd*

ここで利用できる手順には A(Fig.14.10 左, 6 手), B(右, 9 手)とその鏡映 A', B' がある。同じ層に 2 つの同種ピースがある場合はまず A の手順を使い、同種を 2 つの層に分ける(A, B どちらも層間の移動を含むが手数が少ないこちらを使う)。

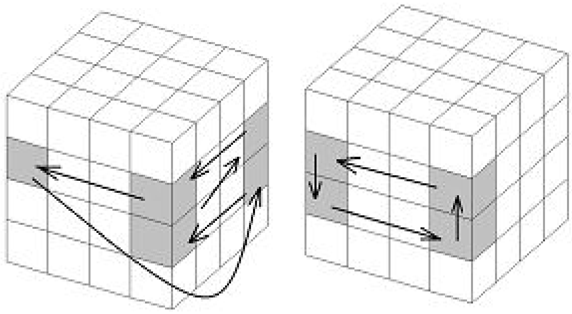


Fig.14.10 エッジピースの交換

2 つの層に同種のピースがありさえすれば、層を互いに回転させることで最低 1 つのペアをつくることができる。1 つのペアが確定すると

a	1	2	3
a	4	5	6

の残り 1～6 の部分への b, b, c, c, d, d の入れ方で場合の数が決まる(表の上下の行は層を表す)。移動させたくない a のペアを Fig.14.10 の奥にして手順を実行するとそれぞれの手順は置換 $A = (1\ \ 2\ \ 3\ \ 5\ \ 6)$, $B = (1\ \ 2\ \ 5\ \ 4)$, $A' = (3\ \ 2\ \ 1\ \ 5\ \ 4)$, $B' = (3\ \ 2\ \ 5\ \ 6)$ にあたる。6 ヶ所に 6 つのものを入れ、そのうち 2 つずつ 3 組が等価であり、3 つの文字の交換に対しても等価なので、場合の数は $6!/(2^3 \times 3!) = 15$ 通りになる。

Table 14.4 ピース配置と移動手順

abcd	*abcd*	*abcd*	*abcd*	*abcd*	*abcd*
abcd	*abdc*	*acbd*	*adbc*	*acdb*	*adcb*
―	A'	A	$A'B$	AB'	A^2A'

abcc	*abcc*	*abcc*	*acbc*	*acbc*	*acbc*	*accb*	*accb*	*accb*
abdd	*adbd*	*addb*	*adbd*	*abdd*	*addb*	*addb*	*adbd*	*abdd*
B'	$B'B$	$B'A$	B^2A'	AA'	$A'A$	B	BB'	BA'

それぞれの場合に Table 14.4 の下の置換をすると 4 つのペアができる(置換は右からかける)。ただし網掛けの並びになった場合は層を互いに回転させて 2 つのペアをつくってから A を施した方が手数が少ない。そこで、この場合分けをする前に「2 つの層を互いに回転させて最もペア数の多い状態にする」という準備をすることでこの 2 つのパターンは 2,3 番目に吸収されるので、13 通りの場合分けをし、手順を実行する。

14.1.12. 非中央 Y 中間層エッジピース位置合わせ

非中央エッジピースがない $N \le 3$ のパズルではこのステップは実行しない。このステップではペアになった Y の第 L 層と第 $N-1-L$ 層のピースを同時に移動させて、4 組のエッジピースのペアの x, z 座標だけを基本状態のものと同じ位置に移動させる。確定したピースに影響を与えずに層

をY軸まわりに回転させることができるので、場合の数(円順列)$3!=6$通りについて手順(最大 16 手)を実行する。これを全ての非中央の層について行い、ペアの位置を合わせる。

これが完了すると、それぞれのエッジピースが正しい位置に納まる場合と、ペアのピースが逆の層に入った状態の 2 通りに分かれる。逆に入ると Fig14.11 右の様に色が反対になる。

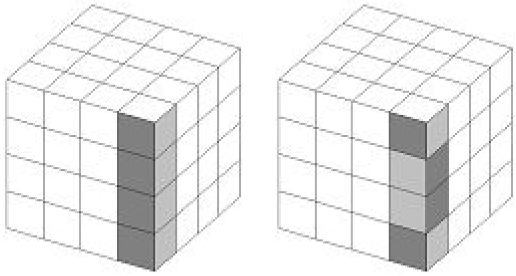


Fig.14.11 正しい状態と逆に入った状態

14.1.13. 非中央Y中間層エッジピース方向合わせ

非中央エッジピースがない$N \le 3$のパズルではこのステップは実行しない。このステップでは逆に入ったピースを交換して全ての非中央エッジピースを正しい位置に入れる。使える手順には隣り合ったエッジのピースを同時に入れ替える手(A ：20 手)と、単独のエッジの層を同時に入れ替える手(B ：21 手)がある。それぞれのエッジでピースが正しい位置にあるものを〇、逆になっているものを×で表し、Y_+側から見たものを表にすると Table 14.5 の様になる。

Table 14.5 エッジピースの正逆

1		2		3		4		5		6	
〇	〇	〇	〇	〇	×	〇	×	〇	×	×	×
〇	〇	〇	×	〇	×	×	〇	×	×	×	×

2 はB, 3 はA, 5 はAB, 6 はAを 2 回施すと 1 の状態にできる。4 はBを 2 回施してもできるが、$X_\pm, Z_\pm$のどれかの面を 180 ° 回転させて 3 の状態にしてからAで方向を変えて、回転された面を元に戻すことで手数を 20 減らす。

14.1.14. 中央Y中間層エッジピース位置合わせ

中央エッジピースがないNが偶数のパズルではこのステップは実行しない。このステップでは中央層(第$(N-1)/2$層)の 4 つのエッジピースを正しい位置に移動させる。非中央エッジピースとは異なり、中央層を回転させるとセンターピースが動いてしまうので、場合の数は円順列では足りない。基本状態のエッジピースを

0	3
1	2

の様に表すとこれらを並べる順列は$4!=24$だが、確定済みの部分に影響を与えない移動は$(1\quad 2\quad 3)$のような 3 個の巡回置換とその組合わせ、すなわち偶置換だけなので、実際には基本状態と同じパリティを持つ 12 通りのみが現れ、その隣同士を置換した並びは出現しない。これらのパターンについてそれぞれ手順(最大 12 手)を実行する。

14.1.15. 中央Y中間層エッジピース方向合わせ

中央エッジピースがないNが偶数のパズルではこのステップは実行しない。このステップでは中央層のエッジピースの向きを正しい方向に変える。

Table 14.6 エッジピースの正逆

1	
○	○
○	○

2	
○	×
○	×

3	
○	×
×	○

4	
×	×
×	×

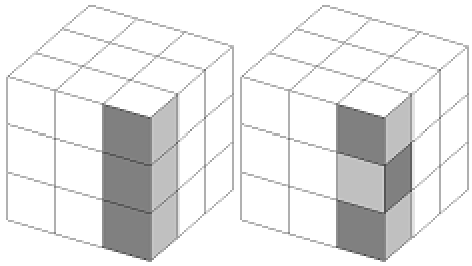


Fig.14.12 正しい状態と逆に入った状態

非中央のエッジピースの場合と似ているが、中央エッジピースは単独で方向を変えることはできないので、奇数個のエッジピースが逆方向を向くことはない。隣り合ったエッジピースの方向を同時に変える手順(14 手)を使って 4 つのエッジピース全てを正しい方向にする。ここまでで$X_\pm, Z_\pm$面のインナーピース以外のピースが全て確定する。

14.1.16. $X_\pm, Z_\pm$面インナーピース合わせ

インナーピースのない$N \leq 3$のパズルではこのステップは実行しない。Y_+面のインナーピースを移動させたときと同様に$(N-1, j, k) \to (j, k, N-1) \to (N-1, k, N-1-j)$をローテーションさせる手順(10 手)を使い、$X_+$面の指定した色のインナーピースを全て$Z_+$面に全て移動させるメソッド$MoveInnerRightToFront$を作り、

全体をZ軸まわりに 180°回転した状態で適用することでX_-面にあるものをZ_+面に移動させる(左)。次に全体を元の状態からY軸まわりに−90°回転させて適用してZ_-面にあるものをX_+面に移動させる(中央)。最後に回転を加えずに

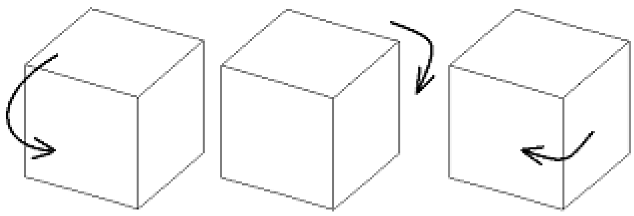


Fig.14.14 Z_+面のインナーピース移動

メソッドを適用してX_+面からZ_+面に移動させることで、全てのZ_+色のインナーピースがZ_+面に集まる。同様にY軸まわり 180°回転(X_-面からZ_-面), Y軸まわり−90°回転(Z_-面からX_+面)の順で適用してX_+面を、Y軸まわり 180°回転(X_-面からZ_-面)で適用することでZ_-面を揃えると、X_-面には自動的にX_-色のインナーピースだけが残る。これで全てのピースの色が正しく入り、パズルが完成したことになる。

14.1.17. 統計

$2 \leq N \leq 10$について、100 回シャッフルした後自動解法を実行して完成までにかかるターン数のデータは以下の通り。(試行回数 100 回の平均と標準偏差) Fig.14.15 の近似式の係数からわかる様に手数はほぼN^2に比例して増える。ピースの数が$N^3 - (N-2)^3 = 6N^2 - 12N + 8$であることからもこれは妥当であると思われる。

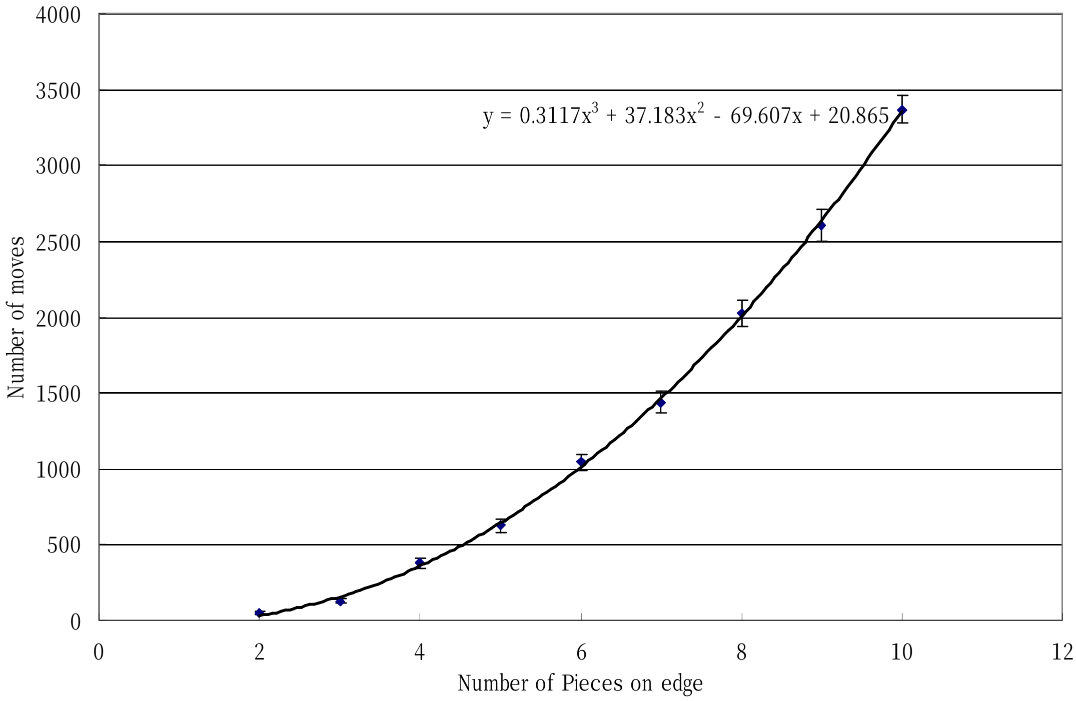


Fig.14.15 一辺のピース数と完成までの手数

Table 14.7 パズル完成までの手数

N	2	3	4	5	6	7	8	9	10
平均	46	124	377	624	1042	1438	2025	2606	3366
標準偏差	8.3	14.5	34.3	40.0	54.3	70.3	84.3	104.2	90.3

14.1.18. 改善案

現在のルーチンではY_-面のコーナーピースを合わせる際に、4 つのピース全てについて同じ手順を使い回している。4 つ目を入れるときには先に入れた 3 つのピースの配置を崩さない様にするために遠回りの手順を使うのもやむを得ないが、1 つめのピースを入れるときには「ありもしないピースを崩さない様に気を使いながら移動」しているため、無駄に手数が多くなっている。段階毎に場合分けすれば少し手数が減るはずである。

またY_+面のエッジピースを入れる際には手数は$X_\pm, Z_\pm$面側からY_+面に移動させるほうがY_+面の別の位置から移動させるより少ない。現在のルーチンでは 4 つのエッジピースを単に時計回りに順に入れているので、目的のピースがY_+面上の他のエッジに収まっていた場合、手数が多くなってしまうが、目的のピースが$X_\pm, Z_\pm$面側にあるものを優先する様にすれば、そこに入っていた別のエッジピースが側面に移動することを利用してトータルの手数を減らせる。

インナーピースを合わせるときに、たとえばある面内の隣り合ったピース 4 つの内 3 つ以上が目的の色であった場合､4つをクラスタにして移動させるなどの手法を使えばより手数を減らせる。

また、Z_-面からZ_+面にインナーピースを移動させる際に一度X_+面を経由しているために手数が多くなっているが、他の配置を崩さずに目的のピースを対面に移動できる手順が見つけられれ

ばここは短縮できる。

N が多い場合にもコーナーピースの数は変わらず、エッジピースは N に比例して増えるのみなので、 N^2 に比例して増えるインナーピースを合わせるステップでの工夫が手数を減らす鍵になると思われる。

14.2. LBL 法

$N=3$ のパズル専用の解法で、Y の第 0 層と第 1 層を先に完成させ、第 2 層を最後に揃えることから LBL(Layer By Layer)法と呼ばれる。完成の速さを競うスピードキュービング大会のトップクラスの記録保持者のほとんどはこの方法に独自の応用を加えて解いている [9]。

1. ある一面のエッジピースを正しい位置に移動させる。同時に方向もそろえる(こちらを Y_- 面とする)
2. Y_- 面のコーナーピースとそれに隣り合う非 Y_- 面のエッジピースを同時に位置･方向とも正しく規定の位置に入れる
3. Y_+ 面のピース全ての y ベクトルに Y_+ 側を向かせる(Y_+ 面の色を揃える)
4. Y_+ 面のピースを正しい位置に移動する

1 には Cross、2 には F2L(First 2 Layer), 3 には OLL(Orienting the Last Layer), 4 には PLL(Permuting the Last Layer)という名前が付いている。

また、それぞれの場合の手順が公式回転記号を使った表記で公開されている。公式回転記号とは、X_+ 面を右、Y_+ 面を上、Z_+ 面を前、右ネジ方向を正として

$L:X$ 第 0 層正回転	$M:X$ 第 1 層正回転	$R:X$ 第 2 層逆回転
$D:Y$ 第 0 層正回転	$E:Y$ 第 1 層正回転	$U:Y$ 第 2 層逆回転
$B:Z$ 第 0 層正回転	$S:Z$ 第 1 層逆回転	$F:Z$ 第 2 層逆回転

で、それぞれの逆方向の回転には ’ をつけてあらわし、180 度の回転は記号の後に 2 をつける。公式回転記号は人間が両手でキューブを回すときに回しにくい方に ’ をつけているので同じ軸まわりの回転を表す記号でも正回転と逆回転が混在している。ツクダ式の解法で用いたコマンドの記法では $UF'U2FU2F'UF$ は yn’ zn ynd zn’ ynd zn yn’ zn’にあたる。ツクダ式のルーチンを考えたときは自前で手順を書き下したのでこのオリジナルの表記を使ったが、公開されている手順をなるべく利用するためこちらは解法の文字列データを公式回転記号で記述し、これを翻訳して実行するメソッドを用意した。

14.2.1. Cross

ツクダ式の面センターピース移動のメソッドをそのまま使って全体回転で Y_- 面のセンターピースを Y_- 側に移動させ、Y_- 面上にあるべき 4 つのエッジピースを正しい位置・方向に入れる。

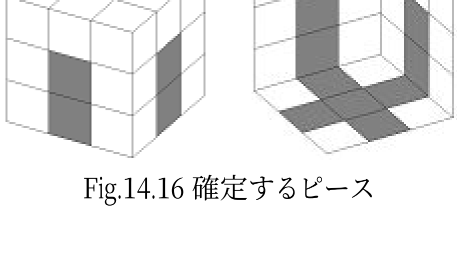


Fig.14.16 確定するピース

基本的にはツクダ式の中央エッジピース合わせと同じ様に 12 ヶ所 2 方向の 24 パターンから目的の位置に移動させる手順を実行するが、手数をなるべく減らすため、何番目のエッジピースを移動するかによって手順を使い分け、無駄な手数を極力抑えた。たとえば 1 つめのエッジピースではセンターピース以外を考慮する必要はないので、目的のピースがY_-面上の別のエッジにあって、yベクトルの向きが正しい場合は単純Y軸周りの全体回転(持ち替え)だけ移動でき、ターン数を増やさずに済む。この場合わけのため手順を 43 通り用意した。
なお、このステップは人間にとっては直感的な作業なので公開されている手順には含まれておらず、場合分けと手順の書き下しは自前で行った。

14.2.2.　F2L

このステップでは(2,0,2)と(2,1,2)にあるべきピース(Fig.14.17 左、太枠)を同時に移動させて正しい位置に入れ、他の 4 つのコーナーピースとエッジピースのペアも同様にあるべき位置に収める。とはいえ、無条件でそれを実行しようとするとコーナーピースは 8 ヶ所×3 方向=24 通り、エッジピースは 8 ヶ所×2 方向=16 通りあるので 384 の場合分けをしなくてはならない。そこで、まずは方向を考慮せずにコーナーピースを(2,0,2)か(2,2,2)の位置(Fig.14.17 中央)に移動させる。この準備は 8 通りの位置のうち 6 通りの場合に短い手順を行うだけで済む(最大 4 手)。
どちらの位置に移動した場合もコーナーピースは 3 通りの方向を向くことができ、それに対してエッジピースは 8 ヶ所×2 方向=16 通りのパターンを持つので$3\times2\times16=96$通りの場合分けをしてそれぞれの場合に規定の手順(最大 13 手)を実行する。

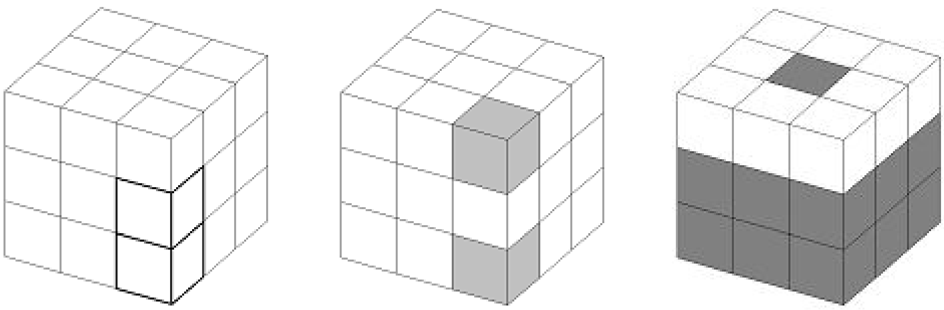


Fig.14.17 移動すべき位置と準備段階の仮移動、確定するピース

なお、手順が公開されている場合分けにはエッジピースが目的の位置以外の非Y_+面のエッジにあるものは含まれていないので、その場合の手順は自前で書き下した。

14.2.3.　OLL

このステップではY_+面のすべてのコーナーピース、エッジピースのyベクトルにY_+側を向かせる、すなわちY_+面側の色を揃えることを目的とする。それに伴いピースの位置が変化するが、次のステップで位置合わせをするのでここではピースの位置については一切考慮しない。
ツクダ式のY_+面コーナーピース方向合わせで見た様に 4 つのコーナーピースの向きの組み合わせは Table 14.2 の 8 パターンある。一方エッジピースはyベクトルがY_+方向を向いているものを 0, 外側を向いているものを 1 であらわすと Table 14.8 の 4 つのパターンだけになる。これを 90

度ずつ回転させたものとコーナーピースの 8 つのパターンを重ね合わせたものがY_+面上のコーナー・エッジピースの取りうる方向状態のすべての組み合わせになる。

Table 14.8 エッジピースの方向

α		
	0	
0		0
	0	

β		
	0	
1		1
	0	

γ		
	0	
0		1
	1	

δ		
	1	
1		1
	1	

α,δは 90 度の回転に対して対称なので Table 14.2 の A～H との組み合わせはそれぞれ 8 ずつしか出てこない。βが 180 度の回転に対して対称であることと、A～H のうち A が 90 度の回転に対して対称であることからβとの組み合わせからは 1+2×7=15 通り、γは 90 度ずつの回転で 4 つのパターンを持つが、A が 90 度, H が 180 度の回転対称性を持つことからγとの組み合わせから出てくるのは 1+4×6+2×1=27 通り。よって組み合わせの数は 8+15+27+8 = 58 通りとなる。このそれぞれのパターンに対して手順(最大 14 手)を実行する。

14.2.4. PLL

このステップでY_+面の全てのピースを正しい位置に移動させる。コーナーピースとエッジピースのグループに分けて考えると、それぞれ$4! = 24$通りの並びが考えられるが、可能な移動はそれぞれのグループに対する偶置換か、それぞれに対する奇置換、すなわちコーナーピース・エッジピースの並びのパリティをどちらも変えないか、同時に変える置換のみである。つまり、並びのパターンは

・コーナー・エッジピースの並びがどちらも基本状態と同じパリティをもつ 12 通り

・コーナー・エッジピースの並びがどちらも基本状態と逆のパリティをもつ 12 通り

のどちらかになる。90°の全体回転はそれぞれについての奇置換に当たり、両方のパリティを変える。ピースを

0	4	3
5		7
1	6	2

の様に表すと、コーナーピースの配置にはこれにC_0(恒等置換), C_1=(0 2)(1 3),C_2=(0 1)(2 3), C_3=(0 3)(1 2), C_4=(0 1 2), C_5=(1 2 3), C_6=(2 3 0), C_7=(3 0 1), C_8=(1 3 2), C_9=(2 0 3), C_{10}=(3 1 0), C_{11}=(0 2 1)の偶置換、C_{12}=(0 1 2 3), C_{13}=(0 3 2 1), C_{14}=(0 2), C_{15}=(1 3), C_{16}=(3 0), C_{17}=(0 1), C_{18}=(1 2), C_{19}=(2 3), C_{20}=(0 2 1 3), C_{21}=(1 3 2 0), C_{22}=(2 0 3 1), C_{23}=(3 1 0 2)の奇置換を加えたもの 24 通り、エッジピースの配置にも同様に$E_0 \sim E_{23}$までの置換(置換の数値はC_iのものに 4 を加えたもの)を加えた 24 通りがある。2 つの順列が同じパリティをもつ組み合わせしか許されないので、12×12×2=288 通りあることになるが、この層をそのまま回転させた状態も等価なので､0 のコーナーピースを固定するという条件を付ければ基本状態にC_0, C_5, C_8と$E_0 \sim E_{11}$, C_{15}, C_{18}, C_{19}と$E_{12} \sim E_{23}$の組み合わせを施してできる 72 通りに絞られる。

さらに、たとえばC_5E_7の後で正方向に 90°回転(0 3 2 1)(4 7 6 5)を加えた変換は$C_{17}E_{19}$になるが、これと$C_{18}E_{16}$を較べると、ピースの配置自体は異なるものの移動するピースの位置関係が同じなので、これを基本状態に戻す手順は(全体回転をしてから行えば)同じものになる。

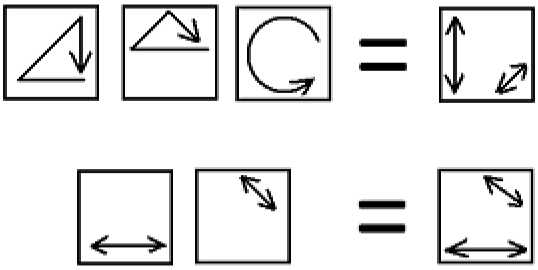


Fig.14.18 ピースの移動と回転

この種のものも同じものに分類すると、状態は 22 通り(コーナーのみの偶置換C_2,C_5,C_8, エッジのみの偶置換E_1,E_2,E_4,E_8, コーナーとエッジの偶置換同士の組み合わせ$C_5E_8,C_5E_9,C_8E_5,C_8E_6$, 奇置換同士の組み合わせ$C_{15}$ ($E_{14},E_{15},E_{18},E_{19}$), C_{18}($E_{14},E_{15},E_{16}E_{17},E_{18},E_{19}$))に分けられる。それぞれの場合について手順(最大 17 手)を実行して位置を合わせれば 6 面の完成となる。

14.2.5.　統計

100 回シャッフルした後自動解法を 100 回試行した。完成までにかかるターン数の平均は 74 回、標準偏差は 6.8 となった。ツクダ式の場合(平均 124 回、標準偏差 14.5 回)に較べかなり短縮できている。場合分けのパターンを多く取りそれぞれの場合について最短の手順を使っていることが短縮の理由の多くを占めると思われる。

14.2.6.　改善案

F2L のステップで、全てのパターンに対応させるため公開されている手順の場合分けに含まれていないものについて手順を書き下した。しかし、たとえば図 14.2.6.1.の様な位置にあるエッジピース(薄網掛け)を目的の位置(太枠)に移動させようとすると、確定したピース(濃網掛け)を移動しない様にするために 11 手、目的の位置の反対側からだと 13 手もの手数を必要とする。なるべくなら他の非Y_+面からの移動を避けた方がトータルの手数は減らせるので、4 ヶ所のペアを揃えていく際に、場合に応じて効率のいい順序で入れていくことができればよい。

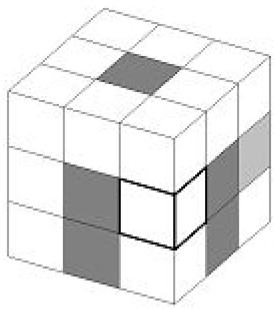


Fig.14.19 非Y_+面からの移動

14.3. 絵柄つきルービックキューブの解法

正規の$N=3$のルービックキューブ以外に、ラベル面に絵柄がついていて、一面に大きな絵柄や 9 個のピース面に同じ絵が描かれているものがある。プログラムでは 3.2.7 で述べたマーク表示を出すことで絵柄と同等のものを再現した。これをツクダ式や LBL 法で解くと、センターピースの方向が周囲のピースと異なって絵がずれてしまうことがある [3]。つまり色だけを頼りに解いたこれらの方法では完全に基本状態にはなっていないということであり、さらにピースの方向を修正する必要がある。

14.3.1.　可能な移動と状態

他のピースの位置･方向を変えずにセンターピースが位置を変えず回転する手順には、1 つの面のセンターピースがその面の法線を軸として正方向に 90°回転し、隣り合った 1 つの面のセンターピースが逆方向に 90°回転する(中央，8 手)手順 1 と、単独の面のセンターピースが 180°方向を変える(右，14 手)手順 2 がある [10]。ここでセンターピースの方向が基本状態と同じかその 180°逆の状態を even，どちらかに 90°ずれた状態を odd と呼ぶことにすると、odd な面は必ず 2 つずつ増減するので可能な状態は Fig.14.21 のどれかになる。

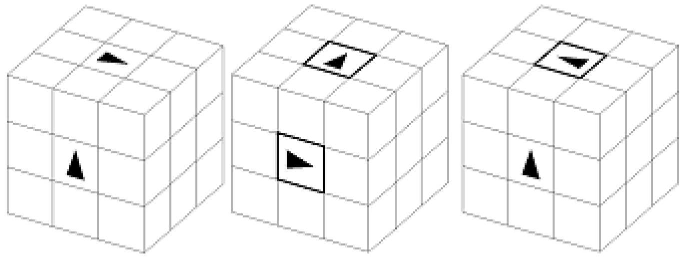


Fig.14.20 ある状態(左)に手順 1，手順 2 を施した状態

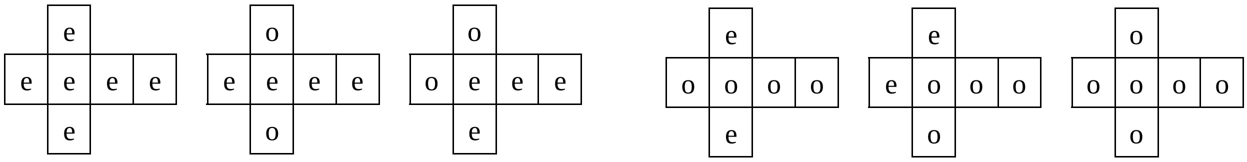
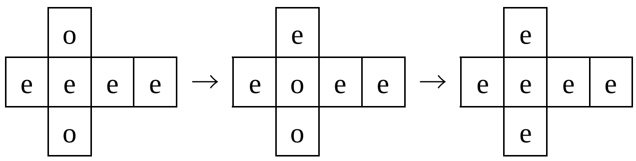


Fig.14.21 センターピースの状態

14.3.2.　odd ピースの even 化

手順 1 を実行すると隣り合った面のセンターピースの方向が変わるので、Fig14.21 の左から 2 番目のパターンのときは 2 つの odd 面に挟まれた 4 つの even 面のうちどれかとセットにして odd 面の一方を even 化し、もう一方とその面をセットにしてすべて even の状態に変える。それ以外の時は隣り合った odd 面同士をセットにして回転させる。また、次の手順での方向転換を最小限にするため、手順 1 とその逆回転を使い分け、手順実行後に少なくとも片方は正しい向きになる様にする。



14.3.3.　180°反転

既に even ピースしか残っていないので、逆向きになっているピースに手順 2 を施して正しい向きに変える。

14.3.4.　統計

100 回シャッフルした後自動解法を 100 回試行した。色が合った状態から完成までにかかるターン数の平均は 42 回、標準偏差は 15.6 回となった。

14.3.5.　改善案

このルーチンは 6 面の色を揃えた後で実行することを前提としている。そのため他のピースを崩

さない様に移動するせいで多くの手数を必要とする。ツクダ式や LBL の途中のステップでセンターピースの方向を条件に入れてさらに場合分けし、その段階でセンターピースの方向を合わせる様にすればトータルの手数は大幅に減らせると思われる。
また、 $N \geq 4$ でツクダ式の解法を行った後はセンターピースだけでなくインナーピースの方向や位置が狂う。これはインナーピースを移動させる際に、同色のピースを区別せずに扱い、さらに方向を考慮せずに移動させていることが原因なので、絵柄からピースの判別がつき、方向も見分けられるものとしてその条件分けを入れれば(ピースの $Number$, 方向データが使える。現在のルーチンでは人間同様見分けがつかないものとしてあえて使っていない)絵柄まで合わせた状態で完成できる。インナーピースを完全に合わせるルーチンの動作を視覚的に確認するには表示系でもピースに絵柄のテクスチャを貼るなどの工夫が必要になる。

14.4. 正四面体・頂点－面型パズルの解法

ピースの名前を定義しておく。
コーナー順ピース: 3 面に属する、パズル全体に当たる正四面体の頂点を含む順方向ピース
コーナー逆ピース: コーナー順ピースに接する逆ピース
エッジ順ピース: 2 面に属する順ピース。 $N = 2$ では存在しない
エッジ逆ピース: 2 面に属する逆ピース。 $N \leq 3$ では存在しない
センターピース:面中央のピース。 N が 3 の倍数＋1 なら順ピース, 3 の倍数＋2 なら逆ピース、3 の倍数のときは存在しない
インナー順ピース:1 面に属するセンターピース以外の順ピース。 $N \leq 4$ では存在しない
インナー逆ピース: 1 面に属する逆ピース。 $N \leq 5$ では存在しない
六面体・面型パズル同様、センターピース同士の位置関係は層の回転によって変化しない。
$N = 2$ のものは 1 つのコーナー逆ピース(八面体)の隣り合わない 4 つの面に 4 つのコーナー順ピースがついているだけなのでパズルというほどのものではなく、コーナー順ピースを回転させて正しい向きにすればそれだけで完成する。ここでは $N = 3$ のものについての解法を考える。

14.4.1. コーナー順ピース位置合わせ

4 個のコーナー順ピース(2, 0, 0, 0), (0, 2, 0, 0), (0, 0, 2, 0), (0, 0, 0, 2)の位置関係は層の回転によって変わらないので、それぞれが指定の位置に来る様に全体回転をする。また、コーナー逆ピース(1, 0, 0, 0), (0, 1, 0, 0), (0, 0, 1, 0), (0, 0, 0, 1)は必ず対応するコーナー順ピースに接しているので順ピースの位置が正しければ自動的にそれらも正しい位置に来る。これは単に持ち替えに相当するのでターン数は増えない。

14.4.2. コーナーピース方向合わせ

コーナー順ピースもコーナー逆ピースも位置が決まれば方向は 3 通りのどれかになる。I 面に対向するコーナー順ピースは I 第 2 層の回転で、逆ピースは第 1 層の回転で他のコーナーピースに影響を与えずに方向を変えられるので、それぞれ独自に合わせる。ターン数は最大でも 1 つのピースにつき 1 手で済む。

14.4.3. I 面のエッジ順ピース合わせ

このステップではI 面に属するエッジ順ピースを合わせる。まず(0, 1, 1, 0)のピースを入れる。

ピースは Fig.14.21 の 6 ヶ所のどこかにあり、方向はそれぞれ 2 通りあるので 12 通りについてそれぞれ手順を実行する(最大 7 手)。(0, 0, 1, 1), (0, 1, 0, 1)についても同様にする。この 3 つを入れるとI 面のピースがすべて揃い、残るのはI 第 1 層の 3 つの順エッジピースだけとなる。

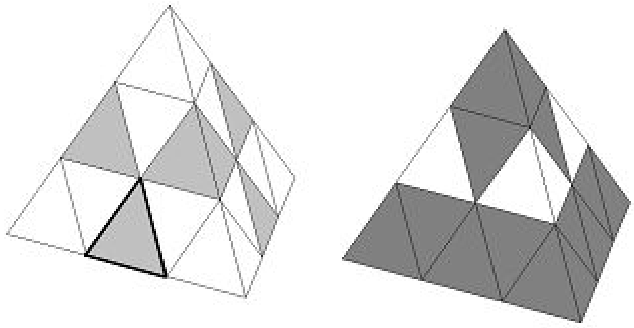


Fig.14.21 I 面エッジ順ピースと確定するピース

14.4.4. I 第 1 層のエッジ順ピース方向合わせ

このステップではI 第 1 層の 3 つのエッジ順ピースの向きを合わせる。位置は次のステップで合わせるのでここでは考慮しない。

既に第 1 層の 3 つのピースは位置の確定していない 3 ヶ所のどこかにあるが、仮に第 1 層を回転させることでそれぞれを規定の位置に移動させた場合に、正しい向きになるか逆になるかをみると、「3 つとも正しい」か「1 つだけ正しく、2 つが逆向き」のどちらかになる。後者であった場合は規定の手順(9 手)でその 2 つも正しい向きに変える。

14.4.5. I 第 1 層のエッジ順ピースの位置を合わせる

このステップは 3 つのピースの位置を合わせてパズルを完成させる。この時点で順列が正しくなっているので、パターンは基本状態と左右どちらかにローテーションした状態の 3 通りしかない。基本状態でなかった場合は規定の手順(11 手)でピースを正しい位置に移動させる。

14.4.6. 統計

100 回シャッフルした後自動解法を 100 回試行した。完成までにかかるターン数の平均は 33 回、標準偏差は 7.8 回となった。

15. パズルの類似性と解法の応用

15.1. 正八面体・頂点型パズルと正六面体・面型パズル

一辺$2N$ の正六面体パズルの面の中央に頂点を接する形で一辺N の正八面体パズルがあると考えれば分割面を共有しているので、基本的には両者は同じ構造をしていると見て良い。違いはピースの取り得る座標で、Fig.15.1 に示す様に$N = 2$の正八面体パズルは$N = 2$の正六面体のパズルのピースに相当する部分と、$N = 4$の正六面体パズルのインナーピースに相当する部分を持つ。それぞれの部分を解くにはツクダ式の解法をそのまま使えば良いが、もう一方を合わせようとすると先に揃えた部分が崩れてしまうので、別な工夫が必要になる。

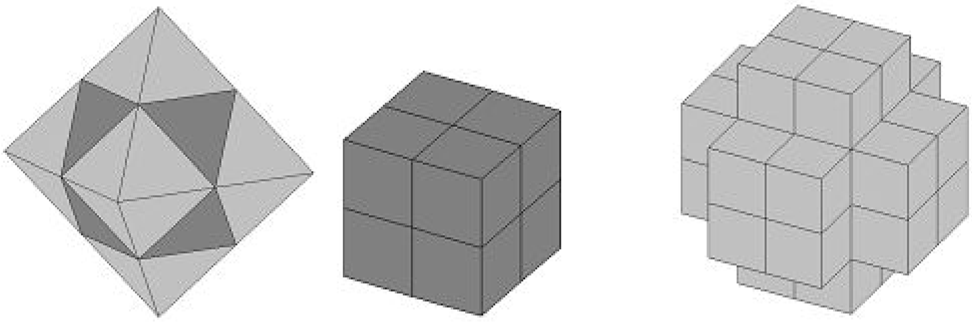


Fig.15.1　$N=2$ の正八面体パズルと等価な正六面体パズル

パズルを考える際にピースが表面上にのみあるとしているためこの様に別扱いになってしまうが、正六面体の中に N^3 個のピースを考えてそれらを全て合わせる解法があれば、それはそのまま正八面体のパズルにも使えるはずである。

一方、正八面体パズルのピースは露出する面が 1 つしかなく、完成の判断は色のみで行うので、Fig 15.2 に示す 3 つの方向は全て同じ扱いとなる。これは解法を考える上で手助けになるかもしれない。

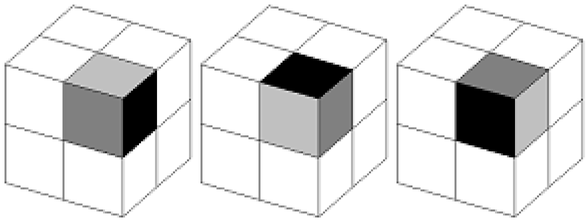


Fig.15.2 正八面体では等価になる 3 つの状態

15.2.　正六面体・頂点型パズルと正八面体・面型パズル

この 2 種類のパズルも回転軸を共有しているので、考慮すべきピースの数を除けば等価なものになる。$N=1$ の正六面体パズルのピースと $N=3$ の正八面体パズル(図では見やすさのため正六面体パズルのピースに対応する 12 個のみ表示しており、そのうち 4 個に色をつけてある)のピースは Fig.15.3 の様に対応している。N が大きくなると互いに対応するもののないピースを持つ様になるが、この場合に限り正八面体パズルが正六面体パズルを含む。正八面体・面型パズルの解法ができれば $N=1$ の正六面体パズルにそのまま応用できるはずである。

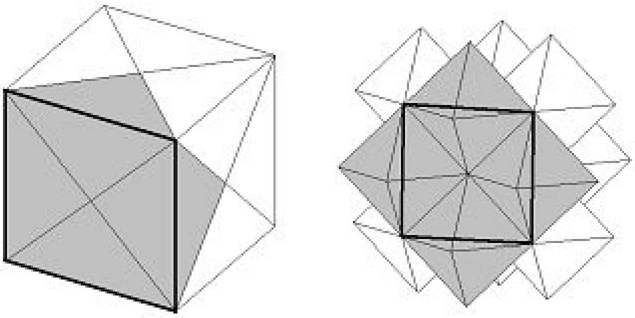


Fig.15.3 正六面体と正八面体パズルのピースの対応

16. 総括

本研究では様々なパズルの可能性を考える際に、正多面体群の持つ頂点の対称性をピースの頂点に応用して、とることのできる回転軸と分割のパターンを数え上げた。ピース自体もこの対称性を持つので、ここで上げた以外のパズルを考える際には原子をピースの頂点かピースに対応させ

ることで結晶学の文献などからヒントが得られるかもしれない。
プログラミングにおいては、ピースの回転・移動の内部的な計算と、表示・アニメーションを切り離して扱うことで、自動解法の開発時の試行がたやすくなった。また、ピースとその頂点に与える整数の座標に規則性が見つけられたことで、これ以外のパズルの再現も少しの応用で容易にできる。
グラフィカルな表示での操作系では、既存の Virtual なパズルで多く見られる、ピースをドラッグして層を回転させる手法をとらず、マウスポインタの位置で回転層と方向を一意に決め、回転方向のプレビュー表示を加えたことで、層の回転とパズル全体のドラッグ回転を混乱することなく操作できる様になった。また他の 3D 系のアプリケーションで例を見ない、視線方向の軸を中心としたパズル全体の回転も可能にしたことでさらに操作性が向上した。
自動解法の開発では、人間的な解法を採用したため場合分けとそれぞれに対する手順の書き下しを行ったが、既存の人間向け解法のガイドを参考にした場合でも、人間が直感的にわかる条件と手順については具体的には示されていないので、全てのパターンを数え上げて手順を書き下すのには注意を要した。追加した手順は最短のものである保証はないため、この推敲だけでトータルの手数を減らせる可能性がある。他の種類のパズルについても解法の示されているサイトがいくつかあるので、ここで採用した「条件と手順の組み合わせ」を使う方法で自動解法が組めそうなものがある。
一方 $N=2$ の正六面体・面型パズルについては、既に基本状態から 14 手で全てのパターンが出尽くすことが証明されているので、Optimal Solver[8)] 的な最短手順を求める手法についても考えてみたい。

17. 謝辞

学科セミナーでの発表の機会を下さった堀端さん、パズルの類似性についての示唆を下さった緑川さんと前田さん、ミラー表示についての示唆を下さった小久保さんと友田さん、研究の発端となった資料を見せてくださった坂井さんにこの場を借りて深く感謝の意を捧げます。

参考文献

1) 物理のかぎしっぽ　http://www12.plala.or.jp/ksp/
2) 頭を鍛えるルービックキューブ完全解析(宝島社) 94 頁 (2005)
3) Rubik’s Cube 展　http://i4no.main.jp/
4) 音と人と情報と 大石研究室のページ　―大同工業大学　情報学部―
http://www.daido-it.ac.jp/~oishi/
5) 立体パズル改造工房 http://puzzle3d.hp.infoseek.co.jp/
6) 立体パズルの部屋 http://homepage3.nifty.com/puzzle_cube/
7) http://byrden.com/puzzles/
8) Optimal Solver , Michael Reid http://www.math.ucf.edu/~reid/Rubik/
9) speedcubing.com http://speedcubing.com/
10) ルービックキューブ攻略
http://www.aomori-u.ac.jp/staff/wajima/