

タイトル	Matlabを用いたルービックキューブのLBL解法
著者	串山， 繁； KUSHIYAMA, Shigeru
引用	北海学園大学工学部研究報告(46)： 119-131
発行日	2019-02-14

Matlabを用いたルービックキューブのLBL解法

串 山 繁*

LBL Solution of Rubik's Cube Using Matlab

Shigeru KUSHIYAMA

要 旨

ルービックキューブはハンガリーの建築学者：Ernő Rubikが1974年に考案し、1980年代初頭に日本で大ブームとなった立体パズルで、 $3 \times 3 \times 3$ タイプのルービックキューブの色の組み合わせ数は約 4.3252×10^{19} 通りある¹⁾。ルービックキューブが如何なる不揃いの配色状態からでも最多で何手で6面揃った状態に戻せるかの手数はGod's Number（神の数）と呼ばれ、ルービックキューブが発売されて以来数学者の関心事であった。その数は、群論に基づくアプローチによりGod's Number=20であることがTomas Rokicki, Herbert Kociemba, Morley Davidson, John Dethridgeにより検証され、それ以上でも以下でもないことが2010年7月に発表されている^{2),3)}。

数学教員志望学生の卒業論文指導として、上記とは異なるアプローチ：ルービックキューブの解法として広く普及しているLBL（Layer-by-Layer）法により単純明快なプログラムを作成したが⁴⁾、その詳細については未公表であったので、本報告で概要について述べる。

1. ルービックキューブの構造

$3 \times 3 \times 3$ タイプのルービックキューブは、図－1に示す様に1つの面が $3 \times 3 = 9$ 分割され、合計 $9 \times 6 = 54$ 面が6色に彩色され、夫々1面体（センターキューブ）6個、2面体（エッジキューブ）12個、3面体（コーナーキューブ）8個の計26個の小キューブから成る。なお、同図は世界基準配色を前面（Front）赤、上面（Up）緑、左面（Left）黄、背面

* 北海学園大学工学部建築学科

* Department of Architecture and Building Engineering, Faculty of Engineering, Hokkai-Gakuen University

(Back) 橙, 右面 (Right) 白, 下面 (Down) 青として眺めた場合を示したものである.

物理的な構造は写真-1 に示す通りで, キューブの中心は26個の小キューブを連結するセンターボールとなっている. センターボールの先端には各面の中心に位置する1面体に取り付けられ, 計6個ある1面体は回転するが相対的位置関係は移動しない. 一方, 辺にある2面体, 頂点にある3面体は自由に回転して位置・向きが変わる仕掛けになっている.

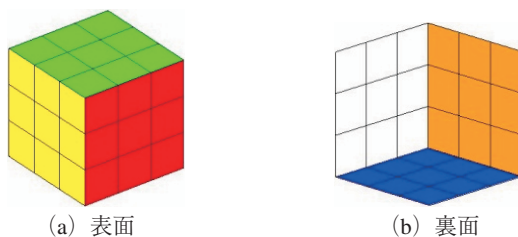
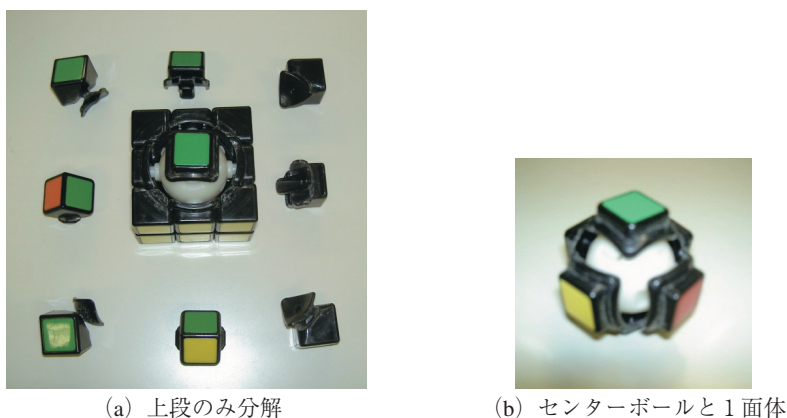


図-1 ルービックキューブ (世界基準配色)



(a) 上段のみ分解

(b) センターボールと1面体

写真-1 ルービックキューブの構造

2. ルービックキューブの色の組み合わせ数

先に述べた様に1面体を除く2面体(12個), 3面体(8個)の小キューブは自由に回転し, 配色可能な組み合わせ数は膨大となる. 以下に組み合わせ数について考える.

2面体についての配置は $12!$ 通り, 更に夫々向きを持つので 2^{12} 通り, 3面体についても同様にその配置は $8!$ 通り, 更に向きは 3^8 通り, したがってそれらを掛け合わせると,

$$12! \times 2^{12} \times 8! \times 3^8 = 519,024,039,293,878,272,000 \text{ 通り}$$

となる. しかし, この組み合わせ数は規則を無視しキューブをバラバラに分解した場合である. 規則に従いキューブをバラバラに分解しない場合には, 以下の3つの制約:

1. ルービックキューブの回転を巡回置換として表すと偶置換となり, 奇置換となる回転が存

$$S_4 = \begin{pmatrix} a & b & c & d \\ b & c & d & a \end{pmatrix} \quad \sim (2)$$

これは、 $a \rightarrow b$, $b \rightarrow c$, $c \rightarrow d$, $d \rightarrow a$ に置換されることを意味している。したがって、単位操作Fは(2)式の表記に倣うと4次の巡回置換5つで、その他の単位操作も列挙すると以下の様に表すことができる。これは図-2に示した展開図を切り抜いて立方体を作れば、容易に確認できる。

$$\begin{aligned} F &= (17 \ 19 \ 24 \ 22), (18 \ 21 \ 23 \ 20), (6 \ 25 \ 43 \ 16), (8 \ 30 \ 41 \ 11), (7 \ 28 \ 42 \ 13) \\ B &= (33 \ 35 \ 40 \ 38), (34 \ 37 \ 39 \ 36), (3 \ 9 \ 46 \ 32), (2 \ 12 \ 47 \ 29), (1 \ 14 \ 48 \ 27) \\ L &= (9 \ 11 \ 16 \ 14), (10 \ 13 \ 15 \ 12), (1 \ 17 \ 41 \ 40), (4 \ 20 \ 44 \ 37), (6 \ 22 \ 46 \ 35) \\ R &= (25 \ 27 \ 32 \ 30), (26 \ 29 \ 31 \ 28), (3 \ 38 \ 43 \ 19), (5 \ 36 \ 45 \ 21), (8 \ 33 \ 48 \ 24) \\ U &= (1 \ 3 \ 8 \ 6), (2 \ 5 \ 7 \ 4), (9 \ 33 \ 25 \ 17), (10 \ 34 \ 26 \ 18), (11 \ 35 \ 27 \ 19) \\ D &= (41 \ 43 \ 48 \ 46), (42 \ 45 \ 47 \ 44), (14 \ 22 \ 30 \ 38), (15 \ 23 \ 31 \ 39), (16 \ 24 \ 32 \ 40) \\ F' &= (17 \ 22 \ 24 \ 19), (18 \ 20 \ 23 \ 21), (6 \ 16 \ 43 \ 25), (8 \ 11 \ 41 \ 30), (7 \ 13 \ 42 \ 28) \\ B' &= (33 \ 38 \ 40 \ 35), (34 \ 36 \ 39 \ 37), (3 \ 32 \ 46 \ 9), (2 \ 29 \ 47 \ 12), (1 \ 27 \ 48 \ 14) \\ L' &= (9 \ 14 \ 16 \ 11), (10 \ 12 \ 15 \ 13), (1 \ 40 \ 41 \ 17), (4 \ 37 \ 44 \ 20), (6 \ 35 \ 46 \ 22) \\ R' &= (25 \ 30 \ 32 \ 27), (26 \ 28 \ 31 \ 29), (3 \ 19 \ 43 \ 38), (5 \ 21 \ 45 \ 36), (8 \ 24 \ 48 \ 33) \\ U' &= (1 \ 6 \ 8 \ 3), (2 \ 4 \ 7 \ 5), (9 \ 17 \ 25 \ 33), (10 \ 18 \ 26 \ 34), (11 \ 19 \ 27 \ 35) \\ D' &= (41 \ 46 \ 48 \ 43), (42 \ 44 \ 47 \ 45), (14 \ 38 \ 30 \ 22), (15 \ 39 \ 31 \ 23), (16 \ 40 \ 32 \ 24) \end{aligned}$$

上記以外の回転操作：S, M, E, S', M', E'は次の様に表示される。

$$\begin{aligned} S &= (4 \ 26 \ 45 \ 15), (5 \ 31 \ 44 \ 10), (1 \ 4 \ 6 \ 2) \\ M &= (7 \ 23 \ 47 \ 34), (2 \ 18 \ 42 \ 39), (1 \ 3 \ 6 \ 5) \\ E &= (21 \ 29 \ 37 \ 13), (20 \ 28 \ 36 \ 12), (3 \ 4 \ 5 \ 2) \\ S' &= (4 \ 15 \ 45 \ 26), (5 \ 10 \ 44 \ 31), (1 \ 2 \ 6 \ 4) \\ M' &= (7 \ 34 \ 47 \ 23), (2 \ 39 \ 42 \ 18), (1 \ 5 \ 6 \ 3) \\ E' &= (21 \ 13 \ 37 \ 29), (20 \ 12 \ 36 \ 28), (3 \ 2 \ 5 \ 4) \end{aligned}$$

更に, Rw, Lw, Fw, Bw, Uw, Dwの2段を纏めて回す操作および(r), (u), (f)の持ち替え操作とそれらに'を付けた逆回転の操作もあるが、いずれも上記に表示した単位操作の積として表示することができる。

なお、任意の置換は巡回置換の積として表せ、n次の巡回置換は(n-1)個の互換の積に分解できる。互換とは文字列中の2つのみを入れ替え、それ以外は動かさない並び替えである。

(2)式の巡回置換は、次の様に互換の積で表される。

$$S_4 = (c \ d)(b \ c)(a \ b) \quad \sim (3)$$

$$\begin{aligned} \therefore (c \ d)(b \ c)(a \ b) &= \begin{pmatrix} a & b & c & d \\ a & b & d & c \end{pmatrix} \begin{pmatrix} a & b & c & d \\ a & c & b & d \end{pmatrix} \begin{pmatrix} a & b & c & d \\ b & a & c & d \end{pmatrix} \\ &= \begin{pmatrix} a & b & c & d \\ a & c & d & b \end{pmatrix} \begin{pmatrix} a & b & c & d \\ b & a & c & d \end{pmatrix} \begin{pmatrix} a & b & c & d \\ b & c & d & a \end{pmatrix} \end{aligned}$$

偶数個の互換の積として表される置換を偶置換といい、奇数個の互換の積として表される置換を奇置換という。したがって、4次の巡回置換は奇置換である。また、偶置換と偶置換の積および奇置換と奇置換の積は偶置換であり、偶置換と奇置換の積と奇置換と偶置換の積は奇置換となる。

ここで、ルービックキューブの小キューブの移動を考える。図-3は、1面体が固定されているとして、2面体、3面体についてのみ識別番号を付番したものである。図-3に単位操作Fを施す場合は次の様に表示できる。

$$\begin{aligned} F &= \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ 3 & 10 & 15 & 4 & 5 & 6 & 7 & 8 & 2 & 14 & 11 & 12 & 1 & 9 & 13 & 16 & 17 & 18 & 19 & 20 \end{pmatrix} \\ &= (1 \ 3 \ 15 \ 13)(2 \ 10 \ 14 \ 9) \quad \sim (4) \end{aligned}$$

(4) 式は4次の巡回置換の積であるから奇置換×奇置換で偶置換となることが分かる。

その他の単位操作も同様に偶置換となり、ルービックキューブにおいて奇置換を生起することは無い。

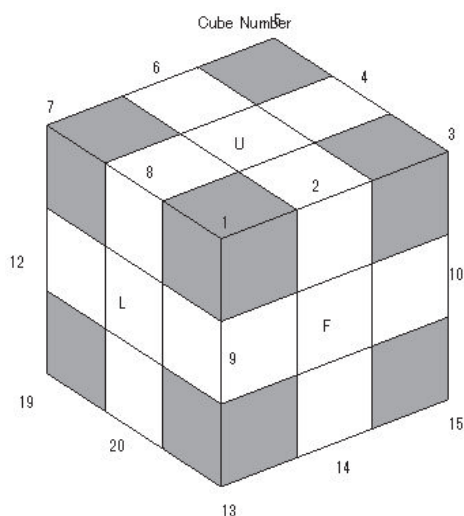


図-3 小キューブの付番

また、単位操作は巡回置換を元とする集合から成り、その意味で対称群といえる。対称群は次の性質を有している。

1. 2つの並べ替えを合成しても既知の並べ替えに等しいものがある。即ち、この集合は演算

の合成に対して閉じている。

2. 結合則が成立する。
3. 単位元（順番を何も変えない操作）が存在する。
4. 逆元（元に戻す逆の並べ替え）が存在する。

4. ルービックキューブのモデル化

群論の文脈によれば、解くべき問題はルービックキューブ群のHTM (Half-Turn Metric) に関するCayleyグラフの直径（即ち、頂点間の最大エッジ距離）を決定することである³⁾。しかし、数学を専門としない者にとってはこの種のアプローチは難解であるため、本報告ではルービックキューブの解法として広く普及しているLBL (Layer-by-Layer) 法を用いることとした。

群論に特化した言語として計算機代数システムGAPをインターフェースとして使用するSAGEが提供されているが、本報告では広く普及している言語を用いて分かり易いプログラムを作成する方針の下にMATLAB言語を使用することとした。

MATLABは、patchコマンドを用いてルービックキューブを図化できる。このために必要なデータは、次の3つ：①キューブを $3 \times 3 \times 3$ に分割するメッシュ交点の座標値、②その交点番号による小キューブ面の定義、③小キューブ面のカラーデータである。ここで、最初の2つのデータは、ルービックキューブに操作を施してもキューブと一緒に動かず、常に最初の位置に固定され、3番目のカラーデータのみがルービックキューブの操作に伴い変わるものと考えらる。小キューブの面番号は先の図-2に示した通りであり、小キューブを規定する小キューブ四隅の頂点番号は、図-4に示す通りとする。

解法のポイントは、特定キューブの位置・向き・カラーの把握を如何に行うかである。これ

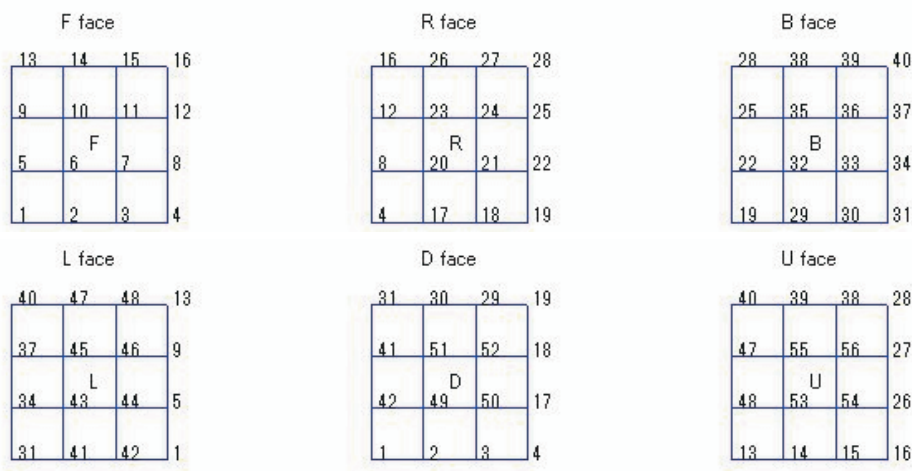


図-4 小キューブ四隅頂点番号の設定

を把握できれば、特定の位置に特定のキューブをLBL法の手順に従い移動すればよい。本プログラムにおいては、図-5、6に示す様にキューブの位置・向きを設定した。

カラーデータについては、初期値として6面体のカラーが揃った状態（以後、ホームポジション状態と呼ぶ）を与えている。カラー行の並び順はU, L, F, R, B, Dであり、先に2, 3面体について与え、最後に1面体について与えた。なお、計算開始時のランダムなカラー配置は、2通りの方法：マニュアル入力、自動生成入力を用意した。図-7はマニュアル入力画面の一例、図-8は図-7に対応した入力データ例（データ入力訂正例を含む）である。

自動生成入力は、最初にホームポジション状態を作成した後、それに対して毎回異なる順序で連続した54種類の単位操作を施して生成した。

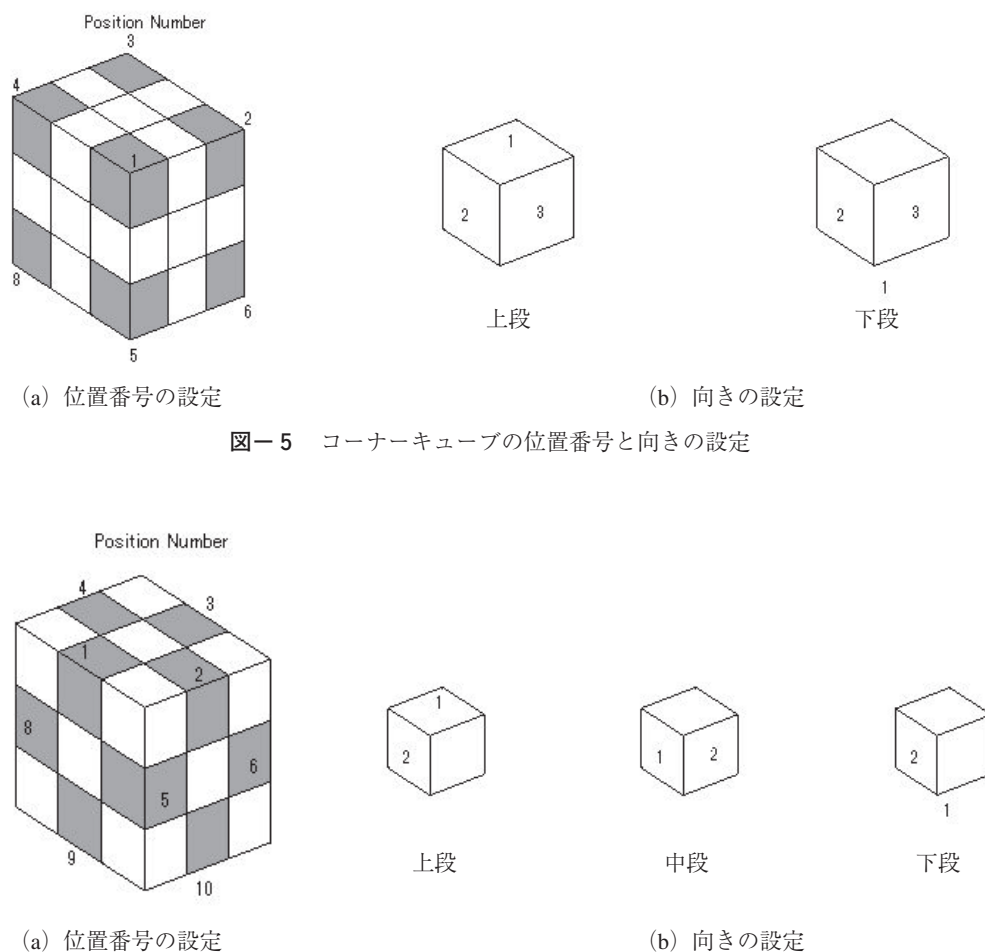


図-5 コーナーキューブの位置番号と向きの設定

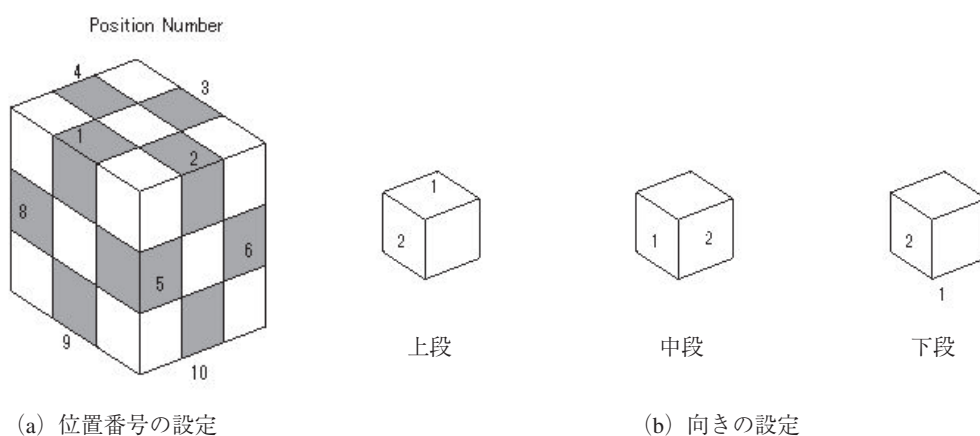


図-6 エッジキューブの位置番号と向きの設定

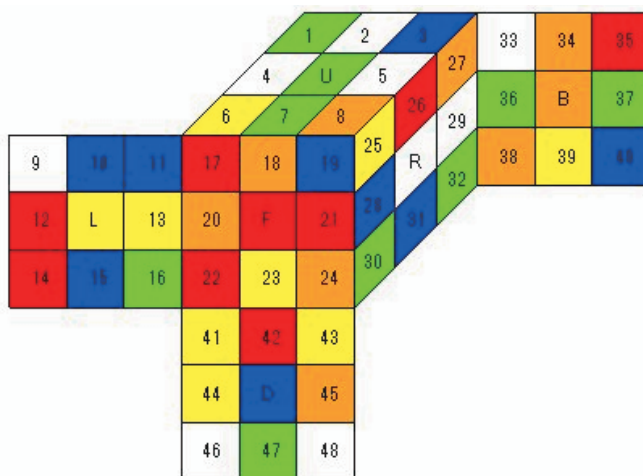


図-7 マニュアル入力データ例

```
>> manual_input_data
Please input Green Color number in [ ]=[1 7 16 30 32 36 37 47]
Please input Yellow Color number in [ ]=[6 11 23 25 39 41 43 44]
Please input Red Color number in [ ]=[12 14 17 21 22 26 35 42]
Please input Orange Color number in [ ]=[8 18 20 24 27 34 38 45]
Please input Blue Color number in [ ]=[3 10 13 15 19 28 31 40]
Please confirm the color arrange!! if there are no mistakes -->0, not yet -->1 1
Please input a mistake number in [ ]=[11 13]
Please input a correct color enclosing G,Y,R,O,W,B with single quotation in [ ]=['B' 'Y']
Please confirm the color arrange!! if there are no mistakes -->0, not yet -->1 0
>>
```

図-8 マニュアルデータ入力 (図-7 に対応, データ訂正を含む)

5. ルービックキューブ解法プログラム

本プログラムは、マニュアル入力状態或いは任意のランダムなスタート状態からホームポジション状態へ導く解を求めるプログラムである。

(1) メインプログラム (Rubik_Cube_Solution_Strategy.m)

まず、メインプログラムの入力データ等について説明する。計算は、次の4つのデータ: cal_cont_flag, figure_flag, ope_disp_flag, cal_numbを用いて指示される。

cal_cont_flag= 0 or 1

= 0 : スタート状態を自動生成してファイルに保存後、実行 (連続計算時使用)

= 1 : 保存スタート状態をファイルから読み込み、実行 (マニュアル入力データ時或いはデバッグ時使用)

figure_flag= 0 or 1

= 0 : ホームポジションに至る単位操作毎の3Dキューブ表示無し

= 1 : ホームポジションに至る単位操作毎の3Dキューブ表示有り

ope_disp_flag

= 0 : ホームポジションに至る一連の操作のみ表示

= 1 : ホームポジションに至る各単位操作詳細情報の表示あり (デバッグ時使用)

cal_numb : 連続計算回数 (ただし, マニュアル入力データ作成時或いはデバッグ時は1とする)

上記データ読み込み後はホームポジション状態を表示した後, cal_cont_flagデータに応じて, 計算開始状態を自動生成するか保存スタート状態をファイルから読み込むかのいずれかへ進み, 計算開始状態のキューブを3D表示する. なお, 任意のランダムな計算開始状態のカラー配置は, 54種類の単位操作の順番をrandperm関数に次の2行を追加して毎回異なる様に生成している.

```
s = RandStream.create('mt19937ar','seed',sum(100*clock));
```

```
RandStream.setDefaultStream(s);
```

それ以降については, LBL法に従った各ステップのfunction.mファイルの実行に移る. 最後に, プログラム末尾で平均計算時間, 平均操作手数等を出力して終了である.

(2) function.mファイル (サブプログラム)

function.mファイルのファンクション名およびそれらの機能は次の通りである. なお, 字下げしたファンクション名は字下げ前のfunction.mファイルにぶら下がっていることを意味する. また, ファンクション名末尾に*)が付随しているのは, そのファンクション名の下に本来記すべきOperation, face_combinationを省略したことを意味している.

なお, LBL法に従った解法パターンについては, 文献5), 7), 8)を参照した.

manual_input_data : マニュアルデータ入力 (メインプログラムとは独立)

color_def_save : 2面体, 3面体および1面体のカラー配置をファイルに書き込み

init_perif_switch : スタート時のランダムなカラー配置の作成

mod_randperm : ランダムな操作手順の生成

Rubik_Cube_operation : ルービックキューブの各単位操作の定義

perif_switch : 2面体, 3面体のカラー配置巡回更新

center_switch : 1面体のカラー配置巡回更新

face_combination : 操作後のカラー並び順番の把握

color_def_save : (前出)

color_def_read：2 面体，3 面体および 1 面体のカラー配置をファイルから読み取り

Rotation_Judge：前面赤・上面緑・左側面黄となる視点から眺めた表示とする調整

adjust_Uface：上段キューブの色，向きを揃える

Operation：一連の操作を単位操作に分割し，操作毎の結果を図化

face_combination：（前出）

adjust_middle_edgface^{*)}：中段の 2 面体を揃える

search_object_position_aspect：始点の位置，向きを探索

adjust_Dface_only^{*)}：下段を反転した上段青面についてカラーのみを揃える

（向きについては無視）

adjust_Dface_corner_aspect^{*)}：青面を前面にしF面の 3 面体コーナー向きを揃える

count_corner_agree：青面を前面にして位置・向きが揃っているコーナー

キューブの個数確認

agree_0^{*)}：青面 3 面体コーナーキューブ 4 個未完成の対応

count_corner_agree：（前出）

agree_2：青面 3 面体コーナーキューブ 2 個完成，2 個未完成の対応

ag_two_110：対角スラッシュ→バックスラッシュに変換

（図－9 参照）

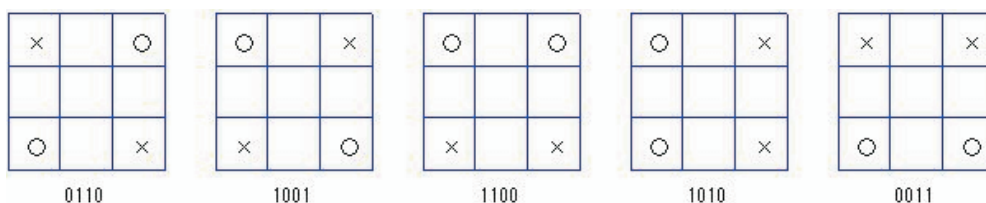
ag_two_1001：バックスラッシュ方向に 2 個一致している状態への対応

ag_two_1100_1010：隣接して上，左縦で 2 個一致している状態への対応

ag_two_11：隣接して下で 2 個一致している状態への対応

adjust_Dface_cross_aspect^{*)}：青面エッジキューブの向きを揃える

count_cross_agree：青面 2 面体の向き確認チェック



○：位置・向きが一致している場合，×：位置・向きが一致していない場合

図－9 青色F面コーナーキューブ位置・向きの一致個数

6. 実行結果

まず，ノートPCを用いて得られた連続計算結果の一例について述べる．表－1 に計算平均所要時間，平均単位操作手数等を示す．ただし，この計算平均時間とはランダムなスタート状

態設定後の一連の単位操作手順を求める時間であり，連続計算は10,000回で且つ解法操作手順のみを表示する場合である．同表によれば，平均計算時間は約0.135 (sec) であり，素早く解が得られることが分かる．ただし，単位操作毎の3Dキューブ表示を行うと20~30 (sec) と非常に長い時間を要する．平均単位操作手数は，解法パターンの場合分け数を極力少なく抑えて（本プログラム例では持ち替えを除くと実質的には総数¹⁹⁾プログラムをシンプルに分かり易くしたため，約160手と多くなった．この原因の一つとして，例えば一連の操作中でD D, D 2 D, U' U, U' 2 U, U 2 U等とあるのをプログラムを分かり易くするため敢えてD 2, D', 単位元, U', U'等と短縮していない箇所があることが挙げられる．これを短縮した場合には表中の値よりも15~25手減少する．なお，表中にadjust_middle_edgeface異常発生率とあるのは，中段のエッジキューブを揃える箇所での異常発生（キューブの移動が4回を超える循環ループに陥ると異常と判定）回数を把握する目的で出力した．異常発生の可能性を否定出来ないのは，ルービックキューブの色の組み合わせ数：約 4.325×10^{19} の全てについて確認することは非現実的で，内一部のケース（約30,000回）についてのみ確認したに過ぎないからである．

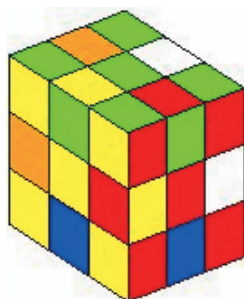
次に，非常に多くの手数を要することが予想されるスーパーフリップについて実行した．スーパーフリップとは，全てのエッジキューブ位置は揃っているが向きが反対である図-10に示した状態である．計算結果の手数は210手となった．なお，計算時間は各単位操作の図を全て表示した場合35.2546 (sec)，その図を表示しない場合0.552632 (sec) であった．

なお，MATLABと互換性のあるフリーソフトであるGNU Octave-3.6.4をスーパーフリップ

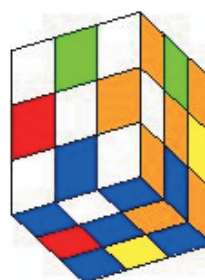
表-1 連続計算10,000回実行時の実行結果

平均計算時間 (sec)	0.135254
平均単位操作手数 (回)	159.313
adjust_middle_edgeface完成時エラー発生率 (%)	0

使用コンピュータ：Panasonic CF-S9 Let's note（プロセッサ：intel (R) Core TM i5 CPU M560@2.67Ghz, RAM：4.00GB, 64bit Operating system)



(a) F, U, L面



(b) B, D, R面

図-10 スーパーフリップ

に適用したところ、各単位操作の図を表示しない場合1.0764 (sec) と約 2 倍の計算時間を要した。しかし、各単位操作の図全てを表示する場合、下記に示すGNU Octaveの警告が多数発生し、途中でプログラムの実行が停止する。これは、画像表示数がOctaveの限界容量（計算例では101画像）を超えると発生する。

Warning : ft_manager : unable to load font : C : /WINDOWS/fonts/arial.ttf

Warning : ft_render : unable to load appropriate font

これを回避するには、単位操作毎の全ての画像表示から各一連操作の最終操作画像のみの表示に切り替え、画像表示数を少なく抑えればよい。

なお、ランダムパターンを自動生成して始める場合には、前項で示したRandStream関数についてOctaveは未対応である。このためmod_randpermファンクションmファイルの代わりにrandpermファンクションmファイルを使用することとなり、Octaveを起動する度にランダムパターンの生成は同じものとなる。

7. まとめ

MATLAB言語を用いてLBL法によるルービックキューブ解法プログラムを作成した。ルービックキューブ群のHTM (Half -Turn Metric) に関するCayleyグラフの直径（即ち、頂点間の最大エッジ距離）を決定する手法によれば、どのようなランダム状態から出発しても20手以内で6面が完全に揃ったホームポジション状態に至ることが判明している。しかし、本報告では解法パターンの場合分け数を極力少なく抑え（本プログラム例では持ち替えを除くと実質的には総数19）、分かり易いプログラムとすることを優先した。

ランダムな色配置の状態から解法操作手順を見出す計算を連続10,000回実行した結果によれば、画像表示を省略した平均計算所要時間は1回当たり0.135254 (sec)、平均単位操作手数は約160手であった。また、スーパーフリップの計算例では計算時間0.552632 (sec)、計算手数210手を要した。ただし、各単位操作の図を全て表示した場合には35.2546 (sec) であった。計算手数が多くことに難があるが、単純明快な本解法プログラムは短時間で解が得られ、計算容量も少なく済む利点を有する。

なお、フリーソフトであるGNU Octave-3.6.4で上記の作成プログラムを実行したところ、表示画像数に制約があるが問題無く動作することが分かった。

参考文献

- 1) <http://ja.wikipedia.org/wiki/ルービックキューブ>
- 2) <http://kociemba.org/cube.htm>
- 3) Tomas Rokicki, Herbert Kociemba, Morley Davidson, and John Dethridge, "THE DIAMETER OF THE RUBIK'S

- CUBE GROUP IS TWENTY”, SIAM J. Discrete Math., Vol.27, No.2, pp.1082–1105, 2013. 6.
- 4) 大友光司, “群論に基づくルービックキューブの数学的表現とLBL解法プログラムの作成”, 北海学園大学工学部平成26年度卒業論文.
 - 5) 島内剛一, “ルービック・キューブ免許皆伝”, 日本評論社, 1981.
 - 6) David Joyner, 川辺治之 (訳), “群論の味わい”, 共立出版, 2010.
 - 7) http://kjtmt.sakura.ne.jp/Rubik_Cube/index.html STEP1.
 - 8) (株)メガハウス社製ルービックキューブ付属LBL法 6 面完成攻略書.