

今そこにあるルービックキューブを解くツール

西本 匡志^{1,a)} 小林 周太郎^{1,b)} 川端 英之^{1,c)} 弘中 哲夫^{1,d)}

概要：

ルービックキューブは手軽に遊べる置換パズルである。しかしながら、予備知識無しで解くことはそれほど容易ではないし、解法の説明文書を頼りに色を揃えようにも、3次元形状のキューブの実体に対して多数の操作を正確に適用し続けるには少なからぬ根気が要求される。これに対し我々は、人間がキューブの色を揃えることを支援するツール「キューブ・ナビ」を開発している。キューブ・ナビはカメラ付きのコンピュータで動作するソフトウェアである。キューブ・ナビは、ユーザがキューブを操作する様子をカメラで監視し、キューブの状態を自動認識し、各時点からゴールに至るまでの操作手順をユーザに提示する。指示に従わないキューブ操作がなされても、あたかもカーナビのように、提示する指示内容を更新してゴールまでユーザを導き続ける。カメラ以外のセンサデバイスをを用いないため、使用するキューブには特殊な仕様を要求しない。本稿ではキューブ・ナビのスマホアプリ版の実装を紹介する。

キーワード：ルービックキューブ、3次元パズル、スマートフォン、ナビゲーション、画像処理

1. はじめに

ルービックキューブ^{*1}は手軽に遊べる置換パズルである [7]。しかしながら、解法についての予備知識の無いままで自力で解くことはそれほど容易ではない [3]。また、解法の説明文書を頼りに解くにしても、手元のキューブがどのような状態にあるのかの把握、その状態から全ての面を揃えるために取るべき手順の選択、さらに、多数の単位操作の系列から成る指示に従って手元のキューブを正確に回す作業が必要で、解くためには依然とし

て少なからぬ根気が必要である。

本稿では、我々が開発中の、人間がルービックキューブを解くことを支援するツールを紹介する。本ツールは、スマートフォンなど、カメラ付きのコンピュータで動作するソフトウェアで、ユーザが手にしているキューブの状態をカメラ画像から自動的に認識し、そこからゴールに至るまでの操作手順をユーザに提示する機能を持つ。本ツールはユーザがキューブに対して加える操作をカメラで監視し続け、状況の変化を検知する度にゴールまでの手順をその都度更新して示す。ユーザがツールの指示に従わない操作を行ったとしても、羅針盤が北極を示し続けるように、本ツールはその位置からゴールまでの道筋をユーザに示し続ける。状況判断は画像認識によってなされるため、使用するキューブには特殊な仕様を要求しない。本ツ

¹ 広島市立大学

a) nishimoto@ca.info.hiroshima-cu.ac.jp

b) kobayashi124@ca.info.hiroshima-cu.ac.jp

c) kawabata@hiroshima-cu.ac.jp

d) hironaka@hiroshima-cu.ac.jp

*1 「ルービックキューブ」は(株)メガハウスの登録商標である。

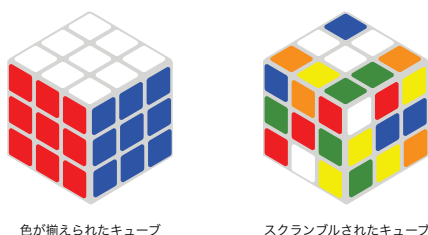


図 1 ルービックキューブ (3x3x3 Cube)

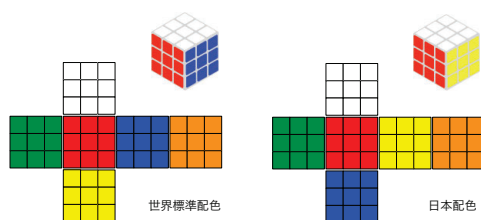


図 2 ルービックキューブの配色

ルを用いれば、ユーザは手元のキューブを気軽に操作して色を揃えることができる。

キューブ・ナビの特徴は、ユーザの手にあるキューブの状態を「一瞥して」現状態を把握できることである。キューブ・ナビは、キューブの一つ面を構成する 9 つの小正方形の色情報のみを得て、6 面全体の色配置をほぼ特定することができる。これにより、ユーザがキューブの各面を一つ一つカメラに向けて状況を確認させる必要性が大幅に削減されるので、ユーザはキューブから手を離すことなくスムーズに操作し続けることができる。

以下、第 2 節ではルービックキューブとその特徴、ルービックキューブを取り巻く現状について述べる。第 3 節では我々が開発している支援ツール「キューブ・ナビ」の設計について述べる。第 4 節ではキューブ・ナビの実装について述べる。第 5 節はまとめである。

2. ルービックキューブとその周辺

2.1 ルービックキューブの解法と遊び方

ルービックキューブは 1974 年に Ernő Rubik によって考案され、商品化されて世界中に広まった。1980 年には日本国内でも販売され^{*2}、大ブームを

^{*2} 第三著者の記憶では、1980 年当時に 1980 円で発売されていた。

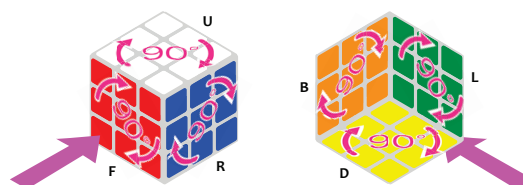


図 3 ルービックキューブの操作

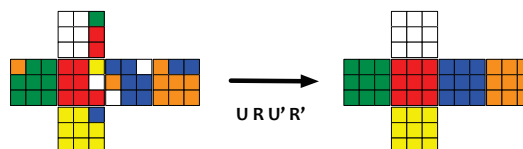


図 4 ルービックキューブを解く

巻き起こしたと言われる。

ルービックキューブは、1 面が 9 つの正方形に分割された立方体形状のパズルである。図 1 に描かれているように、一見すると 27 個の小さな立方体を 3x3x3 の形状に積み重ねたように見えるが、バラバラになるわけではなく、各面の切れ目に沿って回転させられるようになっている。ルービックキューブの表面を構成する 54 個の小さい正方形は図 2 に示すように 6 色で塗り分けられている。各面が異なる色になるように揃えられた状態から、回転操作を複数回加えて色をバラし^{*3}、再度元の状態に戻すのが基本的な遊び方である。

ルービックキューブの操作は、図 3 に示すように、ある面を基準にして各面を 90 度右向きに回転する操作を F, R, U, L, B, D の 6 文字で表現し、また左向き 90 度を F', R', U', L', B', D' で、180 度回転を F2, R2, U2, L2, B2, D2 で、それぞれ表現する方法が一般的である（本稿ではこれらの操作を単位操作と呼ぶ）。ルービックキューブを解く手順は、これら 18 種類の記号の系列で表現される^{*4}。例えば、操作記号列 U R U' R' は、図 4 の左側のスクランブルに対する（中央部が赤い面を手前に、白い面を上に見たときの）「解」の一つである。

ルービックキューブは置換パズルの一種であり、

^{*3} 色をバラす行為をスクランブル (scramble) という。

^{*4} これらに加え、中央部を回転させる操作や持ち替える（正面や上面を変更する）操作に記号を割り当てる方式もよく用いられる。

群論的なアプローチで解くことができる [3], [13]. しかしながら、問題の規模が大きいため（ルービクキューブの取り得る色の組み合わせは 4.3×10^{19} 通りにも及ぶ）、単純な問題である、とは言い難い。例えば、解くために要する最短手順（前述の 18 種類の記号一つを 1 手と数える）が 20 であることが示されたのは 2010 年になってからのことであった*5[6]。また、具体的にある色配置からゴールまでの最短手順を導出するアルゴリズムも調べられている [5] が、階層的・再帰的な探索アルゴリズムに基づく手順導出に要する時間は短くはない。ルービクキューブの遊び方の一つとして、制限時間内に見つけた手順の手数の長さ（Fewest Move Count）を競い合う競技がある。World Cube Association (WCA) によると、長さ 18 の手順を 1 時間以内に見つけられたという記録がある*6。

ルービクキューブの発売後間もなくして、ルービクキューブを解くのに要する時間の短さを競うスピードキュービングと呼ばれる遊び方が現れた。また、スピードキュービングに適した高速回転可能なキューブ*7 も開発され、販売されるようになった。WCA によるスピードキュービング競技会は今日も継続して頻繁に開催されている。スピードキュービングのために用いられる解法としては、必ずしも手数の短さは重要ではない。むしろ、キューブの色配置を確認しながら操作し易い手順や、キューブに対して高速に適用しやすい操作手順、また、そもそも人間にとって覚えやすい手順が求められる。結果として、スピードキュービングにおいては、最短手数の数倍の長大な手数をかけて「目にも留まらぬ速さで」キューブを操作する様子が多く見られる。WCA によると、3x3x3 Cube の色を揃えるのに要する時間の長さの世界記録は、2018 年 11 月 23 日現在、単発で 4.22 秒*8、

5 回の試技の平均で 5.80 秒とのことである。

2.2 普通の人にとってのルービクキューブ

ルービクキューブは安価で入手し易く、カラフルで、手に取りやすいサイズであり、購入意欲や知的好奇心を実に上手くすぐってくるパズルであると言えよう。実際、ルービクキューブを目にする機会は非常に多い*9。しかし、ルービクキューブは、多くの人にとって、インテリアとしてではなく、解くことが目的のパズルと見做されているだろうか。実際のところ、ルービクキューブは、解くことが難しい部類のパズルであろう。解く速さや手順の短さを競う競技会が盛んであること自体が、その傍証であるとも言える。

それでも、ルービクキューブの色を揃えることに対する欲求は世の中から尽きることは無いように、実際、ルービクキューブの色を揃え方を解説するオンラインの情報は無尽蔵に存在する。しかしながら、それらで紹介されている手法では、初心者向けと謳われるものであっても、複数の（細分化すれば 10 近い）ステップを順に辿ることが要求され、また個々のステップがキューブの回転操作を複数個連ねたものの反復によって構成され、そしてその反復の途中には条件分岐が含まれる。手元の問題に対してこのように汎用で複雑なアルゴリズムを当てはめて解くことは、少なからぬ根気が必要な作業である。

一方、眼の前にあるスクランブルされたルービクキューブの色を揃えるための操作手順を直接的に示してくれるような機能を持つツールがいくつも開発されている [1], [8], [11], [12]。これらのツールはいずれも、汎用アルゴリズムを人が苦勞して手元の問題に適用する必要を無くし、手元のスクランブルされたキューブに対する操作手順をしつらえてユーザに示してくれる。すなわち、ルービクキューブの色を揃え方を覚える気は無いけれど手元のキューブの色をただ揃えたいという願望を直接叶えてくれる。しかし、残念ながら、キュー

*5 しかも多量の計算資源を用いた全数調査によって得られた結果である。

*6 <https://www.worldcubeassociation.org/results/regions.php> 参照。

*7 “ルービクキューブ” は (株) メガハウスの、“Rubik’s Cube” は Rubik’s Brand Ltd の、それぞれ登録商標であり、他社製のものは “3x3x3 Cube” などと表現される。

*8 <https://youtu.be/NevGDFBfQGw>

*9 カフェ等で飾ってある例は頻繁に目にするし、土産物屋でもご当地写真等の図柄のキューブを見かける。ANA の機内にて小さな子供用のオモチャとして配っているのを見かけたこともある。

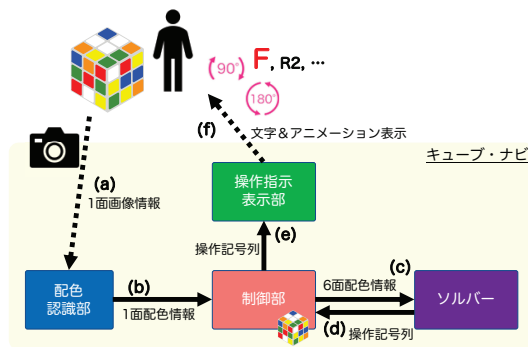


図 5 キューブ・ナビのシステム構成

ブの色を揃える手順は、一般には短くない。最短手順を示してくれるツールを用いるなら、高々20回のキューブの回転操作で目的が達成されるとはいえ、手元の三次元パズルに対して多数回の回転操作を間違いなく適用しなければ、色を揃えることはできない。すなわち、これらのツールは、ゴールに辿り着くための道順を教えてくれるものの、途中で道を外れた場合や迷子になったりした場合への対処の手段を示してはくれない。

3. キューブ・ナビ

キューブ・ナビは、ルービックキューブの色を揃えられるまでユーザに寄り添ってナビゲートするツールである。ユーザがキューブを操作する様子をカメラで捉え、色配置を認識し、ゴールまでの一連の操作手順を提示し続ける。ユーザが指示とは異なる操作を行った場合には、その状況からゴールまでの道筋を改めて提示する。すなわち、カーナビのように、目的地に辿り着くまで「現在地からの経路」を更新しながらユーザに示し続ける。

3.1 キューブ・ナビのシステム構成

キューブ・ナビのシステム構成を図5に示す。図5に示されている4つの構成要素（配色認識部、操作指示表示部、ソルバー、および制御部）はソフトウェア部品である。キューブ・ナビのシステム構成上の特徴として、（今日ではカメラが多くのコンピュータに搭載されていることを踏まえると）スマートフォンを含めてあらゆるコンピュータで動作させることができるという点が挙げられる。

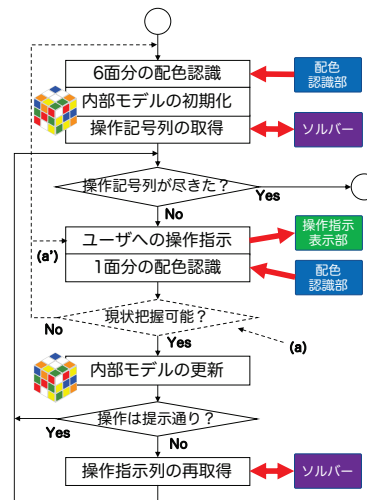


図 6 キューブ・ナビの動作の流れ

以下、図5に示されるキューブ・ナビの構成要素を列挙し、個々の機能の概略とその連携の様子について述べる。キューブ・ナビの全体的な動作については引き続き節で述べる。

- **配色認識部**。カメラから得られた1面分のキューブ画像（図5(a)）から、1面分の配色（9箇所の小領域それぞれの色）がどのようになっているかを判定し、制御部に伝える（図5(b)）。
- **操作指示表示部**。制御部から与えられる情報（図5(e)）に基づき、文字及びアニメーションを使ってユーザに対して操作を指示する（図5(f)）。
- **ソルバー**。キューブの6面全ての色情報（図5(c)）を受け、その状態からゴールまでの操作記号列を返す（図5(d)）。
- **制御部**。各構成要素を制御する。配色認識部からの情報を元にして操作が指示通りに行われたか否かを判断する。指示通りであれば続きの操作指示を行い、指示通りでなければ新たな操作記号列をソルバーから得る。

3.2 キューブ・ナビの動作

図6はキューブ・ナビの動作の流れを示している。キューブ・ナビは、カーナビと同様な振る舞いをする。すなわち、その動作は基本的に、(1) 状況の把握、および (2) 経路指示、これら2つの処

理の反復である。

キューブ・ナビでは、カメラを通して得られる情報を整理して、キューブの現状を内部のモデルに反映させて保持する。これにより、キューブ操作後にカメラ画像から得られる部分的な情報のみに基づいてキューブの全ての面の色配置を把握し続けられるようになっている。(すなわち、後述するが、図6中の(a)においてNo側に分岐する場合は稀である)。

キューブに対する一連の操作は、単位操作の系列である。キューブ・ナビは、キューブに対して一つの単位操作(すなわち、ある面の90度回転もしくは180度回転)が適用されたことを検知するたびに、キューブの現状把握を試みる。そして、ユーザが指示に従ったかどうかを判定する。ユーザが指示に従っていれば「続きの指示」を提示し、そうでなければ、(辿り着くことを想定していなかった)現状からゴールまでの操作記号列を、改めてソルバに問い合わせる。キューブ・ナビは、ユーザに提示する操作記号列が尽きた時点で停止する。このとき、キューブは色が揃った状態になっているはずである。

3.3 キューブ・ナビとカーナビの違い

キューブ・ナビはカーナビと類似の振る舞いをするものの、計算資源やデバイス等への要求の低さと引き換えに、カーナビよりも柔軟性の低い仕様となっている。

カーナビでは、GPS等の高精度のセンシングデバイスから得られるデータが利用可能であるため、十分な精度での現状把握が短時間でできる。従って、物理的な車の位置がナビゲーションシステム上で想定されていた位置から大きくずれることがあっても、短時間での現状把握によって速やかに復帰することができる。一方、キューブ・ナビの場合は、カメラを通して得られるデータに頼るため、キューブの一つの面を認識するだけでも短時間で正確に行うことは容易ではない。また、現状把握に失敗した場合には、6面全ての色配置を把握し直す必要があり、大きな手間がかかる。

カメラによるキューブの現状把握の難しさに対

処し、キューブに対するユーザの操作に追従し易くするため、キューブ・ナビでは、一つの単位操作を行うごとに「変化が確認できる面をカメラに見せる」という行為をユーザに強いる。従って、キューブ・ナビの単位操作への対応処理は、ユーザによって明示的にシステムに与えられるタイミングでなされることになる。ただし、変化が確認できる面をシステム側が短時間に認識できるようであれば、ユーザへの負担は大きくはないと考えられる。既存の関連ツール [1], [8], [11], [12] では操作の区切りごとにユーザがスマホやコンピュータを直接的に操作する必要があるが、キューブ・ナビの場合には、キューブを操作する手を一瞬止めて変化のあった面をカメラに見せるだけでよい。

キューブ・ナビのこの仕様は、カーナビでは見られない制約をユーザに課すことになる。すなわち、ユーザが一時的にキューブ・ナビに頼らずに自律的にキューブに操作を適用し続けた場合には、キューブ・ナビは以降のユーザへの指示を続けることができない。また、キューブ・ナビが示す手順通りにユーザが操作していたとしても、複数の単位操作をまとめて適用した後にキューブをカメラに見せた場合、キューブ・ナビは一面分の色配置情報だけから現状把握をすることができない。このような状況に陥った場合には、改めて6面全体の色配置をキューブ・ナビに伝えることが必要となる(図6の(a)においてNo側に分岐して最初に戻ることに相当)。

4. キューブ・ナビの実装

第3節で述べたキューブ・ナビの設計を踏まえると、キューブ・ナビの実現には次に示す機能の実装が必要である。それぞれについて、引き続き節で詳説する。

- スクラブルされたキューブに対して操作記号列を与えるソルバー
- 1面の色配置認識による現状把握機能
- カメラ画像から色配置を得る配色認識機能

4.1 ソルバー

キューブ・ナビで用いるソルバーは、キューブ・

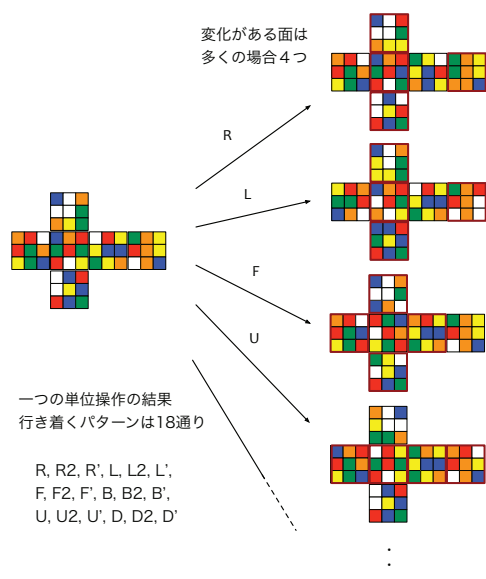


図 7 一つの単位操作後の色配置は 18 通り

ナビに特化したものである必要は無く、置き換えは容易である。例えば、短い操作記号列の反復で構成された単純な手順を示すソルバーにしたり、短い手順の手順が得られるソルバーに置き換えたりなど、用途に応じて変更することも簡単に行える。

キューブ・ナビのスマートフォン版プロトタイプでは、初心者向きと言われる LBL (Layer by Layer) Method (例えば [9]) に基づくソルバー [4] を使用した。このソルバーは、CFOP Method [10] の簡略化版である。CFOP はスピードキュービングにも頻繁に用いられる手法であるが、この手法で得られる手順の長さは短いとはいえない。

4.2 1 面の色配置認識による現状態把握

3.2 節で述べたように、キューブ・ナビは、6 面全ての色配置が把握されている状態から、一つの単位操作によって変化が見られた面の色配置情報のみに基づき、その単位操作後の 6 面全ての色配置情報を把握しようとする。後述の通り、この現状態把握手法は全ての場合に対応できるものではないが、経験的には、対応できない場面に遭遇することは頻繁ではない。

5 まず、一つの単位操作後のキューブの配色のパターンは、図 7 に示すように 18 通りある。1 単

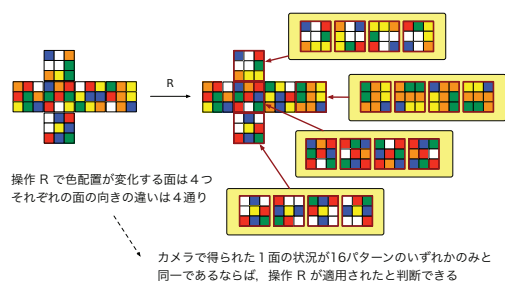


図 8 適用された操作の判定時には向きのみが異なる色配置情報は同一視する

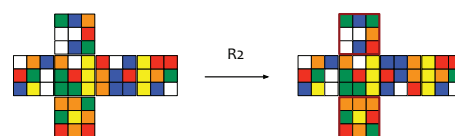


図 9 一つの単位操作で変化する面が二つしかない場合

位操作後のキューブの状態がこの 18 通りのうちのいずれかであることが判別できれば、どの単位操作が適用されたかが判別できる。すなわち、ユーザが指示に従ったかどうかを判定できる。ここで、図 7 に例示されているように、18 通りの各色配置において変化が見られる面は高々 4 つである（また変化が見られる面の位置は個々の操作について不変である）。このことに注目すると、1 単位操作後に変化があった面の色配置がこれら 4 × 18 通りの「いずれか唯一のパターンにのみみ合致すること」を確認できた場合に、ユーザが適用した単位操作を一意に特定できるといえる。

キューブ・ナビは、上の方法により、1 面のみの入力情報を元にしてどの単位操作が適用されたかを判定する。実装および使用に当たっては次の点への考慮が必要になっている：

- ユーザによってカメラ経由で与えられる 1 面分の色配置情報からは、向きがどのようになっているかまでは分からない^{*10}。したがって、図 7 の 18 通りのそれぞれについて、図 8 に示すとおり、向きを含めて 16 パターンとなる色配置情報と、カメラから得られる 1 面分のデータとを比較する作業が必要である（計算

^{*10} キューブを持つ向きをユーザに強制することは、ツールのユーザビリティの観点から望ましくない。

量的には大した問題ではない)。

- ユーザがカメラに向けたキューブの一つの面が、実際には変化していなかった面である可能性もある。ユーザ自身の単純な勘違いである場合のほか、図 9 に示すように、通常なら色配置が変化するはずの面であるにもかかわらずまたま色配置に変化が生じないパターンに遭遇しているという分かりにくい場合もある。キューブ・ナビは、カメラを通して得られる 1 面分の色配置情報に変化が見られるまでは何の処理も進めないで、ユーザは、キューブ・ナビが反応しないことをもって、見せている面が適当ではないということに気付く必要がある。
- 図 10 に示される例のように、変化した面を実際にカメラに向けていたとしても、どの単位操作が適用されたかが特定できない場合も存在する。このような場合には、キューブ・ナビはユーザに対して「特定できない」旨のメッセージを伝え、別の面をキューブ・ナビに向けるように促して 1 面分の認識をやり直す (図 6 の (a) の判定後に No 側に分岐して (a') へ)。これを 3 回繰り返しても依然として特定できない場合 (図 10 (b) はその場合に該当する例である) は、現状把握が不可能と判断し、キューブ・ナビは再度 6 面全ての色情報を確認する作業をユーザに求める (図 6 の (a) の判定後に No 側に分岐して最初に戻る)。

4.3 画像処理によるキューブの色配置把握

本節では、キューブ・ナビにおける配色認識部 (図 5)、すなわち、カメラから得られた画像を解析してキューブの一つの面を構成する 9 つの色の配置情報を得る機能の実装方式について述べる。画像処理による領域認識、領域選択、色判定には、OpenCV ライブラリを用いる。本機能は、次の一連の手続きにより実現されている。それぞれについて、順に述べる。

- (1) OpenCV を用いた領域認識 (ラベリング)。
カメラから得られた画像に対し、グレースケール化の後にエッジ検出を適用し、領域を抽出

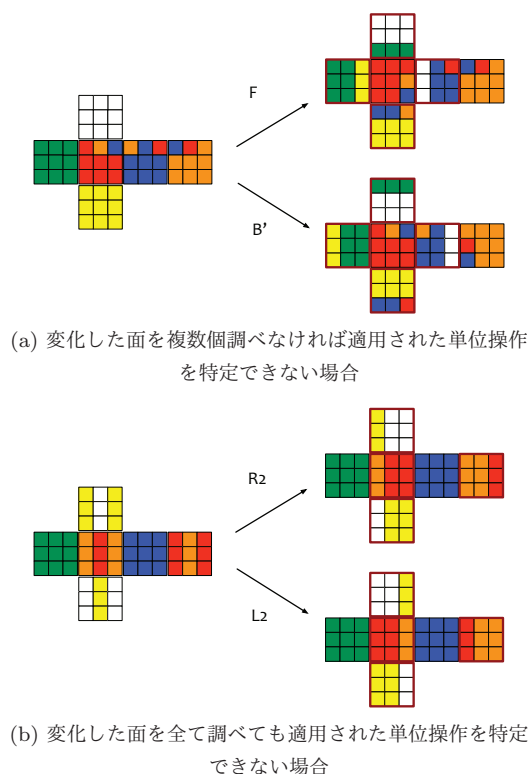


図 10 変化した面を一つ見ただけではどの単位操作が適用されたか特定できない例

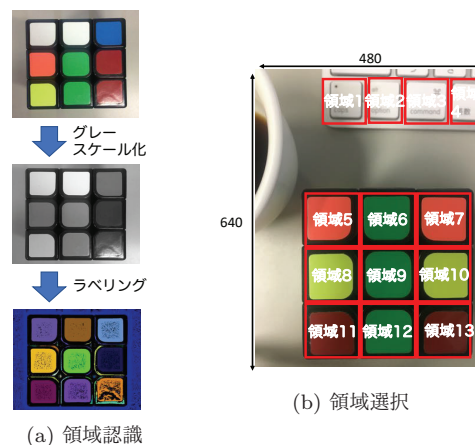


図 11 OpenCV による領域認識 (ラベリング) と領域選択

する (図 11 (a))。一連の処理は OpenCV ライブラリによってなされる。

- (2) キューブの一つの面を構成する領域の選定。
抽出された複数の矩形領域の中から、キュー

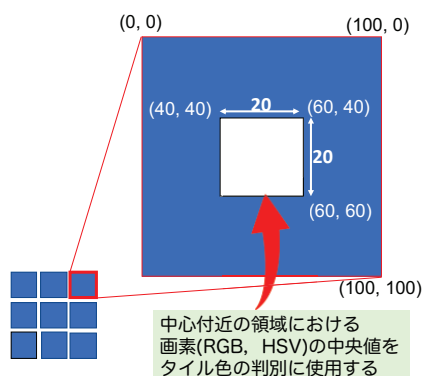


図 12 選択した領域の中央部を見て色を判定

ブの一つの面を構成する 9 つの小領域を選別する。この処理は、ほぼ正方形で均等なサイズの 9 つの領域を選ぶという比較的単純な手順により実現される (図 11 (b))。

(3) 個々の小領域の色の判定.

選択された 9 つの各領域の色の判定は次のように行う。まず、矩形領域を代表する RGB 値及び HSV 値を得るため、矩形領域の中心付近の複数の画素の RGB 値および HSV 値について、その中央値を求める。次に、RGB 値および HSV 値を用いて個々の色を判定する (図 12)。色の判定の際には、まずは RGB 値によるコサイン尺度で大別した後に、僅差の場合に上位二つを HSV の H (色相) によって区別する。

判定される色はキューブの各面を塗り分ける 6 色のうちのいずれかに限定される。そこで、予めキャリブレーションによって得ておいた 6 色の基準値との類似度を踏まえて各領域の色を決定する。これにより、キューブ・ナビは原色系の色以外で塗り分けられたキューブにも対応できるようになっている^{*11}。

4.4 スマートフォンアプリとしての実装

キューブ・ナビのスマートフォン版プロトタイプを、iOS の動作するスマートフォンやタブレッ

^{*11} 標準的に用いられる 6 つの色については、ここで述べた RGB 値および HSV 値に基づく判定法により概ね対応できるものと思われる。

トを対象として実装した。ソフトウェア開発には Xcode を使用した。使用したプログラミング言語は Swift および Objective-C である。

図 13 に、キューブ・ナビを使用している様子を示す。ユーザはスマートフォンに内蔵のカメラを用いてキューブを操作の様子を動画撮影しながら、画面に表示される指示に従い、キューブを揃える。両手を用いてキューブを操作するためには、補助者がキューブ操作者の手元を映し続けられると理想的であるが、適当な台に固定しておくなどすれば一人でも問題なく使用できる (図 13 は、机の上にスマートフォンを立て掛けて一人で操作をしている途中のスマートフォン画面のキャプチャ画像である)。

図 13 (a) は、最初にキューブの 6 面の配色を確認している様子である。6 面の確認が終われば、キューブ・ナビによって認識された配色パターンのミニチュアキューブが画面に表示されるので、ユーザは配色認識が正しく行えているか否かを確認することができる (図 13 (b) の右下参照)。また、即座に、色を揃えるための手順が単位操作ごとに表示される (同時にミニチュアキューブに添えた矢印によってグラフィカルにも示される)。指示された単位操作に引き続く操作列も表示し、手慣れたユーザが先読みし易いように配慮している。ユーザは単位操作を終えるごとに変化のあった面をカメラに向ける必要があるが、通常、1 秒以内に配色が認識される。そして、クリック音とともに画面が書き換えられ、指示が更新される。

5. まとめ

本稿では、キューブ・ナビのプロトタイプを紹介した。キューブ・ナビは、三次元パズルの一つであるルービックキューブの色を揃えることを支援するツールである。キューブ・ナビを使えば、手引書にすがりつつ根気強くキューブと格闘しなくても、導かれるままにキューブを操作することにより色を揃えることができる。

キューブ・ナビが目指すのは、ユーザがパズルを解いている傍らで適当なタイミングで丁度よい指示を出してサポートしてくれる「相棒」のよう



図 13 キューブ・ナビ (スマホアプリ版) の使用例

なツールである。本稿で紹介したプロトタイプ実装では、ツールのサポートを得るためにユーザがキューブを積極的に「見せる」必要がある。これに対し、キューブを覗き見つつ必要に応じてコメントを出してくれるツールを構築したい。これを実現するには、高速で正確な現状把握と適切な指示の提示機能が必要である。例えば、1面ずつの色配置認識ではなく（キューブを斜めから見るなどして）複数の面の配色を同時に認識できるようにしたり（これができれば図 10 (b) のような場合であっても現状把握が可能である）、複数の単位操作を自然に連ねてキューブを操作してもストレスなく追従できることが望まれる。これらの実現には、カメラを介したキューブ状態認識の精度と速度の向上が要となる^{*12}。画像認識の速度と精度を上げることができれば、色柄の描かれたキューブや、光を反射し易い表面を持つキューブなどの扱いも可能になると期待される。

その他の課題としては、ユーザスタディに基づく評価のほか、キューブ・ナビをルービックキューブの解き方指導のための教育的ツールとして活用するなどの発展の余地の検討が挙げられる。

謝辞 多数のご意見を下さった「夏のプログラミングシンポジウム 2018」ご参加の皆様に謝意を表します。

^{*12} キューブの状態認識を瞬時に行う方法として、キューブにセンサを内蔵してコンピュータと接続する方式が挙げられる [2]。しかしながらこの方法では、「今そこにあるルービックキューブを解く」という目的は達成されない。

参考文献

- [1] dikky.net: Cube Solver (online), https://cube-solver.dikky.net/rbk_solver/, (accessed 2018-06-30).
- [2] GoCube Project: GoCube (online), <https://www.producthunt.com/posts/gocube>, (accessed 2018-11-27).
- [3] Joyner, D.: *Adventures in group theory: Rubik's cube, Merlin's machine, and other mathematical toys*, 2nd ed., The Johns Hopkins University Press (2008) (川辺治之訳, 群論の味わい, 共立出版, (2010)).
- [4] Kawabata, H.: Cubing (online), <https://github.com/hide-kawabata/Cubing>, (accessed 2018-08-26).
- [5] Kotaro: ルービックキューブをアルゴリズムで解くということ (online), <https://www.terabo.net/blog/solving-cube-by-algorithms/>, (accessed 2018-11-23).
- [6] Rokicki, T., Kociemba, H., Davidson, M. and Dethridge, J.: God's Number is 20 (online), <http://www.cube20.org>, (accessed 2018-11-23).
- [7] Rubik's Brand Ltd: Rubik's Cube Homepage (online), <https://eu.rubiks.com>, (accessed 2018-06-30).
- [8] Solver, R. C.: Rubik's Cube Solver (online), <https://rubiks-cube-solver.com>, (accessed 2018-06-30).
- [9] Speedsolving.com Wiki: 3x3x3 LBL Method (online), https://www.speedsolving.com/wiki/index.php/Layer_by_layer, (accessed 2018-11-24).
- [10] Speedsolving.com Wiki: CFOP method (online), https://www.speedsolving.com/wiki/index.php/CFOP_method, (accessed 2018-11-24).
- [11] YokahiYokatoki: ルービックキューブ自動解法 (online), <http://www1.kiy.jp/~yoka/>

- gameland/RubiksCube_solve/RubiksCube_solve_swf.html, (accessed 2018-06-30).
- [12] メガハウス:体験! 6面完成ルービックキューブ (online), <https://itunes.apple.com/jp/app/ti-yan!-6mian-wan-cheng-rubikkukyubu/id918465248>, (accessed 2018-06-30).
- [13] 伊知地宏: Personal Communication (2018).

質疑・応答

- Q 色の判別について. 示された使用例では白のタイルに字が書いてあるものがあるが問題は生じないか. 例えば黒に誤認識される恐れはないか. (無地でないものは一般的か?)
- A 白地に黒で文字が書いてあっても, 試行した限りでは問題が生じていない. 白の割合が多いということが理由と思われる. (無地でないものは頻繁に見受けられる)
- Q カメラから見て変化がわからない面をユーザが回したら?
- A 基本的に, 仕様上, 変化があった面をカメラに向けることがユーザに要求される.
- Q ユーザがキューブの操作を間違えた時は後戻りさせたらどうか.
- A 意図して指示に従わない操作を行ったユーザの意思を尊重したい. なお, 一度誤った方向に進んだ場合, 別経路に変更するよりも後戻りするほうが良いとは言い切れない.
- Q 回した面を見せるのなら90度回転か180度回転かがわからないのでは?
- A 「回した面」ではなく「変化した面」を見せるので判別可能.
- Q なぜ一面だけを見せるのか? 斜めにして見せると画像が菱形になるが, 画像を加工して認識するのは容易では?
- A 一面だけを見て9つの色を短時間に正しく認識することさえ現状の実装では難しく, キューブを斜めにして複数面を同時に認識させられるには至っていない. 今後の課題である.
- Q ルービックキューブの色を揃えるのに長けている人がこのツールを使用したらどうなるのか?
- A ソルバーが提示する手順に劣る手順しか知ら

ない人ならばより短時間で解けるようになると考えられる (が, 通常はそんなことはないだろう).

- Q 利き手 (右利き, 左利き) に応じた手順提示等の対応は?
- A 対応は考えていない. なお, 通常の手順書では, 操作しやすいようにキューブの持ち替え操作を組み合わせた手順が提示されていることがあるが, このツールでは「持ち替え操作」の検出はできないため, 必然的に「持ち替え指示」もできない. 結果として, 利き手を全く意識せずに, キューブのあらゆる面をあらゆる方向に回すような手順の提示の仕方になっている.
- Q このツールを使うと, キューブの色を揃えられなかった人が揃えられるようになっていくのか? ルービックキューブの色を揃える手順を学習するためのツールとしての発展の方向性があるか?
- A 現状では, 入門者向けと言われる手順を提示している (そういうソルバーを用いている) ので, 反復していくうちに傾向を把握できるようになるかもしれないが, 期待はし難い. 手順の提示の仕方を工夫し, 教育的なコメントを添えるようにすれば, 教育的ツールとしての可能性はあり得る.
- Q 最初にキャリブレーションして色を登録し, 次いでキューブの6面の配色を登録する, という二段階の手順を踏んでいるが, 色を登録するときに用いる画像を初回の状態認識に使えないのか?
- A 原理的にはできる. 現状では色の登録時にキューブの向きや登録順を意識しなくて済むようにしているが, 指示の出し方を変更すれば実現できる. なお, 現状では, キャリブレーションと配色認識とでは画像の異なる部分にそれぞれ注目して処理をする. 前者では9個の正方形のうち中央のタイルのみに注目して多数の画像を処理し, 後者では少数の画像に対して9個の正方形全体を眺めて処理をしている. 前者の処理には後者よりも長時間を要

するが、用いる画像はキューブの中央部のタイルが明確に見えているだけで十分である。従って、

- キャリブレーションに用いた画像が各面の配色認識の用途の耐えられるものではない場合には、改めて初期状態認識をする必要がある。
- 使うキューブおよび光の当たり加減に変化が少ないようにすればキューブ・ナビを起動する度にキャリブレーションをする必要は無いため、現状の仕様のままでも大きな不便は無いと思われる。

Q ユーザが色を揃える様子を見て、良し悪しをコメントしたりガイドしたりしてはどうか。ゴールまでの距離を用いて手順を評価するなどしてはどうか。

A（参加者からのコメント）感覚的に評価することはできるだろうが、数値化は難しいのではないか。ゴールまでの距離は調べられる。

Q 局面の評価をできるようにしてはどうか？

A 難しいのではないか。

Q 出題をして、1手詰から初めて所要手数を徐々に増やしてユーザを鍛えれば、100手で色を揃える問題でも解けるようになるのだろうか？

A 判断できない。キューブ・ナビは出題機能を持っていない。