Title	ルービックキューブ攻略時における思考過程の分析
Author(s)	菱田，童之
Citation	
Issue Date	2009-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/8085
Rights	
Description	Supervisor:杉山　公造，知識科学研究科，修士

修 士 論 文

ルービックキューブ攻略時における思考過程の分析

北陸先端科学技術大学院大学
知識科学研究科知識システム基礎学専攻

菱田　童之

2009 年 3 月

修 士 論 文

ルービックキューブ攻略時における思考過程の分析

指導教官　杉山　公造　教授

審査委員主査　杉山　公造　教授
審査委員　中森　義輝　教授
審査委員　吉田　武稔　教授
審査委員　由井薗隆也　准教授

北陸先端科学技術大学院大学
知識科学研究科知識システム基礎学専攻

0750043 菱田　童之

提出年月: 2009 年 2 月

目 次

図 目 次

表 目 次

第1章　序論

1.1　背景

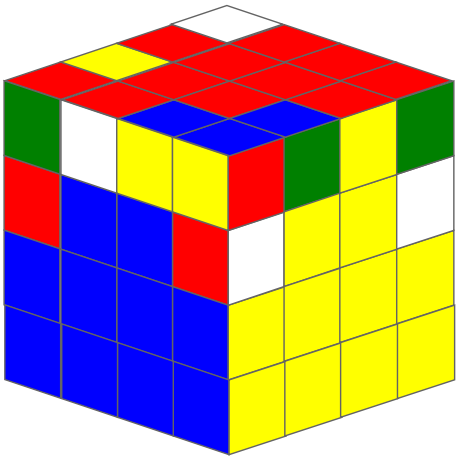
ルービックキューブというパズルがある。一般的には $3 \times 3 \times 3$ のルービックキューブが有名であるが、日本では $2 \times 2 \times 2$〜$5 \times 5 \times 5$ のルービックキューブが販売されている。単純にキューブが取りうる組み合わせの数だけで比較するならば、$2 \times 2 \times 2$ は $3,674,160$ 通り、$3 \times 3 \times 3$ では $43,252,003,274,489,856,000$ 通り、$4 \times 4 \times 4$ はなんと、$7,401,196,841,564,901,869,874,093,974,498,574,336,000,000,000$ 通りになる。しかし、一般に $3 \times 3 \times 3$ のルービックキューブが解ける人であれば、$4 \times 4 \times 4$ のルービックキューブは（確かに難易度は上がっているが）比較的簡単に解けると言われている。

自身の所属する研究室に、ルービックキューブを解く事が出来る人が数名いた。そこで、その人達に「どうやってルービックキューブを解いているか」という質問をした。すると、ある人は「解き方が決まっていてルービックキューブの状態を見ながら論理的に解く」と答え、別の人は「ルービックキューブは勢いで解くもので、普段解き方を意識していない」と答えた。次に、彼らが $4 \times 4 \times 4$ のルービックキューブを解く様子を観察した。すると、まず最初に1面とそこに隣接する1列目を揃え、2段目、3段目と順に揃えていき、最後に上部を揃える（図1.1(a)）人と、最初に各面の中央にある4つのキューブを揃え、次にその中央と角8つのキューブ以外の色の並びを揃えて、$3 \times 3 \times 3$ のルービックキューブと同じような状態（図1.1(b)）にしてから、$4 \times 4 \times 4$ のルービックキューブであるにもかかわらず、$3 \times 3 \times 3$ のルービックキューブを解く時と同じ方法を使って解く人の二種類に分かれた。

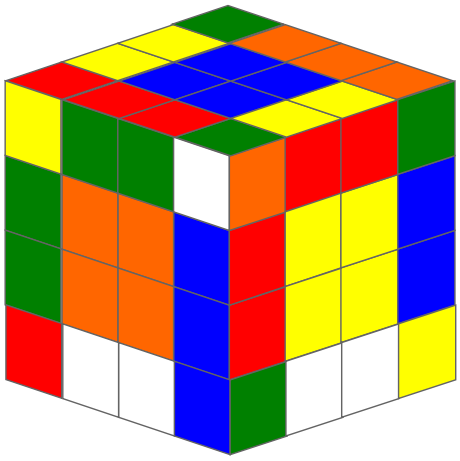
「勢いで解く」、「論理的に解く」という話を聞いた時、解き方の違いは「どうやってルービックキューブを解いているのか」という質問の答えに影響を与えているのだろうか。また、ルービックキューブを「論理的に解く人」と「勢いで解く人」にはどんな違いがあるのだろうか。「勢いで解く」という人もひょっとしたら自身の意識してないところで論理的に解いているのではないか。などのさまざまな疑問が生じた。そこでこれらの疑問を解決するために本研究を開始した。

1.2　本研究の目的

前節で説明したようにルービックキューブを解くときには「ロジカルに解く」・「勢いで解く」等の差が個人間で生じる。本研究では、ルービックキューブ攻略時の思考過程を実



(a) 積み上げ型の攻略（3 段目をそろえている最中）



(b) $3 \times 3 \times 3$ と同じ状態になったところ

図 1.1: $4 \times 4 \times 4$ のルービックキューブの解き方の違い

験によって明らかにし、「ロジカルに解く」といった人と「勢いで解く」といった人は実際にどのような違いがあるのかを調べる事を目的とする。

1.3 ルービックキューブに関する他研究と本研究の特徴

ルービックキューブに関する研究の大半は揃っていない状態のルービックキューブの揃えるための手順を見つけることに関する研究がほとんどである。例えば、データベースを用いて最適解を見つけようとする研究 [1] がある。また、同じ解を求める研究でも、ルービックキューブの上界 (じょうかい, Upper bounds) を複数の CPU を用いて虱潰しに近い方法で探す研究 [2] 等もある[1]。他の研究として、ルービックキューブが置換パズルの一種であり、置換群とかかわっていることを利用し、群論を使って解こうとするものもある [3]。更に、ルービックキューブのメディア変換を行い、別のパズルを創出するという研究もある [4]。

このように、ルービックキューブに関する研究は盛んに行われているが、操作と思考の関係に着目したものは著者の知る限りではほぼ無い。そこで、本研究では特に、ルービックキューブを解ける人を対象とし、ルービックキューブ攻略時の操作と思考の関係に主眼をおいて進めていく。

[1] 2009 年 1 月現在 23 手以内で完成させることが出来るとされている (http://cubezzz.homelinux.org/drupal/?q=node/view/117)。

1.4　本研究の進め方

本研究は、ルービックキューブを既に解ける人を対象として、ルービックキューブ攻略中の様子を撮影し、その時の操作ログと発話プロトコルデータを取得し、解析することで進めていく。

操作ログを取る際、どのような方法でどのようなログを取るかを決めなければいけない。そのために、ルービックキューブの平面グラフへの展開とルービックキューブの操作の記述方法の考案を行った。考案した記述法については3章で詳しく説明する。

実施する実験では、「論理的に解く」と答えた人に$2 \times 2 \times 2$のルービックキューブと$3 \times 3 \times 3$のルービックキューブの二種類のルービックキューブ攻略の実験を行う。そして、「勢いで解く」と答えた人は$3 \times 3 \times 3$のルービックキューブ攻略の実験を行う。「論理的に解く」と答えた人は自分で何のためにどのような操作しているかを意識していることが推測できるため、まずは「論理的に解く」人の$2 \times 2 \times 2$及び$3 \times 3 \times 3$の攻略過程を分析する。その結果を「勢いで解く」と答えた人と比較し、「勢いで解く」と答えた人は実際にはどのようにルービックキューブを解いているかを明らかにしていく。

1.5　本論文の構成

本論文は1章で本研究の背景、目的や研究の進め方等について説明し、続く2章以降に具体的な内容を記述していく。2章では、ルービックキューブの二次元展開法とその利点について説明する。3章では、本研究で使うルービックキューブの操作記述方法について説明し、2章で説明した二次元展開法とその操作記述法との関係を記す。4章には操作記述を行うためのサポートツールの必要性等を論じ、実際に開発したサポートツールの概要を説明する。続く5章では実験方法を説明とその結果の考察を行う。最後に、6章で本研究のまとめと今後の課題について言及する。

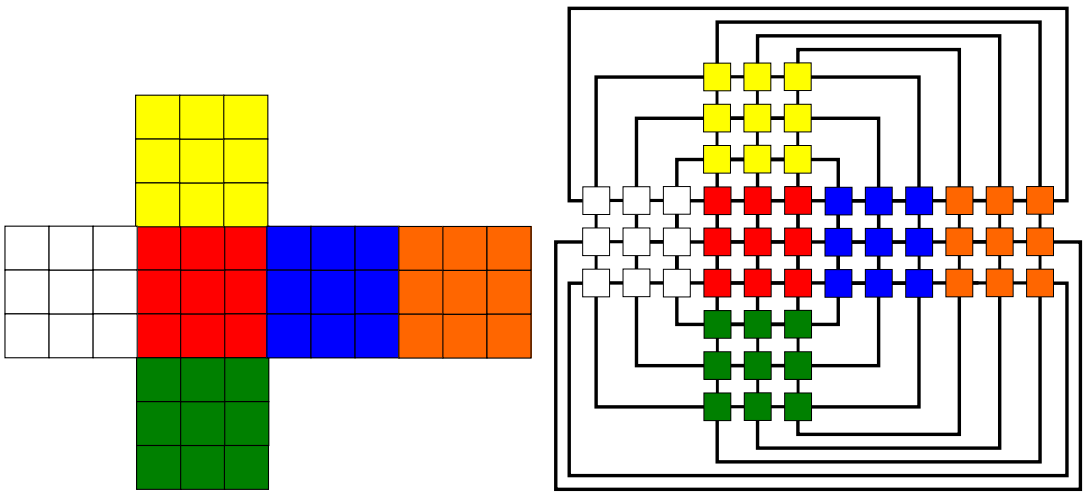
第2章　ルービックキューブの二次元展開

2.1　ルービックキューブの構造の特性

ここでは、ルービックキューブの構造特性について説明する。

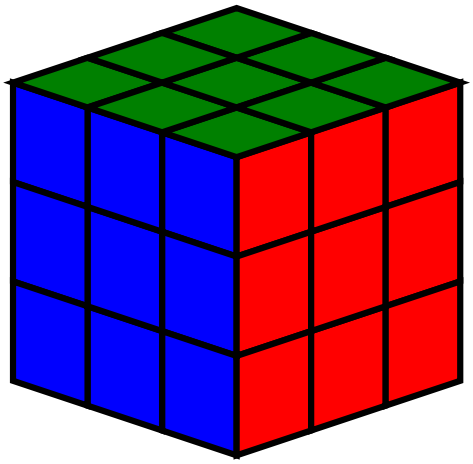
ルービックキューブは構造上ある操作を行っているときに、その操作に対して垂直な操作を行うことは出来ない（2列同時に回すなど平行な操作は可能）。そのため、操作は基本的に一つずつ行われ、平行な操作を除き、整列的な操作は出来ない。一般に、人間はシリアルな思考とパラレルな思考を使い分けている。ルービックキューブの操作を行う際に直列的な操作を要求されるため、たとえ思考が並列的であっても一部例外を除き表出する際に必ず直列的になるのも特徴の一つだといえる。

また、杉山らの研究 [5] により、ルービックキューブは置換パズルの一種であり、平面グラフへの変換が可能なことが分かっている。図 2.1(a) は $3 \times 3 \times 3$ のルービックキューブの展開図である。そして、図 2.1(b) はその展開図を更にルービックキューブのタイル単位に分け、隣接を示すために線で繋いだものである。これは一つの例であるが、このようにルービックキューブは平面グラフに展開することが出来る。

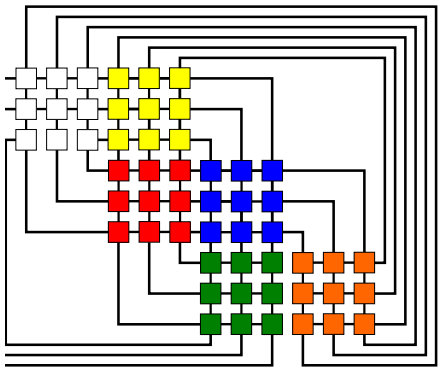


(a) ルービックキューブの展開図　　(b) 更に分割し隣接を線で表す

図 2.1: ルービックキューブの単純な平面グラフ展開



(a) 展開前は一度に 3 面までしか見ることが出来ない



(b) 展開後はすべての面を同時に見ることが出来る

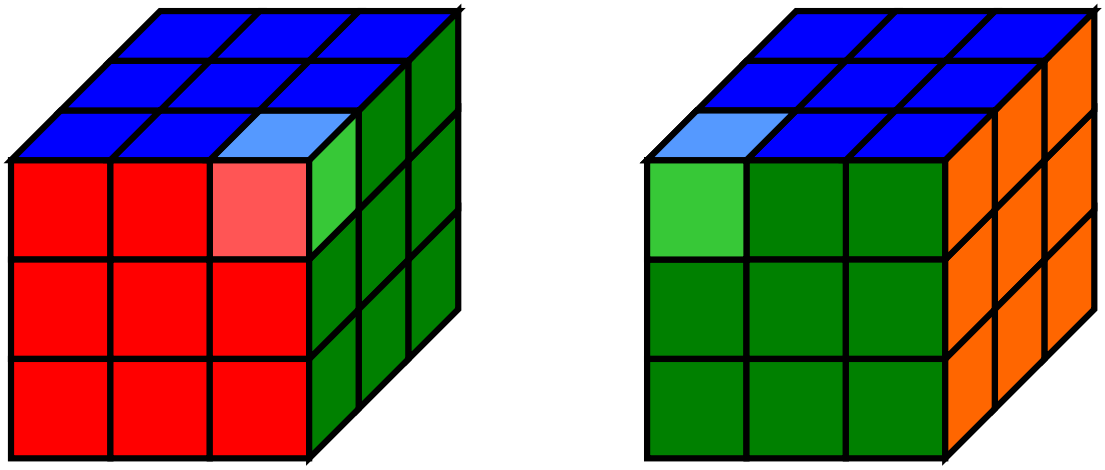
図 2.2: 展開による利点

2.2 平面グラフ展開の利点

前節でルービックキューブを平面グラフへ展開可能だという研究を紹介した。本節では平面グラフに展開することでどのような利点が得られるかを考察していく。

本来、ルービックキューブは立体であるため、現在見えている面の陰になって見えない面が必ず存在する。しかし、平面グラフに展開することですべての面を一度に見ることが出来る。これは平面グラフ展開の利点の一つだといえる。図 2.2 はそれを視覚的に表したものである。図 2.2(a) のように、ルービックキューブは一つの視点からでは多くても 3 面しか見ることが出来ない。しかし、図 2.2(b) のように平面グラフへ展開することで、ルービックキューブの 6 面すべてを同時に見ることが出来る。尚、図 2.2(b) は図 2.1(b) とは違うが、隣接関係さえ崩さなければこのような配置方法の変更が可能である。

また、ルービックキューブはそれ自体の回転により操作を行っていないのにキューブの位置が変化する。図 2.3(a) のルービックキューブの裏側を見るためにはルービックキューブ事態を回転させて図 2.3(b) のようにしなければならない。この時、図 2.3(a) では右上にあった赤・緑・青の組み合わせのキューブが、図 2.3(b) ではルービックキューブの操作をしていないにもかかわらず別の場所に移動してしまっている。しかし、平面グラフに展開すると、操作が行われない限りタイルの位置が変化しない。これもルービックキューブを平面グラフに展開することで得られる利点の一つといえる。



(a) (b)

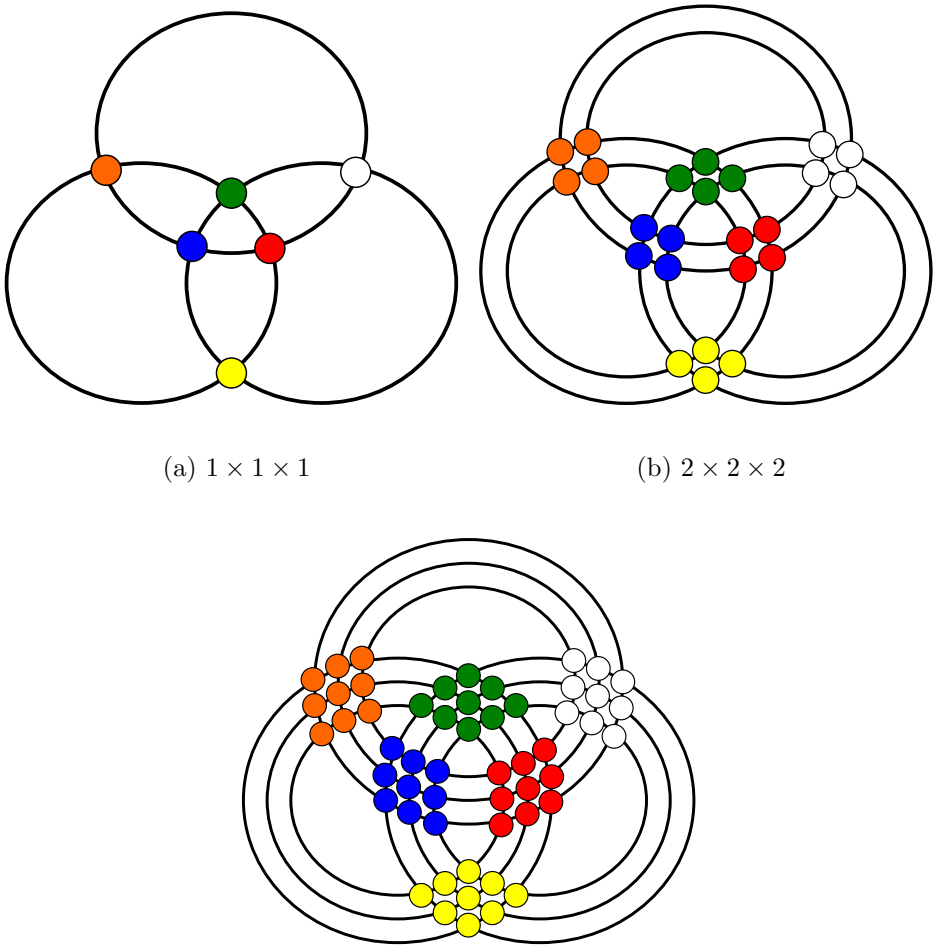
図 2.3: ルービックキューブ自体の回転によるキューブの移動

2.3 平面グラフ展開の方法

ルービックキューブが平面グラフへと展開できることは既に説明した。しかし、単に隣接関係を保ったまま平面グラフへ展開するだけでは、ルービックキューブを操作したときに、同時に動作するタイルの関係が分かりにくい。更に、初期状態では色の隣接関係から操作を推測することが出来るが、何度か操作を行うと、タイルの色がどのように移動するかまったく予測が出来なくなる場合も考えられる。そこで、本研究では、ルービックキューブの操作と、それに対応するタイルが視覚的に判断できるルービックキューブの平面グラフ展開の新しい方法として次に説明する方法を考案する。

展開前と展開後の関係が分かり安いように $1 \times 1 \times 1$ のルービックキューブで考えてみる。まず、互いに交差するように円を 3 つ書く。そして、その円上にある 6 つの交点にタイルを配置する (図 2.4(a))。この時、タイルの配置（配色）は、実際のルービックキューブで隣接する色が展開後に隣接（エッジで結ばれている）するように行う。

$2 \times 2 \times 2$ のルービックキューブを展開する場合も、$1 \times 1 \times 1$ の場合と同様に、円の交点にタイルの隣接関係を崩すことなく配置していく。ただし、$1 \times 1 \times 1$ の場合と円の数が異なり、$2 \times 2 \times 2$ では、同心円を 2 つずつ 3 組書き、同じ中心を持たない円同士が交差するように円を配置し、その交点にタイルを配置していく (図 2.4(b))。同様に $3 \times 3 \times 3$ のルービックキューブを展開する場合は、同心円を 3 つずつ 3 組書き、交点にタイルを配置することで展開することが出来る (図 2.4(c))。



(a) $1 \times 1 \times 1$

(b) $2 \times 2 \times 2$

(c) $3 \times 3 \times 3$

図 2.4: 同心円展開によるルービックキューブの平面グラフ展開

以上より、ルービックキューブのサイズを任意の N（N は正の整数）としたときの $N \times N \times N$ のルービックキューブを二次元展開する方法を以下に示す。

1. N 個の同心円を 3 組作る。
2. 同じ中心を持たない円同士が互いに交差するように円を配置する。
3. 円の交点に隣接関係を崩さないようにタイルを配置する。

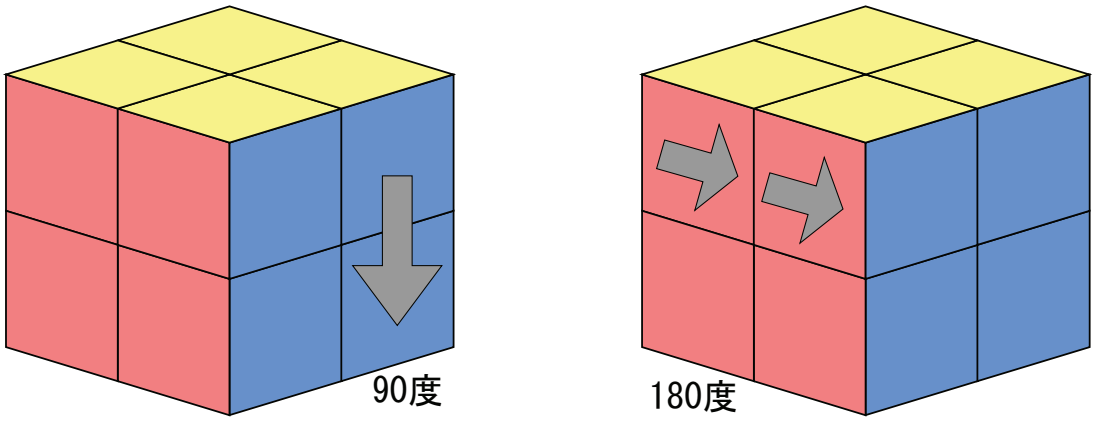
また、この方法を同心円展開法と名付け、同心円展開法によって展開することを同心円展開すると呼ぶことにする。同心円展開されたルービックキューブ上で操作を行うと、同じ円上にあるタイルが N 個隣に移動する。

第3章　ルービックキューブの操作

3.1　操作の記録

ルービックキューブ攻略時における思考過程を分析するにあたり、操作と思考の関係を調べるためにどのような操作をしたかを記録する必要がある。

まず、操作の記述方法としてどのようなものがあるかを調べた。ルービックキューブを購入した際に付属する攻略書では、図3.1のような方法で操作を記述していた。図3.1(a) これは、矢印が書いてある列を矢印の示す方向に90度回転させる事を表している。図3.1(b) のように矢印が2つある場合、その矢印がある列を矢印の示す方向に180度回転させる事を表す。この方法を用いることで操作を特定することが出来る。だが、この方法は視覚的に操作を判断することが出来る反面、記述自体に時間がかかる。本研究では、ルービックキューブ攻略中の操作をすべて記録するため、この方法で記述することは現実的ではない。



(a)　　(b)

図 3.1: $2 \times 2 \times 2$ のルービックキューブ付属の攻略書での操作の説明方法

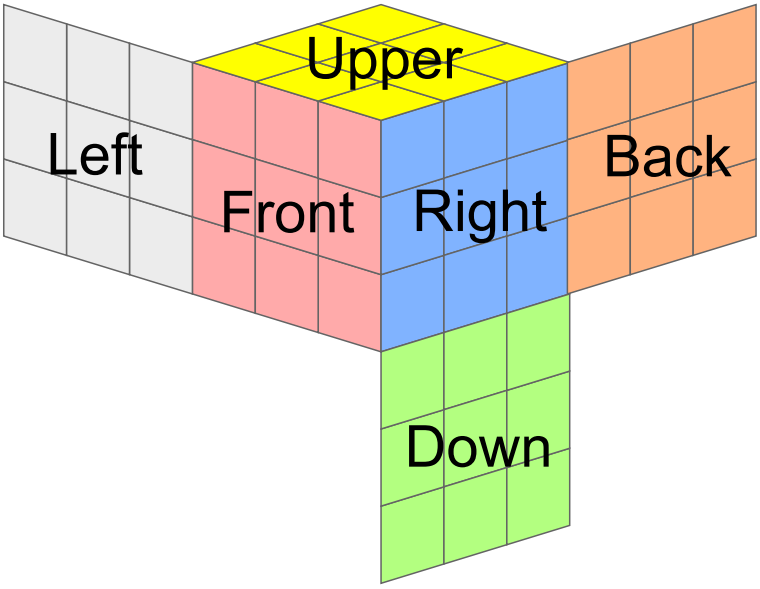


図 3.2: 一部展開したルービックキューブと面の名前

次に、他にどのような方法があるかを調べた。ルービックキューブの攻略法を記載しているウェブサイトの一つ[1] では次のような方法を用いていた。まず、ルービックキューブの各面に名前をつける（図 3.2）。そして、操作は面の頭文字で回転させる面を特定し、その後ろに右回転の時は何もつけず、左回転の時は $'$ をつけて区別する。更に、180 度回転させる時は二度回転させることを示す 2 をつけて表す。Rw のように w を付けると 2 層同時回転することを意味する。図 3.3 に、この方法で表すことが出来る操作と実際のルービックキューブではどのように操作されるかの対応をいくつか示す。この方法は、$2 \times 2 \times 2$ と $3 \times 3 \times 3$ のルービックキューブでは操作を表すことが出来るが、$N \times N \times N (N \geq 4)$ のルービックキューブで $n (2 < n < N)$ 列目のみを回す操作を表すことが出来ない。図 3.4 はその例である。また、この方法では正面を Front とした場合の相対的な操作しか記述することしか出来ない。

これらの方法は、攻略法を説明するためには適している。しかし、本研究ではすべての操作を体系的に扱い、しかも簡単に記述できるような操作記述を用いるため必要があるこれらの方法は適さない。そこで、本研究独自の操作記述法を考案した。次節より考案した記述法について詳細を説明する。

[1] http://rubik.xxxxxxxx.jp/

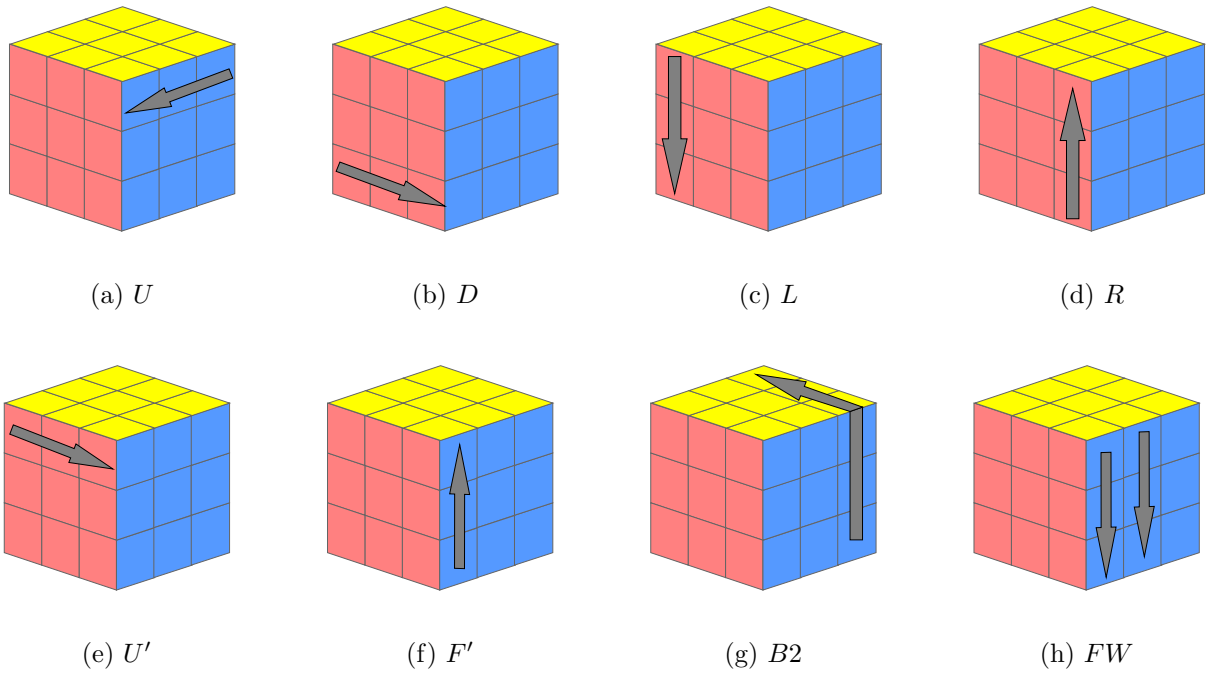


図 3.3: 面に名前を付けたときの操作記述（一部）

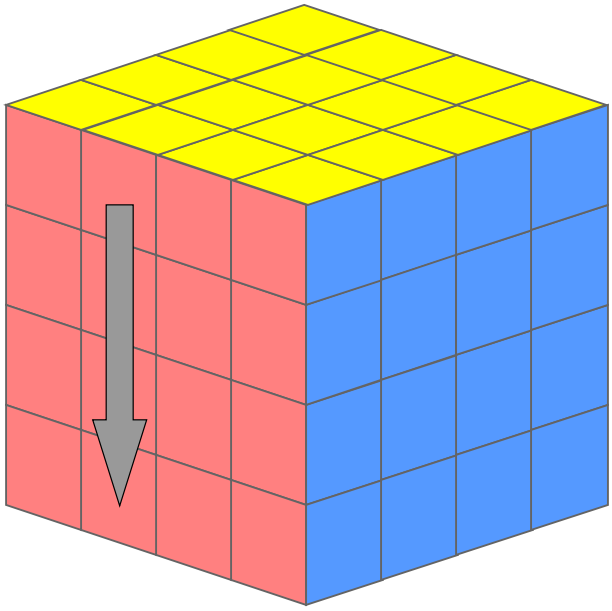


図 3.4: 操作の記述が出来ない例

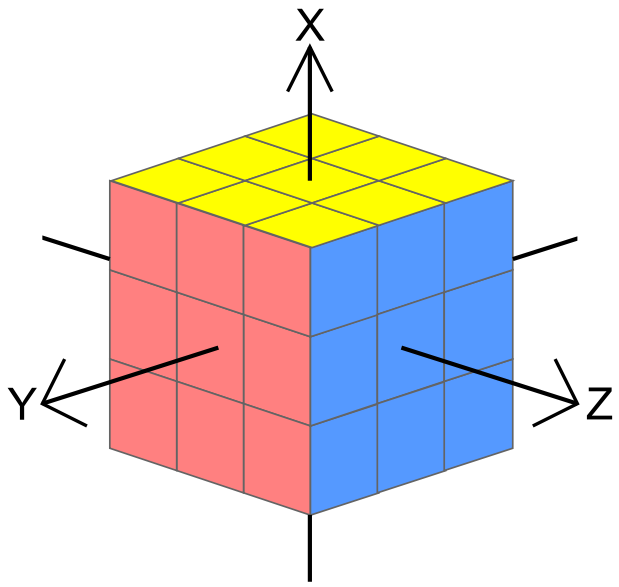


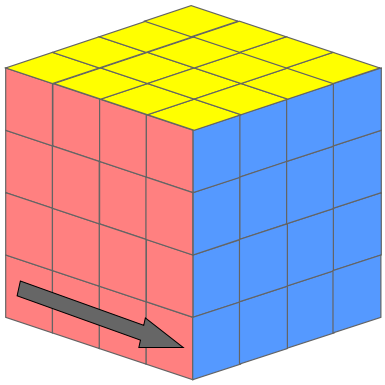
図 3.5: ルービックキューブの中心を通る軸

3.2 操作の記述方法

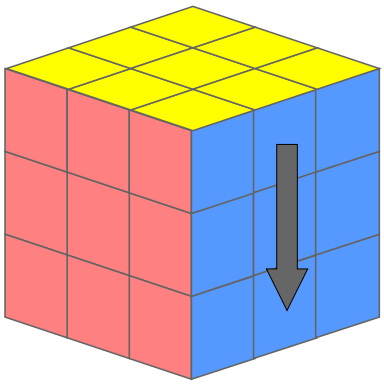
本研究では、操作の記述法に次のような要件が必要である。第一に、簡単に記述できる必要がある。これは、ルービックキューブ攻略時の操作をすべて記述して記録するため、操作の記述に時間がかるのは非効率であるためである。そのため、、ルービックキューブ付属の攻略書に書かれているような図を用いて記述するのではなく、文字だけで記述することにする。また、図 3.3 の記述方法ではルービックキューブ自体を回転させると操作の記述方法が変わってしまう。それは不都合だと考えたので、ルービックキューブ自体を回転させても操作の記述が変わらない絶対的な操作記述方法を決める。

まず、図 3.5 のように、ルービックキューブの中心を通る軸に名前をつけた。ルービックキューブの操作は、必ずどれかの軸に垂直な回転になるため、軸を絶対的なものとして扱い、どの軸に垂直な回転であるかにより名前をつけることにした。

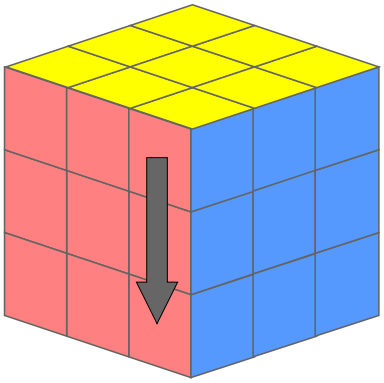
実際の記述は次のように行う。まず、どの軸に対して垂直な回転であるかを軸の名前を取ってそれぞれ X, Y, Z で表し、この時の軸を回転軸と呼ぶことにした。次に、回転軸のプラス方向から見たとき（図 3.5 では矢印のある方向）、回転させる列は何列目であるかを数字で示す。ただし、一番手前にある列を 0 列目とする。最後に、回転軸をプラス方向から見て、回転させる方向が右回りなら R、左回りなら L で表すことにした。そして、回転軸の名前・何列目を回転するか・回転する方向を並べて記述することで一つの操作を記述する。この方法を用いることですべての操作を統一した方法で記述することが出来るようになった。ただし、この方法では複数列同時回転を表すことが出来ない。そのため、複数列同時回転を記述する場合は二つ以上の記述に分けて記述することにした。図 3.6 はこの方法を用いて記述した操作の例である。図 3.6(a) は X 軸に対して垂直な操作であり、



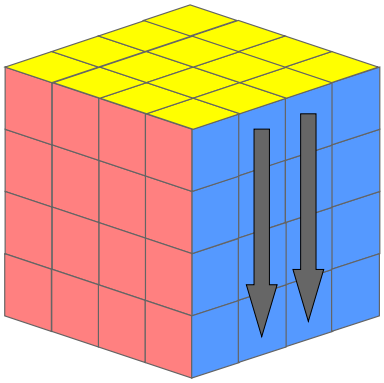
(a) $X3L$



(b) $Y1R$



(c) $Z0L$



(d) $Y1RY2R$

図 3.6: 考案した操作記述の例

X 軸のプラス方向から見たときに、3 列目が左回転する事を表している。図 3.6(b) は Y 軸に対して垂直な操作で、Y 軸のプラス方向から見たときに 1 列目が右回転している事を表す。図 3.6(c) は、Z 軸に対して垂直な操作で、Z 軸のプラス方向から見たとき、0 列目が左回転している事を表す。図 3.6(d) は複数列を同時に操作している。どちらも Y 軸に対して垂直な操作で、Y 軸のプラス方向から見たとき、1 列目と 2 列目が右に回転している。

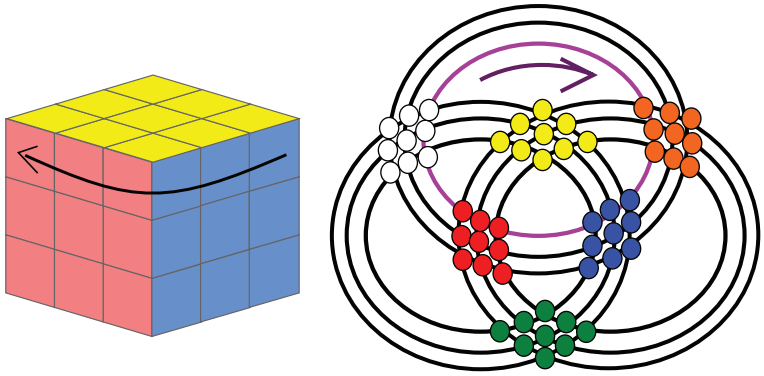


図 3.7: 同一の操作 (X0R) をルービックキューブと同心円展開でそれぞれ表す

3.3 考案した操作記述法と平面グラフ展開との関係

本節では、同心円展開されたルービックキューブと考案した操作記述法の関係について説明する。$3 \times 3 \times 3$ のルービックキューブを同心円展開すると三種類の同心円ができる。この同心円の中心が操作記述で定義した軸にそれぞれ対応している。そして、それぞれの円がその円の中心に対応する軸を回転軸とする操作に対応している。また、この時、最も内側にある同心円を 0 個目とするとき、n 個目の円が展開前のルービックキューブの n 列目に対応する。

図 3.7 はルービックキューブの操作と同心円展開されたルービックキューブの操作の対応の例である。図 3.7 を見て分かるとおり、考案した操作記述法では同心円展開されたルービックキューブとの親和性が高く、操作記述から実際の操作への変換が簡単に行える。

第4章　操作記述サポートツールの開発

4.1　操作記述サポートツールの必要性

ルービックキューブを解く時、ルービックキューブを様々な方向から見て次の操作を決定する。既に説明したように、ルービックキューブは立体であるため現在見えている面の裏側を見るためにはルービックキューブ自体を回転させる必要がある。そのため、操作中の様子を観察しても3章で定めた方法で操作を記述しようとしても、操作の基準となる軸が常に移動し続けるためルービックキューブに対して行った操作を特定し、記録することは非常に困難である。

そこで、ルービックキューブをどのように操作したかを記録するためのサポートツール開発の必要性を感じた。しかし、サポートツールにどのような機能があれば操作の記述を行うことができるようになるかは分からない。そこで、サポートツールに必要な機能を洗い出すために、サポートツールを用いない状態で、何が分かり、何が分からないかを調べる。

4.2　サポートツール無しで分かること

サポートツールに必要な機能を洗い出す為、ルービックキューブを操作する様子を何も使わずに観察した。だが、ルービックキューブを次々に操作するため、どんな操作をしたかの確認をする前に次の操作に移り、「何かの操作をした」という情報しか得ることができなかった。

そこで、ルービックキューブを操作する様子を撮影し、撮影した動画を再生・一時停止することで操作一つ一つをゆっくりと見ることにした。この時、撮影はルービックキューブを操作している人の目線とできるだけ近くなるようにカメラを配置した。

次に、撮影した動画を見ることでどのような情報を得ることができるかを調べた。まず、動画からどのような操作をしているかを抜き出せるかを確認する。動画を見ながら必要に応じて一時停止することで、撮影をすることで操作を一つずつ取り出すことができた。しかし、前節で述べたように撮影した動画ではルービックキューブ自体が回転している。そのため、3章で定めた方法に従い操作の記述を行おうとした時、基準となる部位がどこにあるかが把握できない。3章で説明した記述法は、操作に命名するための基準が重要であるため、サポートツールではこれを解消する必要がある。

(a) 操作前　　(b) 操作後

図 4.1: ルービックキューブの操作前後の画像の比較

図 4.2: 撮影した動画の一場面

そこで、操作前と操作後のルービックキューブの状態に着目し、そこから操作を推測できないかと考えた。図 4.1 は操作前の画像と操作後の画像を比較したものである。図 4.1(a) の操作前では左側最上列にある緑、赤、赤、黄の組み合わせが図 4.1(b) の操作後に右側最上列に移動している。このことから、実際の操作は「最上列を 90 度回転させた」ということが分かる。しかし、このように操作が分かっても基準が分からないため、3 章で定めた記述法で記述することはできない。しかも、撮影用のカメラは一台しか用意していないため、図 4.2 のようにルービックキューブの一面だけしか映っていないこともある。だが、この場合においても前後の動画を見ることで「どこを回転させたか」という情報が得られることを確認している。つまり、動画より得られたルービックキューブの状態変化の推移を確認していくことでルービックキューブのどこを回転させたかという情報が得られることが分かった。

4.3 サポートツールに必要な機能と概要

以上のことを踏まえて、どのような機能を持つサポートツールを開発すれば操作の記録ができるかを考察する。

4.2 節で説明したようにルービックキューブの状態を追う事で、どこを回転させたかという情報を得ることは可能である。しかし、操作の基準となる軸の移動を追いかけるのは難しいため、その操作を 3 章で定めた記述と対応させることは出来ない。そのため、サポートツールでは操作の基準を内部的に管理し、ユーザは操作の記述を意識することなく操作する。サポートツールの内部でその操作を 3 章で定めた操作記述で記録していく。

次にサポートツールのインターフェイスについて考える。操作の記録はルービックキューブの状態推移を追いかけることで実現される。そのため、サポートツールではルービックキューブの状態が一目で確認できるほうが都合がよい。更に、操作の記述法との親和性を考慮し、2 章で説明した展開法を用いて展開した平面グラフをインターフェイスとして利用することにした。こうすることで、ルービックキューブ全体の状態を一目で確認できるようになり、操作の適用ができると考えられる。

また、操作を記録するためのファイル出力機能とそのファイルを読み込み操作を再現するためのファイル入力機能も必要である。

4.4 サポートツールの開発

4.4.1 サポートツールのシステム構成

本節では開発した記述サポートツールについて説明する。サポートツールの開発には Java を利用した。図 4.3 は、システム全体の構成図である。以下に各部の説明を記すが、実際に開発したシステムではデバッグやコーディングのしやすさを向上させるため必要に応じて機構の統合・分割や追加を行っている。

ファイル入出力部

ファイル入出力部では、操作履歴のファイル出力や操作再現のためのファイル読み込みを行う。入出力に利用するファイルはテキスト形式とした。表 4.1 は読み込み可能なファイル規則であり、操作履歴を出力したファイルもこれに従う。一行一命令で記述するため、行頭から読み込み、命令が簡潔した時点で該当行のそれ以降はコメントとして無視することにした。図 4.4 に実際に出力されたファイルの例を示しておく。

パズルモデル管理部

パズルモデル管理部ではルービックキューブの数理モデルを管理する。新しくルービックキューブが生成されたときに初期化され、ユーザの操作がある度に更新される。

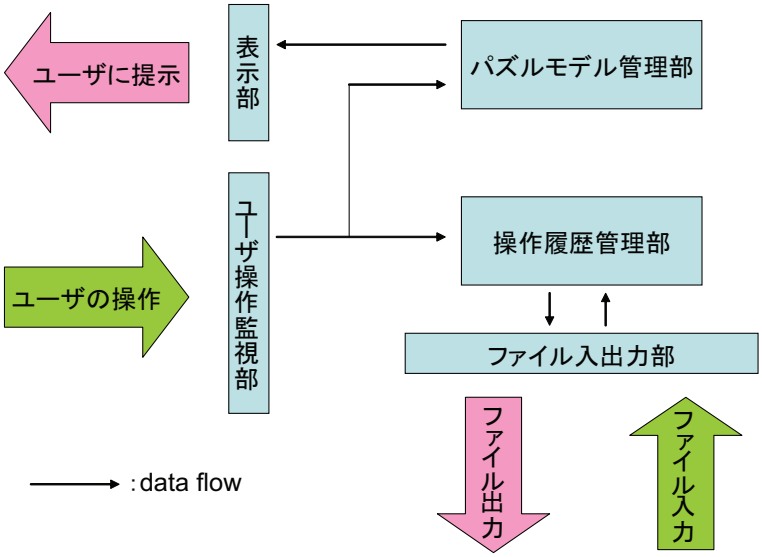


図 4.3: システム全体の構成図

表 4.1: 入出力ファイルの規則

ファイル名	任意
命令文の区切り	改行
[#] 以降	コメントとして無視
rotete [操作]	[操作] で指定された操作を行う
wait [ms]	次の命令の実行を [ms] ミリ秒待つ
new circle [n]	今までの操作を破棄し、新しいルービックキューブを生成する

操作履歴管理部

操作履歴管理部ではユーザの操作履歴を管理する。ここで言うユーザの操作とは、マウスのクリックやドラッグなどではなくルービックキューブに対する操作のことである。ファイル出力時にはここで記録されている操作履歴を表 4.1 に従い出力する。ただし、操作と操作の間隔（時間）は管理していない。

表示部

表示部では二次元展開したルービックキューブの表示やユーザの操作に対応するノードの移動を行う。表示に利用する座標情報の保持や移動距離等の計算時に起こる計算誤差の調整なども表示部で行っている。

図 4.4: 出力ファイルの一例

ユーザ操作監視部

ユーザ操作監視部ではマウスのクリックやドラッグなどのユーザの具体的な操作を監視する。ユーザの操作を常に監視し、意味のある操作（ルービックキューブの操作等）を行ったときにパズルモデル監視部・操作履歴管理部・表示部に必要な指示を与える。そのため、プログラムの状態遷移はここで行われている。

4.4.2 作成したサポートツールの動作

ここでは実際に作成したサポートツールがどのようなものか説明する。図 4.5 は作成したツールの起動直後の状態である。実際に二次元展開されたルービックキューブを表示するには、上部にあるメニューバーの [Edit] → [New Game] にある 2〜5 の数字を選択する。すると、それぞれ $2 \times 2 \times 2$〜$5 \times 5 \times 5$ のルービックキューブが表示される (図 4.6)。尚、表示できるルービックキューブを $2 \times 2 \times 2$〜$5 \times 5 \times 5$ に限定したのは 2008 年 12 月現在日本で発売されているルービックキューブが $2 \times 2 \times 2$〜$5 \times 5 \times 5$ の 4 種類なためである。ルービックキューブの操作を行う場合には、回転させたい組み合わせを繋ぐエッジ上にマウスカーソルを移動させ、回転させたい方向にドラッグすることでできる。図 4.7 は何度か操作を繰り返した後の状態である。尚、操作を行った時にその操作が記録されるが、上部にあるテキストボックスに記入されている内容がコメントとして同時に記録される。このコメント機能は、操作時間の記録を補助する目的で作成した。

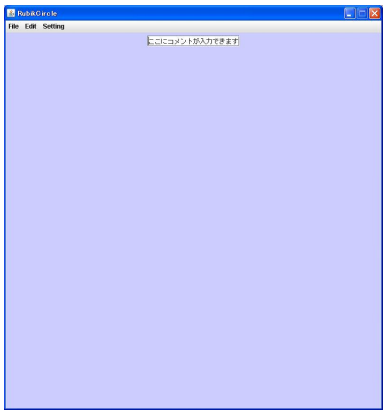
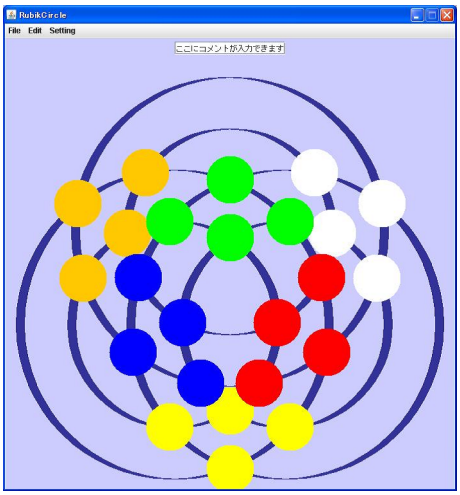


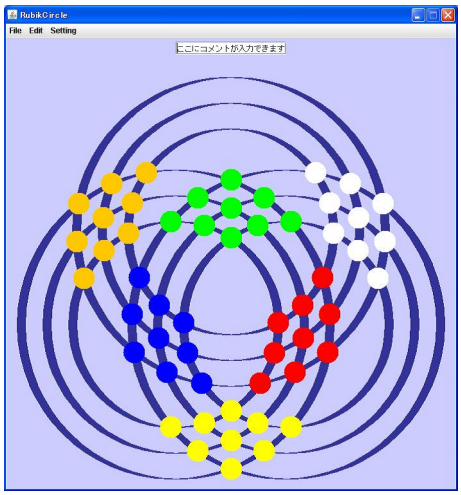
図 4.5: 起動直後の画面

次にファイルの入出力方法について説明する。操作履歴を保存するにはメニューから [File] → [Save] を選択すると、保存するファイル名を指定するダイアログが出てくるのでファイル名を指定することで操作履歴の保存ができる。また、ファイルに出力された操作履歴を読み込みツール上で再現する機能も作成した。メニューから [File] → [Open] を選択すると、読み込むファイルを指定するダイアログが出てくるので、操作履歴が保存されているファイルを選択すること自動的にツール上で再現される。

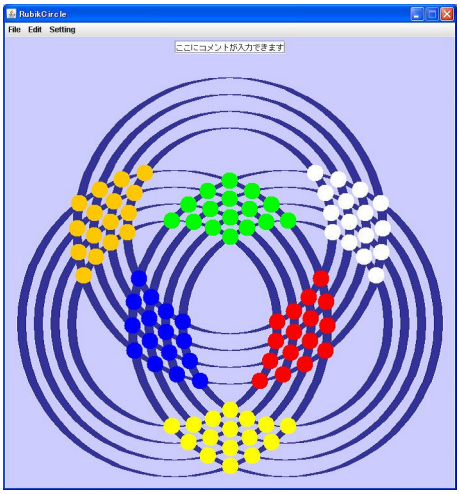
尚、予備実験で実際にこのツールを利用すれば操作記述の記録が可能であることを確認した。



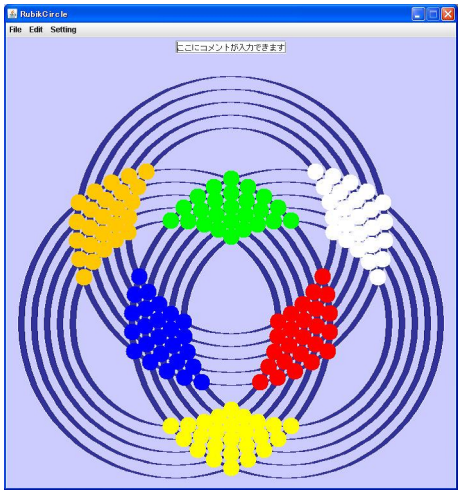
(a) $2 \times 2 \times 2$



(b) $3 \times 3 \times 3$



(c) $4 \times 4 \times 4$



(d) $5 \times 5 \times 5$

図 4.6: 各ルービックキューブの再現

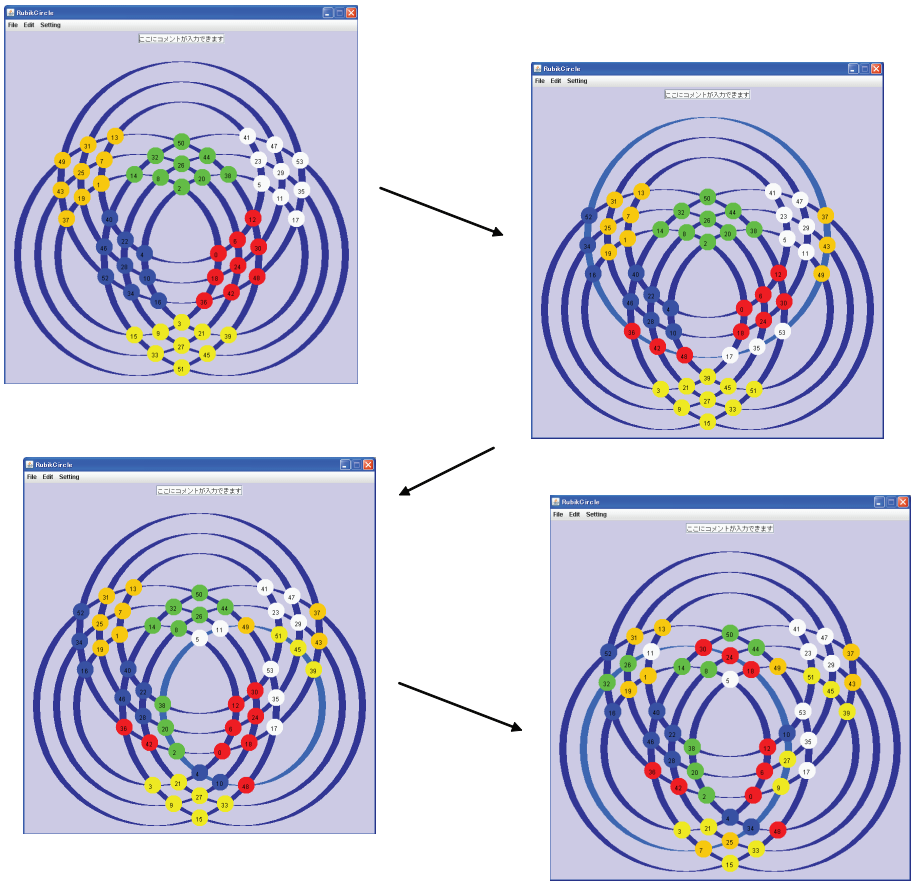


図 4.7: $3 \times 3 \times 3$ のルービックキューブを操作したときの推移

第5章　実験方法と結果

5.1　被験者について

実験の方法を説明する前に、本研究の実験の被験者について説明する。

まず、被験者はルービックキューブを完成させることが出来る人に限定した。これは、本研究では攻略法を知らない人がルービックキューブの攻略法を見つけ出すプロセスを分析するのではなく、攻略法をすでに知っている人がどのように攻略法を適用させていくかに重点を置いているためである。

更に、本研究ではルービックキューブを「勢いで解く」と答えた人が実際にはどのように解いているかを明らかにすることを目的の一つとしている。そのため実験は「論理的に解く」と答えた人（以降 A 氏と呼ぶ）と「勢いで解く」と答えたひと（以降 B 氏と呼ぶ）の 2 名を対象に行った。ルービックキューブをいかに早く解くかを競うスピードキュービックという競技がある。本研究の被験者である 2 名がスピードキュービックの競技者でない事を確認している。尚、2 名の被験者はどちらも 20 代男性である。

5.2　実験方法

本節ではどのような実験を行い、どのようなデータを取得したかについて説明する。

本研究の実験と実験結果の分析は本実験、データ作成、解析の 3 ステップに分かれる。それぞれのステップについて説明する。

5.2.1　本実験

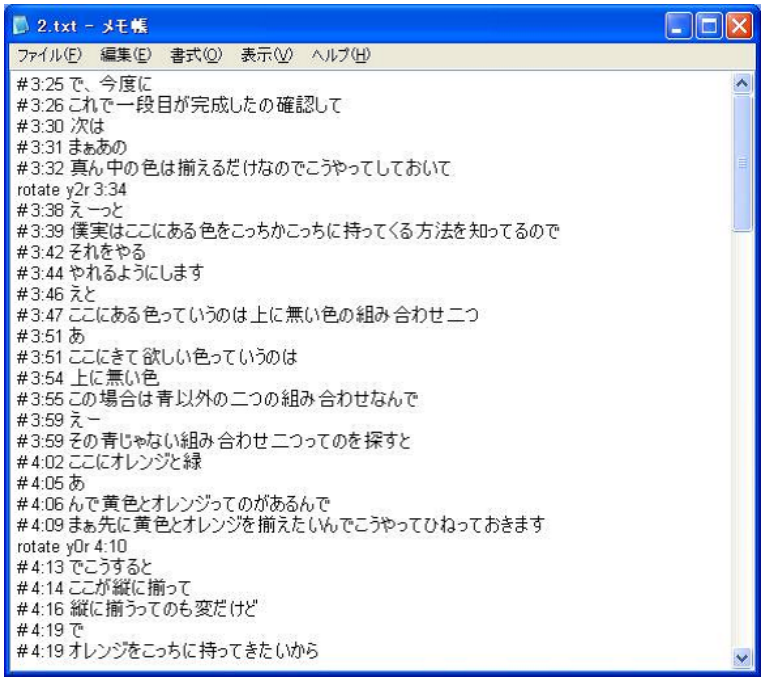
実験は被験者にルービックキューブを解いてもらい、同時に何を考えているかを発話してもらう。この時、被験者には次のように指示した。

- これから渡すルービックキューブを解いてください。
- どの色を揃えようとしてるか・どのような目的で動かすか等なんでもよいので考えてることを自由に話してください。

実際に行った実験を表 5.1 に示す。

表 5.1: 実験の回数と種類

	$2 \times 2 \times 2$	$3 \times 3 \times 3$
A 氏	2 回	1 回
B 氏		1 回


2.txt - メモ帳

ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)

#3:25 で、今度に
#3:26 これで一段目が完成したの確認して
#3:30 次は
#3:31 まぁあの
#3:32 真ん中の色は揃えるだけなのでこうやってしておいて
rotate y2r 3:34
#3:38 えーっと
#3:39 僕実はここにある色をこっちかこっちに持ってくる方法を知ってるので
#3:42 それをやる
#3:44 やれるようにします
#3:46 えと
#3:47 ここにある色っていうのは上に無い色の組み合わせ二つ
#3:51 あ
#3:51 ここにきて欲しい色っていうのは
#3:54 上に無い色
#3:55 この場合は青以外の二つの組み合わせなんで
#3:59 えー
#3:59 その青じゃない組み合わせ二つってのを探すと
#4:02 ここにオレンジと緑
#4:05 あ
#4:06 んで黄色とオレンジってのがあるんで
#4:09 まぁ先に黄色とオレンジを揃えたいんでこうやってひねっておきます
rotate y0r 4:10
#4:13 でこうすると
#4:14 ここが縦に揃って
#4:16 縦に揃うってのも変だけど
#4:19 で
#4:19 オレンジをこっちに持ってきたいから

図 5.1: 得られた実験データの一部

5.2.2 データ作成

本実験で撮影した動画を見ながら実験データの作成を行う。まず、4 章で作成した記述サポートツールを利用することで、被験者がルービックキューブに対して行った操作をすべて記述する。この時、サポートツールについているコメント機能を利用することで各操作が行われた時間を同時に記録していく。サポートツールから出力された操作ログに、発話内容を追加で記述する。発話をはじめた時間を同時に記録し、発話の切れ目は無言時間や意味の切れ目等から主観的に判断した。発話内容を記述した後でもサポートツールで直接読み込めるよう、4 章の表 4.1 に従った。具体的には、操作には「rotate」命令が付き、発話内容をコメントとして記述されている。作成されたデータは、時間情報・発話内容・捜査情報の 3 つで構成されている。図 5.1 に実際に得られた実験データの一部を示す。

表 5.2: 実験結果一覧

	$2 \times 2 \times 2$			$3 \times 3 \times 3$	
A 氏	1 回目	発話数	161	発話数	331
		操作数	53	操作数	116
		時間	6 : 28	時間	12 : 09
	2 回目	発話数	143		
		操作数	35		
		時間	6 : 11		
B 氏				発話数	50
				操作数	120
				時間	2 : 40

5.2.3 解析

得られた実験データを元に解析を行っていく。詳しい解析方法は次節以降に回し、本小節では解析の方針を説明する。表 5.1 に示してあるとおり、A 氏の 2 度、$2 \times 2 \times 2$ のルービックキューブ攻略のデータを取っている。これは、同じルービックキューブを攻略するときに再現性があるかを調べるためである。更に、A 氏には $3 \times 3 \times 3$ のルービックキューブ攻略のデータを取った。A 氏が、$2 \times 2 \times 2$ のルービックキューブ攻略時と $3 \times 3 \times 3$ のルービックキューブ攻略時でどのような違いが出るか、或いはどのような共通点があるのかを調べるためである。$3 \times 3 \times 3$ のルービックキューブ攻略データを A 氏と B 氏の双方で取ったのは、「論理的に解く」A 氏と「勢いで解く」B 氏が同じルービックキューブを攻略する時、どのような違いがあるかを調べるためである。

5.3 実験で集まったデータ

表 5.2 は実験の結果を簡単に纏めたものである。ここでもう一度確認するが、A 氏はルービックキューブを「論理的に解く」と言った人で、B 氏は「勢いで解く」と答えた人である。

この表を見るだけ、A 氏の $3 \times 3 \times 3$ のルービックキューブ攻略と B 氏の $3 \times 3 \times 3$ のルービックキューブ攻略では、操作数はほとんど変わらないが、発話数は A 氏が 331 回に対し、B 氏は 50 回と大きな差があることが分かる。更に、攻略にかかる時間も明らかに差がある。A 氏は $2 \times 2 \times 2$ のルービックキューブの攻略データを 2 回分集めたが、1 回目は操作数 53 回、2 回目は操作数 35 回と大きく差があるにもかかわらず、攻略にかかった時間は 6 : 28 と 6 : 11 でほとんど違いがない。次節からはこれらのデータを使って解析を進めていく。

5.4 ルービックキューブ攻略の再現性

本節では A 氏が $2 \times 2 \times 2$ のルービックキューブを攻略した際、再現性があったかどうかを考察していく。

表 5.2 を見ると分かるが、操作数が 1 回目の攻略時には 53 回、2 回目の攻略では 35 回と大きく差があり、単純に再現性があったとは言うことが出来ない。発話数は 1 回目には 161 回、2 回目は 143 回であった。操作数の比が約 10 : 7 になっているのに対し、発話数の比は約 8 : 7 となっている。単純に考えると、操作数が増えれば攻略に要する時間が増加し、それに従い発話数も増えるはずである。しかし、実際にはそのようになっていない。なぜこのような結果になったのかを調べていく。

5.4.1 操作数の違い

まずは操作数の違いについて調べてみる。A 氏が $2 \times 2 \times 2$ のルービックキューブを攻略する際に記録した発話データと操作ログを見ていると、1 回目の攻略時に次のような発話を見つけた。

3:10 さ、ちょっと間違えて
3:12 さっきやら無くていい事をやっちゃいました

この二つの発話から、操作にミスがあったことと、その操作ミスが本来ならばする必要がない操作だったことが分かる。A 氏にこの必要がない操作とはどの操作のことかを動画を見せて確認した。すると、操作ミスは間違いがあったという発言の前 8 回の操作だと言うことが分かった。更に、その操作ミスを直すために行った本来必要でない操作を 7 回していたことも分かった（図 5.2）。

この事から、A 氏は $2 \times 2 \times 2$ のルービックキューブ攻略時（1 回目）に、計 15 回の無駄な操作をしていたと言える。動画で確認したことで操作ミスと操作ミスを修正する操作を行った後に、操作ミスを行う前とまったく同じ状態になっているわけではないことも分かった。しかし、操作ミスを修正後には操作ミス前と似た状態になっているはずである。そのため、A 氏が $2 \times 2 \times 2$ のルービックキューブ攻略の 1 回目の操作数から、操作ミスとその修正に使った 15 回の操作数を引くと、操作ミスをしなかった場合必要だった操作数に近い値になると考えられる。元の操作数は 53 回なのでそこから 15 回を引くと、38 回となる。2 回目の攻略時には自身で自覚し、そうだと発話した操作ミスはなかった。1 回目の攻略は、本来ならば 38 回程度の操作で出来た。1 回目が 38 回程度の操作で攻略できたならば、2 回目の操作数 35 回とはそれほど違いがあるとはいえない。つまり、A 氏が $2 \times 2 \times 2$ のルービックキューブ攻略時、1 回目と 2 回目の操作数が 53 回と 35 回で大きく離れている原因は、1 回目の攻略の際に操作ミスをし、そのミスを修正するために本来攻略に不必要な操作をしたためだといえる。

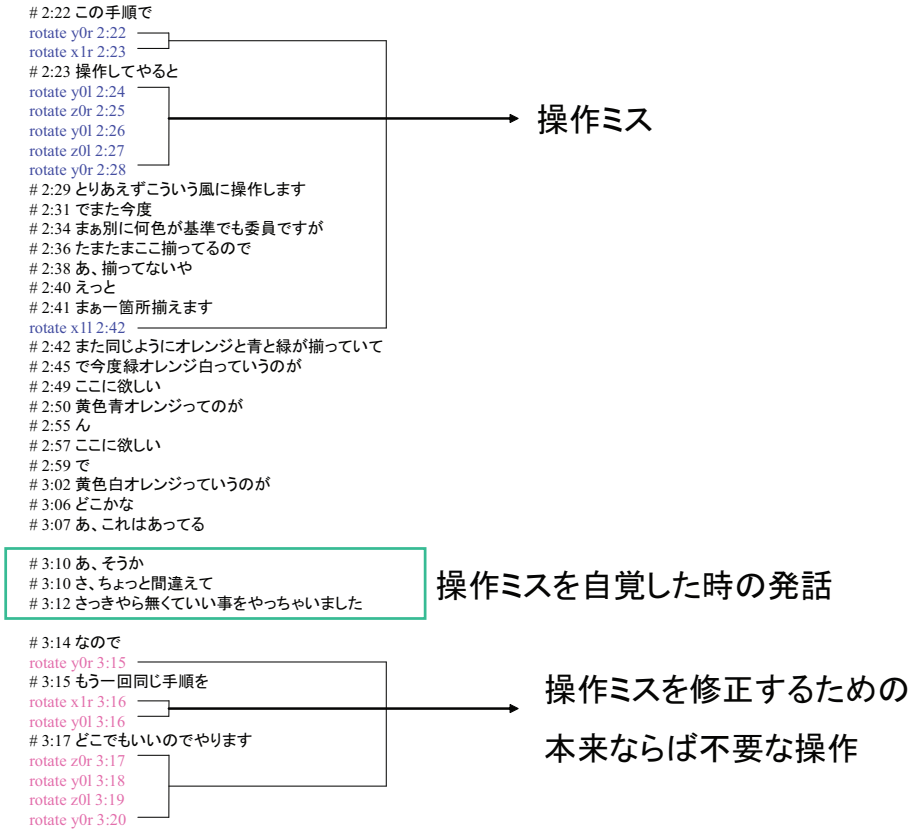
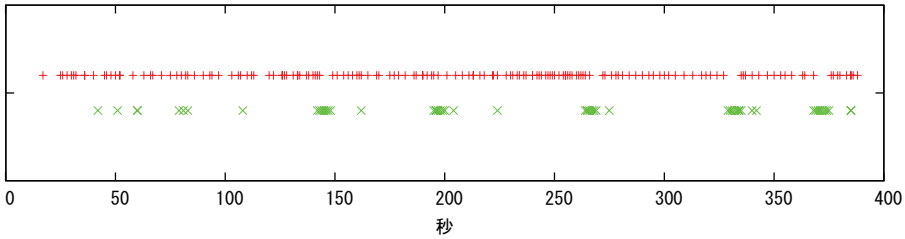


図 5.2: 操作ミスと不要な操作

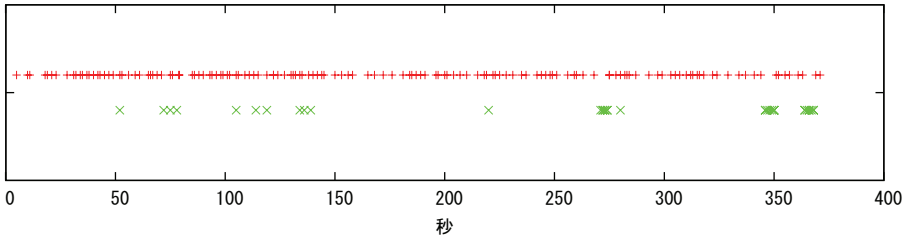
5.4.2 操作数・発話数と攻略に要した時間の関係

被験者にはあらかじめ考えている事を発話するように指示しているため、被験者は操作を決定するために思考し、その思考を発話という形で表出化している。そのため、操作が増えれば増えるほど発話は増加する。単純に考えると、操作に対して発話は一定の割合で増えるはずである。しかし、実際にはそうなっていない。なぜそのようになるかを明らかにしていく。

そのために、まず操作と発話がいつ行われているかを調べてみた。図 5.3 は A 氏がいつ操作と発話をしたかを示す図である。横軸に秒を取り、発話と操作が始まった時間にそれぞれ緑と赤でプロットした。この図を見ると、攻略中はほぼ途切れることなく発話を続けていることが分かる。攻略に要した時間はそれぞれ、6 分 28 秒と 6 分 11 秒であった。そして、発話数が 161 回と 143 回である。そこから、発話数の差は、約 6 分間で 20 回程度であったといえる。20 回というと、多く感じるが 1 分間当たり約 3 回であり「え〜っと」「そして」等の短い発話も 1 つの発話としてカウントしてるため誤差の範囲内といえる。攻略に要する時間は同程度であり、発話数も誤差の範囲内であるといえるため、攻略に要する時間は操作数よりも発話数に依存していたことが分かった。では、なぜ操作ミスをし



(a) A 氏の $2 \times 2 \times 2$ 攻略（1 回目）



(b) A 氏の $2 \times 2 \times 2$ 攻略（2 回目）

図 5.3: 発話と操作の開始時間

た 1 回目と操作ミスをしなかった 2 回目ので攻略に要する時間にほとんど差がなかったのか。それを調べるために、発話と操作について詳しく調べてみた。

5.4.3 操作と発話の関係

操作と発話の関係を調べるために、実験によって実際に得られた発話データと操作ログを見てみた。すると、図 5.4 のように、発話を何度か続け、次に操作を一度だけ行う時と発話を何度か続け次に操作を連続して行う時があることが分かった。 そこで、本研究では操作の開始時間と次の操作の開始時間が 2 秒以内のとき、その操作は連続していると定義し、連続した操作と呼ぶことにした（図 5.5）。単発の操作も連続回数が 1 回の連続した操作と考える。

ルービックキューブ攻略中に連続した操作がどれくらいあるのかを調べてみた。図 5.6 は連続した操作の出現順とその回数をグラフに表したものである。縦軸に連続操作回数をとり、ルービックキューブ攻略中に連続した操作があらわれた順に左から並べた。この図を見ると、連続した操作が 1 回目の攻略では 16 回、2 回目の攻略時には 14 回あったことが分かる。図 5.4 から、発話は、連続した操作の前にあることが分かる。そこで、一つの連続した操作に平均してどれくらい発話をしているかを調べてみた（表 5.3）。表 5.3 から分かるように、どちらも一度の連続した操作に対する発話数は約 10 回と同程度であっ

0:25 そうですね
0:26 この場合はちょっと
0:28 特に意味はないですけど
0:30 黄色から
0:31 あ、黄色じゃない
0:32 赤をそろえていきます
0:36 なので
0:36 ここに黄色と赤二つあるので
0:40 これそのまま持ってくればとりあえず赤二面が揃います

rotate y0r 0:42 一度だけ操作

0:45 で白がここにあって
0:46 まぁ赤揃えるので白と赤の組み合わせ

5:09 ここにある色とここにある色とここにある色が上に来る方法があります
5:13 今オレンジをひとつだけ上に出したいので
5:17 こうこうこうでやるとオレンジが二つきてしまう
5:19 こうこうこうでやれば
5:21 こことここここだとオレンジはひとつだけ
5:24 ひとつだけ上に来るので
5:27 その方法をやります

rotate x0l 5:29
rotate y1l 5:30
rotate y1l 5:31
rotate x1l 5:32
rotate y1l 5:33
rotate x0l 5:34
rotate z1l 5:34
rotate x0r 5:35

何度も連続して操作

5:35 どうでしょうか
5:36 オレンジが上だけきました

図 5.4: 単発の操作と連続した操作

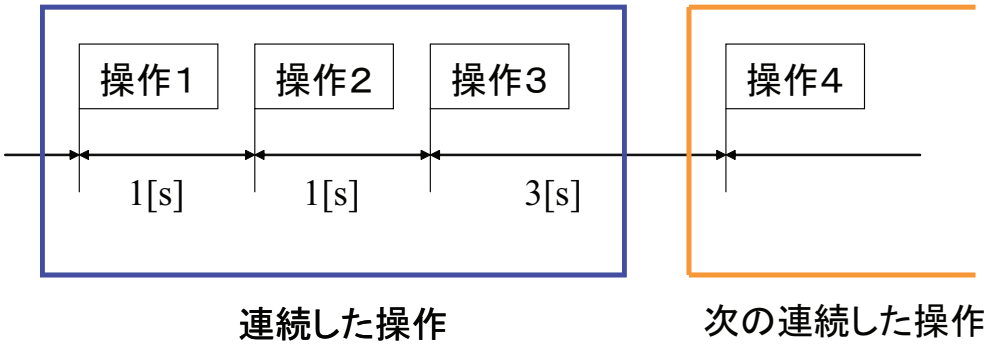
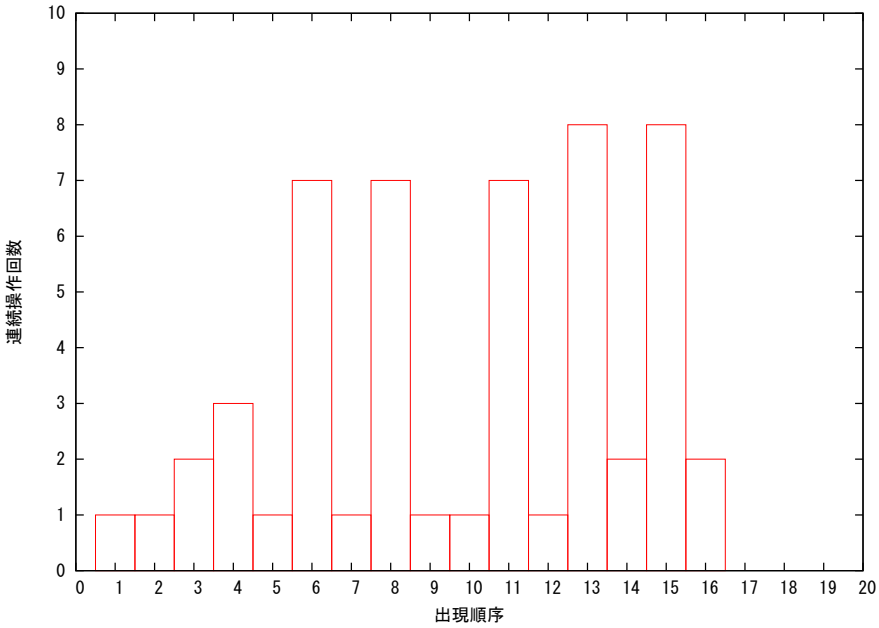


図 5.5: 連続した操作

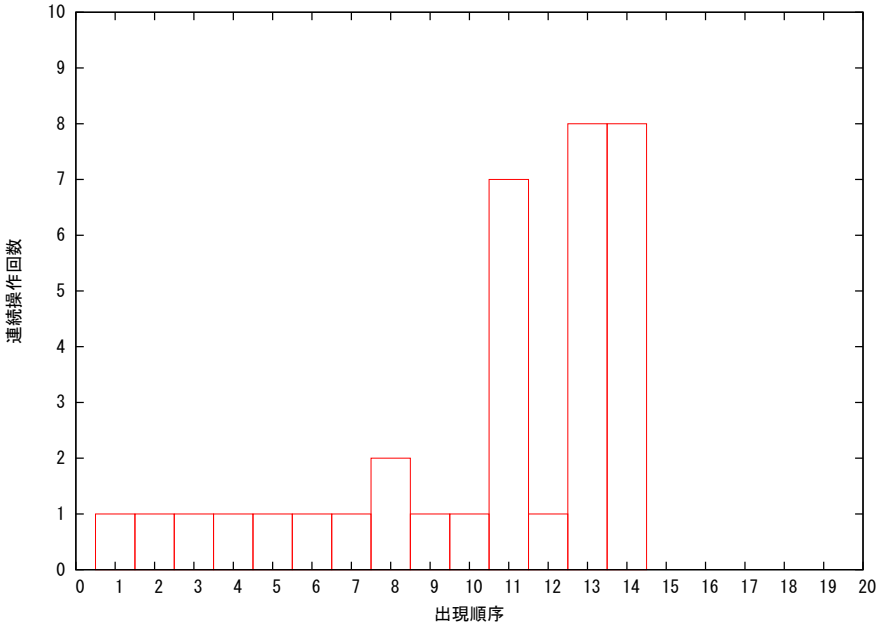
た。操作数が 53 回で発話数が 161 回と操作数が 35 回で発話数 143 回では、操作数と発話数のバランスに違いがあり、操作数と発話数は関係がないように思える。しかし、連続した操作という観点からみると、操作数と発話数は関わりが強いことが分かった。

表 5.3: 連続した操作一回当たりの発話数

	総発話数	連続した操作の出現数	連続した操作一回当たりの発話数
1 回目	163 回	16 回	10.19 回
2 回目	143 回	14 回	10.21 回



(a) A 氏の $2 \times 2 \times 2$ 攻略（1 回目）



(b) A 氏の $2 \times 2 \times 2$ 攻略（2 回目）

図 5.6: 連続した操作の出現順とその回数

4:08 要するに
4:09 一箇所だけ揃ってて
4:11 後は一個ずつ隣にずれるっていう風にしてやります
サブゴールの決定
4:16 でこれで今目的の形になったんで
4:19 でこの時に
4:20 この組み合わせ三箇所をこういう風に
4:23 この三つを一個隣一個隣二個隣って入れ替えてやる方法をがあるんで
4:28 それをやります
サブゴールを実現する操作の決定
rotate y1r 4:31
rotate x1l 4:32
rotate y0r 4:32
rotate x1r 4:33
rotate y1l 4:33
rotate x1l 4:34
rotate y0l 4:34
操作
4:35 はい
4:35 これで出来ました
4:38 で、え一っと
rotate x1r 4:40
操作
4:40 まぁ確認してみましょうか
4:42 まぁこの二個は調べるまでも無いですね
4:43 見たまんまなんで
4:44 組み合わせどころか上も揃ってます
4:47 でここは緑白赤ってのがきて正しい
4:53 ここは赤黄色白ってのがきてるから正しい
ルービックキューブの状態の確認
4:57 で今度はえ一っと
4:59 一箇所だけ白が上にきて残り全部白以外っていう風にやっていきます
5:03 今二箇所ここに白がきてるんで
5:05 これはダメなんで
ルービックキューブの状態の確認
5:07 えっと一
5:10 まぁ回してやって
5:12 えっと
5:13 一箇所白が来るようにします
サブゴールの決定
5:15 ここに白
5:16 まぁ白を一箇所にする方法なんですけど
5:18 一箇所基準というか動かない面ってのを固定してやります
5:22 まぁこのこれを
5:24 この黄色が見えている面を動かない面として扱ってやると
5:29 ここにある色とここにある色とここにある色を
5:34 上に持ってくる方法があるんでそれをやってやります
サブゴールを実現する操作の決定
5:37 そうするとこの白と緑とオレンジが上にきて
5:41 この黄色はそのままなんで
5:44 白が一箇所だけ上ってなるはずです
rotate y0r 5:46
rotate x1r 5:46
rotate x1r 5:47
rotate y0l 5:48
rotate x1r 5:49
rotate y0r 5:50
rotate x1r 5:50
rotate y0l 5:50
操作
5:51 はいできました
ルービックキューブの状態の確認
5:52 でこんどどうしてやるかっていうと
5:55 ここの白と
5:57 こことこことここを上に持ってくる方法があります
サブゴールの決定
6:01 まぁそれをやってやると
6:03 今度は全部白が上にきます
ルービックキューブの状態の確認

図 5.7: A 氏の実験データと意味づけ

5.4.4 ルービックキューブ攻略時の操作と思考のモデル

今まで分析を進めてきて、攻略時の思考と操作にはある程度の規則性があるのではないかと考えた。図 5.7 は A 氏の実験データの一部と発話等に意味づけを行ったものである。これを見ると、ルービックキューブの攻略は表 5.4 に纏めた行動の組み合わせであることが分かる。これらの行動の順番には規則性があるのかを調べてみる。尚、ここで言うサブゴールとは、「まずは赤を揃える」等のルービックキューブ攻略のために自ら設定した目標のことである。そこで、実験で得られたデータにラベル付けを行った。ここで利用したラベルは表 5.4 と同様である。データにラベル付けを行う時、連続して行われた操作や、複数の連続した発話を合わせて一つの意味になるものには 1 つのラベルしか付けていない。また、攻略に直接関係のない発話にもラベル付けは行わなかった。そのため、操作数と発話数の合計がラベル数と同数になっていない。

図 5.8 は A 氏の $2 \times 2 \times 2$ のルービックキューブ攻略から作成した発話・操作データにラベル付けを行い、そのラベルのみを表示したものである。まずは、図 5.8(a) のみに注目し

表 5.4: データに付けるラベルとその意味

ラベル名	意味
サブゴールの決定	「まずは赤を揃える」等の小さな目標の設定
操作の決定	サブゴールを達成するための操作の決定
状態を調べる	ルービックキューブの色を見て、状態を確認
操作	ルービックキューブを実際に操作

てみる。「状態を調べる」、「操作の決定」、「操作」の 3 つを続けて行うことが多いのが分かる。更に、「状態を調べる」を省略し、「操作の決定」、「操作」を行うこともある。普通に考えると、ルービックキューブの状態を知ることなく操作を決定することは出来ない。しかし、実際の実験データでは「状態を調べる」ことなく「操作を決定」を行っている。何故このようなことが起きたのだろうか。本研究では、思考の表出化を発話に頼っているため、無意識の思考や、言語化できない思考を表出化することは出来ない。そのため、実際には行っている「状態を調べる」という思考を省略されているのではないかと推測できる。ただし、実験データの中に、「後は目をつぶっていても解ける」という発話データがあった。つまり、「状態を調べる」ことなく「操作の決定」が行われる事がまったくないとは言えない。図 5.8(a) とそれに対する考察から、ルービックキューブ攻略時の思考と操作のモデルを作成した。図 5.9 に作成したモデルを示す。このモデルはルービックキューブを解く時の思考と操作が次のようになっている事を示している。まず「サブゴールを決定」する。次にルービックキューブの「状態を調べ」、ルービックキューブに行う「操作の決定」する。その後ルービックキューブの「操作」を行う。この時、場合によっては「操作の決定」と「操作」を繰り返すことがある。

図 5.8(b) を見ると、作成したモデルは A 氏の $2 \times 2 \times 2$ のルービックキューブ攻略 1 回目の実験データを元に作成したモデルが部分的に例外があるが、2 回目にも適応できることが分かる。そのため、A 氏のルービックキューブ攻略には再現性があると言える。

サブゴールの決定 -> 状態を調べる -> サブゴールの決定 -> 操作の決定 -> 操作 ->
操作の決定 -> 操作 -> 状態を調べる -> 操作の決定 -> 操作 ->
サブゴールの決定 -> 操作 -> 操作の決定 -> 操作 -> 操作の決定 ->
操作 -> 状態を調べる -> サブゴールの決定 -> 状態を調べる -> 操作の決定 ->
操作 -> 状態を調べる -> 操作の決定 -> 状態を調べる -> 操作の決定 ->
操作 -> 状態を調べる -> 操作の決定 -> 操作 -> 状態を調べる ->
操作の決定 -> 操作 -> 操作の決定 -> 操作 -> 状態を調べる ->
操作の決定 -> 操作 -> サブゴールの決定 -> 状態を調べる -> 操作の決定 ->
操作 -> 操作 -> 状態を調べる -> サブゴールの決定 -> 操作の決定 ->
操作 -> 状態を調べる -> 操作 -> 状態を調べる -> 操作の決定 ->
操作 -> 状態を調べる -> 操作の決定 -> 操作 -> 状態を調べる

(a) $2 \times 2 \times 2$（1 回目）のラベル

サブゴールの決定 -> 状態を調べる -> 操作の決定 -> 操作 -> 状態を調べる ->
操作の決定 -> 操作 -> 操作の決定 -> 操作 -> 操作の決定 ->
操作 -> 操作の決定 -> 操作 -> 状態を調べる -> 操作の決定 ->
操作 -> 状態を調べる -> 操作の決定 -> 操作 -> 操作の決定 ->
操作 -> 状態を調べる -> 操作の決定 -> 操作 -> 操作の決定 ->
操作 -> 操作の決定 -> 操作 -> 状態を調べる -> サブゴールの決定 ->
状態を調べる -> 操作の決定 -> 操作 -> 状態を調べる -> 操作の決定 ->
操作 -> 状態を調べる -> 操作 -> 状態を調べる -> サブゴールの決定 ->
状態を調べる -> 操作の決定 -> 操作の決定 -> 操作 -> 状態を調べる ->
操作の決定 -> 操作 -> 操作の決定-> 状態を調べる

(b) $2 \times 2 \times 2$（2 回目）のラベル

図 5.8: $2 \times 2 \times 2$ のルービックキューブ攻略データへのラベル付け

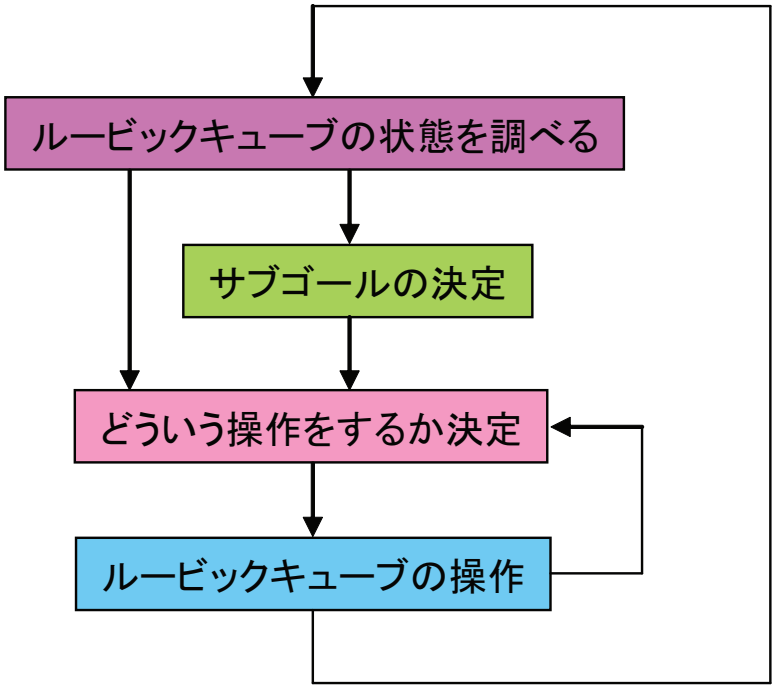
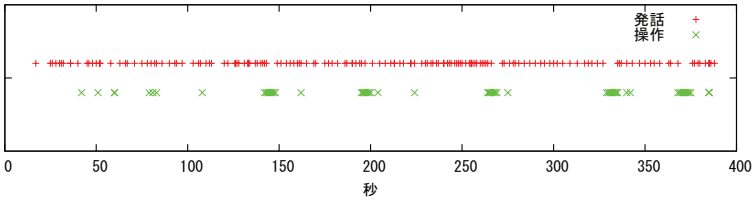
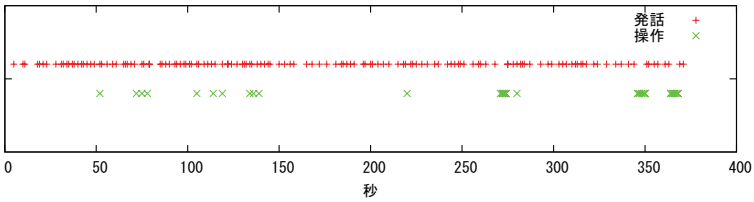


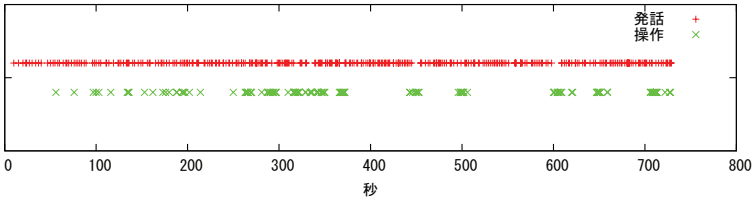
図 5.9: ルービックキューブ攻略時の思考と操作のモデル



(a) A 氏の $2 \times 2 \times 2$ 攻略（1 回目）



(b) A 氏の $2 \times 2 \times 2$ 攻略（2 回目）



(c) A 氏の $3 \times 3 \times 3$ 攻略

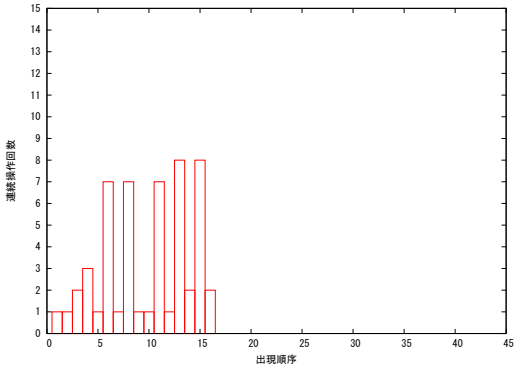
図 5.10: 発話と操作を時間軸上にプロット

5.5 大きさの違うルービックキューブ

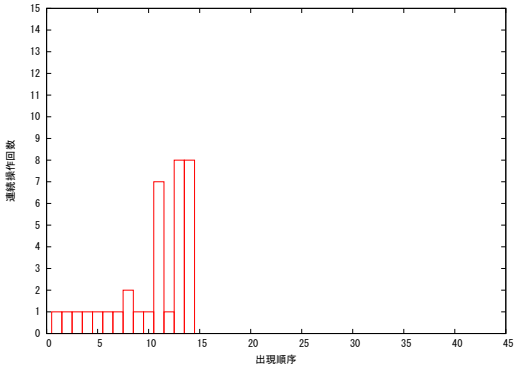
本節では、A 氏の $2 \times 2 \times 2$ のルービックキューブと $3 \times 3 \times 3$ のルービックキューブ攻略の様子を比較する。

まず、操作と発話が攻略中にどのようになっていたかを調べてみることにした。図 5.10 は、時間軸上に操作と発話の始まった時間をそれぞれプロットしたものである。これを見ると、$2 \times 2 \times 2$ のルービックキューブ攻略時と同様に、$3 \times 3 \times 3$ のルービックキューブ攻略時に攻略中はほぼ絶え間なく発話を続けていることが分かる。更に、操作に注目すると、まったく操作をしていない期間が存在していることも分かる。つまり、その期間の前後で操作が途切れているといえる。そこで、連続した操作について調べてみた。

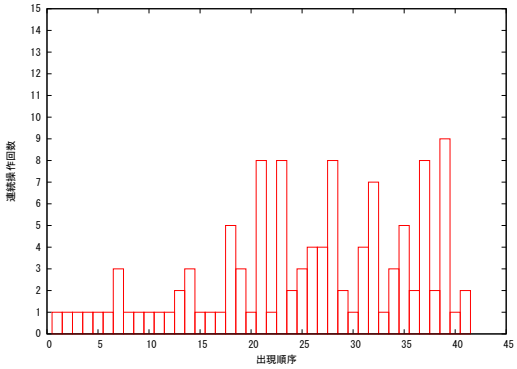
図 5.11 は A 氏がルービックキューブ攻略した時の操作から連続した操作を見つけ、それを出現順に並べたものである。これを見ると、$2 \times 2 \times 2$（2 回目）と $3 \times 3 \times 3$ 攻略時は、前半は連続操作数が少ないのに対し、後半では連続操作数が多くなっている。$2 \times 2 \times 2$



(a) A 氏の $2 \times 2 \times 2$ 攻略（1 回目）



(b) A 氏の $2 \times 2 \times 2$ 攻略（2 回目）



(c) A 氏の $3 \times 3 \times 3$ 攻略

図 5.11: 連続した操作とその出現順

（1 回目）は、他の 2 つほどはっきりと差が現れてないが、同様の傾向がある。なぜこのように、前半と後半ではっきりとした違いが出るのだろうか。

5.5.1 序盤の戦略とそれ以降の戦略

まず、はっきりと違いが現れている $2 \times 2 \times 2$（2 回目）を調べてみる。図 5.11(b) から、操作の連続回数増えるのは 11 回目の連続した操作だということが分かる。実際の操作数に直すと、12 回目の操作から長く連続した操作が始まっている。そこで、実際に 12 回目の操作が始まる直前ではルービックキューブの状態と発話がどうなっているかを調べた。図 5.12 を見ると、11 回目の操作の 1 つ前で 1 面を完成させていることが分かる。更に、12 回目以降の操作から 2 段目以降をそろえようとしている。図 5.12 は 11 回目の操作が終わった直後の状態だが、発話でも示している通りに 1 面とその面に隣接する部分の色が

rotate z11 2:19 ← 10回目の操作

2:20 こういう風にやるとこうやって青が揃います
2:22 で、まぁあの
2:24 やってる最中に
2:25 一面を揃える最中についでに横のこの部分もそろえてあるんで
2:30 これで二列揃ったことになります
2:33 次は上を揃えてくんですけど
2:36 そろえるときに何をやるかって言うと
2:38 色の組み合わせが揃っているかっていうのを確かめます
2:45 まぁここの場合なんかだと
2:48 黄色と赤と白っていう風になってて
2:52 ここは色の組み合わせは揃ってます
2:56 ここに来るべきキューブは上に白もきちゃってますけど
3:01 組み合わせは正しいってことになります
3:04 で、別のところもみてくと
3:05 ここの横は黄色とオレンジなんで
3:07 まぁ上はしろ
3:09 上は全部白が来るのが決まってるんで
3:11 ここに黄色と白とオレンジって組み合わせができてれば正しいわけです
3:16 でこれは
3:17 黄色と白とオレンジができてるんで正しいです
3:20 で、今度こっち
3:21 横がオレンジと緑で上白なんで
3:24 オレンジ白緑ってのが来なくちゃいけないんですけど
3:27 赤白緑ってのができてるんで間違ってます
3:30 でこっちも赤緑白がこなくちゃいけないのに
3:35 違う色ができてるんで間違ってます
3:38 この場合はえーっと
3:39 一個ずらしてやって

1面＋一段目を揃えたことを確認する発話

rotate x1r 3:40 ← 11回目の操作

3:42 でそうすると
3:43 一箇所見たときに
3:45 白赤緑はここで正しい
3:48 でここは黄色白オレンジってのは
3:51 一個となり
3:55 で、今度その隣の部分にある
3:57 白赤黄色ってのが一個となり
4:02 でその隣
4:04 今度こっちにある白緑オレンジってのが
4:06 二個となり
4:08 要するに
4:09 一箇所だけ揃ってて
4:11 後は一個ずつ隣にずれるっていう風にしてやります
4:16 でこれで今目的の形になったんで
4:19 でこの時に
4:20 この組み合わせ三箇所をこういう風に
4:23 この三つを一個隣一個隣二個隣って入れ替えてやる方法をがあるんで
4:28 それをやります

二列目を揃えるための思考

rotate y1r 4:31 ← 12回目の操作
rotate x1l 4:32
rotate y0r 4:32
rotate x1r 4:33
rotate y1l 4:33
rotate x1l 4:34
rotate y0l 4:34

図 5.12: $2 \times 2 \times 2$（2 回目）の 11 回目の操作前後の発話

揃っており、2 段目が揃っていないことが分かる。この事から、1 面を揃え終わり、2 段目以降を揃える時を境に連続操作数が変化するのではないかと考えた。それを確認するために、$2 \times 2 \times 2$ の 1 回目の攻略と $3 \times 3 \times 3$ の攻略でも同様に 1 面とそれ以降で連続操作数が変化するかを調べた。

攻略時の発話から、実際に 2 段目を揃え始めるのは $2 \times 2 \times 2$（1 回目）は 9 回目の操作から、$3 \times 3 \times 3$ では 23 回目の操作からだということが分かった。つまり、それぞれ 1 面とその面に隣接する部分の色を揃えるために、それぞれ 8 回と 22 回必要だったといえる。$2 \times 2 \times 2$ の 8 回目の操作と $3 \times 3 \times 3$ の 22 回目の操作が、何個目の連続した操作に当たるかを調べた（図 5.13）。$2 \times 2 \times 2$（1 回目）では 5 回目、$3 \times 3 \times 3$ では 17 回の連続した操作で 1 面を完成させていた。図 5.11 を見ると、$2 \times 2 \times 2$（1 回目）では 6 回目、$3 \times 3 \times 3$ では 18 回から連続操作回数が多くなっている。つまり、A 氏は、ルービックキューブの 1 面を揃えるまでは比較的短い操作を繰り返し、2 段目以降を揃える時は長い操作を繰り返しながら攻略していることが分かった。

rotate yl1 1:23 ← 7回目の操作

1:23 という風にやってとりあえず一面をそろえます
1:26 で横もしっかり揃ってることを確認して
1:30 次上を揃えていきます
1:33 えーっと
1:34 まず一箇所組み合わ
1:37 色の組み合わせそろえるってことをします
1:43 ここに緑と青と
1:46 上はオレンジですかね
1:47 オレンジが来るようにします

1面＋一段目を揃えたことを確認する発話

rotate x1r 1:48 ← 8回目の操作

1:50 でここの色の組み合わせは揃いました
1:52 次に
1:53 他のところも同じように色の組み合わせをそろえるって事をやっていきます
2:00 ここに黄色とろとオレンジがありますが
2:02 黄色と白とオレンジの組み合わせは本当はここになくてはいけません
2:06 んで
2:06 オレ
2:07 このL
2:08 このオレンジと白と緑の組み合わせは
2:11 こっちにになくちゃいけない
2:13 で
2:13 ここの組み合わせは
2:14 黄色オレンジ青であってる
2:17 この場合
2:18 えーっと
2:20 一回
2:21 こう
2:22 この手順で

二段目の揃えるための思考

rotate y0r 2:22 ← 9回目の操作
rotate x1r 2:23
2:23 操作してやると
rotate y0l 2:24
rotate z0r 2:25
rotate y0l 2:26
rotate z0l 2:27
rotate y0r 2:28

(a) A 氏の $2 \times 2 \times 2$ 攻略（2 回目）

rotate y2r 3:34 ← 7回目の操作

3:38 えーっと
3:39 実はここにある色をこっちかこっちに持ってくる方法を知ってるので
3:42 それをやる
3:44 やれるようにします
3:46 えと
3:47 ここにある色っていうのは上に無い色の組み合わせ二つ
3:51 あ
3:51 ここにきて欲しい色っていうのは
3:54 上に無い色
3:55 この場合は青以外の二つの組み合わせなんで
3:59 えー
3:59 その青じゃない組み合わせ二つってのを探すと
4:02 ここにオレンジと緑
4:05 あ
4:06 んで黄色とオレンジってのがあるんで
4:09 まあ先に黄色とオレンジを揃えたいんでこうやってひねっておきます

rotate y0r 4:10 ← 22回目の操作

4:13 でこうすると
4:14 ここが縦に揃って
4:16 縦に揃うってのも変だけど
4:19 で
4:19 オレンジをこっちに持ってきたいから
4:21 その時は

rotate y0l 4:23 ← 23回目の操作
4:23 こうやって動かしてやると
rotate z2r 4:24
rotate y0r 4:25
rotate z2l 4:26

発話には出ていないが実際に
二段目を揃え始めたのはここから
（動画で確認）

(b) A 氏の $3 \times 3 \times 3$ 攻略

図 5.13: 2 段目を揃え始める時の発話と操作

状態を調べる -> サブゴールの決定 -> 状態を調べる -> サブゴールの決定 -> 操作の決定 ->
操作 -> サブゴールの決定 -> 操作 -> 状態を調べる -> サブゴールの決定 ->
操作 -> 操作の決定 -> 操作 -> 操作の決定 -> 操作 ->
状態を調べる -> 操作の決定 -> 操作 -> サブゴールの決定 -> 操作の決定 ->
操作 -> 操作の決定 -> 操作 -> 操作の決定 -> 操作 ->
状態を調べる -> 操作の決定 -> 操作 -> 状態を調べる -> 操作の決定 ->
操作 -> 操作の決定 -> 操作 -> 操作の決定 -> 操作 ->
操作の決定 -> 操作 -> 状態を調べる -> 操作 -> 操作の決定 ->
操作 -> 操作の決定 -> 操作 -> 操作 -> 操作の決定 ->
状態を調べる -> 操作の決定 -> 操作 -> サブゴールの決定 -> 状態を調べる ->
操作の決定 -> 操作 -> 状態を調べる -> サブゴールの決定 -> 操作 ->
状態を調べる -> サブゴールの決定 -> 操作 -> 状態を調べる -> 操作 ->
サブゴールの決定 -> 操作 -> 状態を調べる -> 操作 -> 操作の決定 ->
操作 -> サブゴールの決定 -> 操作 -> 状態を調べる -> 操作 ->
操作 -> 状態を調べる -> サブゴールの決定 -> 状態を調べる -> 操作 ->
状態を調べる -> サブゴールの決定 -> 状態を調べる -> 操作の決定 -> 操作 ->
操作の決定 -> 操作 -> 状態を調べる -> 操作の決定 -> 操作 ->
状態を調べる -> 操作の決定 -> 操作 -> 操作 -> 状態を調べる ->
サブゴールの決定 -> サブゴールの決定 -> 状態を調べる -> 操作の決定 -> 操作 ->
状態を調べる -> 操作の決定 -> 操作 -> 状態を調べる -> 操作の決定 ->
操作 -> 操作の決定 -> サブゴールの決定 -> 操作 -> 状態を調べる ->
操作の決定 -> 操作 -> 操作 -> 状態を調べる -> 操作の決定 ->
操作 -> 操作の決定 -> 操作

図 5.14: A 氏の $3 \times 3 \times 3$ のルービックキューブ攻略時のラベル

表 5.5: $3 \times 3 \times 3$ のルービックキューブ攻略

	$3 \times 3 \times 3$	
A 氏	発話数	331
	操作数	116
	時間	12 : 09
B 氏	発話数	50
	操作数	120
	時間	2 : 40

5.5.2 作成したモデルの適応

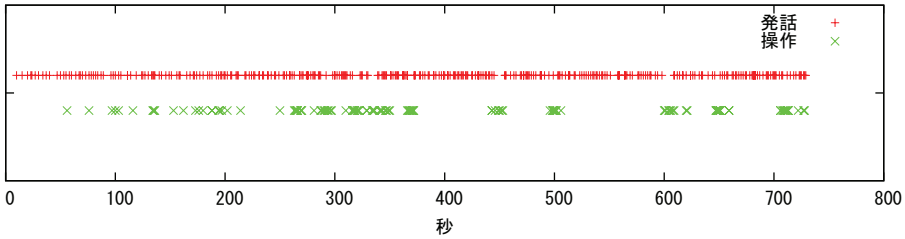
本研究では既に、A 氏の $2 \times 2 \times 2$ のルービックキューブ攻略時に得られた実験データからルービックキューブ攻略時の操作と思考のモデルを作成している（図 5.9）。このモデルが A 氏が $3 \times 3 \times 3$ のルービックキューブ攻略時にも適応することが出来るのかを調べる。

図 5.14 は A 氏の $3 \times 3 \times 3$ のルービックキューブ攻略時に得られた実験データにラベルを付け、そのラベルだけを取り出したものである。尚、この時のラベルの付け方は 5.4.4 節で示したものと同様である。ラベルの内容を見てみると、$2 \times 2 \times 2$ の時と同様に、「状態を調べる」「操作の決定」「操作」の 3 つがセットになりあらわれている事がわかる。更に、「サブゴールの決定」も全体を通して見つかり、5.4.4 節で示した図 5.9 が適応できるといえる。

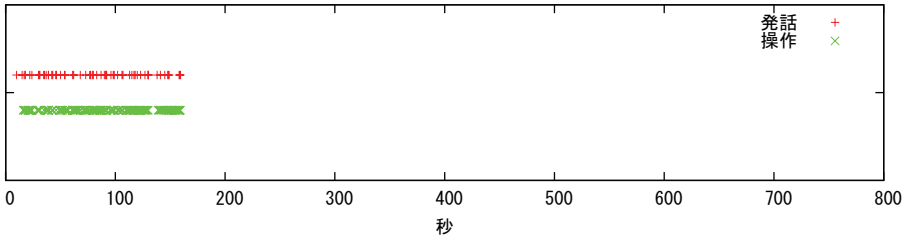
5.6 $3 \times 3 \times 3$ のルービックキューブの攻略者による違い

「論理的に解く」という A 氏と、「勢いで解く」という B 氏が同じルービックキューブを解くときにどんな違いがあるかを調べる。

表 5.5 にもう一度 $3 \times 3 \times 3$ のルービックキューブ攻略時の操作回数と発話数、攻略に要した時間を纏めた。この表を見ると、操作数は A 氏が 116 回、B 氏は 120 回とほとんど差はない。しかし、発話数と攻略時間には大きな差がある。操作数と発話数がどのような関係になっているのかを視覚的に確認するため、操作と発話の開始時間をグラフに示した（図 5.15）。既に説明したように、A 氏は操作をしていない期間がかなりある。しかし、B 氏は A 氏よりもかなり短い時間で同程度の操作をしているため、操作をしていない期間がほとんどないように見える。本当にそうなのだろうか。図 5.15 は比較のためにグラフのスケールを合わせてあるため分かりにくい。そこで、B 氏のみを詳しく見るため、図 5.16 に B 氏の操作と発話をグラフのスケールを変更し書き直した。図 5.16 を見ると、確かに操作をしていない期間が存在する。しかし、A 氏はそれがはっきりと分かり、何度も出現するのに対し、B 氏は操作していない期間が分かりづらく、その出現回数も少ない。



(a) A 氏の $3 \times 3 \times 3$ 攻略



(b) B 氏の $3 \times 3 \times 3$ 攻略

図 5.15: 発話と操作の開始時間

グラフのスケールを変更することで先ほどよりは分かりやすくなったが、まだ分かりにくい。そこで連続した操作がどのようになってるのかを調べてみることにした。

5.6.1 連続した操作

既に、「論理的に解く」A 氏は 1 面を揃えるまでとそれ以降では操作の連続する回数が明らかに違う事を示している。では、「勢いで解く」B 氏にも同じようなことは言えるのだろうか。それを調べるために B 氏が連続した操作をどれくらい行っていたかを調べてみた。図 5.17 は B 氏の連続した操作とその出現回数を示すグラフである。比較のため A 氏

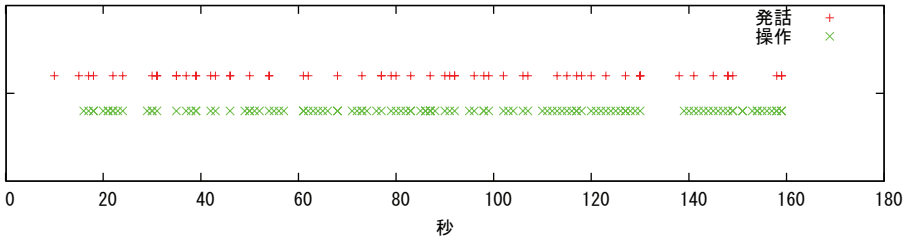
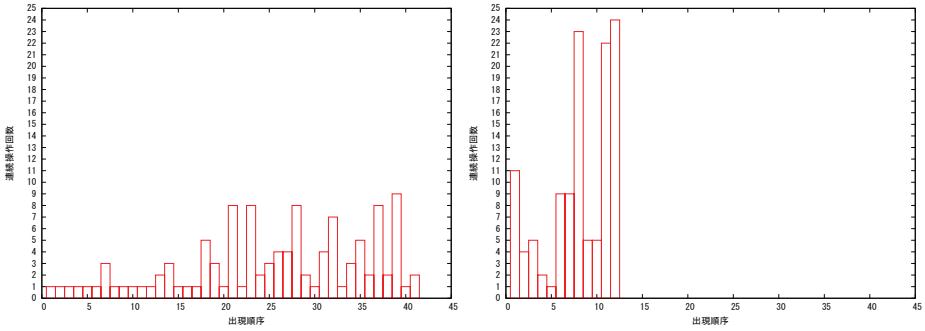


図 5.16: B 氏の発話と操作の開始時間



(a) A 氏の $3 \times 3 \times 3$ 攻略　　(b) B 氏の $3 \times 3 \times 3$ 攻略

図 5.17: A 氏と B 氏の連続した操作

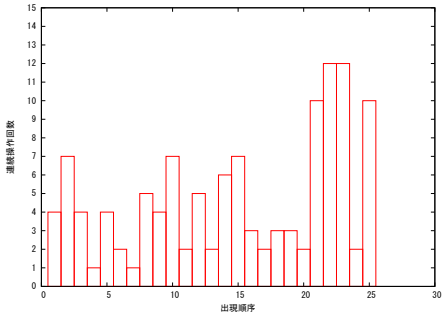


図 5.18: B 氏の連続した操作（1 秒以内）

のグラフも記した。グラフを見ると二人には明らかな違いがある。両氏とも総操作回数は約 120 回で違いはないが、連続した操作を行った回数が A 氏が 41 回であるのに対し、B 氏は 3 分の 1 以下の 12 回しかない。更に、B 氏は最初の連続操作回数が 11 回もあり、A 氏とはまったく違う結果になっていることが分かる。ところで、A 氏は 12 分 09 秒で 116 回の操作をしているのに対し、B 氏は 2 分 40 秒で 120 回の操作をしている。つまり、B 氏は A 氏よりも操作を行う密度が高いといえる。連続した操作は 2 秒以内に行われた操作であると定義した。この 2 秒というのは、A 氏には適切だったが、A 氏よりも操作密度の高い B 氏には適正な値でなく、本来別の連続した操作に分類されるべき操作が同じ連続した操作に属してしまった可能性がある。そこで、閾値を 1 秒以内とした場合の B 氏の連続した操作を調べてみた。図 5.18 はある操作の開始から次の操作の開始までの時間が 1 秒以内の時を連続した操作としたときの B 氏の連続操作回数とその出現順を示すグラフである。閾値を低く設定したため、図 5.17(b) と比べると連続操作回数は短くなっている。しかし、A 氏は明らかに序盤では連続操作回数が少なく、それ以降では連続操作回数が多いのに対して、B 氏にはそのような現象が見られなかった。

サブゴールの決定 -> 操作 -> 操作 -> 状態を調べる -> 操作 ->
状態を調べる -> 操作 -> サブゴールの決定 -> 操作 -> 状態を調べる ->
操作 -> サブゴールの決定 -> 操作 -> 状態を調べる -> 操作 ->
サブゴールの決定 -> 操作の決定 -> 操作 -> 状態を調べる -> 操作 ->
操作の決定 -> 操作 -> 状態を調べる -> 操作 -> 状態を調べる ->
サブゴールの決定 -> 操作 -> 操作の決定 -> 操作 -> 操作の決定 ->
操作 -> 状態を調べる -> サブゴールの決定 -> 操作 -> 状態を調べる ->
操作 -> 操作の決定 -> 状態を調べる -> 操作の決定 -> 操作 ->
操作の決定 -> 操作 -> 操作 -> 操作の決定 -> 操作 ->
状態を調べる

図 5.19: B 氏の $3 \times 3 \times 3$ のルービックキューブ攻略時のラベル

5.6.2 A 氏の実験データから作成されたモデルの適応

5.4.4 節で A 氏の $2 \times 2 \times 2$ のルービックキューブ攻略時の実験データからモデルを作り、5.5.2 節ではそのモデルが $3 \times 3 \times 3$ のルービックキューブ攻略時にも当てはまることを明らかにした。本小節では A 氏の実験データから作られたモデルが、B 氏でも通用するのかを調べる。図 5.19 は B 氏が $3 \times 3 \times 3$ のルービックキューブを攻略する時に得られた実験データにラベルをつけ、そのラベルのみを示したものである。ラベルの意味やラベル付けの方法は 5.4.4 節と同様の方法で行った。「勢いで解く」と答えた B 氏は、「論理的に解く」A 氏と違い操作を体系的に説明できないと考えていた。しかし、図 5.19 を見ると分かるように、A 氏の時と同様に部分的に「状態を調べる」「操作の決定」「操作」を繰り返していた。だが、A 氏の時と違い「状態を調べる」の直後に「操作」を行っていることが多かった。A 氏の実験データを元に作成したモデルは B 氏には当てはまらないのだろうか。しかし、いくら「勢いで解く」と言っても状態を調べている以上、その状態に必要な操作を決定せずに出鱈目な操作を行っているとは考えられない。つまり、B 氏はルービックキューブの状態を確認は意識的に行っているが、操作の決定を無意識、若しくは言語化できない方法で行っていると考えるのが妥当である。従って、B 氏の操作と思考も 5.4.4 節の図 5.9 に示すモデルが当てはまると言える。つまり、「論理的に解く」A 氏と「勢いで解く」B 氏も表現方法や、自分の意識するレベルでは違いがあるが、本質的には同様の方法でルービックキューブを解いていることが明らかになった。

5.7 操作ミスに気付いたとき

次は、ルービックキューブ攻略中に、操作ミスが見つかった時にどのように対処しているかを調べた。図 5.20 は A 氏と B 氏が操作ミスを自覚した発話と、その前後の操作を抜き出したものである。操作ミスをした場合、単純にその直前に行った連続した操作の、逆操作をしてやれば操作ミスをする前の状態に戻る。しかし、図 5.20 をみると、操作ミスを自覚した発話（緑枠で囲まれている部分）の前後で直前の操作の逆操作を行っていない。実験を行った範囲内で操作ミスが発話として表出化した箇所をすべて調べたが、その中に

```
rotate y0r 2:22
rotate x1r 2:23
rotate y0l 2:24
rotate z0r 2:25
rotate y0l 2:26
rotate z0l 2:27
rotate y0r 2:28
rotate x1l 2:42

# 3:10 あ、そうか
# 3:10 さ、ちょっと間違えて
# 3:12 さっきやら無くていい事をやっちゃいました

rotate y0r 3:15
# 3:15 もう一回同じ手順を
rotate x1r 3:16
rotate y0l 3:16
# 3:17 どこでもいいのでやります
rotate z0r 3:17
rotate y0l 3:18
rotate z0l 3:19
rotate y0r 3:20
```

(a) A 氏の操作ミス

```
rotate y0r 2:19
rotate x2r 2:20
rotate y0r 2:21
# 2:19 まぁ
rotate x2l 2:22
rotate y0r 2:23
rotate x2l 2:24
rotate y0r 2:25
# 2:25 こうやってやると
rotate x2r 2:26
rotate y0l 2:27
rotate x2r 2:28

# 2:28 あれ
# 2:28 間違えちゃった

rotate y0l 2:29
rotate y0l 2:29
# 2:29 勢いでやらないと間違える
rotate y0r 2:31
rotate x2r 2:31
rotate y0r 2:33
rotate x2l 2:34
rotate y0r 2:34
rotate x2l 2:35
rotate y0r 2:36
rotate x2r 2:37
rotate y0l 2:38
```

(b) B 氏の操作ミス

図 5.20: 表出化した操作ミスとその前後の操作

逆操作を行い操作ミスを修正していたものは無かった。この事から、操作ミスが見つかってもそれを逆操作で修正することなく、操作ミスが見つかった段階でルービックキューブの状態を調べ、新たにサブゴールを決定しているということが分かった。

第6章　結言

6.1　まとめ

本研究では、「ルービックキューブをどのように解いているのか」という疑問から始まり、それを実験により明らかにするという事を目的に進めてきた。実験に必要なルービックキューブの操作ログを取るための方法として、操作の記述方法を考案した。そして、操作の記述を行うために必要なツールの開発と、そのツールに利用するルービックキューブを二次元展開する方法の一つとし、同心円展開法を考案した。

ルービックキューブ攻略の様子をビデオに撮影し、同時にプロトコル分析を行い思考を表出化する事を試みた。その時に撮影したビデオを見ながら、開発したツールを利用し、操作の記述も行った。その結果得られたデータを元に操作と思考のモデルを作成し、同じ被験者が $2 \times 2 \times 2$ のルービックキューブを解く時に再現性があるかを調べた。さらに、$2 \times 2 \times 2$ のルービックキューブと $3 \times 3 \times 3$ のルービックキューブ攻略に違いがあるかを調べ、一段目を揃えるまでは共に一手ずつ確認しながら操作し、それ以降は操作を連続して行なっていることを明らかにした。それだけでなく、$2 \times 2 \times 2$ のルービックキューブ攻略時の実験データを元に作成した操作と思考のモデルが、$3 \times 3 \times 3$ のルービックキューブ攻略時にも当てはまる事を明らかにした。そして、「勢いで解く」と言った人と「論理的に解く」と言った人の双方の $3 \times 3 \times 3$ のルービックキューブ攻略時のデータを比較した。ルービックキューブ攻略時の思考と操作のモデルを利用することで「論理的に解く」人と「勢いで解く」人は表現方法や思考の表出化の度合いに違いがあるが、本質的には同様の方法でルービックキューブを解いている事を明らかにした。

それとは別に、ルービックキューブ攻略中、操作ミスに気付いても逆操作でその操作ミスを修正するのではなく、操作ミスに気付いた時点でルービックキューブの状態を確認し、新たにサブゴールを設定していることが明らかになった。

6.2　今後の課題

本研究では思考の分析のためにプロトコル分析を用いた。ルービックキューブを「論理的に解く」と言った人は、意識して操作を決定しているため発話プロトコルとして思考の表出化が行われた。しかし、「勢いで解く」と言った人は解き方をそれほど強く意識していないため、発話プロトコルを十分に集めることが出来なかった。そのため、自分で意識していない思考（操作の決定）をどのように表出化するかは今後の課題である。

また、本研究では軸を基準とし、ルービックキューブ自体の回転などを考慮しない絶対的な操作記述法を用いた。しかし、実際に操作を行う者はルービックキューブ自体を回転させ、現在自分が見ている面を基準とした相対的な操作を意識している可能性がある。そのため、相対的な操作を考慮した記述等、より被験者の思考が反映する分析を行う事も今後の課題の一つといえる。

謝辞

本研究を進めるにあたり、多くの方にご協力頂いた。特に、主指導教官である杉山先生をはじめ、由井薗先生、小倉先生には熱心に指導して頂き、深く感謝する次第である。また、実験に協力して下さった被験者の皆様にもこの場を借りて御礼申し上げる。

参考文献

[1] Richard E. Korf, Finding optimal solutions to Rubik ’s Cube using pattern databases, AAAI National Conference, pp700–705, 1997,

[2] Daniel Kunkle Gene Cooperman, Twenty-six moves suffice for Rubik’s cube, Proceedings of the 2007 international symposium on Symbolic and algebraic computation 2007, pp235–242, 2007

[3] David Joyner, Adventures in Group Theory, The Johns Hopkins University Press, 2002

[4] 前田篤彦 杉山公造 間瀬健二, 置換パズルのメディア変換とパズル・ジェネレータの試作, 情報処理学会研究報告. HI, ヒューマンインタフェース研究会報告 Vol.2002 No.111, pp33–40, 2002

[5] 杉山公造　前田篤彦　大澤亮　水元明法, 新しいパズルの創出：「抽象化とメディア変換」戦略の実践と考察, 日本創造学会論文誌 Vol.8,pp1–20, 2004

付 録A　実験データ一覧

A.1　A氏の$2 \times 2 \times 2$（1回目）攻略時の発話と操作

```
# 0:17 まずルービックキューブ全体を見て何色を揃えるかを決めます
# 0:25 そうですね
# 0:26 この場合はちょっと
# 0:28 特に意味はないですけど
# 0:30 黄色から
# 0:31 あ、黄色じゃない
# 0:32 赤をそろえていきます
# 0:36 なので
# 0:36 ここに黄色と赤二つあるので
# 0:40 これそのまま持ってくればとりあえず赤二面が揃います
rotate y0r 0:42
# 0:45 で白がここにあって
# 0:46 まぁ赤揃えるので白と赤の組み合わせ
# 0:48 ここにあるので
# 0:50 これもまぁひねれば簡単に
rotate x1r 0:51
# 0:52 で
# 0:52 最後のひとつの赤がここにあるので
# 0:58 これをまぁ一番近いところまでとりあえず持ってくる
rotate z1l 1:00
rotate z1l 1:00
# 1:03 で次なんですが
# 1:06 これは
# 1:07 えっとこの赤をこっちにこの
# 1:11 こう回転させればうまくいくわけですけど
# 1:15 そうするとこれがどいてしまうので
# 1:18 これをどかしておいて
rotate y0l 1:19
# 1:20 一度ひねる
```

```
rotate x1r 1:21
# 1:22 そしてまだ戻す
rotate y1l 1:23
# 1:23 という風にやってとりあえず一面をそろえます
# 1:26 で横もしっかり揃ってることを確認して
# 1:30 次上を揃えていきます
# 1:33 えーっと
# 1:34 まず一箇所組み合わ
# 1:37 色の組み合わせそろえるってことをします
# 1:43 ここに緑と青と
# 1:46 上はオレンジですかね
# 1:47 オレンジが来るようにします
rotate x1r 1:48
# 1:50 でここの色の組み合わせは揃いました
# 1:52 次に
# 1:53 他のところも同じように色の組み合わせをそろえるって事をやっていきます
# 2:00 ここに黄色としろとオレンジがありますが
# 2:02 黄色と白とオレンジの組み合わせは本当はここにこなくてはいけません
# 2:06 んで
# 2:06 オレ
# 2:07 このし
# 2:08 このオレンジと白と緑の組み合わせは
# 2:11 こっちにこなくちゃいけない
# 2:13 で
# 2:13 ここの組み合わせは
# 2:14 黄色オレンジ青であってる
# 2:17 この場合
# 2:18 えーっと
# 2:20 一回
# 2:21 こう
# 2:22 この手順で
rotate y0r 2:22
rotate x1r 2:23
# 2:23 操作してやると
rotate y0l 2:24
rotate z0r 2:25
rotate y0l 2:26
rotate z0l 2:27
```

```
rotate y0r 2:28
# 2:29 とりあえずこういう風に操作します
# 2:31 でまた今度
# 2:34 まぁ別に何色が基準でも委員ですが
# 2:36 たまたまここ揃ってるので
# 2:38 あ、揃ってないや
# 2:40 えっと
# 2:41 まぁ一箇所揃えます
rotate x1l 2:42
# 2:42 また同じようにオレンジと青と緑が揃っていて
# 2:45 で今度緑オレンジ白っていうのが
# 2:49 ここに欲しい
# 2:50 黄色青オレンジってのが
# 2:55 ん
# 2:57 ここに欲しい
# 2:59 で
# 3:02 黄色白オレンジっていうのが
# 3:06 どこかな
# 3:07 あ、これはあってる
# 3:10 あ、そうか
# 3:10 さ、ちょっと間違えて
# 3:12 さっきやら無くていい事をやっちゃいました
# 3:14 なので
rotate y0r 3:15
# 3:15 もう一回同じ手順を
rotate x1r 3:16
rotate y0l 3:16
# 3:17 どこでもいいのでやります
rotate z0r 3:17
rotate y0l 3:18
rotate z0l 3:19
rotate y0r 3:20
# 3:21 ではもう一度オレンジを
rotate x1l 3:24
# 3:25 またさっきと同じように青緑オレンジを基準にして見ましょう
# 3:28 黄色オレンジ青という組み合わせは
# 3:31 ここに欲しい
# 3:33 で
```

3:33 オレンジ黄色白っていうのは逆
3:36 この位置が入れ替わるとちょうどいい
3:38 この位置はオレンジ緑白で揃ってるので
3:42 えーっと
3:42 この場合は
rotate x1l 3:44
3:44 一個ずらしてやると目的の形になります
3:48 目的の形ってのは
3:50 とりあえず色の組み合わせ
3:51 黄色白
3:53 黄色
3:54 黄色と白とオレンジは揃ってる
3:56 んで
3:57 緑と青とオレンジは
4:00 えーっと
4:00 ここに来る
4:02 白と緑とオレンジは
4:03 ここにくる
4:04 黄色オレンジ青っていうのが
4:06 ここにくる
4:07 でちょうど
4:08 これがここ
4:09 これがここ
4:10 これがここって
4:12 このこれがこういう風に入れ替わると
4:14 えーっと
4:15 ちょうどあの
4:15 組み合わせだけで言うと
4:16 上の面
4:17 上のあのー
4:18 上にオレンジがきてる必要は無くて
4:20 組み合わせだけで言うと
4:21 えーっと
4:22 揃います
4:23 そのときは
rotate y1l 4:24
4:24 まぁさっきからやってるんですけど
rotate x1r 4:25

```
rotate y0l 4:25
rotate x1l 4:26
# 4:26 こうやって操作をしてやります
rotate y0l 4:27
rotate z0l 4:28
rotate y0r 4:29
# 4:32 そうすると
# 4:33 揃ってるはずなんですけど
rotate z0r 4:35
# 4:36 どっか適当に一回まわして見ましょう
# 4:38 白黄色
# 4:39 白オレンジ緑はここでいい
# 4:41 オレンジ黄色白はここでいい
# 4:44 青黄色オレンジってのはここでいい
# 4:47 緑オレンジ青ってのはここでいい
# 4:50 で次に
# 4:53 オレンジを一面だけ上に出して他の
# 4:55 他はオレンジがこないようにってする
# 4:58 まぁ一箇所基準を決めます
# 5:00 でここで
# 5:02 オレンジの位置を見ます
# 5:05 ここにある色とここにある色とここにある色が上に来る方法と
# 5:09 ここにある色とここにある色とここにある色が上に来る方法があります
# 5:13 今オレンジをひとつだけ上に出したいので
# 5:17 こうこうこうでやるとオレンジが二つきてしまう
# 5:19 こうこうこうでやれば
# 5:21 こことここととここだとオレンジはひとつだけ
# 5:24 ひとつだけ上に来るので
# 5:27 その方法をやります
rotate x0l 5:29
rotate y1l 5:30
rotate y1l 5:31
rotate x1l 5:32
rotate y1l 5:33
rotate x0l 5:34
rotate z1l 5:34
rotate x0r 5:35
# 5:35 どうでしょうか
```

5:36 オレンジが上だけきました
5:37 オレンジだけ上にきました
5:40 同じように
rotate y1r 5:40
rotate y1r 5:42
5:43 上の色をまた調べます
5:47 この方法を使ってやればさっき見た色の組み合わせは変わってないので
5:50 次の方法に行きたいと思います
5:53 でさっき言ったみたいに
5:55 こことこことここを上に持ってくる方法がある
5:58 でこの色の組み合わせは変わらないってことは
6:03 これみてみると
6:04 こことこことここでオレンジが全部上に来るので
6:08 その方法を試します
rotate z1r 6:08
rotate x1l 6:09
rotate x1l 6:10
rotate z0r 6:11
rotate x1l 6:12
rotate z1r 6:13
rotate y0r 6:14
rotate z1l 6:15
6:16 はい、できました
6:17 オレンジ上に着ました
6:19 で
6:20 これ２ｘ２ｘ２なので３と違って真ん中が無いので
6:23 もうそれだけで上が全部揃います
6:25 最後仕上げに
rotate x1l 6:25
6:25 こうやってまわしてやって
rotate x1l 6:25
6:26 ルービックキューブを完成させます
6:28 以上です

A.2 A氏の$2 \times 2 \times 2$（2回目）攻略時の発話と操作

```
# 0:05 まずは全体をこうやって眺めて何色が揃えやすいかを調べます
# 0:10 うーん
# 0:11 別に揃えやすい色っていうのは無いんで
# 0:18 うーん
# 0:19 まぁなんとなく
# 0:21 自分が好きな色から
# 0:23 まぁ僕青好きなんで青から揃えていきます
# 0:28 どこか一箇所基準になる青を決めます
# 0:31 ここにしようかな
# 0:32 この青は
# 0:34 えっとー
# 0:35 横が黄色とオレンジなんで
# 0:37 黄色と
# 0:38 黄色かオレンジがある青を探します
# 0:40 探すとここにあるんで
# 0:42 黄色の
# 0:43 黄色がある青がここにあるんで
# 0:45 これをまず
# 0:47 ここに持ってきます
# 0:49 そのためにはまず
# 0:52 こうやって一回まわして
rotate z0l 0:52
# 0:53 出来るだけ近くになるようにまわします
# 0:56 でこうやってまわしてやると
# 0:59 隣に移動すればいいんですけど
# 1:01 えっとこれ横に回せば
# 1:05 この色がこっちに来るんですけど
# 1:06 その時にこいつが
# 1:07 横に回したときにずれちゃうんで
# 1:09 それを防ぐために
# 1:11 このずれたら困る奴を一回よけておいてから
rotate y1l 1:12
# 1:15 回してやって
rotate x0r 1:15
# 1:16 で 元に戻すってやります
rotate y1r 1:18
# 1:19 そうすると
```

1:19 ここ青が二つでしかも黄色が横に並ぶってなります
1:25 こんど
1:26 この
1:28 別のもう一個の
1:30 残りの青を揃えていきます
1:33 で
1:34 まぁ目に付いた青がこれなんで
1:36 これを揃えていく
1:38 これを
1:39 えっとオレンジ緑ってのは
1:41 まぁオレンジが
1:42 オレンジがここにあるんで
1:45 とりあえずこうやって
rotate x1l 1:45
1:46 一番近いところまで持ってきます
1:49 今度はよける必要ないですね
1:51 さっきみたいに
1:53 だからこれはこのままこうやって回してやります
rotate z0l 1:54
1:55 で、最後の青がここにあるんで
1:59 とりあえず近いところにこうやって持ってくる
rotate x1r 1:59
2:02 で
2:02 これ最初の奴みたいに
2:04 このまま前に倒してやる
2:07 回してやると
2:10 えーっと
2:11 ここの青がどっかいっちゃうんで
2:12 それを防ぐために
2:14 よけておいて
rotate z1r 2:14
2:15 で回してやって
rotate y0l 2:16
2:18 よけた分を元に戻してやる
rotate z1l 2:19
2:20 こういう風にやるとこうやって青が揃います
2:22 で、まぁあの
2:24 やってる最中に

2:25 一面を揃える最中についでに横のこの部分もそろえてあるんで
2:30 これで二列揃ったことになります
2:33 次は上を揃えてくんですけど
2:36 そろえるときに何をやるかって言うと
2:38 色の組み合わせが揃っているかっていうのを確かめます
2:45 まぁここの場合なんかだと
2:48 黄色と赤と白っていう風になってて
2:52 ここは色の組み合わせは揃ってます
2:56 ここに来るべきキューブは上に白もきちゃってますけど
3:01 組み合わせは正しいってことになります
3:04 で、別のところもみてくと
3:05 ここの横は黄色とオレンジなんで
3:07 まぁ上はしろ
3:09 上は全部白が来るのが決まってるんで
3:11 ここに黄色と白とオレンジって組み合わせがきてれば正しいわけです
3:16 でこれは
3:17 黄色と白とオレンジがきてるんで正しいです
3:20 で、今度こっち
3:21 横がオレンジと緑で上白なんで
3:24 オレンジ白緑ってのが来なくちゃいけないんですけど
3:27 赤白緑ってのがきてるんで間違ってます
3:30 でこっちも赤緑白がこなくちゃいけないのに
3:35 違う色がきてるんで間違ってます
3:38 この場合はえーっと
3:39 一個ずらしてやって
rotate x1r 3:40
3:42 でそうすると
3:43 一箇所見たときに
3:45 白赤緑はここで正しい
3:48 でここは黄色白オレンジってのは
3:51 一個となり
3:55 で、今度その隣の部分にある
3:57 白赤黄色ってのが一個となり
4:02 でその隣
4:04 今度こっちにある白緑オレンジってのが
4:06 二個となり
4:08 要するに
4:09 一箇所だけ揃ってて

4:11 後は一個ずつ隣にずれるっていう風にしてやります
4:16 でこれで今目的の形になったんで
4:19 でこの時に
4:20 この組み合わせ三箇所をこういう風に
4:23 この三つを一個隣一個隣二個隣って入れ替えてやる方法をがあるんで
4:28 それをやります
rotate y1r 4:31
rotate x1l 4:32
rotate y0r 4:32
rotate x1r 4:33
rotate y1l 4:33
rotate x1l 4:34
rotate y0l 4:34
4:35 はい
4:35 これで出来ました
4:38 で、えーっと
rotate x1r 4:40
4:40 まぁ確認してみましょうか
4:42 まぁこの二個は調べるまでも無いですね
4:43 見たまんまなんで
4:44 組み合わせどころか上も揃ってます
4:47 でここは緑白赤ってのがきて正しい
4:53 ここは赤黄色白ってのがきてるから正しい
4:57 で今度はえーっと
4:59 一箇所だけ白が上にきて残り全部白以外っていう風にやっていきます
5:03 今二箇所ここに白がきてるんで
5:05 これはダメなんで
5:07 えっとー
5:10 まぁ回してやって
5:12 えっと
5:13 一箇所白が来るようにします
5:15 ここに白
5:16 まぁ白を一箇所にする方法なんですけど
5:18 一箇所基準というか動かない面ってのを固定してやります
5:22 まぁこのこれを
5:24 この黄色が見えている面を動かない面として扱ってやると
5:29 ここにある色とここにある色とここにある色を
5:34 上に持ってくる方法があるんでそれをやってやります

```
# 5:37 そうするとこの白と緑とオレンジが上にきて
# 5:41 この黄色はそのままなんで
# 5:44 白が一箇所だけ上ってなるはずです
rotate y0r 5:46
rotate x1r 5:46
rotate x1r 5:47
rotate y0l 5:48
rotate x1r 5:49
rotate y0r 5:50
rotate x1r 5:50
rotate y0l 5:50
# 5:51 はいできました
# 5:52 でこんどどうしてやるかっていうと
# 5:55 ここの白と
# 5:57 こことここことここを上に持ってくる方法があります
# 6:01 まぁそれをやってやると
# 6:03 今度は全部白が上にきます
rotate y1r 6:04
rotate x1l 6:04
rotate x1l 6:05
rotate y1l 6:06
rotate x1l 6:06
rotate y1r 6:07
rotate x1l 6:08
rotate y1l 6:08
# 6:09 はいできました
# 6:11 これで全部揃いました
```

A.3 A氏の$3 \times 3 \times 3$攻略時の発話と操作

```
# 0:10 えーと、まずは全体を見て何色をそろえようかなと考えます
# 0:15 まぁ何色でもいいんですけど
# 0:20 なんとなく白を揃えていきましょう
# 0:23 で
# 0:24 白を揃えるために真ん中を白をまず探して
# 0:27 そこの周りの色を調べて
# 0:30 えーっとどうすれば白が揃うか
# 0:34 まぁたまたま目に付いた白と赤って組み合わせを
# 0:37 他のところで探します
# 0:40 ひとつここに白と赤があるのでこの白と赤をこっちにもってきたいなぁ
# 0:47 なんで、えーっとどうしようかな
# 0:50 とりあえず、えーっとまぁど
# 0:53 この赤ちょっと離れてるんで
# 0:55 近いところに持ってこよう
rotate y0r 0:56
# 0:58 で、こうなったら
# 1:00 今度はここにあって
# 1:04 これをこっちに倒すと、し
# 1:07 えーっと
# 1:08 上の色が黄色になっちゃうんで
# 1:10 ここに赤をこさせたいんで
# 1:13 えーここにはこれないんで別の場所に移動させる
rotate y0r 1:16
# 1:16 要するにこっちの角に移動させる
# 1:18 こうやって
# 1:20 で、今度
# 1:22 んーっと
# 1:24 これをこっちに倒せばいいんだけど
# 1:27 こっちに倒すとえーズレちゃうん
# 1:29 ズレちゃうというかここのせっかく揃ってる赤がどっかいっちゃうんで
# 1:36 こうやってよけておいて
rotate z0l 1:37
# 1:38 で
# 1:40 さっきやりたかったほうに倒して
rotate y0l 1:40
# 1:43 戻してやる、と赤が揃う
rotate z0r 1:43
```

1:45 で今度こっちのもう一個の赤と白を探します
1:51 あった
1:51 この赤と白をここに持ってくるので
1:54 えーっと、遠いと考えにくいからまた近くにもってきておいて
rotate y0l 1:56
1:59 で、今度これをこっちに倒せばいいんだけど
2:04 たまたまここにある緑を
2:05 えーそのままよう
2:07 そのまま使いたいんで
2:10 さっきみたいにこれをよけてからまわしておく
2:13 よけてから
rotate x0l 2:14
2:14 回して
rotate y0l 2:15
2:15 戻す
rotate x0r 2:16
2:16 ってやるとこうやって揃うからこうしておく
2:20 で今度はここかここの色を探す
2:22 黄色と白ってやつか
2:26 それとも緑と白ってやつを探します
2:29 あ、あった緑と白
2:31 えーっと、これはまぁこのまますっと戻すだけでできます
rotate x2l 2:33
2:36 で、えーっと黄色と白
2:38 あった
2:39 黄色と白もあったんで近くまで持ってきて
rotate y0r 2:42
2:45 で、えーっと
2:48 この黄色と白もさっきと一緒で
2:51 えー倒す
2:52 よけてから
rotate z0r 2:53
2:54 戻しておいて
rotate y2l 2:56
2:57 よけてから移動させて戻す
rotate x0l 3:00
3:00 で、またよけた分を戻すとこういう風になって
3:04 いい感じにできます

3:06 で、後最後の白はここにあるんで
rotate y0r 3:08
rotate y0r 3:08
3:08 一番近くまで持ってきておいて
3:12 で、これも同じように
3:14 よけて
rotate z0l 3:14
rotate y2r 3:15
3:16 戻す
rotate y0r 3:17
3:17 戻すというか倒してやって
3:19 倒すじゃないな
3:20 まぁいいや
rotate x2l 3:22
3:22 で、よけた分を戻す
3:25 で、今度に
3:26 これで一段目が完成したの確認して
3:30 次は
3:31 まぁあの
3:32 真ん中の色は揃えるだけなのでこうやってしておいて
rotate y2r 3:34
3:38 えーっと
3:39 僕実はここにある色をこっちかこっちに持ってくる方法を知ってるので
3:42 それをやる
3:44 やれるようにします
3:46 えと
3:47 ここにある色っていうのは上に無い色の組み合わせ二つ
3:51 あ
3:51 ここにきて欲しい色っていうのは
3:54 上に無い色
3:55 この場合は青以外の二つの組み合わせなんで
3:59 えー
3:59 その青じゃない組み合わせ二つってのを探すと
4:02 ここにオレンジと緑
4:05 あ
4:06 んで黄色とオレンジってのがあるんで
4:09 まぁ先に黄色とオレンジを揃えたいんでこうやってひねっておきます
rotate y0r 4:10

```
# 4:13 でこうすると
# 4:14 ここが縦に揃って
# 4:16 縦に揃うってのも変だけど
# 4:19 で
# 4:19 オレンジをこっちに持ってきたいから
# 4:21 その時は
rotate y0l 4:23
# 4:23 こうやって動かしてやると
rotate z2r 4:24
rotate y0r 4:25
rotate z2l 4:26
rotate y0r 4:26
# 4:28 えーっと
# 4:28 オレンジが下に降りてきます
rotate x2l 4:29
rotate y0l 4:29
rotate x2r 4:30
# 4:30 ここ揃いました
# 4:31 でさっきと同じように
# 4:34 今度上の色
# 4:35 緑とオレンジ
# 4:35 あ
# 4:36 要するに青の組み合わせ
# 4:38 青が上に入ってない奴をここで探すと
# 4:40 緑とオレンジがあるので
# 4:41 それもこうやって
rotate y0l 4:41
# 4:42 まず取り合えず縦にそろえて
# 4:45 今度はこっちに持ってきたいんで
# 4:46 さっきとは逆の
rotate y0r 4:47
# 4:47 まぁ左右逆の操作を行ってやる
rotate z2l 4:48
rotate y0l 4:50
rotate z2r 4:51
# 4:52 と
rotate y0l 4:53
rotate x0l 4:54
```

```
rotate y0r 4:56
rotate x0r 4:57
# 4:58 こういう風にできます
# 5:00 で
# 5:00 次は
# 5:02 えーっと
# 5:04 黄色とオレンジがいい
# 5:06 黄色とオレンジも
# 5:07 あ
# 5:08 黄色と赤だ
# 5:09 黄色と赤も同じように
rotate y0r 5:10
# 5:10 こっちに
# 5:11 とりあえず縦にそろえてから
# 5:14 こっちにおろす方法を僕は知ってるんで
rotate y0r 5:16
# 5:16 こうやってやります
rotate z0r 5:16
rotate y0l 5:17
rotate z0l 5:18
rotate y0l 5:18
rotate x2r 5:20
rotate y0r 5:21
rotate x2l 5:22
# 5:23 で
# 5:24 次が赤と緑なんで
# 5:25 これもう目の前にあります
rotate y0r 5:29
# 5:29 これもまず縦にそろえて
rotate y0r 5:29
# 5:30 この状態からまずこっちにおろす方法を使うと
rotate x0r 5:34
rotate y0l 5:35
rotate x0l 5:36
# 5:39 あれ
# 5:40 ああいいのか
rotate y0r 5:41
rotate x0r 5:42
```

```
rotate x0l 5:42
rotate y0l 5:43
# 5:43 あ
# 5:44 間違えた
# 5:45 えーっと
# 5:46 どうやるんだっけ
rotate y0l 5:46
rotate z0l 5:47
# 5:47 こうかな
rotate y0r 5:49
rotate z0r 5:50
# 5:51 よし
# 5:51 できました
# 5:53 で次は
# 5:55 えーっと
# 5:56 まず
# 5:57 上の十字
# 5:58 上を十字型に同じ色にそろえます
# 6:02 えーっと
# 6:02 今この
# 6:03 十字になってないんで
# 6:04 それを直してやりたいと思います
# 6:06 で
# 6:06 この場合は
rotate z2r 6:06
# 6:06 こうやってやってやると
rotate x0r 6:07
rotate z2l 6:08
rotate x0l 6:09
rotate y0l 6:10
rotate x0l 6:11
rotate y0r 6:11
# 6:12 できます
rotate x0r 6:12
# 6:13 で
# 6:13 ここで青青青青青青っていう十字型に揃いました
# 6:17 で
# 6:18 次は四つのブロックをそろ
```

6:20 角四つのブロックをそろえるんですけど
6:22 その時にまず一個基準を決めます
6:25 基準
6:26 まぁ目の間にあるんでこの色を基準にして
6:30 えーっと
6:30 オレンジオレンジ青ってこの角
6:32 色の組み合わせだけを揃ってるか確認を
6:35 揃ってるんで他を見ます
6:37 えーっと
6:39 他
6:39 青緑赤
6:40 青緑赤の組み合わせはここに欲しい
6:43 あ
6:43 青緑赤の組み合わせは
6:45 ここはオレンジ黄色青の組み合わせなんで
6:48 えーっと
6:49 上に青がある必要は無くてその組み合わせだけを調べてます
6:50 今は
6:52 えーっと
6:53 青緑赤はこっち
6:55 なんでここは違いますね
6:56 んで
6:57 黄色赤青ってのは大丈夫
6:59 黄色青緑っていう
7:00 ん
7:01 黄色オレンジ青っていうのはここで大丈夫なんで.
7:05 あ
7:05 ここじゃないや
7:08 黄色青オレンジ
7:10 黄色青オレンジ
7:11 あ
7:11 ここか
7:13 ここなんで大丈夫なんで
7:15 こことここがいれ
7:17 組み合わせだけで見ると入れ替わってほしい
7:18 えっとこうやって対角線上に
7:20 えーっと
7:21 組み合わせが入れ替わってる場合は

```
rotate x2l 7:23
# 7:23 一回取り合えず
rotate y0l 7:23
# 7:25 こういう操作を行います
rotate x0l 7:26
rotate y0r 7:29
rotate x2r 7:30
rotate y0l 7:32
rotate x0r 7:33
# 7:35 で
# 7:35 一回やりました
# 7:36 でまた同じようにどこか一個基準を決めます
# 7:40 まぁこの場合緑赤青ってのが揃ってるんで
# 7:43 ここを基準にします
# 7:45 でここの組み合わせ見ると
# 7:49 赤黄青
# 7:50 赤黄青はここにほしい
# 7:52 で
# 7:52 黄色オレンジ青っていうのは
# 7:54 ここに欲しい
# 7:55 で
# 7:56 青
# 7:57 青緑オレンジってのは
# 7:59 組み合わせがここなんで
# 8:01 これがこの基準とした一点を除いて
# 8:03 時計回りに入れ替わると
# 8:07 えーっと
# 8:08 そろ
# 8:08 揃います
# 8:11 で時計回りに入れ替えるって方法も決まってて
# 8:15 そのときはこうやって
rotate x2r 8:16
rotate y0r 8:18
rotate x0r 8:19
# 8:19 やってやると
rotate y0l 8:19
rotate x2l 8:20
rotate y0r 8:21
```

rotate x0l 8:22
8:23 入れ替わります
8:24 えーっと今入れ替えたはずなんで
rotate y0l 8:26
8:25 もう一回確認してみます
8:27 ここはオレンジ赤青で正しい
8:30 ここも黄色赤青で正しい
8:33 で
8:33 黄色青オレ
8:34 黄色青オレンジはここも正しい
8:38 でこの場合
8:39 この場合上に出て無くても黄色と青とオレンジがあることが大事
8:43 あれば大丈夫です
8:45 これも緑青オレンジなんで
8:47 ふちにきてるんでちゃんと大丈夫です
8:49 でこうなると
8:51 今度はき
8:53 この四箇所の
8:55 を全部上の
8:56 上に青をそろえます
8:58 えー
8:59 その時どうするかっていうと
9:01 まず
9:03 実は僕あのー
9:04 三つの方法を知ってて
9:06 ここにある色
9:08 まぁひとつどっか基準を決めたとき
9:11 ここにある色とここにある色とここにある色を上に出すって方法を知ってます
9:18 で
9:18 なんでそ
9:19 つまり
9:19 こことこことここの三色に青がくればいいんですけど
9:24 むし
9:24 むしろ
9:25 というよりもあのー
9:26 次の目標の形としては
9:29 一箇所だけこういう風に青が出ていて
9:31 こことこことここが青以外の色が出てるっていう形を次の目標にしています

9:36 なんで
9:37 えーっと
9:39 一箇所青を出すためにはここに
9:41 ここはまぁそのまま色に
9:43 色の変化が無い部分として使うと
9:45 ここの色が上に来る
9:47 ここの色が上に来る
9:48 ここの色が上に来るって方法が僕出来るんで
9:52 その方法をやってやると
9:54 ここだけ青になって他全部青以外ってなるはずなんで
9:58 それをやります
rotate x0r 10:00
rotate y0l 10:01
rotate y0l 10:01
rotate x0l 10:04
rotate y0l 10:05
rotate x0r 10:06
rotate y0l 10:08
rotate x0l 10:09
10:09 はい
10:09 青が上一箇所だけきました
10:12 このふちだけ青になって
10:14 ここの十字もそのままで
10:17 えーっと
10:17 あー今ちょっと位置がずれてるんでずらして
rotate y0r 10:20
rotate y0r 10:21
10:20 確認しましょうか
10:24 えーっと
10:24 黄色赤青っていうのが
10:26 組み合わせがあってるけど青がきてない
10:28 ここも緑赤青で組み合わせがあってるけど青が上にきてない
10:32 ここも同じです
10:34 でこうなると今度
10:35 さっきはこことここことここを出す方法なんですけど
10:40 逆の手順をするとこことここことここを出す方法があるんで
10:44 こことここことここを出せば全部上が青になるんで
10:46 それをやります

```
rotate x2r 10:47
rotate y0r 10:48
rotate y0r 10:48
rotate x2l 10:49
rotate y0r 10:49
rotate x2r 10:50
rotate y0r 10:51
rotate x2l 10:51
# 10:51 これはさっきの逆の手順
# 10:53 左右とか入れ替えて
# 10:55 でこれで上が青きました
# 10:57 で
# 10:58 せっかく角そろえたんで角が揃うようにしておきます
rotate y0r 10:59
rotate y0r 10:59
# 11:02 でこうなると最後
# 11:03 こことここととここの入れ替えなんですけど
# 11:06 赤と青はここにきて欲しい
# 11:08 オレンジと青はここにきて欲しい
# 11:10 緑と青はここにきて欲しいってなって
# 11:12 黄色はそのままで揃ってます
# 11:14 まぁ揃ってる基準に
# 11:15 揃ってる面を考えずに
# 11:18 えーっと
# 11:20 緑は
# 11:21 あ
# 11:21 これはここ
# 11:22 これはここ
# 11:22 これはここ
# 11:23 また今度
# 11:23 今度も時計回りに
# 11:25 入れ替えると
# 11:27 このブロックとこのブロックとこのブロックを
# 11:30 時計回りに入れ替えてやると.
# 11:33 えーあの
# 11:34 うまくいきます
# 11:35 うまくいくって言うか
# 11:37 まぁこのエッジの部分を入れ替えてやると
```

11:39 うまくいきます
11:40 なんで
11:41 その方法を
11:42 まぁこれも覚えてるんでできるから
11:45 やります
rotate x0r 11:46
rotate x0r 11:46
11:47 まずこうしておいて
rotate y0r 11:48
11:48 こうして
rotate z0r 11:49
rotate z2r 11:50
11:5０ あげる
rotate x0l 11:51
rotate x0l 11:52
11:52 戻して下げて
rotate z0l 11:53
rotate z2l 11:53
11:55 でこっからはえーっと
11:58 こっからさきは覚えてないんですけど
12:00 まぁ見りゃわかる程度なんで
rotate y0r 12:02
12:03 青はこっちに回してやって
12:04 で
12:05 えーっと
12:07 これを
rotate x0l 12:07
12:08 二回ひねってやると
rotate x0l 12:08
12:09 完成します

A.4 B 氏の $3 \times 3 \times 3$ 攻略時の発話と操作

```
# 0:10 じゃあまず緑色の面を探して上にきてるから
# 0:15 それでまずバツを作ります
rotate y2r 0:16
rotate z2r 0:17
# 0:17 えっと
rotate z2l 0:18
rotate z2l 0:18
# 0:19 もうほとんどバツ出来上がってるから
rotate y0l 0:20
rotate x2r 0:21
rotate y0r 0:21
rotate x2l 0:22
rotate x2l 0:22
# 0:22 まずこれでバツができたと
rotate z0r 0:23
rotate z0r 0:24
# 0:24 そして後は緑色の面の角を揃えていく
rotate x2l 0:29
rotate z0r 0:30
# 0:30 で結構なんか
rotate x2r 0:30
rotate z0l 0:31
# 0:31 あんま崩れてないから楽かな
# 0:34 ということで
rotate x2l 0:35
# 0:35 んで
# 0:36 えーっと
rotate z0l 0:37
# 0:37 角を合わせていきます
rotate x2l 0:38
rotate z0r 0:39
rotate x2l 0:39
# 0:40 んで
# 0:41 えーっと
rotate z2l 0:42
# 0:42 角を合わせれましたと
rotate x2r 0:43
```

```
# 0:44 そして
rotate z2r 0:46
# 0:46 これ完全一面の状態になってるんで
# 0:47 次に二段目を揃えていきます
rotate x2l 0:49
# 0:50 二段目は一回こうやって退避させて
rotate x2r 0:50
rotate y0l 0:50
rotate x2l 0:51
rotate y0r 0:52
rotate x2l 0:54
# 0:53 んで
# 0:54 入れたい奴を揃えてまぁ角をまた戻す
rotate z2l 0:55
rotate x2r 0:56
rotate z2r 0:57
# 1:00  そして
rotate x2l 1:01
# 1:01 えーっとまた
rotate z0r 1:01
rotate x2r 1:02
# 1:03 後はもう同じ作業を何回か繰り返すだけで
rotate z0l 1:03
rotate x2r 1:04
rotate y2r 1:05
rotate x2l 1:06
rotate y2l 1:08
# 1:08 でなっていきます
rotate x2r 1:08
rotate z0l 1:11
rotate x2l 1:12
rotate z0r 1:13
rotate x2l 1:13
# 1:13 えっと後たぶん一回で
rotate y0r 1:14
rotate x2r 1:16
rotate y0l 1:17
# 1:17 完全二面っていうのかな
```

```
# 1:18 あーもうできちゃった
# 1:19 んで
rotate y2r 1:19
rotate x2r 1:20
# 1:20  そん次に上でプラスを作ります
rotate z2r 1:21
rotate x2l 1:22
rotate z2l 1:23
rotate y2l 1:23
# 1:24 でプラスを作るのは
rotate y0l 1:25
rotate x2r 1:26
rotate x2r 1:26
rotate z0l 1:27
rotate z0r 1:27
# 1:27 こうやってまわす
rotate x2l 1:27
rotate z0l 1:28
rotate x2l 1:30
# 1:30 なんていえばいいんだろこれ
rotate z0r 1:31
# 1:31 まぁこうする
rotate y0r 1:32
# 1:33 そしたらプラスが出来ます
# 1:34 そん次
rotate z0l 1:35
rotate x2r 1:36
# 1:36 一回まぁ上の面を
rotate z0r 1:38
rotate x2l 1:38
# 1:38 揃えますと
rotate z2l 1:39
# 1:40 かた、硬い
rotate x2r 1:42
# 1:43 で上の面揃えるのはウェブページ見たんで
rotate z0l 1:43
rotate x2l 1:44
rotate z2r 1:46
```

```
# 1:46 だいたいそのパターンにあったようにしかならないんで
rotate z0r 1:47
# 1:50 で後は角の残りかたをみて
rotate z2l 1:50
rotate z0l 1:51
rotate x2l 1:52
rotate z2r 1:53
# 1:53 後は目をつぶっても
rotate x2r 1:54
rotate z0r 1:55
# 1:55 こっからさきは
rotate x2l 1:56
rotate z2l 1:57
rotate x2r 1:57
rotate z2r 1:58
# 1:58 まわす一方なんで
rotate x2r 2:00
# 2:00 硬
rotate x2r 2:01
rotate z0l 2:02
rotate x2r 2:03
# 2:04 硬いなこれ
rotate z0r 2:04
rotate x2l 2:05
rotate z2l 2:06
rotate x2r 2:07
# 2:07 んでこういう風にすると
rotate z0l 2:07
rotate x2l 2:08
rotate z2r 2:09
rotate z0r 2:10
# 2:11 まぁなんか
# 2:21 穴抜け状態になってる
# 2:23 これだとこうまわせばいいのかな
# 2:18 なんで
rotate y0r 2:19
rotate x2r 2:20
rotate y0r 2:21
```

```
# 2:19 まぁ
rotate x2l 2:22
rotate y0r 2:23
rotate x2l 2:24
rotate y0r 2:25
# 2:25 こうやってやると
rotate x2r 2:26
rotate y0l 2:27
rotate x2r 2:28
# 2:28 あれ
# 2:28 間違えちゃった
rotate y0l 2:29
rotate y0l 2:29
# 2:29 勢いでやらないと間違える
rotate y0r 2:31
rotate x2r 2:31
rotate y0r 2:33
rotate x2l 2:34
rotate y0r 2:34
rotate x2l 2:35
rotate y0r 2:36
rotate x2r 2:37
rotate y0l 2:38
# 2:38 こうやると
rotate x2r 2:38
rotate y0l 2:39
# 2:39 出来上がります
rotate y0l 2:39
# 2:40 一応出来ました
```

付 録B　ルービックキューブに類似したパズル

本研究ではルービックキューブのみに焦点を当てて研究を進めた。ルービックキューブは置換パズルの一種であるが、置換パズルにはルービックキューブ以外に様々なものがある。そこで、本節では本研究の被験者が他の置換パズルを攻略する際どのような反応を示すかを簡単に述べる。

B.1　ボイドキューブ

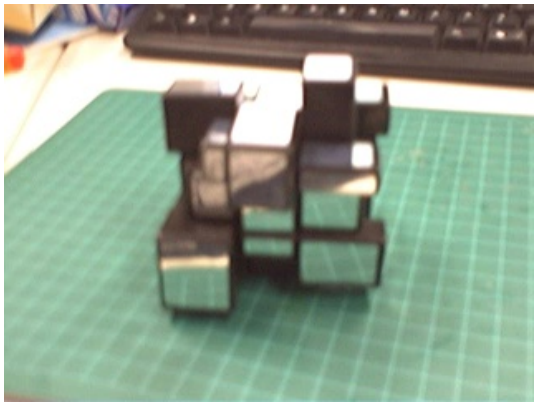
ボイドキューブは$3 \times 3 \times 3$のルービックキューブと似ているが、中央のキューブが存在せず、空洞になっているものである（図B.1(a)）。中央のキューブが存在しないだけで、パズルとしての基本的な構造と可能な操作は$3 \times 3 \times 3$のルービックキューブと同一である。そのため、$3 \times 3 \times 3$のルービックキューブが攻略可能ならば、ボイドキューブも解くことができる。しかし、実際にやってみると$3 \times 3 \times 3$のルービックキューブを解く事が出来る人でも確実に解く事が出来るとは言え無かった。、中央が存在しないゆえの特殊な色の配置になることがあり、その場合にはランダムに何度か操作し、もう一度最初から色を揃え直さなければならなかった。

B.2　ミラーブロックス

ミラーブロックスは、$3 \times 3 \times 3$のルービックキューブと同じ構造を持つ。しかし、ルービックキューブが色を合わせるのに対して、ミラーブロックスは形を合わせる。図B.1(b)は揃った状態のミラーブロックスである。これを何度か操作すると図B.1(c)のようになる。このように、色はすべて同じであるが、それぞれのキューブの形が違うためにパズルとして成り立っている。これは、色が形に変わっただけで$3 \times 3 \times 3$のルービックキューブとまったく同じ構造を持つためルービックキューブが解ける人ならば簡単に攻略が可能であると考えた。しかし、実際に操作をしてみるとA氏は色が形に変わっただけなので簡単だといいながら解いたが、B氏はミラーブロックスを完成させることが出来なかった。

(a) ボイドキューブ

(b) ミラーブロックス（操作前）



(c) ミラーブロックス（操作後）

(d) メガミンクス

図 B.1: ルービックキューブ以外の置換パズル

B.3 メガミンクス

メガミンクスはルービックキューブが正六面体であるのに対し、正十二面体の構造を持つ（図 B.1(d)）。置換パズルであり、一つの角に集まる辺が 3 つであるため、$3 \times 3 \times 3$ のルービックキューブと似ている部分もあるが、面一つ一つが五角形になっているため、通常のルービックキューブが解ける人にとっても攻略は難しい。事実、A 氏・B 氏ともに完成させることは出来なかった。