Roux, an advanced approach to cubing

Dominic Zimmer

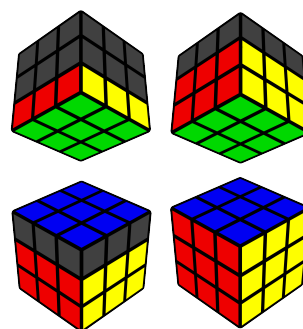January 6, 2016

Contents

1 Prologue

1.1 Abstract

I'm going to introduce, explain, and discuss a method to solve the Rubiks Cube called *Roux*. I will not assume the reader knows to how to solve the Rubiks Cube. However the geometric understanding of the cube is useful. Thus I will also mention some very basic information which might seem redundant to others. After I ensured we are on the same level of understanding the notation, I will first give a rough overview of what Roux is. Then I will continue by going into great detail on how to approach each step.

1.2 Perspectives

Apart from Roux, there are plenty methods out there of which I will mention the most prevalent ones.

First off, I am going to cover the *Beginner's Method*, also known as *Layer by Layer Method*. The Method does what the name implies: it solves the cube layer by layer. This method seems pretty intuitive to most people as it starts off by solving one face (which is easy to begin with).

The most common method is closely related to the Beginner's Method: *Fridrich's Method* or more commonly known as *CFOP*. CFOP begins by forming a **c**ross on one side, extending that cross to the **f**irst two layers, **o**rienting, and **p**ermutating the last layer (leading to its unique name). You can see how closely related it is to the Beginner's Method: The key difference is that the Beginner's

The four steps of using the Beginners Method

Method splits building the first two layers into two seperate steps whereas CFOP does this more efficiently. There are some more methods which deserve to be named here, for instance *Petrus' Method*, but I'm going to leave it at that.

1.3 Notation and terminology

1.3.1 Face turns

To be able to communicate on this abstract level of thinking, we are going to use the standart notation and terminology which I will explain below.

In our notation we are not going to consider the colors of the individual facelets but rather keep the cube in one orientation. Thus we can easily refer to the six faces of the puzzle as **u**p, **f**ront, **r**ight, **l**eft, **b**ack and **d**own and abbreviate them each by their initial letter. Keeping that in mind, we intuitively define the moves U, F, R, L, B and D as clockwise 90° turns of their respective face.

To denote the counterclockwise turn of a side, we add an apostrophe to the respective turn. If we wanted to perform a 180° rotation of a face, we would simply append the number two to the move.

1.3.2 Observations

For clarification: we consider a respective side of the cube to face us as we turn it. For instance U and D rotate opposite sides and turn in opposite directions. We certainly could also use $U3$ with the same logic as we defined $U2$ yet we would notice that $U3$ is the very same as U'. We also see that $R2$ and $R2$' result in the same turns. So would someone want to use $R2$' at all? Yes - sometimes it can be useful to hint using two R' moves over two R moves for more efficient turning. Note how we are using uppercase letters for the basic turns. Lowercase letters are reserved for an upcoming notation.
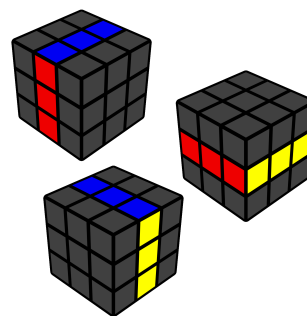
1.3.3 Slice turns, wide turns and cube rotations

Sometimes the basic turns are inconvenient to use in a certain case, for instance if you wanted to turn the middle slice, i.e., the slice *sandwiched* between the right and the left side.

For that reason we define the **m**iddle slice turn, the **e**quatorial slice turn and the **s**tanding slice turn. We again abbreviate them respectively with their initials. However we need to declare that M turns in the same direction as L, E turns in the same direction as D, and S turns in the same direction as F. Intuitively, we can append the earlier defined suffixes ' and 2 with their corresponding interpretations.

For a similar motivation as why we introduced the slice turns, we also define *wide* moves. Wide moves are pretty self-explanatory: you grab two layers by expanding a face turn by the adjacent slice. We respectively add w for *wide* to our collection

From top to bottom: the middle, equatorial and standing slice

of suffixes. As we add more suffixes, we want to introduce an abbreviated form for wide moves: replacing the uppercase letter by a lowercase one.

The last notation we want to introduce are *cube rotations*. Cube rotations require you to turn the entire cube inside your hands. We denote them as the three axes X, Y and Z. Again we need to clarify which axis turns in which direction: X turns in the same direction as R, Y turns in the same direction as U and Z turns in the same direction as F.

Since you might run into this question: we do not differentiate between uppercase and lowercase cube rotations.
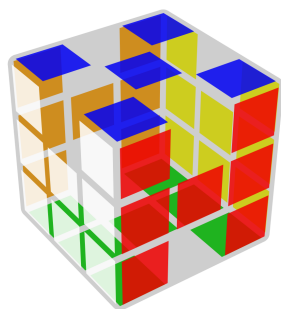
2 Roux

2.1 Brief overview

Before we go into detail, I am going to give you a rough idea of how roux works.

You start off by what is called the *First Block*: you build a 1x2x3 block of one color around the matching center. Next, we will mirror the first block onto the opposite side. This leaves us in the state of the *First Two Blocks* already solved. If we were looking from the front side, we would see that we are left with a T-shaped area which is yet to be solved. Notice how every piece that is yet to be solved is on either the middle slice or the upper layer. Mathematicians like to call these pieces the *(linear) span*, denoted as $\langle M, U \rangle$, of the turns U and M.
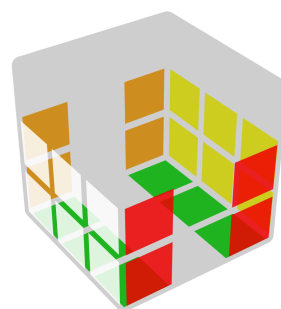
We now want to solve the corners of the upper layer to be left with six unsolved edges and four unsolved centers. Solving the corners of the upper layer will be split into two

The First Two Blocks

seperate steps. First, we orient the corners to have their common color face up. Then we permute them to the places where they belong. Both of these steps are not very trivial and thus will require some algorithms. If you want to figure some algorithms out by yourself, you usually want to try to break up the First Two Blocks and then reconstruct them slightly differently. But there is no big learning effect involved, so I would not recommend doing so.

As mentioned previously, we are now left with six edges and four centers unsolved. As you realize quickly, the centers are at most two M-slice moves from being in their correct positions, leaving us with the *Last Six Edges* unsolved. You can see in the figure beside how the cube would look so far. What speeds up the process of solving the Last Six Edges is the fact that you can now solve the cube by only using algorithms of $\langle M, U \rangle$, i.e., you do not have to scramble what you have solved so far.

Now we would like to split solving the Last Six Edges into three substeps: orienting the edges, solving the left and right upper edge, and finishing the middle slice.

Before the Last Six Edges

By orienting the edges we mean that we want them to only show bottom or top layer stickers on the bottom and top layer. In our example, this would make the U-face only contain blue and green stickers. This can either be

achieved intuitively with a little bit of practise or by folloing the patterns and steps mentioned in the according section below. Now we intuitively insert the last remaining edges on the left- and right-hand side. All that is left to do is resolving one out of four cases to finish off the middle layer.

Now we will precisely go into detail of each step.

2.2 First block

Our goal is to build a 1x2x3 block on our first color side. Notice that we require edges to be matching to consider the block solved. This can be achieved in various ways. I will introduce some basic concepts and leave it up to you make those more efficient.

You can start off by building a 1x1x2 block out of a center and edge piece. By adding more 1x1x2 blocks, you can first expand that block to a 1x2x2, then to a 1x2x3 block.
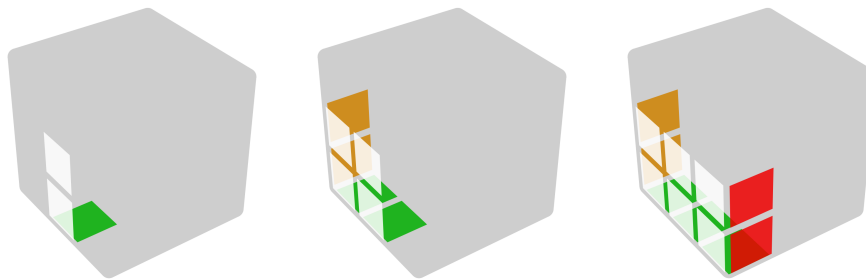
Building the first block with three 1x1x2 blocks

Alternatively, we could also start by putting the edges in place and inserting the corners afterwards. As shown in the picture, we want to keep the 1x2x3 block on the left bottom edge.

Alternatively, you could bring the edges in place first and insert the corners afterwards.

2.3 Second block

2.4 Corners Last Layer

2.5 Last six edges

3 Appendix: Algorithms

3.1 2 Look CLL

3.2 CMLL

3.3 LSE