Solving Rubik's Cubes

V. Nardozza*

Jan 2022†

Abstract

In these notes I present all beginner methods I have used to solve the cubes I own. These notes are for myself, as a reference for the algorithm I have used and to have these algorithms at hand when I pick up a cube I have not solved for a long time. However, these notes are available for anybody interested in it[1].

Contents

*Electronic Engineer (MSc). Turin, IT. Any comment to: vinardo@nardozza.eu
†Posted at: http://vixra.org/abs/2201.0038 - Current version: v3 - Aug. 2022
[1]A more recent version may be found at: https://www.nardozza.eu/vinardo

Introduction

In these notes I present all beginner methods for solving the most common Rubik Cubes. Following the terminology used by cube solver, I will call sequences of moves "algorithms" although they are not technically so.

Every time I mention a symmetric algorithm I mean basically two things. For algorithms permuting three pieces, the symmetric algorithm is the one that permutes the pieces in the other way round (e.g. clockwise, anticlockwise). For other algorithm, I mean the algorithm that changes the cube configuration in a symmetric way with respect to the plane dividing your body in two symmetric parts. Often the two definitions coincide.

Symmetric algorithms will not be given and they need to be worked out by the reader. This is always an easy task. However, symmetric algorithms, although useful, are usually not essential for solving the cubes.

Rubik Puzzles come in families. Member of one family can be either mechanically equivalent (although sometimes it is difficult to tell) or one is a mechanical subset of the other. Generally speaking, member of the same family can be solved using a single method. Sometimes a puzzle that is a mechanical superset of another, it has simply more "layers" of is "smaller" sibling. This is the case of the Rubik's revenge, with respect of the Classic Rubik's Cube or the Gigamix, with respect of the Megamix. Usually, cubes of this type can be solved by reduction to one of the smaller member of the family.

For this type of cubes, and with abuse of language, we will call a specific one the Size N of the family, like you would do with a pair of trousers, where N is the number of "layers" and when the words size implies (olthough not always true) that the more are the "layers", the bigger is the cube!!

1 Classic Rubik's Cube

Figure 1: Classic Rubik's Cube

Notation. *A classic Rubik's Cube has three type of pieces. Centres, Edges and Corners. I will use the standard notation for algorithms where faces are denoted by:*

F: front, R: right, U: up, L: left, D: down, B: back

- Moves are intended to be a clockwise rotation. For an anti-clockwise rotation, the letter of the relevant move will be followed by the prime (i.e. ') symbol (ex. D').

- A number after a letter of a move shall indicate the number of 90deg rotations to be applied when different from 1 (ex. D2 is a 180 deg rotation of the down layer).

- When two slices have to be rotated (the external slice and the one next to it i.e. the middle slice in the case of a 3x3x3 cube), the letter relevant to the move shall have a w as a lower index (ex. D_w).

Classic Rubik's Cube can be solved by layers.

Layer 1:

Starting always from the layer relatives to a specific colors helps a lot to visualise the moves. It is custom in western countries to start always from the white one.

- Complete the white cross placing the edges so that they match the white and the other color centre piece (intuitive moves, no algorithm).

- Place the corners moving them from the 3rd layer to the 1st layer with the algorithm classic(1) and its symmetrical. It is always possible to take a 1st layer corner on layer 3 and with the right orientation (white on one side) by using algorithm classic(1) in various ways. Algorithm classic(1) is called **Sledgehammer** and is one of the most used algorithm is various situations and for all cubes.

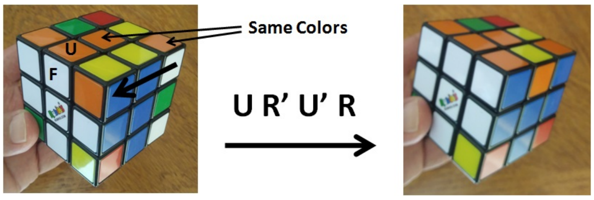


Figure 2: classic(1): $U\ R'\ U'\ R$

Layer 2:

- Place edges on the second layer. To do that match one 2nd layer edges piece (the ones without yellow on them) present on the 3rd layer with the relevant corner piece of the 1st layer with the algorithm classic(2) and its symmetrical. Algorithm classic(2) is also a **Sledgehammer** and therefore up to symmetries algorithms classic(1) and classic(2) are the same one.

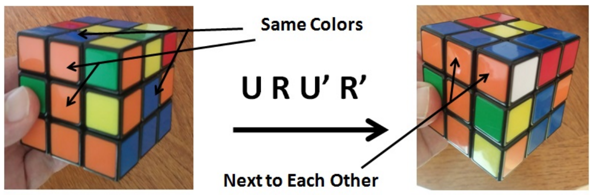


Figure 3: classic(2): $U\ R\ U'\ R'$

- now the 1st layer corner will not be in place any more. Put it in place by using algorithm classic(1). This will eventually put in place also the 2nd layer edges as desired.

It is always possible to remove a 2st layer edge which is misplaced by replacing it with any edge on layer 3 by using algorithms classic(1) and classic(2) as described above.

Layer 3:

- Make the yellow cross on the 3rd layer. This can be done using the algorithm classic(3). If only one yellow piece is present on the 3rd layer then apply algorithm classic(3) with any front face. This will make an L pattern of yellow pieces on the 3rd layer.

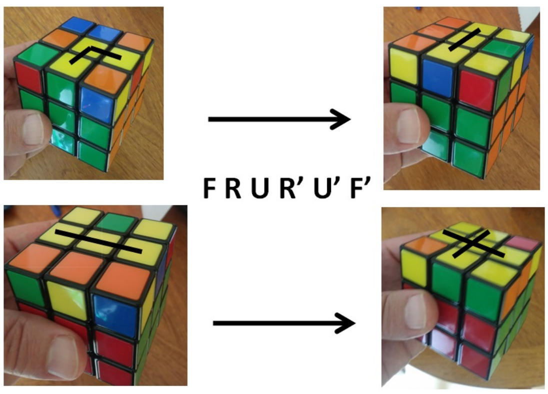


Figure 4: classic(3): $F\ R\ U\ R'\ U'\ F'$

There are two option to complete the 3rd layer.

Layer 3 Option 1:

- Place Layer 3 corners in their correct position with respect to colors of the lateral faces by using algorithm classic(4) and its symmetrical. This algorithm, permutes 3 corner of 3rd layer, and swaps two edges (but we do not care).

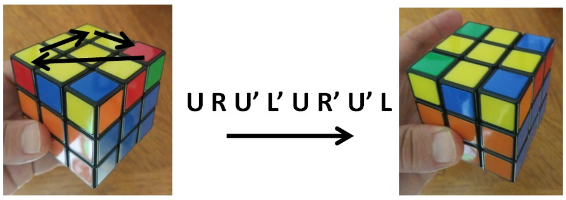


Figure 5: classic(4): $U\,R\,U'\,L'\,U\,R'\,U'\,L$

- Match the color of the edges of Layer 3 with the color of the lateral faces without moving corners from their own position by using algorithm classic(5) and its symmetrical. This algorithm, permutes 3 edges of 3rd layer, rotates corners around their axis (but we do not care) and it is known as "Sune". Note that the Sune can be performed also inverting the sense of rotation of the move before the last as $R\,U\,R'\,U\,R\,U2'\,R'$ and this is the way it is normally presented in tutorials because it is better to be performed in speed cubing. However, I prefer the other way because I do not do speed cubing and I like to turn the Up layer always in the same sense.

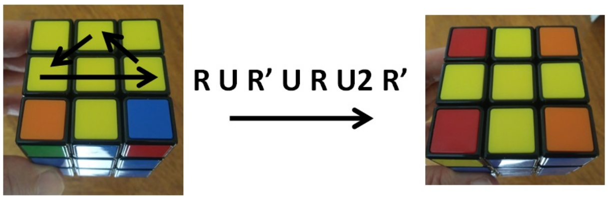


Figure 6: classic(5): $R\,U\,R'\,U\,R\,U2\,R'$ (Sune)

- Now all pieces are in place but corner of 3rd layer are rotated around their own axis. To rotate them in the right way we can use algorithm classic(6) which is twice algorithm classic(1) in disguise. Algorithm classic(6) rotate one corner of the 3rd layer without changing the other pieces of the layer. Moreover if applied 3 times it makes the cube to go to the original configuration. To rotate a corner, apply algorithm classic(6) with the corner you want to rotate placed in the up-right-front position of the cube till the corner is rotated in the correct way. Rotate the up face in order to put the next corner to rotate in the up-right-front position. Keep applying algorithm classic(6) for a total number of time which is a multiple of 3 till the cube is solved.

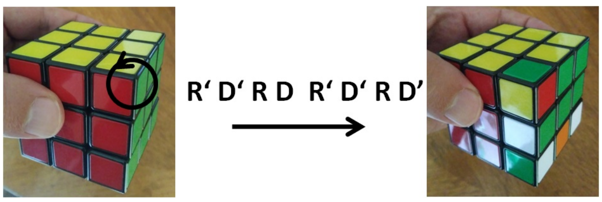


Figure 7: classic(6): $R'\,D'\,R\,D\;\;R'\,D'\,R\,D$

Layer 3 Option 2:

- Place corners in the correct position even if not rotated in the correct way by permuting them using algorithm classic(4) and its symmetric.

- Orient Last Layer (OLL), i.e. make the up face of the cube completely yellow by using algorithm classic(5) and its symmetric (Sune). It should be possible to do it using algorithm classic(5) and/or its symmetric at most twice.

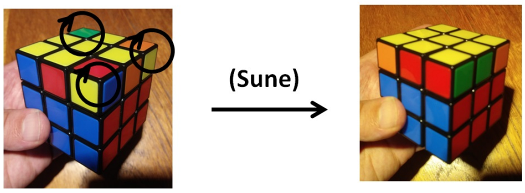


Figure 8: Orient Last Layer (OLL)

- Turn the up layer till the 4 corners go in the correct position. The edges will generally not be in the correct position.

- Permute the edges of layer 3 using algorithm classic(7). This is called a U-permutation (U-perm).

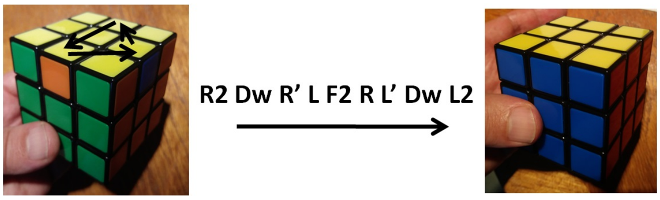


Figure 9: classic(7): $R2\ D_w\ R'\ L\ F2\ R\ L'\ D_w\ L2$

Note that a U-perm can be also performed by applying a Sune (algorithm classic(5)) and its symmetric in sequence. I let the reader to find out how to do it by his own.

2 Rubik's Revenge Et Al

This type of Rubik's cubes (apart from the 2x2x2 one) can be solved by reducing them to a classic 3x3x3 Rubik Cube. For these cubes, some parity cases arise that have to be solved with specific algorithms.

These days, commercial Rubik's cubes exist till the size 17x17x17 or even more. After the 12x12x12 they usually exist in odd sizes only, which are mechanically more stable.

New algorithms are required to solve cubes up to the 5x5x5. After that all cubes of these type can be solved without additional algorithms.

Notation. *We need some additional notation.*

- A lower index on a move, if different from 1, will indicate that an internal layer shall be rotated. If equal to 1 the index is normally omitted. For example, in a 4x4x4 cube $R_1 = R$ *and* $R_4 = L'$*. Moreover, as additional examples,* R_2 *and* R_3 *are rotation of internal slices,* $R\,R'_w = R'_2$ *and* $R2\,R_w2 = R_22$*.*

These cubes (apart from the 2x2x2 one) are solved in 4 steps:

- **Step 1**, make the composite centres of the reduced 3x3x3 cube by putting all small centres of the same color together. Composite centres mast be in the correct position to each other. If the yellow composite centres is on the up face, you must place the red one on the left and the green one on the right. Moreover, blue must be opposite to green, orange to red and yellow to white.

- **Step 2**, make the composite edges by putting all small edges with same colors together.

- **Step 3**, solve the cube as a 3x3x3 classic cube.

- **Step 4**, solve parities.

It is important to remember that external layers can be always turned without messing up edges and centres.

2.1 2x2x2

Figure 10: 2x2x2

This cube can be solved as a classic 3x3x3 cube just taking into account that the middle layers do not exist as follows:

- Using algorithm classic(1), place the 1st layer corners in the right place in such a way colours matches.

- Using algorithms classic(5) (Sune), Orient Last Last layer (OLL), i.e. make the top of last layer completely yellow.

2.2 4x4x4 - Rubik's Revenge

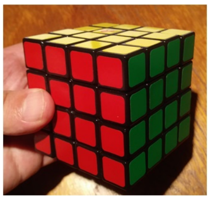


Figure 11: 4x4x4 - Rubik's Revenge

Step 1:

With a little bit of practice to build your intuition, centres can be solved using the algorithm revenge(1): $R_w\, Un\, (Fm)\, R'_w$ and its symmetric where n and m are any integers. This move can be used to make cluster of two centres as well as to put 2 clusters of two in a clusters of 4. In particular for n=2, revenge(1) take two clusters of two centres on the 3rd layer (longitudinally) on the front and up face and put them together to form a centre of 4.

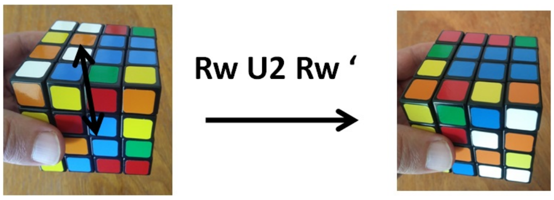


Figure 12: revenge(1): $R_w\, Un\, (Fm)\, R'_w$

It may happen that to complete the last two centres there is one last piece to swap. This can be done with algorithm revenge(2) which is a commutator that can be used to swap pieces on centre's diagonals.

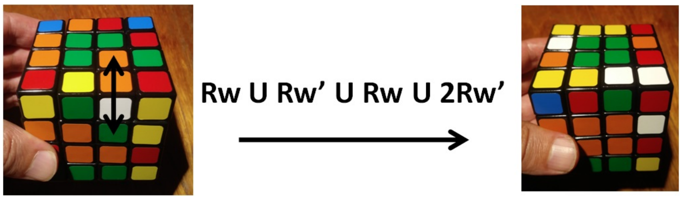


Figure 13: revenge(2): $R_w\, U\, R'_w\, U\, R_w\, U2\, R'_w$

Step 2:

To complete each edge use algorithms revenge(3).

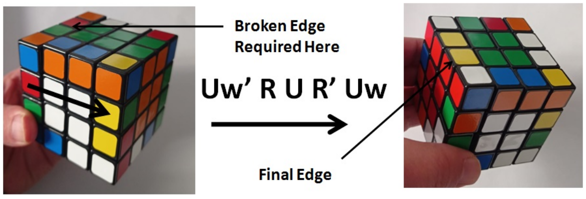


Figure 14: revenge(3): $U'_w\, R\, U\, R'\, U_w$

By rotating external layers it is always possible to place the edges, to which apply the algorithm, in the correct position. For example you can turn the edge on the 4th column of the front face upside down using the sequence $R'\ D\ B\ R'2$, which is so intuitive that I do not even give it as an algorithm.

To use algorithm revenge(3) you need at least a 3rd broken edge (i.e. not completed) to be put in the back of the up layer. It may happen that only two edges are left to be completed. In this case you can use algorithm revenge(4).

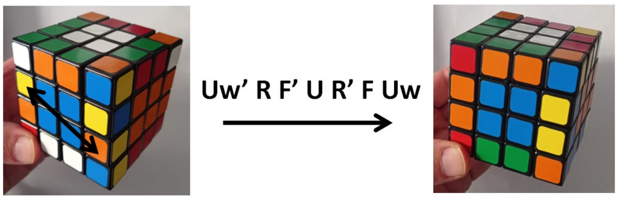


Figure 15: revenge(4): $U'_w\, R\, F'\, U\, R'\, F\, U_w$

Note that in the above algorithm, U'_w is a set up move, $R\ F'\ U\ R'\ F$ is a flipping algorithm that puts the right edge upside down and U_w undoes the set up move. A slightly longer algorithm equivalent to revenge(4) which to me is easier to remember is the the following:

$$\text{revenge}(5):\quad \overbrace{U'_w}^{\text{set-up}} \quad \underbrace{\overbrace{R'\,F\,R\,F'}^{\text{sledgehammer}} \quad \overbrace{R\,U'\,R'}^{\text{re-insert}}}_{\text{flipping alghoritm}} \quad \overbrace{U_w}^{\text{undo set-up}}$$

where the reinsert move takes the edge, that ended up on the top after the sledgehammer, and reinsert it on the right hand side of the cube.

Step 3:

Solve the cube as a classic 3x3x3 cube. Be aware that due to the presence of a parity a complete cross may not be possible to be achieved and one of the yellow edges may be missing making the cross to look like a T instead. With a bit of experience this cases will be easy to be recognised.

Step 4:

For this cube there are two parity cases possible. Many tutorials describe 3 parity cases. However, when solving the cube in Step 3, it is always possible to permute edges of the yellow layer by using algorithm classic(7) in order to reduce the cube to one of the two parity cases (or both at the same time) described below.

Parity case 1 (OLL parity):

This parity case is when one edge of the cube is flipped with respect to its correct orientation. This parity has a very long and awful algorithm to be solved. In order to help memorizing this algorithm we will use a special notation:

$$\left\{ \begin{array}{ccccc|ccccc} R\downarrow & = & R'_wRU2 & = & R'_2U2; & R\uparrow & = & R_wR'U2 & = & R_2U2 \\ R\updownarrow & = & R_w2R2U2 & = & R_22U2 & & & & & \\ L\downarrow & = & L_wL'F2 & = & L_2F2; & L\uparrow & = & L'_wLF2 & = & L'_2F2 \end{array} \right.$$

To solve this parity apply algorithm revenge(6).

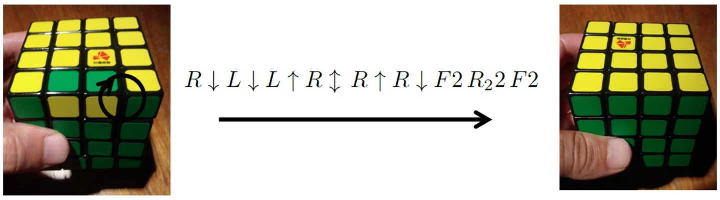


Figure 16: revenge(6): $R\downarrow\ L\downarrow\ L\uparrow\ R\updownarrow\ R\uparrow\ R\downarrow\ F2\ R_2 2\ F2$

Parity case 2 (PLL parity): This parity case is when two opposite edges on a face are swapped. To solve this parity apply algorithm revenge(7).

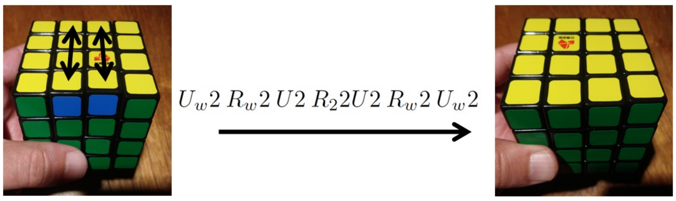


Figure 17: revenge(7): $U_w2\ R_w2\ U2\ R_22U2\ R_w2\ U_w2$

2.3 Bigger Sizes

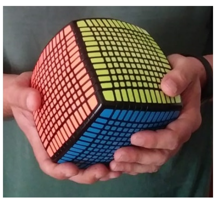


Figure 18: 13x13x13

Once you are able to solve the 5x5x5 cube, you have all the algorithms required to solve any size. In this section I will describe the additional algorithms required.

Step 1:

Complete the first 4 centres using algorithm biggersize(1), which is the equivalent of algorithm revenge(1), till you have two last contiguous centres undone. If the first move R_{ilk} in the algorithm is R_2 or L'_2, then you can use respectively R_w and L'_w (see example below) and change the last move of the algorithm accordingly. This is very useful for the $5x5x5$ cubes and it helps to speed up this step a bit.

With a bit of practice the above task can be done intuitively. Complete one line at a time in a free (not yet done) centres and then insert the line in the centre you are working at the moment. For even sizes, be sure the colour of the composite centres are in the correct position with respect to each other. For odd sizes, the central piece of each face cannot be moved and this will automatically ensure that the relative position of the composite centres is correct.

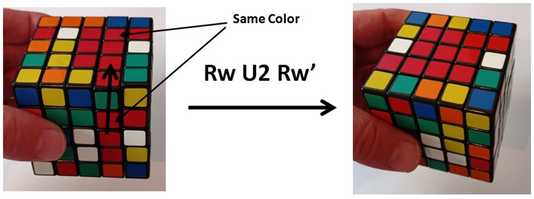


Figure 19: biggersize(1): $R_{ijk}\, Un\, (Fm)\, R'_{ijk}$

To complete the last two centres you need a centres commutator. Before using a commutator, play a bit with algorithm biggersize(1) in order to complete as much of the centrers as possible on the two last faces and in particular try to complete the crosses if the size is odd. This because commutators are boring and the less you use them, the better.

Up to size 5x5x5 the equivalent of algorithm revenge(2) is enough to complete the centres:

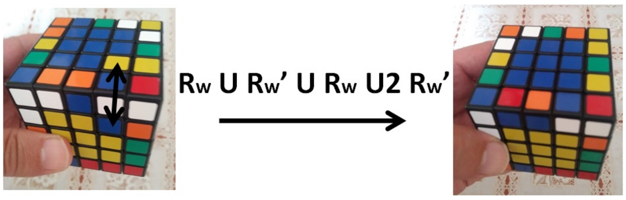


Figure 20: biggersize(2): $R_i\; U\; R'_i\; U\; R_i\; U2\; R'_i$

The above algorithm can actually be used to swaps pieces on centre’s diagonals of cubes of any size.

However, for bigger sizes you need a commutator which swaps any couple of pieces in a (x,y) position. This is algorithm biggersize(3), which is a general commutator and, by being more general then biggersize(2), it can also do the job of algorithm biggersize(2) which therefore it is not strictly needed.

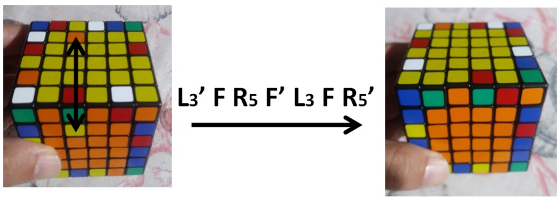


Figure 21: biggersize(3): $L'_x\; F\; R_y\; F'\; L_x\; F\; R'_y$

Step 2:

Edges can be completed with algorithm biggersize(4).

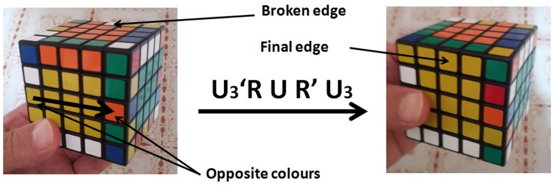


Figure 22: biggersize(4): $U'_{ijk}\,R\,U\,R'\,U_{ijk}$

Where more then one pieces at a time can be inserted if necessary.

When only two undone edges are left, place the two renaming edges on the front face. The two edges can be fixed using algorithm biggersize(5). When placing the pieces in the edges do not worry about orientation, this will be fixed later.

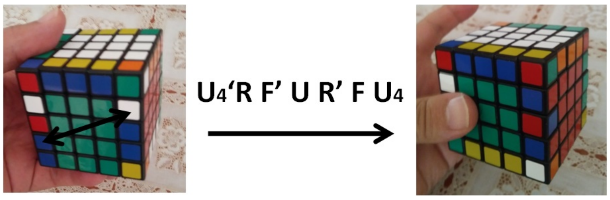


Figure 23: biggersize(5): $U'_{ijk}\,R\,F'\,U\,R'\,F\,U_{ijk}$

In the above algorithm, more then one piece at a time can be inserted if necessary.

Now you need to orient the pieces of the two last edges. With algorithm biggersize(6) it is possible to fix the edge on the right of the front face but the other edge will in general not be fixable. When this happen is because there is an OLL parity (actually, in general, a multiple parity depending on the size of the cube). This can be fixed now or at the end as described in step 4. The layers in the algorithm are the ones relevant to pieces that we do not want to flip. If we turn a layer, we have also to turn the layer symmetric with respect to the middle horizontal layer (the ones with the "*"). The middle horizontal layer is the symmetric to itself.

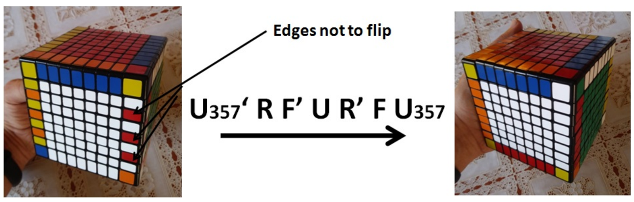


Figure 24: biggersize(6): $U'_{ii^*jj^*}\,R\,F'\,U R'\,F\,U_{ii^*jj^*}$

Regarding the last three algorithms above, two things have to be noticed. As for algorithm biggersize(1), if the first move U'_{ilk} in the algorithm is U'_2 or D_2, then you can use respectively U'_w and D_w and change the last move of the algorithm accordingly. This is very useful for the $5x5x5$ cubes and it helps to speed up this step a bit. Moreover, algorithms biggersize(5) and biggersize(6) can be decomposed in a set-up move, a flipping algorithm (for the right edge) and an undo set-up move. As for algorithm revenge(4), the flipping algorithm can be substituted by a sledgehammer and a re-insert move as in algorithm revenge(5). For example, for algorithm biggersize(5) an equivalent algorithm is:

$$\text{biggersize(7):}\quad \overbrace{U'_{ijk}}^{\text{set-up}}\quad \underbrace{\overbrace{R'\,F\,R\,F'}^{\text{sledgehammer}}\quad \overbrace{R\,U'\,R'}^{\text{re-insert}}}_{\text{flipping alghoritm}}\quad \overbrace{U_{ijk}}^{\text{undo set-up}}$$

Step 3:

Nothing to add with respect to smaller size cubes.

Step 4: Odd size cubes can have only the the OLL parity. Even size cubes can have OLL and PLL parities. OLL Parity can occur with multiplicity depending on the size of the cube.

OLL parity can be fixed using algorithm revenge(5) where, being (i,j,k) the layers of the pieces to fix (on one side only), the notation has to be interpreted as follows:

$$\left\{ \begin{array}{rcl|rcl} R\downarrow & = & R'_{ijk}U2; & R\uparrow & = & R_{ijk}U2 \\ R\updownarrow & = & R_{ijk}2U2 & & & \\ L\downarrow & = & L_{ijk}F2; & L\uparrow & = & L'_{ijk}F2 \end{array} \right.$$

where each index will fix a couple of symmetric pieces to be flipped (even if the layer are rotated only on one side) and pieces can be fixed all at the same time. Note that fix one couple at a time, although boring, would work as well.

PLL parity can be fixed using algorithm revenge(6) where U_w, D_w and R_2 have to be intended as all the internal layers on one side of the median plane of the big size cube replacing the 2 layer of the 4x4x4 cube.

3 Cubes Mechanically Equivalent to the Classic 3x3x3 Cube

3.1 3x3x3 Sizes Cubes

Figure 25: 3x3x3 Mechanically Equivalent to the classic 3x3x3 Cube

In the picture above, from left to right and from top to down we have: Axel or Axis cube, Mastermorphix, 3x3x3 Rhomboidal Dodecahedron, Penrose Cube, Fisher Cube, Windmill or Wheel Cube, Pandora cube and Heart shaped.

These cubes are mechanically equivalent to the classic cube and can be solved using the same method. With a bit of practice it will be possible to visualise centre pieces, edges and corners, although they may have a different shape and a number of colours different from the classic cube. I suggest to use option one for solving the last layer.

For these cubes, after completing the first layer, the centre pieces of the middle layer will not be in general oriented correctly although they will always be in the correct position. Before completing the second layer it is therefore necessary to orient them using algorithm equivalent(1):

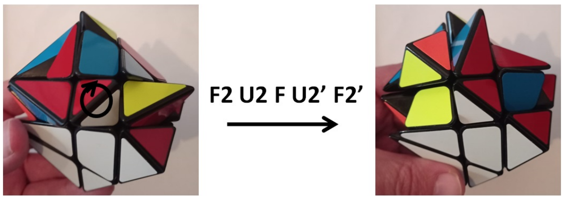


Figure 26: equivalent(1): $F2\,U2\,Fn\,U2'\,F2'$

where n can be any integer needed to orient the centre pieces.

Parity: Some of these cubes also have parity cases which are specific of each different cube. The reason for the parity is that each cube may have pieces that are identical and that may get swapped while solving it. Unless you want to work out a specific algorithm for each parity case of each cube, a way to fix it is to find out what pieces are swapped causing the parity at hand, swap them back in the correct position trying to mess up the cube as little as possible, and complete the cube from that point. This will allow you to fix any possible parity you will come across with this kind of cubes.

3.2 Megamorphix Bigger Sizes

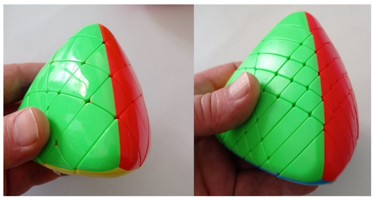


Figure 27: Megamorphix Bigger Sizes

A special attention is deserved by the Megamorphix Bigger Sizes puzzles. They can be done with the same algorithms of the equivalent bigger size classic cube. However, the first time you do them they can be unsettling. They have parities cases that can be solved with the same approach of the cubes mechanically equivalent to the 3x3x3 but that are difficult to be worked out. They are long to be completed but they are fun.
NOTE: For even sizes, before you scramble this cubes check the relative position of the face colours. I have seen on the marked cubes where colours are arranged in different ways which are not always standard.

4 Cuboids

4.1 2x2x3

Figure 28: 2x2x3 - Tower

This cuboid can be solved with the same algorithms of the 2x2x4 cuboid with the difference that there is only a single central slice (instead of two) that can be solved easily and intuitively. Once the central slice is solved, the rest of the cuboid can be solved in the same way as for a 2x2x4 cuboid.

4.2 2x2x4 - Rubik's Tower

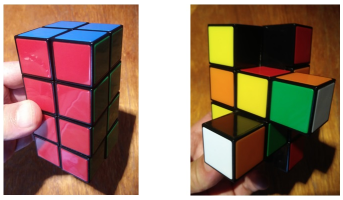


Figure 29: 2x2x4 - Tower

As a first remark, this cube should be scrambled initially in its original tower shape since, once the external layers do nor form a 2x2 cube with the next slice, it it impossible to rotate them along the tower axis and the cube may be non scrambled properly.

In the cube in the above fig the down face is white and the up face is blue. They may exist cubes with different color arrangement but I will refer to the up and down faces in the description below as the white and blue faces.

- Solve the inner 2x2x2 cube with the standard algorithms for it. At first it may be difficult to visualise it as a 2x2x2 cube but this can be done with a bit of practice. For each edge of the two inner layers there are two pieces that have the same two lateral colors, one for each of the two inner layer. This two pieces may have two white, two blue or a white and a blue piece attached to their 3rd face. All combinations are possible. It does not matter which of the two identical pieces it is used when completing the 1st layer of the inner 2x2x2 cube and which of the two identical pieces goes to the 2nd layer.

- When the inner 2x2x2 cube is completed the cube will have its original tower shape. Find out which face is white and which face is blue by rotation on on the external layer till a piece matches with the color of the inner 2x2x2 cube.

- By using algorithm cuboid(1) place all the pieces of the first layer (white face) in the correct positions. This is always possible by swapping pieces between the 1st and the 4th layer. Algorithm cuboid(1) will rotate the right layer of the internal 2x2x2 cube. this is not a problem and can be fixed later as long as when applying the algorithm the rotated layer is kept always on the right.

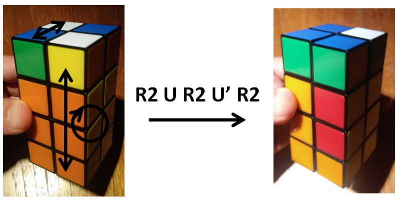


Figure 30: cuboid(1): $R2\,U\;R2\,U'\;R2$

- When the 1st layer is completed, if the right layer of the inner 2x2x2 cube is rotated, rotate it back by using algorithm cuboid(2).

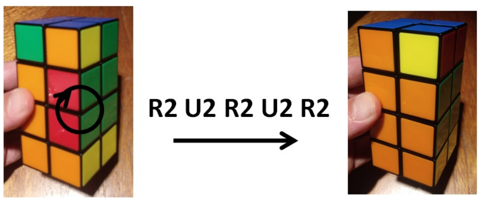


Figure 31: cuboid(2): $R2\,U2\;R2\,U2\;R2$

- Now the solve the last layer applying algorithm cuboid(3) to swap two pieces of 4th layer. It is always possible to sove the last layer applying algoritm cuboid(3) at most twice.

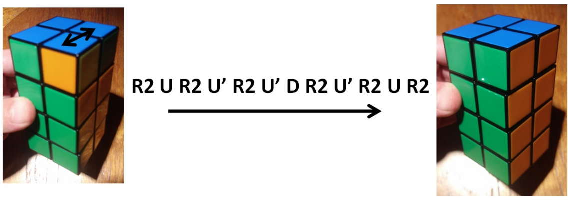


Figure 32: cuboid(3): $R2\,U\;R2\,U'\;R2\,U'\;D\;R2\,U'\;R2\,U\;R2$

4.3 2x3x3 and Barrel

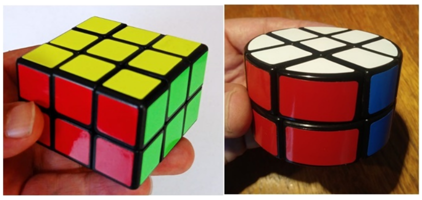


Figure 33: 2x3x3 and Barrel

TBD

4.4 Rubik’s Edge

Figure 34: Rubk’s Edge

TBD

4.5 1x2x3

Figure 35: 1x2x3

TBD

5 Megamix Puzzles

This are very nice puzzles that, although mechanically different from classic cubes, can be solved with algorithms very similar to the classic 3x3x3 ones.

5.1 Megamix Size 2 (Kilomix)

Figure 36: Megamix Size 2 (Kilomix)

This cube can be solved as a standard Megamix just taking into account that the middle ”layers” do not exist.

5.2 Megamix

Figure 37: Megamix

The Megamix has a 5 edge star instead on the cross if compared with a classic 3x3x3 cube. Moreover, it has 5 lines of pieces (that cannot be moved with respect to each other) that have to be completed from the down face to the up face. The central line (line 3) is a zigzag line with down and up corners. Lines 2 and 4 in five points are thicker then the other lines since they include a central piece.

As a matter of fact, a Megamix can be solved using exactly (or almost exactly) the same algorithm of a classic 3x3x3 cube.

Line 1:

I suggest to use the white face as down face. Complete the white star. This can be done intuitively. Insert the corner of the down layer. this also can be done intuitively. With a bit of practice you will notice that the algorithm to insert corners is algorithm classic(1) used for the classic cube where Up and Right faces have to be interpreted in the obvious way. Once the corners are in place the first line will be completed.

Line 2:

To complete this line you need to insert edges in the right place. This is equivalent to complete the second layer of a classic cube and algorithms classic(1) plus classic(2) can be used where Up and Right faces have to be interpreted in the obvious way. With a bit of practice you will understand that if you can complete the second layer of a classic cube then you can complete the second line of a Megamix.

Line 3:

Insert the edges in the right position and with the right orientation. Insert the down corners in the zigzag line using algorithm megamix(1).

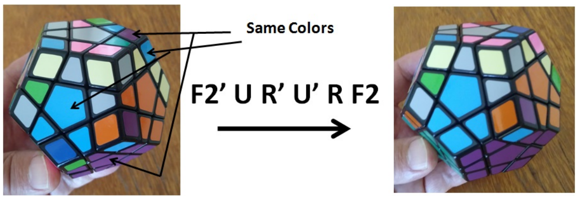


Figure 38: megamix(1): $F2' \, U \, R' \, U' \, R \, F2$

Insert the up corners. This can be done intuitively. With a bit of practice you will notice that the algorithm to be used is the same for inserting corners in the first layer of a classic cube which is algorithm classic(1).

Line 4:
Insert the up edges. This can be done intuitively. With a bit of practice you will notice that the algorithms to be used are the same for inserting edges in the second layer of a classic cube which are algorithms classic(1) plus classic(2).

Line 5:
Make the star (usually grey depending on the cube) on the top layer. This can be done with algorithm megamix(2) which is identical to algorithm classic(3). To do this you need to use the above algorithms several times.

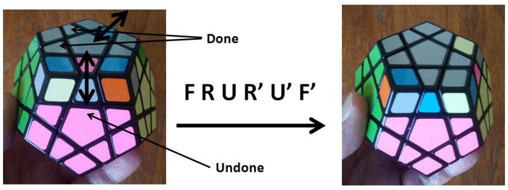


Figure 39: megamix(2): $F \, R \, U \, R' \, U' \, F'$

Permute edges till each edge is in the right position. This can be done with algorithm megamix(3) which is the equivalent of the Sune for classic cubes, which is algorithm classic(3). To do this you need to use the above algorithms several times. Note that the algorithm can be performed also inverting the sense of rotation of the move before the last as "$R \, U \, R' \, U \, R \, U2' \, R'$" and this is the way it is normally presented in tutorials because it is better to be performed in speed cubing. However, I prefer the other way because I do not do speed cubing and I like to turn the Up layer always in the same sense. When presented this way, algorithm megamix(3) it is identical to a Sune where Up and Right faces have to be interpreted in the obvious way.

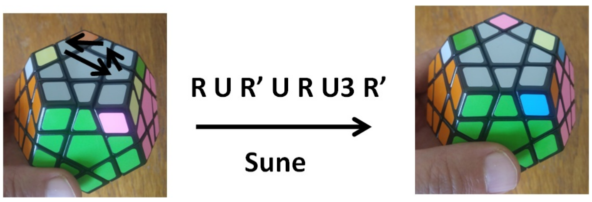


Figure 40: megamix(3): $R \, U \, R' \, U \, R \, U3 \, R'$ (Sune)

Permute corners till each corner is in the right position. Do not worry about orientation This can be done with algorithm megamix(4) which is the equivalent of algorithm classic(4). To do this you need to use the above algorithms several times.

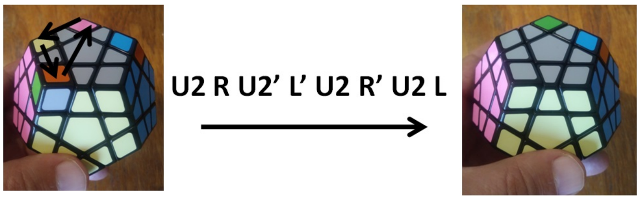


Figure 41: megamix(4): $U2\, R\, U2'\, L'\, U2\, R'\, U2\, L$

In tutorials it is often presented algorithm megamix(5) that does the same job and that it is presented here for completeness.

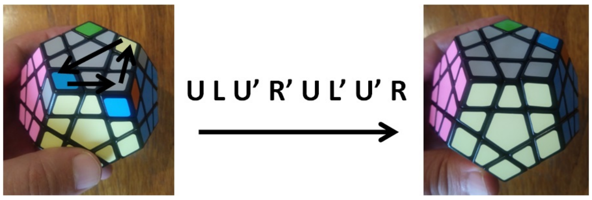


Figure 42: megamix(5): $U\; L\; U'\; R'\; U\; L'\; U'\; R$

Orient the last layer (OLL) using algorithm classic(6) where Up and Right faces have to be interpreted in the obvious way. This bit is identical to solving a classic cube using option 1 for completing the last layer as presented in the paragraphs above.

5.3 Megamix Size 4 (Master Kilomix)

Figure 43: Megamix Size 4 (Master Kilomix)

For classic cubes we gave the solution of the $4x4x4$ (Rubik Revenge) cube first and then we went to higher dimensions cubes. For the Megamix, we will do the opposite and we will not give the solution of the Size 4 but we will rather go directly to the size 5 Megamix (Gigamix) which is the more common one. The Size 4 Megamix can be solved as a Size 5 just taking into account that the mid "layers" do not exist and that faces have no unmovable centre pieces that gives a reference for the color of faces. Centre faces have to be solved pacing them in the right position with respect to each other.

Parity: The Size 4 Megamix (Master Kilomix), as for the the Rubik's Revenge cube, has the PLL parity case. The difference with the Rubik's Revenge is that it has not OLL Parity. The PLL Parity can be solved with the following algorithm:

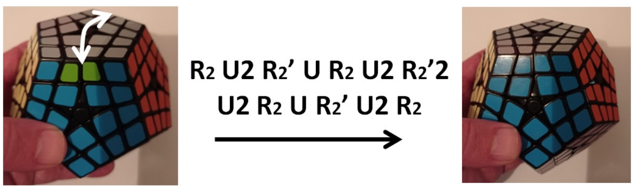


Figure 44: masterkilo(1): $R_2\, U2\, R_2'\, U\, R_2\, U2\, R_2'2\, U2\, R_2\, U\, R_2'\, U2\, R_2$

5.4 Megamix Size 5 (Gigamix)

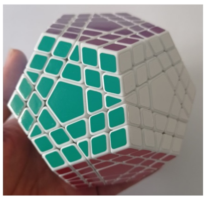


Figure 45: Megamix Size 5 (Gigamix)

Before we give a solution, we state two facts about the Gigamix puzzle. **Fact number one:** the Gigamix can be solved by reducing it to a standard Megamix. This reduction process can be done with the very same algorithms of the $5x5x5$ classic cube. For the above reason the reader does not strictly need this section. **Fact number two:** the Gigamix simply has not parity cases.

Solution... TBD

5.5 Bigger Size Megamixes

Magamixes of size N can be solved with the same algorithms of the relevant NxNxN classic Ribik's cubes where the reader may also look at some of the advises given for the Megamix Size 5 above. The main difficulties are of course in adapting the algorithms used to complete centres and edges in $NxNxN$ cubes to Megamixes. However, we are confident that the reader will be able to do it.

Parity: Size N Megamixes have the PLL parity case for N even and no parity cases at all for N odd. The PLL parity can be solved with algorithm masterkilo(1) given for the Size 4 Megamix (Master Kilomix) where the inner layers, represented by the w index, can be moved together (as if they were glued) in two even groups to represent the two single inner layer of the Kilomix.

6 Square-one

A Square-one Cube has three type of pieces. Edges, Corners and pieces of the middle layer. The standard notation used for the square-one is very different from the notation for other cubes.

Figure 46: Square-one

Notation. *We will use the standard notation for the Square-one cube. Up and down layer can be rotated of a multiple of the angle of an edge piece (360/12 degs). Note that the angle of a corner piece is 2*360/12 degs. Only 3 moves are possible:*

- The up layer can be rotated by a multiple m of 360/12 Degs clockwise (m positive) or anti-clockwise (m negative).

- The down layer can be rotated by a multiple n of 360/12 Degs clockwise (n positive) or anti-clockwise (n negative).

- The right part of the cube can be rotated by 180 degs and it is notated with a slash.

- Moves are grouped in three moves at a time and notated as $m, n/$ meaning: Up rotation by n, Down rotation by m, right part of the cube rotation for the slash.

- If the firs up and down layer rotations are both zero, these are not written down and the algorithm start with a slash. Same applies for the last slash of the notation that will be omitted if not required.

For example with the notation "/-3,0/0,3" we mean a rotation of the right part of the cube followed by a rotation anti-clockwise by $3 * 360/12$ degs of the up layer only, another rotation of the right part of the cube, etc..

Note that apart from algorithm square-one(1) (where there is no proper square in the top layer), in all other algorithms the slash move is to be intended along the plane that goes from 1/3 on the left on the front edge of the top layer to 1/3 on the right on the back edge of the top layer.

Square-one Cube can be solved in 7 steps:

- **Step 1**: Make a Cube Shape.
- **Step 2**: Corner Orientation.
- **Step 3**: Edges Orientation.
- **Step 4**: Corner Swap.
- **Step 5**: Edges Permutation.
- **Step 6**: Parity.
- **Step 7**: Fix the Middle Layer.

Before we start, I would like to introduce a simple algorithm which will be needed only at the very end in step 7 but that it can be used to familiarise with the notation. This algorithm can be used to flip the right central layer piece by 180 degs and, of course, it can be used to flip both central layer pieces just by turning the cube by 180 degs around the z axis.

Although not really needed till the end, especially at the beginning, we may use this algorithm to flip central pieces and give to the central layer a shape we like, for example, for better grabbing the cube.

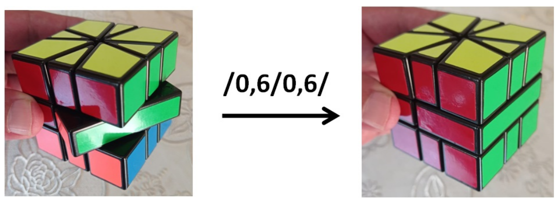


Figure 47: square-one(1): **/0,6/0,6/**

Step 1: Make a cube shape

This step consists in giving the right cube shape to the Square-one. This applies to the up and down layer while we do not care for the moment to the shape of the middle layer. Do do so, we group all the edges together on one layer as shown in the left hand picture of the fig. below. Although not completely trivial, this can be done intuitively and I will not give algorithms for it at least in this revision of this tutorial. The first time you do it, it may take 10 or 20 minutes. As long as you get familiar with the cube it will take much less.

Once you have grouped all edges in one layer, facing the centre of the edges on the up layer, this step can be completed by applying algorithm square-one(2).

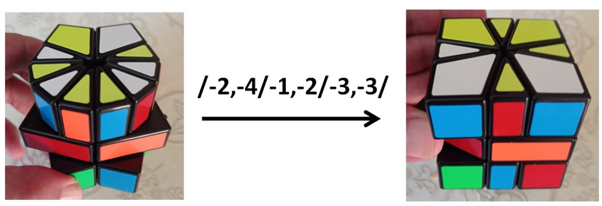


Figure 48: square-one(2): **/-2,-4/-1,2/-3,-3/**

Step 2: Corners orientation

This step consists in putting all the corner with the same color on the same face. This step can be performed intuitively. The only need to be careful not to break the square shape of up and down layers. In order to do so, we misalign the up and down face by 360/12 degs as in the left hand picture of the figure below. Once we do that, using only 90 degs rotations of up and down layers and the slash move we can easily orient all the corners without the need for any algorithm.

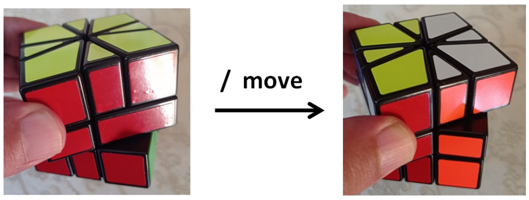


Figure 49: Never break the up and down layers square shape

Step 3 - Edge Orientation: For this step we need two algorithms:

$$\begin{cases} 1) & \text{sqare-one}(3): \quad 1,0/\text{-}1,\text{-}1/ \\ 2) & \text{sqare-one}(4): \quad /3,0/\text{-}3,\text{-}3/0,3/ \end{cases} \tag{1}$$

The figure below shows the edges of the wrong color with respect to the face where they are placed. Algorithms 1 and 2 above can be used to cycle among the various configuration and you can join the chain of configurations at any point according to the configuration you have and you need to fix.

In the picture below, configurations with 3 or 4 edges on the wrong face are missing. However, by applying one of the two algorithms above to these configuration in a more or less obvious way, you can always get to one of the configurations shown in the figure below.

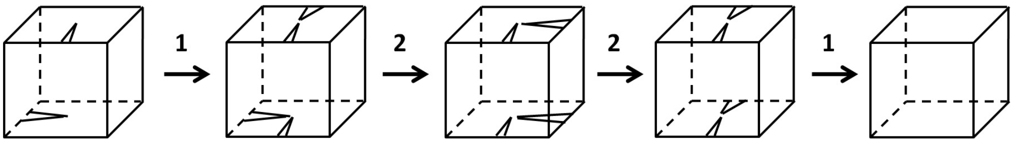


Figure 50: Edge Orientation

Step 4: Corners Swap

Aim of this step is to put the corners in their correct position. To do that, place two corner with matching colors on the left and use algorithm square-one(5). If you do not have two pieces with matching colors, perform the algorithm once and you will have them.

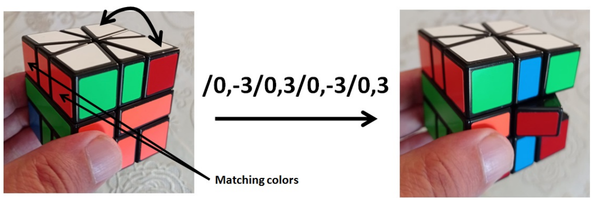


Figure 51: square-one(5): **/0,-3/0,3/0,-3/0,3**

Once you have completed one layer, turn the cube upside down and do the other one. Please bear in mind that algorithm square-one(5) will mess up the down layer. However, if you perform the algorithm a second time without turning the down layer, this will fix it. This means that the second layer has to be done using the algorithm an even number of time.

To do that, when working on the second layer, place the pieces with the matching colors at the back. Perform the algorithm once. This will not solve the up layer and will mess up the down layer. Turn the up layer in order to have the matching colors on the left. Apply the algorithm again.

Step 5: Edges Permutation

Aim of this step is to permute edges and complete the up and down layer. To permute the edges we use algorithm square-one(5) as we did in previous step. This time we perform the algorithm, we do the set-up move 1,0, we perform the algorithm again and we undo the set-up move. This will turn the edges on the right clockwise. by performing algorithm square-one(5) twice, we do not mess up corners.

If you cannot complete the task (i.e. two edges swapped) it means you have a parity.

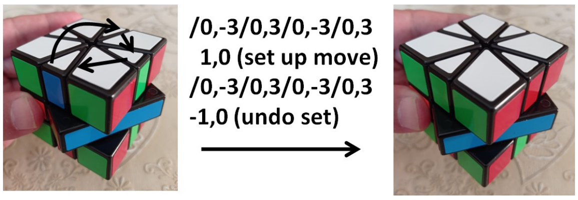


Figure 52: square-one(6): square-one (5) 1,0 (set-up move) square-one(5)

Finally, the symmetric algorithm turning the edges anticlockwise is as follows: we perform the set-up move first, we perform algorithm square-one(5), we undo the set-up move with the move -1,0, we perform the algorithm square-one(5) again.

Step 6: Parity

You have parity when two edges of the same face are swapped. This can happen to one or both faces. To fix the party there is a long and nasty algorithm that swaps the two edges.

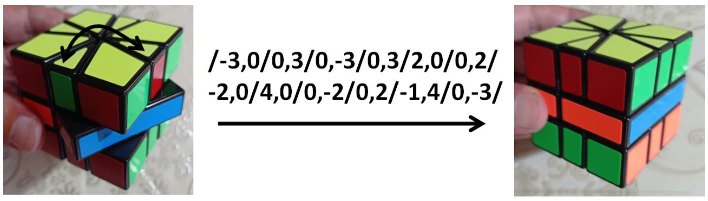


Figure 53: square-one(7): **/-3,0/0,3/0,-3/0,3/2,0/0,2/-2,0/4,0/0,-2/0,2/-1,4/0,-3/**

Step 7: Fix the Middle Layer

In this last step we fix the middle layer in terms of shape and color matching. This can be done using one or twice algorithm square-one(1).

7 Pyramidal Puzzles

This section contains puzzle which have a pyramidal shape. They are grouped all together although they may be mechanically uncorrelated.

7.1 Pyramix

Figure 54: Pyramix

This puzzles can be solved according the following steps:

First Layer: Choose a color (e.g. blue) and orient centre pieces in order to move them on the relevant face. Corners can be oriented at any time but is is useful to orient them at this stage to have a reference for the color of the adjacent faces to the one you are competing.

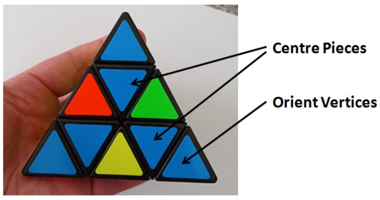


Figure 55: Pieces Names

Complete the first layer using algorithm pyramix(1):

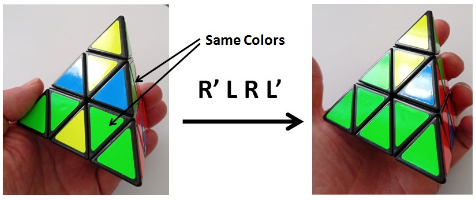


Figure 56: pyramixb(1): R' L R L' (Sledghammer)

Second Layer: Orient the centre piece of the second layer (by turning it) till they match the color of the first layer. Once you have done it, you need to fix the edges of the second Layer. You can have two cases:

Case 1: One edge of the second layer is in place and two are swapped. Use algorithm pyramix(2) to solve the second layer.

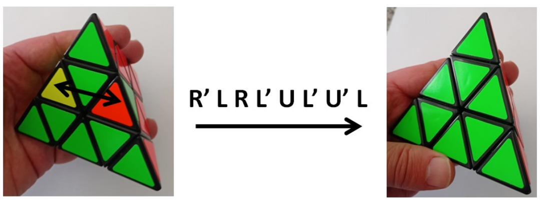


Figure 57: pyramixb(2): R' L R L' U L' U' L (2 x Sledghammer)

Case 2: None of the edges of the second layer is in place and two are swapped. Use algorithm pyramix(3) to rotate edges till the second layer is solved or you end up to Case 1:

Figure 58: pyramixb(3): $R' \; U \; R' \; U \; R \; U \; R'$

Third layer Layer: Trivial.

Note that algorithm pyramix(3) is very similar to a Sune. Moreover, algorithm pyramix(2) is basically a sequence of two Sledgehammers with a rotation of the cube by 120deg anticlockwise around the axes that from you goes toward the puzzle.

7.2 Flower/Petal Pyramix

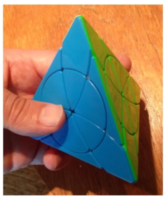


Figure 59: Flower/Petal Pyramix

TBD

7.3 Pyramix Duo

Figure 60: Pyramix Duo

TBD

8 The Skewb Family

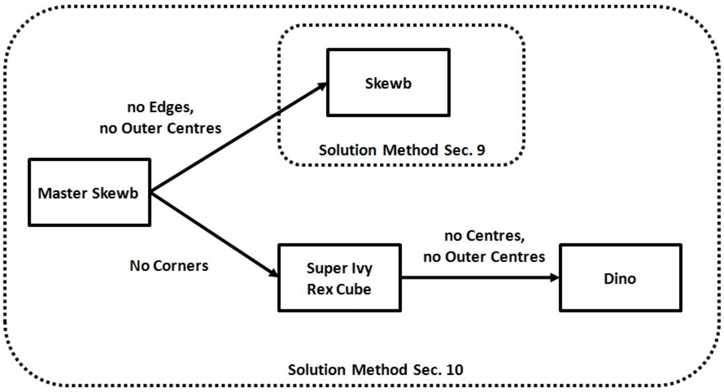


Figure 61: Skewb Family Classification

The Skewb family is composed by several members which, as shown in the picture above, do not fall all in a straight mechanical inclusion line. The Master Skewb is the mechanical superset of all of them and if you can solve it, you can solve all the family. The reader my try, as an exercise, to solve the Skewb using a subset of the steps needed to solve the Master Skewb (Steps 1, 4, 5 and 6). However, solve the Skewb using the Master Skewb method is not efficient. In this section we will give a method to be used specifically for the Skewb. In the next section we will give a method to be used for all other members of the family.

8.1 Skewb

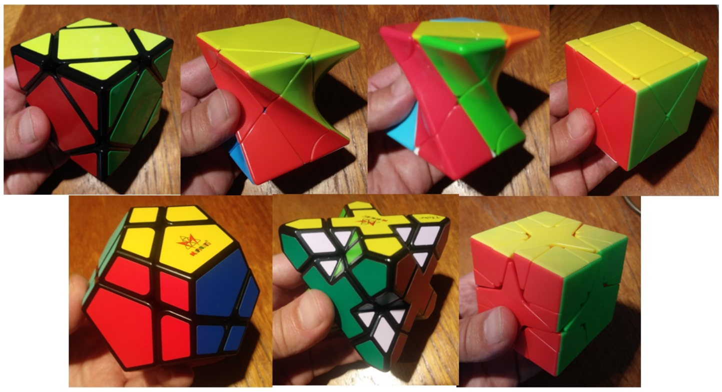


Figure 62: Skewb

In the picture above, from left to right and from top to down we have: Skewb, Twisted Skewb, Twisted Skewb (again but this is a second variant), Fisher Skewb, Skewb Ultimate, Skewb Xtreme and Polaris Cube.

All the above cubes are mechanically equivalent and they can be solved with the same method. We will describe how to solve the most common ones present in the picture above.
Notation. *For the Skewb we need a new notation. The Skewb can be solved (beginner method) just rotating along the two axis in the figure below. As usual, R and L will be used for clockwise rotations, R' and L' will be used for anti-clockwise rotations. In the figure I show also the z axis since we often need to rotate the cube by 180 degs around this axis. This move will be notated as $z2$.*

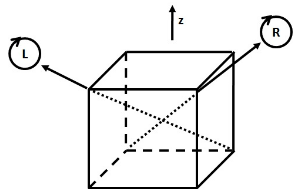


Figure 63: Skewb Notation

Skewbs have centre pieces and corners. For some Skewbs, as the standard Skewb, centre pieces are symmetric in shape and have one colour. For such pieces we do not need to bother rotating them to solve the Skewb. For some other Skewbs, as the twisted Skewb, this is not true. For the above reason, the steps to solve the standard Skewb is a subset of the steps required for the twisted Skewb and will not be presented here. For the solution of these cubes the reader can refer to the section for the twisted Skewb and just ignore the last step where you rotate centres which, for what we said, is not needed.

8.2 Twisted Skewb

To solve the Skewb we basically need only one algorithm, called Sledgehammer, reported in the figure below. Some tutorial give a Sledgehammer different from the one below performed rotating different axis or with a different order of Left and Right rotation. Hoverer, given the due symmetries all algorithms are equivalent.

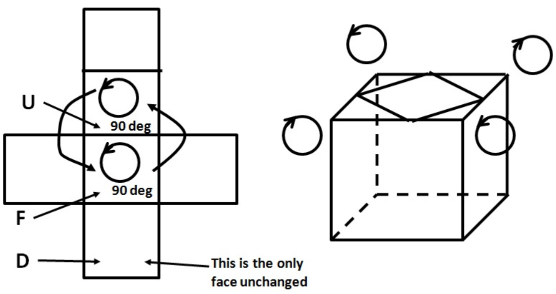


Figure 64: skewb(1): $R'\ L\ R\ L'$ (Sledghammer)

In addition to the Sledgehammer, we need two more algorithms which are reported in the figure below and are Sledghammer performed in sequence with additional 180 degs rotations around the z axis.

- skewb(2): $(R'\ L\ R\ L')\ z2\ (R'\ L\ R\ L')\ =\ SH\ z2\ SH$
- skewb(3): $(R'\ L\ R\ L')\ z2\ (R'\ L\ R\ L')\ (R'\ L\ R\ L')\ z2\ (R'\ L\ R\ L')\ =\ SH\ z2\ SH\ SH\ z2\ SH$

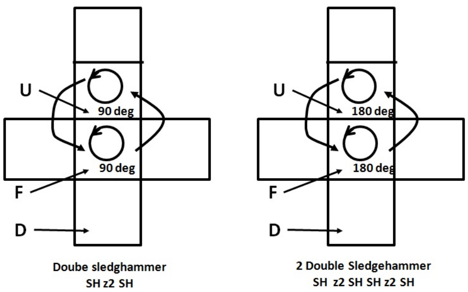


Figure 65: skewb(2): Double Sledghamme and skewb(3): Two Double Sledghammer

To solve the Skewb, perform the following steps.

Step 1:
Complete the white face (or whatever face with a symmetric and one coloured centre piece you have on your cube). This is intuitive. Take the corner to the white face in the correct position. To rotate a corner without rotating the other ones on the same face use algorithm skewb(4) where A and B are the axis in the fugures below. This is, as a matter of fact and once again a Sledghammer.

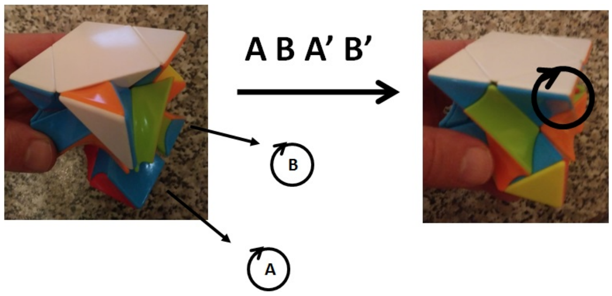


Figure 66: skewb(4): $A\ B\ A'\ B'$

Step 2:
Orient Yellow corners. this should be possible to be done at most using twice algorithm skewb(1) (i.e Sledgehammer) with the white face pointing in the down direction.

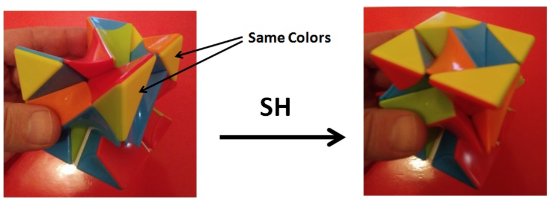


Figure 67: Orient Corners

Step 3:
Complete yellow face. To do so, with the white face pointing down and the yellow centre in the back face, use algorithm skewb(2) (i.e. $SH\ z2\ SH$) once.

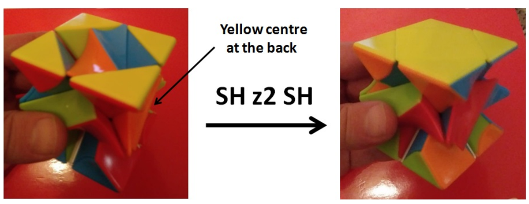


Figure 68: Complete Yellow Face

Step 4:
Give it the right shape and put centre pieces in place. In order to do that, find a centre piece that is already in the right position. If the piece is turned by 180 degs and color do not match, it is not important. Pieces can be turned by 180 degs at the end. If you do not find such a piece,

apply algorithm skewb(2) (i.e. $SH\ z2\ SH$) with white and yellow face on the right and left face (or vice versa). By applying algorithms skewb(2) with the piece already in place at the bottom and the pieces to be turned by 90 degs on the top and back face, you will be able to put each piece in place and correctly oriented (apart the colors). If you have done all correctly, this will give to the skewb the right shape. If you do not do it correctly you may get the skewb with the correct shape but with centre pieces swapped for opposite faces.

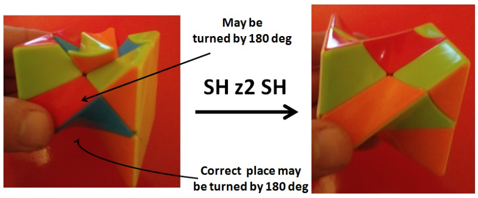


Figure 69: Give it the Right Shape

Step 5:

Rotate the centre pieces by 180 degs to make colors match. Do that by using algorithm Skewb(3) (i.e. $SH\ z2\ SH\ SH\ z2\ SH$) with pieces to rotate on the top and back faces. The algorithm will rotate two pieces at a time. If the pieced to rotate are odd, this can be fixed by rotating also the white or yellow piece that are psychometric and it does not matter if they are rotated by 180 deg. this can be done by applying the algorithm with the white and yellow pieces on the front and back faces or vice versa.

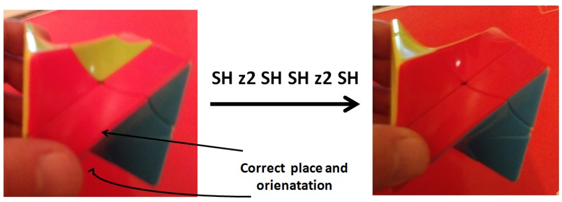


Figure 70: Rotate Centres Faces

8.3 Polaris Cube

This puzzle is a mechanical superset of the Skewb and it can be solved by reduction to it.

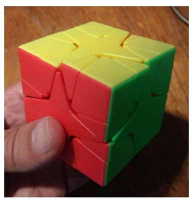


Figure 71: Polaris Cube

TBD

9 Other Members of the Skewb Family

This section is mainly devoted to the solution of the Master Skewb which gives also a solution for the Super Ivy (equivalent to Rex cube) and the Dino puzzles.

Despite misleading names, the Super Ivy (equivalent to Rex Cube) is not a mechanical superset of the Ivy Cube but rather a mechanical subset of the Master Cube. This is why it is addressed in this section.

9.1 Master Skewb

Figure 72: Master Skewb

Pieces Name. *We have four kind of pieces in the Master Skewb. Corners, Edges, Centre Pieces, Outer Centre Pieces. You can easily tell which one is which.*

In the notation, rotations are around axis that go trough the corners of the relevant letter. The ’ sign, as usual indicates anticlockwise rotation. Master Skewb can be solved in 6 steps:

- **Step 1**: Avoid Parity.
- **Step 2**: Edges Permutation.
- **Step 3**: Centres Permutation.
- **Step 4**: Outer Centre Pieces Swap.
- **Step 5**: Corners Swap.
- **Step 6**: Corners Orientation.

Step 1: Avoid Parity

The fist step consist in putting corner in the correct position. Orientation of the corners is not important. When putting corner in place it is important to respect color of faces which are the same of the classic 3x3x3 cube. White opposite to yellow, red to orange blue to green with white green and red in the going anticlockwise around the common corner.

This step may seems useless since corner will be messed up by further move. However, the Master Skewb has a parity on corner permutation. Although further moves proposed in this paper will mess up the corner positions, they will not break the parity that will therefore not show up at the end.

Step 2: Edges Permutation

This step consists in moving all edges in their correct position. Once again, as per previous step, face colors have to respect the position of colors on the classic 3x3x3 cube as explained above. This step can be performed easily using the three algorithms below to make the edges travel around the cube till they are in place and choosing the correct algorithm that does not perturb the pieces already in place. Moreover, using the symmetric algorithms that permute edges in the opposite way, may help.

This algorithms will nor scramble the corners. However, corner will be scrambled in further steps. It is convenient to start always from the same face (e.g. white), however everyone can do the way more natural to him.

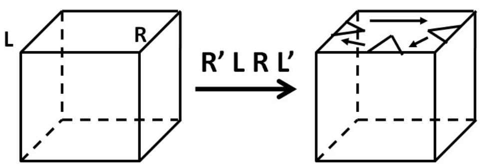


Figure 73: master-skewb(1): $R' \, L \, R \, L'$

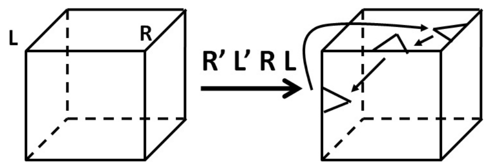


Figure 74: master-skewb(2): $R' \; L' \; R \; L$

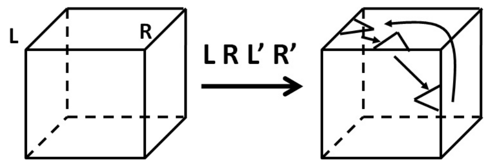


Figure 75: master-skewb(3): $L \; R \; L' \; R'$

At the end of the this step the cube will look as follows:

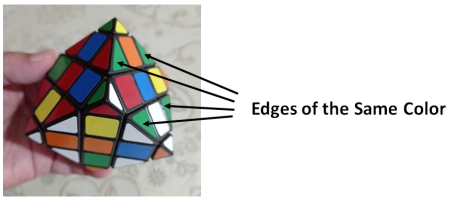


Figure 76: End of Step 2

Step 3: Centres Permutation

Aim of this step is to place the centre pieces in place. This can be easily done with algorithm master-skewb(4) which permute centres anticlockwise and its symmetric that permutes centres clockwise.

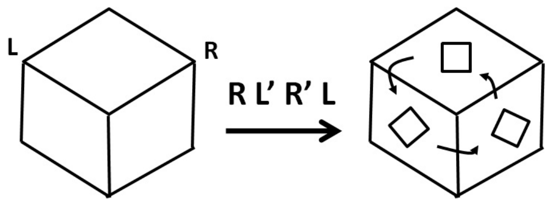


Figure 77: masters-kewb(4): $R \; L' \; R' \; L$

Step 4: Outer Centre Pieces Swap

Aim of this step is to put the outer centre pieces in their correct position. This can be done with algorithm master-skewb(5) (and its symmetric) which swaps outer centre pieces on the same face and for two faces that are next to each other. Note that master-skewb(5) is composed by two parts. The first one is a rotation anticlockwise of the centre pieces (i.e. algorithm master-skewb(5)). The second part is its antisymmetric with a clockwise rotation. The 120 deg Rotation between the two parts will let to a final effect of swapping outer centre pieces.

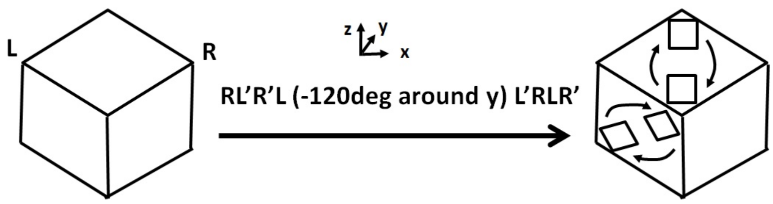


Figure 78: master-skewb(5): $R \; L' \; R' \; L$ **(-120deg [y])** $L' \; R \; L \; R'$

Since to perform our task we need to swap pieces on two different faces, the trick to use is to do a set-up move that puts the two pieces to be swapped on the same face, to perform algorithm master-skewb(5) and to undo the set-up move at the end.

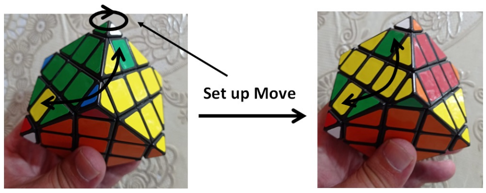


Figure 79: Set-up Move

Remember that the move will swap pieces on two different faces. For the other face use a face with colors that have not been addressed yet or swap pieces of the same color.

Sometimes the piece you want to move to the face you are completing is not on the correct side of the centrepiece. In this case you can use algorithm master-skewb(5) without a set-up move to move it on its face.

It may happen that the piece that you need is not on a face next to the one you are completing. In this case you can move to the correct face in two steps applying twice the algorithm. Another way to do it is to use two set-up moves and move the piece on the correct face in one go. This is a bit more advanced but I am sure you can figure out how to do it.

Step 5: Corners Swap

In the last step we want to put each corner in the correct position. This can be done with algorithm master-skewb(6). Note that this algorithm is simply algorithm master-skewb(4) repeated 3 times with a net effect of leaving the centre pieces unchanged but with an effect o the corners.

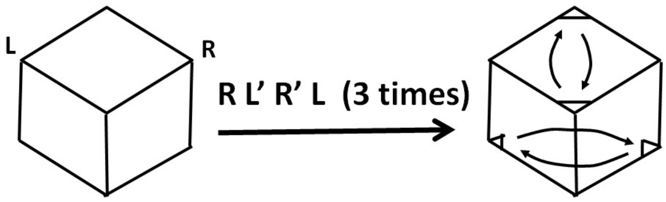


Figure 80: master-skewb(6): $R\,L'\,R'\,L$ **(3 times)**

When you do this, you want your corner (the one closer to you) to go in the correct position and with the correct orientation. In order to do that you may need to perform a set-up move to make the up color of the corner match with the color of the centre piece on the up face (see fig below). Once you have performed algorithm master-skewb(6), remember to undo the set-up move.

Remember that algorithm master-skewb(6) swaps two couple of corners and this may be confusing. What I usually do to complete this step is to move the corners belonging to the white face in their place and this will fix the other one but you can use the method you like more.

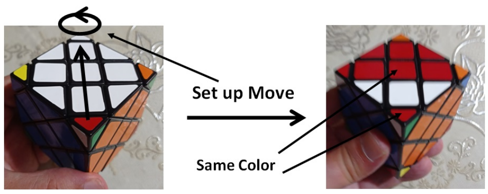


Figure 81: Set-up Move

If you performed correctly step 1, you should be able to complete this step easily. Otherwise you will incur in a parity. I have no algorithm to fix this so go back to step 1, perform it to fix the parity trying to mess up the cube as little as possible, and go on from there.

Step 6: Corners Orientation

Once you have placed all corners in the correct position, some of them may still need to be orientated correctly. In order to do this, you can rotate the cube in order to have the corner you want to rotate on the top face further away from you. Then swap twice the two corners on the top face using algorithm master-skewb(6) twice and before the second swap you use a set-up move to put the corner closer to you (i.e. the one you wanted to rotate) back to its position with the correct orientation. After a few attempts, this will eventually make you able to complete the cube.

9.2 Dino

Figure 82: Dino

The Dino puzzle is a mechanical subset version of the Super Ivy (equivalent to Rex Cube) which in turn is a mechanical subset of the Master Skewb. With respect to the latter, it has no corners and no centre pieces. It is a very simple puzzle and it can be solved almost intuitively.

To solve it you need just to apply Step 2 of the Master Skewb where you permute edges. The main difference with the Master Skewb edge solving method is that for the Dino you do not need to avoid any parity because you do not have corners and therefore, when fixing edges, you do not need to use Sledgehammer moves only. This means that you can solve as many edges as you can intuitively and apply the edge permutation algorithms of the Master Skewb Step 2 only to fix the remaining ones.

9.3 Super Ivy and Rex Cube

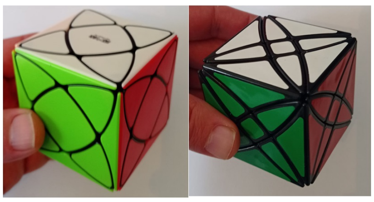


Figure 83: Super Ivy Cube and Rex Cube

The super Ivy puzzle (mechanically equivalent to the Rex Cube) is a mechanical subset of the Master Skewb. With respect to the latter, it has no corners.

You can solve it using the same method of the master Skewb where the first and the last two steps (which apply to corners) are not needed. When solving edges (Master Skewb Step 2) the same comment we made for the Dino (you do not need to avoid corners parity) applies.

10 Other Diagonal Axis Turning Puzzles

This section includes some other diagonal turning axis puzzles.

10.1 Ivy Cube

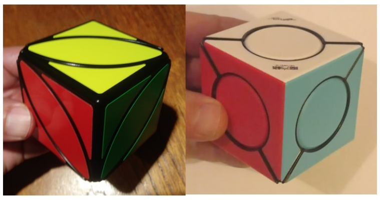


Figure 84: Ivy Cubes

TBD

10.2 Face Turning Octahedron

Figure 85: Face Turning Octahedron

Face Turning Octahedron has not to be confused with Corner Turning Octahedron with is identical when you see it but that has a complete different mechanics. This puzzle can be solved with a method which reminds a lot the one used for the Master Skewb.

In the notation, rotations are around axis that go trough the corners of the relevant letter. The ' sign, as usual indicates anticlockwise rotation. We have three kind of pieces in this puzzle. Corners, Edges and Centre Pieces Pieces. You can easily tell which one is which.

Face Turning Octahedron can be solved in 3 steps:

- **Step 1**: Solve Corners.
- **Step 2**: Centres Swap.
- **Step 3**: Edges Permutation.

Step 1: Solve Corners

Aim of this step is to place corner in their correct position and orientation. Place two corners is straight forward. The third corner can be placed with the move in the figure below which is very similar to the move we use to place corners for the Skewb.

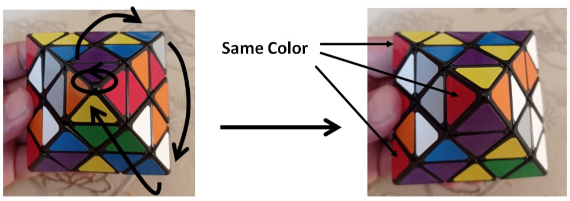


Figure 86: Turn 3rd Corner

At this point we can be in two different cases. The other three corners are already solved or two corners need to be rotated. The second case can be solved using algorithm FTO(1).

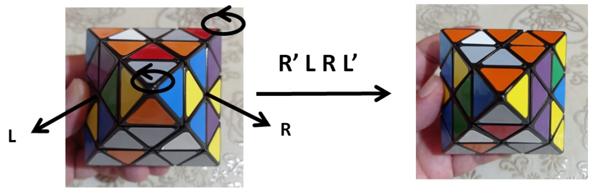


Figure 87: FTO(1): $R' \; L \; R \; L'$

Step 2: Centres Swap

Aim of this step is to place centres in their correct position. This can be done using algorithm FTO(2), which rotates all pieces on a face leaving the rest of the cube unaffected.

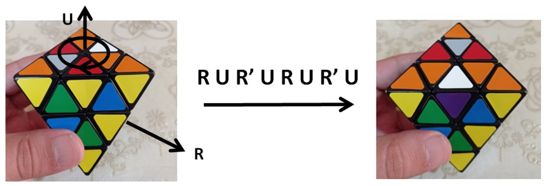


Figure 88: FTO(2): $R \; U \; R' \; U \; R \; U \; R' \; U$

Since to complete the task we need to swap two pieces on two faces, in order to do that we need to use a set-up move, apply algorithm FTO(2), and then undo the set-up move. This is shown in the figure below. Sometimes, before apply the set up move, we need to rotate one of the two faces using algorithm FTO(2).

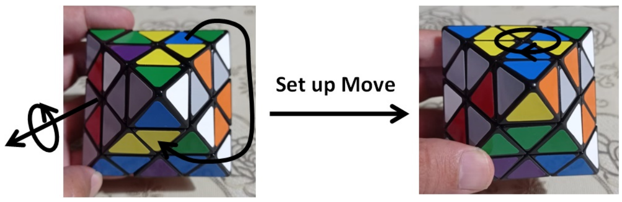


Figure 89: Set-up Move

This is very similar to what we did to solve Outer centre pieces of the Mastercube.

Step 3: Edges Permutation

Aim of this step is to solve the edges. This will complete the cube. We can use algorithm FTO(2) again and make edges travelling around the cube till they are in place. This is very similar to what we did for solving edges of the Mastercube. Once we solve four faces on one side of the cube we may end-up in three cases on the other four faces: the cube is solved. Three pieces are out of place, four pieces are out of place.

If three pieces are out of place, we can permute them using algorithm FTO(3) and its symmetric.

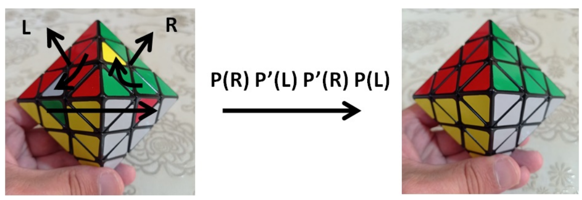


Figure 90: FTO(3): $P(R) \; P'(L) \; P'(R) \; P(L)$

Were with $P(R/L)$ I mean algorithm FTO(2) applied to face R or L (P for permutation) and with P' I mean the symmetric (i.e. rotating face R/L anticlockwise) of algorithm FTO(2). The reader can work out by himself the symmetric algorithm to permute edges clockwise.

If four pieces are out of place, apply algorithm FTO(3) to three of them. One of the 3 pieces will go to the right position and you will be left with only three pieces out o place as per case before.

11 Puppet Cubes

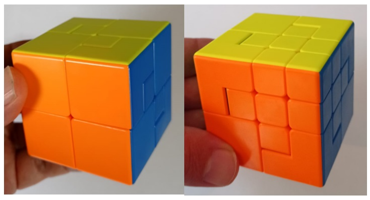


Figure 91: Puppet Cube v1 and v2

These are nice puzzles although a bit long to solve. They come in two versions and they are fun. I recommend everyone to have a go!

11.1 Puppet Cube Version 1

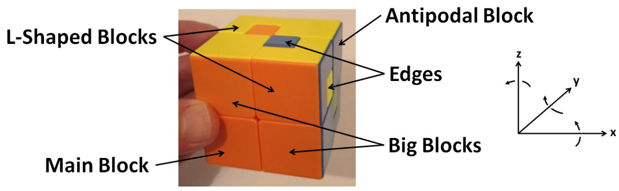


Figure 92: Puppet Cube v1 and v2

We start by giving names to the various pieces of the puzzle. There are 2 Solid Blocks (SB), 3 Big Blocks (BB), 3 L-Shaped Blocks (LB) and 9 Edges (3 of winch hidden inside the cube). The edges are called so because they are actual edges of a classic $3x3x3$ cube hidden inside the Puppet cube. It is difficult to tell the difference between SB and BB the first time you see them. SB come in two different types. I call them with names I made up. The first SB is the vertex of the 3 faces of the cube with no Edges. I call it the Main Block (MB) and it protrudes out from the inner $3x3x3$ cube although you do not see it. In standard colours it is the white-green-orange block. The second SB is the vertex of the 3 faces of the cube with Edges. I call it the Antipodal Block (AP) and it is actually part of inner $3x3x3$ cube since it is made of 8 pieces of the inner cube fused in one single piece.
Cube Position: algorithms have the same notation of a classic cube and are always performed with the MB in the Front Down position. An (R) or an (L) before the algorithm will tell you the position in which the aghoritm has to be performed. An (R) is used when you have to perform the algorithm holding the cube respectively with the Main Bock in the Front Left Down Position, i.e. holding the cube with the Left hand but performing the algorithms mostly with the (R)ight hand. An (L) is used when you have to perform the algorithms with the Main Cube in the Front Left Down Position, i.e. holding the cube with the Right hand but performing the algorithms mostly with the (L)eft hand.

Puppet v1 cube can be solved in 5 steps:

- **Step 1**: Give it a cube shape.
- **Step 2**: Swap Big Blocks.
- **Step 3**: Swap L-Shaped Bocks.
- **Step 4**: Place Edges in their own place.

- **Step 5**: Flip Edges.

Step 1: Give it a cube shape. The fist step consists in getting the puzzles in its original cube shape. This step is quite intuitive, and sometimes frustrating. Try to solve the first layer of the cube (where first layer is the one with the Main Block in it) as it was a classic $2x2x2$ cube. This is a bit more difficult with respect to the classic cube because many move are forbidden by the fact that pieces interfere. Focus on the main block and try to put next to it two any Big Blocks and finally an L-shaped one. Once you complete the first layer you will find the second layer in a specific configuration. If the puzzle has not a cube shape, try to substitute the L-shaped bock with another one if possible and play around with that configuration. If you cannot sort it out scramble a bit the cube and start again. With a bit of practice you will complete the first step in a few minutes.

The reason why completing the first layer gives you a good chance to complete this step is because there are very few configurations of the second layer and therefore there is a good chance that by completing the first layer you give a cubic shape to the puzzles. However, to shorten the time for completing this step, in the figure below I give you some configuration you may get in with the relevant algorithm to complete the step. If each configuration happened with the same probability (which is probably not true), knowing only one of the configurations in the figure below would statistically shorten the time required to complete the cube by half.

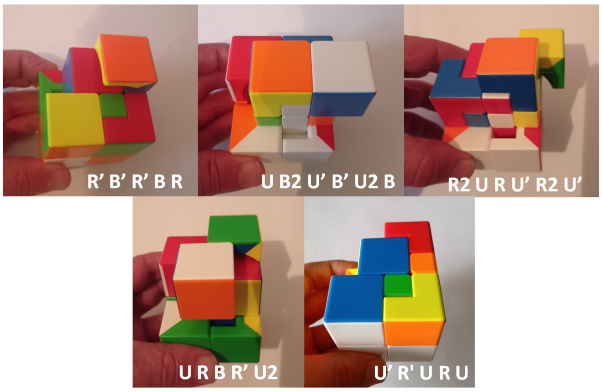


Figure 93: Algorithms for Step 1

All configurations in the figure above are shown with the Main Bock in the Left-Down-Front position and therefore all algorithms are given as (R) algorithms, which is cube to be hold with the left hand and algorithms to be performed with the (R)ight hand.

You Should also be able to recognize symmetric configurations to be solved with the relevant symmetric (L) algorithm (invert verse of all moves and change R moves in L moves) as shown in the pictures below. This will automatically double the number of configurations you can solve.

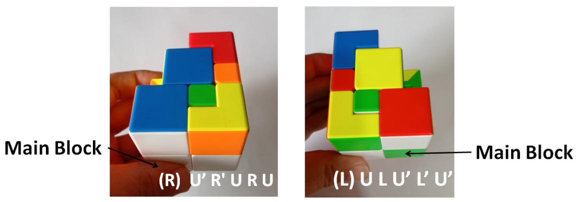
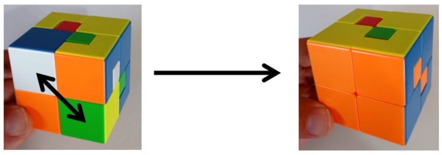


Figure 94: Symmetric Algorithm for Step 1

Step 2: Swap Big Blocks. Step 2 and step 3 can be done in either order because they do not affect each other. This step consists in placing the big blocks in they correct position. This can be done swapping Big Blocks two at a time with the algorithm of the figure below.



(R) U2 R U’ R’ U2 R’ U’ R U R’ U’ R’ B’ R2 B R

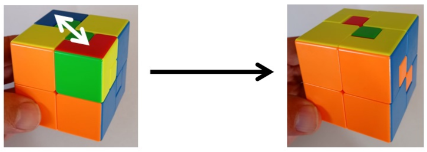
Figure 95: Swap Big Blocks

This algorithm is the classic one presented in tutorials. However, it is very long. I have wrote a code to look for better algorithms and I have found 2 algorithms (and their 2 reverse algorithms) that do the job in 10 moves. Pick the one you prefer. My code shows also that we cannot do better than that even thought I have not proven my code to be error free.

	Algorithm	Reverse Algorithm
1	R U2 R’ U R’ B’ U2 R2 U’ B’	B U R2 U2 B R U’ R U2 R’
2	B’ R’ U2 R2 B’ U’ R U’ R2 U	U’ R2 U R’ U B R2 U2 R B

Puppet Big Block Swap Algorithms

Step 3: Swap L-Shaped Blocks. Step 3 and step 2 can be done in either order because they do not affect each other. This step consists in placing the L-shaped blocks in they correct position. This can be done swapping L-Shaped Blocks two at a time with the algorithm of the figure below.



(R) R’ U R’ U’ R’ U R U’ R U2 R’ U’ R U R’ U R U B U R B’ R’ B’

Figure 96: Swap L-Shaped Bocks

This algorithm is the classic one presented in tutorials. However it is very long. I have wrote a code to look for better algorithms and I have found 2 algorithms (and their 2 reverse algorithms) that do the job in 11 moves. Pick the one you prefer. My code shows also that we cannot do better than that even thought I have not proven my code to be error free.

	Algorithm	Reverse Algorithm
1	U2 R B2 R’ B R B2 R2 U R U	U’ R’ U’ R2 B2 R’ B’ R B2 R’ U2
2	U2 B’ R2 B R’ B’ R2 B2 U’ B’ U’	U B U B2 R2 B R B’ R2 B U2

Puppet L-Shaped Block Swap Algorithms

Step 4: Place Edges in their own place. This step consists in placing each edge in its own position. To do that you need the 5 and 3 cycle algorithms reported in the two tables below. Note that the algorithms are reported with the W notation used as an index (meaning two layers are moved at the same time) to show what move is actually applied to the classic inner $3x3x3$ cube hidden inside the puzzle. Having this notation, you can practice the algorithms on a classic $3x3x3$ cube and see the effect on the edges, if you like.

In addition to algorithms you need U-perm tasks (i.e. sequences of algorithm) and Flip Edge Tasks. These are reported in Appendix A1 and A2. Tasks sometimes require to rotated the cube between algorithm. Axes conventions are given in the figure at the beginning of this section.

Solve edges means position them and flip them (next step) if they are uncorrected oriented. This is a long procedure and in this step and the next one I give my own method. However, given the cycle algorithm below, you can work out your own method if you like.

Edges are named with 2 letter corresponding to the the two faces of the inner and hidden classics cube they belong. I use the following convention to order the 2 letters: F and B come first and R and L come before U and D.

I consider a cube solved when hidden edges are also correctly placed and oriented although you cannot see them. However, you can choose your own criteria. I personally could not sleep at night if I knew that some hidden edges of my cube are not correctly oriented! At any time there are 3 hidden edges around the Main Block. You can see them by opening gently the space between the Main Block hidden face and the one of any of the 3 adjacent blocks.

The following table contains the 5-cycle and 3-cycle (R) algorithms:

	5-cycles and 3-cycles	Algorithm	Reverse Algorithm
1	(R) FU FR FD **BL** FL	$R_W\, U\, R'_W\, U'_W\, R'\, U'_W\, R'\, U_W\, R\, U_W$	$U'_W\, R'\, U'_W\, R\, U_W\, R\, U_W\, R_W\, U'\, R'_W$
2	(R) FU FR FD **BD** FL	$R_W\, U\, R_W\, U'_W\, R'\, U'_W\, R'\, U'_W\, R\, U_W$	$U'_W\, R'\, U_W\, R\, U_W\, R\, U_W\, R'_W\, U'\, R'_W$
3	(R) FU FL **BL** **RD** FD	$R_W\, U\, R_W\, U\, R_W\, U'_W\, R'_W\, U'_W\, R'_W\, U'_W$	$U_W\, R_W\, U_W\, R_W\, U_W\, R'_W\, U'\, R'_W\, U'\, R'_W$
4	(R) FR FL **BL**	$R_W\, U\, R_W 2\, U'_W\, F'\, U'_W\, L'_W\, B_W 2\, L_W\, F$	$F'\, L'_W\, B_W 2\, L_W\, U_W\, F\, U_W\, R_W 2\, U'\, R'_W$

Table puppet.1 : 5-cycle and 3-cycles (R) Algorithms

The following table contains the 5-cycle and 3-cycle (L) algorithms:

	5-cycles and 3-cycles	Algorithm	Reverse Algorithm
5	(L) FU FL FD **BR** FR	$L'_W\, U'\, L_W\, U_W\, L\, U_W\, L\, U'_W\, L'\, U'_W$	$U_W\, L\, U_W\, L'\, U'_W\, L'\, U'_W\, L'_W\, U\, L_W$
6	(L) FU FL FD **BD** FR	$L'_W\, U'\, L'_W\, U\, L_W\, U\, L_W\, U_W\, L'\, U'_W$	$U_W\, L\, U'_W\, L'_W\, U'\, L'_W\, U'\, L_W\, U\, L_W$
7	(L) FU FR **BR LD** FD	$L'_W\, U'\, L'_W\, U'\, L'_W\, U_W\, L_W\, U_W\, L_W\, U_W$	$U'_W\, L'_W\, U'_W\, L'_W\, U'_W\, L_W\, U\, L_W\, U\, L_W$
8	(L) FL FR **BR**	$L'_W\, U'\, L_W 2\, U_W\, F\, U_W\, R_W\, B_W 2\, R'_W\, F'$	$F\, R_W\, B_W 2\, R'_W\, U'_W\, F'\, U'_W\, L_W 2\, U\, L_W$

Table puppet.2 : 5-cycle and 3-cycles (L) Algorithms

For a start you need to choose your own main front face from which performing your algorithms in the (R) position. In standard colors you can choose between orange and white. I usually choose orange.

Sub-step 4.1: Given your front face you need to solve the edges which are not on the front nor on the back face. In the (R) position they are LD (hidden), RU and RD. Choose two edges that have a color in common and place the other one in its position. Take the remaining two edges to the common face they belong and place them in their place. For this last operation you may need Front Face U-perm tasks reported in appendix A1.

Sub-step 4.2: Position the edges on the Back Face using (R) algorithms only and choosing the algorithm that do not have effect on the blocks fixed during the previous sub-step.

Sub-step 4.3: The only edges left to solve are now the ones on the main front face. To solve them you need Front Face U-perm tasks reported in Appendix A1.

Step 5: Flip Edges. Now that all the edges are in pace, you will notice that some of them need to be flipped. I report the main Flip Edge tasks in Appendix A2.

There are always an even number of edges to be flipped. You can flip 2 or 4 at a time using the specific task. Before using flip edge tasks you need to place the edges to flip in the correct spots where the flipping task acts. You can do that with a set-up move made of several algorithms and or tasks. Keep in mind what set up move you did because you need to unmake it after the edges have been flipped reversing exactly the algorithms and tasks in the opposite order.

We want to show how the flip edge tasks work. We will focus on task 9 in Appendix A2. We have:

$$((FU\ FR\ FD\ BL\ FL)(FU\ FL\ BL\ RD\ FD))^2 = ((FU\ FR)(FD\ RD))^2 = \text{Null Permutation}$$

and every edge goes back to its original position. However, the final effect is that 4 edges get flipped.

11.2 Puppet Cube Version 2

The Puppet Cube Version 2 is a mechanical superset of the v1 where the Antipodal Clock (AP) is replaced by a smaller cube of half a size (as an edge) and one height the size as a volume. In this way there is a proper 3x3x3 cube inside the puzzle three faces of which can be seen from outside (in standard colors the orange, blue and yellow faces). It may be solved by reduction to the Puppet v1 but of course this is not the best way.

Puppet v2 cube can be solved in 5 steps:

- **Step 1**: Give it a cube shape (identical to Puppet v1 Step 1).
- **Step 2**: Swap Big Blocks (identical to Puppet v1 Step 2).
- **Step 3**: Swap L-Shaped Bocks (identical to Puppet v1 Step 3).
- **Step 4**: Make The Inner Cube White Cross.
- **Step 5**: Fix Remaining Edges.

Before we proceed we need to give names to different types of edges. In the Pupped v2 cube there are hidden edges (3 of them) which are inside of the cube and out of sight, outer edges (3 of them) which are edges of the inner cube which are also edges of the whole puzzle (the outer cube), and finally we have inner edges (6 of them) which are the ones on faces of the outer cube but not on its edges.

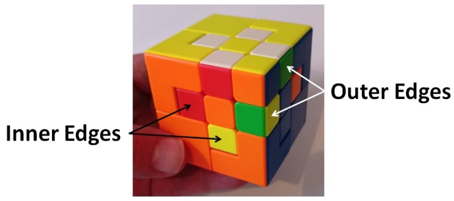


Figure 97: Outer and Inner Edges

Step 4: Make The Inner Cube White Cross. Remember that the corners of the inner cube are already fixed because they are part of the outer cube and centres are easy to fix and therefore you need to fix only edges. To do this, as for a classic cube, you need to do the white cross first. This is quite intuitive and one easy way to do it is making the white cross on the top face (yellow face) with opposite matching colors and then transfer it to the down face.

Note that centres can be placed in the correct position as a reference to place the edges of the cross, but since they can be fixed easily, they may be put in pace after the cross is done. However, at the end of this step you have also to ensure that centres are placed in their correct position which is the one matching the colors of external cube faces.

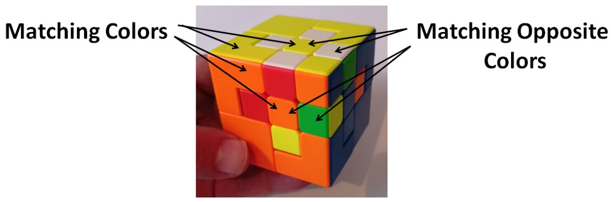


Figure 98: Make White Cross

Step 5: Fix Remaining Edges. Finally you need to fix the remaining edges and this can be done with the four algorithms presented in the table below. All algorithms are performed with the Main Block in the Down position and the cube oriented according to the task at hand. Note that in standard colours the main block is the white-green-red-block. This is different from the Puppet v1 where in standard colours the Main Bock was the white-green-orange block. You may try the algorithms below on a classic $3x3x3$ cube and see the effect on the edges.

Action	Algorithm
Clockwise	M2 U’ M U2 M’ U’ M2
Anticlockwise	M2 U M U2 M’ U M2
Clockwise Flipping FU and RU	M U M’ U2 M U M’
Anticlockwise Flipping FU and LU	M U’ M’ U2 M U’ M’

Puppet v2 - (RU FU LU) Edge Permutation Algorithms (M: Middle Layer positive verse as L)

Fore example, the first algorithm of the above table has the effect shown in the figure below.

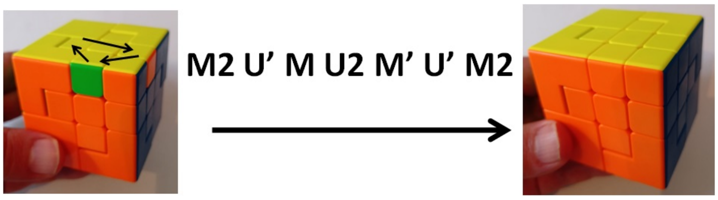


Figure 99: Edge Permutation

Fix remaining edges its all about fix one edge at a time by swapping the wrong edge in the correct position and the correct edge in the wrong position. This can be done by making set-up moves of the medium layers of the inner cube in order to to take the two edges to swap on the same face and than apply one of the four algorithms above, with flipping or without in order to pace the correct edge with the correct orientation. The Idea is to cycle three edges, the two edges we want to swap and a third one among the ones not fixed yet.

Once the white cross is done, there will be one inner edge not fixed yet (in standard colors the green and red edge). This is the first we want to sort out by bringing it to one of the faces with a set up move and performing the swap. After that, we fix the inner edges one at a time. We can always permuting the swap by cycling the two edges at a time and one of the outer edges not fixed yet.

When the inner edges are done, we will be left with the three outer edges to fix. We can bring them to the same face by inner cube middle layers set up moves and we cycle them. Finally, ff we are left with two edges in the correct position, two of which flipped, we can sort this out by cycling and flipping the edges and than cycle them back without flipping.

12 Bandaged Cubes

Bandaged cubes are standard cubes where some faces are blocked to stay together like there was a bandage. This reduces the number of possible moves and make the task of solving the cube more difficult.

When it comes to bandaged cube the possibility are endless an cubes have in general no standard names. As far as $3x3x3$ bandages cubes are concerned, in this tutorial we well consider only three types shown in the figure below and we will name them as type A, B and C. These names, although sometimes are used on the internet, are no standard names at all.

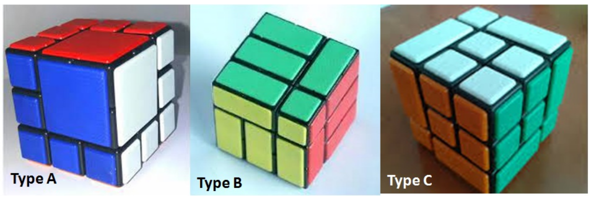


Figure 100: $3x3x3$ Bandaged Cubes

12.1 AI Bandaged Cube

Figure 101: AI Bandaged

The AI Bandaged cube is made from a standard $4x4x4$ cube where some faces are bandaged. In most of the configurations of the cube the possible moves are the same of a $2x2x2$ cube. However, when the big blocks are arranged in one layer, the cube become a 3 layer cube with one tick layer and two thin ones. With the cube in this configuration it is possible to solve the little faces which is the difficult task of this cube. To do that we have two possible things we can do. We can turn the two up thin layers and we can rotate the up $2x2x2$ composite corners. This can be done with algorithm bandaged(1) which we will notate with a P'.

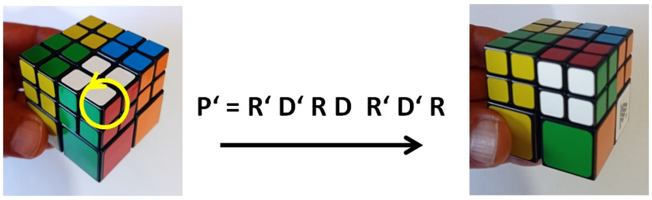


Figure 102: bandaged(1) $= P' : R'\ D'\ R\ D\ R'\ D'\ R$

Note that algorithm P' is a double Sledgehammer where we do not perform the last D move because we do not care about the configuration of the down layer and it is the very same algorithm of classic(6) above. The P' move is an anticlockwise rotation. The clockwise notation will be noted with a P or a $P'2$ since after 3 rotation the corner gets to the original orientation and you can perform a P move by performing 2 P' moves in sequence unless you want to memorize the inverse of algorithm bandaged(1), which by the way it is an easy thing to do.

Note also that the cube cannot be properly scrambled if you do not go, time to time, in the 3 layer configuration because the only move we can do in the other configurations are the one of a classic $2x2x2$ cube and therefore after the scramble the cube can be solved as a $2x2x2$ cube.

Solution: The AI Bandages Cube can be solved in 6 steps:

- **Step 1**: Get to the 3 layer configuration.
- **Step 2**: Solve one composite corner.
- **Step 3**: Solve a composite cornet next to the previous one.
- **Step 4**: Solve Edges of remaining composite corners.
- **Step 5**: Permute little centres.
- **Step 6**: Solve as a "2x2x2".

Step 1: *Get to the 3 layer configuration* - This can be done using the algorithms of a $2x2x2$ classic cube.

Step 2: *Solve one composite corner* - Join the 4 pieces of the top layer. This can be done intuitively rotating the two top layer and rotation corners. You can join the 4 pieces 2 by 2 in two separate corners and then put them together. Save the four pieces configuration on the top layer by rotation it.

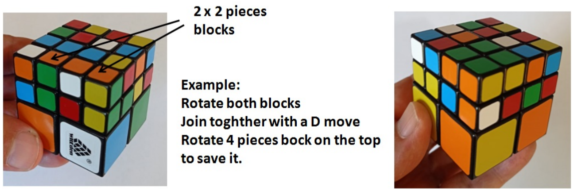


Figure 103: 4 Pieces Up Layer

Join the 3 pieces of the middle layer on a separate block. Put the 2 layer together.

Figure 104: First Corner Completed

Step 3: *Solve a composite cornet next to the previous one* - For this step the most important thing to remember is that you do not want to scramble the first corner you did. Every time you move one of the two up layers to perform a task, you have to remember to undo the move at the end to preserve the first corner. For the rest you proceed as for the previous step. You make the up layer of the corner on the right hand side next to the previous one you did. If you make a different corner, remember that you can always swap corners using algorithm classic(4) applicable also to $2x2x2$ cubes. This algorithm permutes 3 corners and therefore it allows also to swap two corners. For the middle layer, the easiest way to do it is to put only a little centre pierce on the face of opposite color on the corner as shown in the figure below:

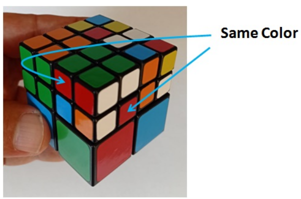


Figure 105: Configuration to Solve the Middle Layer

and to join the two remaining pieces in the corner next to it. At this point, you can complete the corner using algorithm bandaged(2).

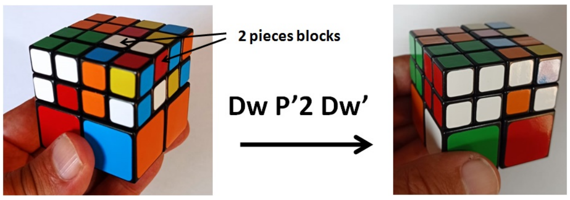


Figure 106: bandaged(2): $D_w\, P'2\, D'_w$

where with D_w we mean we are rotating both the thick down layer and the thin middle layer.

Step 4: *Solve Edges of remaining composite corners* - This step consists in f solving the little edges in the two renaming compost corners. As for the step before you do not want to scramble the two corners you fixed before. In order to do to do that, every time you move one of the two top layers, you have to remember to undo the move at the end to preserve them. To solve edges, you match a corner and a edge by moving the top layers, and you rotate the corner to save the match you have just done.

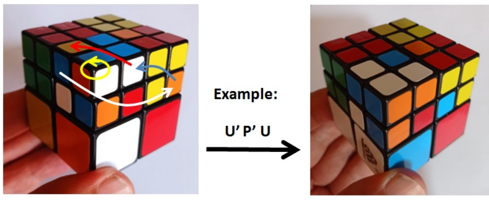


Figure 107: Permute 3 Edges

All the simple moves described above will permute 3 edges. You may end up in a situation where two edges are left to solve. To solve this case we need to introduce algorithm bandaged(3) to swap two corners of the $2x2x2$ cube. This algorithm is equivalent to algorithm classic(4) but it does not rotate the corners, which is what we want. We will notate this algorithm with a Q.

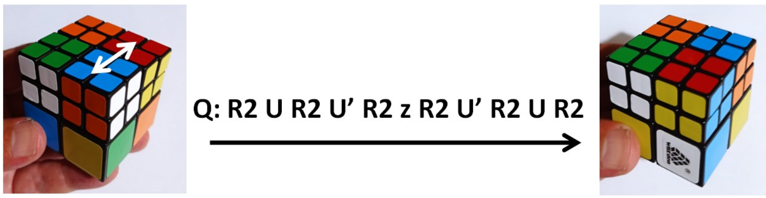


Figure 108: bandaged(3) $= Q : R2\, U\, R2\, U'\, R2\, z\, R2\, U'\, R2\, U\, R2$

To swap two edges, place the two edges to solve on the middle layer and use algorithm bandaged(4).

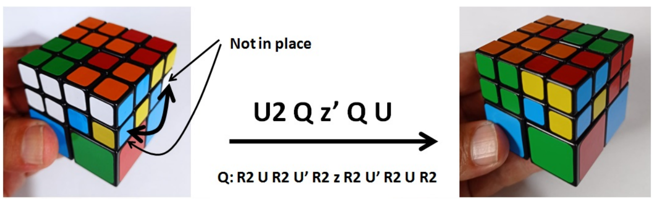


Figure 109: bandaged(4): $U2\, Q\, z'\, Q\, U$

At the end of the above algorithm, the two original solved corners are not next to each other any more. To place them next to each other, you can use algorithm Q again or use algorithms standard(4). The result, up to rotation of corners, is the same.

Step 5: *Permute little centres* - You can complete the two renaming composite cubes by using algorithm bandaged(5) which permutes 3 centres. You can rotate corners if needed before permuting centres. Since the algorithm permute two centres on the left hand side cube and one on on the right hand side, to complete the cube you may want to swap two composite corners you are working on. You can do it using the Q algorithm or algorithm classic(4).

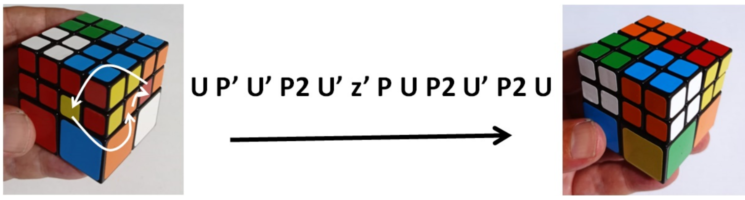


Figure 110: bandaged(5) $U\,P'\,U'\,P2\,U'\,z'\,P\,U\,P2\,U'\,P2\,U$

Step 6: *Solve as a "2x2x2"* - Once the have solved the for composite corners, you can complete the cube solving it as a $2x2x2$ classic cube.

12.2 Pocket Cube

Although the Pocket cube looks very similar to Puppet cubes it is mechanically equivalent to the Bandaged cube Type C and it can be solved with the same algorithms. This is the reason why it has been inserted in this section.

Figure 111: Pocket Cube

Solution: All algorithm to solve the Pocket cube have to be performed with the composite 2x2x2 corner (if present) in the up-back-left position. The Pocket Cube can be solved in 5 steps:

- **Step 1**: Give it a cube shape.
- **Step 2**: Fix bars and corners.
- **Step 3**: Swap little edges.
- **Step 4**: Orient little edges.
- **Step 5**: Orient the little corner.

Step 1: *Give it a cube shape* - This step is identical to the same step for puppet cubes. You can have a look at the description of step 1 for puppet cubes as a reference. This part has to be done intuitively till we get to a known configuration. We give below a few configurations. they may seams not very helpful but they are especially designed to help for the Bandaged cube Type C which will be addressed later on. Moreover this step for the Pocket cube is easer that the relevant one for the Puppet cube because the number of possible moves and configuration is limited. In the configurations below, note that some of the big corners correspond actually to bars (i.e. long edges), the little corner and the big corner of the Bandaged Cube Type C. Taking into account that the Bandaged Cube Type C never looses its original shape, try to identify them.

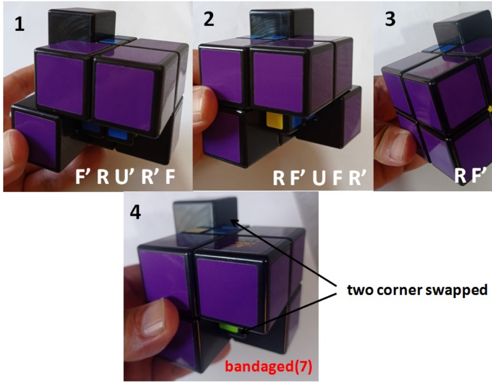


Figure 112: Step 1 Configurations

Note that the algorithm applied to configuration 3 will take the cube to configuration 4 which is the configuration where the two corners of the bandaged cube Type C are swapped. The algorithm to solve this configuration is bandaged(7) given further on in this section.

Step 2: *Fix bars and corners* - [**NOTE: this step is not correct and it needs to be amended. However, this step is not essential to solve the cube.**] The original configuration of the Pocket cube is the one with the two logos and the yellow face of the compost cube on the same face.

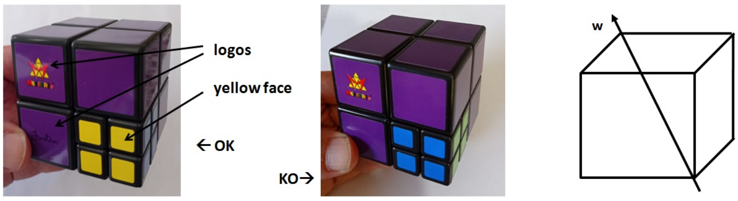


Figure 113: Correct final configuration

In the above figure we define also an axis w that we will need in the next algorithm. If you want the cube to have the original factory configuration, you need to fix the corners position that correspond to bars (i.e. long edges) in the Bandaged cube Type C with the position of the Composite corner and its opposite. To fix the bars we need algorithm bandaged(6).

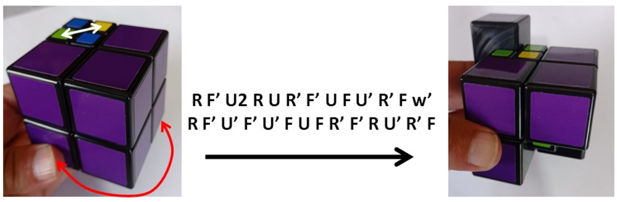


Figure 114: Algorithm bandaged(6)

$$bandaged(6):\ R\,F'\,U2\,R\,U\,R'\,F'\,U\,F\,U'\,R'\,F\,w'\,R\,F'\,U'\,F'\,U'\,F\,U\,F\,R'\,F'\,R\,U'\,R'\,F$$

where with w' we mean a rotation of the cube that take the right face of the cube on the front. If you want the bars properly placed, you apply algorithm bandaged(6) for a few times and twice at a time till you have the two logos and the yellow centre of the composite cube on the same face. What the algorithm does is to rotate clockwise the composite corner and the corner opposite to it and swap the bars on the down face. You may need to apply this algorithm quite a few time before all matches.

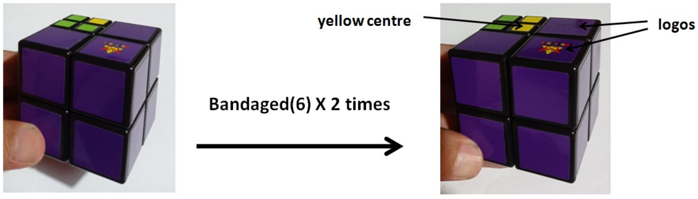


Figure 115: Alghoritm bandaged(6) x 2 times

Since all the corners of the Pocket cube have the same color, you may decide to ignore this step if you do not mind where the logos are.

Step 3: *Swap little edges* - This step consists in placing the little edges in their correct position. This can be done using algorithm bandaged(7), which swaps corner positions but it has the side effect to swap also the little corner and its opposite. For this reason you will need to use algorithm bandaged(7) an even number of times. Note that algorithms bandaged(6) and bandaged(7) do the same job but algorithm bandaged(7) also swaps two bars and therefore it cannot be used after you fixed the bars in the previous step.

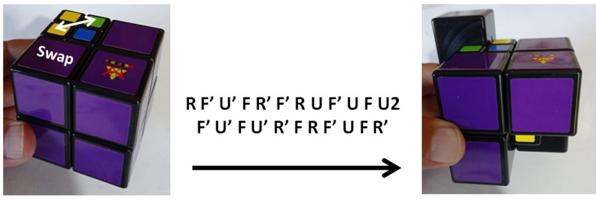


Figure 116: Algorithm bandaged(7)

$$bandaged(7) : \; R\, F'\, U'\, F\, R'\, F'\, R\, U\, F'\, U\, F\, U2\, F'\, U'\, F\, U'\, R'\, F\, R\, F'\, U\, F\, R'$$

Step 4: *Orient little edges* - This step consist in flipping two lettle edges at a time till all edges are correctly oriented. This can be done with algorithm bandaged(8).

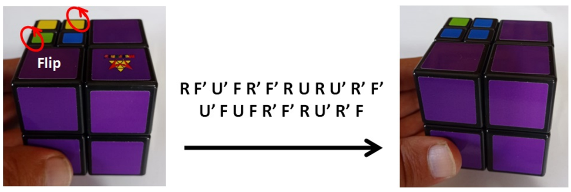


Figure 117: Algorithm bandaged(8)

$$bandaged(8) : \; R\, F'\, U'\, F\, R'\, F'\, R\, U\, R\, U'\, R'\, F'\, U'\, F\, U\, F\, R'\, F'\, R\, U'\, R'\, F$$

Step 5: *Orient the little corner* - When you get to this step all faces are in place apart from the ones of the little corner that in general will need to be oriented. This can be done applying algorithm bandaged(8) an even number of times on the same face till the cube is solved. Applying algorithm bandaged(8) twice, edges on that face will be flipped twice and they will go to the original position but the little corner will rotate with respect to the w axis.

12.3 Camouflage Cube $3x3x3$

This cube is mechanically equivalent to the AI Bandaged cube and it can be solved using the same algorithms. For this reason it has been inserted in this section.

NOTE: I do not own this cube. The picture comes from the internet as well as the information that this puzzle is mechanically equivalent to the AI Bandaged cube. All this will be confirmed in the next issue of this tutorial once I have bought one of this puzzles.

Figure 118: Camouflage Cube

TBD

12.4 Bandaged Cube Type A

TBD

12.5 Bandaged Cube Type B

TBD

12.6 Bandaged Cube Type C

TBD

13 The Redi Family

These are very easy puzzles where you can turn corners together with the edges next to them. This puzzles are easy because every possible move has a local effect and it does not affect the configuration of pieces far away.

13.1 Redi Cube and Redi Barrel

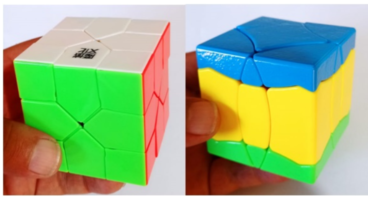


Figure 119: Redi and Redi Barrel

TBD

13.2 Redimix

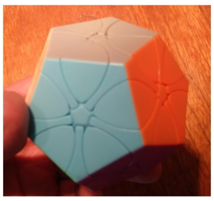


Figure 120: Redimix

TBD

14 Curvy Copter and Helicopter Puzzles

This section contains solutions for the Curvy Copter and Helicopter Puzzles.

14.1 Curvy Copter

This is a nice puzzle with a mechanics which is quite different from all other puzzles present in this notes.

Figure 121: Curvy Copter

The Curvy Copter has Edges Corners and centre pieces (they are actually outer centre pieces). You can easily tell which one is which.

Notation: Being a edge rotating puzzles, the notation for the Curvy copter refers to rotation of edges as in the picture below. standard rotation for edges is of 180 degs and therefore no number after the edge letter is needed in the notation. We will see that it is possible to have also rotation of edges by an angle less then 180 deg. This moves will be notated putting the letter in brackets. We will call moves without brackets, regular moves.

In addition to the above notation we will use also the notation J_R for the Right Jumbling Move and J_L for the Left Jumbling Move, both described below.

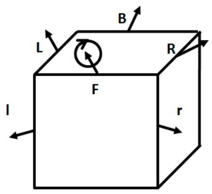


Figure 122: Curvy Copter Notation

Orbits and the Jumbling Move: By just using regulars moves only (i.e. 180 degs move, which are notated without a tilde) it is possible to move centre pieces around the puzzles but only in their own orbits. On the Curvy Cube there are four separate orbits one of which is shown in the picture below:

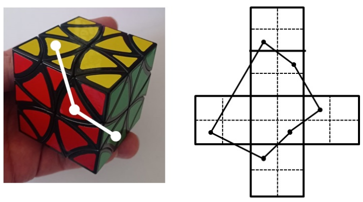


Figure 123: Curvy Copter Orbits

In order to move a centre pieces from one orbit to the other, you need algorithm copter(1) which is called the Jumbling move. This move swaps two couples of centres, two on the same orbit and two on two separate orbits (the centres on the opposite left and right faces). We can use the Jumbling move to make these latter centres to jump from one orbit to another.

Figure 124: copter(1): **(R) (l) F (l') (R')** - Right Jumbling Move

The above algorithm is the J_R (Right Jumbling Move). In order to solve the puzzle we need also its symmetric, which is the J_L (Left Jumbling Move). Both versions of this algorithm are presented below:

$$\begin{cases} J_R: & (R)\,(l)\,F\,(l')\,(R') \\ J_L: & (L')\,(r')\,F\,(r)\,(L) \end{cases}$$

For example, the Jumbling move can be used to swap two centres on the same face. To do that, we need to take the two pieces on opposite faces with two regular moves, apply the Jumbling move, undo the two regular move. The procedure is shown in the picture below:

Figure 125: copter(2): $L\ R\ J_R\ L\ R$

Solution: The curvy Copter puzzles can be solved in 7 steps:

- **Step 1**: Make a Cube Shape.
- **Step 2**: Solve white Centres.
- **Step 3**: Solve pieces on the white side of the cube.
- **Step 4**: Solve Remaining Centres.
- **Step 5**: Orient Edges on yellow Face.
- **Step 6**: Permute Corners.
- **Step 7**: Orient Corners.

Step 1: *Make a Cube Shape* - When the Curvy Copter is scrambled, it does in general lose its original cube shape.

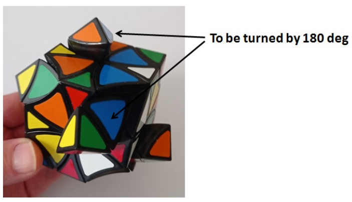


Figure 126: Scrambled Curvy Copter

We need to return the puzzle to its originate shape. This can be done with algorithm copter(3) that fixes two corners at a time. In a scrambled cube it is always possible two find two non oriented corners that can be paired and solved with this algorithm.

When the cube is a lot out of shape, some moves are prevented by the fact that some pieces have a bit of interference. However, if you force the cube you will be able to perform those moves anyway. These moves are illegal and you should be able to solve the cube using only legal moves. Sometimes, when the cube is completely messed up, it is difficult to find legal moves though. I will not blame you if sometimes you decide to use them.

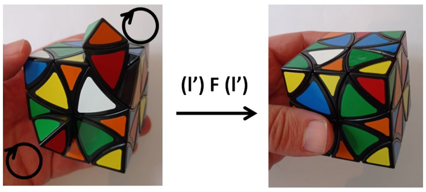


Figure 127: copter(3): **(l') F (l')**

Step 2: *Solve white Centres* - In this step you solve the 4 white centres (or whatever color you chose for your first face). Edges can rotate but cannot be moved from their own position and therefore they give a reference for the color of faces.

Identify the white face which is the one where each edge has one white side. Place as many white centres on that face. This can be done intuitively. Do it in such a way edges are oriented correctly with the white side on the white face. If a centre cannot be solved, it means that it is not on the correct orbit (all four orbits pass on each face). Put the centre to be solved on the correct orbit using the Jumbling move (more then once if needed).

Step 3: *Solve pieces on the white side of the cube* - In this step you need to solve the corners on the white face, the edges on the lateral face and the two centres on the lateral faces next to the white face.

For each corner of the white face, find it on the cube and put it in place taking care that the relevant edge on the lateral faces is oriented correctly. To orient the corner correctly you can make it going around the yellow face. Each round will torn the corner by 120deg clockwise or anti-clockwise according to the direction you make it travel around the yellow face.

Place the two centres using algorithm copter(4). If you cannot place a centre in the correct spot to apply the algorithm it means that it is not in the correct orbit. Use Jumbling moves to place it in the correct orbit (more then once if necessary.

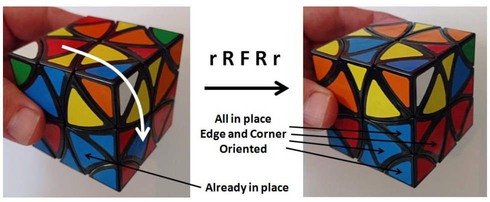


Figure 128: copter(4): **r R F R r**

Step 4: *Solve Remaining Centres* - In this step you solve the remaining centres without bothering about the orientation on the edges on the yellow face. To do that, for each centre on the lateral face, put it in its correct orbit using the Jumbling move (more then once if needed). Once lateral centres are in the correct orbit, the yellow centres will be in the correct orbit too. At this point they can be placed in their correct position easily and intuitively.

Parity: Sometimes you cannot complete this step because two yellow centres are swapped (and you cannot tell because they are of the same colour). This is a parity that can be solved with algorithm copter(2):

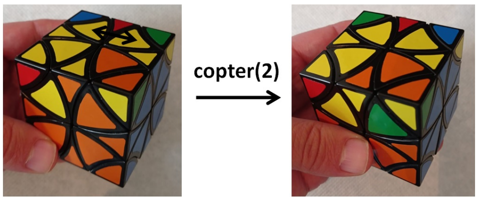


Figure 129: Fix Parity with Algorithm copter(2)

And from this point the step can be easily completed.

Step 5: *Orient Edges on yellow face* - Use algorithm copter(5) to orient edges of the yellow face:

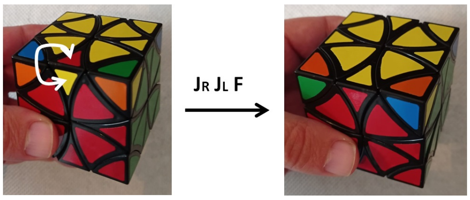


Figure 130: copter(5): J_R J_L F

Step 6: *Permute Corners* - Use algorithm copter(6) to permute corners of the yellow face:

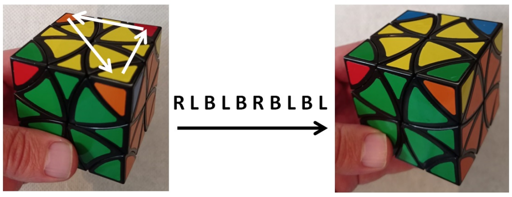


Figure 131: copter(6): **R L B L B R B L B L**

Step 7: *Orient Corners* - Use algorithm copter(7) to orient corners of the yellow face:

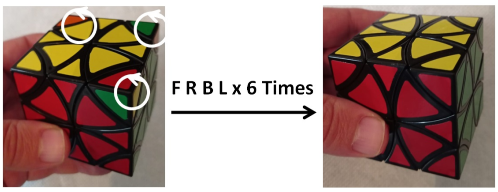


Figure 132: copter(7): **F R B L** repeat 6 times

14.2 Helicopter Cube

Figure 133: Helicopter Cube

The Helicopter cube is a mechanical subset of the Curvy copter puzzle where the edges are missing. It can be solved with the same method of the Curvy Copter. The main difference is that you have no reference for the white face and therefore you can choose any face to start the cube and you never need to bother about edges orientation (since they do not exist).

Another thing to note about the Helicopter cube is that when you get to Step 7 (orient corners) you may end up in a situation in which two corners are correctly oriented and two are not. You can see this as a parity and it is due to the fact that one of the ”non existing” edges is not correctly oriented. You can fix this easily using algorithms copter(5) to flip any of the ”non existing” edges on the yellow face end start again from step 6 to the end.

A simpler way to solve this is to have the face of the cube such that the two non oriented corner are on the front face and apply half algorithm copter(7) (i.e. (F R B L) x 3 times) till the yellow face is completed and only one corner is correctly oriented (it will be an odd number of times). From there you can complete the cube using Step 7.

15 Siamese Cubes

Siamese cubes are cubes joint edge to edge. This reduces the possible moves available to solve each cube.

15.1 Siamese Cube - Two Cube Version

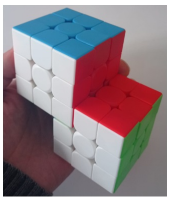


Figure 134: Siamese Cube - Two Cube Version

TBD

16 Mirror and Ghost Puzzles

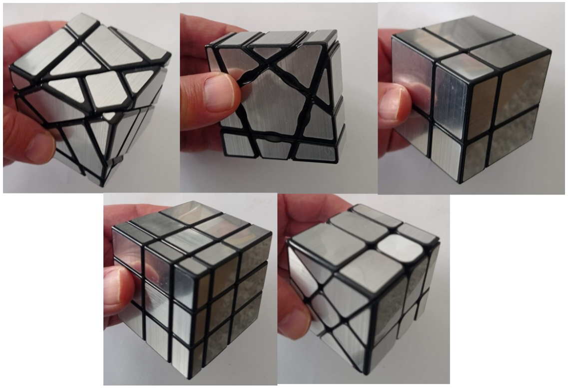


Figure 135: Mirror and Ghost Puzzles

There is no much to say about Mirror and Ghost Puzzles. At the beginning they may seem difficult but if you can do the coloured version, you can do it. The cube on the up left corner of the picture above is the Ghost 3x3x3 cube which is a mirror version of the Axel Cube. There are also bigger size versions of it (e.g. the 4x4x4). The Ghost Cube may be a real challenge and my advise is to choose your down layer (i.e. the equivalent down layer of the classic cube), identify the central piece of it and take a picture of the edges of the middle layer (which are difficult to tell each other apart) before you scramble it the first time.

Appendix

A.1 Puppet Front Face U-perm Tasks

Tasks in this section are designed to be performed using the minimum number of reverse algorithms. This in case you want to memorize the algorithms and you do not want to memorize all the reverse ones. For example, task 2 may have been implemented just reversing all algorithms of task 1 from the last to the first, but we made a different choice.

	Task	Implementation	Algorithms
1	(R) FR FU FL	(R) FU FL **BL** **RD** FD	R U R U R U’ R’ U’ R’ U’
		(R) FR FL **BL**	R U R2 U’ F’ U’ L’ B2 L F
		(R) Rev(FU FL **BL** **RD** FD)	U R U R U R’ U’ R’ U’ R’
2	(R) FL FU FR	(R) FU FL **BL** **RD** FD	R U R U R U’ R’ U’ R’ U’
		z - (L) FL FR **BR** - z’	z - L’ U’ L2 U F U R B2 R’ F’ - z’
		(R) Rev(FU FL **BL** **RD** FD)	U R U R U R’ U’ R’ U’ R’

Table puppet.A1 : Front Face Up U-Perm Tasks - (R) Position

	Task	Implementation	Algorithms
3	(R) FR FD FL	z’ x’ - (R) FR FL **BL** - x z	z’ x’ - R U R2 U’ F’ U’ L’ B2 L F - x z
		(R) FR FU FL (task 1)	R U R U R U’ R’ U’ R’ U’
			R U R2 U’ F’ U’ L’ B2 L F
			U R U R U R’ U’ R’ U’ R’
		y’ - (L) FL FR **BR** - y	y’ - L’ U’ L2 U F U R B2 R’ F’ - y
4	(R) FL FD FR	z’ x’ - (R) FR FL **BL** - x z	z’ x’ - R U R2 U’ F’ U’ L’ B2 L F - x z
		(R) FL FU FR (task 2)	R U R U R U’ R’ U’ R’ U’
			R U R2 U’ F’ U’ L’ B2 L F
			U R U R U R’ U’ R’ U’ R’
		y’ - (L) FL FR **BR** - y	y’ - L’ U’ L2 U F U R B2 R’ F’ - y

Table puppet.A2 : Front Face Up U-Perm Tasks - (R) Position

Tasks in this page (i.e. (L) U-perms) may be useful if you want to implement my method changing it for left handed people or to make a new method on your own.

	Task	Implementation	Algorithms
5	(L) FL FU FR	(L) FU FR **BR LD** FD (L) FL FR **BR** (L) Rev(FU FR **BR LD** FD)	L' U' L' U' L' U L U L U L' U' L2 U F U R B2 R' F' U' L' U' L' U' L U L U L
6	(L) FR FU FL	(L) FU FR **BR LD** FD z' - (R) FR FL **BL** - z (L) Rev(FU FR **BR LD** FD)	L' U' L' U' L' U L U L U z' - R U R2 U' F' U' L' B2 L F - z U' L' U' L' U' L U L U L

Table puppet.A3 : Front Face Up U-Perm Tasks - (L) Position

	Task	Implementation	Algorithms
7	(L) FL FD FR	z x' - (L) FL FR **BR** - x z' (L) FL FU FR (task 5) y - (R) FR FL **BL** - y'	z x' - L' U' L2 U F U R B2 R' F' - x z' L' U' L' U' L' U L U L U L' U' L2 U F U R B2 R' F' U' L' U' L' U' L U L U L y - R U R2 U' F' U' L' B2 L F - y'
8	(L) FR FD FL	z x' - (L) FL FR **BR** - x z' (L) FR FU FL (task 6) y - (R) FR FL **BL** - y'	z x' - L' U' L2 U F U R B2 R' F' - x z' L' U' L' U' L' U L U L U z' - R U R2 U' F' U' L' B2 L F - z U' L' U' L' U' L U L U L y - R U R2 U' F' U' L' B2 L F - y'

Table puppet.A4 : Front Face Up U-Perm Tasks - (L) Position

A.2 Puppet Flip Edge Tasks

This section contains flip edge tasks.

	Task	Implementation	Algorithms
9	(R) Flip FU FR FD **RD**	[(R) FU FR FD **BL** FL (R) FU FL **BL** **RD** FD] ×2	[R U R' U' R' U' R' U R U R U R U R U' R' U' R' U'] ×2
10	(R) Flip **BL** **RD** FD FR	[(R) FU FL **BL** **RD** FD (R) FU FR FD **BL** FL] ×2	[R U R U R U' R' U' R' U' R U R' U' R' U' R' U R U] ×2

Table puppet.A5 : 4-Edge Flip Tasks - (R) Position

	Task	Implementation	Algorithms
11	(L) Flip FU FL FD **LD**	[(L) FU FL FD **BR** FR (L) FU FR **BR** **LD** FD] ×2	[L' U' L U L U L U' L' U' L' U' L' U' L' U L U L U] ×2
12	(L) Flip **BR** **LD** FD FL	[(L) FU FR **BR** **LD** FD (L) FU FL FD **BR** FR] ×2	[L' U' L' U' L' U L U L U L' U' L U L U L U' L' U'] ×2

Table puppet.A6 : 4-Edge Flip Tasks - (L) Position

	Task	Implementation	Algorithms
13	(R) flip FU **BL**	(R) flip FU FR FD **RD** (task 9) (R) flip **BL** **RD** FD FR (task 10)	[R U R' U' R' U' R' U R U R U R U R U' R' U' R' U'] ×2 [R U R U R U' R' U' R' U' R U R' U' R' U' R' U R U] ×2
14	(L) flip FU **BR**	(L) flip FU FL FD **LD** (task 11) (L) flip **BR** **LD** FD FL (task 12)	[L' U' L U L U L U' L' U' L' U' L' U' L' U L U L U] ×2 [L' U' L' U' L' U L U L U L' U' L U L U L U' L' U'] ×2

Table puppet.A7 : 2-Edge Flip Tasks