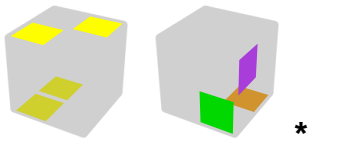
2x2 1-Look

Assume first face is yellow. Most L/R mirror cases are not included. Left image shows the case, right shows prediction by the suggested algorithm. First predict OLL: track any 3 white stickers; Then predict EG: track any 3 non-white stickers on any 3 white pieces. For the cases where all the yellow pieces are in first layer, one can do U move first to make better EG.



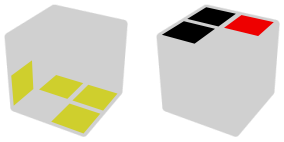
R U R' U R2 **y2 z (+, F)**

Alternative algorithms

Alg of left image **R U R' U R2** moves purple, green, orange pieces in right image by **y2 z**, **y2 z +**, **y2 z F**. Notation: fixed piece, piece by pure rotating, twist piece, piece by 1 or 2 moves, piece by 2 moves. case with 3 relatively fixed piece [8], same case*. + = CW, - = CCW.

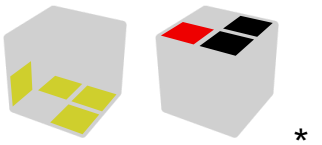
Bar [36]

LS (Last Slot) [5]



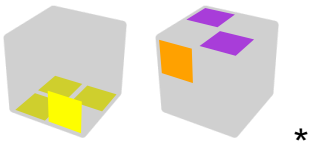
F' R' F R **U**

= y L' F' L F = y2 z F' R' U R

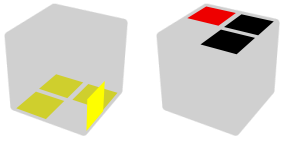


R U R' F' **U**

= y' F R F' R' = y2 z U R U' F'

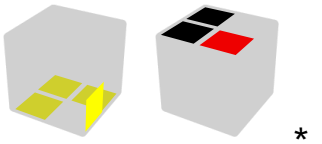


R' U R U R' **y F R**



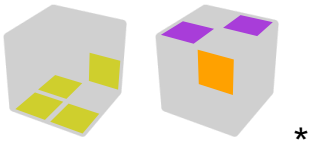
F R U' R' **U'**

= y' R F R' F' = y2 z' F L U' L'

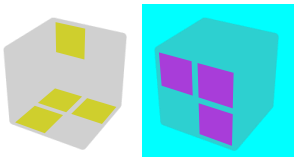


R' F' R F **U'**

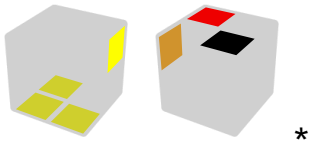
= y F' L' F L = y2 z' U' L' U F



R U' R' U' R **y' R' F'**

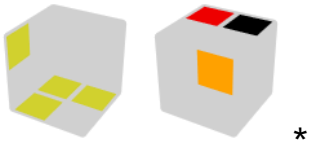


R U2' R' **z2 x**



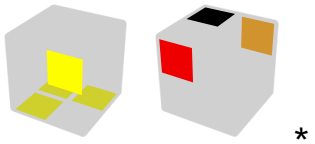
R' U' R **U', U' R**

= y' F' L' F = y2 R' F' R



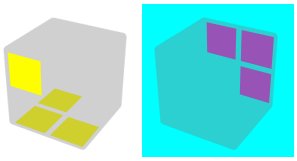
R U' R' **U', R U'**

= y' F R' F'

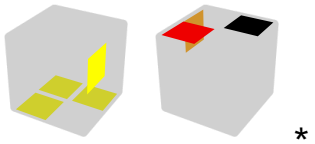


R' F R F' **F R, R' U**

= y F' R U R'

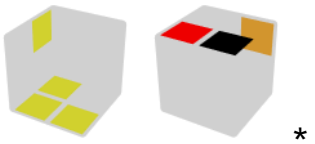


R' U2 R **z2 x'**



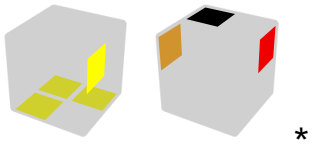
R U R' **U, U R'**

= y' F R F'



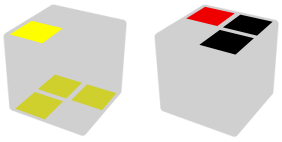
R' U R **U, R' U**

= y' F' L F = y2 R' F R

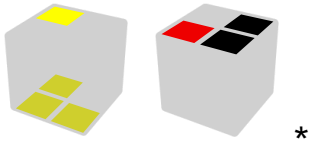


F R' F' R **R' F', F U'**

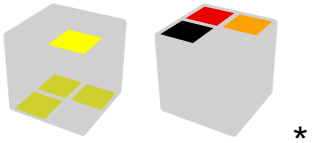
= y R U' R' F



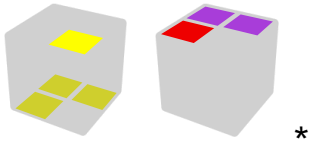
R2 U' R2' **U'**



R2 U R2' **U**

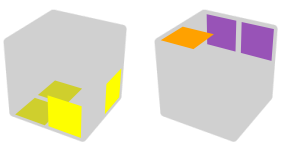


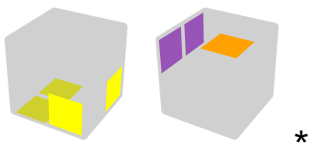
R2 U R2'
U' R2 **U, U'**

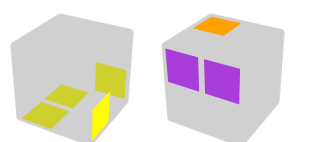


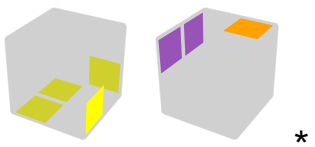
R2 U R2'
U' F2 R2 **y' U'**

DD [4]

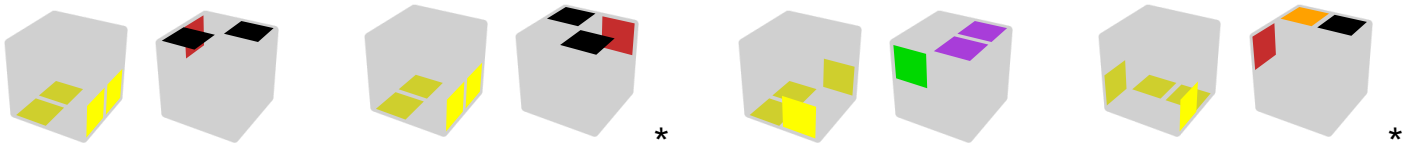








R' U2 R U' R' | z' x' D' R' | R' U' R U2' R' | z y D' B' | R U2' R' U R | z' x D R | R U R' U2' R | z y' D F

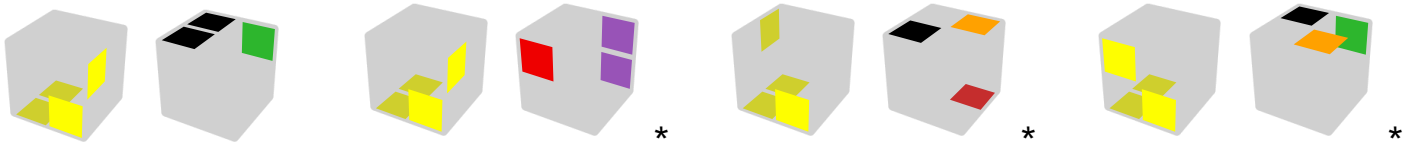


R' U R2 U' R | U R' | R U' R2' U R' | R' U | R U2' R2' U R2 | y2 - | F R2 U' R2' | F, U'

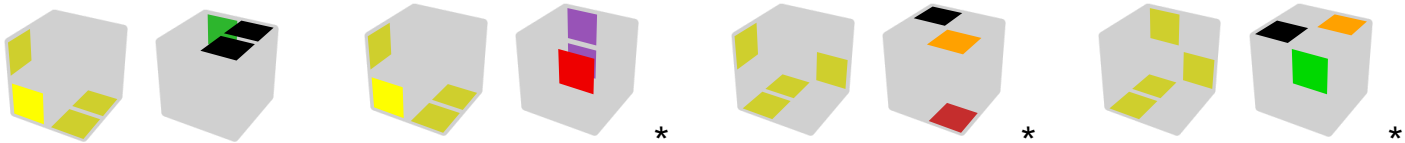
z ->other case

UD - 1 move D [6]

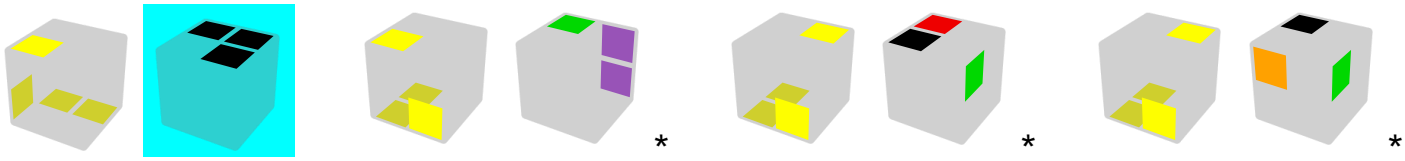
R' | R' | R | R



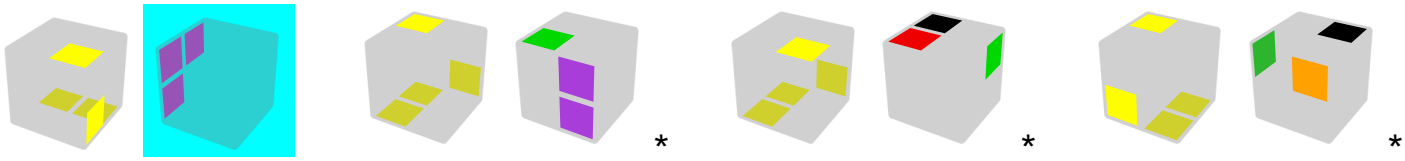
R2' F' R F | - | R' F R U' R' | x' L2 | R U R' U' R2 | R2, U' | R' U' R U R' | -, U



L2 F L' F' | + | L F' L' U L | x' R2 | R' U' R U R2' | R2, U | R U R' U' R | +, U'

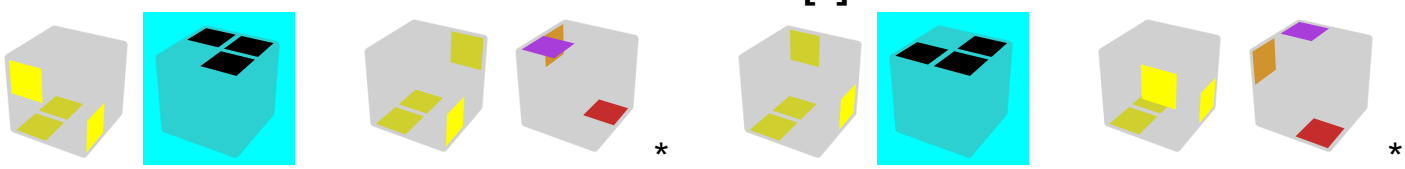


R2 U' R' U R' F' | R U' R2' | x' - | R' U R U' R' | -, U | R2' F R F' | -, F R



R U R' U' F | z | R' U R2 | x + | R U' R' U R | +, U' | L2 F' L' F | +, F' L'

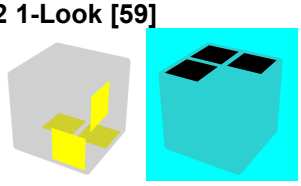
UD - 3 move D [6]



F R U' R U R2' | R2 U R' | y (F2, R') | B' R' U R' U' R2 | R2' U' R | y' (B2, R)

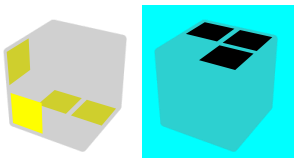
U z y ->other case

U' z y' ->other case



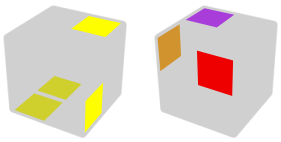
R' D R

= y F' R F



L D' L'

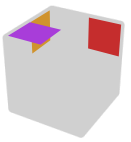
= y' F U' R'



R U' R

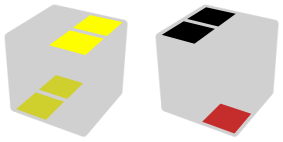
y' (B, R)

R' U R'



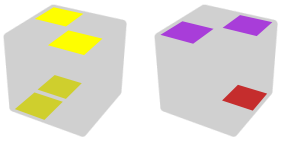
y (F', R')

UU - 2 Up [2]



R2

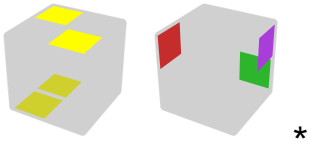
R2



R2 U' R2' U' R2

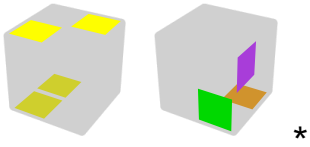
U' R2 U R2' U R2

y' B2



R' U' R U' R2'

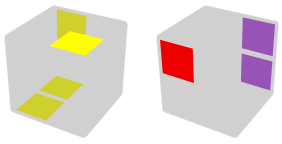
y2 z (-, B2)



R U R' U R2

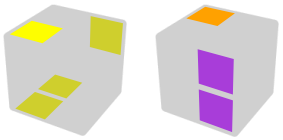
y2 z (+, F)

UU - 1 Up [6]



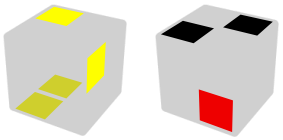
R' U2 R'

z2 x' R2



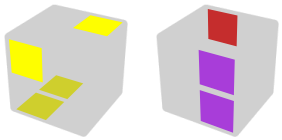
R' U' R2

x L U'



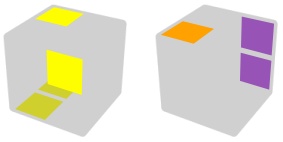
R' U R U' R

R



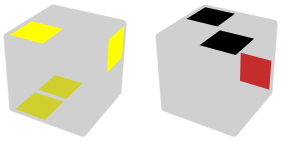
R U2 R

z2 x R2



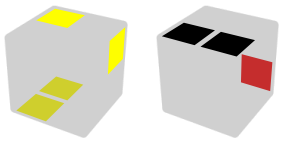
R U R2'

x' L' U



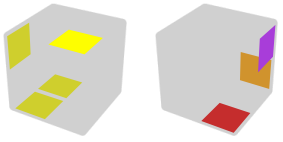
R U' R' U R'

R'



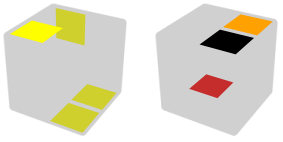
R2' U R U' R

R'



R' U' R'

y2 z (B', L')

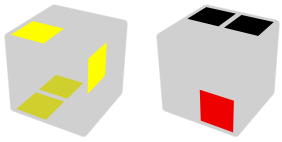


L' F U' F'

= y' F' R U' R'

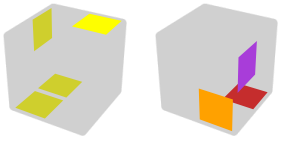
R2, U'

= y2 R' F R' F'



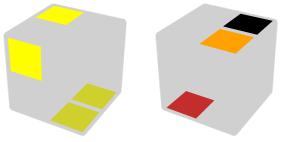
R2 U' R' U R'

R



R U R

y2 z (F, L)

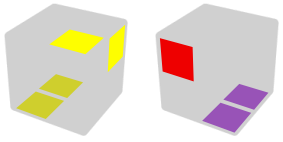


L F' L F

= y' F R' F R

L2, U

= y2 R F' U F



R2' F R U' R'

x2 L

R2 U R' U' R

R', R U'

L2 F' L' U L

x2 R'

R2' U' R U R'

R, R' U

UU - 0 Up [7]

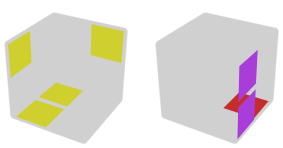
R2 U2' R' z2 R

R2' U2' R z2 R'

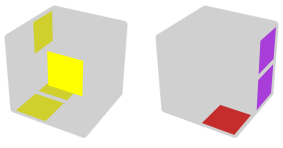
R U' R U R' R U'

R' U R' U' R R' U

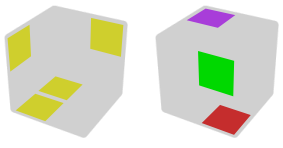
U' R' U' R2 U R'
U' R U R2' U' R



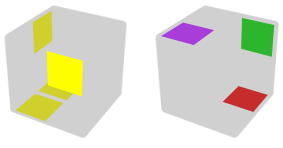
R U R U2' R' z' B



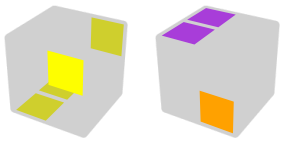
R' U' R' U2' R z' F'



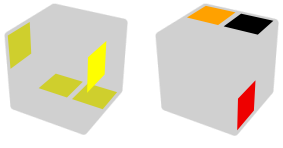
R2 U' R' y' (+, B2)



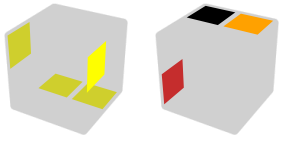
R2' U R y' (-, F2)



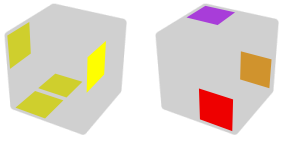
R U2' R2' U' R2 y2 L U'
= R' U2' R2 U R2'



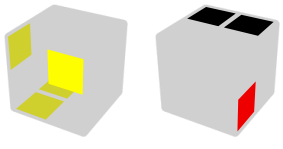
F' R2 U' R2' F', U'



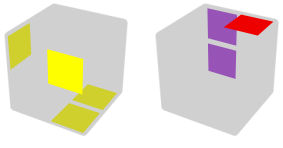
F L2' U L2 F, U



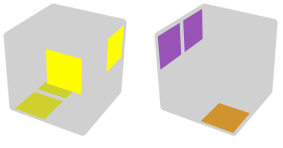
R U' R' U R' U' R y' (B, B')
= R' U R U' R U R'



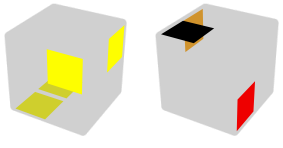
R U' R U R2' F'
R' U2 R' U' R



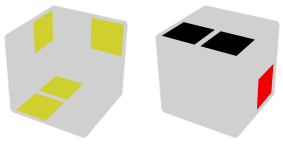
L F' L' U L' U' L2 x' R



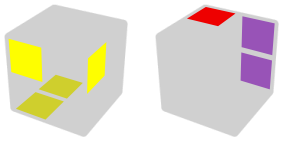
R' U R' U2' R z y' D' F
R' U2 R' U R



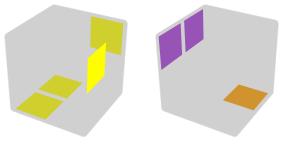
R' U' F R F' F', B'



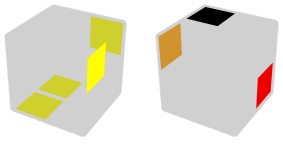
R' U R' U' R2 B
R U2 R U R'



R' F R U' R U R2' x' L'

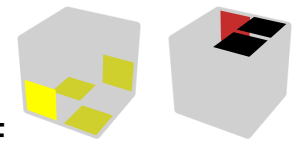


R U' R U2' R' z y D B'
R U2 R U' R'

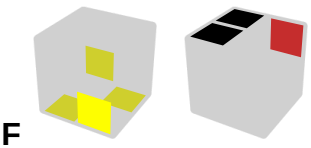


R U F' U' F B, F

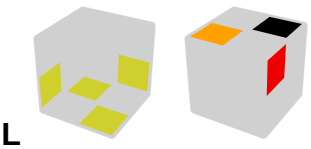
(T)CLL [4]



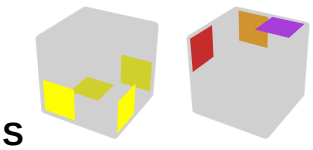
F R2 U R' F' R' L



F L2' U' L F L R'



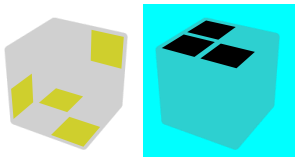
L R2 U R2' F' F', U



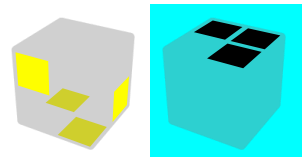
S F U2 R y2 (B, R)

Easy Checker [12]

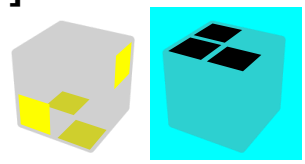
UD [6]



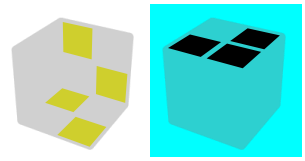
R D R'



R F R'

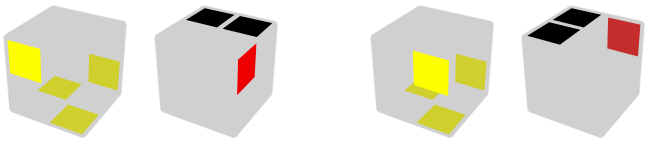


F' R' F



x R U R' F' R'

U' R F U R'



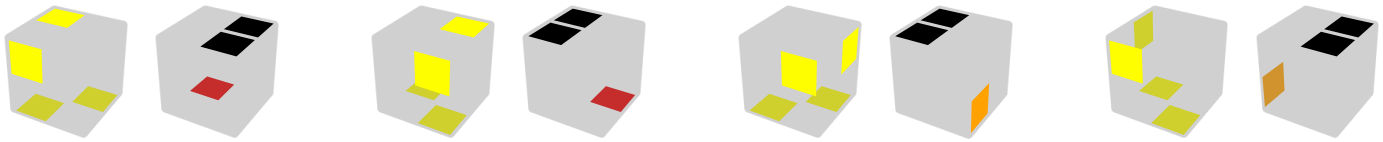
x U' R U' R' U x'

F'

x R U' R' U R' x'

R'

UU [6]



x U L U' L x'

L2

x U' R' U R' x'

R2

x2 R U' R' U L' x

D R'

x2 L' U L U' R x

D' L

R2' F R2

F

L2 F' L2

F'

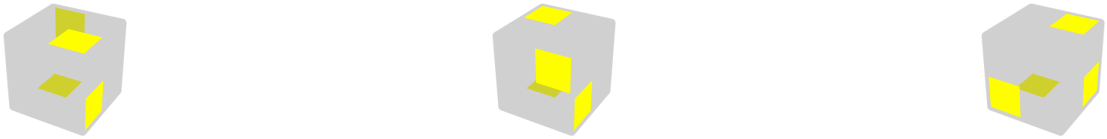
Others [7]

F R2

R' F' R' F' U' R'

F' U2 R

R2 F' R'



U' F' R2

U F' R2

R F U2 R