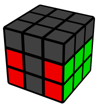
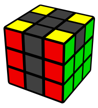
Roux CMLL [42]



Intermediate



CubeR oot

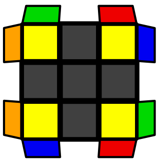
Roux CMLL Algorithms

(Corners Orientation & Permutation of Last Layer Ignoring M-Slice)

Algorithms from Kian Mansour http://sites.google.com/view/kianroux, Video showing execution http://bit.ly/2j5WX86

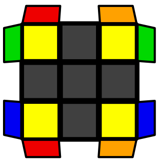
CMLL, the third step of Roux, orients and permutes corners simultaneously, without regard to the M-slice edges or U layer corners. #1 alg that preserves EO, #2 alg that flips edges, **suggested alg,** alternative alg, **OH alg.**

O

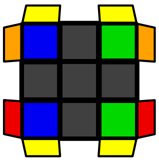


(R U R' F') (R U R' U') (R' F R2 U' R')
(r U R' F') (R U R' U') (R' F R2 U' r')
(R U R' U') (R' F R2 U') R' U' (R U R' F')
(R U R' U') r' F (R2 U' R' U') (R U R' F')
U (r U' L U2) (R' U R U2) r' L'
(R U2 R' U') (R U2 L' U) R' U' L
Adj Swap

F (R U' R' U') (R U R' F') (R U R' U') (R' F R F')
r2 D (r' U r D') R2' U' (F' U' F)
(r' U' r' D') (r U' r' D) (r U r' D') (r U r' D) r2
F (R U' R' U') (R U R' F') (r U R' U') (r' F R F')
F (R U' R' U') (R U R' F') (R U R' U') (R' F R F')
(R U R' U) (R' F R F') R U2' R2 U' R U' R' U2' R
Diag Swap

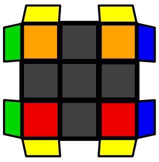


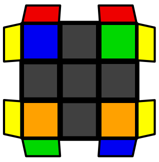
H



(R U2' R' U') (R U R' U') (R U' R')
(r U2' R' U') (R U R' U') R U' r'
U (R U R' U) (R U' R' U) (R U2' R')
Columns

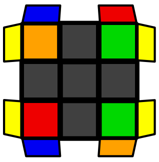
F (R U R' U') (R U R' U') (R U R' U') F'
(R' F R U2') (R' F' R U) F (R U R' U) F'
Rows



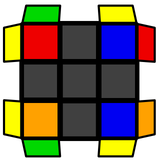


F (R U' R' U) (R U2 R' U') (R U R' U') F'
R U2' (R2' F R F') U2' (R' F R F')
Column

(R U R' U R U) r' F R' F' r
U' (r U' r2' D') (r U' r' D) r2 U r'
(r U' r' F) U2' (r2' F r U' r)
R' U' (R U' R' U') L U' R U L'
Row

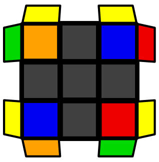


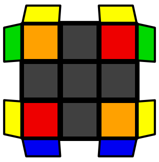
Pi



R U2' R2' U' R2 U' R2' U2' R
F (R U R' U')2 F'
f (R U R' U')2 f'
U2 F (U R U' R')2 F'
r' U r2 U' r2' U' r2 U r'
r U' r2' U r2 U r2' U' r
R' U2 (r U' r' U2) (r U r' U2) R
Right Bar

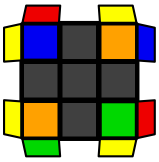
F U (R U' R' U) (R U' R2' F') (R U R U' R')
(F R' F' R) U2 (R U' R' U) (R U2' R')
(R U2' R' U') (R U' R2' U) (L U' R U L')
\

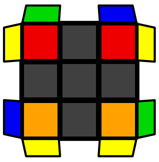




U R' F2 D (R2 U' R2' D') F2 R
(R' F R U) F U' (R U R' U') F'
F (R U' R' U') (R U R' U) (R U' R' F')
Checker

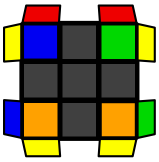
(R U R' U') R' F (R2 U R' U') (R U R' U') F'
(R U2' R' U') (R U R' U2') (R' F R F')
U' R' U2' (R U R' U) R2 U' (L' U R' U' L)
/



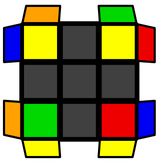


U' (r' F R F') (r U' R' U') (R U' R')
(r U' r2' D') (r U r' D) r2 U r'
U' (R' U R U') (R2' F R2 U) R' U' F' R
U (R' U L U') R U' (L' U' L U' L')
U' (L' U R U') (L U' R' U') (R U' R')
Columns

(R U2 R' U') F' (R U2' R' U') (R U' R' F) (R U' R')
R' U' (R' F R F') R U' R' U2 R
U R' U' (R U' R' U) F' U F R
Left Bar



U



R2 D (R' U2 R D') R' U2 R'

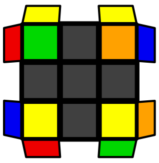
U2 (r U' r' U')2 F' U2 F

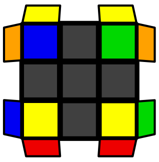
/

R2' D' (R U2 R' D) R U2 R

R' F (R U R' F) R U (F U2' F')

\





U R U2' (R2' U' R2 U') (R' U R' U') (R U R' U) R

R2' F U' F U F2 R2 U' (R' F R)

(R' U' R U') (R' U2' R2 U) (R' U R U2' R')

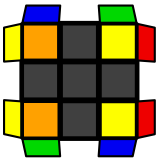
Front Row

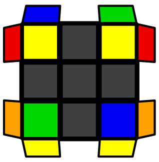
U R' F (R U' R' U') R U R' F' (R U R' U') (R' F R F') R

F R2 D (R' U R D') R2' U' F'

U F U' R U' R' U2' R U' R' U2' (R U' R' U') F'

Rows





U2 (R U R' U') (R' F2 R2 U') (R' U' R U) R' F2

(r U' r' U) r' D' (r U' r' D) r

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

U R' D (R U' R U') (R' U R' D') R

F (R U' R' U) (R U R' U) (R U' R' F')

Checker

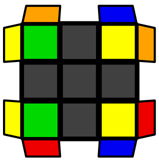
U (R U' R' U') (R U R D) (R' U R D') R2

F (R U R' U') F'

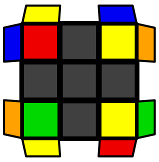
f (R U R' U') f'

U2 F (U R U' R') F'

Back Row



T



(r U R' U') (r' F R F')

(R U R' U') (R' F R F')

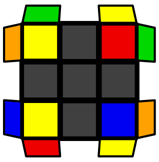
Left Bar

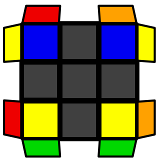
(l' U' L U) (l F' L' F)

(L' U' L U) (L F' L' F)

U2 F (R U' R' U) (R U R' F')

Right Bar





U' (R U R' U) (R U' R' U) R' U' (R2 U' R2' U2' R)

(F R' F R2) U' (R' U' R U) R' F2

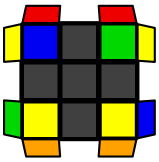
(R U2' R' U') (R U' R2' U2') (R U R' U R)

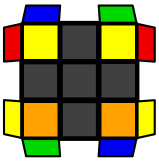
Rows

(R' U r U2') (R2' F R F') r

(r' U r U2') (R2' F R F') R

Front Row





F (R U R' U') (R U' R' U') (R U R' F')

U2 r' D' (r U r' D) (r U' r U r')

U' R' D (R U' R U) (R' U R' D') R

Back Row

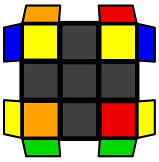
U2 (R' U R2 D) (r' U2 r D') (R2' U' R)

r2' D' (r U r' D) (r2 U' r' U' r)

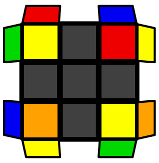
(r U' r2' D') (r U2 r' D) (r2 U r')

U2 R' U' (R U' R' U)2 F' U F R

Columns



L



U2 R2' D' (R U' R' D) R U R

F (R U' R' U') (R U R' F')

U F' (r U r' U') r' F r

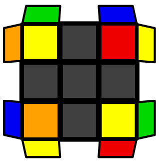
U F' (r U R' U') r' F R

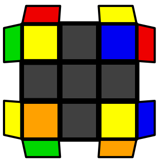
Mirror

(F R' F' r) (U R U' r')

(F R' F' R) (U R U' R')

Inverse





(R U2 R' U') (R U R' U')2 (R U' R')

(R U R' U') (R' F R F') (R' F R F') r U r'

(R U R' U) (R U' R' U)2 R U2' R'

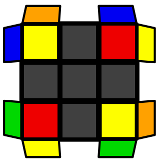
Pure

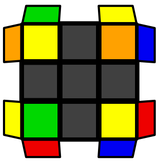
(R U2 R D) (R' U2 R D') R2'

U2 (R U R' U) (R' F R F') U2 (R' F R F')

(R U R' U') (F' U2' F) (U R U R')

Front Com





R' U' (R U R' F') (R U R' U') R' F R2

U2 R U2' (R2' F R F') R U2' R'

U2 r U2' (R2' F R F') R U2' r'

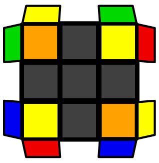
Diag

(R' U2 R' D') (R U2 R' D) R2

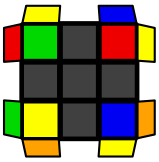
U (R U R' U') (R' F R2 U') (R' U R U) R' F'

U' (R U R' U2') (L U' R U) (L' U R')

Back Comm

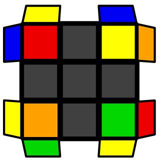


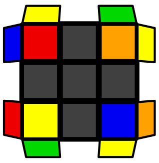
S



R U R' U R U2' R'
r U R' U R U2' r'
Left Bar

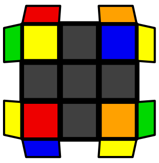
U2 (L' U2 L U2') (l F' L' F) M'
U2 (L' U2 L U2') (L F' L' F)
(R' U2' R U2) R f' U' f
Checker

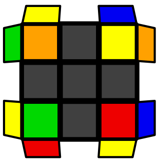




M (F R' F' R) (U2 R U2' r')
(F R' F' R) (U2 R U2' R')
/

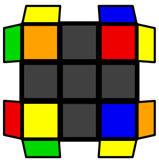
R' U' (R U' R2' F') (R U R U') R' F U2' R
U' (R U R' U') (R' F R F') (R U R' U R U2' R')
U2 (R U R' U) (R U' R D) (R' U' R D') R2'
Columns



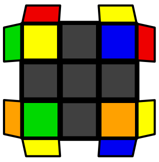


(R U R' U) (L' U R U') L U2 R'
(R U R' U) (R' F R F') R U2' R'
Right Bar

R U' L' U R' U' L
(r U' r') (F R' F' R)
U2 L U' R' U L' U' R
\

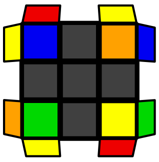


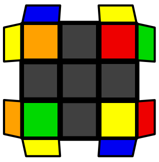
AS



R' U' R U' R' U2 R
r' U' R U' R' U2 r
U R U2' R' U' R U' R'
Right Bar

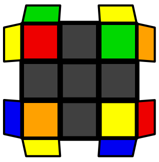
R2 D (R' U R D') (R' U R' U') (R U' R')
r R D (R' U R D') (R' U R' U') R U' r'
Columns

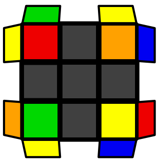




M (F' L F L') (U2' L' U2 L) M'
(F' L F L') (U2' L' U2 L)
F' (r U r' U2) (r' F2' r)
\

(R U2' R' U2) (r' F R F') M'
(R U2' R' U2) (R' F R F')
Checker





L' U R U' L U R'
(R' F R F') r U r'
U2 R' U L U' R U L'
/

U' (R' U' R U') (L U' R' U) (L' U2 R)
R' U' (R U' R' U) (R' F R F') U R
Left Bar