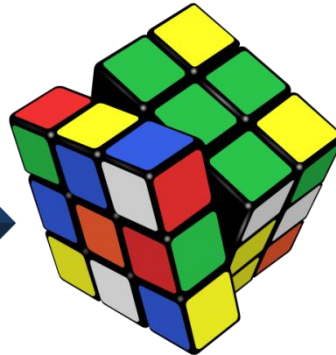


## Baxter Research Robot

Active Robots Systems Integration Example:  
Solving Rubik's Cube



# Baxter Solving the Rubik's Cube

Code: Python

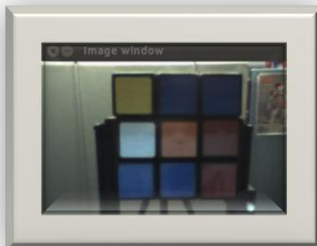


Using edge detection to look for defined areas of colour within black borders. Detection of 9 areas of uniform colour. Colours as RGB tuples.

Take colour tuples and match them in vector space. Verify that the result is a valid mapping for a Rubik's Cube

A search based algorithm for solving any mapping of the Rubik's Cube. 2-stage Kociemba reduces the search space using group theory.

Baxter arm movement using MoveIt for OMPL based IK, interfaced using the MoveIt Commander Python wrapper. Additional direct joint control /feedback directly using RSDK API.



```
baxter@Baxter-PC: ~/git/rdk-examples
> ./rb33
6 colours found
Failed to find 9 stickers of each colour
false True
FAIL on first attempt
-----
0 0 0 0 0 0 0 0
1 3 1 1 1 1 1 1
2 2 2 2 2 2 2 2
3 3 3 3 3 3 3 3
4 4 4 4 4 4 4 4
5 5 5 5 5 5 5 5
-----
6 colours found
9 of each colour found
Faces have unique colour
Edges have valid sticker colours
Edges are unique
Corners have valid colours
Corners are unique
True True
Success on retry
-----
0 0 0 0 0 0 0 0
```

```
baxter@Baxter-PC: ~/git/rdk-examples
Cube Mapping: 'ddruu1lbbdbdrfbuuurrrfrfrffddibbbfrfudluuldbflfr'
Sending Cube Mapping to Kociemba Algorithm
Kociemba algorithm solved this cube in 8 moves
List of Singmaster manipulations to solve this cube mapping: ['F1', 'B2', 'U1',
'L1', 'D1', 'R2']
Manipulating series of Singmaster moves:
['F1', 'B2', 'U1', 'L1', 'D1', 'R2']
Parsed Singmaster Manipulation = 'F1'
...
>>> doing manipulation 'F1'
current: 'DRL'
new: 'DRL'
current hold = new hold -> no hold change required
closing
close cmd
opening
open cmd
F1'
...
Parsed Singmaster Manipulation = 'B2'
```

