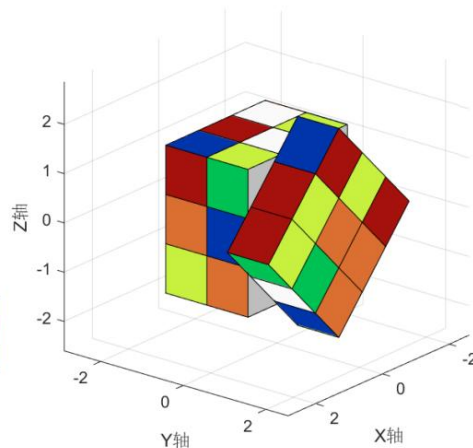
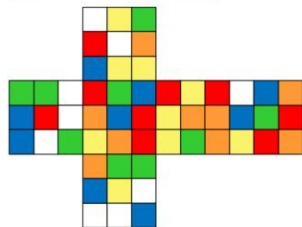


一. 问题分解

任务 3：魔方识别与还原

- **要求：**读取三阶魔方六个面的图像，识别每个面的颜色，求解并展示旋转过程与结果。测试图不少于 10 组。



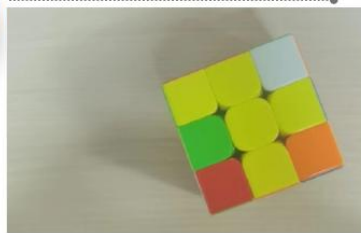
0. 识别输入图片中的魔方区域，并变换为规则的输入图片
1. 识别输入图片中每个面的颜色，并映射到魔方数组中；
2. 求解魔方；
3. 展示旋转过程和求解结果。

二. 求解过程

0. 预处理

(1) 预处理主要步骤:

STEP 1 检测魔方有效区域



高斯滤波 `imgaussfilt()`

边缘检测 `edge()`

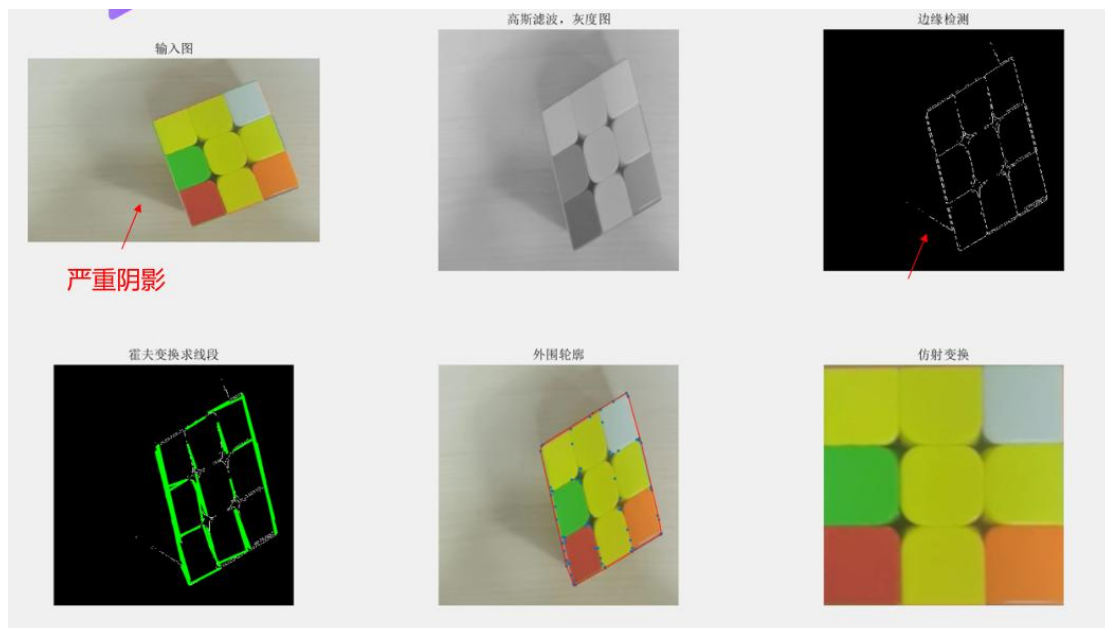
霍夫变换 `hough()`
`houghpeaks()`
`houghlines()`

求轮廓 `boundary()`

求角点

仿射变换 `fitgeotrans()`
`imwarp()`

(2) 每一步的中间结果 (可用 `preprocess.m` 中的绘图语句分步输出)



1. 颜色识别和魔方构造

(1) 颜色归类方法

以中心块的 RGB 值为基准, 计算其他块到 6 块中心块的欧氏距离。每个中心块选择与它颜色最为最近的其他 8 个块归为一类, 将同一类面的颜色统一为中心块的颜色。由于使用了两个上色不同的魔方图片作为输入, 红色和橙色的分辨出现了一点困难。

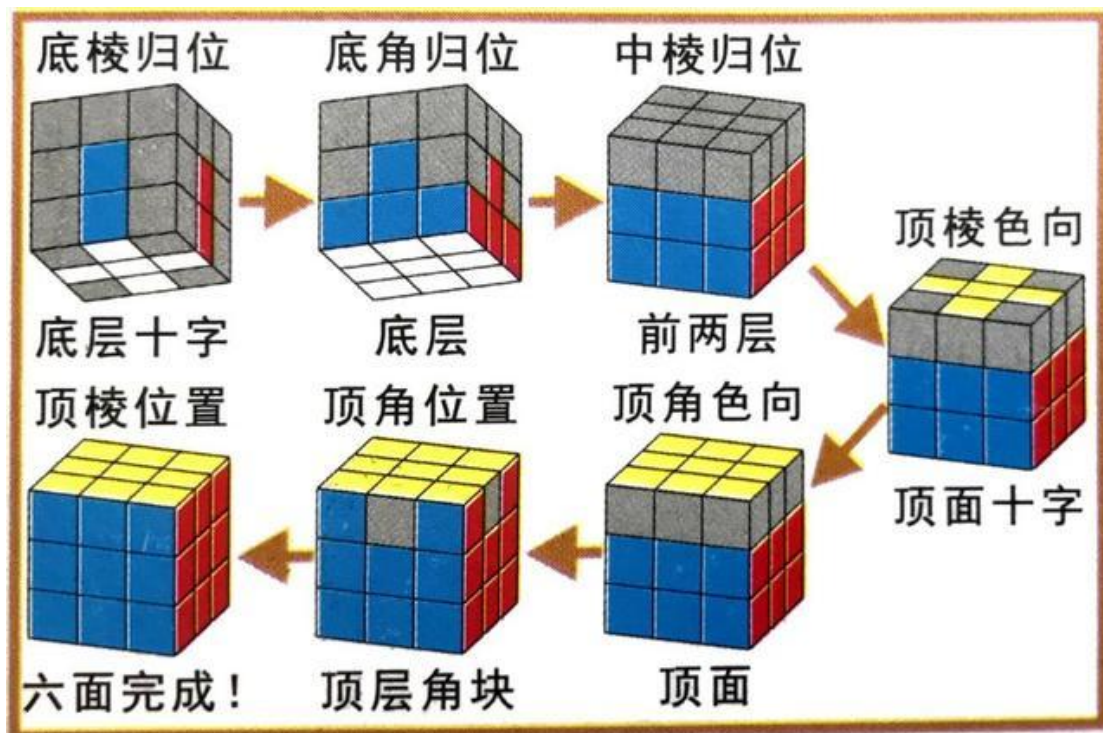


解决方法是，计算颜色之间的欧氏距离时，将 R 通道和 G 通道的权重提升至 1.2 倍，增大红色和橙色的距离，结果红色和橙色的分辨更准确了。

(2) 魔方的表示

用 `cub[27]` 矩阵表示魔方，cub 有 27 个小魔方块，正面右下角编号为 1，编号按 xyz 轴顺序增加背面右上角编号为 27. 每个小魔方块有 6 个色块 face，按前左右上下后顺序编号，每个色块有 RGB 三个属性。

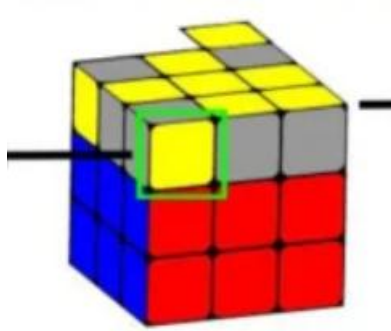
2. 魔方解法：层先法



实现思路：前两层的复原中，每四个块的复原方式是一致的，用同样的解法循环四次。例如底层的四个角块，可以将正面右下角的槽位作为目的位置，找到目标块，对目标块所在的位置进行分类讨论。复原了一块以后转动底层将正面右下角的槽位让给还没复原的角块，循环知道底层角块全部复原。

顶层的复原就是套公式，每种公式的适用情况都有它独特的特征，通过魔方颜色可以判

断出来该用哪一条公式。比如这条“小鱼公式”的判断逻辑就是，4 个顶层角块的右侧面中，有 3 个面与顶层中心块同色。如果对魔方还原足够熟悉，就可以知道每条公式的适用情况有什么独有的特征。



3. 复原展示

绘图函数使用 `fill3`，直接传入魔方数组的顶点坐标和颜色就可以了。通过对图形句柄属性的更新实现对动画效果。