(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 111383352 A
(43)申请公布日 2020.07.07

(21)申请号 202010205243.4

(22)申请日 2020.03.20

(71)申请人 北京工业大学
地址 100124 北京市朝阳区平乐园100号

(72)发明人 康存锋 罗明睿 石春阳 陈东临

(74)专利代理机构 北京思海天达知识产权代理有限公司 11203
代理人 刘萍

(51)Int.Cl.
G06T 19/20(2011.01)
G06T 5/00(2006.01)
G06T 7/13(2017.01)
G06T 7/136(2017.01)
G06T 7/90(2017.01)

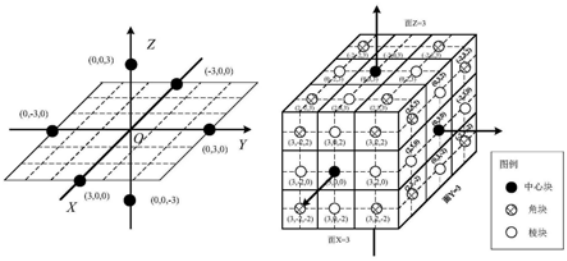
权利要求书3页 说明书9页 附图4页

(54)发明名称

一种针对三阶魔方的自动填色及抽象化方法

(57)摘要

本发明公开了一种针对三阶魔方的自动填色及抽象化方法,在保证识别效率变化不大的情况下解决了颜色识别成本过高的问题,并且提出的魔方抽象化模型可以简化魔方的表达,更适合高阶魔方的算法编程。本发明包含三个部分,分别为魔方的抽象化表达模型:魔方状态模型,基于单目视觉的魔方色块识别,以及基于上述两者的三阶魔方自动填色方法。



CN 111383352 A

1.一种魔方的抽象化表达与魔方色块识别方法，其特征是：构建以魔方中心O为坐标系原点的右手系空间直角坐标系XYZ，坐标轴通过各个面的中心块，各个面上的各个色块均由该空间内的点{(x,y,z)|x,y,z∈{-3,-2,0,2,3}}唯一确定，并且色块的颜色即f(x,y,z)；之后通过图像去噪、轮廓识别、网格划分、色彩比对这四步完成魔方色块识别。

2.根据权利要求1所述的方法，其特征在于，魔方的抽象化表达具体为在构建以魔方中心O为坐标系原点的右手系空间直角坐标系XYZ后，魔方的12种基本操作方法即各个面的顺时针90°旋转以及各个面的逆时针90°旋转，都转换成色块坐标的空间变换，设魔方旋转前色块坐标为(x,y,z)，顺时针90°旋转后色块坐标为(x′,y′,z′)，逆时针90°旋转后色块坐标为(x″,y″,z″)，需要计算的色块坐标集合为P：

对面X＝±3旋转：P＝{(x,y,z)|x＝±3,±2}

$$\begin{bmatrix}x'\\y'\\z'\\1\end{bmatrix}=\begin{bmatrix}1&0&0&0\\0&0&-1&0\\0&1&0&0\\0&0&0&1\end{bmatrix}\begin{bmatrix}x\\y\\z\\1\end{bmatrix}\qquad\begin{bmatrix}x''\\y''\\z''\\1\end{bmatrix}=\begin{bmatrix}1&0&0&0\\0&0&1&0\\0&-1&0&0\\0&0&0&1\end{bmatrix}\begin{bmatrix}x\\y\\z\\1\end{bmatrix}$$

对面Y＝±3旋转：P＝{(x,y,z)|y＝±3,±2}

$$\begin{bmatrix}x'\\y'\\z'\\1\end{bmatrix}=\begin{bmatrix}0&0&1&0\\0&1&0&0\\-1&0&0&0\\0&0&0&1\end{bmatrix}\begin{bmatrix}x\\y\\z\\1\end{bmatrix}\qquad\begin{bmatrix}x''\\y''\\z''\\1\end{bmatrix}=\begin{bmatrix}0&0&-1&0\\0&1&0&0\\1&0&0&0\\0&0&0&1\end{bmatrix}\begin{bmatrix}x\\y\\z\\1\end{bmatrix}$$

对面Z＝±3旋转：P＝{(x,y,z)|z＝±3,±2}

$$\begin{bmatrix}x'\\y'\\z'\\1\end{bmatrix}=\begin{bmatrix}0&-1&0&0\\1&0&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}\begin{bmatrix}x\\y\\z\\1\end{bmatrix}\qquad\begin{bmatrix}x''\\y''\\z''\\1\end{bmatrix}=\begin{bmatrix}0&1&0&0\\-1&0&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}\begin{bmatrix}x\\y\\z\\1\end{bmatrix}$$

旋转后

g(x′,y′,z′)＝f(x,y,z)，g(x″,y″,z″)＝f(x,y,z)

最终形成的$f(x,y,z)\leftrightarrow g(x,y,z)$即为旋转前后的魔方状态变化关系。

3.根据权利要求1所述的方法，其特征在于，轮廓识别方法具体为：设置六条直线l_A、l_B、l_C、l_D、l_E、l_F为魔方边界逼近线，且k_A、k_B、k_C、k_D、k_E、k_F分别为六条逼近线斜率，边界点提取算法通过l_A、l_B、l_C、l_D、l_E、l_F六条直线逼近轮廓边缘，各边界点满足以下数学条件：

设U＝{(x,y)|g(x,y)＝255}则

A＝{(x,y)|$y=\min(y_1,\ldots,y_j)$,(i,j)∈U}

B＝{(x,y)|$y-k_Bx=\min(y_1-k_Bx_1,\ldots,y_j-k_Bx_i)$,(i,j)∈U}

C＝{(x,y)|$y-k_Cx=\max(y_1-k_Cx_1,\ldots,y_j-k_Cx_i)$,(i,j)∈U}

D＝{(x,y)|$y=\max(y_1,\ldots,y_j)$,(i,j)∈U}

E＝{(x,y)|$y-k_Ex=\max(y_1-k_Ex_1,\ldots,y_j-k_Ex_i)$,(i,j)∈U}

F＝{(x,y)|$y-k_Fx=\min(y_1-k_Fx_1,\ldots,y_j-k_Fx_i)$,(i,j)∈U}

其中k_B、k_C、k_E、k_F为逼近线l_B、l_C、l_E、l_F的斜率，需满足条件：k_B,k_E<0,k_C,k_F>0；

具体实现过程，提供两种搜索方式，分别为多边形逼近搜索和水平扫描搜索，根据搜索效率选用原则如下：

设n为大于0的自然数，另设$\sum_{n=1}^{6} S_n$为多边形逼近重复搜索面积，S_{abcdef}为多边形逼近未搜索面积，如果$\sum_{n=1}^{6} S_n \le S_{abcdef}$则选用多边形逼近搜索，否则选用水平扫描搜索；

多边形逼近搜索的具体实现如下：

逼近直线l_A、l_B、l_C、l_D、l_E、l_F的解析式分别为：

l_A:f(x)＝b

l_B:f(x)＝k_Bx+b

l_C:f(x)＝k_Cx+H-b

l_D:f(x)＝H-b

l_E:f(x)＝k_E(W-x)+(H-k_EW-b)

l_F:f(x)＝k_F(W-x)+(-k_EW+b)

其中W为图像的宽度，H为图像的高度，b控制搜索范围，b>max(W,H)。

4.根据权利要求1所述的方法，其特征在于，网格划分方法具体为：魔方各色块的位置由$\vec{V_1}$、$\vec{V_2}$、$\vec{V_3}$线性导出，即$\vec{V_1}$、$\vec{V_2}$、$\vec{V_3}$为魔方空间的基向量；六个边界点，确定两组重复的基向量，利用这个特点矫正边界点定位误差，具体算法如下：

$$\vec{V_1}=\begin{bmatrix}\frac{x_B+x_D-x_A-x_E}{6}\\ \frac{y_B+y_D-y_A-y_E}{6}\end{bmatrix}\quad \vec{V_2}=\begin{bmatrix}\frac{x_F+x_D-x_A-x_C}{6}\\ \frac{y_F+y_D-y_A-y_C}{6}\end{bmatrix}\quad \vec{V_3}=\begin{bmatrix}\frac{x_C+x_E-x_B-x_F}{6}\\ \frac{y_B+y_D-y_A-y_E}{6}\end{bmatrix}$$

ABGF面各色块中心点计算公式：

$$\begin{bmatrix}x_i\\ y_j\end{bmatrix}=\begin{bmatrix}x_A\\ y_A\end{bmatrix}+(i+p)\vec{V_1}+(j+q)\vec{V_2}$$

BCDG面各色块中心点计算公式：

$$\begin{bmatrix}x_i\\ y_j\end{bmatrix}=\begin{bmatrix}x_B\\ y_B\end{bmatrix}+(i+p)\vec{V_3}+(j+q)\vec{V_2}$$

GDEF面各色块中心点计算公式：

$$\begin{bmatrix}x_i\\ y_j\end{bmatrix}=\begin{bmatrix}x_F\\ y_F\end{bmatrix}+(i+p)\vec{V_3}+(2\text{-}j+q)\vec{V_1}$$

其中x_i,y_j(i,j＝A、B、C、D、E、F,)表示六个边界点的任意一点基坐标，p为i方向上的坐标偏移值，q为j方向上的坐标偏移值，标准魔方p＝0.5,q＝0.5。

5.根据权利要求1所述的方法，其特征在于，三阶魔方自动填色方法具体为：摄像头位于魔方的对角线上，一次拍摄魔方的三个面，一次性获取三个面上所有色块的颜色；将魔方初始状态记为状态(1)，第一次拍摄取样获取色块{(x,y,z)|(x＝3)or(y＝3)or(z＝3)}上

的颜色;第一、二步操作同时对面Z＝±3连续两次顺时针旋转90°得到状态(2),此时进行第二次拍摄取样,获取色块{(x′,y′,z′)|x′＝3,z′＝±2}以及{(x′,y′,z′)|y′＝3,z′＝±2}上的颜色;第三、四步操作继续对面Z＝±3连续两次顺时针旋转90°回到状态(1);第五、六步操作对X＝±3连续两次顺时针旋转90°得到状态(3),此时进行第三次拍摄取样,获取色块{(x′,y′,z′)|z′＝3,x′＝±2}以及{(x′,y′,z′)|y′＝3,x′＝±2}上的颜色;第七、八步操作继续对面X＝±3连续两次顺时针旋转90°回到状态(1);第九、十步操作对Y＝±3连续两次顺时针旋转90°得到状态(4),此时进行第四次拍摄取样,获取色块{(x′,y′,z′)|z′＝3,y′＝±2}以及{(x′,y′,z′)|x′＝3,y′＝±2}上的颜色;第十一、十二步操作继续对面X＝±3连续两次顺时针旋转90°回到状态(1);根据上文所述的状态模型对所有的旋转后色块(x′,y′,z′)逆向转换即可得到旋转前即状态(1)中色块(x,y,z)的颜色f(x,y,z)。

一种针对三阶魔方的自动填色及抽象化方法

技术领域

[0001] 本发明涉及一种针对三阶魔方的自动填色及抽象化模型建立方法，具体涉及魔方色块的单目视觉识别及魔方色块的抽象化表示方法。

背景技术

[0002] 魔方是生活中常见的一种益智玩具，而三阶魔方由于上手容易，特别适合对初学者进行教学和训练，也常常作为魔方自动复原装置的操作对象。自动装置还原魔方的第一步即需识别魔方各个面上颜色块，现有的技术往往采用多颜色传感器或摄像头分别对单一面或单一区块进行颜色识别，因此装置成本较高，不适合推广应用，并且在魔方色块的抽象化表示中往往采用展开图数字标记的方法，需要在计算机中储存大量规则，尤其对于高阶魔方，抽象化模型建立过程相对复杂。

发明内容

[0003] 本发明旨在提供一种针对三阶魔方的填色及抽象化模型建立方法，在保证识别效率变化不大的情况下解决了颜色识别成本过高的问题，并且提出的魔方抽象化模型可以简化魔方的表达，更适合高阶魔方的算法编程。

[0004] 本发明包含三个部分：分别为魔方的抽象化表达模型即魔方状态模型、基于单目视觉的魔方色块识别以及基于上述两者的三阶魔方自动填色方法。

[0005] 本发明定义魔方的状态模型用于描述魔方每次旋转后各个色块的移动规律，通过魔方的状态模型可以将魔方的每次操作、每种状态抽象化。通过构建以魔方中心O为坐标系原点的右手系空间直角坐标系XYZ，坐标轴通过各个面的中心块，各个面上的各个色块均可由该空间内的点{(x,y,z)|x,y,z∈{-3,-2,0,2,3}}唯一确定，并且色块的颜色即f(x,y,z)。

[0006] 在本发明定义的魔方状态模型中，对魔方只需要12种基本操作方法，即各个面的顺时针90°旋转以及各个面的逆时针90°旋转。并且所有的基本操作都可以转换成色块坐标的空间变换，设魔方旋转前色块坐标为(x,y,z)，基本操作后色块坐标经过空间变换为(x′,y′,z′)，新位置下色块颜色g(x′,y′,z′)＝f(x,y,z)，最终形成的$f(x,y,z)\leftrightarrow g(x,y,z)$即为旋转前后的魔方状态变化关系。

[0007] 本发明提出的基于单目视觉识别的魔方色块提取可细化为图像去噪、轮廓识别、网格划分、色彩比对这四步。

[0008] 图像去噪步骤包含对图像进行的两次滤波过程。首先使用双边滤波对摄像头获取到的魔方照片进行处理。双边滤波是一种非线性滤波器，它可以达到保持边缘、降噪平滑的效果，使用该滤波器可以在降噪的同时保存魔方的边缘，为后面的边缘检测、网格划分提供便利。在双边滤波结束后，继续对图片进行非局部平均降噪。该降噪与前面的双边滤波不同的是，该算法利用整幅图像来进行降噪，在图像中以块为单位寻找相似区域，然后对相似区域的颜色求平均值。经过图像去噪步骤后干扰魔方识别的背景噪点会被精准的过滤掉。

[0009] 边缘识别步骤包含梯度计算、梯度二值化、边界点定位三个过程。图像边缘的像素点梯度值会趋于一个较大的数值，因此可以通过求取图像各像素点的梯度来判断图像的边缘。为了提高色块提取效率，本发明摄像头安装于魔方对角侧，可以同时获取魔方三个面的图像，通过l_A、l_B、l_C、l_D、l_E、l_F六条直线逼近轮廓边缘定位魔方六个边界点的位置，从而便于对其进行网格划分。具体实现过程，本算法提供两种搜索方式，分别为多边形逼近搜索和水平扫描搜索，根据搜索效率选用原则如下：设$\sum_{i=1}^{6} S_i$为多边形逼近重复搜索面积，S_{abcdef}为多边形逼近未搜索面积，如果$\sum_{i=1}^{6} S_i \le S_{abcdef}$则选用多边形逼近搜索，否则选用水平扫描搜索。

[0010] 网格划分步骤包含基向量计算、色块中心点定位、图像取样三个过程。魔方各色块的位置均可由边界点构成的魔方空间的基向量$\vec{V_1}$、$\vec{V_2}$、$\vec{V_3}$线性导出，六个边界点，可以确定两组重复的基向量，利用这个特点可以矫正边界点定位误差。根据中心点坐标(x_i,y_j)截取中心矩形区域，即完成对色块的取样。

[0011] 色彩对比步骤包含色彩校正、色彩空间二值化、可能性计算三个过程。由于魔方色块颜色识别的准确率直接影响魔方的求解，通过色彩校正可以极大提高颜色识别的准确率，在摄像头固定位置放置比色卡，比色卡包含标准魔方所需的六种颜色，每次拍摄时比色卡与魔方处于相同的光照环境，因此可以直接从比色卡相应区域提取色彩空间二值化的高阈值T_H及低阈值T_L。色彩空间二值化即每个像素点(x,y)判断其HSV值是否位于高低阈值内，如果位于高低阈值内，则此点像素值g(x,y)＝255，否则此点像素值g(x,y)＝0。对六种颜色分别进行色彩空间二值化得到图像矩阵M_i(i＝0,1,2,3,4,5)，每种颜色的可能性则可由矩阵的和S_i＝sum(M_i)表示，使得矩阵和S_i最大的那种颜色即为可能性最大的颜色，作为色块颜色识别的最终结果。

[0012] 利用本发明提出的魔方色块单目视觉识别以及魔方状态模型，可以对三阶魔方进行自动填色。本发明提出12步操作方法，对魔方进行4次取样即可完整获取魔方的初始状态，以下是具体的实现方法：

[0013] 摄像头位于魔方的对角线上，可以一次拍摄魔方的三个面，由上文提出的色块提取方法，可以一次性获取三个面上所有色块的颜色。将魔方初始状态记为状态(1)，第一次拍摄取样可以获取色块{(x,y,z)|(x＝3)or(y＝3)or(z＝3)}上的颜色。第一、二步操作同时对面Z＝±3连续两次顺时针旋转90°得到状态(2)，此时进行第二次拍摄取样，可以获取色块{(x′,y′,z′)|x′＝3,z′＝±2}以及{(x′,y′,z′)|y′＝3,z′＝±2}上的颜色；第三、四步操作继续对面Z＝±3连续两次顺时针旋转90°回到状态(1)；第五、六步操作对X＝±3连续两次顺时针旋转90°得到状态(3)，此时进行第三次拍摄取样，可以获取色块{(x′,y′,z′)|z′＝3,x′＝±2}以及{(x′,y′,z′)|y′＝3,x′＝±2}上的颜色；第七、八步操作继续对面X＝±3连续两次顺时针旋转90°回到状态(1)；第九、十步操作对Y＝±3连续两次顺时针旋转90°得到状态(4)，此时进行第四次拍摄取样，可以获取色块{(x′,y′,z′)|z′＝3,y′＝±2}以及{(x′,y′,z′)|x′＝3,y′＝±2}上的颜色；第十一、十二步操作继续对面X＝±3连续两次顺时针旋转90°回到状态(1)。根据上文所述的状态模型对所有的旋转后色块(x′,y′,z′)逆向转换即可得到旋转前即状态(1)中色块(x,y,z)的颜色f(x,y,z)。至此，魔方54个色块

已经获取51个色块的颜色，剩余3个无法获取的色块颜色为三个面的中心块颜色，根据魔方复原原理，魔方只需要获得所有角块、棱块的颜色即可，因此本方法满足魔方复原的要求。

附图说明

[0014] 图1为本发明的魔方状态模型示意图

[0015] 图2为本发明的魔方边界点提取算法示意图

[0016] 图3为本发明的魔方空间网格划分算法示意图

[0017] 图4为本发明的三阶魔方自动填色方法流程图

[0018] 图5为本发明的三阶魔方自动填色方法示意图

具体实施方式

[0019] 以下，结合附图以及具体实施方法，对本发明做进一步阐述。

[0020] 本发明定义魔方的状态模型用于描述魔方每次旋转后各个色块的移动规律，通过魔方的状态模型可以将魔方的每次操作、每种状态抽象化，具体的实现方法如下：

[0021] 如说明书附图1所示，魔方具有每一个中心块与每一个面的绝对位置关系唯一确定的特性，该特性是分析魔方并确定魔方打乱状态后的棱块、角块位置的重要参考标准。根据这一特性，构建以魔方中心O为坐标系原点的右手系空间直角坐标系XYZ，坐标轴通过各个面的中心块，各个面上的各个色块均可由该空间内的点{(x,y,z)|x,y,z∈{-3,-2,0,2,3}}唯一确定，并且色块的颜色即f(x,y,z)。

[0022] 六个面可分别用X,Y,Z标号，即X＝±3,Y＝±3,Z＝±3，若定义目标面标号为A，则相对面标号为-A，相邻面为非目标面、非相对面。魔方三类色块的坐标也满足一定特点：中心块坐标满足(X,0,0)or(0,Y,0)or(0,0,Z)，角块坐标满足{(X,y,z)or(x,Y,z)or(x,y,Z)||x|＝2,|y|＝2,|z|＝2}，棱块则为非中心块、角块的色块。

[0023] 在本发明定义的魔方状态模型中，对魔方只需要12种基本操作方法，即各个面的顺时针90°旋转以及各个面的逆时针90°旋转。由于色块即为空间中的点，因此对魔方的旋转可以转化为空间中点的旋转，也就得到魔方旋转与魔方状态的映射关系：

[0024] 设魔方旋转前色块坐标为(x,y,z)，顺时针90°旋转后色块坐标为(x′,y′,z′)，逆时针90°旋转后色块坐标为(x″,y″,z″)，需要计算的色块坐标集合为P

[0025] 对面X＝±3旋转：P＝{(x,y,z)|x＝±3,±2}

[0026]
$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \qquad \begin{bmatrix} x'' \\ y'' \\ z'' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{1}$$

[0027] 对面Y＝±3旋转：P＝{(x,y,z)|y＝±3,±2}

[0028]
$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \qquad \begin{bmatrix} x'' \\ y'' \\ z'' \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{2}$$

[0029] 对面Z＝±3旋转:P＝{(x,y,z)|z＝±3,±2}

[0030] $$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \qquad \begin{bmatrix} x'' \\ y'' \\ z'' \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{3}$$

[0031] 旋转后

[0032] g(x′,y′,z′)＝f(x,y,z),g(x″,y″,z″)＝f(x,y,z) (4)

[0033] 最终形成的$f(x,y,z) \leftrightarrow g(x,y,z)$即为旋转前后的魔方状态变化关系。

[0034] 魔方的状态即魔方各个面中所有色块的分布,此分布具有一个重要性质:魔方一旦建立,不论如何打乱各面的颜色,魔方复原后各面色块的分布是相同的。由此产生推论,魔方建立后的任何状态都可以唯一确定这个魔方。为节约商业成本,本发明仅采用单一摄像头,单次拍摄可以获取3个面的色块信息,通过对魔方进行12次旋转、4次拍摄获取魔方各个面的色块分布,即魔方的状态。如说明书附图4所示,魔方状态的获取过程可以划分为两个步骤,步骤一是基于单目视觉识别的魔方色块提取,步骤二是基于状态模型的魔方色块合成。其中,步骤一可细化为图像去噪、轮廓识别、网格划分、色彩比对这四步;步骤二可细化为魔方执行规定动作、摄像头取样、色块提取、合成状态模型这四步。

[0035] 图像去噪步骤包含对图像进行的两次滤波过程。首先使用双边滤波对摄像头获取到的魔方照片进行处理。双边滤波是一种非线性滤波器,它可以达到保持边缘、降噪平滑的效果,使用该滤波器可以在降噪的同时保存魔方的边缘,为后面的边缘检测、网格划分提供便利。在双边滤波结束后,继续对图片进行非局部平均降噪。该降噪与前面的双边滤波不同的是,该算法利用整幅图像来进行降噪,在图像中以块为单位寻找相似区域,然后对相似区域的颜色求平均值。经过图像去噪步骤后干扰魔方识别的背景噪点会被精准的过滤掉。

[0036] 双边滤波具体算法如下:

[0037] 设(i,j)为摄像头采集到的魔方图像中的某个像素点坐标,S则为(i,j)的相邻像素区域,其中(k,l)∈S。经过双边滤波器后,(i,j)点输出的像素值g(i,j)可以表达为

[0038] $$g(i,j) = \frac{\sum_{(k,l)\in S} f(k,l)\omega(i,j,k,l)}{\sum_{(k,l)\in S} \omega(i,j,k,l)} \tag{5}$$

[0039] 其中

[0040] $$\omega(i,j,k,l) = \exp\left(-\frac{(i-k)^2+(j-l)^2}{2\sigma_d^2} - \frac{\|f(i,j)-f(k,l)\|^2}{2\sigma_r^2}\right) \tag{6}$$

[0041] 对图像中的每一个像素点(i,j)求取g(i,j)即完成双边滤波过程。

[0042] 在本算法的实现过程,需要设定三个参数的取值,分别为决定像素邻域S大小的邻域直径d,本发明中设定为9较为合适;决定坐标空间范围的σ_d,该参数数值越大,意味着越远的像素会相互影响,从而使更大的区域足够相似的颜色获取相同的颜色,本发明中设定为300较为合适;决定颜色空间范围的σ_r,该参数的值越大,就表明该像素邻域内有更宽广的颜色会被混合到一起,产生较大的半相等颜色区域,本发明中设定为300较为合适。

[0043] 非局部平均降噪具体算法如下：

[0044] 设经过双边滤波后的魔方图像中以x为中心的某个像素块为搜索窗口I,以y为中心的某个像素块为邻域窗口V。本算法基本过程是邻域窗口在搜索窗口中滑动,通过计算两个邻域窗口间的相似程度为y赋以权值ω(x,y),具体计算方法如下：

[0045] $$\omega(x,y)=\frac{1}{Z(x)}\exp(-\frac{\|V(x)-V(y)\|^2}{h^2}) \quad (7)$$

[0046] 其中

[0047] $$Z(x)=\sum_y \exp(-\frac{\|V(x)-V(y)\|^2}{h^2}) \quad (8)$$

[0048] 设邻域窗的像素值为v(y),去噪后图像的像素值为g(x),则：

[0049] $$g(x)=\sum_{y\in I}\omega(x,y)v(y) \quad (9)$$

[0050] 对图像中的每一个像素点x求取g(x)即完成非局部平均降噪过程。

[0051] 在本算法的实现过程,需要设定三个参数的取值,分别为平滑参数h,该参数控制高斯函数的衰减程度,越大高斯函数变化越平缓,去噪水平越高,但同时也会导致图像越模糊,越小,边缘细节成分保持得越多,但会残留过多的噪声点,本发明中的具体取值为10较为合适;邻域窗口V的大小,本发明中设定为7较为合适;搜索窗口I的大小,本发明中设定为21较为合适。

[0052] 边缘识别步骤包含梯度计算、梯度二值化、边界点定位三个过程。

[0053] 图像边缘的像素点梯度值会趋于一个较大的数值,因此可以通过求取图像各像素点的梯度来判断图像的边缘。

[0054] 梯度计算公式如下：

[0055] $$G=\sqrt{G_x^2+G_y^2} \quad (10)$$

[0056] $$tan(\theta)=\frac{G_x}{G_y} \quad (11)$$

[0057] 其中G_x为沿x方向导数的模,G_y为沿y方向导数的模,G为梯度的模,θ为梯度方向。

[0058] 但考虑设备上微型电脑的算力,本发明采用简化梯度模的计算公式：

[0059] $G=|G_x|+|G_y|$ (12)

[0060] 同样考虑设备算力,本发明采用Sobel算子,即：

[0061] $$S_x=\begin{bmatrix}-1&0&1\\-2&0&2\\-1&0&1\end{bmatrix},\ S_y=\begin{bmatrix}1&2&1\\0&0&0\\-1&-2&-1\end{bmatrix} \quad (13)$$

[0062] 则像素点(x,y)的梯度的模G_p可计算为：

[0063] $G_p=|S_x*A|+|S_y*A|$ (14)

[0064] 其中A为以像素点(x,y)为中心的像素块。

[0065] 将梯度映射至边缘可称为梯度二值化。设像素点x的梯度的模为G_p,梯度方向为θ,是边缘的梯度必须同时满足以下三个条件：

[0066] (1)如果沿θ方向的相邻像素点梯度为G_{p1},沿π-θ方向的相邻像素点梯度为G_{p2},满足$G_p \geq G_{p1}$且$G_p \geq G_{p2}$则为轮廓边缘,否则抛弃该梯度值。

[0067] (2)设置高阈值T_H及低阈值T_L,如果$G_p \geq T_H$则G_p为强边缘,如果$T_L \leq G_p \leq T_H$则G_p为弱边缘,如果$G_p \leq T_L$则抛弃该梯度值。

[0068] (3)如果G_p属于弱边缘,则检测其周围是否存在强边缘,如果存在则G_p为强边缘,否则抛弃该梯度值。

[0069] 如果像素点x满足上述条件,则为边缘,此点像素值g(x,y)=255,否则此点像素值g(x,y)=0。

[0070] 在本算法的实现过程,需要设定两个参数即高阈值T_H和低阈值T_L的取值,该参数共同决定了像素点成为边缘的难易程度,如果T_L和T_H均较大,则只会有很少的像素点成为边缘,反之则会有较多的像素点成为边缘。本发明中T_L设定为50,T_H设定为100较为合适。

[0071] 为了提高色块提取效率,本发明摄像头安装于魔方对角侧,可以同时获取魔方三个面的图像,通过边界点提取算法,定位魔方六个边界点的位置,从而便于对其进行网格划分。为了提高算法的普适性,本发明假定魔方可能位于摄像头取景框内任何位置,如说明书附图2所示,所需提取的边界点按逆时针顺序依次为A、B、C、D、E、F。

[0072] 边界点提取算法通过l_A、l_B、l_C、l_D、l_E、l_F六条直线逼近轮廓边缘,各边界点满足以下数学条件:

[0073] 设$U=\{(x,y)\,|\,g(x,y)=255\}$则

[0074] $A=\{(x,y)\,|\,y=\min(y_1,\ldots,y_j),(i,j)\in U\}$

[0075] $B=\{(x,y)\,|\,y-k_Bx=\min(y_1-k_Bx_1,\ldots,y_j-k_Bx_i),(i,j)\in U\}$

[0076] $C=\{(x,y)\,|\,y-k_Cx=\max(y_1-k_Cx_1,\ldots,y_j-k_Cx_i),(i,j)\in U\}$

[0077] $D=\{(x,y)\,|\,y=\max(y_1,\ldots,y_j),(i,j)\in U\}$

[0078] $E=\{(x,y)\,|\,y-k_Ex=\max(y_1-k_Ex_1,\ldots,y_j-k_Ex_i),(i,j)\in U\}$

[0079] $F=\{(x,y)\,|\,y-k_Fx=\min(y_1-k_Fx_1,\ldots,y_j-k_Fx_i),(i,j)\in U\}$

[0080] 其中k_B、k_C、k_E、k_F为逼近线l_B、l_C、l_E、l_F的斜率,需满足条件:$k_B,k_E<0,k_C,k_F>0$,一般取值为$k_B,k_E=-2,k_C,k_F=2$,此斜率取值与魔方的摆放姿态相关。

[0081] 具体实现过程,本算法提供两种搜索方式,分别为多边形逼近搜索和水平扫描搜索,根据搜索效率选用原则如下:

[0082] 设$\sum_{i=1}^{6} S_i$为多边形逼近重复搜索面积,S_{abcdef}为多边形逼近未搜索面积,如果$\sum_{i=1}^{6} S_i \leq S_{abcdef}$则选用多边形逼近搜索,否则选用水平扫描搜索。

[0083] 多边形逼近搜索的具体实现如下:

[0084] 逼近直线l_A、l_B、l_C、l_D、l_E、l_F的解析式分别为:

[0085] $l_A:f(x)=b$

[0086] $l_B:f(x)=k_Bx+b$

[0087] $l_C:f(x)=k_Cx+H-b$

[0088] $l_D:f(x)=H-b$

[0089] $l_E:f(x)=k_E(W-x)+(H-k_EW-b)$

[0090] $l_F: f(x) = k_F(W-x) + (-k_E W+b)$

[0091] 其中W为图像的宽度，H为图像的高度，b控制搜索范围，一般b>max(W,H)即可满足搜索要求。

[0092] 伪代码如下

```
Begin
    for i ← 0 to 5
        flag ← 0
        do for b ←     0 to H
            if  flag =1
                then break
            do  for x ← 0 to W
                do  y ← f(x)
                if  i=0 or i=3
                    then if y<0 or y> H
                                then break
                if  i=1 or i=5
                    then if y> H
                                then continue
                            if y<0
                                then break
                if  i=2 or i=4
                    then if y> H
                                then break
                            if y<0
                                then continue
                if  g(x , y) > 0
                    then save this point
                    flag ← 1
                    break
End
```

[0093]

[0094] 水平扫描搜索的具体实现伪代码如下：

```
Begin
    for y ← 0 to H
        do for x ← 0 to W
            do if g(x , y) > 0
                    then for i ← 0 to 5
                        do now[i] ← y - k[i] * x
                        if i=0 or i=1 or i=5
                            then     if now[i] > last[i]
                                then last[i] ← now[i]
                                update point
                        if i=2 or i=3 or i=4
                            then     if now[i] < last[i]
                                then last[i] ← now[i]
                                update point
End
```

[0095]

[0096]

[0097] 网格划分步骤包含基向量计算、色块中心点定位、图像取样三个过程。

[0098] 如说明书附图3所示，魔方各色块的位置均可由$\overrightarrow{V_1}$、$\overrightarrow{V_2}$、$\overrightarrow{V_3}$线性导出，即$\overrightarrow{V_1}$、$\overrightarrow{V_2}$、$\overrightarrow{V_3}$为魔方空间的基向量。六个边界点，可以确定两组重复的基向量，利用这个特点可以矫正边界点定位误差，具体算法如下：

[0099] $$\overrightarrow{V_1}=\begin{bmatrix}\frac{x_B+x_D-x_A-x_E}{6}\\ \frac{y_B+y_D-y_A-y_E}{6}\end{bmatrix}\quad \overrightarrow{V_2}=\begin{bmatrix}\frac{x_F+x_D-x_A-x_C}{6}\\ \frac{y_F+y_D-y_A-y_C}{6}\end{bmatrix}\quad \overrightarrow{V_3}=\begin{bmatrix}\frac{x_C+x_E-x_B-x_F}{6}\\ \frac{y_B+y_D-y_A-y_E}{6}\end{bmatrix}\tag{15}$$

[0100] ABGF面各色块中心点计算公式：

[0101] $$\begin{bmatrix}x_i\\ y_j\end{bmatrix}=\begin{bmatrix}x_A\\ y_A\end{bmatrix}+(i+p)\overrightarrow{V_1}+(j+q)\overrightarrow{V_2}\tag{16}$$

[0102] BCDG面各色块中心点计算公式：

[0103] $$\begin{bmatrix}x_i\\ y_j\end{bmatrix}=\begin{bmatrix}x_B\\ y_B\end{bmatrix}+(i+p)\overrightarrow{V_3}+(j+q)\overrightarrow{V_2}\tag{17}$$

[0104] GDEF面各色块中心点计算公式：

[0105] $$\begin{bmatrix}x_i\\ y_j\end{bmatrix}=\begin{bmatrix}x_F\\ y_F\end{bmatrix}+(i+p)\overrightarrow{V_3}+(2\text{-}j+q)\overrightarrow{V_1}\tag{18}$$

[0106] 其中p为i方向上的坐标偏移值，q为j方向上的坐标偏移值，标准魔方p＝0.5，q＝0.5，如果网格划分有错位，可以微调p和q的数值修正网格划分。

[0107] 根据中心点坐标(x_i,y_j)截取中心矩形区域，即完成对色块的取样。

[0108] 色彩对比步骤包含色彩校正、色彩空间二值化、可能性计算三个过程。

[0109] 由于魔方色块颜色识别的准确率直接影响魔方的求解，通过色彩校正可以极大提高颜色识别的准确率，具体方法如下：

[0110] 在摄像头固定位置放置比色卡，比色卡包含标准魔方所需的六种颜色，每次拍摄时比色卡与魔方处于相同的光照环境，因此可以直接从比色卡相应区域提取色彩均值T_M，则色彩空间二值化的高阈值T_H及低阈值T_L可以表达为：

[0111] $T_H = T_M + R$ (19)

[0112] $T_L = T_M - R$ (20)

[0113] 其中R为阈值半径，半径越小色彩识别的容差就越小，本发明HSV色彩空间中色相H的阈值半径设置为2，饱和度S和明度V的阈值半径设置为50。

[0114] 色彩空间二值化即每个像素点(x,y)判断其HSV值是否位于高低阈值内，如果位于高低阈值内，则此点像素值g(x,y)＝255，否则此点像素值g(x,y)＝0，具体表达如下：

[0115] $$\begin{cases} g(x,y) = 255, T_L \le HSV \le T_H \\ g(x,y) = 0, (HSV > T_H) \text{or} (HSV < T_L) \end{cases} \quad (21)$$

[0116] 对六种颜色分别进行色彩空间二值化得到图像矩阵M_i(i＝0,1,2,3,4,5)，每种颜色的可能性则可由矩阵的和S_i＝sum(M_i)表示，使得矩阵和S_i最大的那种颜色即为可能性最大的颜色，作为色块颜色识别的最终结果。

[0117] 根据魔方的状态模型，对魔方进行的任何一种基本操作都可以根据操作前魔方的状态得到操作后魔方的状态，因此获取魔方最初的状态就成为复原魔方的关键。如说明书附图5所示，本发明提出12步操作方法，对魔方进行4次取样即可完整获取魔方的初始状态，以下是具体的实现方法：

[0118] 摄像头位于魔方的对角线上，可以一次拍摄魔方的三个面，由上文提出的色块提取方法，可以一次性获取三个面上所有色块的颜色。魔方初始状态如附图中状态(1)所示，第一次拍摄取样可以获取色块{(x,y,z)|(x＝3)or(y＝3)or(z＝3)}上的颜色。第一、二步操作同时对面Z＝±3连续两次顺时针旋转90°得到状态(2)，此时进行第二次拍摄取样，可以获取色块{(x′,y′,z′)|x′＝3,z′＝±2}以及{(x′,y′,z′)|y′＝3,z′＝±2}上的颜色；第三、四步操作继续对面Z＝±3连续两次顺时针旋转90°回到状态(1)；第五、六步操作对X＝±3连续两次顺时针旋转90°得到状态(3)，此时进行第三次拍摄取样，可以获取色块{(x′,y′,z′)|z′＝3,x′＝±2}以及{(x′,y′,z′)|y′＝3,x′＝±2}上的颜色；第七、八步操作继续对面X＝±3连续两次顺时针旋转90°回到状态(1)；第九、十步操作对Y＝±3连续两次顺时针旋转90°得到状态(4)，此时进行第四次拍摄取样，可以获取色块{(x′,y′,z′)|z′＝3,y′＝±2}以及{(x′,y′,z′)|x′＝3,y′＝±2}上的颜色；第十一、十二步操作继续对面X＝±3连续两次顺时针旋转90°回到状态(1)。根据上文所述的状态模型对所有的旋转后色块(x′,y′,z′)逆向转换即可得到旋转前即状态(1)中色块(x,y,z)的颜色f(x,y,z)。至此，魔方54个色块已经获取51个色块的颜色，剩余3个无法获取的色块颜色为三个面的中心块颜色，根据魔方复原原理，魔方只需要获得所有角块、棱块的颜色即可，因此本方法满足魔方复原的要求。

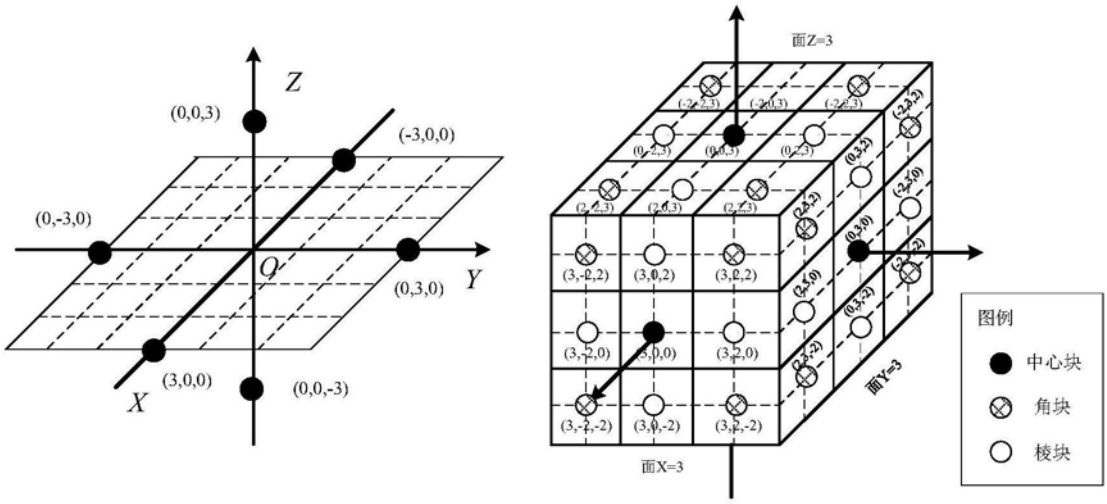

Z
(0,0,3)
(-3,0,0)
(0,-3,0)
O
Y
(0,3,0)
X
(3,0,0)
(0,0,-3)
面Z=3
面Y=3
面X=3
图例
中心块
角块
棱块

图1

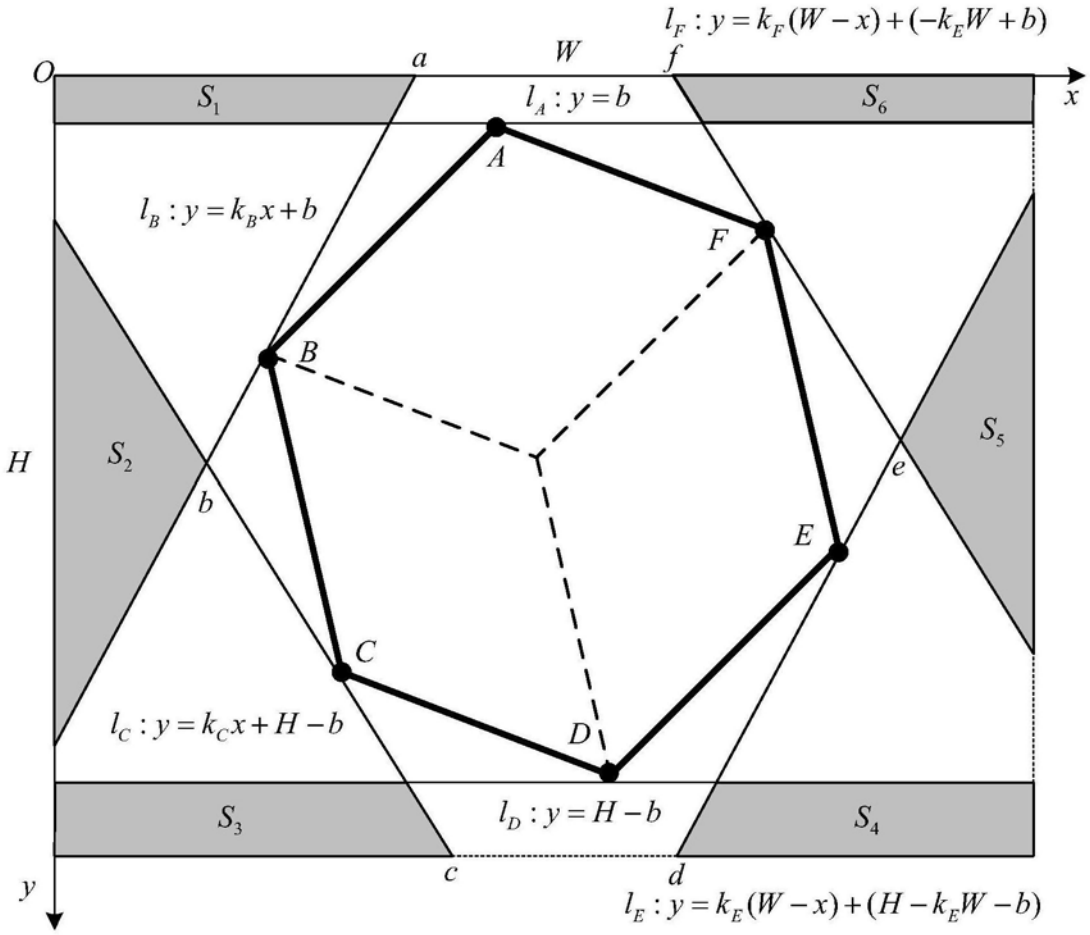

$l_F : y = k_F(W - x) + (-k_E W + b)$
O
a
W
f
x
S_1
$l_A : y = b$
S_6
A
$l_B : y = k_B x + b$
F
B
H
S_2
S_5
b
e
E
C
$l_C : y = k_C x + H - b$
D
S_3
$l_D : y = H - b$
S_4
y
c
d
$l_E : y = k_E(W - x) + (H - k_E W - b)$

图2

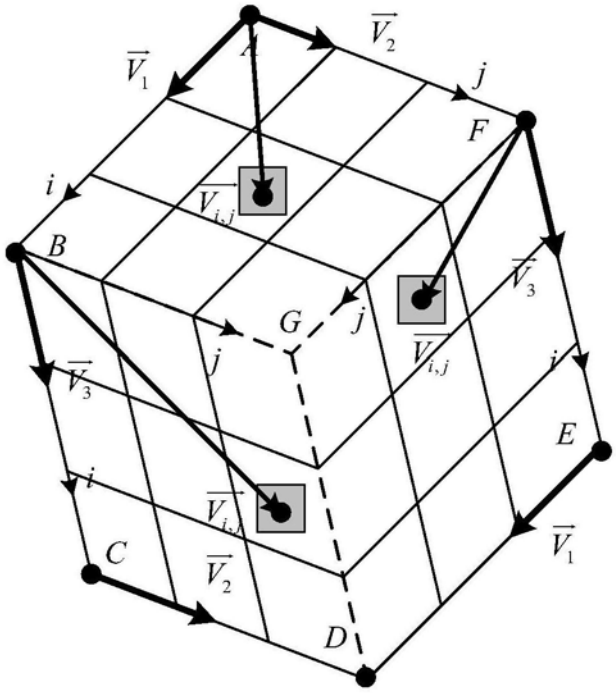

A
$\vec{V}_2$
$\vec{V}_1$
j
F
i
$\vec{V}_{i,j}$
B
G
j
$\vec{V}_3$
$\vec{V}_{i,j}$
$\vec{V}_3$
j
i
E
$\vec{V}_{i,j}$
i
C
$\vec{V}_2$
$\vec{V}_1$
D

图3

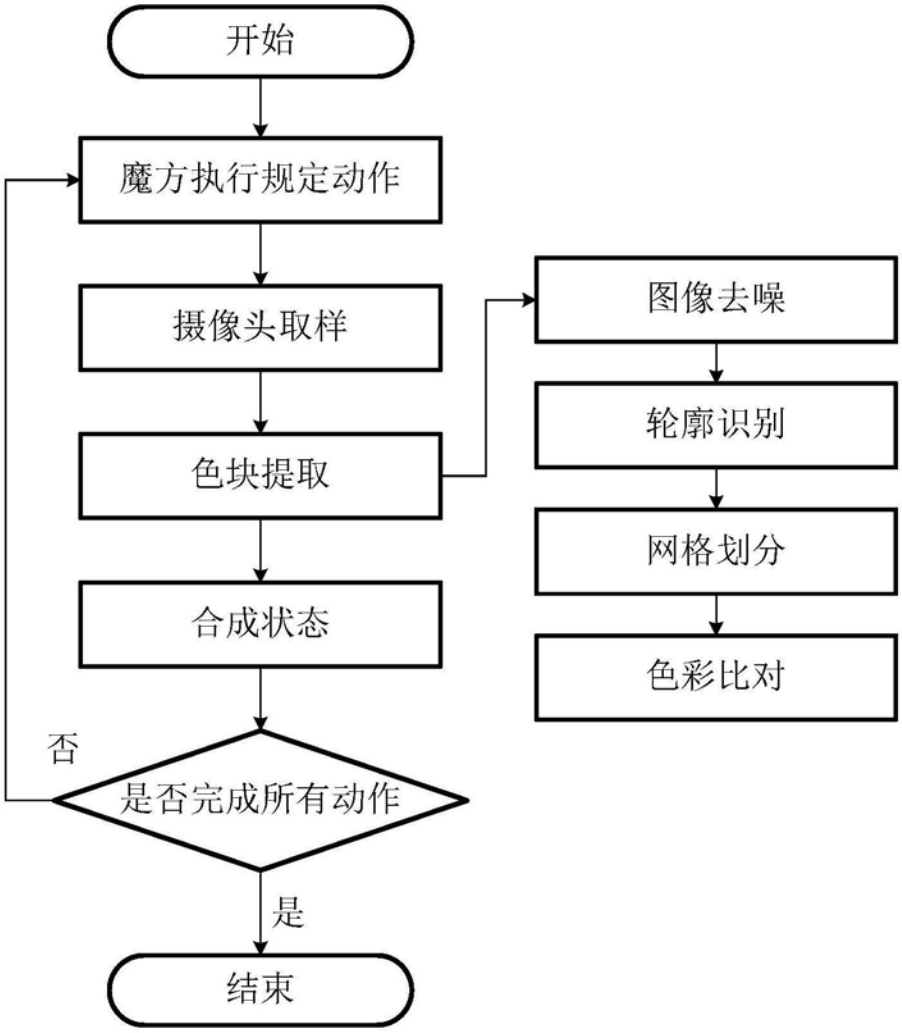

开始
魔方执行规定动作
摄像头取样
色块提取
合成状态
否
是否完成所有动作
是
结束
图像去噪
轮廓识别
网格划分
色彩比对

图4

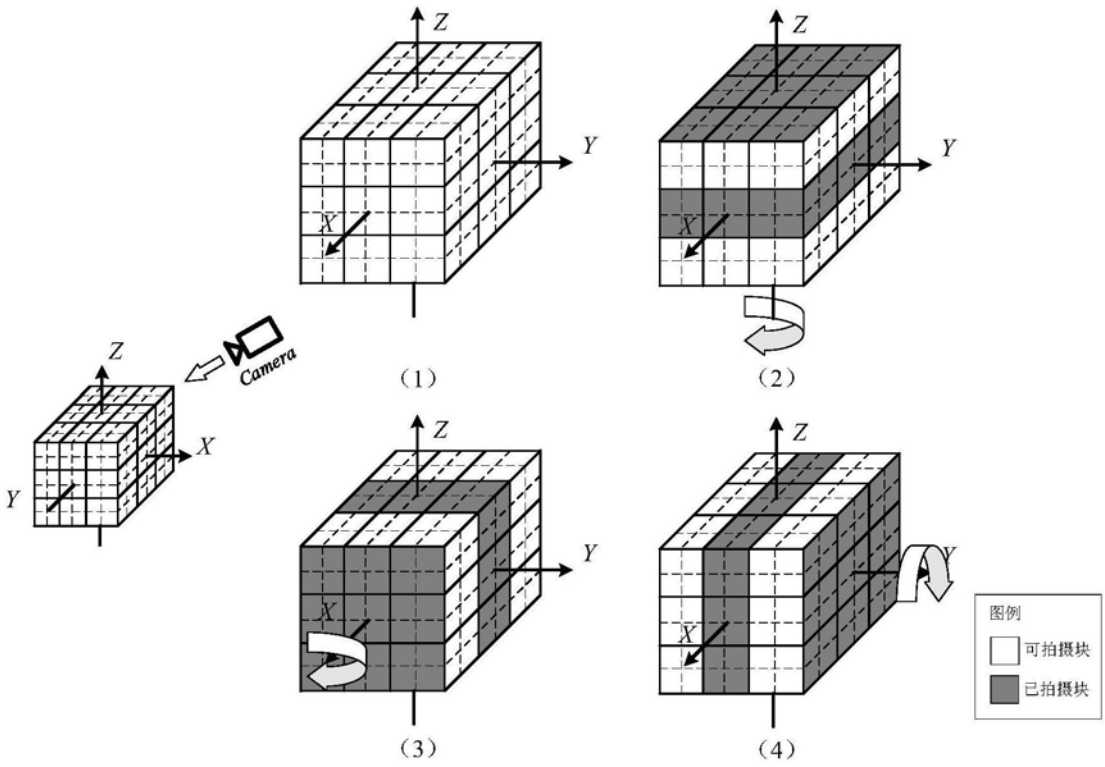


图5