

**本科生毕业论文（设计）**

基于视觉伺服控制的转魔方机械装置设计与实现

**Design and implementation of Rubik’s cube turning mechanism based on visual servo control**

学生姓名： 吴小末

指导教师： 胡标副教授

专业名称： 机械电子工程

所在学院： 工学院

2022年 5 月

# 摘 要

魔方自上世纪七十年代被发明出来之后，以其独特的魅力迅速在全球流行。时至今日，人类挑战三阶魔方，世界纪录已经达到3.47s。自从李世石、柯洁等人类棋手在传统围棋项目比赛中输给“Alpha go”之后，人类对人工智能的探索愈发深入，而本文设计旨在打造一款稳定性高、还原速度快的转魔方机械装置。

本文通过对国内外解魔方机器人的研究，对比他们的优缺点，最终决定整体结构采用STM32单片机为主控，安卓手机为辅助控制的搭建方案。双控制器之间通过蓝牙通讯模块实时交互，可以实现控制指令快速传递，是整个视觉伺服控制的基础。

较之传统的魔方还原平台，本设计在视觉伺服控制系统的设计过程中，用人为设置搜索时间的方法避免了Kociemba还原算法陷入无限搜索以至于耗时较长的状况；在视觉识别算法中摒弃了传统的阈值法，并引入k-means聚类算法，将魔方六个面的中块色片颜色作为聚类中心，同时设置每个类别聚集的数据数量为9，这样大大削减了外界环境光强因素对颜色信息提取的影响，成功提高了视觉识别的准确度；在伺服舵机执行动作流程中，先根据四臂两指型装置的四自由度设计出一套新的魔方优化还原指令序列，在此基础上，引入二叉树模型对舵机的执行动作进行分解判断，将其转换为模型的最短路径问题，对整个系统的运行效率有着重要作用。

最后通过系统测试分析出，本文设计的转魔方机械装置能够在70秒内有效还原被打乱的魔方。综上所述，本文设计的机械装置达到了设计方案的预期效果，魔方还原平台总体上呈现出稳定性与高效性。

**关键词：魔方还原,STM32单片机,视觉识别,Kociemba算法**

# Abstract

The Rubik's cube swept the world in a short time because of it's unique charm after invented in the 1970s. Up to now, the world record of human solving the third-order Rubik’s cube has reached 3.47s. Since human chess players such as Lee Sedol and Jie Ke lost to "Alpha go" in the traditional go competition, the exploration of artificial intelligence has become more and more depth. This paper is designed to create a Rubik’s cube turning mechanical device with high stability and fast reduction speed.

Through the research on the Rubik’s cube solving robots at home and abroad, this paper compares their advantages and shortcomings. Finally, the overall case adopts STM32 microcontroller as the main control and Android mobile phone as the auxiliary control. The dual controllers are able to interact in real time through the Bluetooth communication module, which can realize the rapid transmission of control instructions, the basis of the whole visual servo restoration.

Compared with the traditional Rubik's cube restoration platform, to avoid the situation that Kociemba restoration algorithm falls into infinite search and takes a long time ,this design uses the method of artificially setting the search depth in the design of visual servo restoration system. In the visual recognition algorithm, the K-means clustering algorithm is introduced to replace the traditional threshold method. The six central blocks of the Rubik’s cube are used as the clustering center, which are allocated 9 data each. In this way, the influence of external environmental light intensity on color information extraction is greatly reduced, leading to a great improvement in the accuracy of visual recognition. In the execution process of servo actuator, a new set of Rubik’s cube optimization reduction instruction sequence is designed according to the four degrees of freedom of this device. On this basis, the binary tree model is introduced to judge the execution action of actuator and convert it into the shortest path problem of the model, which plays an important role in the efficiency of the whole system.

Finally, through the system analysis, the Rubik’s cube turning mechanical device designed in this paper can effectively restore the disturbed Rubik’s cube in 70 seconds, which has achieved the expected effect of the design scheme.

**Keywords: Rubik's Restoration,STM32 microcontroller,** **Visual recognition，Kociemba algorithm**

# 目录

[摘要 I](#_Toc103780774)

[Abstract II](#_Toc103780775)

[第一章 绪论 1](#_Toc103780779)

[1.1 研究背景及意义 1](#_Toc103780780)

[1.1.1课题研究背景与目的 1](#_Toc103780781)

[1.2 国内外研究现状 1](#_Toc103780782)

[1.2.1 国内研究现状 2](#_Toc103780783)

[1.2.2 国外研究现状 3](#_Toc103780784)

[1.3 文章主要工作和章节内容安排 3](#_Toc103780785)

[第二章 魔方的数学模型与基本原理 5](#_Toc103780786)

[2.1 魔方的概念描述与符号表达 5](#_Toc103780787)

[2.1.1 魔方的立体结构 5](#_Toc103780788)

[2.1.2 魔方的配色 6](#_Toc103780789)

[2.1.3 魔方的符号表达 6](#_Toc103780790)

[2.2 魔方的数学模型与状态转换 7](#_Toc103780791)

[2.2.2 魔方色片转换表示 8](#_Toc103780792)

[2.2.3 魔方色块转化表示 9](#_Toc103780793)

[2.3 本章总结 11](#_Toc103780794)

[第三章 转魔方机械装置的设计 12](#_Toc103780795)

[3.1系统整体设计规划 12](#_Toc103780796)

[3.1.1 Android手机与STM32单片机方案 12](#_Toc103780797)

[3.1.2 系统设计结构 13](#_Toc103780798)

[3.2 硬件电路与软件设计 15](#_Toc103780799)

[3.2.1 电路设计框图 15](#_Toc103780800)

[3.2.2 XL4015E1稳压电路 16](#_Toc103780801)

[3.2.3软件整体框架设计 17](#_Toc103780802)

[3.3 蓝牙通讯 17](#_Toc103780803)

[3.4 本章总结 18](#_Toc103780804)

[第四章 魔方还原算法的研究 19](#_Toc103780805)

[4.1 经典魔方还原算法研究 19](#_Toc103780806)

[4.1.1 层先法 19](#_Toc103780807)

[4.1.2 CFOP法 19](#_Toc103780808)

[4.1.3 Kociemba还原算法 20](#_Toc103780809)

[4.2 魔方还原算法的选择与优化 21](#_Toc103780810)

[4.2.1 还原算法的选择 21](#_Toc103780811)

[4.2.2 还原算法的优化 21](#_Toc103780812)

[4.3 本章总结 22](#_Toc103780813)

[第五章 转魔方机械装置的算法实现 23](#_Toc103780814)

[5.1魔方特征信息提取 23](#_Toc103780815)

[5.1.1 魔方图像采集 23](#_Toc103780816)

[5.1.2 图像预处理 23](#_Toc103780817)

[5.2 魔方视觉识别 23](#_Toc103780818)

[5.2.1 阈值法 23](#_Toc103780819)

[5.2.2 k-means聚类算法 24](#_Toc103780820)

[5.3 优化还原指令序列 26](#_Toc103780821)

[5.3.1 优化魔方还原指令序列 26](#_Toc103780822)

[5.3.2 二叉树模型优化算法 26](#_Toc103780823)

[5.4 系统测试 27](#_Toc103780824)

[5.5 本章总结 28](#_Toc103780825)

[第六章 结论与展望 29](#_Toc103780826)

[6.1 结论 29](#_Toc103780827)

[6.2 展望 29](#_Toc103780828)

[参考文献 30](#_Toc103780829)

[致谢 32](#_Toc103780830)

[作者简介 33](#_Toc103780831)

# 插图和附表清单

图2-1 魔方的几何结构 5

图2-2 官方配色标准 6

图2-3 棱块与角块的表示 7

图2-4 魔方各面的编号 8

图2-5 魔方F面顺时针旋转90° 9

图2-6 逆置换的置换积 9

图2-7 角块方向变化 10

图2-8 魔方F面顺时针旋转90° 10

图3-1 操作臂 13

图3-2 夹具 13

图3-3 转魔方机械装置设计图 14

图3-4 转魔方机械装置实物图 15

图3-5 魔方还原流程框架图 15

图3-6 电路框图 15

图3-7 稳压电路原理图 16

图3-8 软件系统设计 17

图4-1 层先法步骤 19

图4-2 CFOP法 20

图4-3 两种状态下舵机步数对比 21

图5-1 高斯滤波效果 23

图5-2 二维平面上点的聚类 24

图5-3 k-means算法流程图 25

图5-4 转魔方机械装置的自由度 26

图5-5 二叉树模型 27

图5-6 系统还原时间测试结果 27

表2-1 初始指令序列 7

表2-2 优化指令序列 7

表2-3 魔方置换后色片位置变化 9

表2-4 魔方F面顺时针旋转90°角块表示 10

表2-5 魔方R面顺时针旋转90°角块表示 11

表3-1 蓝牙模块AT指令 18

表4-1 多种算法实验分析 21

表5-1 优化指令序列对应的舵机执行步数 26

# 主要符号和术语表

|  |  |  |
| --- | --- | --- |
| 变量符号 | 变量单位 | 变量名称 |
| T | S | 时间 |
| U | V | 电压 |
| I | A/mA | 电流 |
| R | Ω | 电阻 |

# 第一章 绪论

## 1.1 研究背景及意义

1.1.1课题研究背景与目的

魔方是匈牙利建筑学教授暨雕塑家埃尔诺·鲁比克在1974年发明的机械益智玩具，最开始是用来练习学生的立体结构思维，然而当他将魔方第一次打乱后，却发现很难将其复原。魔方在上世纪80年代最为流行，至今未有颓势，目前世界魔方大赛中三阶魔方的世界纪录为3.47s，由中国学生杜宇生创造。

当今社会科学技术日益进步，高度智能化、集成化的自动化操作平台登上历史舞台，并在各行各业中处于至关重要的地位，如今，世界各国和各地区对自动化智能平台上的研究越来越深入[1]。机器人行业作为自动化智能平台的一个重要组成部分，涉及到机械设计、传感技术、计算机视觉、电气电子、人工智能等等多学科的知识与技术，在各领域都发挥着重要作用。2020年疫情爆发后，我国机器人与人工智能制造都处于被市场急速激发的状态，自动化、智能化制造设备大量投入进市场应用不仅能优化行业生产结构，提高企业竞争力，同时还可以改善用工环境，降低工作风险，形成新的就业机会以提高就业率，因此自动化、智能化制造模式是各国各地区未来发展兴盛的必要趋向。

将魔方这种日常可见的智力型玩具与机器人相结合形成解魔方机器人，无论是理论研究还是参与性、趣味性等方面，都能很全面地考验学生的综合技术水平。首先，解魔方机器人的研究对于自动化、智能化技术有着重要的应用意义，它的构建需要将几个不同的子系统如机械系统、电控系统、视觉伺服系统等组合成一个完整的机械装置，此过程可以为集成化，特别是在研究机电一体化时提供很多有用的经验；其次，解魔方机器人研究对于科技科普启蒙也有着重要影响，魔方机器人将机械设计、计算机视觉、电子理论知识等与实践相结合，将相对来说复杂的解魔方步骤以快速、准确、机器化的方式展现给大众，能够使得大众，特别是青少年学生在接触魔方的同时，启迪他们的智慧，将他们引入科学的大门，这样不仅能够达到研究自动化与智能化技术的目的，还可以将其包装成一种科普产品出现在大众视野中，这对于科教兴国战略、前沿理论知识的普及、全民科学知识文化的提高具有深远的意义。

## 1.2 国内外研究现状

纵观解魔方机器人发展的历史，在大部分的设计中已经提出了一些稳定、高效的解决方案，从1981年克劳德·香农设计出的简单魔方机械手到运动更集成的两臂、四臂、六臂的机械装置，从LEGO MINDSTORMS NXT机器人[2]到Symbian手机、Android手机控制平台，这其中大多数使用了NXT组件，但也有一些被构造成气动机器人[3]。无论使用何种机制，魔方求解过程的原理都是相同的，包括三个阶段：立方体状态检测、生成解和实现特定动作。

1.2.1 国内研究现状

目前，国内研究出的解魔方机器人大部分原理相似，在结构上却各有不同。其中，在驱动模块上主要有气动驱动、伺服驱动等，在机械装置上则有四臂两指、两臂两指等不同的结构设计。

济南大学的田田[4]在前人研究的基础上开发了一款四臂两指型解魔方气动组合机械手[5]，这套方案在视觉采集的过程中选用了传统的CMOS摄像头，从拍摄得到的魔方图像中提取颜色信息，再使用RGB颜色模型与之对比，最后实现魔方各个面的颜色采集分析。以计算机为主控，以TM法为基础，设计出魔方的还原步骤，在下位机则采用低成本的PLC，运用气动驱动技术，以四个气动手爪为主要执行元件，通过TM法设计出的组合运动，控制气动机械手进行魔方复原。

与之原理同源但机械结构有所区分的，还有云南师范大学的刘远法[6]选用Arduino单片机为主控、舵机为驱动来构成的解魔方机器人。在设计中，以Arduino Mega2560作为主控平台，颜色传感器作为主要视觉媒介，在机械结构的设计上采用底板与机械臂配合的结构，达到底板带动机械臂旋转、机械臂前进后退的动作要求。这样既可以让魔方各个面独立翻转和，同时各层的旋转也不会影响到其他面。此外，合肥工业大学张宏平[7]设计了一款双臂两指型解魔方机器人。在整个设计中，张宏平对其机器人机械装置部分的整体结构及手臂的结构的创新进行了详细描述。

北京理工大学的卢桂萍[8]设计的双臂两指型解魔方机器人中，则对于二阶段算法，即Kociemba还原算法进行了深入的探讨研究。传统的一些翻滚式、六轴自由度魔方机器人都是在基于层先法或者CFOF等人类适应的解法基础上来搭建的，识别效率、速度有限，而卢桂萍则基于还原算法对视觉系统和机械传动控制系统做出改变。使用Kociemba算法结合OpenCV获取魔方相关的图像信息不仅能够缩短还原魔方的步数，更能使得颜色分割识别速度、效率大大提升。在机械结构上，采用双气动驱动手爪，适应新算法还原魔方的速度，在机械结构上也更加合理。

1.2.2 国外研究现状

在国外开发的解魔方机器人，Cubinator[9]机器人最早出现在大众视野中。这是一款在2010年世界制造者博览会上进行展出的机器人，它由Pete Redmond设计，这种两臂型解魔方机器人结构可以通过操作臂间的相互配合来实现魔方复原动作。

Jozef Varga、Frantisek Durovsky[10]等人则完成了气动解魔方机器人的设计，此机器人从两爪升级到四爪，并对求解过程优化所需的Thistlethwait的45算法进行了统计分析，成功搭建了结构更为简单的气动机器人。

Vasile Dan, Gabriel Harja[11]等人提出了一种利用PC机和Arduino Due板作为处理单元，构建高级魔方算法求解器(ARCAS)的技术方案。机器人的设计允许即时识别颜色，通过使用4个网络摄像头定位捕捉立方体的每个面。接着利用彩色图像分割，找到置乱魔方的HSV掩码，从而得到图案。得到的图像在特定的C#桌面应用程序中处理，利用Kociemba算法、蒙眼方法Old Pochmann和M2来求解魔方。该方法以一种新颖的方式实现，通过适应机器人的机械结构，来减少求解时间。

## 1.3 文章主要工作和章节内容安排

本文设计了一款可以将处于任意正确状态下的三阶魔方还原的四臂两指型转魔方机械装置。基于传统的解魔方机器人，本设计在此基础上进行了一些整合改善，使得转魔方机械装置在还原魔方的速度、效率上得到了明显的提升，大量的实验结果表明本文设计的转魔方机械装置能够在70秒以内的时间还原绝大多数魔方。

以研究一款转魔方机械装置为基础，现在对本文的主要工作与章节内容进行合理安排：

第一章：主要简单介绍魔方的由来，以及它于自动化、智能化技术结合形成的解魔方机器人对于自动化操作平台技术运用的意义和对于全民科普科教、挺高科学素养的深远影响。其次，基于解魔方机器人这个研究对象，对国内外发展现状进行综述比较，在学习他们优点的同时进行创新改善。

第二章：首先，直观分析了魔方的立体结构，在辛马斯特标记法的基础上对魔方的基本符号表示进行定义，优化了还原魔方的指令序列。同时，对魔方的数学模型进行研究，给出魔方置换后不同的表示方法。总体从魔方的一些特殊性质与原理入手讨论，这对于魔方的还原具有非常重要的意义和作用。

第三章：本章对转魔方机械装置的构建进行了详细描述，从总体方案设计思路到硬件与软件的实现，再将视觉识别结果通过蓝牙通讯的方式与主系统相联结，整体呈现出系统的稳定性与设计的创新性。

第四章：主要介绍了依旧流行与当下的魔方还原的一些典型算法，通过对这些魔方还原算法实验研究，总结出它们各自的优缺点，最后选择基于TM法还原法形成的Kociemba还原算法，也称二阶段算法。此外，针对Kociemba算法的深度搜索特点，本设计通过实验设置搜索时间阈值，极大地减少了因算法运行时间过长带来的时间消耗，总体上提高了魔方还原效率。

第五章：传统的解魔方机器人在视觉识别过程中，经常采用阈值法、滤波法等方法分析提取魔方的颜色信息，这种方法的识别错误率较高，且极易受环境因素的影响，比如魔方的色差、环境的光强等，稳定性较差。本文采用了k-means聚类算法，这种算法在机器学习领域中被广泛运用，能够通过聚类运算能够消除大部分外界环境因素对视觉识别准确度的不利影响，大大提高了视觉识别的准确率，同时也能够让整个转魔方装置系统处于的稳定状态运行。

第六章：对本文的设计流程进行归纳总结，同时在此系统上提出了优化结构、改良算法的展望。

# 第二章 魔方的数学模型与基本原理

从三阶魔方被发明出来至今已经衍化出多种高阶的魔方，在追求解魔方的同时，人们还发明出许多具有奇思妙想的异型魔方，例如：移棱魔方、圆柱魔方、枫叶魔方、不等边魔方等等。尽管如此，三阶魔方始终是最为大众所熟知大众所熟知、研究最多的魔方，想要研究其他种类的魔方，必须先以普通三阶魔方作为根基。所以，本设计将研究对象设定为三阶魔方，并在下文简称为魔方。本章重点讨论魔方的一些特殊性质与原理，这对于魔方的复原具有非常重要的意义和作用。

## 2.1 魔方的概念描述与符号表达

2.1.1 魔方的立体结构

三阶魔方是由一个连接着6个中心块的中心轴以及结构不一的20个方块构成的六面立方体。这20个方块分为两种，一种是位于立方体8个角落位置的角块，一种是位于外层棱上中点的棱块。其中角块绕中心轴做转动，它裸露在外侧的三个面都是不同的颜色。棱块亦是绕中心轴转动，与之对应的是它的两个表面也是不同的颜色。与中心轴连接的6个中心块相对于轴固定不动且只有一种颜色。这些方块组合在中心轴周围，与中心色块连接形成一个完整的魔方，其任何一层都能够单独旋转且不影响到其他的面。当转动发生，54个着色面形成不同的排列组合，魔方的状态也随之改变[12]。其结构如图2-1所示。

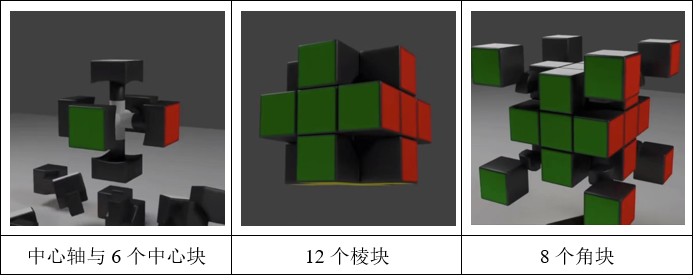


图2-1 魔方的几何结构

2.1.2 魔方的配色

从魔方发明至今，它的配色经历过很多变化，现在大众中所流行的是官方版本配色，这一版本的着色融合了大量色彩研究者的见解，将魔方相对两面的颜色安排为相同色系，如图2-2所示，也就是白色对黄色、红色对橙色、蓝色对绿色，且蓝色、橙色、黄色三色以顺时钟排列[13]。

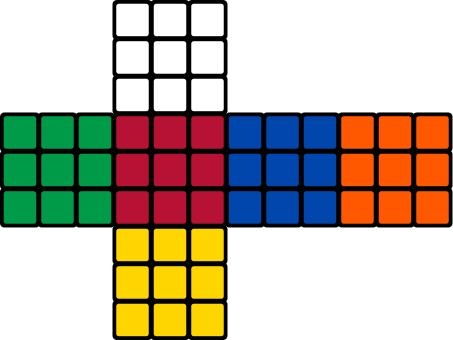


图2-2 官方配色标准

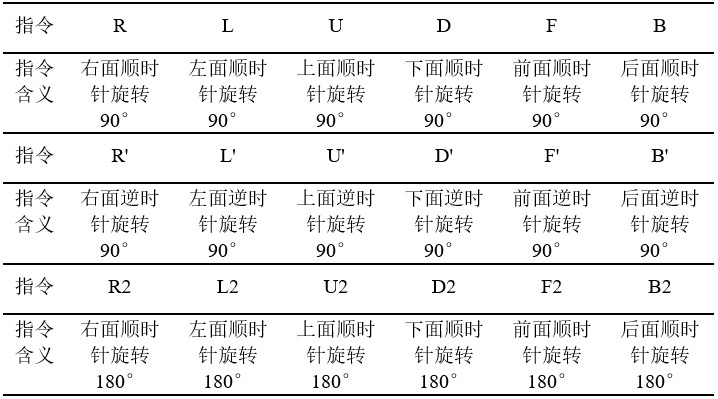
2.1.3 魔方的符号表达

在魔方的符号表达中，辛马斯特发明了一种简洁的魔方旋转记录方法，并很快在世界上流行开来。如今，辛马斯特标记法（Singmaster notation）已经成为世界通用标准，也被称为“魔方公式符号”[14]。

在这种标记法中，主要分为“各层代号”、“旋转方向”两部分。在各层代号表达中，魔方各层以其英文首字母指代，R（Right）、L（Left）、U（Up）、D（Down）、F（Front）、B（Back）分别指代右、左、顶（上）、底（下）、正（前）、背（后）层。在旋转方向的表达中，当魔方的某一个面顺时针旋转90°时，直接写这层的代号；当某一面做逆时针旋转90°时，在这层代号后加上后缀“’”或“\*”；当某一面任意旋转180°时，在这层的代号后加上后缀“2”或“2”（顺时针方向旋转180°与逆时针方向旋转180°均可）。

用这一套方法，可以将魔方的旋转解释为“以面向指代层的视角，按方向进行旋转”。为了魔方研究的方便与简洁，在本文的研究中将出现两套符号指令：初始指令序列，优化指令序列。前者所代表的是以辛马斯特标记法为基础的Kociemba算法解算后得到的魔方指令序列，如表2-1所示。

表2-1 初始指令序列



本文设计的四臂两指型转魔方机械装置可以通过四个夹具扭转对应的四个魔方表面层，这样整个机械装置就拥有四个自由度；但是当人类在手动还原打乱的魔方过程中，自然而然会使用到魔方上、下、左、右、前、后这六个自由度，在这种操作过程中，前文提及的标记法可以涵盖所有的转魔方动作，即初始指令序列。通过对比发现，本文的设计需要将剩余两个自由度的动作分解组合成新的指令，因此在基于（L,R,F,B,L’,R’,B’,L’,L2,R2,F2,B2）的动作序列上，增加了一系列新的优化指令序列，如表2-2所示。

表2-2 优化指令序列

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 指令 | LL | RR | FF | BB |
| 指令含义 | 向左翻转90° | 向右翻转90° | 向前翻转90° | 向后翻转90° |

## 2.2 魔方的数学模型与状态转换

基于魔方的12个棱块与8个角块，本文使用符号来表示其状态。它们的表示优先级依次为上、下、前、后、左、右。例如，如图2-3所示，上面、前面和左面的角块记为UFL，上面与左面的棱块记为UL[15]。

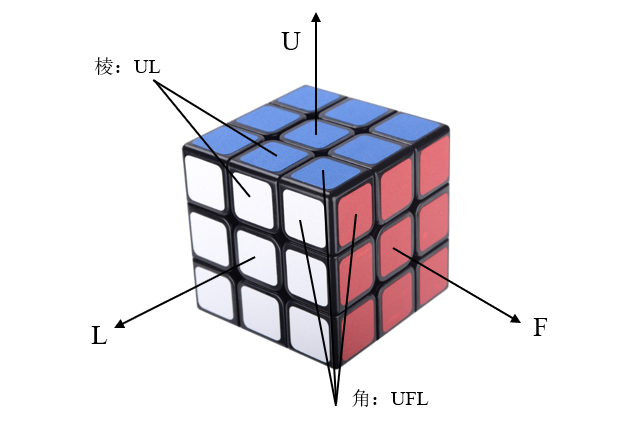


图2-3 棱块与角块的表示

由此得到12个棱块的表示矩阵：

8个角块的表示矩阵：

这样一个处于初始状态的魔方也可以用矩阵来表示：

2.2.2 魔方色片转换表示

当魔方处于还原的初始状态时，可以将其看作由3\*3\*6个色片组成，其编号如图2-4所示。

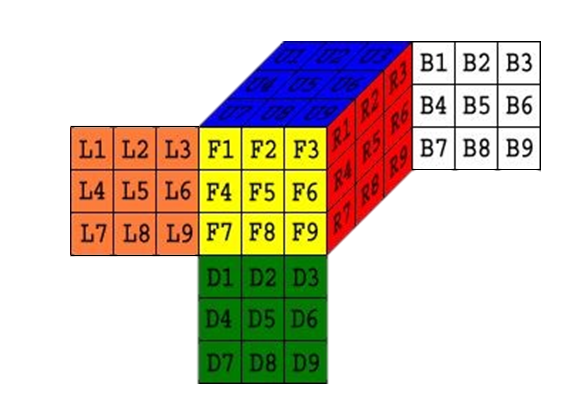


图2-4 魔方各面的编号

每一次转动会将这些色片的位置重排，称为置换[16]。从整个魔方来看，经过旋转所置换的并不是它的各个小面，而是其8个角块与12个棱块。比如现在将图2-4所示的魔方黄色面顺时针旋转90°，即做一次F变换，就可以得到如图2-5所示的魔方状态。经观察黄色面上的色片发现，魔方置换状态如表2-3所示。各个色片的置换表达有以下两种方式：F1被移动到了F3(F1→F3)，F2→F6，F3→F9，F4→F2，F5→F5，F6→F8，F7→F1，F8→F4，F9→F7；F1被替换为了F7(F1←F7)，F2←F4，F3←F1，F4←F8，F5←F5，F6←F2，F7←F9，F8←F6，F9←F3。

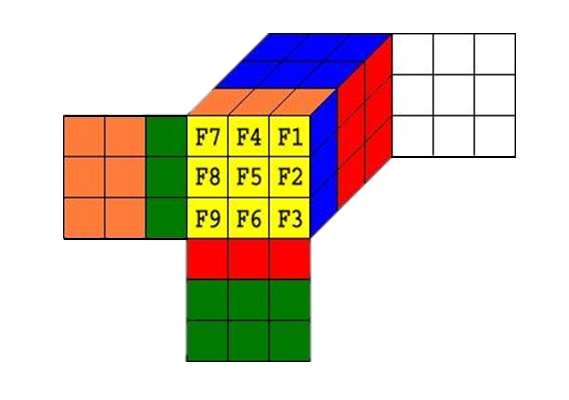


图2-5 魔方F面顺时针旋转90°

表2-3 魔方置换后色片位置变化

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 前 | F1 | F2 | F3 | F4 | F5 | F6 | F7 | F8 | F9 |
| 后 | F3 | F6 | F9 | F2 | F5 | F8 | F1 | F4 | F7 |

已知（F3,F6,F9,F2,F5,F8,F1,F4,F7）是在“被移动到”表示方式下的变化，而（F7,F4,F1,F8,F5,F2,F9,F6,F3）是在“被替换为”表示方式下的变化，那么这两个置换的积则可以表示为F\*F'，也就是F\*F'=1。这样的两个置换成为逆置换，其结果如图2-6所示。

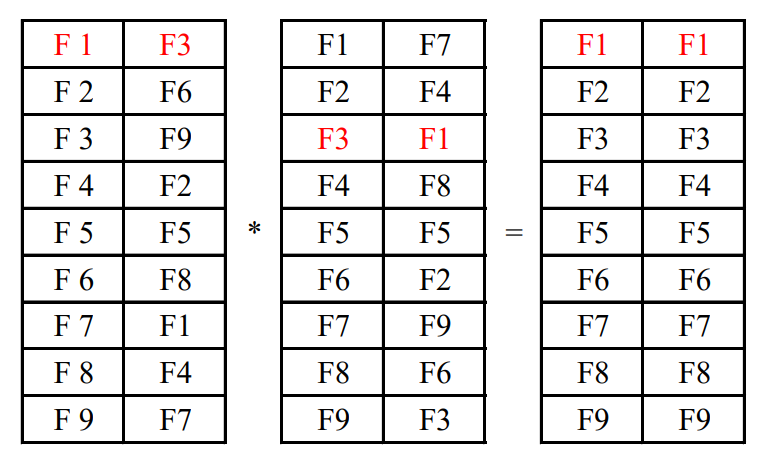


图2-6 逆置换的置换积

2.2.3 魔方色块转化表示

虽然色片表示的方式能在魔方结构展开时直观表达出位置转换状态，但这种表示方式并不合适快速计算。因此本文继续讨论在色块层这一状态下魔方的转换过程，此时研究旋转魔方时置换的对象不是色片，而是8个角块与12个棱块。这种表示方法跟现实中的魔方结构最接近。角块可以在原地扭转，棱块也可以翻转过来。如图2-7所示，这些色块在他们的原位，但是方向却发生了变化。

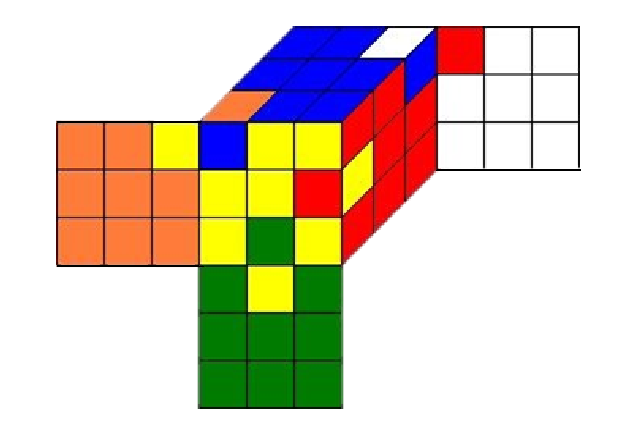


图2-7 角块方向变化

当魔方于初始状态做一次F（前面顺时针旋转90°）旋转时，如图2-8所示，位于 URF和 DLF的角块被顺时针扭转了，位于UFL和DFR的角块被逆时针扭转了，位于UF，DF，FL和FR位置的棱块被翻转了。使用“被替换为”的表示方式来记录位置变化，如表2-4所示上面的例子可以表示为：URF被替换为UFL(URF←UFL)，UFL←DLF，ULB←ULB，UBR←UBR，DFR←URF，DLF←DFR，DBL←DBL，DRB←DRB。用"0"表示角块方向没有变化，用"1"表示角块顺时针的原地扭转，用"2"表示角块逆时针的原地扭转。棱块使用"1"表示棱块被翻转，"0"表示没有被翻转。表2-5展示了魔方右面顺时针旋转90°后的角块表示。

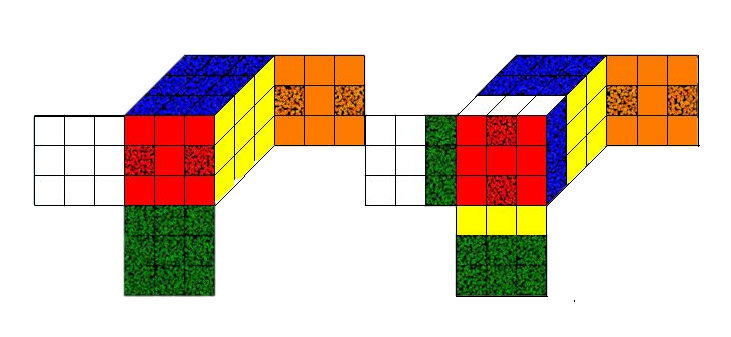


图2-8 魔方F面顺时针旋转90°

表2-4 魔方F面顺时针旋转90°角块表示

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 前 | URF | UFL | ULB | UBR | DRB | DFR | DLF | DBL |
| 后 | c:UFL;o:1 | c:DLF;o:2 | c:ULB;o:0 | c:UBR;o:0 | c:DRB;o:0 | c:URF;o:2 | c:DFR;o:1 | c:DBL;o:0 |

表2-5 魔方R面顺时针旋转90°角块表示

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 前 | URF | UFL | ULB | UBR | DRB | DFR | DLF | DBL |
| 后 | c:DFR;o:2 | c:UFL;o:0 | c:ULB;o:0 | c:URF;o:1 | c:UBR;o:2 | c:DRB;o:1 | c:DLF;o:0 | c:DBL;o:0 |

当F\*R变换作用于UFL角上时，魔方会前面顺时针旋转90°，接着右面旋转顺时针90°，这样8个角块的位置将发生变化。由表2.4中的URF←UFL和表2.5中的UBR←URF可以得出UBR←UFL的结论，即UBR角块被替代为了UFL角块。判断完角块的位置，还需考虑它的方向，在第一步中UFL角块替代URF角块时，o←1。这代表着，从位置UFL移动到位置URF的过程中，角块进行了顺时针原地扭转，这种方向改变增加了从URF移动到UB（UBR←URF）时的方向变化，所以最终角块进行了相对于原位置的逆时针原地扭转。

## 2.3 本章总结

本章主要分析了魔方的数学模型与基本原理。首先对魔方的立体结构特点与官方流行配色进行展开描述，结合其爆炸图直观给出魔方的转动原理与特点。接着，在辛马斯特标记法的基础上，对魔方的层代号与旋转动作进行符号定义，并对初始魔方指令序列进行优化，设计出一套适合四臂两指型机械装置的优化指令，极大地缩短了在研究魔方过程中的步骤。基于此，本文还讨论了魔方状态转换时两种不同的表达方式，通过对比分析，色块表示法比色片表示法更具有代表性。使用色块表示法能够通过置换与逆置换的原理直观呈现出魔方变换后各个角块、棱块的位置与方向，这是对还原算法研究的极大简化。

# 第三章 转魔方机械装置的设计

## 3.1系统整体设计规划

3.1.1 Android手机与STM32单片机方案

STM32是近年来用途十分广泛的单片机，它集成了非常丰富的接口，通信模块以及其他功能模块，处理速度与实时性都很强[17]。同时，STM32拥有大量的用户群体，这些用户能够源源不断地为STM32增加项目应用案例，这对于构建基于STM32软硬件生态非常重要，可以激发大量的用户学习参与。因此，考虑到低成本、高性能、低功耗以及大量的网络学习资源，本文选择STM32单片机作为转魔方机械装置的主控。

考虑到成本与开发的自由度，选择小米2手机作为视觉识别的处理器[18]。该手机后置800万像素的摄像头，运行基于安卓4.4系统[19]，完全适用于本文设计所需的视觉识别处理模块，且手机成本低，性价比极高。在本文的四臂两指型转魔方机械装置设计中，采用STM32单片机与安卓手机结合的方式，能够在手机APP与STM单片机主控间使用蓝牙模块通讯。基于此，使用STM32作为主控、安卓手机作为辅助控制器具有鲜明的优点：

（1）将视觉识别过程及还原算法运算的工程移植到安卓手机上，这样能够让视觉识别与还原算法的开发连成一个整体。安卓手机完成运算后，直接在通过蓝牙通讯模块输出优化指令，可以省去额外的系统串口设计，让整个项目的任务更加集中。

（2）本文设计中转魔方机械装置的操作臂由伺服舵机驱动，在操控魔方还原过程中，控制器的主频不会影响舵机的运行速度，所以采用STM32单片机作为主控与别的高频处理器没有任何区别[20]。此外，考虑到STM32单片机拥有成熟的软件库、直观简明的技术文档以及内容全面的教学资料，完全满足于独立自主学习探索的需求，用来控制转魔方机械装置的动作非常合适。

（3）在本设计的软件开发中，Kociemba算法与k-means聚类算法的引入，让算法运行变得复杂。因此，将这些核心算法移植到安卓手机上，能够大幅度节省STM32单片机解算运行的时间，这对于缩短整个系统的运行时间有着重要的作用。

（4）双控制器的选择优化了繁琐的串口设计，本文将设计重点放在蓝牙通讯串口上，极大的简化了整体装置的设计步骤。

因此，采用STM32单片机和安卓手机双控制器的方案是本设计的不二选择。

3.1.2 系统设计结构

由图3-1可知，四臂两指型的转魔方机械装置主要由STM32单片机、伺服舵机、蓝牙通讯模块、安卓手机、稳压模块、操作臂、夹具、基座等组成。下面将一一介绍这些部件的功能：

（1）STM32单片机：整个机械装置的主控，负责控制伺服舵机的转动，完成优化后的动作指令。

（2）安卓手机：负责视觉识别魔方颜色信息，完成魔方还原算法的计算。

（3）伺服舵机：采用MG995型号舵机，负责驱动机械装置进行转动，对STM32单片机发出的PWM波进行反馈，旋转还原指令中设计的角度[21]。

（4）蓝牙通讯模块：在STM32单片机与安卓手机之间构建起通讯，实现信息交互，考虑到性能与成本问题，此设计选用了HC-05的串口模块[22]。

（5）操作臂：如图3-1所示，操作臂与伺服舵机的十字形喇叭形成曲柄连杆结构，通过舵机带动喇叭旋转，可以将动作转换成操作臂的前进与后退两个动作，由此魔方能够实现翻转。

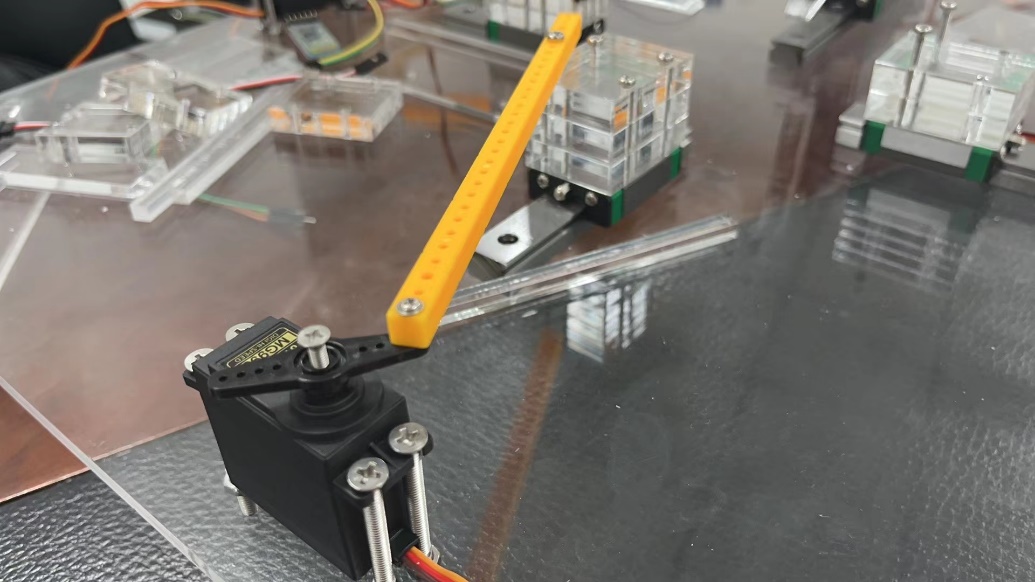


图3-1 操作臂

（6）夹具：如图3-2所示，夹具安装在伺服舵机上，当舵机旋转时带动夹具旋转，并完成扭转魔方一个面的目的。

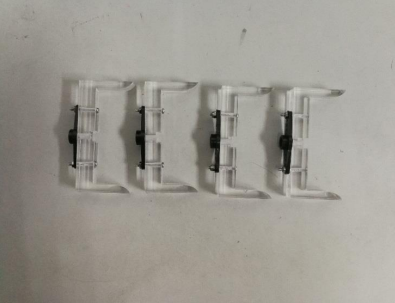


图3-2 夹具

（7）稳压板、锂电池：锂电池能够提供7.4V的电压，通过XL4015E1稳压板的稳压，将电压保持在额定6V，这样能够维持舵机的稳定动作[23]。

（8）基座：保障所有零部件处于正确的位置上，为整个机械装置的平稳运行提供物理支撑。

图3-3是转魔方机械装置的设计图，图3-4是转魔方机械装置的实物图。

图3-5是整个转魔方机械装置的流程框架图与技术路线。在整个流程中，首先由STM32单片机发送指令，使舵机将魔方加紧并处于等待拍照的状态。这里本设计将单片机上的四个按键分别设置为：Key0为松开魔方，四个伺服舵机带动操作臂向后退；Key1为开始拍照，STM32下达指令让舵机完成系列动作；Key2为加紧魔方，四个操作臂向前运动应将魔方夹在操作臂末端的夹具上；Key3为强制操作臂后退，接着停止所有操作臂与夹具的动作。当执行Key1动作时，夹具加紧魔方，等待2秒后，蓝牙给STM32单片机发送指令，控制舵机旋转，使得安卓手机完成魔方6个面的视觉识别。接下来通过调取魔方照片，对图像进行预处理、颜色聚类结算等分析6个面的颜色状态，利用Kociemba还原算法得到初始指令，再进行指令序列的优化。在伺服还原模块中，安卓手机通过蓝牙通讯模块将还原指令发送给STM32单片机，单片机读取分析后再利用二叉树模型优化指令，伺服舵机执行其指令，一步一步还原魔方。

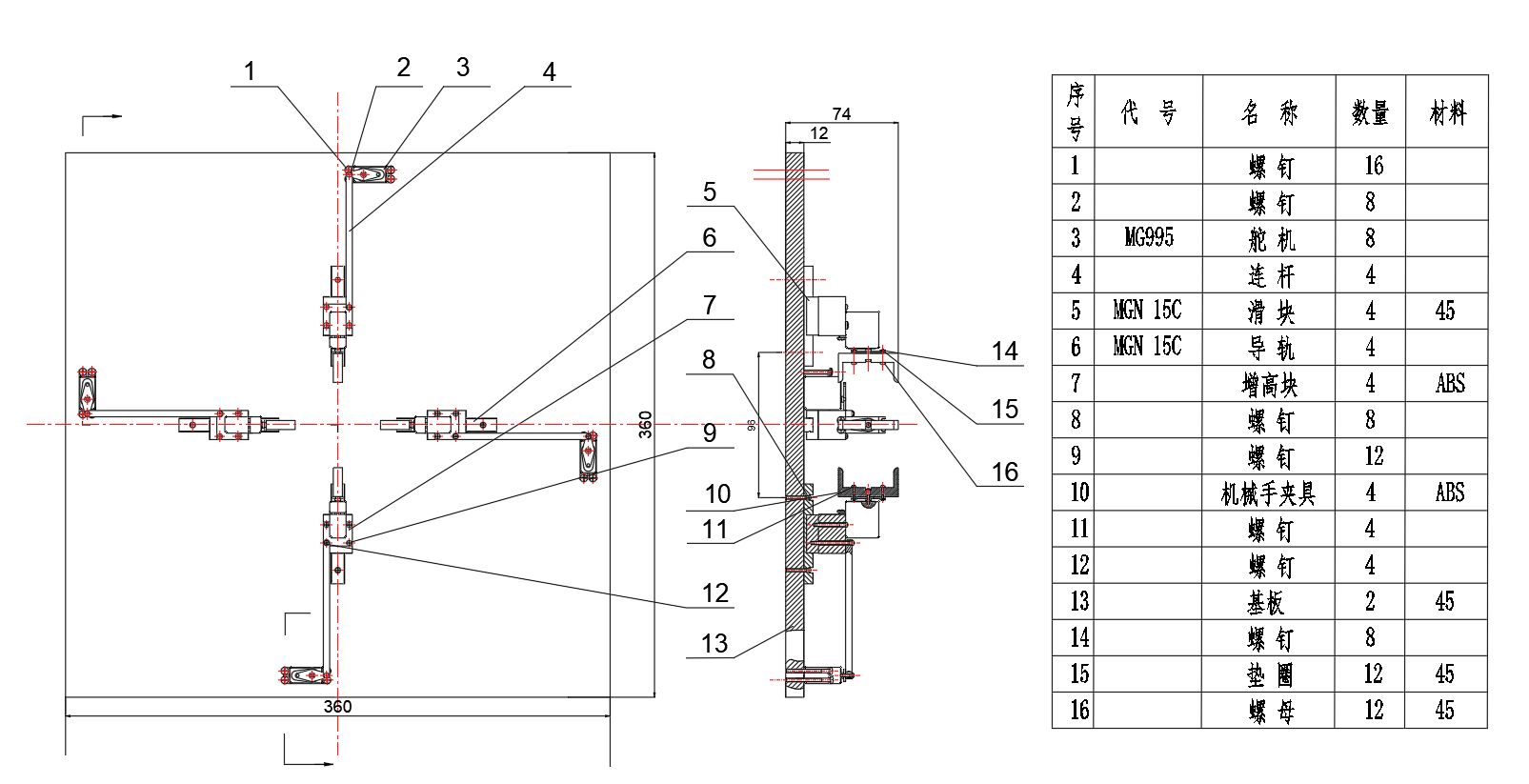


图3-3 转魔方机械装置设计图

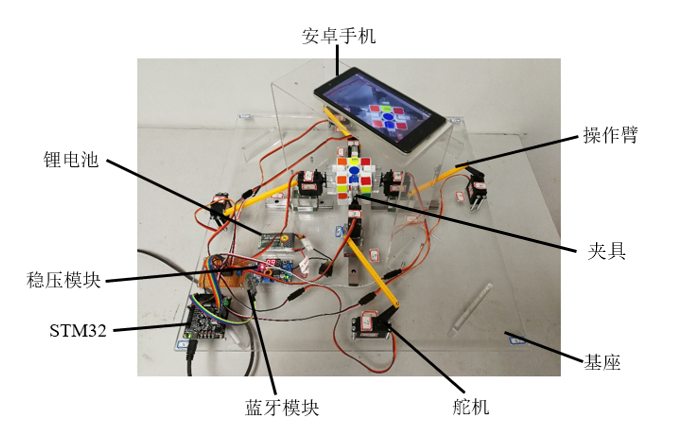


图3-4 转魔方机械装置实物图



图3-5 魔方还原流程框架图

## 3.2 硬件电路与软件设计

3.2.1 电路设计框图

考虑到伺服舵机因电源电压电流问题运行不稳定的因素，本装置专门为舵机设计了一套电源电路：采用7.4V锂电池为8个舵机供电；而STM32单片机则由USB数据线供电。如图3-3所示，此电路系统搭载有两个供电电源，USB数据线连接在电脑上专为STM32单片机供电；蓝牙通讯串口通过杜邦线连接在STM32单片机上，并由此供电；7.4V的锂电池经XL4015E1稳压板获得稳定为6.0V的电压，并为舵机供电。

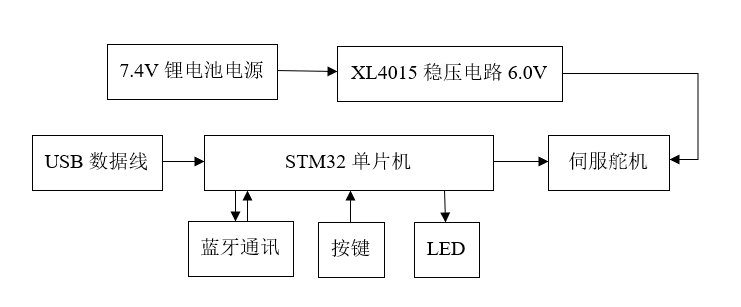


图3-5 电路框图

3.2.2 XL4015E1稳压电路

本文选用MG995伺服舵机搭建末端执行结构，这种舵机所需驱动额定电流大约为500mA，其中4个舵机用于带动操作臂前进与后退，4个舵机用于扭转魔方各面，这样一来同时控制8个舵机则至少需要4A的驱动电流，而舵机的驱动电压为4.8V-6.5V。考虑到舵机电压与电流的需求，本文选用XL4015E1稳压器对7.4V锂电池进行稳压调节，经过电压的调节，可使稳压器最终输出电压保持在系统所需的6.0V，同时输出电流为4A，基本满足了伺服舵机组的驱动电流和电压需求[24]。

XL4015E1稳压器的稳压电路的原理如图3-4所示。从中可以看出，经过各环节调整，电路的最终输出电压为UOUT=1.25\*(1+R1/R2)。其中1.25为参考电压，单位为V，R1的阻值为固定值10K，R2为可调电阻，最大阻值为10K。当输出电压为6.0V时，R2的阻值约为2.6K。

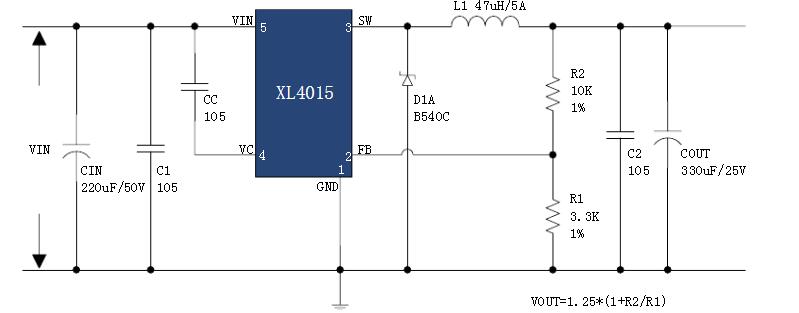


图3-6 稳压电路原理图

3.2.3软件整体框架设计

软件系统的设计包括STM32程序的设计与安卓手机APP程序的设计，软件整体框架如图3-5所示。

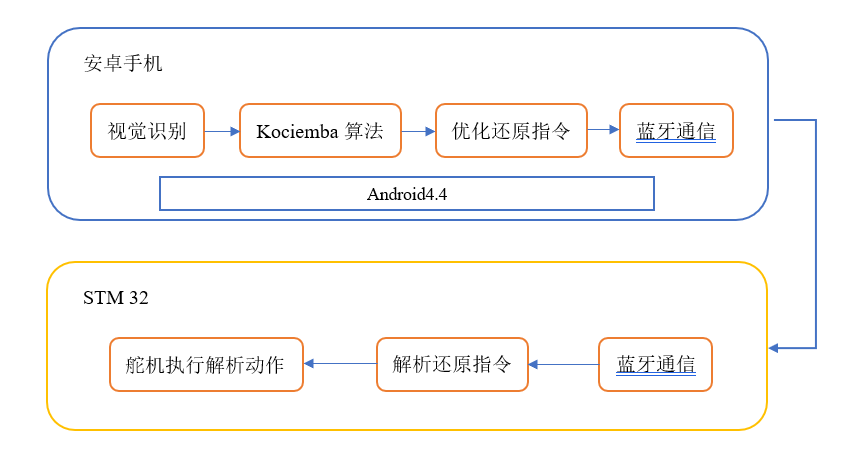


图3-7 软件系统设计

移植到安卓APP的程序主要包括视觉识别程序，Kociemba算法的使用，优化还原指令程序以及蓝牙通信程序等。在视觉识别部分调用了安卓手机的后置摄像头，先依次拍摄魔方的6个面，将拍摄好的照片储存后续再读取分析。此外，为了最大程度排除外界因素的影响，还使用了k-means聚类算法提高视觉识别的准确度。还原算法程序采用了由TM法优化而来的Kociemba还原算法，因其将TM法的四个解算阶段合成为两个阶段，所以也被称为二阶段算法。Kociemba算法是全平台还原魔方所需步数最短的算法，并且运算时间较快。为了对从Kociemba算法得到的还原指令进行优化以适应四个自由度下舵机的运动，程序中使用了二叉树模型进行优化。

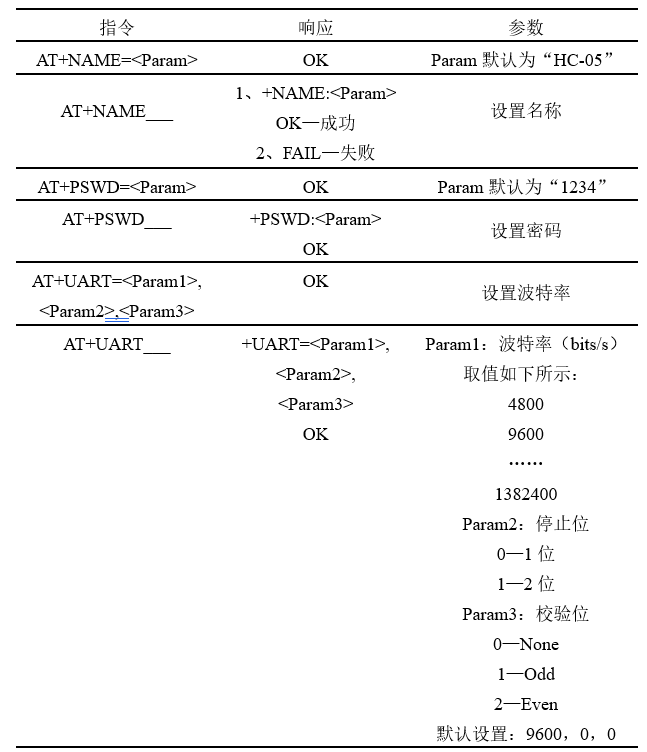
STM32程序主要包括：对从APP传送过来的优化还原指令进行解析，控制舵机组完成相应的执行动作[25]，一步一步完成魔方的还原。

## 3.3 蓝牙通讯

本设计所选取的蓝牙串口模块型号为HC-05，其采用蓝牙2.0通信协议。此模块应用了蓝牙芯片的透传功能，这种功能能够便捷转换蓝牙协议和串口协议，使用者不需要了解蓝牙协议也可以实现数据的无线收发，其也叫做无线串口[26]，非常适用于本文的研究设计。

HC-05模块使用AT指令来设置其参数，例如模块名称，配对密码、波特率等[27]。使用到的AT指令如表3-1所示。

表3-1 蓝牙模块AT指令



## 3.4 本章总结

本章先阐述了选择STM32单片机与安卓手机进行双控制方案的优点，接着对转魔方机械装置整体结构框图进行了详细介绍，讲解了各个零部件的作用以及装置运行过程。此外，对于整个装置的硬件电路设计以及软件编程实现，都做出了直观清楚的分析。

# 第四章 魔方还原算法的研究

## 4.1 经典魔方还原算法研究

4.1.1 层先法

层先法是三阶魔方的入门解法，也是大多数初学者接触魔方时使用的解法。这种魔方还原方法符合人类的思考方式，可以按照思维流程分为6个步骤[28]，如图4-1所示，每个步骤代表着各自所需要实现的目标。

层先法顾名思义是以层为考虑对象，首先第一步是顶层十字，指选定魔方上面的4个棱块，并将其还原到正确位置，形成一个十字形图案；第二步为复原顶层角块，在第一步中选定面的基础上，将其4个角块也扭转到正确位置，同时要保证每个角块的色片颜色朝向是正确的；第三步是指还原魔方中间层的四个棱块，并与前面一样保证色片的朝向正确；第四步的目标是在底层拼出一个与顶层颜色相对的十字图案，在这种状态下，底层4个棱块都处于正确位置，但是角块色片朝向却不一致；第五步将魔方翻转180°，观察4个角块的位置，并将它们扭转以与初始状态的色片朝向一致；第六步就能够将整个魔方还原到初始状态。

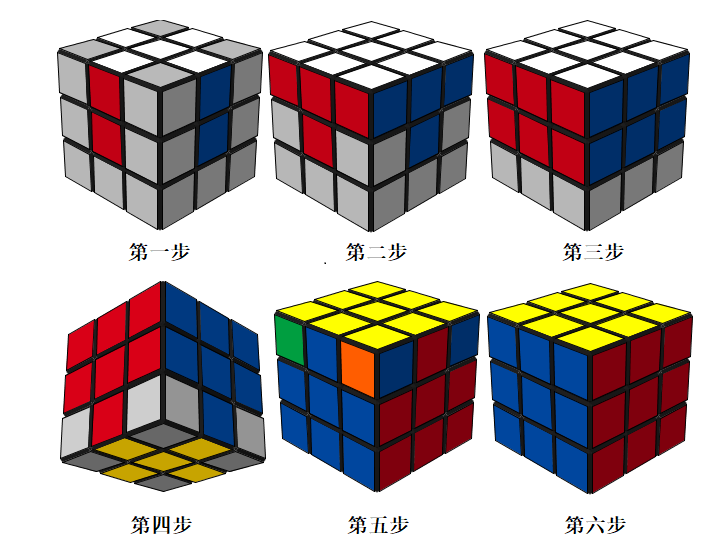


图4-1 层先法步骤

4.1.2 CFOP法

CFOP是层先法的进阶版算法，CFOP法将层先法的第二步和第四步融合到其余步骤中，最后整合成一、三、五、六这四个步骤[29]。目前大部分魔方高手都是采用这种方法还原魔方，在世界魔方大赛上的使用也是很常见的。

CFOP法是Cross、First 2 layers、Orientation of last layer、Permutation of last layer的简称，如图4-2所示，在这种方法中，要先在底面拼起一个同色的十字图案，并保证这层的4个棱块都处于正确的位置；接着在底面十字图案上还原底层与中间层的角块和棱块，这样一来，还原魔方的效率得到提高；顶层的还原要先确保棱块与色块的色片颜色朝向与初始状态一致，这样就能通过简单的层扭转来还原魔方。

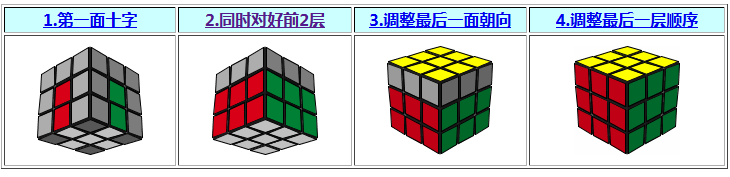


图4-2 CFOP法

4.1.3 Kociemba还原算法

Kociemba算法是由Kociemba基于TM法设计的，通常被称为Two Phase Algorithm，即二阶段算法[30]，目前可在GitHub上找到各种语言版本的算法。

在讨论Kociemba算法时，可以引入群的性质。例如，在使用除了<U1，U3，B1，B3，D1，D3，R1，R3>之外的方法来拧动处于初始状态下的魔方时，这样形成的魔方状态就属于魔方所有可能状态群中的一个子群，是一种包含关系。这个子群可以用一个公式A1=<F1，L1，R2，U2，B2，D2，R2>来表示，用特定子群表示的魔方在该种状态下所有棱块和角块的色片颜色方向不会产生变化。也就是说，魔方棱块或者角块的色片颜色与其所在位置有关，且当处于同一个位置的时候，这些颜色的朝向是一致的。此外，根据魔方的立体结构特点得出，任何打乱动作都不会影响到中间层棱块的位置，但是其色片颜色在不同子集中有所区别。将状态设定为群，能够将魔方的每一种打乱图案用数学语言表达出来。这样一来，在研究魔方还原过程时可以合理由魔方当前状态反推出其还原算法。

Kociemba算法的解魔方步骤分为两个阶段，首先在第一阶段中要使用到迭代加深的A\*算法，通过对目标的合理搜索与剪枝，保证魔方在转动后达到A1群内的状态。换种说法就是说，完成所有块的色片颜色方向调整，同时让原本属于中层的棱块在还原后也位于中间层位置。

在第二阶段中，Kociemba算法在魔方A1子群的基础上，调整魔方8个角块的相对位置，同时魔方的F面、B面，U面、D面，L面、R面两两相对的面上所有棱块处于正确位置，在此基础上，完成还原的所有步骤。当Kociemba算法求解出一个可行方案后，还会根据算法中提前设定的参数决定是否继续搜索，这是它的一种特点。若设定在最优步数模式下，算法会通过对第一阶段的不断优化来减少第二阶段的步数，直到第二阶段步数为零，此时的方案就是全局最优解。在实际设计过程中，为了符合设计要求减少算法运行的时间，在程序中可人为调整算法找出局部优解，以防止进入长时间的运算，保证装置整体的效率。

## 4.2 魔方还原算法的选择与优化

4.2.1 还原算法的选择

如今主流的魔方算法包括层先法、角先法、Kociemba算法等等。基于上文对一些经典算法的研究，步骤简洁的还原步骤与庞大的魔方状态群落分析是分不开的。本文在选择还原算法时主要考虑的是魔方还原步数以及魔方解算时间，针对层先法、CFOP法以及Kociemba算法的还原步数进行试验统计分析，得到的情况如表4-1所示。不难看出，在同等条件下，想要兼顾简洁与快速两个方面的优点，就必须选择Kociemba还原算法。

表4-1 多种算法实验分析

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 算法 | 实验次数 | 步数求和 | 平均步数 | 方差 |
| 层先法 | 20 | 2668 | 133.4 | 56.12 |
| CFOP法 | 20 | 1564 | 78.2 | 23.98 |
| Kociemba算法 | 20 | 404 | 20.2 | 0.86 |

4.2.2 还原算法的优化

然而，在实际运用中，Kociemba还原算法在运行时，求解出一个可行解之后，还会根据提前设定的参数决定是否继续搜索。虽然在经过更长时间的运算后，Kociemba算法确实能给出较之以前更优的解，但是平均达到15分钟的用时不在本设计的考虑范围内。针对这种现象，必须要在保证算法运行时效强的同时，也能够找到优解。于是，本设计做了一些对比实验，在程序中人为调整算法参数终止搜索，以防止进入长时间的运算成为了一种可行的方法[31]。本文统计了在1s和10s用时下，由Kociemba算法解算出的还公式所对应的舵机步数，其对比结果如图4-3所示。

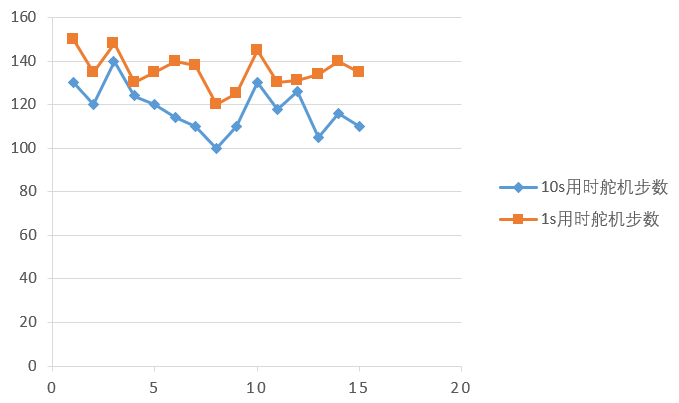


图4-3 两种状态下舵机步数对比

根据实验数据分析，在提前设定了搜索深度下，Kociemba算法短时给出的最优解与长时间运算的最优解并没有显著的差别。过分追求最优解带来的舵机动作简化大于寻找最优解花费的时间消耗。因此，设定特定的参数限制搜索时间与搜索深度来优化Kociemba算法，对本文来说是很合理的选择。

## 4.3 本章总结

本章节从魔方入门解法层先法、专业速拧CFOP法入手，介绍了他们的具体步骤与优缺点。接着介绍目前主流的Kociemba还原算法，这种算法能够兼具还原步数少和还原时间短的优点。Kociemba算法在求解出一个可行解之后，还会根据提前设定的参数决定是否继续搜索，若将算法设定在最优步数模式下，算法会通过对第一阶段的不断优化来减少第二阶段的步数，直到第二阶段步数为零，此时的方案就是全局最优解。这种无意义的长时间搜索并不会影响解算结果，反而不利于系统的用时，因此本文对其进行优化，在程序中人为调整算法运行时间和搜索深度终止搜索，在保证算法运行时效强的同时，也能够找到优解。

# 第五章 转魔方机械装置的算法实现

## 5.1魔方特征信息提取

5.1.1 魔方图像采集

在APP调用摄像头拍摄照片的流程中，首先按下Key1键，STM32单片机发送指令，控制舵机将魔方夹紧并处于等待拍照的状态。等待2秒后，蓝牙给STM32单片机发送指令，控制舵机旋转，使得手机摄像头依次完成魔方6个面的图像采集。图像采集完成后，拍摄好的照片将会按照拍摄的顺序编号命名，自动保存进设置好的路径中，等待调用。

5.1.2 图像预处理

在图像在形成、传输的过程中，图像质量容易受到外界噪声影响而产生退化，因此，为了减小噪声影响、维持照片的质量细节，可采取各种滤波方法对图像进行去噪处理，提高图像效果，使之更适合于后续图像分析[32]。

在手机摄像头获取魔方6个面的图像过程中，由于受到外界环境光照强度影响、魔方色差因素影响等，获取到的魔方图像极容易产生模糊、形变、色差不准等现象。这种图像的质量退化对于后续图像分析是非常不利的。因此，通过对比学习，本文选取了OpenCV中常用的高斯滤波[33]图像处理方法，通过滤波，能够使有效去除图像中的噪声，平滑的图像更易于后续分析处理。其效果如图5-1所示，从左到右依次是原图、高斯滤波后图、黄色区域HSV图以及识别后得到的黄色块图。

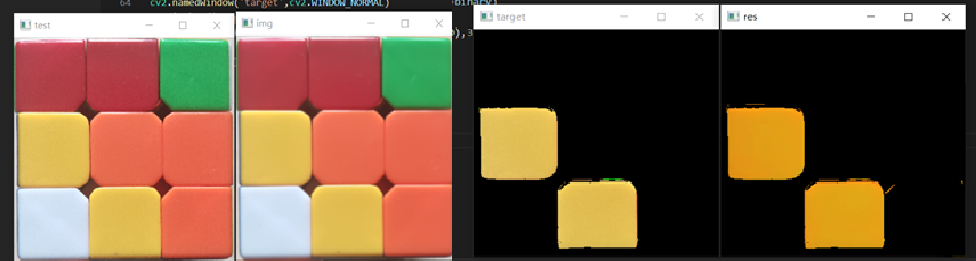


图5-1 高斯滤波效果

## 5.2 魔方视觉识别

5.2.1 阈值法

以往的解魔方机器人大多数选择阈值法作为视觉识别的方法，该方法是在提取魔方特定位置或平均位置的RGB或HSV值后，将其与设定好的颜色阈值作比较，若提取到的颜色信息在这一范围内，则认为该颜色与设定的颜色一致，不然，则不是这种设定颜色[34]。

阈值法计算简单、运算效率较高、速度快，但是对于本文的视觉识别设计来说其缺点也是不容忽视的。首先，阈值法的识别准确率与图像的质量息息相关，如果魔方图像质量受到外界环境光强或魔方色块色差等因素影响，图像的质量退化会对阈值法产生极大干扰，阈值法对比出来的颜色信息准确度也会大大下降。其次，手机图像存储与调用的过程中，也会影响到图像的细节，并会造成魔方实物图像与所设定的颜色阈值不相符的情况。总体而言，装置视觉识别的可靠性收到阈值法的极大限制。

5.2.2 k-means聚类算法

考虑到阈值法的局限性，本文在研究此问题的过程中引入了一种聚类算法。将物理或者抽象对象的集合分成由类似的对象组成的多个类的过程被称为聚类。一般意义上来讲，一个数据集内的每个样本之间往往有各种各样的特性，将这些特性数值化、规范化，并且在高维空间中描绘出来，对于相聚比较近的点分为一类，将整个数据集分成若干类的过程便是聚类[35]。

图5-2所示的就是一个典型的聚类例子，此聚类为二维平面上点的聚类。在平面中有一定数量的点，每个点都有属于自己的（x，y）坐标，将这些点绘制在二维平面上之后，可明显看到这些点分成三簇，而聚类算法的实质就是给出每个点属于哪一簇的索引。

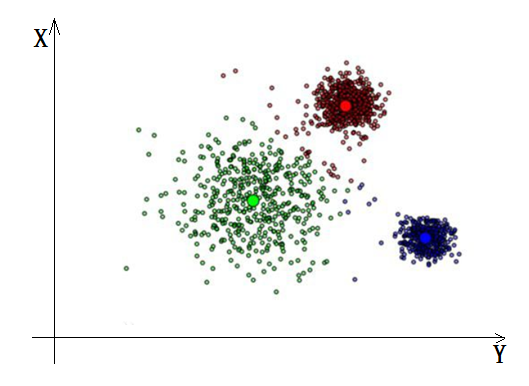


图5-2 二维平面上点的聚类

如果数据点较少，数据特点维度较低，人为很容易将一批数据集分成若干类。但在面对大批量数据时，尤其是对于高维度数据，人为很难靠想象力分析数据之间的关系，因此可以通过计算机的计算量来帮助寻找海量数据之间的关系。在实际商业经营中，聚类通常可以将特性群体分成若干类，不同的类别有不同的职业划分、消费观念、兴趣爱好等等，生产厂家或者销售人员可以基于此对生产结构和销售策略做出调整，以更符合客户群体的特性。

聚类在机器学习中属于无监督学习，也就是说其仅靠本身算法就能够将数据集分成不同种类，而不需要提前训练。本设计中采用的聚类算法为k-means聚类算法。经过学习研究，传统的k-means聚类算法通过3个步骤对数据集进行聚类分类：随机选取聚类中心、簇分类、移动聚类中心。如流程图5-3所示，k-means聚类算法通过对数据的聚类判断，将其分为不同的类别。

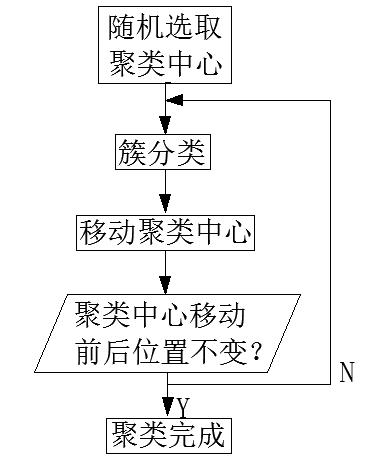


图5-3 k-means算法流程图

结合转本文的魔方机械装置的特点，在实际设计过程中需要对k-means聚类算法进行优化。

第一点，由于初始聚类中心的选择不同，算法的聚类速度也会随之受到影响，与此同时，当一些数据集内部样本信息较为相似时，其最终聚类结果也因初始数据点的选择不同而背离预期效果[36]。本文设计的转魔方机械装置在进行视觉识别时，其各个面在被拍摄时中心块的位置不变，所以本设计将聚类中心设置为每个面的中心块色片颜色，这样就可以避免因随机点不具有代表性而影响聚类结果的现象。

另外一点，在以往的例子中也会出现使用k-means算法无法保证聚类结果每一簇数据点的个数的状况。而对于本设计的颜色聚类算法，魔方立体结构特点决定每种颜色的色块数量一定是9，所以在实际算法设计中，可以将每次聚类过程各簇分到的数据数量设定为9个。

## 5.3 优化还原指令序列

5.3.1 优化魔方还原指令序列

由第二章的介绍可知，本文涉及了两套魔方还原指令，在原始的还原指令序列中，去掉了U、U’、U2、D、D’、D2的指令，结合舵机的动作设计了魔方整体翻转的指令。之所以做出这样的优化，是因为转魔方机械装置在机械结构上有所不足。不同于人类在还原魔方时天然使用到了手的六个自由度，为了维持整个机械装置的稳定性，在保证性能的前提下节省材料成本，本文设计的转魔方机械装置受其结构影响只拥有四个自由度，如图5-4所示，整个机械结构在上、下、左、右四个方向自由运动，加入了诸如翻转、复原等指令后，新的优化还原指令序列能够大大减少舵机运动的步数。

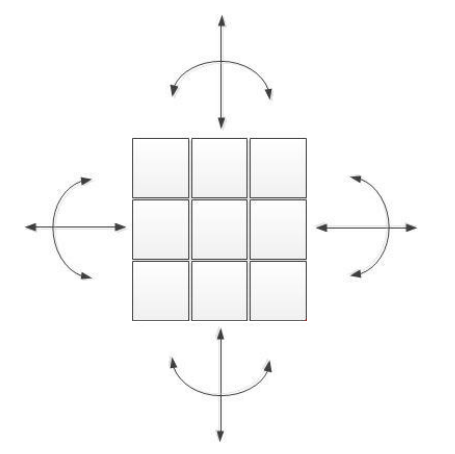


图5-4 转魔方机械装置的自由度

在优化还原指令序列中每个单指令所对应的舵机执行步数是一定的，因此，当舵机运行的速度一定时，魔方还原时间和舵机执行步数成正比例关系。优化后的指令序列中每一个单指令所对应的舵机执行步数如表5-1所示，在这种执行条件下，舵机步数的减少对提高整个装置的还原效率具有重要作用。

表5-1 优化指令序列对应的舵机执行步数

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 指令 | L | R | F | B | L’ | R’ | F’ | B’ | L2 | R2 | F2 | B2 | LL | RR | FF | BB |
| 步数 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 8 | 8 | 8 | 8 | 6 | 6 | 6 | 6 |

5.3.2 二叉树模型优化算法

然而在实际设计中，STM32单片机接收到来自安卓手机的求解字符串后，在操作过程中，舵机执行整体翻转指令之后是否需要再松开魔方翻转回原位，这是一个值得讨论的问题。当舵机想要回到原来位置时需要执行相对应的逆翻转运动，这样舵机就要执行额外的6步动作；若是不翻转回原来的位置，那么翻转之后的所有魔方还原指令都受翻转影响发生变化。这两种选择的差距关乎到系统运行的流畅性和效率，为了解决这一疑惑，本文引入了二叉树模型，将找到舵机执行动作的最优解转换为求二叉树的路径问题[37]。如图5-5所示，二叉树模型的流程设计符合本文探索最优解的目的，能够很好的优化舵机执行动作，从而提高整体装置的魔方还原效率。

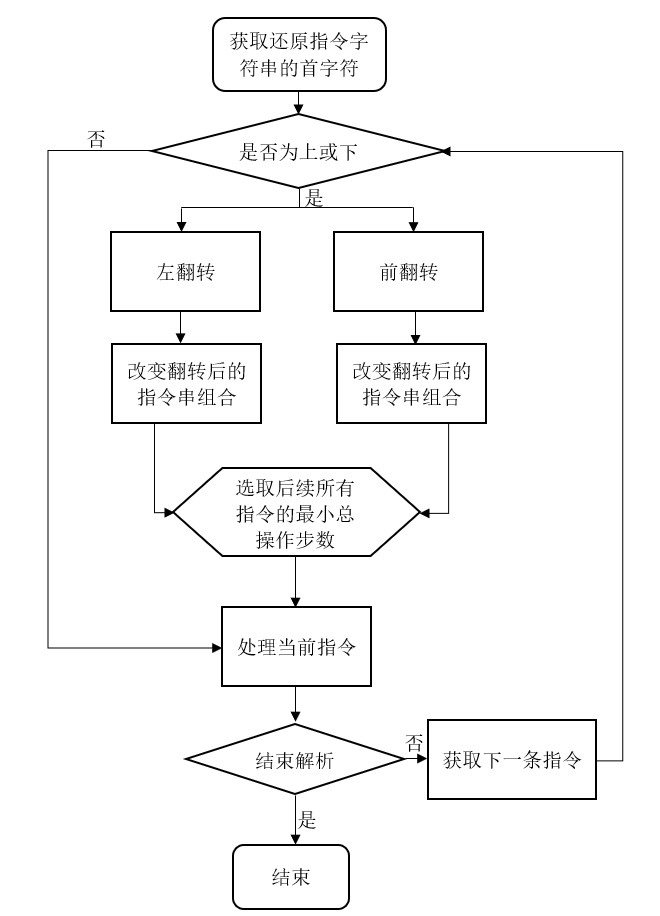


图5-5 二叉树模型

## 5.4 系统测试

本文设计的四臂两指型转魔方机械装置能够将打乱的魔方还原到初始状态，在这一过程中，时间参数与还原准确度是重要的测试标准。在系统测试阶段，在本文搭建的机械装置平台上连续进行了15次魔方复原测试。

魔方完全严格地由人工打乱，为了维持整个系统测试的合理性，在每次的打乱状态下，不允许出现相邻棱块和角块颜色大幅度重合的情况。本文设计了两种时间测试参数分别是：视觉识别时间是指从STM32单片机与安卓手机进行蓝牙通讯后得到魔方各个面颜色信息并解算出优化还原序列；伺服舵机还原时间是指STM32单片机获取解析优化指令序列后控制舵机运动直到还原整个魔方的时间。这样两者相加即可得到总体还原时间。整个系统测试实验的结论如图5-6所示。

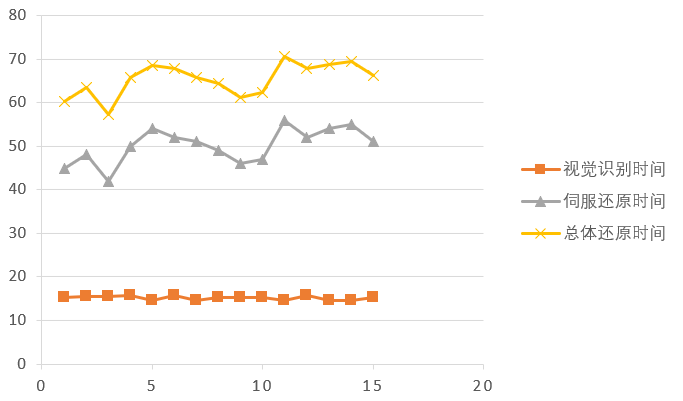


图5-6 系统还原时间测试结果

经过分析，本设计视觉识别时间较为稳定，其平均用时为15秒左右，舵机执行时间最大为56秒，最小为42秒，平均用时为50秒。由此可见，系统整体还原平均时间为65秒左右，并且测试结果显示本文设计的转魔方机械装置能够在70秒以内的时间还原绝大多数打乱的魔方。

## 5.5 本章总结

本章从视觉识别的算法入手，首先对影响图像质量的因素做出分析并给出了高斯滤波的图像处理方法。接着，通过研究阈值法在传统解魔方机器人当中的运用，分析其优缺点，认为不适合本文的设计思路将其摒弃。选取颜色聚类算法k-means聚类算法实现对魔方色块的聚类分类，极大地避免了外界环境因素对整个视觉识别过程的影响。在还原指令优化的环节中，先根据四臂两指型装置的四自由度设计出新的一套魔方优化还原指令序列，在此基础上，引入二叉树模型对舵机的执行动作进行分解判断，将其转换为模型的最短路径问题，对整个系统的运行效率有着重要作用。最后通过系统测试分析出，本文设计的转魔方机械装置能够在70秒内有效还原被打乱的魔方。

# 第六章 结论与展望

## 6.1 结论

本文介绍了一种四臂两指型转魔方机械装置的设计过程，整体结构采用STM32单片机为主控，安卓手机为辅助控制的搭建方案。双控制器通过蓝牙通讯模块实时交互，能够有效便捷地传递信息，最终实现三阶魔方的还原过程。该机械装置结构简单，系统稳定，成本低廉，是学习自动化与智能化尤其是机电一体化技术的良好选择，同时，也可以将其包装成科普产品走入大众视野，这对于科教兴国战略、前沿理论知识的普及、及全民科学知识文化的提高具有深远的意义。

在视觉伺服还原系统的设计过程中，用人为设置搜索深度的方法避免了Kociemba还原算法陷入无限搜索以至于耗时较长的状况；在视觉识别算法中摒弃了传统的阈值法，并引入k-means聚类算法，将魔方六个中心色块作为聚类中心，同时设置各簇分到的数据数量为9，这样大大削减了外界环境光强因素对颜色信息提取的影响，使其成功率接近100%；在伺服舵机执行动作流程中，先根据四臂两指型装置的四自由度设计出一套新的魔方优化还原指令序列，在此基础上，引入二叉树模型对舵机的执行动作进行分解判断，将其转换为模型的最短路径问题，对整个系统的运行效率有着重要作用。

最后通过系统测试分析出，本文设计的转魔方机械装置能够在70秒内有效还原被打乱的魔方。综上所述，本文设计的机械装置达到了设计方案的预期效果，魔方还原平台总体上呈现出稳定性与高效性。

## 6.2 展望

本文设计的转魔方机械装置在还原时间上还有进一步提升空间，这主要与伺服舵机组搭建的驱动模块有关。伺服舵机在运动时稳定性不高，速度较慢，因此，今后的研究方向是将驱动模块升级为速度更快、稳定性更好的驱动部件，以此提高整个系统的还原效率。

此外，曲柄连杆结构的可靠性不高，在操作臂前进后退过程中容易导致连接有夹具的舵机速度波动起伏较大。因此，今后会在整个系统的基础上优化伺服舵机速度控制算法，使机械装置在稳定性与速度上达到平衡。

# 参考文献

[1]黄铭,陆思良,孔凡让.魔方还原机器人的视觉子系统设计与实现[J].机械与电子,2013,(05):60-64.

[2]李海龙.基于LEGO平台的智能解魔方机器人系统研究与实现[D].东北大学,2015.

[3]Lu, S. L., Huang, M., & Kong, F. R. (2013). The Design of a Rubik's Cube Robot. Advanced Materials Research, 709, 432-435.

[4]田田,徐林,赵洪华,昃向博.解魔方四爪机械手结构设计与操作[J].机器人技术与应用,2014,(05):38-39.

[5]田田.解魔方气动组合机械手功能结构设计与系统分析[D].济南大学,2015.

[6]刘远法,周屹.基于Arduino单片机的解魔方机器人——控制部分[J].电脑知识与技术,2016,12(07):171-173.DOI:10.14004/j.cnki.ckt.2016.0831

[7]张宏平.魔方机器人手臂设计及运动学分析与动态特性研究[D].合肥工业大学,2016.

[8]卢桂萍,程开,罗泽奇,吴荣鑫,陈永键,康诗铄.基于Kociemba算法的双臂解魔方机器人还原算法研究[J].机电工程技术,2021,50(09):100-103.

[9]纪海彬.基于AVR的解三阶魔方机器人研究[D].长春理工大学,2013.

[10]Jozef Varga, Frantisek Durovsky, Juraj Kovac，Design of Pneumatical Rubik’S Cube Solver，Applied Mechanics and Materials,2014,613:265-272.

[11]Vasile Dan, Gabriel Harja and Ioan Nascu.Advanced Rubik's Cube Algorithmic Solver:IEEE,2021,90-94.

[12]哈金才,李若雪,哈瑞.魔方的数学模型研究及其应用[J].创新创业理论研究与实践,2018,1(19):83-86.

[13]梁小龙.解魔方算法的研究和系统实现[D].东北大学,2013.

[14]郑瑜.魔方原理及其应用[D].浙江大学,2009.

[15]喻腾飞.魔方群的研究[D].中国科学技术大学,2014.

[16]卢文娟,吴华芳,刘甜璐,曾达幸.三阶魔方机构的拓扑与运动简图[J].机械工程学报,2020,56(23):89-97.

[17]勾慧兰,刘光超.基于STM32的最小系统及串口通信的实现[J].工业控制计算机,2012,25(09):26-28.

[18]李培林.安卓系统的应用及发展趋势展望[J].计算机光盘软件与应用,2012,15(18):161-162.

[19]高达.基于STM32双臂魔方机器人的设计[J].电子产品世界,2018,25(11):51-53.

[20]Jing Chen, Litian Liu, Zhijian Li, et al. On the single-chip condenser miniature microphone usingDRIE and backside etching techniques[J]. Sensors amp; Actuators: A. Physical, 2003, 103(1):23-26.

[21]程院莲.浅析步进电机、伺服电机和舵机的区别[J].轻工科技,2020,36(01):36-37+62.

[22]陈思颖,邹乐瑶,王少坤.基于ATMEGA32A单片机和HC05的蓝牙信号传输体操机器人设计[J].信息与电脑(理论版),2021,33(07):106-108.

[23]刘拓晟.基于STM32的倒立摆控制系统设计[J].电脑知识与技术,2018,14(36):213-215.DOI:10.14004/j.cnki.ckt.2018.4289

[24]唐金元,王翠珍.0～24V可调直流稳压电源电路的设计方法[J].现代电子技术,2008,(04):12-14.DOI:10.16652/j.issn.1004-373x.2008.04.007

[25]杨玲,钱百青,徐晨光,徐子文.基于STM32输出多路PWN波控制八足蜘蛛机器人的方法研究[J].电脑知识与技术,2017,13(23):156-158.DOI:10.14004/j.cnki.ckt.2017.2432

[26]刘兵.基于ADXL362的低功耗计步器的设计与实现[D].大连理工大学,2017.

[27]郑昊.基于Arduino/Android的蓝牙通信系统设计与实现[D].湖北大学,2012.

[28]黄晋,王健行,吴恩华.一种魔方求解游戏的实现与交互[C] 全国教育游戏与虚拟现实学术会议. 2010.

[29]张雪娇.智能魔方机器人的视觉感知与复原算法研究[D].东北大学,2011.

[30]谭水生.解三阶魔方智能机器人系统的设计与研究[D].五邑大学,2020.

[31]苏磊.双臂解魔方机器人的算法实现[D].河北工程大学,2020.

[32]张旭明,徐滨士,董世运.用于图像处理的自适应中值滤波[J].计算机辅助设计与图形学学报,2005,(02):295-299.

[33]王丹.基于各向异性高斯滤波的图像边缘检测方法[D].西安电子科技大学,2010.

[34]谢鹏鹤.图像阈值分割算法研究[D].湘潭大学,2012.

[35]吴超辉.K-means改进算法在多元分系统中的研究和应用[D].武汉理工大学,2016.

[36]孙林,刘梦含,徐久成.基于优化初始聚类中心和轮廓系数的K-means聚类算法[J].模糊系统与数学,2022,36(01):47-65.

[37]王雪,钱志鸿,胡正超,李奕男.基于二叉树的RFID防碰撞算法的研究[J].通信学报,2010,31(06):49-57.

# 

# 致谢

大学四年的时光匆匆而过，转眼间本科生活就已经来到了尾声。在中国农业大学的四年学习生活中，我接触到了一片新的天地。还记得从紧张枯燥的高中学习中刚踏进大学校园时，我怀着激动向往的心情让自己融入周围的一切，这期间，我遇到了很多良师益友，在我前进的道路上给予我许多帮助。他们的鼓励与支持，我将用一生铭记。

感谢冉寒学长在我刚入大学时传授给我的学习生活经验，我能够很快适应新的学习与工作同这离不开关系。很怀念在新闻通讯社一起工作的日子，那两年的工作经验让我从一个新媒体小白成长为进阶版新媒体小白，也认识到很多志同道合的新朋友。此外，也很感谢学长在我面临学业、工作选择的时候为我提供的诚恳的、实际的分析，我也从中收益良多。

感谢黄煜杰同学在我学业和生活上对我的支持。每当我遇到困难阻碍的时候，他都会无条件地为我提供帮助，鼓励我克服困难继续前进。在我有所收获的时候，与我一起分享喜悦，感谢他对我的帮助与照顾。

感谢阎欣峰班长对我大学四年的学习指导。在班级缺少团支书时，是他鼓励踌躇犹豫的我竞选团支书，并在我正式成为团支书后与我一起进行班级工作，为班级建设提出了宝贵的意见。

感谢我的舍友崔芯文、柳怡宁同学在宿舍生活中包容我的一些习惯，感谢她们对我的体贴照顾，让我在大学四年中一直拥有良好的宿舍氛围。

感谢导师胡标老师对我毕业设计的莫大支持，让我能够全身心地投入到研究设计中，同时保证每一个设计环节都有序进行。

最后，感谢始终在我背后默默支持我的家人们，他们的爱给了我面对未知的勇气与底气，他们的尊重与支持让我能做出自己想要的选择，从不吝啬在我身上投入一切，让我能够站在他们的肩膀上拥抱整个世界。

大学四年的经历让我的生活丰富多彩，在未来的前行道路上，我会将这四年的宝贵回忆铭记于心，砥砺前行。

# 

# 作者简介

**基本介绍（姓名、性别、出生日期、籍贯等）:**

吴小末，女，1999年12月31日生，江苏省扬州市高邮市人。

**教育经历：**

2018年09月-2022年06月 中国农业大学 机械电子工程 工学学士

**本科期间主持/参与的科研项目：**

校级本科生科研项目，基于51单片机的四路抢答器设计，2019.5-2020.04，参与。

**本科期间获得的奖励和荣誉：**

（1）全国大学生英语竞赛（NECCS）省部级优秀奖、学业奖学金（2021）

（2）优秀团员称号、学习优秀奖学金、学习进步奖学金（2020）