

LEGO 机器人魔方还原基本方法与过程

李国军¹, 钟志强², 张毅宁¹

(1. 鞍山师范学院 物理科学与技术学院, 辽宁 鞍山 114007; 2. 鞍山师范学院 教育科学与技术学院, 辽宁 鞍山 114007)

摘要:LEGO 机器人魔方还原问题是一个持续受关注的问题,但其工作基本原理和解决方法仍不能被广泛理解.本文以层先法为例说明了魔方还原基本方法,结合 leJOS 编程软件解析了 LEGO 机器人还原魔方中的扫描魔方、识别色彩、魔方求解运算、机械实施还原的主要过程.

关键词:LEGO 机器人;魔方;模式识别

中图分类号:G434 **文献标识码:**A **文章篇号:**1008-2441(2014)02-0071-04

1974 年,匈牙利布达佩斯建筑学院厄尔诺·鲁比克教授发明魔方(Rubik's Cube),之初仅作为增强学生空间思维能力的教学工具^[1].其中三阶魔方由一个连接着 6 个中心块的中心轴以及 8 个角块、12 个棱块构成.变化数是 $(8! \times 38 \times 12! \times 212)/(2 \times 2 \times 3)$ 约等于 4.33×10^{19} .如今魔方风靡世界,奥妙无穷.

LEGO 机器人主要用于 10 岁以上青少年机器人教育和学习,由丹麦乐高(LEGO)公司、美国麻省理工大学(MIT)和美国国家仪器(NI)公司共同开发研究,其硬件由控制器、伺服电机、传感器等组成^[2].2013 年 8 月上市的 EV3(Evolution)主处理器为 32 位 ARM9,主频 300 MHz.乐高官方推荐的软件系统是 MINDSTORMS Education EV3,其使用 LabView 引擎开发的图形化编程软件,其功能丰富且入手简便^[3].第三方代码编写软件有 leJOS NXJ, BricxCC, ROBOTC, NxtJLib 等.其中本文使用的 leJOS 是 SourceForge.net 上的开源软件,是可执行于 LEGO 机器人主机上的 JVM(Java Virtual Machine),在 Eclipse 中只需引用 classes.jar 即可.2013 年 9 月 22 日 leJOS-EV3 0.4.0-alpha 发布.

LEGO 机器人解魔方已不为鲜见,原创有瑞典 Hans Andersson 的 Tilted Twister1.2(2008 年 9 月 21 日)、Tilted Twister2.2(2010 年 8 月 15 日,蓝牙数据传输数据到 PC 端完成魔方求解计算)和 David Gilday 的 MindCuber 2.2(2013 年 7 月 13 日),等等.但由于程序代码较大,其工作基本原理和解决方法仍不能在短时内让一般研究者们理解,本文以此为题,结合 leJOS 代码概述 LEGO 机器人魔方还原基本方法和主要过程.

1 魔方还原基本方法

三阶魔方的还原方法包括层先法(Layer-First Solution Method)、角先法(Corners-First Solution Method)、棱先法(Edges-First Solution Method)、二阶段法(The Two-Phase-Algorithm)、桥式法(Bridge Solution Method)、CFOP(Cross、F2L(First 2 Layer)、OLL(Orientation of Last Layer)、PLL(Permutation of Last Layer))和笑面虎法等.其中层先法适合魔方初学者使用;角先法公式少,最适合初级乐高机器人编程实现,如 Tilted Twister1.2^[4];二阶段法以算法的速度为目标,而不以最优解为目标,方便在机器人编程实现,如 Tilted Twister2.2^[5];CFOP 由捷克密码学教授 Jessica Fridrich 发明用于竞速,是目前世界上

最流行的方块解法^[6]. 现以层先法为例说明魔方还原过程(如图 1).

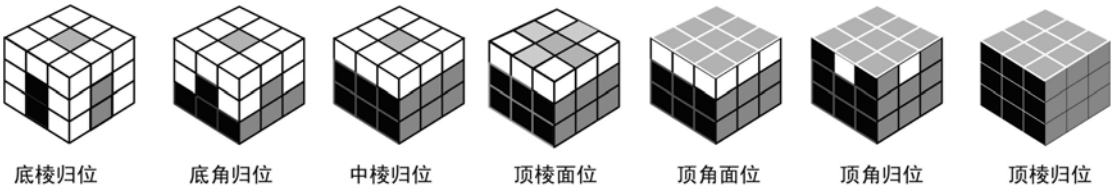


图 1 层先法魔方还原过程

- 第一步:底棱归位又称底部架十字,实现较容易.
- 第二步:底角归位,复原魔方第一层 4 个角块有 5 种情况分别考虑. 作者以其中一情况作为说明如图 2,其他情况可参考相关材料. 其公式表征为 $R_1 U_1 R_3$. 第一个字母表示操作的面(F:front,B:back,R:right,U:up),第二个字母表示旋转的方向,1 是顺时针,3 是逆时针,2 是旋转 180°.
- 第三步:中棱归位,复原魔方中层 4 个棱块,有类似 3 种情况(略).
- 第四步:顶棱面位,也称顶层架十字,有类似 2 种情况(略).
- 第五步:顶角面位,魔方的 4 个顶角的顶面色全部调至顶面有类似 2 种情况(略).
- 第六步:顶角归位.
- 第七步:顶棱归位^[7]. 也就是说当机器人解魔方时,可能对应算法不一致,但是算法的每个步骤都有相应确定公式.

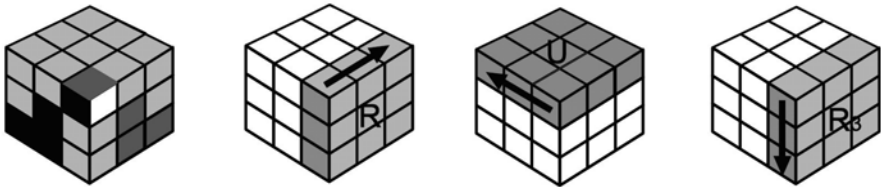


图 2 底棱归位情况之一的魔方操作图

2 机器人魔方还原过程

2.1 魔方的数组表示

魔方有 6 个面,每面含 9 个色块,按图 3 顺序确定编号. 其数组 `String SideColors [] = { "yyyyyyyy", "rrrrrrrr", "ooooooo", "bbbbbbbb", "wwwwwww", "gggggggg" }`. 在翻转魔方的时候,为使坐标系永远跟魔方同步,使用 `CenterColor` 与 `Sides` 类,用来记录 6 个面的中心在旋转后的位置和面上颜色信息^[8]:

```
string [ ] CenterColor = new string [ 6 ] { "U", "R", "D", "L", "F", "B" };
if ( clockwise ) {
    string n = CenterColor [ 5 ];
    CenterColor [ 5 ] = CenterColor [ 1 ];
    CenterColor [ 1 ] = CenterColor [ 4 ];
    CenterColor [ 4 ] = CenterColor [ 3 ];
    CenterColor [ 3 ] = n;
}
if ( ClockWise ) {
    for ( i = 0; i < 3; i++ ) {
        temp = Sides [ 5 ] [ 0 ] [ i ];
        Sides [ 5 ] [ 0 ] [ i ] = Sides [ 3 ] [ 2 - i ] [ 0 ];
        Sides [ 3 ] [ 2 - i ] [ 0 ] = Sides [ 4 ] [ 2 ] [ 2 - i ];
        Sides [ 4 ] [ 2 ] [ 2 - i ] = Sides [ 1 ] [ i ] [ 2 ];
        Sides [ 1 ] [ i ] [ 2 ] = temp;
    }
}
```

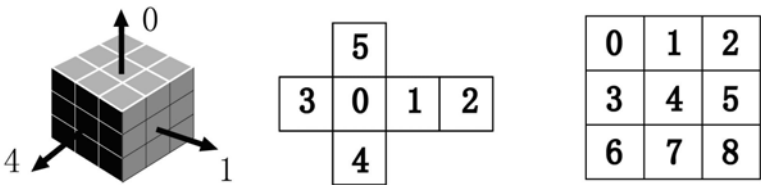


图 3 魔方的数组顺序及编号图

2.2 扫描魔方、识别色彩

机器人的视觉识别多使用摄像头,经过图像分割、聚类算法,在 HSV (Hue, Saturation, Value) 颜色空间下实验分析.而 LEGO 机器人则通过颜色传感器 (Color Sensor, 采样率为 1KHz, 可检测 8 种颜色) 经过伺服电机转动实现识别色彩.事实上这个数值受环境光线强度影响非常大,即使相同的环境下,读数也会有波动,何况必须扫 54 个点.为此定义分辨颜色的判断为如下规则^[8]:

- (1) 假设 RGB3 个值的和为数组 TargetExists [], 按降序排序, 前 9 个是白色.
- (2) 再按 G 分量降序排序, 前 9 个是黄色.
- (3) 再按 B 分量降序排序, 前 9 个是蓝色.
- (4) 再按 R 分量降序排序, 前 9 个是绿色.
- (5) 再按 $R+2 * G-2 * B$ 降序排序, 前 9 个是橙色.
- (6) 剩下的颜色全是红色.

```
ColorSensor color = new ColorSensor (SensorPort. S3);
```

```
int R= color. getRawRed (); int G= color. getRawGreen (); int B= color. getRawBlue ();
```

```
int[] TargetExists = R + G + B; 并以此作为依据生成魔方数组.
```

```
for (int i=0; i < 6; i++) { for (int j=0; j < 3; j++) { for (int k=0; k < 3; k++) {
```

```
int c=i * 100 + j * 10 + k; string r=ColorSortResult [c]; ColorStr += ColorValue (r);
```

```
RealStr += r; } } }
```

2.3 魔方求解算法

魔方求解算法国内外已有多种方式, 可参看文献[4]、[5]、[9]. 作者参考了 Tilted Twister 二阶段法魔方求解算法, 使用了 org. kociemba. twophase. Search 类; 将 SideColors 信息进行转化, 最后将颜色数组转换为解魔方的步骤. 结果保存在 solutions 变量中, 格式为: F1 U2 F2 D3 L2 D1 F1 U3 L2 D1, 其中每两个字符表示一个旋转步骤,

```
String msg = new String (SideColors). substring (0, 54);
```

```
String cube = msg. substring (4 * 9, 4 * 9 + 9) + msg. substring (2 * 9, 2 * 9 + 9) + msg. substring (1 * 9, 1 * 9 + 9)
```

```
+ msg. substring (5 * 9, 5 * 9 + 9) + msg. substring (0 * 9, 0 * 9 + 9) + msg. substring (3 * 9, 3 * 9 + 9);
```

```
String solutions = Search. solution (cube, 30, 30, false);
```

2.4 机械实施还原

LEGO 机器人伺服电机 (Server Motor) 测速反馈精度 160 ~ 170 RPM (Revolutions Per Minute), 内置角度传感器 (Rotator Sensor) 可校准控制精度到 1 度. Motor paw = Motor. A; // 电机 A 口接机器手的电机 Motor bottom = Motor. C; // 电机 C 口接魔方底座的电机. 其机械搭建作者参考了 MindCuber 2.2, 如图 4 所示.

建立 Robot 类, 通过调整 nQuarter 参数值实现翻转面和顺 (逆) 时针旋转 90° 或 180° 实现 solutions 变量中相应旋转步骤.

```
public class Robot { public void RotateBottomSide (int nQuarter) { ... }
```

```
public void RotateBottom (int nQuarter) { ... } public void RotatePaw () { ... } }
```

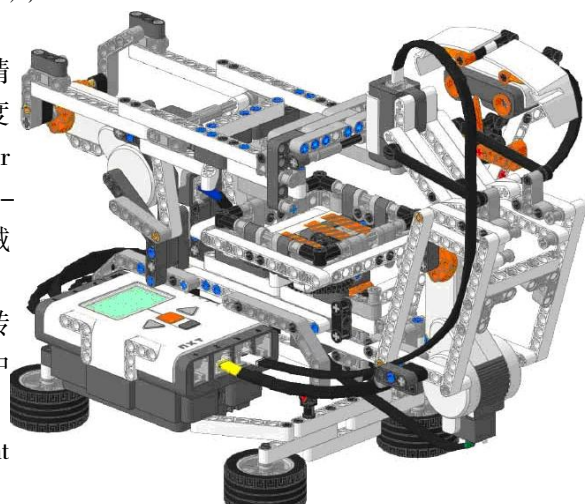


图 4 MindCuber 2.2 整体图

3 结语

LEGO 机器人魔方还原过程还有诸多问题没有探讨, 如机器手伺服定位、转动扭矩与校验调整、PC

端通信、摄像头视频辅助分析色块问题、超声波传感器做开关工作问题.

参考文献:

- [1] 魔方_百度百科[EB/OL]. <http://baike.baidu.com>,2013-08-10.
- [2] 钟志强. Android 系统协同 LEGO 机器人工作关键技术研究[J]. 电脑编程技巧与维护,2013(20):78-80.
- [3] 李国军,钟志强. 乐高 NXT 光线传感器自平衡机器人教学开发[J]. 鞍山师范学院学报,2013,15(2):34-37.
- [4] Solving Corners[EB/OL]. <http://rubikscube.info>,2013-07-10.
- [5] Cube Explorer [EB/OL]. <http://kociemba.org/cube.htm>,2013-09-10.
- [6] 郑瑜. 魔方原理及其应用[D]. 杭州:浙江大学,2009.
- [7] 中国魔方俱乐部[EB/OL]. <http://www.mf8-china.com>,2013-12-10.
- [8] 魔方机器人[EB/OL]. <http://www.diy-robots.com>,2013-10-10.
- [9] rokicki[EB/OL]. <http://tomas.rokicki.com/cubecontest>,2013-10-10.

The Basic Method and Process of Lego Robot Working Rubik's Cube Work

LI Guo-jun¹, ZHONG Zhi-qiang², ZHANG Yi-ning¹

(1. School of Physical Science and Technology, Anshan Normal University, Anshan Liaoning 114007, China;

2. School of Educational Science and Technology, Anshan Normal University, Anshan Liaoning 114007, China)

Abstract: Rubik's Cube solution for LEGO robot is receiving constant concern, but The basic method and process of Lego robot working is perplex for LEGO fans. This paper illustrates Rubik's Cube Layer-First Solution Method, and analyses scanning color, pattern recognition, Cube solution and mechanize recovering process.

Key words: LEGO robot; Rubik's Cube; Pattern recognition

(责任编辑:张冬冬)