

《数字图像处理基础》课程实践

结题报告

项 目 名 称 ： 数字图像处理在俄罗斯方块游戏运用

姓 名 ： 阮正鑫

学 号 ： 516021910587

联 系 电 话 ： 18205953938

电 子 信 箱 ： 1051826257@qq.com

合作人姓名 ： 徐双捷、李宇凡

2019 年 1 月

请参照下列提纲填报

（一）研究内容综述

1. 对比开题报告，说明各个研究内容的完成情况
2. 未完成的研究指标应说明原因

我们小组研究的是基于图像处理的自动俄罗斯方块游戏攻略，开题报告中设计的研究技术点有：

（1）、通过图像处理技术，获得俄罗斯方块游戏中，下落物块的属性以及底部物块的状态，进而进行处理。

我们小组采用的方法是：截取当物块落到第四行时的图片，将其转化为二值图之后，遍历图像上各个方格的像素值。经过我们实验得出，当像素在某一个范围之内时，此方格中有物块，而当像素值在范围之外是，方格中没有物块。我们构建了一个 22×10 的矩阵，在矩阵中将有物块的格子置为 1 而没有物块的格子置为 0。这样的一个 22×10 大小的元素都为 0/1 的矩阵保存了游戏棋盘中的所有物块的信息，用于游戏策略部分的设计。

（2）、游戏策略算法部分，利用处理图像获得的信息，判断当前下落的物块应该落下的位置，得到对应的操作数

游戏策略部分我们小组采用了 Pierre Dellacherie 算法。通过读取游戏棋盘矩阵，从中获得六个参数并加权，得出物块下落在相应位置的得分。穷举物块下落在各个可能位置的得分，获得一个最好的下落位置。输出了移动操作数与旋转操作数。在这个算法上，我们小组对算法做出了一定的调整，对于六个参数的权重，以及计算方法进行了正定。使得得到的操作数更为合理，准确。

（3）、利用游戏策略中得到的移动操作数与旋转操作数，控制机械臂自动进行游戏。

策略算法输出对落块的操作指令，包含旋转次数和移动格数（用符号表征方向），设备据此进行物理操作。经调整，设备能够较为精准地进行位移和点击，但存在无效点击和连点的状况。

（二）研究方案

请详细撰写各个研究内容的研究过程、原理、实验设计与实验分析

根据我们的研究内容，大致可将总任务分为三部分：图像处理、游戏策略的生成、机械臂的运动控制。

1、图像处理

该部分主要由李宇凡同学负责。我们通过二值图的形式，遍历二值图的各个点的像素，获得游戏棋盘的 22×10 的 0/1 矩阵。

1) 图像处理原理

为实现后续对于游戏的控制，我们需要获取棋盘的信息。构建了 22×10 的矩阵其中的元素为 0/1，这样能较为准确的获取完整的棋盘信息。图像处理主要是运用 opencv 库中的函数。

首先根据助教给的原始代码，截取录屏软件获得的游戏图像，然后使用 cvtcolor、threshold 以及改变图片像素值得到黑白图。使用 mat 类下的 at 函数读取像素值，将 220 个格子像素值存储在 $t[220]$ 。数组中像素值为 255 时，相当于方格中有物块。像素值为 0 时，方格中没有物块。判断 t 数组中像素值，

将其转换为二维矩阵 board[22][10]。

2) 研究过程

1、熟悉 mat 类以及图像处理的函数

我们小组主要通过在网上学习代码，查阅资料，尝试着做一些图像处理小程序，来熟悉 opencv 基础函数的用法。小组完成了老师在课上布置的题目，以及完成大作业，基本掌握了 opencv 的使用。

2、思考运用边缘提取算法的处理方法

开始时，我们小组是想通过 sobel 算子，来获取图像的边缘可以获得物块的形状。但后来发现，由于俄罗斯方块下部的棋盘会有许多的井，使用边缘提取，既无法准确完整的获取到棋盘的信息，又难以进行处理。所以最终我们小组放弃了这个想法。

3、思考图像遍历处理方法

为了准确完整的获取图像信息，我想到了构建一个 22*10 的矩阵的方法，这样就不会有边缘提取的算法中出现的问题。最后我们选择使用 mat 类下的成员函数 at 读取像素的方法，这样可以快速的访问 mat 类的成员，得到相应的 0/1 矩阵。我们在设计过程中发现 at 函数后面坐标参数，与我们常用的横纵坐标相反，这个点卡了好久。最终得到解决，项目才得以进行下去。

3) 实验设计及分析

我们将 ipad 上的图像进行录屏截图，然后使用完成的算法进行处理，得到的棋盘矩阵 board[22][10]，与实际的游戏棋盘进行对比。输出的矩阵与实际的游戏棋盘信息完全相同，图像处理的效果很好。

2、游戏策略的生成

该部分主要由我完成。主要是基于较为成熟的 Pierre Dellacherie 算法，通过获得的棋盘矩阵，进行处理得到六个反应局势的参数，并利用该算法得到一个最好的物块落点，输出对应的旋转操作数、移动操作数。

1) 策略算法原理

Dellacherie 评分函数的基本原理是，根据已知的落块，对落块各个角度以及每个可能的落点进行遍历，并对各个落点计算 6 个指标，加权得到最终评分，以此评分作为找出最佳落点的依据。指标详解如下：

1、落点高度，对应代码中变量 landing_height。该指标衡量落块在该落点的最终高度，通常来说我们希望落点较低。该指标的评分加权为负。

2、消行数与贡献方块数的乘积，对应代码中变量 row_eroded。该指标的评分加权为正，且是唯一一项正加权指标。

3、行变换数，对应代码中变量 row_transition。该指标意为某落点落块落下并消行（如果能够）后，方块堆的各行内的“变换次数”，某行中当一个方块的状态与旁边的方块状态不同则行变换数加一，用以评判方块堆的“平整度”。该指标的评分加权为负。

4、列变换数，对应代码中变量 col_transition。类似行变换数，一位方块堆的各列内的“变换次数”。该指标的评分加权为负。

5、空洞数，对应代码中变量 buried_holes。在某落点落块落下并消行（如果能够）后得到的方块堆中，如某一列的最上方方块下有 N 个空位，则该指标加 N。该指标的评分加权为负。

6、井，对应代码中变量 wells。在某落点落块落下并消行（如果能够）后得到的方块堆中，如果某空位的左右两侧都有方块，则该空位构成井的一部分。

对一个高度为 N 的井，该指标加 $1+2+\dots+N$ 。该指标的评分加权为负。

以上六个指标中，`landing_height` 和 `row_eroded` 是与方块本体相关的，另四个指标与最后形成的新方块堆有关。

该评分函数的核心思路是，在追求低落点和消行数的同时，兼顾形成的新方块堆的内外部“平整性”，避免不利的布局比如内部空洞、过高的井以及过于不平坦的表面出现。

由于这个算法是只与当前下落物块有关的，所以为了与图像处理部分对接，我们选择截取正在下落的物块下落到第四行时的图片，通过分析上面四行的 0/1 情况可以得到正在下落物块的种类，而分析下面几行的 0/1 情况可以得到下方堆叠的物块的情况，即可得到以上需要的指标。

我们小组对于各个参数的权重做了一定的微调，使得这个算法更为精确。

2) 研究过程

我从网上论坛中第一次看到了这个算法，觉得这个算法十分适合这个项目。通过网上搜索我们发现，这个算法已经十分成熟并且被广泛的运用。所以我们从网上找到了这个算法的介绍文档，以及该算法的代码，进行了深入的学习。最后在参考资料的基础上，完成了我们自己的代码。

之后在实验的基础上对于代码中的参数进行微调，多次的实验重复，使得算法更为适合这个游戏。

3) 实验设计及分析

由于这个算法完成时，图像处理部分成功做完了，而机械臂控制部分还有较多的问题。我们小组采用的实验方法是，由录屏软件截取游戏画面并有程序处理后，输出了旋转操作数与移动操作数。然后我们人工手点，观察物块的落点和旋转次数是不是合适的。如果出现问题，我们可以针对每种物块的下落的情况进行 debug。多次实验调整后，算法最终完成。

3、机械臂的运动控制

该部分主要有徐双捷同学负责。机械臂完成移动到对应的按钮，并按次数点击。

1) 机械臂运动控制的原理

移动操作我们采用了 `device->comMoveTo(,)` 函数，该函数的坐标值为绝对坐标值，方便我们进行标定。

点击我们采用了 `device->comHitOnce()` 函数，使用较为方便，触笔点击也没有出现过问题。

指令传输以及延迟是机械臂控制中比较大的问题。由于我们小组的算法是每次物块下落到第四行时进行一次运算，然后有一个输出，机械臂做一次操作。我们在程序中使用了 `sleep` 函数，使得物块下落到第四行计算输出后，机械臂完成操作，程序停止运行。这样就不会使得程序在物块下落到第四行是多次运算发送多次指令，对机械臂发指令过快，出现 bug。等待下一个物块落到第四行时，再次触发程序运行。

另一个延迟就是机械臂控制中的延迟。延迟操作的时间，不会使得触笔卡住。

2) 研究过程

我们按照一开始的想法，完成了移动及点击的操作。我们发现这两个操作较为简单，并没有什么问题。问题主要是出现在延迟上。我们的设定是在第四行有东西，可以判断出正在下落的物块类型时，执行俄罗斯方块算法。但是这个时间点整个图像处理函数会执行好几遍，也就意味着会重复发送命令，导致错误。

所以我们用 sleep 函数增加延迟,使得程序只发送一遍指令。

第二个延迟部分就是机械臂控制中,当触笔移动到按钮上方时与点击之间需要一个延迟,以及两次点击之间也需要一个延迟。我们一开始使用了 qtcyberdip 中的 comDeviceDelay(delay)函数,一直出现问题。我们寻求助教帮助,后来发现似乎是这个 comDeviceDelay(delay)函数本身存在问题,无法使得机械臂延迟操作。后来我们使用 window 自带的 sleep 函数才解决了这个问题。

最终完成了机械臂控制的部分

3) 实验设计及分析

我们通过不使用 pad 进行虚点的方法,标定坐标值点击次数。并多次调整了延迟时间,最终完成了我们的代码。

(三) 研究成果

对比开题报告中的研究指标,详细论述研究成果

● 方块的识别

我们采用静态方块的读取算法,准确、快捷地读取了游戏棋盘方块的情况,为后面的游戏处理做好了基础。

● 俄罗斯方块策略算法

我们根据已有的成熟算法进行改进,使得我们的算法消除行数的上限足够高,机器的操作时间足够,并且再出现因下落物块不理想而行数较高时算法能够进行处理。在机械臂操作的情况下,消除的行数能达到 40 到 50 行左右。

(四) 研究总结

1. 研究中存在问题、建议及需要说明的问题

本次研究中还存在这一些问题

(1) 触笔点击有时还存在问题。有时候会出现触笔点击,而实际上没有点到 pad 屏幕的情况。

(2) 程序的稳定性不够完美。大部分时候程序能正常运行,不过有的时候机械臂操作会出现 bug,突然移动到游戏区域外,卡住的情况。由于这种情况大部分时不出现,我们没有想到好的方法 debug 解决这个问题

(3) 策略算法仍有改进的空间。由于这个游戏会提供之后将要下落的两个物块的情况,我们并没有使用到这个信息。我们设想可以进行多层的搜索,使得算法更为完美。不过我们时间不充裕,且这个多级的策略难度很大,我们没有实现。

(4) 项目过程前松后紧,整体有些拖沓。

2. 在本次课程实践中的感想与体会

通过这次课程设计,我学习了图像处理的一些基础知识,并把这个知识投入了应用,成功实现了一个自动的游戏操作仪器。我们从无到有,从一开始对于这方面的知识一无所知,自己查阅资料并进行实践,最后取得了很大的进步。完成整个程序的时候,还是有很大的成就感的。

同时,我也体会到了小组合作与交流的重要性,几个人一起想点子找解决方法,工作效率会有很大的提升。

整个过程中助教很辛苦,从软件的安装、硬件的调试包括课程设计本身都给了大家很多帮助,非常感谢助教。

3. 对本次课程实践的意见与建议

我觉得这个课程实践给我们带来挺大困扰的有两点。一是项目开始时,没有入门的学习材料,全靠自己摸索,上手太慢了,后面的时间就不够充裕。

二是那个 dip 机器以及它的代码部分似乎不够完美。我们小组在机器上就遇到了不少问题，出现过触摸笔头掉落，机器卡顿，以及控制函数无法使用的问题。希望这个机器能够有一些改进，使它更为完美，方便之后的使用者。