机器视觉课程设计

报告

**姓名：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**学号：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**学院：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**专业：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

目录

[一、 概述 5](#_Toc43282332)

[二、 课程设计任务及要求 5](#_Toc43282333)

[三、 算法设计 5](#_Toc43282334)

[四、 实验及数据分析 5](#_Toc43282335)

[五、 结论 6](#_Toc43282336)

[六、 收获、体会和建议 6](#_Toc43282337)

[七、 参考文献 6](#_Toc43282338)

1. **概述**
2. **课程设计任务及要求**

实现8层金字塔的NCC模板匹配算法。

1. **算法设计**

算法分为模板构建和递归匹配两部分。

对于构建模板，注意到NCC计算公式

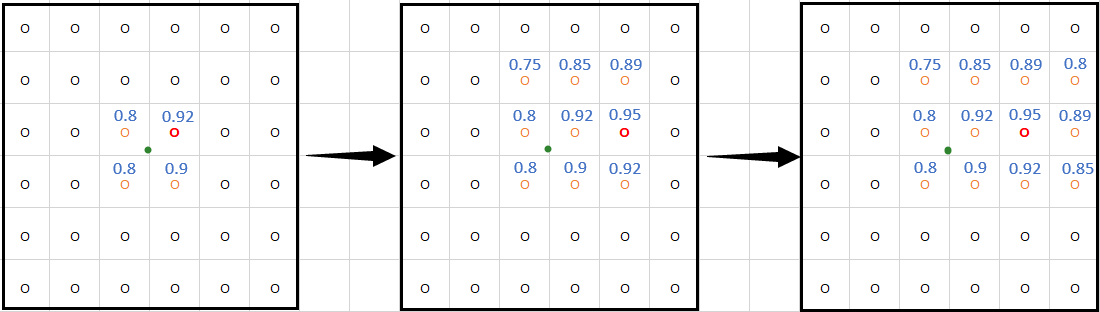
其中

这一部分仅与模板有关，可以提前算出模板在金字塔各层的矩阵，以节省与图像匹配时的计算开销。

具体算法为：将模板图像转化为单通道、归一化（0.0~1.0）图像，逐层使用3x3的高斯核滤波后，缩小图像长宽为原来的一半，将结果图像作为上一层。然后计算该层模板的矩阵。

递归匹配部分，从高斯金字塔最高层向下递归。首先类似模板金字塔的构建，生成图像的高斯金字塔。由于课题所提供Opencv1.0库中的矩阵数据类型有按元素乘算的方法，所以只需算出

然后与对应层的矩阵按元素乘算，并计算结果矩阵各元素加和，即可得到图片目标ROI位置与模板的NCC值。移动ROI，算出并储存对应ROI的NCC值。对于金字塔顶层的图片，ROI移动应遍历所有可能像素；对于低层的图片，则从上一层结果来计算该层临近的四个像素，并以这四个像素开始，计算搜索周围8邻域内NCC最大值所在位置，如图：



搜索到局部最大值、并保证其周围8邻域的NCC全被计算后，通过曲面拟合的方式计算大致的亚像素位置。由于曲面拟合时，若采用实际坐标矩阵来进行运算，过于繁琐，于是采用如下常数坐标值，计算目标亚像素相对于NCC最大值点的偏移坐标，以降低部分运算量以及代码复杂度：

从而

从而

将9个NCC值代入运算，算出

前五个值，然后利用

即可得到当前层目标亚像素值。向下层递归时，将亚像素值除以当前图像长宽，再乘以下层图像长宽，即可得到对应于下层的坐标，并以这个坐标开始重复上述步骤。直至最底层，最底层得到的亚像素值即为目标结果。

1. **实验及数据分析**
2. **结论**
3. **收获、体会和建议**

此次课程设计，巩固了我们写算法的能力，让我们了解到了不少优化手法。

此外，在设计过程中，我们注意到了一种优化方法：

由于在构建金字塔时，我们使用了3x3的高斯核，所以在向下层计算时，通过上一层估计的极点坐标值与实际极点坐标值之间相差大概率在1.6个像素以内，从而可以根据这前后1.6的范围，简化ROI的移动规则，以减少NCC值的计算次数。

实际结果比起上面所述的算法平均节约了四分之一的时间，而且仅在亚像素精度上有所差异。

1. **参考文献**