

机器视觉课程设计 报告

姓名： 吕家昊 李君程

学号： 210320111 210320412

学院： 机电工程与自动化

专业： 自动化

目录

一、	概述.....	3
二、	课程设计任务及要求.....	3
三、	算法设计.....	3
四、	实验及数据分析.....	6
五、	结论.....	6
六、	收获、体会和建议.....	6
七、	参考文献.....	错误!未定义书签。

一、概述

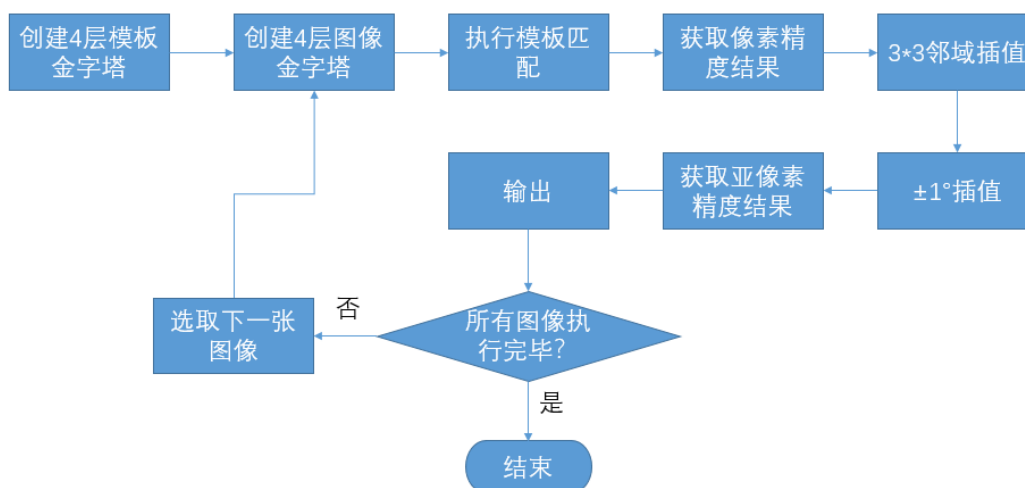
在一个 4 层金字塔 NCC 模板匹配算法基础上增加角度的输出，即模板匹配可以输出模板的(x,y)坐标以及模板和目标间的角度。

二、课程设计任务及要求

实现 NCC 模板匹配算法的角度输出。

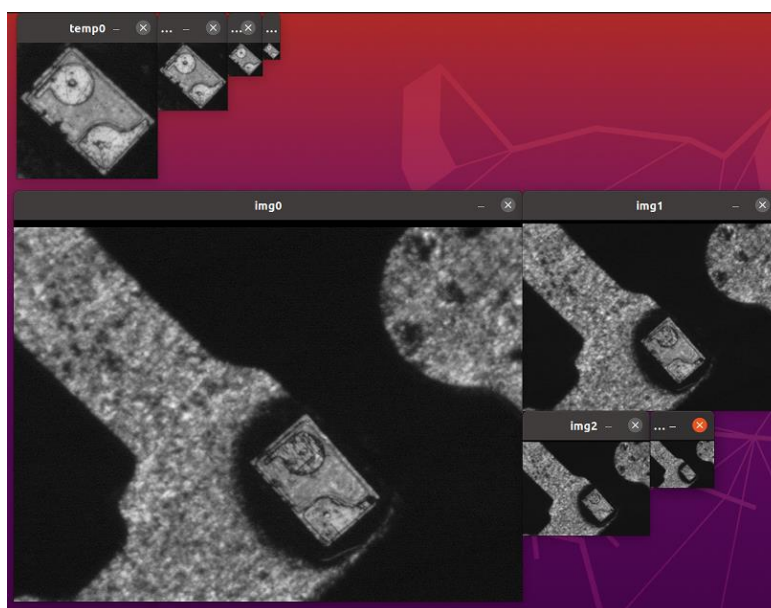
三、算法设计

程序使用 ROS20.04 + OpenCV2 进行构建。程序总体流程图如下。



程序开始，读取模板图像并转换为灰度图，将其作为第 4 层模板。对此图像进行 3*3 高斯滤波，并将图像长宽分别缩放至上一层的一半，依次得到第 3,2,1 层模板。

当读取一张图像后，以同样的方式依次进行高斯滤波、缩放，得到 4 层图像金字塔，模板与图像金字塔如下图（temp 为模板，img 为图像，0~3 依次表示尺寸从大到小）。

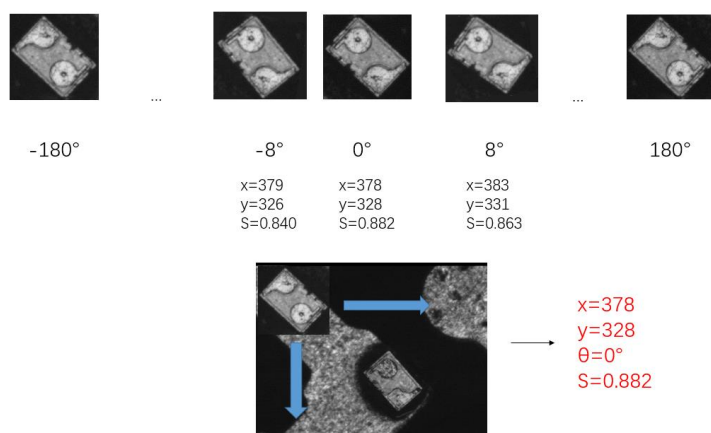


对最上层进行模板匹配。使用 NCC 模板匹配时，采用归一化相关系数匹配法（CV_TM_CCOEFF_NORMED），输出结果为

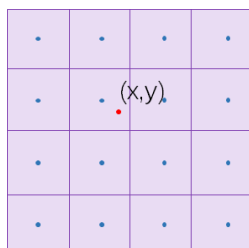
$$S = \frac{\sum_{x',y'} [T'(x',y') \cdot I'(x+x',y+y')]}{\sqrt{\sum_{x',y'} T'(x',y')^2 \cdot \sum_{x',y'} I'(x+x',y+y')^2}}$$

即 S 在[-1,1]之内，S 越大，表示图像对应 ROI 与模板相似度越高。在图像中依次移动 ROI，最终会得到一个 $S_{(x_{img}-x_{tmp}) \times (y_{img}-y_{tmp})}$ 矩阵，矩阵中最大元素 S_{max} 的坐标对应相似度最高的 ROI。

对该层模板依次进行旋转，旋转角度 $\theta \in \{-180^\circ, -172^\circ, \dots, 180^\circ\}$ ，并将旋转后的模板分别进行匹配。由于每个角度会对应一个最高分值 $S_{max}(\theta)$ ，因此对 $S_{max}(\theta)$ 关于 θ 取最大值即可得到最匹配的位置 (x_1, y_1) 与角度 θ_1 。



对于图像旋转，为减少变换过程中图像的失真，变换后采用三次插值与复制填充（边缘像素值）。在 OpenCV 中实现为构建仿射变换矩阵，然后对图像应用仿射变换。此后，将转换后的各像素点对应到原图的坐标 P_1 的 4*4 邻域，如下图所示。



由于每一个像素点与 P_1 的距离均小于 2，构造以下函数：

$$W(x) = \begin{cases} 1.5|x|^3 - 2.5|x|^2 + 1, & |x| \leq 1 \\ 0.5|x|^3 - 2.5|x|^2 + 4|x| - 2, & 1 < |x| < 2 \\ 0, & |x| \geq 2 \end{cases}$$

该函数为每个像素点在 x,y 方向上分别对应的“权重”值。插值得到的结果为 $\sum I(x,y)W(x)W(y)$ ， $I(x,y)$ 为选取的邻域内像素点。

得到第 1 层的结果位置与坐标后，接下来对第 2 层进行匹配。由于第 2 层图像与模板长宽均为第 1 层的 2 倍，因此对应坐标均变为上一层 2 倍，即 $(2x_1, 2y_1)$ 。为提高运算速度，仅对 $2x_1 - 6 \leq x \leq 2x_1 + 6$ ， $2y_1 - 6 \leq y \leq 2y_1 + 6$ 范围内的点进行匹配，即对图像裁剪后执行 matchTemplate 函数。类似地，仅取旋转角度 $\theta \in$

$\{\theta_1 - 16, \theta_1 - 12, \dots, \theta_1 + 16\}$ 。经过本层模板匹配后，可得到精度更高的像素级匹配结果（相对原图为 4 像素、 4° ）。

类似地，执行第 3、第 4 层模板匹配，同样根据上一层的结果减小 ROI 与角度范围，取角度步进值为 2° 、 1° ，位置精度为 2 像素、1 像素。执行完成后，可得到最匹配的坐标 (x_4, y_4) 与角度 θ_4 。由于模板匹配输出为图像左上角坐标，因此匹配中心坐标应为 $(x_4 + \frac{x_{tmp}}{2}, y_4 + \frac{y_{tmp}}{2})$ 。

接下来进行亚像素精度求解。对坐标求解过程如下：取上述得到的角度 θ_4 ，依次对 3×3 邻域求匹配分值，拟合二次曲面 $k_0x^2 + k_1xy + k_2 + k_3x + k_4y + k_5 = S(x, y)$ 。方便起见，令 3×3 邻域坐标为

(-1,-1)	(-1,0)	(-1,1)
(0,-1)	(0,0)	(0,1)
(1,-1)	(1,0)	(1,1)

将 $[x^2 \ xy \ y^2 \ x \ y \ 1]$ 代入以上值得

$$A_{9 \times 6} = \begin{bmatrix} 1 & 1 & 1 & -1 & -1 & 1 \\ 0 & 0 & 1 & 0 & -1 & 1 \\ 1 & -1 & 1 & 1 & -1 & 1 \\ 1 & 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & -1 & 1 & -1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

而 $F_{9 \times 1} = [S(-1, -1), S(0, -1), \dots, S(1, 1)]^T$ ，由最小二乘可得

$$[k_0, k_1, \dots, k_5]^T = (A^T A)^{-1} A^T F$$

对二次曲面在 x, y 方向上求偏导，则

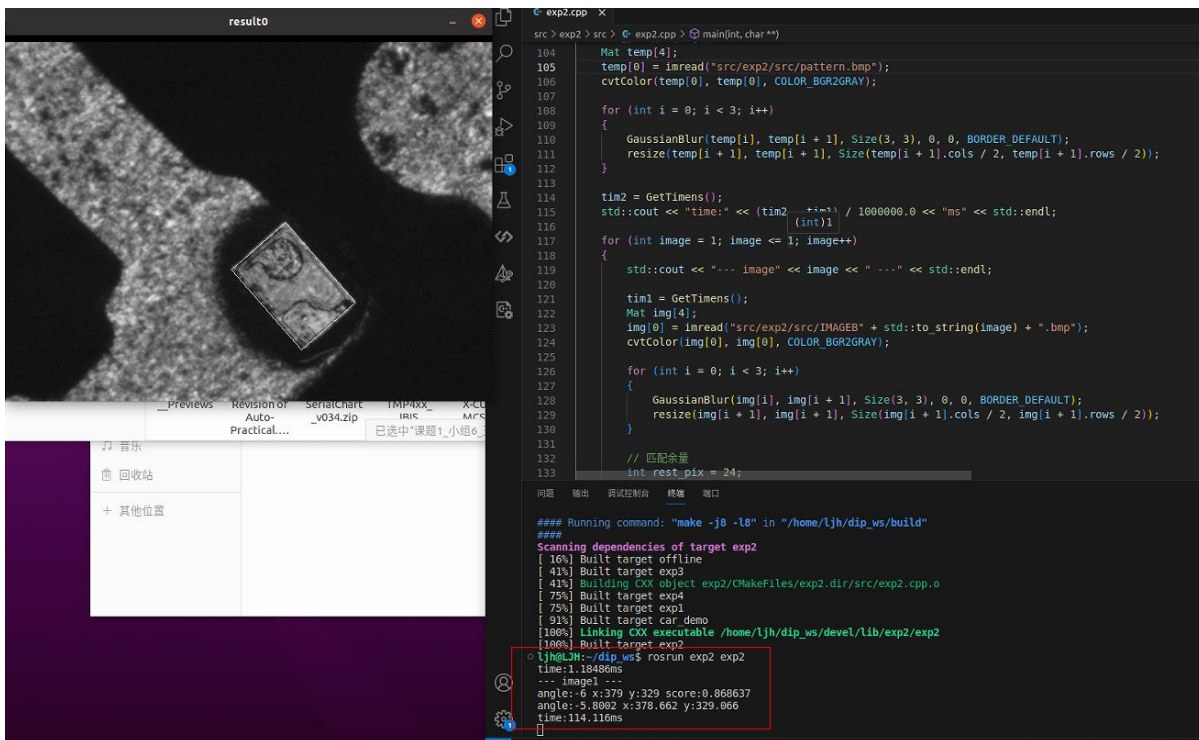
$$\begin{cases} x_{sub} = \frac{2k_1k_3 - k_2k_4}{4k_0k_1 - k_2^2} \\ y_{sub} = \frac{2k_0k_4 - k_2k_3}{4k_0k_1 - k_2^2} \end{cases}$$

将其转换到实际坐标即 $(x_4 + \frac{x_{tmp}}{2} + x_{sub}, y_4 + \frac{y_{tmp}}{2} + y_{sub})$ 。

对角度求解采用类似的方式，取 $\theta_4 - 1^\circ, \theta_4, \theta_4 + 1^\circ$ 在左上角为 (x_4, y_4) 处匹配得到的分值，拟合二次函数 $k_0\theta^2 + k_1\theta + k_2 = S(\theta)$ （代入 $\theta = -1^\circ, 0, 1^\circ$ 求解），得到相对角度 $\theta_{sub} = \frac{2(k_0 - k_2)}{k_0 + k_2 - 2k_1}$ ，输出结果为 $\theta_4 + \theta_{sub}$ 。

最后，在匹配完成后的图像上绘制出模板对应轮廓。为更直观展示匹配效果，窗口中绘制出物体的长方形轮廓，其长宽分别为 95 与 140，旋转角度为 $\theta_4 - 47^\circ$ （由于模板中该图形倾斜，此参数为绘制后调整得到，仅作为效果展示，并不完全准确）。

运行结果如下图：



四、 实验及数据分析

4 层模板金字塔构建时间为 1.18486ms。

对 34 张图像执行该模板匹配算法，平均每幅图像用时为 60.1688ms，最大用时 115.806ms，最小用时 55.1377ms，识别率 100%。

使用 Halcon 进行模板匹配，平均用时 1.32ms，最大用时 1.74ms，最小用时 0.96ms，识别率 100%。

五、 结论

以上算法能在 4 层金字塔 NCC 模板匹配的基础上，实现角度的输出。同时利用二次曲线（面）插值方法，可对亚像素精度的位置与角度进行求解。

六、 收获、体会和建议

这次课设加深了我对模板匹配算法的理解，也使我学习到了一些模板匹配算法中提高效率的方法。

此算法相比于 Halcon 成熟的算法，执行时间约是其 46 倍，说明此算法有较大的优化空间。

当识别一层后，下一层选取了较大的位置与角度范围进行匹配，事实上这一范围可进一步缩小。

对于亚像素精度的位置匹配， $A_{9 \times 6}$ 矩阵为定值，因此可直接使用数组进行储存，减少代码运算量。更进一步，可直接计算出 $(A^T A)^{-1} A$ 并储存在代码中。

第 4 层匹配后，可直接返回分值矩阵，便于下一步亚像素计算，而不需要重新计算 9 次 1 像素点的模板匹配（3*3 邻域）。