



哈爾濱工業大學(深圳)

HARBIN INSTITUTE OF TECHNOLOGY, SHENZHEN

机器视觉课程设计 结题汇报

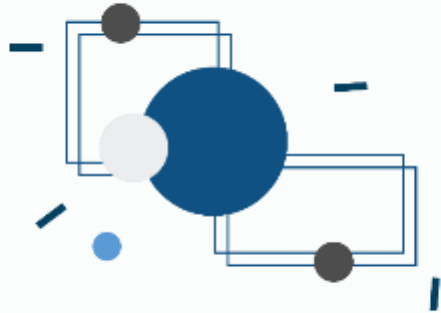
小组成员:

2024. 5. 31

规格严格 | 功夫到家



目 录

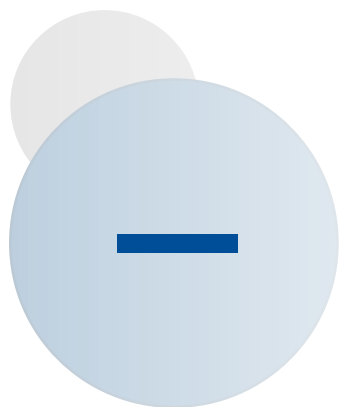


一 选题分析

二 算法原理与程序设计

三 运行效果展示

四 总结与展望



选题分析



一、选题分析



- **题目：**(第2题) 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。
- **基本思路：**将给定的模板进行旋转，**生成旋转 0° 、 1° 、.....的模板**，并计算这些模板与图像间的NCC，得出**NCC匹配分数最高**的模板和目标点，此模板的旋转角度即为所求原模板和目标间的角度，此目标点坐标即为所求(x,y)坐标。
- **注意到：**需充分利用源代码中实现的图像金字塔处理方法，**减少匹配角度的次数与目标点数**。





二

算法原理与程序设计

1 在打开模板图像时，按照 1° 的步长生成旋转 $0-359^\circ$ 的图像

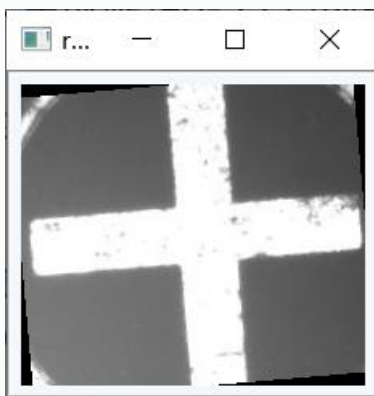
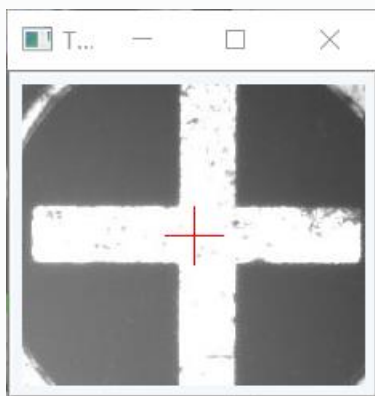
```
cv2DRotationMatrix(cvPoint2D32f(theCpTemplate->width / 2, theCpTemplate->height / 2),  
index, 1, matrix);
```

模板中心点

旋转角度 缩放因子 生成的旋转矩阵

```
cvWarpAffine( 源图像 , 变换后图像 , 旋转矩阵 , 双线性插值 +  
补全旋转后的空位 , 补为全0 (黑色) );
```

```
TemplateArr[index] = TempImagePtr;
```



加速策略1.1：模板在检测前预处理



- 1 在打开模板图像时，按照 1° 的步长生成旋转 $0-359^\circ$ 的图像
- 2 对于每个待匹配的角度，都从刚才生成的若干模板中取出对应角度的模板，并进行匹配，获取分数与匹配点的位置

```
cvCopy(TemplateArr[index], theTemplate, NULL);
```

```
c_imgproc.AllocateMemory(theImage->width, theImage->height, theTemplate->width, theTemplate->height);
```

```
c_imgproc.PreProcessTemplate(theTemplate);
```

取出模板并进行匹配

```
c_imgproc.m_pTemplate = theTemplate;
```

```
c_imgproc.SingleTemplateMatch(theImage->imageData, theImage->width, theImage->height, theImage->widthStep, theImage->nChannels);
```

```
location_fxy[index] = c_imgproc.GetTargetLocation();
```

```
rotation_score[index] = c_imgproc.m_MaxNccPoint.score;
```

获取分数及匹配点位置

```
location_point[index][0] = c_imgproc.m_MaxNccPoint.width;
```

```
location_point[index][1] = c_imgproc.m_MaxNccPoint.height;
```





- 1 在打开模板图像时，按照 1° 的步长生成旋转 $0-359^\circ$ 的图像
- 2 对于每个待匹配的角度，都从刚才生成的若干模板中取出对应角度的模板，并进行匹配，获取分数与匹配点的位置
- 3 将本次匹配的分数与之前存储的最大分数相比较，若更大则将最佳匹配点换成本次匹配点

加速策略1.2: 若分数过小, 则增大搜索步长

```
if (abs(rotation_score[index]) < 0.5 && (index + 5 < m_SetMaxRotationAngle ||  
((index >= 360 - m_SetMaxRotationAngle - 1) && index <= 355))) index += 4;
```

加速策略1.3: 利用设定的角度范围, 跳过不在所要求角度范围内的点

```
if (m_SetMaxRotationAngle > 0){  
    if (index >= m_SetMaxRotationAngle && index <= 360 - m_SetMaxRotationAngle - 1){  
        index = 360 - m_SetMaxRotationAngle - 1;  
    }  
}
```





二、算法原理——方法一

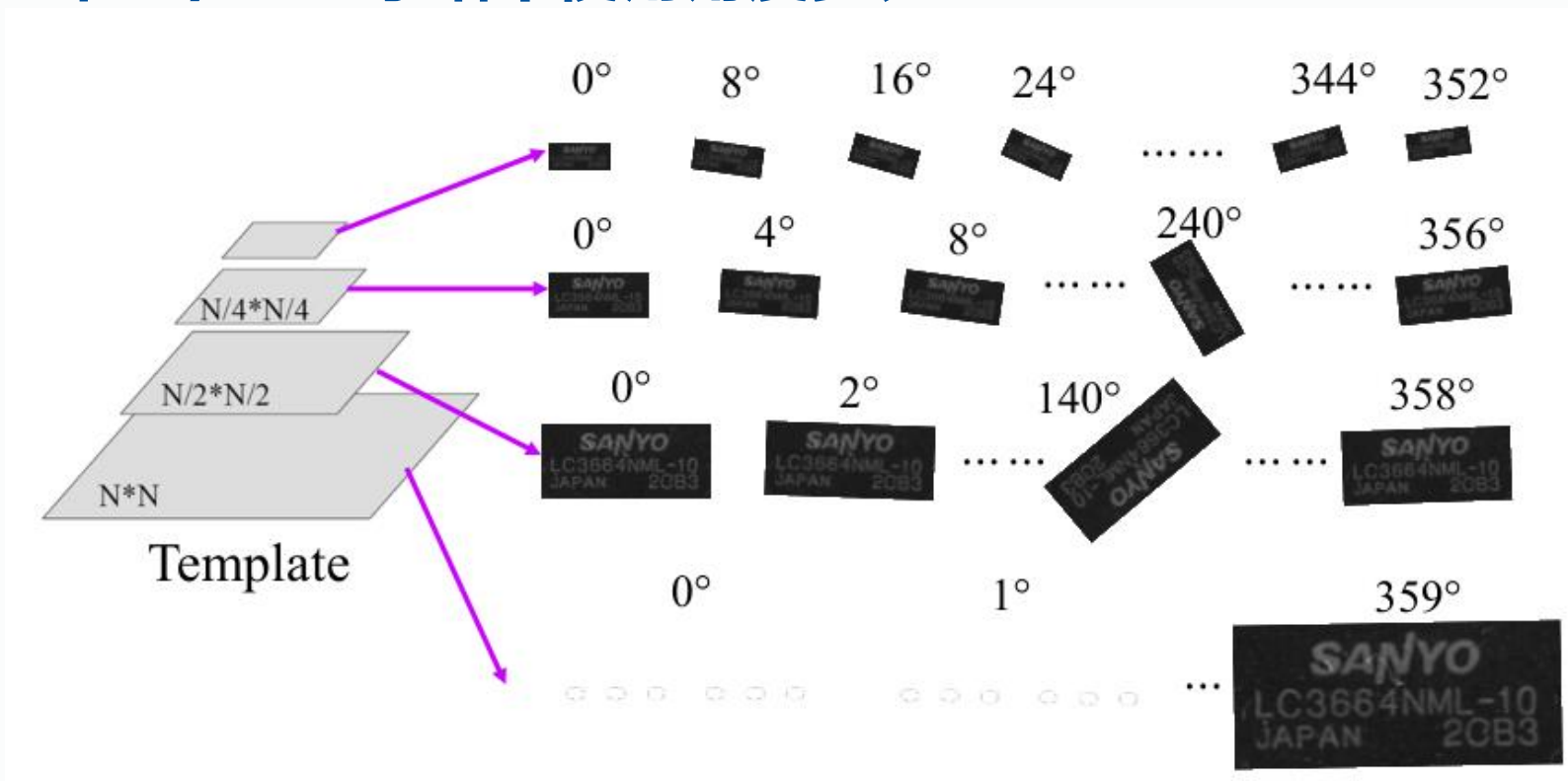


- 1 在打开模板图像时，按照 1° 的步长生成旋转 $0-359^\circ$ 的图像
- 2 对于每个待匹配的角度，都从刚才生成的若干模板中取出对应角度的模板，并进行匹配，获取分数与匹配点的位置
- 3 将本次匹配的分数与之前存储的最大分数相比较，若更大则将最佳匹配点换成本次匹配点
- 4 匹配结束后，存储最佳匹配点坐标及分数，停止计时并显示时间，绘制结果图形。



问题：用于每层金字塔上的角度搜索步幅是相同的 (1°)，而在高层金字塔上不需要如此小的角度步幅。

加速策略2.1：在 $1/2^k$ 金字塔中使用角度步长 2^k





二、算法原理——方法二



- 1 先在最高层（1/8 层）上按旋转角度 0° 、 8° 、 16°进行匹配，对于每个角度，都选出 NCC 分数最高的 12 个点。比较每个角度的最高分，**选出最高分最大的角度 θ_8**

加速策略 2.2: (提供的代码中的思路)

`NCC = 256.0 - (double)nSubSquareSum / SquareFunc(nTemplateWidth * nTemplateHeight);`

本层的 NCC 计算与其余各层不同，仅计算模板像素值与目标图像 ROI 像素值之差的平方和，并未进行归一化的计算，再通过排序选出该指标最大的点（更像 SSD）。这样既可以初步筛选出差异较小的若干个点，也显著降低了计算量。





二、算法原理——方法二



- 1 先在最高层（1/8 层）上按旋转角度 0° 、 8° 、 16°进行匹配，对于每个角度，都选出 NCC 分数最高的 12 个点。比较每个角度的最高分，**选出最高分最大的角度 θ_8**
- 2 将上述选出的角度下 NCC 分数最高的 12 个点映射到 1/4 层（得到 108 个点），按旋转角度 $\theta_8 - 4$ 、 θ_8 、 $\theta_8 + 4$ 来匹配其中每个点，**选出分数最高的一个点并记下旋转角度 θ_4**
- 3 将上一步所得点映射到 1/2 层（得到 16 个点），按旋转角度 $\theta_4 - 2$ 、 θ_4 、 $\theta_4 + 2$ 来匹配其中每个点，**选出分数最高的一个点并记下旋转角度 θ_2**
- 4 将上一步所得点映射到 1/1 层（得到 9 个点），按旋转角度 $\theta_2 - 1$ 、 θ_2 、 $\theta_2 + 1$ 来匹配其中每个点，**选出分数最高的点并记下旋转角度 θ** ，匹配完成
- 5 匹配结束后，存储最佳匹配点坐标及分数，停止计时并显示时间，绘制结果图形。

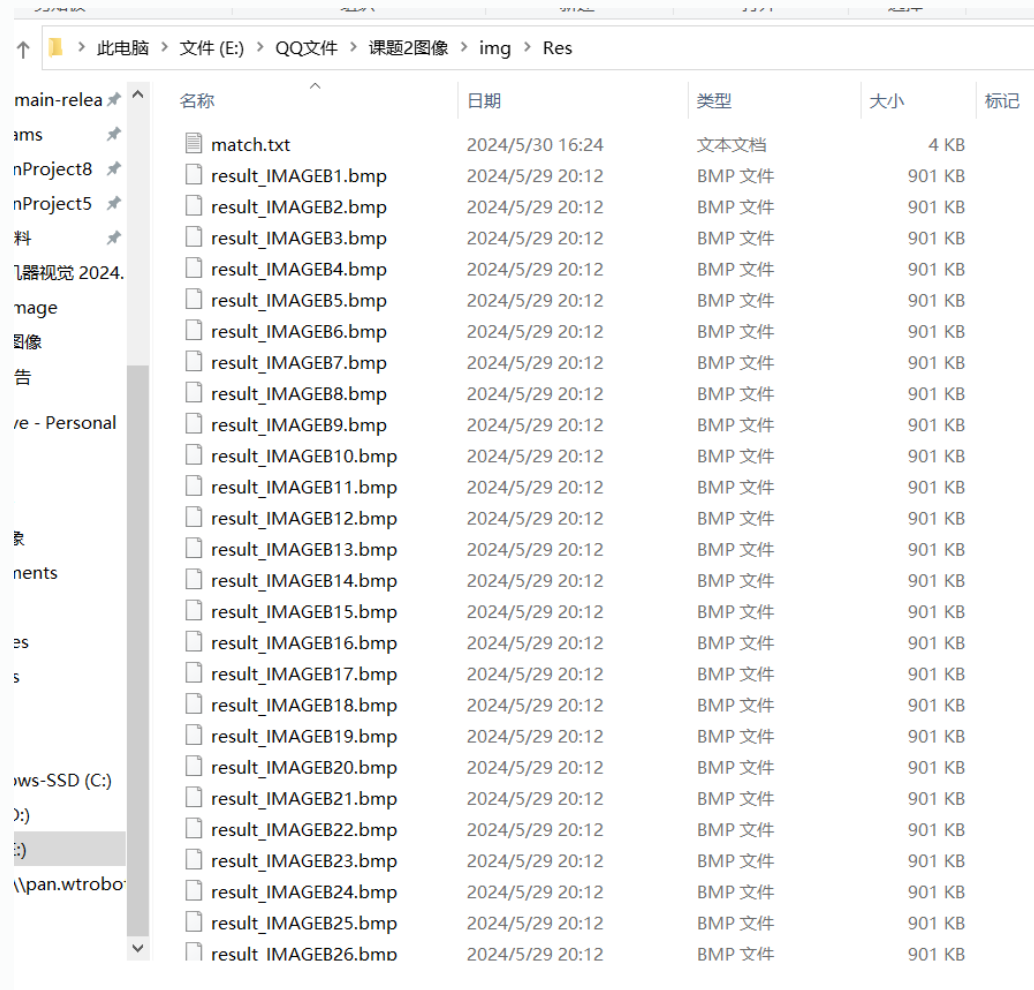




二、程序设计——批量处理及打印结果



修改了图形界面，点击“打开文件夹”，再选择“图像匹配(文件夹)”，设置好扩展名即可批量匹配该文件夹下所有符合格式的图像。





三

运行效果展示



三、运行效果展示



The screenshot displays the Visual Studio IDE with the following components:

- Code Editor:** Shows the implementation of `TemplateMatchDlg.cpp`. The code includes a check for folder and template paths, directory creation, and file processing using `cvLoadImage`.
- Template Matching Dialog:** A modal dialog box with the following sections:
 - 系统初始化 (System Initialization):** Buttons for "打开图像" (Open Image), "打开文件夹" (Open Folder), and "打开模板" (Open Template).
 - 控制部分 (Control Section):** Buttons for "设置方法与参数" (Set Method and Parameters), "图像匹配(单文件)" (Image Matching (Single File)), and "图像匹配(文件夹)" (Image Matching (Folder)).
 - 旋转角度设置 (Rotation Angle Settings):** Input fields for "设置最大旋转角度" (Set Maximum Rotation Angle) and "设置最小旋转角度" (Set Minimum Rotation Angle), both set to 0. A text field for "图像文件扩展名" (Image File Extension) is set to ".bmp". A "退出系统" (Exit System) button is also present.
 - 输出 (Output):** Fields for "匹配时间" (Matching Time) in ms, "最大匹配值" (Maximum Matching Value), and "是否报警" (Whether to Alarm).
 - 处理文件 (Process File):** A large text area for file processing details.
- Output Window:** Shows the application's execution output, including the message "请先选取待处理文件夹以及模板!" (Please select the folder to be processed and the template first!).
- Solution Explorer:** Shows the project structure for "TemplateMatch", including source files and resource files.



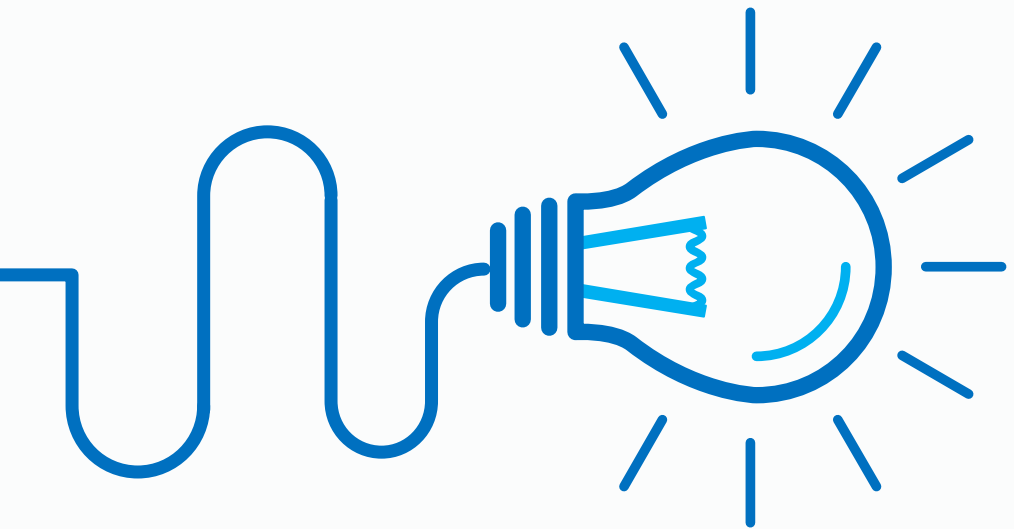


四

总结与展望



四、总结与展望



- 所实现的算法**能完成规定的任务**，且速度较快，与原先无旋转的模板匹配时间差异不大；
- 在此基础上，增加了**整文件夹图像批量识别**功能并可打印结果、保存匹配图形。
- **改进：**
 - 暂时只能实现目标图像中**单一目标**的匹配，日后可尝试将其扩展到**多目标匹配**；
 - 匹配的角度**只是整数**，角度范围仅有最大值限制而**最小值固定为0**；
 - 与成熟的视觉、图像处理库中的模板匹配算法相比，**仍有加速的空间**。





哈爾濱工業大學(深圳)

HARBIN INSTITUTE OF TECHNOLOGY, SHENZHEN

谢谢!

小组成员:

2024. 5. 31

规格严格 | 功夫到家