

机器视觉课程设计 报告

姓名：___ 赵乙霖 、 苏晗 ___

学号：___ 200320404 、 200320409 ___

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	5
五、	结论	6
六、	收获、体会和建议	6

一、概述

在一个 4 层金字塔 NCC 模板匹配算法基础上增加了角度的输出，运用双线性插值算法进行模板旋转构建四层金字塔匹配，即模板匹配最终可以输出模板的(x,y)坐标以及角度。

二、课程设计任务及要求

在一个 4 层金字塔 NCC 模板匹配算法的 C++VS2008 和 Opencv1.0 的源码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。

三、算法设计

旋转图像函数 `IplImage* CDBImageProcess::Rotate(IplImage* pImage, double angle)`

基本思路：利用仿射变换，计算出旋转后图像对应的旋转前坐标，再利用双线性插值得出该点的灰度值

输入：需旋转的 `IplImage*` 类型图像，旋转角度

输出：旋转后的 `IplImage*` 类型图像

```
IplImage* CDBImageProcess::Rotate(IplImage* pImage, double angle)
{
    int Width = pImage->width;
    int Height = pImage->height; // 定义原图像长宽
    // 计算旋转后的长宽
    double theta = angle * (3.1415926 / 180.0);
    double costheta = cos(theta);
    double sintheta = sin(theta);
    int nWidth = ceil(fabs(double(Width) * costheta) + fabs(double(Height) * sintheta));
    int nHeight = ceil(fabs(double(Width) * sintheta) + fabs(double(Height) * costheta));
    CvSize size{nWidth, nHeight};
    IplImage* temp = cvCreateImage(size, 8, 1); // 新建一个图像 大小为旋转后的长宽 8depth 1channel
    temp->imageData = (char*)cvAlloc(temp->widthStep * temp->height);
    temp->imageDataOrigin = temp->imageData;    cvSetData(temp, temp->imageData, nWidth
* 1 * sizeof(unsigned char));    // 旋转逆运算变换矩阵
    double m1[3][3] = {1, 0, 0, 0, -1, 0, 0.5 * double(nHeight), 0.5 * double(nWidth), 1};
    double m2[3][3] = {costheta, sintheta, 0, (-1) * sintheta, costheta, 0, 0, 0, 1};
    double m3[3][3] = {1, 0, 0, 0, -1, 0, 0.5 * double(Height), 0.5 * double(Width), 1};
    for (int w = 0; w < nWidth; w++) // 对于旋转后的新图像    {
        for (int h = 0; h < nHeight; h++)
        {
            double ncoordinate[3] = {w, h, 1}; // 新图坐标
            double coordinate[3] = {0.0, 0.0, 0.0}; // 计算原图坐标
            for (int i = 0; i < 3; i++)
            {
```

```
        coordinate[i] = multimatrix(ncoordinate, m1, i);
    }
    double tempcoordinate1[3] = { coordinate[0],coordinate[1],coordinate[2] };
    for (int i = 0; i < 3; i++)
    {
        coordinate[i] = multimatrix(tempcoordinate1, m2, i);
    }
    double tempcoordinate2[3] = { coordinate[0],coordinate[1],coordinate[2] };
    for (int i = 0; i < 3; i++)
    {
        coordinate[i] = multimatrix(tempcoordinate2, m3, i);
    }
    //coordinate 数组中是 xy 是原图坐标 [x y 1]
    double col, row;
    col = coordinate[0];//原图纵坐标
    row = coordinate[1];//原图横坐标
    if (row <= 1 || col <= 1 || row > Width || col > Height)
        (temp->imageData + (temp->widthStep) * h)[w] = 0;
    else
    {
        //取周围四个点
        int left = floor(col);
        int right = ceil(col);
        int top = floor(row);
        int bottom = ceil(row);
        //记
        int a = col - left;
        int b = row - top;
        //双线性插值计算
        (temp->imageData + (temp->widthStep) * h)[w] = (1 - a) * (1 - b) *
        (pImage->imageData + (pImage->widthStep) * top)[left]
        + a * (1 - b) * (pImage->imageData + (pImage->widthStep) * top)[right]
        + (1 - a) * b * (pImage->imageData + (pImage->widthStep) * bottom)[left]
        + a * b * (pImage->imageData + (pImage->widthStep) * bottom)[right];
    }
}
}
return temp;
}
```

四、实验及数据分析

能够按要求输出角度值，精度为 1 度，检测结果与参考有 1.5 度左右的误差。

部分检测结果与参考如下：

IMAGE1

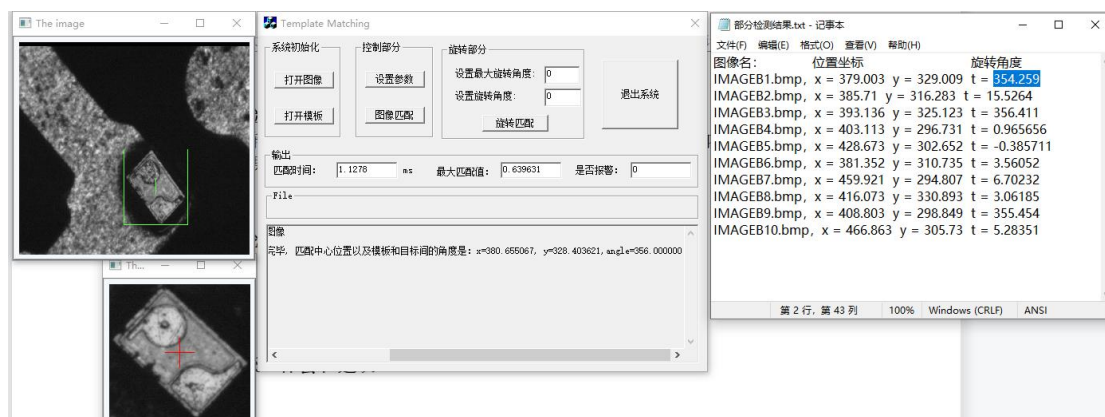


IMAGE2

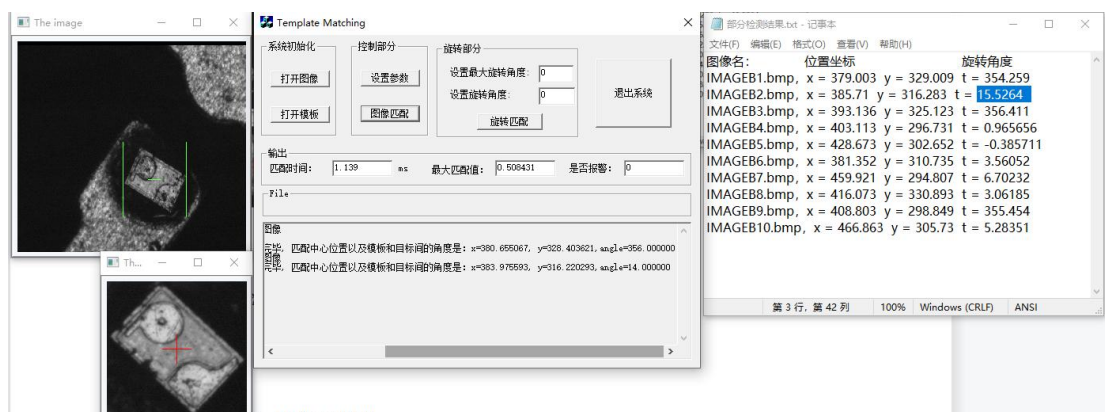
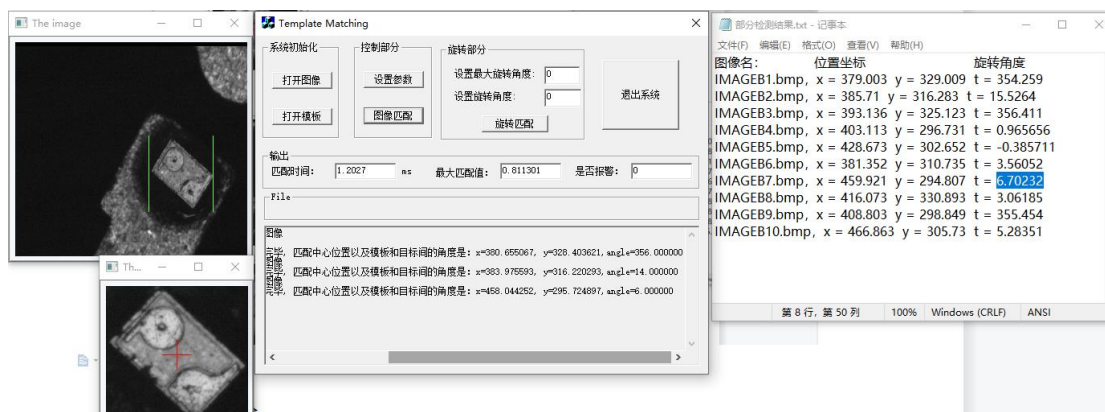


IMAGE7



五、 结论

在误差允许范围内，基本实现课程设计任务目标。

原图片目标区域灰度值与模板相差较大时，检测结果误差更大，例 IMAGE6

六、 收获、体会和建议

①学习 opencv1.0 库函数使用

②接触较大型 MFC 项目工程实现

③需深入了解算法原理及工程代码逻辑后，再进行代码编写，可以大大节省 debug 时间

七、 参考文献

<http://t.csdn.cn/Cjgzz>