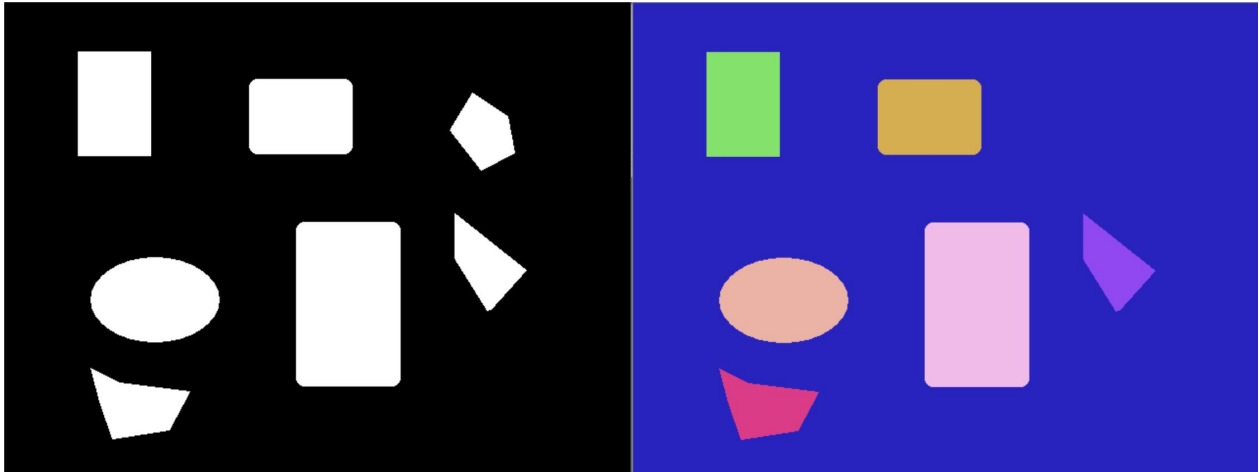


## 计算机视觉 课程实验报告

学号：201800130086	姓名：徐鹏博	
实验题目：图像结构 1		
<p>实验过程中遇到和解决的问题：</p> <p>实验 7.1 连通域</p> <p>对一幅输入二值图像进行如下操作（连通域计算可以调用 <code>cv::connectedComponent</code> 函数）：</p> <ul style="list-style-type: none"> <li>- 对每个连通区域进行随机上色，如下图所示；</li> <li>- 删除较小的前景区域，只保留最大的一个；</li> </ul> <p>这里我找到函数 <code>connectedComponentsWithStats</code> 可以实现连通域的计算，并且可以利用它对连通域面积进行筛选，较小的区域和背景区域同色，其他使用随机颜色。</p> <pre> void exp_1(Mat input,int AreaValue) {     Mat BW, ColorImage, stats, centroids;     threshold(input, BW, 150, 255, THRESH_BINARY);     imshow("原图 1", input);     int labels = connectedComponentsWithStats(BW, ColorImage, stats, centroids);//连通域计算     vector&lt;Vec3b&gt; colors(labels);     for (int label = 0; label &lt; labels; ++label) {         if (stats.at&lt;int&gt;(label, cv::CC_STAT_AREA) &lt; AreaValue)             colors[label] = colors[0];         else             colors[label] = Vec3b((rand() &amp; 255), (rand() &amp; 255), (rand() &amp; 255));     }     Mat output(input.size(), CV_8UC3);     for (int r = 0; r &lt; output.rows; ++r) {         for (int c = 0; c &lt; output.cols; ++c) {             int label = (int) ColorImage.at&lt;int&gt;(r, c);             output.at&lt;Vec3b&gt;(r, c) = colors[label];         }     }     imshow("补色图", output); } </pre> <p>效果: (这里面积阈值设定使得)</p>		

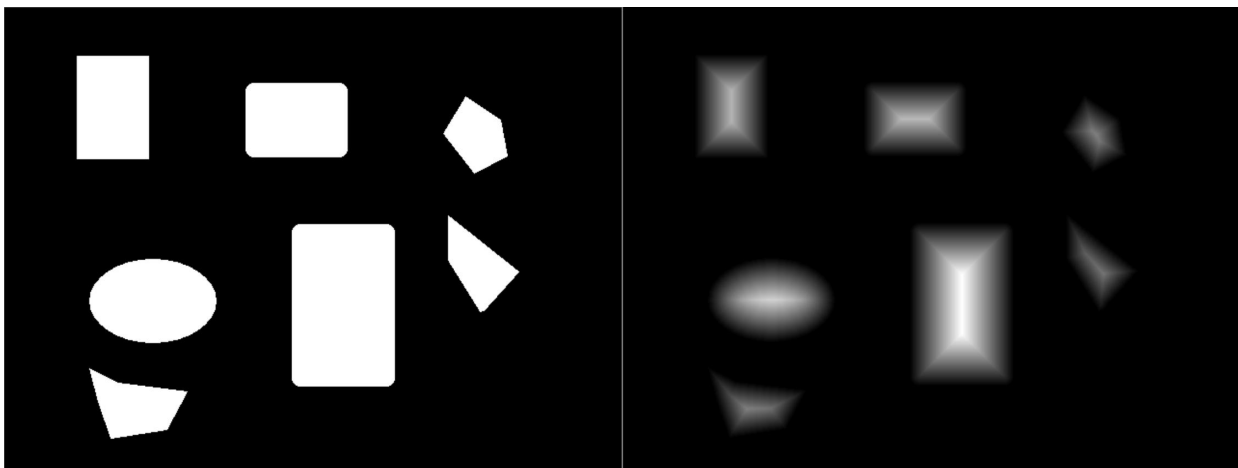


## 实验 7.2 距离变换

了解 OpenCV 的距离变换函数 `distanceTransform`，使用合适的测试图像进行测试，并将距离场可视化输出

```
void exp_2(Mat input) {
    Mat Bin;
    threshold(input, Bin, 60, 255, THRESH_BINARY);
    imshow("原图 2", input);
    Mat output, Labels;
    distanceTransform(Bin, output, Labels, DIST_L2, 3, DIST_LABEL_CCOMP); //距离变换
    normalize(output, output, 0, 1, NORM_MINMAX); //归一化
    imshow("距离转换", output);
}
```

效果：





### 结果分析与体会：

`connectedComponentsWithStats` 函数参数值包括输入的二值图，会输出连通域的 `label` 数组，`Stats` 数组，`centroids` 数组。`stats` 对应每个轮廓的 `x,y,width,height` 和面积，`centroids` 是中心点，`label` 是第几个轮廓。

距离变换是对二值图的一种变换，是计算并标识空间点对目标点的距离的过程，最终将二值图变为灰度图(每个点的灰度值等于它到最近目标点的距离)。