

第 4 章 PCA 原理和应用.....	2
4. 1 PCA 原理介绍.....	2
4. 1. 1 PCA 概念.....	2
4. 1. 2 opencv 中 PCA 相关接口.....	2

第 4 章 PCA 原理和应用

4.1 PCA 原理介绍

4.1.1 PCA 概念

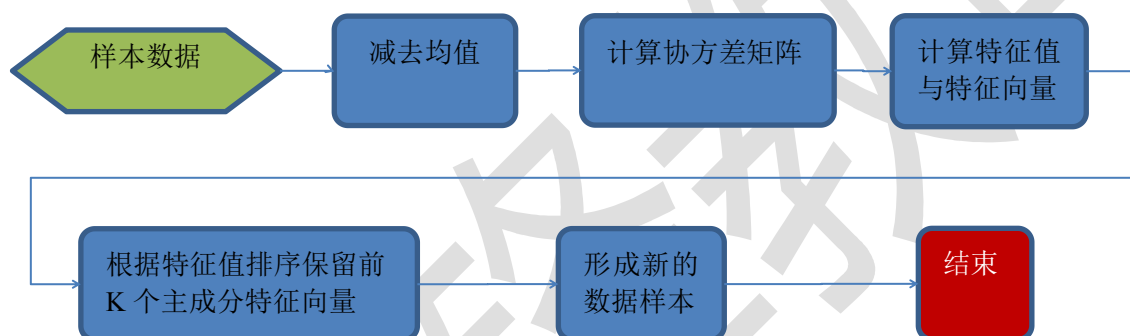
PCA (Principal Component Analysis) 是一种常用的数据分析方法。PCA 通过线性变换将原始数据变换为一组各维度线性无关的表示，可用于提取数据的主要特征分量，常用于高维数据的降维处理。

PCA 特点:

主成分不变

细微损失

高维数据到低维数据



数据互换:

输出数据 = 前 K 个特征向量组合 * 均值调整后的数据

均值调整后的数据 = 前 K 个特征向量行组合^T * 输出数据

原始数据 = 前 K 个特征向量行组合^T * 输出数据 + 均值数据

4.1.2 opencv 中 PCA 相关接口

4.1.2.1 cvtColor()

```
void cvtColor( InputArray src, OutputArray dst, int code, int dstCn = 0 );
```

功能:颜色空间转换

参数:

src: 原图

dst: 保存转换后的图

做真实的自己，用良心做教育

dstCn: 转换方式, 比如 COLOR_BGR2GRAY 转换成灰度图

4.1.2.2 threshold()

```
double threshold( InputArray src, OutputArray dst, double thresh, double maxval, int type );
```

功能: 图像的二值化, 就是将图像上的像素点的灰度值设置为 0 或 255, 也就是将整个图像呈现出明显的只有黑和白的视觉效果

参数:

src: 原图

dst: 保存转换后的图

thresh: 设定的阈值

maxval: 当灰度值大于 (或小于) 阈值时将该灰度值赋成的值

type: 当前二值化的方式

4.1.2.3 findContours()

```
void findContours( InputOutputArray image, OutputArrayOfArrays contours, OutputArray hierarchy, int mode, int method, Point offset = Point());
```

功能: 轮廓检测

参数:

image: 一般是灰度图或者二值化图, 多是二值化图

contours: 定义为 “vector<vector<Point>> contours”, 是一个双重向量, 每一组点集就是一个轮廓, 有多少轮廓, contours 就有多少元素

hierarchy: 定义为 “vector<Vec4i> hierarchy”, hierarchy 是一个向量, 向量内每个元素都是一个包含 4 个 int 型的数组

mode: 定义轮廓的检索模式

CV_RETR_EXTERNAL, 只检测最外围轮廓, 包含在外围轮廓内的内围轮廓被忽略

CV_RETR_LIST, 检测所有的轮廓, 包括内围、外围轮廓, 但是检测到的轮廓不建立等级关系

CV_RETR_CCMP, 检测所有的轮廓, 但所有轮廓只建立两个等级关系

CV_RETR_TREE, 检测所有轮廓, 所有轮廓建立一个等级树结构

method: 定义轮廓的近似方法

CV_CHAIN_APPROX_NONE, 保存物体边界上所有连续的轮廓点到 contours 向量内;

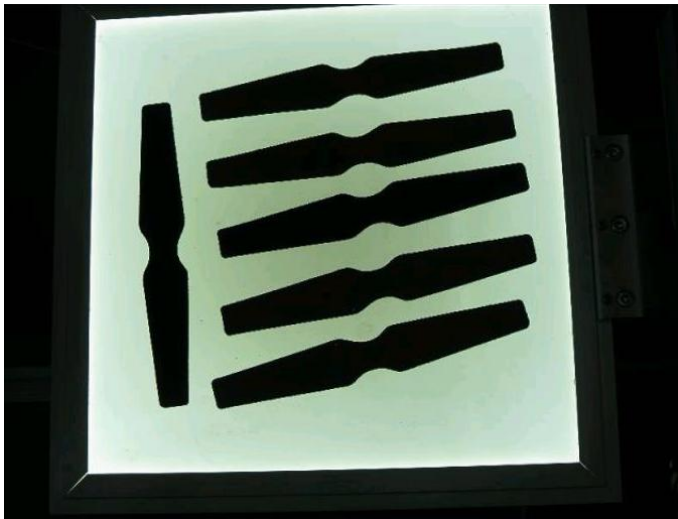
CV_CHAIN_APPROX_SIMPLE, 仅保存轮廓的拐点信息, 把所有轮廓拐点处的点保存入 contours 向量内, 拐点与拐点之间直线段上的信息点不予保留; CV_CHAIN_APPROX_TC89_L1, 使用 teh-Chinl chain 近似算法;

CV_CHAIN_APPROX_TC89_KCOS, 使用 teh-Chinl chain 近似算法。

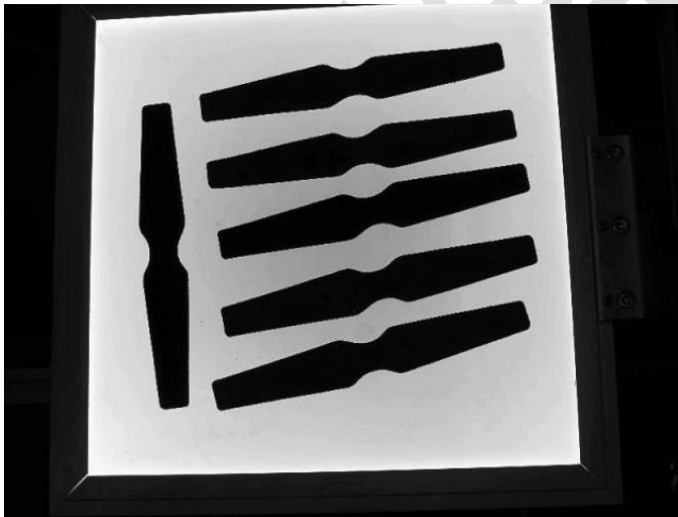
offset: Point 偏移量, 所有的轮廓信息相对于原始图像对应点的偏移量, 相当于在每一个检测出的轮廓点上加上该偏移量, 并且 Point 还可以是负值

案例: [参看基础代码](#)

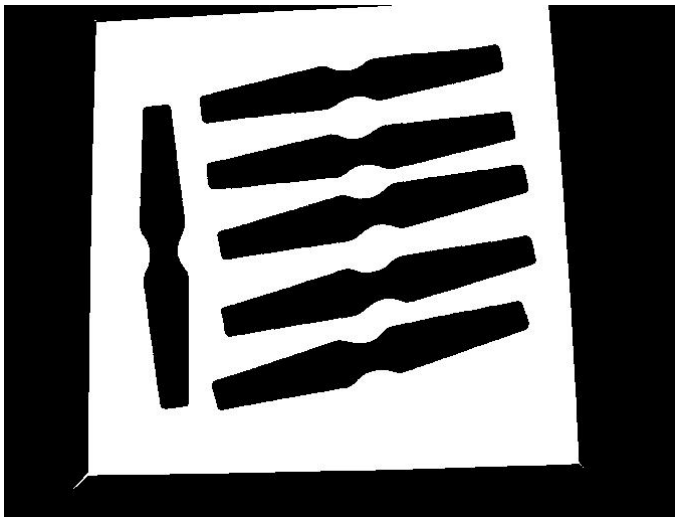
原图:



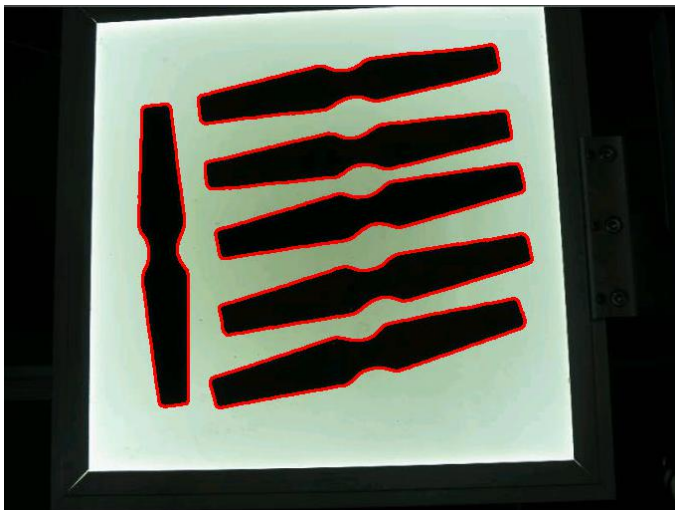
灰度图:



二值化图:



轮廓检测图:



最终图:

