

实战：基于OpenCV实现偏斜文档校正

gloomyfish 小白学视觉 今天

点击上方“小白学视觉”，选择加“星标”或“置顶”

重磅干货，第一时间送达

本文转自 | OpenCV学堂

纸质文档扫描中经常会发生扫描出来的图像有一定角度的偏斜，对后期的文档信息化OCR提取造成很大的干扰，导致OCR识别准确率下降从而影响文档信息化的结果。这个时候可以使用OpenCV对文档进行纠偏，最常见的文本纠偏算法有两种，分别是

- 基于FFT变换以后频率域梯度
- 基于离散点求最小外接轮廓

这两种方法各有千秋，相对来说，第二种方法得到的结果更加准确，第一种基于离散傅立叶变换求振幅的方法有时候各种阈值选择在实际项目中会有很大问题。

基于FFT变换以后频率域梯度

主要思路是先把图像转换为灰度图像，然后使用离散傅立叶变换得到图像在频率域空间的振幅，对其二值化之后，使用霍夫直线检测得到角度，然后根据角度完成旋转校正。代码实现如下：

```
1. Mat src = imread("D:/vcprojects/images/rotate_text.png");
2. Mat gray, binary;
3. cvtColor(src, gray, COLOR_BGR2GRAY);
4.
5. //expand input image to optimal size
6. Mat padded;
7. int m = getOptimalDFTSize(gray.rows);
```

```

8.   int n = getOptimalDFTSize(gray.cols);
9.   // on the border add zero values
0.   copyMakeBorder(gray, padded, 0, m - gray.rows, 0, n - gray.cols, BORDER_CONSTANT, Scalar::all(0));
1.   Mat planes[] = { Mat_<float>(padded), Mat::zeros(padded.size(), CV_32F) };
2.   Mat complexI;
3.   // Add to the expanded another plane with zeros
4.   merge(planes, 2, complexI);
5.   // 离散傅立叶变换
6.   dft(complexI, complexI);
7.
8.   // 实部与虚部得到梯度图像
9.   // planes[0] = Re(DFT(I), planes[1] = Im(DFT(I))
0.   split(complexI, planes);
1.   magnitude(planes[0], planes[1], planes[0]);
2.   Mat magI = planes[0];
3.   magI += Scalar::all(1);
4.   log(magI, magI);
5.   // crop the spectrum, if it has an odd number of rows or columns
6.   magI = magI(Rect(0, 0, magI.cols & -2, magI.rows & -2));
7.   // rearrange the quadrants of Fourier image  so that the origin is at the image center
8.   int cx = magI.cols / 2;
9.   int cy = magI.rows / 2;
0.   Mat q0(magI, Rect(0, 0, cx, cy)); // Top-Left - Create a ROI per quadrant
1.   Mat q1(magI, Rect(cx, 0, cx, cy)); // Top-Right
2.   Mat q2(magI, Rect(0, cy, cx, cy)); // Bottom-Left
3.   Mat q3(magI, Rect(cx, cy, cx, cy)); // Bottom-Right
4.   Mat tmp;
5.   // swap quadrants (Top-Left with Bottom-Right)
6.   q0.copyTo(tmp);
7.   q3.copyTo(q0);
8.   tmp.copyTo(q3);
9.   q1.copyTo(tmp);
0.   q2.copyTo(q1);
1.   tmp.copyTo(q2);
2.
3.   // 归一化与阈值化显示
4.   normalize(magI, magI, 0, 1.0, NORM_MINMAX);
5.   Mat dst;
6.   magI.convertTo(dst, CV_8UC1, 255, 0);
7.   threshold(dst, binary, 160, 255, THRESH_BINARY);
8.
9.   // 霍夫直线
0.   vector<Vec2f> lines;
1.   Mat linImg = Mat::zeros(binary.size(), CV_8UC3);
2.   HoughLines(binary, lines, 1, (float)CV_PI / 180, 30, 0, 0);
3.   int numLines = lines.size();
4.   float degree = 0.0;
5.   for (int l = 0; l < numLines; l++)
6.   {
7.       float rho = lines[l][0], theta = lines[l][1];
8.       float offset = CV_PI / 12.0;
9.       if (abs(theta) > offset && abs(theta) < (CV_PI / 2.0 - offset)) {
0.           printf("theta : %.2f\n", theta);
1.           degree = (theta)*180-90;
2.       }
3.       Point pt1, pt2;
4.       double a = cos(theta), b = sin(theta);
5.       double x0 = a*rho, y0 = b*rho;
6.       pt1.x = cvRound(x0 + 1000 * (-b));
7.       pt1.y = cvRound(y0 + 1000 * (a));

```

```

8.     pt2.x = cvRound(x0 - 1000 * (-b));
9.     pt2.y = cvRound(y0 - 1000 * (a));
0.     line(linImg, pt1, pt2, Scalar(0, 255, 0), 3, 8, 0);
1. }
2. imshow("lines", linImg);
3.
4. // 旋转调整
5. Mat rot_mat = getRotationMatrix2D(Point(binary.cols/2, binary.rows/2), degree, 1);
6. Mat rotated;
7. warpAffine(src, rotated, rot_mat, src.size(), cv::INTER_CUBIC, 0, Scalar(255, 255, 255));
8. imshow("input", src);
9. imshow("deskew-demo", rotated);
0. imwrite("D:/deskew_text.png", rotated);

```

基于离散点求最小外接轮廓

其主要思路是先把图像二值化，得到一系列离散的前景像素点集合，然后利用轮廓的最小外接矩形函数，得到偏斜的矩形大小与角度，通过仿射变换完成校正。代码实现如下：

```


1. Mat src = imread("D:/vcprojects/images/rotate_text.png");
2. Mat gray, binary;
3. cvtColor(src, gray, COLOR_BGR2GRAY);
4. threshold(gray, binary, 0, 255, THRESH_BINARY_INV | THRESH_OTSU);
5. imshow("binary", binary);
6. imwrite("D:/binary_text.png", binary);
7.
8. vector<Point> points;
9. findNonZero(binary, points);
0. RotatedRect box = minAreaRect(points);
1.
2. double angle = box.angle;
3. if (angle < -45.)
4.     angle += 90.;
5. printf("angle : %.2f\n", angle);
6.
7. Point2f vertices[4];
8. box.points(vertices);
9. for (int i = 0; i < 4; ++i)
0.     line(src, vertices[i], vertices[(i + 1) % 4], Scalar(0, 0, 255), 2);
1. imshow("box", src);
2. imwrite("D:/box_text.png", src);
3.
4. Mat rot_mat = getRotationMatrix2D(box.center, angle, 1);
5. Mat rotated;
6. warpAffine(src, rotated, rot_mat, src.size(), cv::INTER_CUBIC, 0, Scalar(255, 255, 255));
7. //bitwise_not(rotated, rotated);
8. imshow("deskew-demo", rotated);

```

运行结果

- 原图

A fork of Tesseract Tools for Android ([tesseract-android-tools](#)) that adds some additional functions. Tesseract Tools for Android is a set of Android APIs and build files for the [Tesseract OCR](#) and [Leptonica](#) image processing libraries

 OpenCV学堂

- 最小外接矩形

A fork of Tesseract Tools for Android ([tesseract-android-tools](#)) that adds some additional functions. Tesseract Tools for Android is a set of Android APIs and build files for the [Tesseract OCR](#) and [Leptonica](#) image processing libraries

OpenCV学堂

- 校正之后

A fork of Tesseract Tools for Android ([tesseract-android-tools](#)) that adds some additional functions. Tesseract Tools for Android is a set of Android APIs and build files for the [Tesseract OCR](#) and [Leptonica](#) image processing libraries



下载1：OpenCV-Contrib扩展模块中文版教程

在「小白学视觉」公众号后台回复：**扩展模块中文教程**，即可下载全网第一份OpenCV扩展模块教程中文版，涵盖**扩展模块安装**、**SFM算法**、**立体视觉**、**目标跟踪**、**生物视觉**、**超分辨率处理**等二十多章内容。

下载2：Python视觉实战项目52讲

在「小白学视觉」公众号后台回复：**Python视觉实战项目**，即可下载包括**图像分割**、**口罩检测**、**车道线检测**、**车辆计数**、**添加眼线**、**车牌识别**、**字符识别**、**情绪检测**、**文本内容提取**、**面部识别**等31个视觉实战项目，助力快速学校计算机视觉。

下载3：OpenCV实战项目20讲

在「小白学视觉」公众号后台回复：**OpenCV实战项目20讲**，即可下载含有**20个基于OpenCV实现20个实战项目**，实现OpenCV学习进阶。

交流群

欢迎加入公众号读者群一起和同行交流，目前有**SLAM**、**三维视觉**、**传感器**、**自动驾驶**、**计算摄影**、**检测**、**分割**、**识别**、**医学影像**、**GAN**、**算法竞赛**等微信群（以后会逐渐细分），请扫描下面微信号加群，备注：“昵称+学校/公司+研究方向”，例如：“张三 + 上海交大 + 视觉SLAM”。**请按照格式备注，否则不予通过**。添加成功后会根据研究方向邀请进入相关微信群。**请勿**在群内发送**广告**，否则会请出群，谢谢理解~



A promotional banner for '小白学视觉' (Xiao Bai Xue Shi Jue). The background is black with a white piano keyboard on the right. On the left, there is a blue circular logo with a white robot icon. The text '小白学视觉' is in large white characters. Below it, '计算机视觉' (Computer Vision) is written in white. Further down, '论文解读 求职感想' (Paper interpretation, job-seeking thoughts) and 'SLAM技术 深度学习 学习感受' (SLAM technology, deep learning, learning experience) are listed in white. A dashed line separates this from the bottom text: '距离我们只差一个' (Only one step away from us) in white, and '长按关注' (Long press to follow) in red. On the right side, the text '聚集地 计算机视觉学者' (Gathering place for computer vision scholars) is written vertically in white. At the bottom right, there is a QR code with a blue robot icon in the center, framed by a blue border.

喜欢此内容的人还喜欢

欢迎阅读 OpenCV 指南—第2部分

深度学习与计算机视觉



使用 Python 创建自己的文档扫描仪

深度学习与计算机视觉



如何使用OpenCV为图像加水印

深度学习与计算机视觉

