

# 使用OpenCV實現偏斜文檔校正

gloomyfish 小白學視覺 2022-08-17 10:05 發表於香港

點擊上方“[小白學視覺](#)”，選擇加“[星標](#)”或“[置頂](#)”

重磅乾貨，第一時間送達

## 使用OpenCV實現偏斜文檔校正

紙質文檔掃描中經常會發生掃描出來的圖像有一定角度的偏斜，對後期的文檔信息化OCR提取造成很大的干擾，導致OCR識別準確率下降從而影響文檔信息化的結果。這個時候可以使用OpenCV對文檔進行糾偏，最常見的文本糾偏算法有兩種，分別是

- 基於FFT變換以後頻率域梯度
- 基於離散點求最小外接輪廓

這兩種方法各有千秋，相對來說，第二種方法得到的結果更加準確，第一種基於離散傅立葉變換求振幅的方法有時候各種閾值選擇在實際項目中會有很大問題。

### 基於FFT變換以後頻率域梯度

主要思路是先把圖像轉換為灰度圖像，然後使用離散傅立葉變換得到圖像在頻率域空間的振幅，對其二值化之後，使用霍夫直線檢測得到角度，然後根據角度完成旋轉校正。代碼實現如下：

```
. Mat src = imread("D:/vcprojects/images/rotate_text.png");
. Mat gray, binary;
. cvtColor(src, gray, COLOR_BGR2GRAY);
.
. //expand input image to optimal size
. Mat padded;
. int m = getOptimalDFTSize(gray.rows);
. int n = getOptimalDFTSize(gray.cols);
. // on the border add zero values
. copyMakeBorder(gray, padded, 0, m - gray.rows, 0, n - gray.cols, BORDER_CONSTANT, Scalar::all(0));
. Mat planes[] = { Mat_<float>(padded), Mat::zeros(padded.size(), CV_32F) };
. Mat complexI;
. // Add to the expanded another plane with zeros
. merge(planes, 2, complexI);
. // 离散傅立叶变换
. dft(complexI, complexI);
.
. // 实部与虚部得到梯度图像
. // planes[0] = Re(DFT(I)), planes[1] = Im(DFT(I))
. split(complexI, planes);
. magnitude(planes[0], planes[1], planes[0]);
. Mat magI = planes[0];
. magI += Scalar::all(1);
. log(magI, magI);
. // crop the spectrum, if it has an odd number of rows or columns
. magI = magI(Rect(0, 0, magI.cols & -2, magI.rows & -2));
. // rearrange the quadrants of Fourier image  so that the origin is at the image center
```

```

.   int cx = magI.cols / 2;
.   int cy = magI.rows / 2;
.   Mat q0(magI, Rect(0, 0, cx, cy)); // Top-Left - Create a ROI per quadrant
.   Mat q1(magI, Rect(cx, 0, cx, cy)); // Top-Right
.   Mat q2(magI, Rect(0, cy, cx, cy)); // Bottom-Left
.   Mat q3(magI, Rect(cx, cy, cx, cy)); // Bottom-Right
.   Mat tmp;
.   // swap quadrants (Top-Left with Bottom-Right)
.   q0.copyTo(tmp);
.   q3.copyTo(q0);
.   tmp.copyTo(q3);
.   q1.copyTo(tmp);
.   q2.copyTo(q1);
.   tmp.copyTo(q2);
.
.   // 归一化与阈值化显示
.   normalize(magI, magI, 0, 1.0, NORM_MINMAX);
.   Mat dst;
.   magI.convertTo(dst, CV_8UC1, 255, 0);
.   threshold(dst, binary, 160, 255, THRESH_BINARY);
.
.   // 霍夫直线
.   vector<Vec2f> lines;
.   Mat linImg = Mat::zeros(binary.size(), CV_8UC3);
.   HoughLines(binary, lines, 1, (float)CV_PI / 180, 30, 0, 0);
.   int numLines = lines.size();
.   float degree = 0.0;
.   for (int l = 0; l<numLines; l++)
.   {
.       float rho = lines[l][0], theta = lines[l][1];
.       float offset = CV_PI / 12.0;
.       if (abs(theta) > offset && abs(theta) < (CV_PI / 2.0 - offset)) {
.           printf("theta : %.2f\n", theta);
.           degree = (theta)*180-90;
.       }
.       Point pt1, pt2;
.
.       double a = cos(theta), b = sin(theta);
.       double x0 = a*rho, y0 = b*rho;
.       pt1.x = cvRound(x0 + 1000 * (-b));
.       pt1.y = cvRound(y0 + 1000 * (a));
.       pt2.x = cvRound(x0 - 1000 * (-b));
.       pt2.y = cvRound(y0 - 1000 * (a));
.       line(linImg, pt1, pt2, Scalar(0, 255, 0), 3, 8, 0);
.   }
.   imshow("lines", linImg);
.
.   // 旋转调整
.   Mat rot_mat = getRotationMatrix2D(Point(binary.cols/2, binary.rows/2), degree, 1);
.   Mat rotated;
.   warpAffine(src, rotated, rot_mat, src.size(), cv::INTER_CUBIC, 0, Scalar(255, 255, 255));
.   imshow("input", src);
.   imshow("deskew-demo", rotated);
.   imwrite("D:/deskew_text.png", rotated);

```

## 基於離散點求最小外接輪廓

其主要思路是先把圖像二值化，得到一系列離散的前景像素點集合，然後利用輪廓的最小外接矩形函數，得到偏斜的矩形大小與角度，通過仿射變換完成校正。代碼實現如下：

```

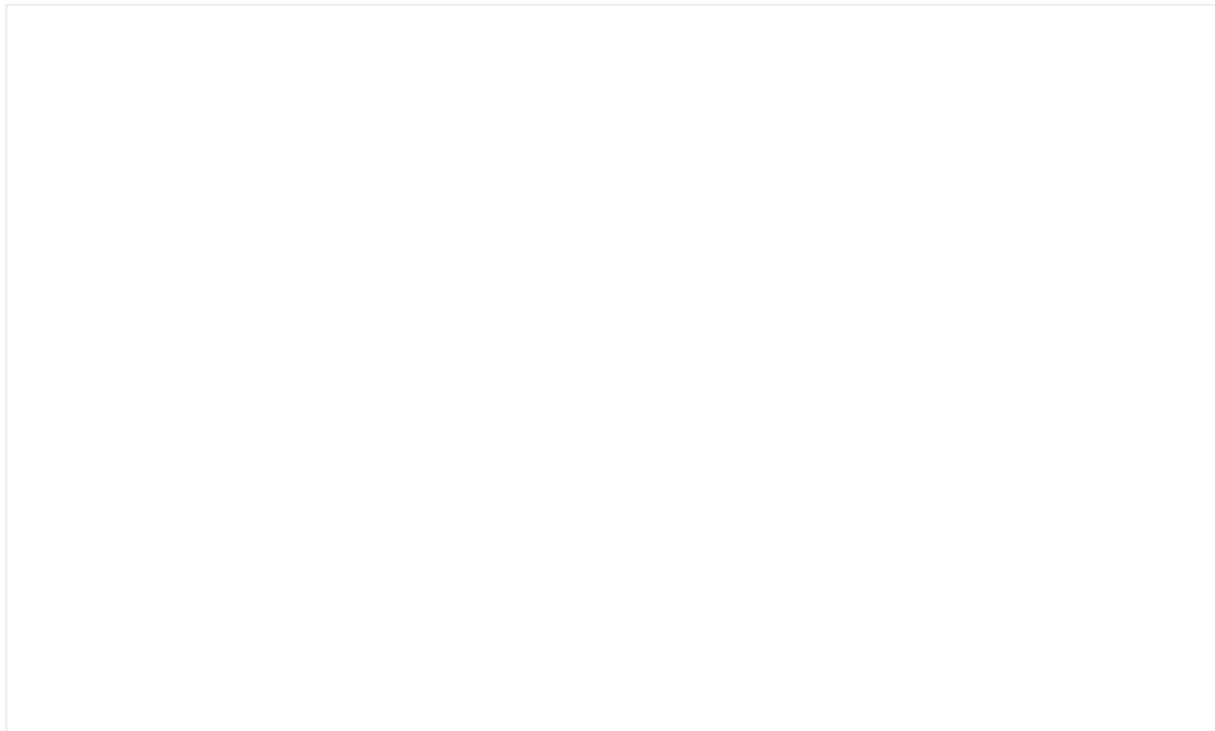
.   Mat src = imread("D:/vcprojects/images/rotate_text.png");
.   Mat gray, binary;
.   cvtColor(src, gray, COLOR_BGR2GRAY);
.   threshold(gray, binary, 0, 255, THRESH_BINARY_INV | THRESH_OTSU);
.   imshow("binary", binary);
.   imwrite("D:/binary_text.png", binary);
.
.   vector<Point> points;
.   findNonZero(binary, points);
.   RotatedRect box = minAreaRect(points);
.
.   double angle = box.angle;
.   if (angle < -45.)
.       angle += 90.;
.   printf("angle : %.2f\n", angle);
.
.   Point2f vertices[4];
.   box.points(vertices);

```

```
.    for (int i = 0; i < 4; ++i)
.        line(src, vertices[i], vertices[(i + 1) % 4], Scalar(0, 0, 255), 2);
.    imshow("box", src);
.    imwrite("D:/box_text.png", src);
.
.    Mat rot_mat = getRotationMatrix2D(box.center, angle, 1);
.    Mat rotated;
.    warpAffine(src, rotated, rot_mat, src.size(), cv::INTER_CUBIC, 0, Scalar(255, 255, 255));
.    //bitwise_not(rotated, rotated);
.    imshow("deskew-demo", rotated);
```

## 運行結果

- 原圖



- 最小外接矩形



• 校正之後



好消息！

小白學視覺知識星球

開始面向外開放啦👉👉👉



### 下載1：OpenCV-Contrib擴展模塊中文版教程

在「小白學視覺」公眾號后台回復：**擴展模塊中文教程**，即可下載全网第一份OpenCV擴展模塊教程中文版，涵蓋擴展模塊安裝、SFM算法、立體視覺、目標跟蹤、生物視覺、超分辨率處理等二十多章內容。

### 下載2：Python視覺實戰項目52講

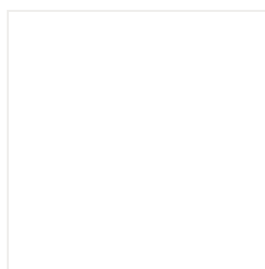
在「小白學視覺」公眾號后台回復：**Python視覺實戰項目**，即可下載包括圖像分割、口罩檢測、車道線檢測、車輛計數、添加眼線、車牌識別、字符識別、情緒檢測、文本內容提取、面部識別等31個視覺實戰項目，助力快速學校計算機視覺。

### 下載3：OpenCV實戰項目20講

在「小白學視覺」公眾號后台回復：**OpenCV實戰項目20講**，即可下載含有20個基於OpenCV實現20個實戰項目，實現OpenCV學習进阶。

### 交流群

歡迎加入公眾號讀者群一起和同行交流，目前有SLAM、三維視覺、傳感器、自動駕駛、計算攝影、檢測、分割、識別、醫學影像、GAN、算法競賽等微信群（以後會逐漸細分），請掃描下面微信號加群，備註：“暱稱+學校/公司+研究方向”，例如：“張三 + 上海交大 + 視覺SLAM”。請按照格式備註，否則不予通過。添加成功後會根據研究方向邀請進入相關微信群。請勿在群內發送廣告，否則會請出群，謝謝理解~





喜歡此內容的人還喜歡

全面講解十大經典排序算法（Python實現）  
小白學視覺



如何使用OpenCV測量圖像中物體之間的距離  
小白學視覺



全連接神經網絡的原理及Python實現  
小白學視覺

