Assignment 1 - Page Detection

Document Analysis

Timon Höbert(01427936)

Manuel Mayerhofer (01328948)

Stefan Stappen(0132)

May 5, 2018

1 Definition of Task

Page detection is the task of detecting and segmenting the page outline of a document within an image. The ICDAR2015Competition on Smartphone Document Capture and OCR (SmartDoc) [1] is the first competition for document page detection and Smartphone OCR. The first assignment is the page detection.

The input is a set of video clips containing a document at different backgrounds. The output contains the bounding box of the document in quadrilateral coordinates per each frame of the video clip.

2 Dataset

The dataset consists of six different document types: Datasheet, letter, magazine, paper, patent, and tax. For each document type, there are five different documents, which cover different document layout schemes and contents. The six document types are recorded on five different backgrounds. A result is a number of 30 videos per background and a total of 150 videos. The videos are recorded using FULL HD resolution at variable frame rate. Every video has a duration of around 10 seconds. The videos include distortions like focus and motion blur, the perspective of the document, change of illumination and partial occlusions of the document pages. Each frame is ground-truthed by annotating the quadrilateral coordinates of each frame in an XML file.

3 Implementation

Our method is based on the method of P Li, Y Niw and X. Li from NetEase.

3.1 Preprocessing

First, the image is converted in a gray value image. Then we use the canny edge detector with a sigma of 3 to create a binary image of the document. We chose the canny detector because it found most of the edges in the document frames, unlike Sobel or Prewitt. However, there were many distortions on the edges, so we increased the sigma from $\sqrt{2}$ to 4 to smooth the image a bit. The result of the preprocessing steps looks like Figure 1.
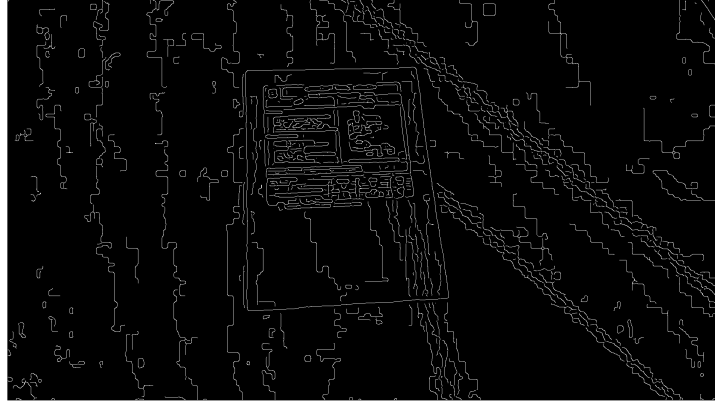

Figure 1: Document frame of the image after the preprocessing step.

3.2 Line Segment Detector

After the preprocessing we use the line segment detector (LSD) [4] to detect the lines in the image. We use the LSD-OpenCV-MATLAB [3]. The LSD algorithm detects the essential lines in the image, but because we use the canny edge detector as a preprocessing step. There are many disturbances in the image, and the LSD algorithm detects lines that are no line in the real image. However, we need the canny detector because in frames, where the document and the background have similar color the LSD algorithm detects no edges of the document.

The founded set of lines is reduced as follows. Lines are removed which have a minimum length of the image size. Furthermore, lines are removed which are near the border of the image or which have their starting point on the border of the image.
The result of the reduced set of lines with the LSD algorithm is shown in Figure 2

3.3 Bounding Box Calculation

The line segments are grouped in horizontal and vertical segments. After that, we merge lines that have the same direction with a certain margin between the line segments. Then we select two horizontal and two vertical lines and calculate the bounding boxes of the document. When selecting horizontal and vertical lines, only those with at least 3 of 4 lines have a similar start, or end point are selected.
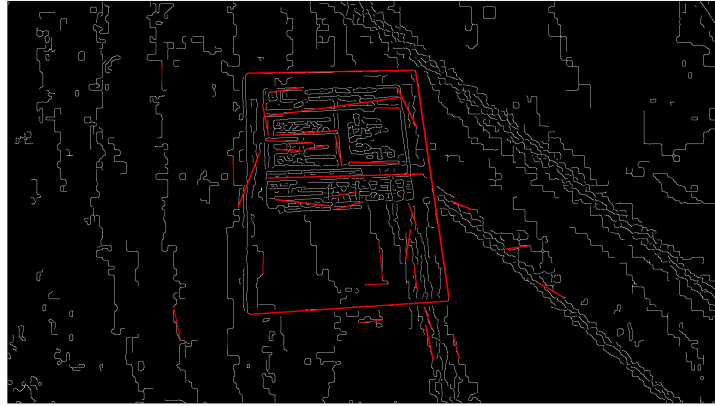
Figure 2: Document frame of the image after the preprocessing step.

3.4 Best Bounding Box Selection

For every bounding box we calculate the following properties:

- Area
  - The area is the most important property because we choose the bounding box with the greatest area, but we also check if the area exceeds a limit because sometimes the table is also recognized and this would form the greatest area.

- Length of the horizontal and vertical lines
  - We check if the two horizontal lines are equal in length with a little threshold, for vertical lines respectively.

- Aspect ratio
  - We assume that we recognize documents in the A4 format in portrait format or horizontal format.

- Inner angles
  - We check if the inner equals are similar to 90 degrees, but because of the perspective distortion, we accept a high deviation of the angles to achieve suitable results.

If the bounding box complete the property checks, we choose the bounding box with the greatest area.

3.5 Frame Interpolation

If we get no bounding box from the algorithm or the bounding box is smaller than the average bounding boxes in the complete video we interpolate between frames with areas that have an area greater than the average area of the bounding boxes.

3

4 Performance

One big drawback of the algorithm is the execution time that depends on the number of founded line segments. We calculate for every combination of two horizontal and two vertical lines a bounding box. This is huge expense even with a small number of lines due to the permutations. Another problem is that the LSD algorithm takes only an input image path as input. So we have to create an image for every frame and write it out.

5 Evaluation

To measure the performance the Jacard index measure [2] is used. The Jacard index is the intersection of the detected bounding box with the ground truth document bounding box divided through the union of the bounding boxes (see equation 1).

$$JI(frame) = \frac{area(G \cap S))}{area(G \cup S))} \tag{1}$$

The score for one video is the average of the frame scores, and the overall accuracy of the implemented method is average of all videos with different backgrounds.

6 Results

References

[1] Jean-Christophe Burie, Joseph Chazalon, Mickaël Coustaty, Sébastien Eskenazi, Muhammad Muzzamil Luqman, Maroua Mehri, Nibal Nayef, Jean-Marc Ogier, Sophea Prum, and Marçal Rusiñol. Icdar2015 competition on smartphone document capture and ocr (smartdoc). In *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pages 1161–1165. IEEE, 2015.

[2] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[3] primetang. Lsd-opencv-matlab. https://github.com/primetang/LSD-OpenCV-MATLAB, commit: 0b035a8c5b105091604d71f1570ac4d4526aec8d, 2014.

[4] Rafael Grompone Von Gioi, Jeremie Jakubowicz, Jean-Michel Morel, and Gregory Randall. Lsd: A fast line segment detector with a false detection control. *IEEE transactions on pattern analysis and machine intelligence*, 32(4):722–732, 2010.