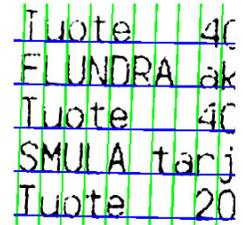


Image distortion estimation and compensation

Description	Robustly undistorting receipt images for easier OCR analysis.
Period	Summer 2014
Languages & Libs	Matlab
Tags	Computer Vision



This project's goal was to automatically and robustly estimate and compensate distortion from any receipt photos. The user is able to just snap the photo and OCR could accurately identify bought products and their prices. However this task is somewhat challenging because typically receipts tend to get crumpled and bent. Thus they won't lie nicely flat on a surface for easy analysis. This set of algorithms solves that problem and produces distortion-free thresholded images for the next OCR step.

The first step is to convert the image to black & white and to adaptively threshold it to separate black letters from white-to-gray background. Typically the light comes from top so the used camera, phone or a tabled tends to cast a shadow on it. Hopefully its border will be at least slightly "smooth", so that it won't get mis-thresholded as actual ink on the receipt. I used this formula to compensate light and shadow effects from the image:

$$I'_{x,y} = \text{Clamp}_{[0,1]} \left(a + b \cdot \log \frac{I_{x,y}}{\text{GaussBlur}_{\sigma,x,y}(I)} \right) \quad (1)$$

Basically each (x,y) location's brightness is compared to it surrounding brightness, which is determined by applying the standard Gaussian blur at chosen σ . Parameters a and b adjust the "blackness" range and the result is clamped to have values between 0.0 and 1.0.

The resulting black pixels are separated to clusters based on their connectivity with neighbouring pixels. Clusters with 60 to 500 pixels are assumed to be individual characters. Each valid cluster's major and minor axes are determined by running the standard PCA algorithm and checking its major and minor singular values. Clusters with 75% of variance on the major direction and having a major singular value of at least 50 are accepted for orientation voting. These thresholds filter out spurious votes from too round or too small clusters. The outcome can be seen in Figure 1.

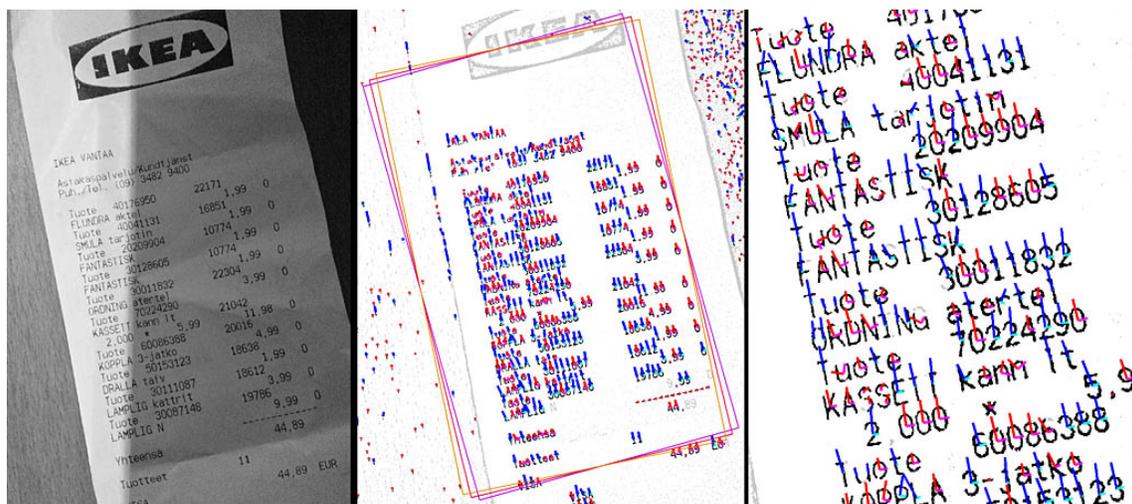


Figure 1: The initial rotation estimation is based on assumptions on the font's height to width ratio. Dark blue major axes are used to find the median rotation.

After the main rotation has been compensated, the next step is to estimate distortions caused by the perspective and/or bent receipt. It is accomplished by generating the [Delaunay triangulation](#) of observed characters and identifying strictly horizontal and vertical segments. An example result can be seen in Figure 2. These lines are used to robustly fit the following linear models for horizontal direction α , vertical direction β and the scale (S) at the location (x, y) :

$$\alpha(x, y) = a_{1,x}x + b_{1,y}y + c_1 \tag{2}$$

$$\beta(x, y) = a_{2,x}x + b_{2,y}y + c_2 \tag{3}$$

$$S(x, y) = \exp(a_{3,x}x + b_{3,y}y + c_3) \tag{4}$$

Since the main rotation has been compensated already, c_1 and c_2 should be approximately zero. The distortion is compensated by following the dynamic model for horizontal (p_h) and vertical (p_v) movement:

$$\frac{\partial}{\partial x} p_h(x, y) = \cos(\alpha(x, y)) \cdot S(x, y) \tag{5}$$

$$\frac{\partial}{\partial y} p_h(x, y) = \sin(\alpha(x, y)) \cdot S(x, y) \tag{6}$$

and

$$\frac{\partial}{\partial x} p_v(x, y) = -\sin(\beta(x, y)) \cdot S(x, y) \tag{7}$$

$$\frac{\partial}{\partial y} p_v(x, y) = \cos(\beta(x, y)) \cdot S(x, y) \tag{8}$$

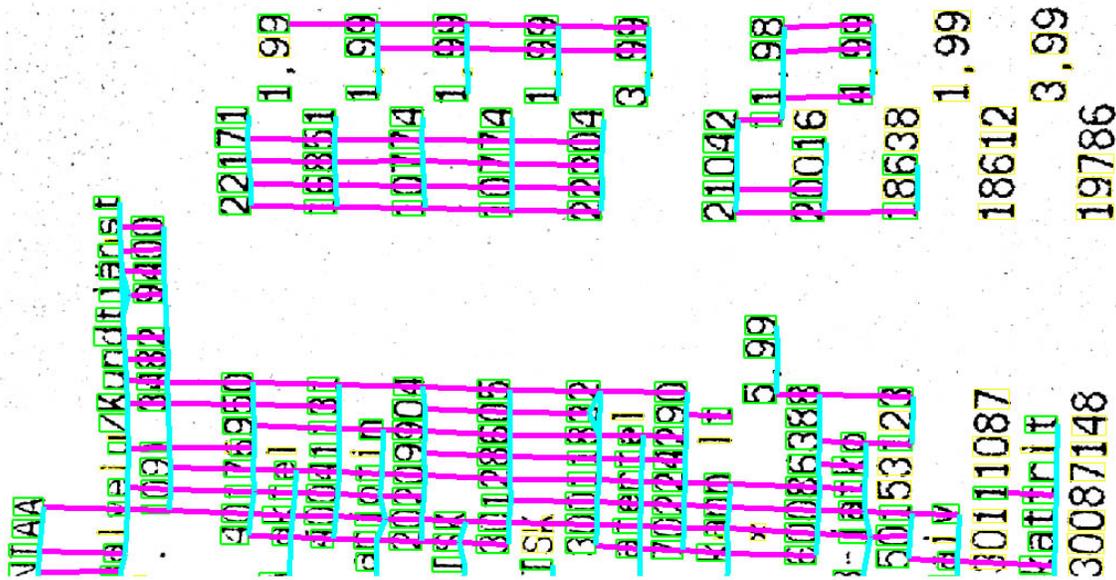


Figure 2: Identified horizontal (magenta) and vertical (cyan) lines (a partial crop of the whole image).

These can be used to construct an inverse mapping which straightens out remaining curvatures and scale changes. Its outcome is visualized at Figure 3, where it is apparent that vertical and horizontal lines align very well with the underlying monospace font.

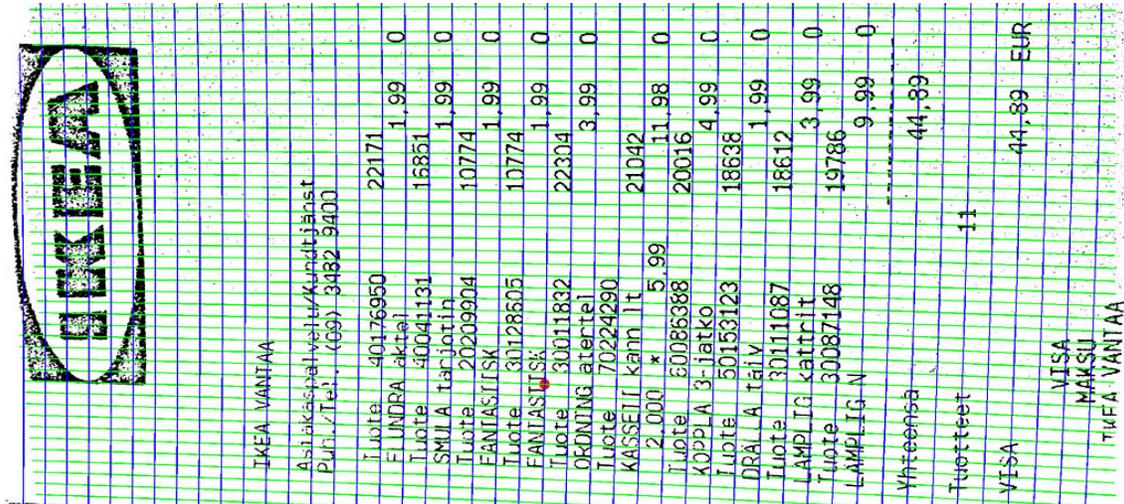


Figure 3: Estimated underlying monospace font's grid is shown in green and blue lines.

The final step is to render the undistorted image and to do final receipt and text line detection. Receipt is horizontally cropped if significant black border separates it from the left and right parts of the image. Because distortions have been corrected already these lines should be fairly accurately purely vertical. Then text lines are detected and separated from others based on white horizontal rows between them. This output should be fairly easy for OCR algorithms to analyze.

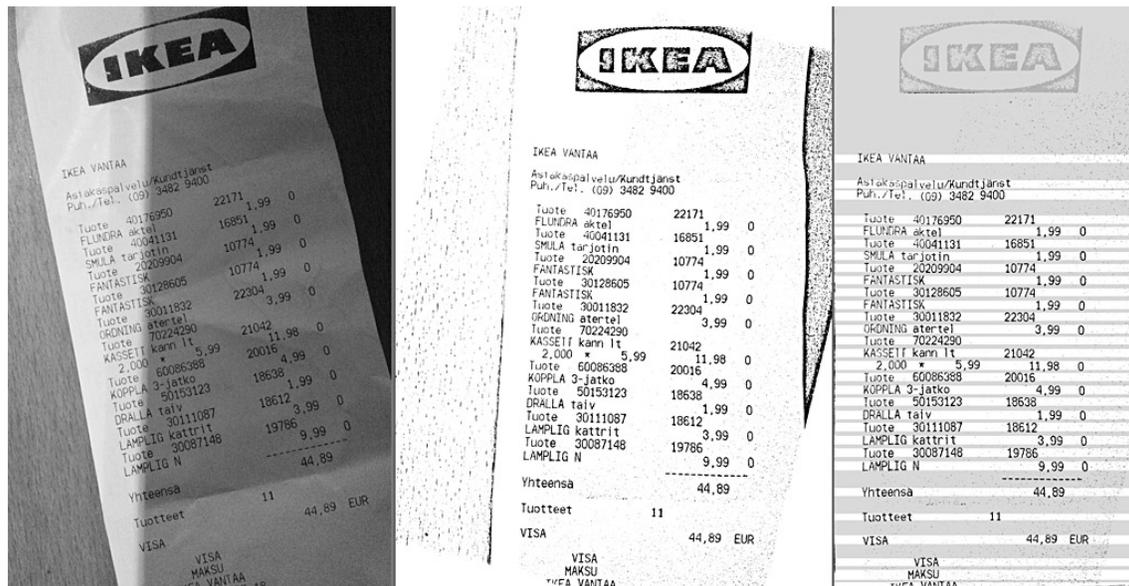


Figure 4: The processing starts from the image on the left, it is thresholded and rotated (shown in middle) and finally remaining distortions are corrected, the receipt is cropped and individual lines are detected.