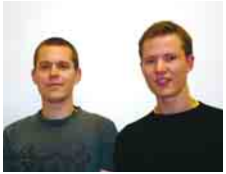

Martin Lüssi
Marco Eichmann

Multilevel Image Thresholding

Graduate candidate	Martin Lüssi, Marco Eichmann
Examiner	Prof. Guido Schuster
Co-examiner	Gabriel Sidler, Eivycom GmbH, Zürich
Subject area	Digitale Medien
Project partner	Northwestern University, USA

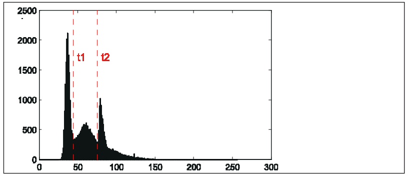

Figure 1: Histogram from PCB image with two thresholds. Example for multilevel-thresholding; t1 = 44, t2 = 75.

Thresholding is one of the most widely used image segmentation operations; one application is foreground background separation. The thresholding operation separates the pixels of an image into different classes. The class to which a pixel is assigned depends on the intensity of the pixel. A special case of thresholding is multilevel thresholding, which means that the pixels are classified into more than two classes. In order to find the thresholds, the histogram of the image is analyzed.

One well known thresholding method is the one proposed by N. Otsu. The Otsu method finds the thresholds by minimizing the sum of the weighted variances of the classes. A simple algorithm finds the thresholds by performing an exhaustive search, which means calculating the objective function for every possible combination of thresholds. The problem with this approach is that when the histogram contains L grey levels and is segmented into M classes, the time needed by the algorithm grows more than exponentially in M. More sophisticated algorithms use dynamic programming and have a time complexity of O(ML^2).

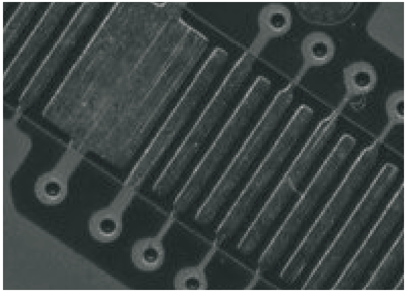

Figure 2: Original PCB image.

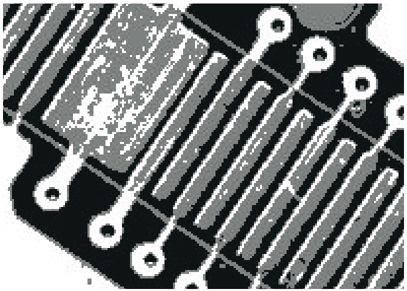

Figure 3: Segmented PCB image in three classes with thresholds from Figure 1.

In our thesis, we show that Otsu thresholding is the same as optimal scalar quantization (in the minimal mean-square error sense), therefore fast algorithms for optimal scalar quantization can be employed for Otsu thresholding. These algorithms exploit a special property of the objective function and use a combination of dynamic programming and divide-and-conquer or SMAWK matrix searching; they have time complexities of O(ML log L) and O(ML) respectively. By implementing the different algorithms and comparing their execution times, we can establish that the O(ML log L) algorithm outperforms both the O(ML^2) and the O(ML) algorithms for reasonably sized histograms.

Another thresholding criterion was proposed by Kittler and Illingworth. The underlying model assumes that the histogram is a mixture of Gaussian distributions with distinct means and variances. The known dynamic programming algorithm for Kittler and Illingworth thresholding has a time complexity of O(ML^2). We propose an algorithm which does not always find the optimal solution but is much faster with linear time complexity.

How many classes an image should be segmented into is another interesting question. We use modified dynamic programming algorithms for Otsu and Kittler-Illingworth thresholding, which automatically determine a good number of classes and calculate the corresponding thresholds.