An image enhancement method for scanned textual and line-art data

Volker Schatz http://volkerschatz.com/science/nonpatents/ November 2015

Keywords: document scanning, image enhancement, OCR

Introduction

As low-cost document scanners are available, digital document archival becomes increasingly feasible and attractive. However, Optical Character Recognition (OCR) is not yet 100% reliable. Therefore it is desirable to store scanned page images along with the text resulting from OCR, so as to allow re-processing in the future with improved OCR algorithms. Ideally, the images should compress well but still retain all information that may be helpful for OCR. For example, storing an unmodified scanned page would waste space on storing variations in background brightness, whereas performing a thresholding before storage would cut image size but discard grey levels that contribute to character outlines.

This disclosure describes an image enhancement method that improves contrast while reducing image file size and does not impact OCR significantly. It is based on analysing the histogram of the image and identifying the peaks of the foreground and background colour. It is therefore applicable to text content and line art, and with minor modifications to grey images embedded in a document (see below).

Description of the method

The image-enhancement method operates on the images that result from grey-scale scan of a document. Each page of the document is processed separately.

The first step is to analyse a histogram of the image. For a textual or line-art document page, the histogram will look as shown in Figure 1. The white background forms the highest peak, and the black ink a lower peak. The peaks are detected by determining which histogram bins exceed a given threshold, represented by the horizontal dashed lines in Figure 1. The threshold is lowered successively until two intervals of histogram bins exceed it, i.e. the threshold intersects the two tallest histogram peaks.

The second step consists of a global linear transformation of pixel values. The interval of values between the peaks is mapped to complete pixel value range. The information contained in the intervals between the zero pixel value and the black peak, and the background peak and the maximum pixel value, respectively, is discarded, thereby reducing the information content and improving the maximum compression rate. The grey values between the dominant foreground and background colours are retained, so intermediate grey levels can be used by later OCR.

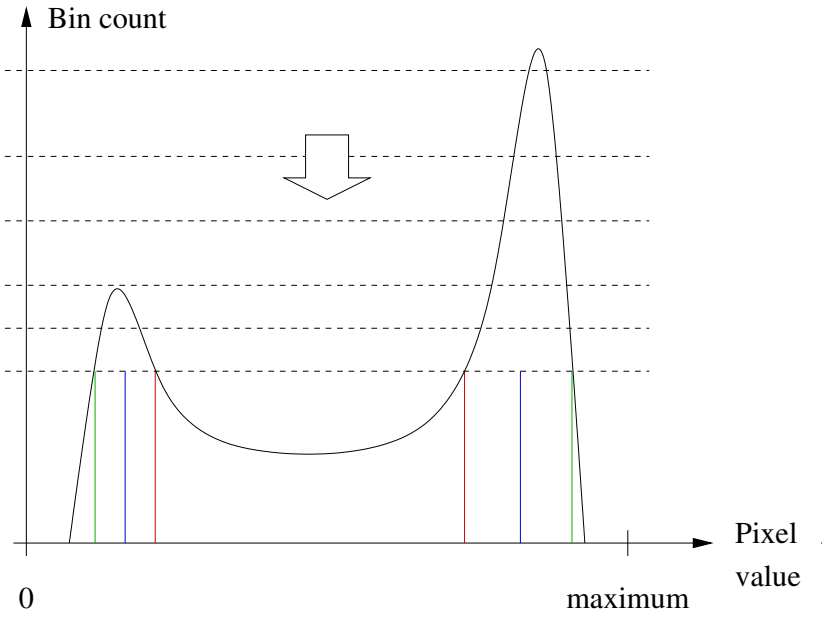

Figure 1: Example histogram of a scan of a text document or line art. Dark pixel values lie at the left end of the scale, bright pixels to the right. The right, higher peak represents the paper background, the smaller peak the foreground colour. In typical scanned documents, both have a distance from the maximum and minimum pixel value, respectively. The method detects the peaks and maps the interval between them to the complete value range, see text.

Details and variants

Both steps of the method can be performed in multiple ways that differ in detail.

- The threshold for peak detection can be initialised to the maximum histogram value (which can be determined while generating the histogram) or any value larger than the maximum of the secondary peak, if such a value can be determined.
- The lowering of the threshold can be done linearly, i.e. by repeated subtraction of a constant; logarithmically, i.e. by repeated multiplication with a constant less than 1; or by a different monotonically decreasing scheme.
- The position at which the threshold is stopped can vary. It can be the highest level at which the histogram exceeds the threshold in two separate intervals, the lowest such level (just above the bottom of the trough between the black and white peak, or a possible third peak), or at a defined level in between, such as the arithmetic or geometric mean.
- The range of pixel values to be mapped to the whole output data range is determined by the final threshold position. It can be the interval between the outer intersection of the threshold with the histogram (green vertical lines in Figure 1), between the inner intersection (red vertical lines), the position of the peak maximum, or a function of those three positions (such as a weighted average; blue vertical lines).

- In order to make the method suitable for greyscale images, the lower bound of the range of pixel values to retain can be set to zero or to the very left end of the populated part of the histogram. This avoids squashing the dark peak and thereby the loss of relevant information when a brighter pixel value is more frequent than that of black ink.

Pseudocode listing

The following listing shows pseudocode for one variant of the method, with logarithmical reduction of the threshold and using the midpoints of the top of the peaks to determine the new pixel value range.

```
Compute a histogram of the image and determine maximum bin value

Initialise threshold to maximum bin value
Initialise peak count to 1
While peak count < 2 and threshold > minimum threshold > 0
    Multiply threshold with reduction factor (< 1)
    Set peak count to 0
    Initialise peak flag to false
    For all histogram bins
        If peak flag is false and bin value > threshold
            Increment peak count
            Save bin index as left side of this peak
            Set peak flag to true
        If peak flag is true and bin value <= threshold
            Save bin index - 1 as right side of this peak
            Set peak flag to false
    If peak flag is true
        Save maximum bin index as right side of last peak

If peak count != 2
    Abort with error

Set left value to (left side + right side of first peak) / 2
Set right value to (left side + right side of second peak) / 2

For all pixels in the image
    If pixel value < left value
        Set pixel value to 0
    Else if pixel value > right value
        Set pixel value to maximum possible pixel value
    Else
        Set pixel value to (pixel value - left value)
            * maximum possible pixel value / (right value - left value)
```