Having an Edge

Image Processing Basics: Edge Detection

Extracting quantitative features of objects from a scene is one of the major tasks in industrial image processing: the diameter and the position of a hole are typical examples. Best practice yields images with good grey-level contrast between objects and background, and the objects clearly show up due to their prominent edges. This article describes some simple methods for edge detection.

Contours in Binary Images

When images are grabbed by properly designed acquisition systems, grey levels of objects are substantially different from grey levels in the background. Objects and background can thus easily be differentiated by a single threshold. Pixels with grey levels below the threshold are set to zero, all the other pixels to 255, resulting in a binary image. The image is now beautifully segmented into two classes, background and objects. Edges, however, are not yet detected, although this special task may well be a crucial step in image processing. Decoding barcode patterns, for instance, is based on the determination of the positions of the edges of dark and bright regions. The various methods of Hough transform usually are applied to edge images with well prepared edges, preferably thinned to a width of a single pixel. Generally, edges are at the heart of image processing as a field of metrology: edges are local differences in contrast, showing up in the image due to proper lighting of the scene.

A well-prepared raw image and the subsequent extraction of edges are shown in figure 1. The source image can simply be segmented with a single constant grey level threshold. The next step, labeling, is an analysis of the neighborhood relations between pixels. The result is not only a well-defined relation between every pixel in the foreground and a group of connected pixels ("blob", in this case six separated objects), but also the identification of the so-called contour pixels, thus the edges of the objects. The labeling algorithm yields these results on the fly: contour pixels are object pixels with grey level 255 like any other object pixel, but they can easily be differentiated from inner pixels, since they are connected to the background. Labeling and highlighting only the contour pixels of the blobs in the resulting image turns out to be a proper method of edge detection in binary images. A similar approach is to extract a chain code for the contour of binary objects, based on contour tracing, which directly features the edges in an image.

Search Lines

Labeling and contour tracing are used in most image processing libraries, but they are by no means trivial. Every labeling algorithm first has to detect an object in the image. For this purpose, the binary image is checked line by line from the upper left to the lower right corner, and the grey level of every pixel is checked. The first pixel encountered with grey level 255 belongs to an object and necessarily is a contour pixel, thus is part of an edge. This simple, very efficient method of edge detection is widely used in industrial applications. It is immediately clear that search lines from the left, the right, from the top and from the bottom sides of the image will touch the adjacent con-

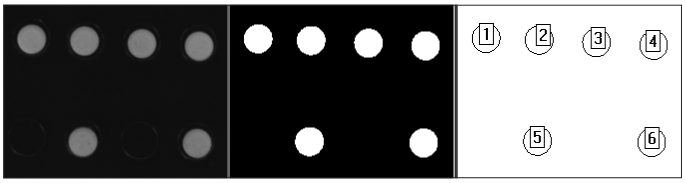


Fig.1: Grey level image, binary image and label image showing the contour

tour pixels. Search lines yield good results, whenever the position of an edge is roughly known, whereas the exact position and orientation may vary due to the characteristics of the handling system for the parts under investigation. Edges of this type are "caught" by use of search lines in predetermined regions of interest. Although a single search line might be sufficient in some cases, it is worthwhile to use several parallel, only closely separated search lines, checking for consistency to be one the safe side. An alternative approach might be to average the signals of several lines to eliminate noise or artifacts. In general, search lines are a very efficient method for edge detection, whenever certain information about the structure of the scene exists, much more efficient than labeling, which is based on the analysis of the whole image rather than single lines. A refinement of the use of simple search lines along the x- and y-axis only are search arrows, probing a ROI from several directions, possibly pointing to or emerging from a common center, to check the compactness of a hole, for instance.

Search Lines in Grey Level Images

When implementing a search line in software, you will immediately try to apply the idea directly to grey level images. It makes no sense at all to create a binary image and just look for the first white pixel on a line; it is much easier to take the grey levels from the line, check for the first pixel with a grey level above the threshold and stop searching, since the edge is already detected. The next step in the evolution of your method may be to get rid of the global threshold; why not just look at the difference between the grey levels of two adjacent pixels in the line? Once this difference is above a certain threshold, the edge will clearly be segmented from the background, and that is all you want to see! This works fine, and even inhomogeneous lighting or deviations from a flat optical field will have no significant influence on the result, since only the local (and not the global) differences in grey level between background and object are probed. Signals must not be in saturation or near to darkness, of course, and the threshold for the grey level difference has to exceed the noise level clearly. Apart from these restrictions, this method of edge detection is fairly robust, being a differential method based on the evaluation of the local contrast.

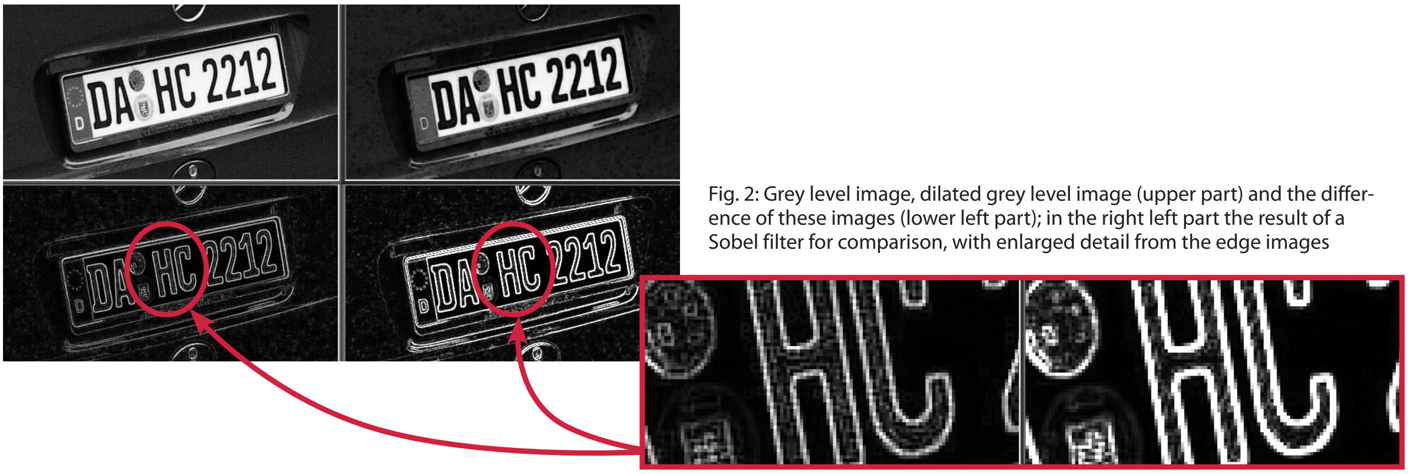


Fig. 2: Grey level image, dilated grey level image (upper part) and the difference of these images (lower left part); in the right left part the result of a Sobel filter for comparison, with enlarged detail from the edge images

A systematic approach along these lines of thought directly leads to the idea of the classical edge filter. Every mathematician in his original right mind, not yet spoilt by the strange ideas of engineers and the rude methods of physicists, will immediately see an edge as a region of rising or falling of a grey level function, the position of the edge given by the maximum of the derivative of said function, the derivate being approximated on the discrete grid of pixels on a 2D-plane by a difference-quotient. Indeed, the simple difference between the grey levels of adjacent pixels on a line (parallel to the x-axis, e.g.) is nothing else but the most primitive, crude and dirty implementation of the first derivative along that line (there are much better versions, of course). In order to make sure that edges are found independent of their orientation in the image, possibly with the same probability and even without shifting their gracious position, filters have to be carefully constructed. The absolute value of the gradient-vector roughly does the job. It may be plotted as the grey level in a resulting image, showing edges as bright areas and regions with slowly varying intensity as dark areas. Edge filters are broadly dealt with and very well documented in the literature [1, 2]. We shall not dare to touch them any more in this article.

Thinning

A typical result of an edge filter applied to an image is shown in figure 2 on the lower right hand side. This operation was a so-called Sobel filter, which is a clever combination of computing a derivative in one direction and of averaging the signals in the perpendicular direction, all within a single scan of the image. The edges are clearly visible as bright regions. The Sobel filter apparently does a good job in isolating the general region of an edge as a zone with a width of several pixels. To find the unique position of the edge, however, the edge image has to undergo further image processing. This is a well-known problem with edge filters. Although post-processing is possible, of course (looking for local maxima in the edge image, e.g.), further operations will consume valuable computing power and have to be robust, at least in industrial applications. Methods which directly result in edges thinned to a width of a single pixel, would probably have a valuable advantage. For binary images, there exist some reliable mechanisms of this kind. An example is shown in figure 3. A so-called dilation was applied to the binary source, resulting in objects which have a line of single additional pixels around their contour. In the next step, the difference between the source and the dilated image is computed, resulting in an image with the contour as the single remaining feature, depicted as a chain of single white pixels like pearls on a string. Dilation can also be applied to grey level images; this operation, however, does not only build up a single pixel along the edges. The difference image calculated by means of this procedure is quite similar to the result of a typical edge filter, as can be seen in figure 2, detail. In general, dilation is a rank filter and consumes a lot of computing power. If filter operations are definitely a no-go, you might try to simply shift the whole image by one pixel to the left and to the bottom and look at the results of the Boolean operations between the source and the shifted image. That can be really fast!

Conclusion

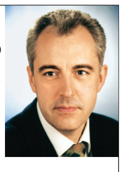
Thinned edge images can easily be computed with binary sources by labeling, contour tracing or by means of search lines. An alternative approach is to use a dilated image, subtracted from the binary source; this operation directly yields a thinned edge image. Edges in grey level images may be found by search lines and analysis of the slope of the grey level function along the line. This method gives good results, when the position and orientation of the edge is roughly known. Generalized methods have to make use of edge filters. After these operations, edges appear as bright bands, which usually have to be thinned with further image processing algorithms.

References

[1] B. Jähne, Digitale Bildverarbeitung, Springer-Verlag

[2] W. Burger, M. J. Burge, Digitale Bildverarbeitung, Springer-Verlag 2005, S. 111 ff.

2 Author
Prof. Dr. Christoph Heckenkamp



2 Contact
Darmstadt University of Applied Sciences
Optical Technology and Machine Vision
Darmstadt, Germany
heckenkamp@h-da.de
www.fbmn.h-da.de

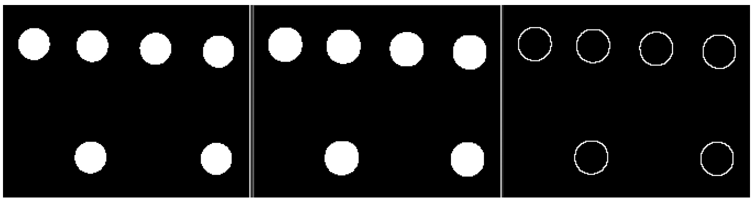


Fig. 3: Binary image, dilated binary image and the difference of these images

Key to markets Messe Stuttgart

VISION 2009

22nd International Trade Fair for Machine Vision and Identification Technologies

Neue Messe Stuttgart
3 to 5 November 2009

It's a VISION - It's a community

Every year companies from all over the world come to exhibit at VISION, the leading trade fair in the machine vision industry. And it is more and more as each year goes by. They come from 28 different countries and encounter visitors from an endless variety of industries. Highly-qualified visitors who are ready to invest and are in search of new products and application-oriented solutions. Will you be one of them?

www.vision-messe.de

Information and registration

Florian Niethammer
Tel.: +49 (0)711 18560-2541 | Fax: +49 (0)711 18560-2657
E-mail: florian.niethammer@messe-stuttgart.de