The optimal threshold algorithm and canny based edge detection in optical area and dimension measurement

Jaroslav Třeštík[1], Jan Fischer[2]

[1, 2] Department of Measurement, Faculty of Electrical Engineering, Czech Technical University in Prague, Technická 2, 166 27, Prague 6, Czech Republic, E-mail: [1]trestj1@fel.cvut.cz [2]fischer@fel.cvut.cz,

Abstract- The fluctuating light conditions cause troubles in image processing based area and dimension measurement. This article describes the hardware implementation and measured properties of canny-based edge detection [5] and an optimal thresholding method. Both units were implemented on proprietary FPGA-based image processing system for "on the fly" measurement. The optimal threshold algorithm presented is able to adapt to changed light conditions and calculates the object size counting the pixels above threshold. The other method uses the first and the second differentiation combined with low-pass filter do locate the object edges and calculates the object size by subtracting their positions. As it uses the maximal slope as edge criteria it is also resistant against illumination changes.

I. Introduction

The results of simple dimension and area measurement based on simple thresholding strongly depend on light conditions. The situation is depicted in figure 1 and figure 2. To get proper results at varying illumination the threshold level must be dynamically adapted. The other approach is the derivation-based, also called canny-based after John F. Canny edge detection method. The main disadvantages of canny-based edge detection methods [5] are higher complexity of the hardware and higher noise sensitivity. Several adaptations of the algorithm are proposed here to reduce the hardware complexity and increase the robustness.

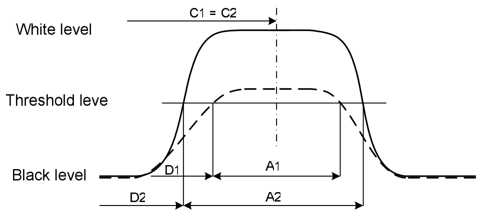

Figure 1 The impact of fluctuating light on dimension measurement

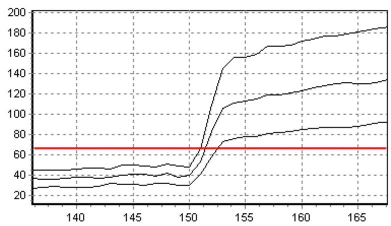

Figure 2 The real samples of the same line profile for three level of light

II. Optimal thresholding

The optimal threshold iterative algorithm uses the image histogram. It assumes the image is composed from foreground (the object) and background both with normally distributed pixel intensity. The histogram of the image is thus given by a weighted sum of these two probability density curves (see fig. 4). The difficulty of optimal threshold algorithm lies in estimating the normal distribution parameters together with the uncertainty whether the distribution may be considered normal. Deeper analysis of the optimal thresholding method are in [1], [2], [3] and [4]. Even if the method used in this work is fairly simple, it can be used even if the image histogram is not bimodal. In case of real measuring applications, the system is configured to produce images with two well separated regions present in the histogram. In this case, the eight bit resolution of the threshold level is reached after four to eight iterations. As the threshold algorithm is executed on the live data stream from the image sensor no image frame memory needs to be included into the camera module. Four images are usually enough to find the optimal threshold level. If the light changes faster, it affects the results of single frame measurement and none of these methods are usable.

$$\mu_B(t) = \frac{\sum_{(x,y)\in background} image(x,y)}{background\_pixels\_count} \qquad (1)$$

$$\mu_O(t) = \frac{\sum_{(x,y)\in object} image(x,y)}{object\_pixels\_count} \qquad (2)$$

$$T(t+1) = \frac{\mu_B(t) + \mu_O(t)}{2} \qquad (3)$$

The equations (1), (2) and (3) represent a single iteration step. μ_B is the mean value for the background part of the histogram (*image(x,y)* < *T*) and μ_O is the mean value for the object part of the histogram (*image(x,y)* > *T*). The *T(t+1)* is the new threshold level that will by used in the next step. The method converges fast even if the initial threshold value is far from the optimum. The examples of the algorithm behavior are presented in figures 3 to 7. The first image (figure 3) has histogram with well separated background and object histograms, the second image (figure 5) has histogram with only one dominant peak. The optimal threshold level is reached faster for the first image (up to fourth iterations). The second image was used only for the verification of the algorithm convergence speed as it does not represent contain any objects whose dimension can be measured. Images like figure 3 are more common in measurement tasks.

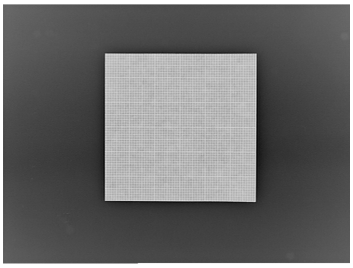

Figure 3 Easy recognizable object

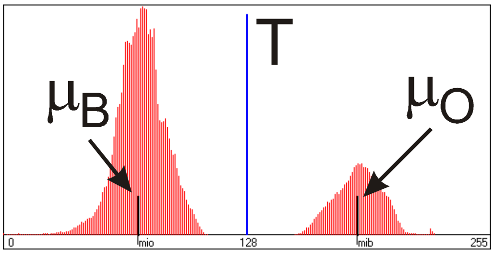

Figure 4 Histogram of the square

Figure 5 Real life photo

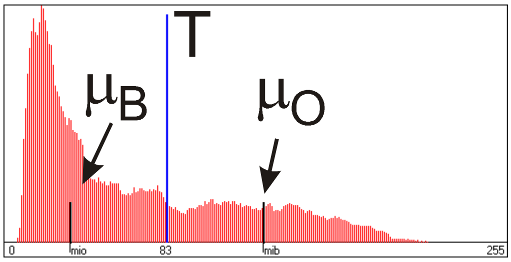

Figure 6 Histogram of the real life poto

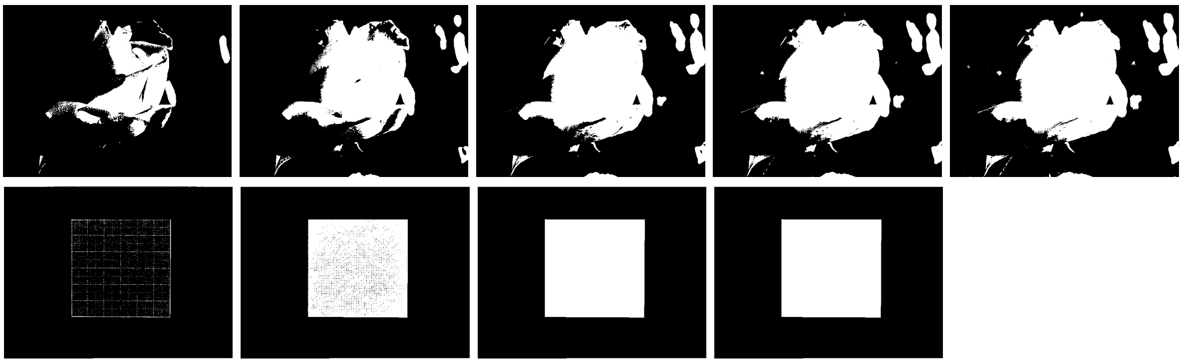

Figure 7 The series of threasholded images, the threshold level estimation for the easy recognizable object is faster then the estimation for the real life photo

III. Canny based edge detection

The 1D canny edge detector uses Gaussian smoothing and derivation. Equation (4) defines the Gaussian operator function with zero mean value and standard deviation σ. The input signal is filtered (smoothed) by finite impulse response filter (FIR) with Gaussian shape impulse response (5). The differenciation follows the filter block and can be described as simple FIR filter (6). As both filters are linear, they can by combined into a single filter with differentiated Gaussian (7) impulse response shape.

$$G(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{\frac{-x^2}{2\sigma^2}} \qquad (4)$$

$$F_G(z) = \frac{1}{\sqrt{2\pi\sigma}} \sum_{i=0}^{2K} z^{-i} e^{\frac{-(i-K)^2}{2\sigma^2}} \qquad (5)$$

$$F_D(z) = 1 - z^{-1} \qquad (6)$$

$$F_{GD}(z) = \frac{-1}{\sqrt{2\pi}\sigma^3} \sum_{i=0}^{2K} z^{-i}(i-K)e^{\frac{-(i-K)^2}{2\sigma^2}} \qquad (7)$$

The positions of edges are represented by minimums and maximums in the output signal. The task of edge detection is thus replaced by the local extreme detection task. By applying another difference filter, the local extremes are converted to zero crossing with slope proportional to the extreme. The zero crossing detection task can be easily implemented in the FPGA. The second derivation can be included into the convolution operator too, but it will be more sensitive to the noise contained in the image. When a nonlinear dead-zone unit is inserted between the first and second convolution (derivation), only the extremes pass through and the noise sensitivity is reduced. The order of the first derivation filter need to be adjusted depending on required precision and hardware complexity constrains. Three examples of convolution operators are shown in figure 8.

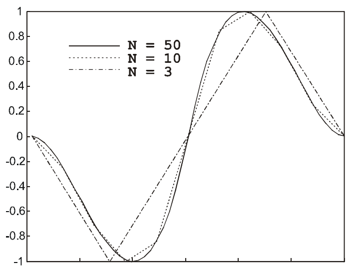

Figure 8 Convolution operator

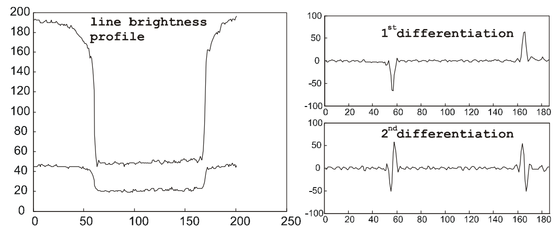

Figure 9 Line profile and its throughpass via first and second convolution

N is the order of the convolution operator (length of the filter). As the edge detection method was developed for low-end image sensors, the properties of 3rd order operator are the most interesting. The higher order operators imply more complex implementation in the FPGA, but the impact of increasing order is neglectable. The details of edges after filtering by third and fifth order operator are shown in figures 10 and 11 the zero crossing position is important.

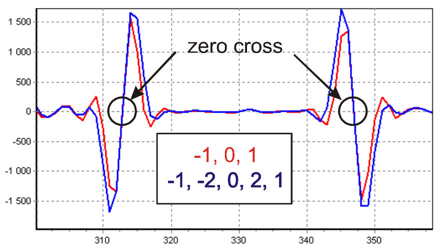

Figure 10 Third a fifth order canny detection without threshod after first pass

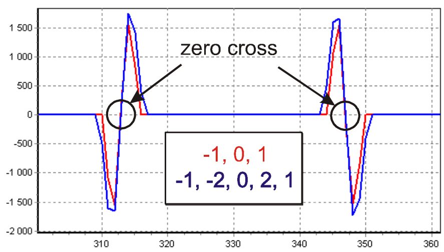

Figure 11 Third a fifth order canny detection with threshod after first pass

IV. Experimental setup

The abilities of the optimal threshold and canny based algorithm for the area and dimension measurement had to be checked on real data. All the measurements were carried out on the special image processing system (figure 12), with the Kodak CMOS image sensor KAC9638 used as the sensing element.

It is important to test all the algorithms on the same image data. The varying day light, varying temperature and sensor-related image noise can cause differences between two samples of the scene captured with the same settings. There is no image memory on the camera system therefore the algorithms should not be tested directly on this system. The camera system was used to capture the images and transfer them to the PC via USB interface. The algorithms were then simulated in Matlab environment.

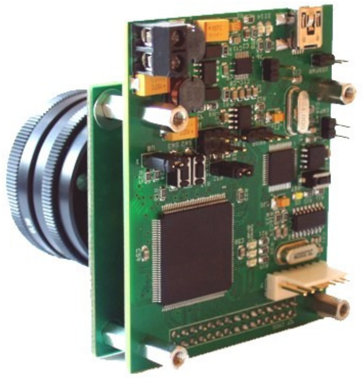

Figure 12 Realised Spartan 3 FPGA based camera system

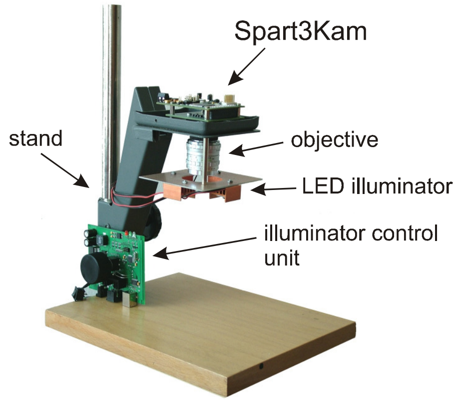

Figure 13 Measure setup

A high power LED light source was used to assure stable light conditions. Several testing models ware prepared. All the images were acquired with the same lens and mechanical settings. The detail of one edge of all captured images is presented in figure 16, the least illuminated image is in figure 14 the most illuminated one is in figure 15.

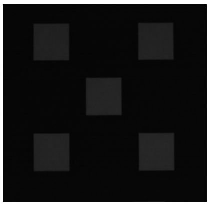

Figure 14 The lowest illuminated test image

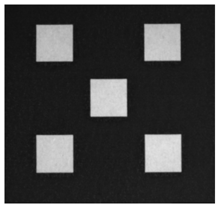

Figure 15 The highest illuminated test image

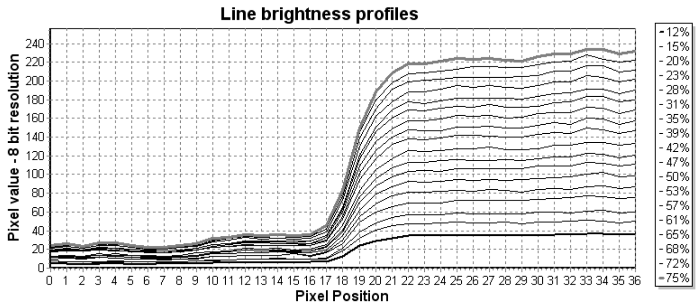

Figure 16 Detail of the edge

V. Area measurement

The area measurement was performed for all the captured images with the same area of interest settings (AIS). The number of iterations was experimentally set to six in case of the optimal threshold algorithm. The 8-bit threshold level was stable for all captured images after four iterations. The result of area measurement is the sum of pixels in AIS, where the pixel level is higher then the threshold level.

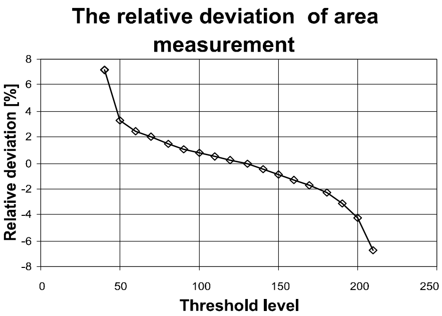

Figure 17 Manual thresholding impact on area measurement

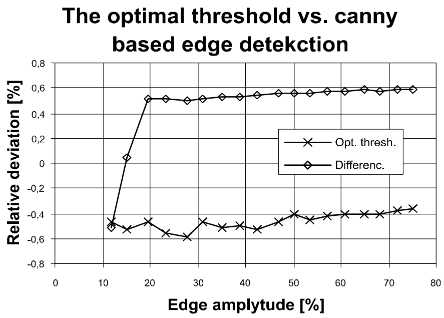

Figure 18 The errors of measured area obtained by both methods under different light contitions.

The dead-zone level was experimentally set to get no visible errors in the AIS for the canny-based edge detection. The result area is then the sum pixels between rising and falling edges of the object.
The manual threshold setting measurement was realized on the most illuminated image. The threshold was set from 40 to the 220. The relative deviation is less than ±8 %, the results are plotted in figure 17.
The results of both implemented methods is plotted in figure 18, here the correct size of the object is taken from the reference measurement with manually set threshold level at 50% of the edge height. Separately the relative deviation is less than ±0,15 %. The canny-based method fails at less illuminated images where the slow edges are rejected by the dead-zone block). The difference between both methods is approximately 1% and is constant.

VI. Dimension measurement

The dimension measurement experiment was realized on the same set of images. To achieve sub-pixel resolution, Booth algorithms were improved by adding a linear interpolation block before the zero cross detection block in the canny-based algorithm and before the threshold detect block in the optimal threshold algorithm.
Like in the area measurement the object width was first measured with manual. At the most illuminated image the threshold was set from 40 to 200. The ±3% deviation of the obtained object width was noted during test (figure 19). The results both methods are compared in figure 20. The relative deviation of both methods results is lower than ±0,1 % and in absolute numbers, the optimal threshold based method gives again lower values than canny-based.

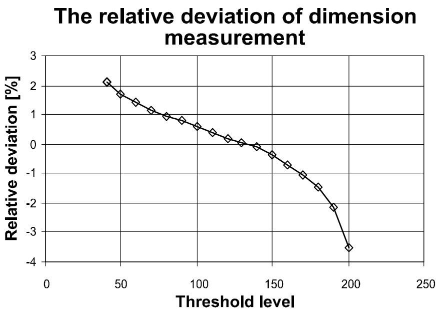

Figure 19 Manual thresholding inpact on dimension measurement

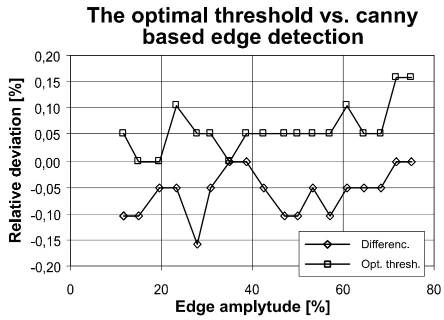

Figure 20 The errors of measured object width obtained by both methods under different light contitions.

VII. Conclusion

Two different methods eliminating the impact of fluctuating light conditions in image-based measurement were outlined. The optimal thresholding and canny-based edge detection methods have been realized in an FPGA based measurement camera system, developed in laboratory of videometry CTU FEE and as test scripts in Matlab environment.
The thresholding is easy implementable for many measurement tasks but for the fluctuating light conditions the optimal threshold algorithm has to be used. The system is able to calculate only one iteration in single frame as it does not contain the image frame memory. Four and more iteration steps has to be computed to achieve the optimal threshold level in case of steps change of the illumination intensity. The canny-based edge detection is more complex and can be difficult to use in more complex measurement tasks eg. complex shaped objects area measurement. No iteration is needed there and proper result is obtained in each frame. The aim of this article is to check and specify the abilities of the canny-based edge detection algorithm and the optimal threshold algorithm in optical measurement tasks. A set of measurements of the object dimension and area had been realized under different light conditions.
The experiment produces several conclusions for the measurement in varying light environment:

- Dimension and area measurements are moving within ± 2% to ± 9% deviation in case of no experience manual threshold settings.
- The area and dimension measurement results are within ±0,15 % error margin when the more complex algorithm is used.

The optimal threshold algorithm is suitable especially for area measurement where the light conditions changes slowly (no significant change within 4 adjacent frames). Algorithm is well implementable in target system. The 1D canny-based area measurement generates errors when the edge is not perpendicular to the scan line direction.
The canny-based algorithm is suitable for high-speed dimension measurement where the environment can produces fast light changes. This algorithm offers good results even if the object is not illuminated uniformly and reduces thus the required lighter complexity or vignetting error.

Acknowledgement

The research was supported by research program No. MSM6840770015 "Research of Methods and Systems for Measurement of Physical Quantities and Measured Data Processing" of the CTU in Prague sponsored by the Ministry of Education, Youth and Sports of the Czech Republic.

References

[1] Sonka M., Hlavac V., Boyle R., *Image Processing, Analysis, and Machine Vision,* 3rd ed., Toronto, 2008, ISBN: 978-0-495-08252-1

[2] Per Christian Henden, "Exercise in Computer Vision A Comparison of Thresholding Methods", *NTNU – Norwegian University of Scienc and Technology*, Faculty of Information Technology, Mathematics and Electrical Engineering, 2004

[3] Madria K. V., Haninsworth T. J., "A Spatial Thresholding Method for Image Segmentation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 10, No. 6, 1988

[4] Trussell H. J., "Picture Thresholding Using an Iterative Selection Method", *IEEE Transactions on Systems, Man and Cybernetics,* Vol. SMC-9, No. 5, 1979

[5] J. Třeštík, J. Fischer, "The Single Dimensional Edge Detection Method for the Intelligent Image sensor", *International Baltic Electronics Conference 2006*, Tallinn, ISBN: 1-4244-0415-0