Precise Edge Detection in 1-D Images for Contactless Measurement of Object

M. Hagara, P. Kulla

Institute of Electronics and Photonics FEI STU, Bratislava, Slovakia
Email: miroslav.hagara@stuba.sk

Abstract. *In some industrial applications it is necessary to measure only a single dimension of object. In such cases 1-D camera is sufficient and non-contact measurement is mostly based on edge detection with a precision of one pixel. If we need to increase the accuracy of measurement, we have two options, either use a 1-D optical sensor with more pixels or increase the accuracy of edge detection. In this paper we present comparison of four methods for edge detection with sub-pixel accuracy in 1-D images: method based on approximation of real image function with erf function, moment-based edge operator, technique using spatial moments of the image function and the method based on wavelet transform of image. The paper presents results of simulations as well as of experiments to compare methods in terms of accuracy, the standard deviation of the edge localization error is chosen as precision criterion.*

Keywords: Edge Detection, Sub-pixel Accuracy, Image Processing

1. Introduction

Some applications, e.g. measurement of the objects with high precision, need to detect edges with sub-pixel accuracy in 1-D images. There are a lot of methods for edge detection with sub-pixel accuracy and user can have a problem, which method is most suitable for his concrete purpose. This is a reason why we present in this paper the comparison of four methods for edge detection with sub-pixel accuracy in 1-D images: method based on approximation of real image function with erf function (AEF) [1], gray level moment (GLM) edge operator [2], spatial moment (SM) edge detector [3] and edge detector [4] based on wavelet transform (WT). We used simulations as well as experiments to compare methods in terms of accuracy and we chose standard deviation of the edge localization error as precision criterion.

2. Edge Detectors with Sub-pixel Accuracy for 1-D Images

Tabatabai and Mitchel proposed grey level moment (GLM) edge operator for 1-D image [2] based on the first three moments of the input data sequence:

$$m_i = \frac{1}{n}\sum_{j=1}^{n} x_j^{\,i} \quad ...\ i = 1,2,3 \quad , \tag{1}$$

where x_1, x_2, ... x_n are image samples. Let suppose that they are the samples of ideal step edge (Fig. 1a) and p_h is a number of samples with gray level h (they are the pixels on the left of the edge). If we define the densities p_1 and p_2 as:

$$p_1 = \frac{p_h}{n} \tag{2}$$

$$p_2 = \frac{n - p_h}{n} = 1 - p_1 \tag{3}$$

then solution of three equations

$$m_1 = (1 - p_2)h + p_2(h + k) \quad (4)$$

$$m_2 = (1 - p_2)h^2 + p_2(h + k)^2 \quad (5)$$

$$m_3 = (1 - p_2)h^3 + p_2(h + k)^3 \quad (6)$$

with three unknown variables h, k, p_2 results in

$$p_2 = \frac{1}{2}\left(1 + s\sqrt{\frac{1}{4 + s^2}}\right) , \quad (7)$$

where
$$s = \frac{m_3 + 2m_1^3 - 3m_1 m_2}{\left(m_2 - m_1^2\right)^{3/2}} . \quad (8)$$

In the case of real image, $p_h = n.p_1$ is not integer and represents sub-pixel edge location. Another sub-pixel edge detector [3] is based on spatial moments (SM) of continuous function $f(x)$ of order p, which are defined

$$M_p = \int x^p f(x) dx . \quad (9)$$

Let function $f(x)$ represents step edge and x is from –1 to +1 (Fig. 1b), to simplify calculations. Then Eq. 9 for $p = 0, 1$ and 2 can be written as

$$M_0 = h\int_{-1}^{1} dx + k\int_{l}^{1} dx = 2h + k(1 - l) \quad (10)$$

$$M_1 = h\int_{-1}^{1} x\, dx + k\int_{l}^{1} x\, dx = \frac{1}{2}k\left(1 - l^2\right) \quad (11)$$

$$M_2 = h\int_{-1}^{1} x^2\, dx + k\int_{l}^{1} x^2\, dx = \frac{2}{3}h + \frac{1}{3}k\left(1 - l^3\right) . \quad (12)$$

The solution of these equations results in formula for edge location l

$$l = \frac{3M_2 - M_0}{2M_1} . \quad (13)$$

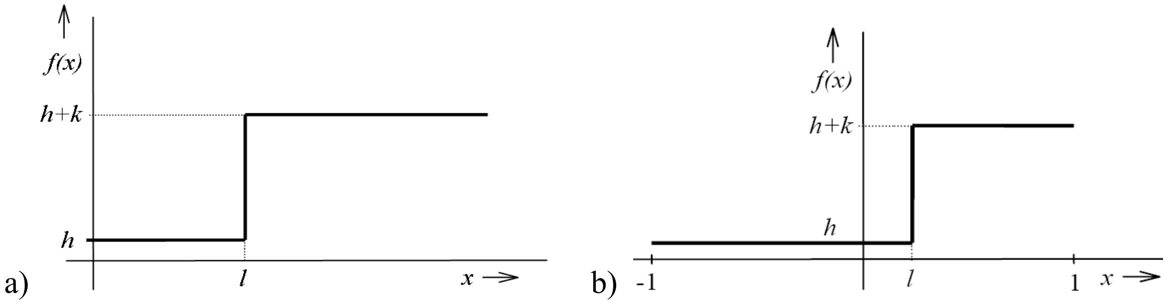

Fig. 1. a) Step edge model. b) Edge model for spatial moment edge detector.

Third edge detector used for comparison is based on wavelet transform [4]. If x_1, x_2, ... x_n are the samples of the image and $W_s(x_i)$ are values of wavelet transform, then

$$p_i = \frac{W_s(x_i)}{\sum_{j=1}^{n} W_s(x_j)} \quad (14)$$

is the probability that the sample $W_s(x_i)$ of wavelet transform is a local maximum. Position l of the edge is calculated using the formula

$$l = \sum_{i=1}^{n} i.p_i \ . \tag{15}$$

Last sub-pixel edge detector used for comparison is based on approximation of real image function $f_r(i)$ with function $f_a(x)$ [1], which is equal to

$$f_a(x) = \frac{k}{2}\left(erf\left(\frac{x-l}{\sqrt{2}\sigma} \right) + 1 \right) + h \ , \tag{16}$$

where

$$erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \ . \tag{17}$$

Function $f_a(x)$ has four parameters –h, k, l and σ. The core of this AEF edge detector is parametric fitting by minimizing a difference between the real image function $f_r(i)$ and function $f_a(i)$. This difference is defined

$$E(h,k,l,\sigma) = \sum_{i=1}^{N} \left(f_r(i) - f_a(i) \right)^2 \tag{18}$$

where N is a number of samples. Minimizing the difference $E(h,k,l,\sigma)$ gives subpixel edge location l. Edge detection algorithm based on approximation consists of three steps: edge detection with pixel accuracy, initial values estimation of parameters (h,k,l,σ) and parametric fitting by minimizing difference function $E(h,k,l,\sigma)$. First step can be done by any edge detection method with pixel accuracy. One can find how to estimate initial values σ_0, h_0 and k_0 in [1]. To minimize difference function $E(h,k,l,\sigma)$ we apply Matlab function *fminsearch*.

3. Results of Simulations and Experiments

We did all simulations in program Matlab. Let there is 1-D image sensor which consists of elements with width w and gap g between two sensor elements. Let the brightness around the edge is constant in time and varies only in the direction x according to blurring edge model [5]. Then simulated noiseless output signal $f_{rs}(i)$ is [1]

$$f_{rs}(i) = \gamma T_a \int_{i-w/2}^{i+w/2} \left(\frac{k}{2}\left(erf\left(\frac{x-l}{\sqrt{2}\sigma} \right) + 1 \right) + h \right) dx \, , \tag{19}$$

where σ represents edge blurring parameter, γ is sensor integral sensitivity and T_a is accumulation time. For simulations we can set $\gamma T_a = 1$, $w = 1$, $h = 0.1$ and $k = 1$. To add noise to signal defined in Eq. 19 we apply Matlab function *randn*, which returns a pseudorandom value with a normal distribution. We did simulations for signal-to-noise ratio SNR = 30dB. To get random difference c between actual position of the edge and centre of corresponding sensor element we used Matlab function *rand*, which returns pseudo-random values drawn from a uniform distribution on the unit interval, so $c = rand - 0.5$. We calculated noisy signal for different values of blurring parameter σ with random c two thousand times and for each noisy signal we used 31 samples around the edge to find sub-pixel edge location. Then we calculated edge location error for all realizations and for these errors we determined the standard deviation STD_{cn}. Results are presented in Table 1 and in Fig. 2a. For experimental verification we shot few images of car engine valve. We shot some images with manual focusing so we got unfocused images with different values of blurring parameter. Since the valve must be perfectly straight, computed sub-pixel edge positions should create a straight line, which can be represented as polynomial. Difference between the computed edge position and the value of the polynomial can be considered to be the edge location error. Obtained results are presented in Table 2 and in Fig. 2b.

Table 1. Standard deviation STD_{cn} (in px) of edge location error of simulated noisy signal.

σ [px]	0.75	1	2	3	4
AEF	0.029	0.032	0.048	0.060	0.082
GLM	0.030	0.033	0.050	0.720	0.127
SM	0.083	0.082	0.082	0.115	0.241
WT	0.093	0.087	0.097	0.122	0.234

Table 2. Standard deviation STD_e (in px) of edge location error (experiments).

σ_s [px]	0.84	1.54	2.18	3.25	4.25
AEF	0.031	0.043	0.060	0.075	0.092
GLM	0.031	0.043	0.060	0.094	0.129
SM	0.045	0.045	0.062	0.129	0.236
WT	0.054	0.047	0.063	0.144	0.265

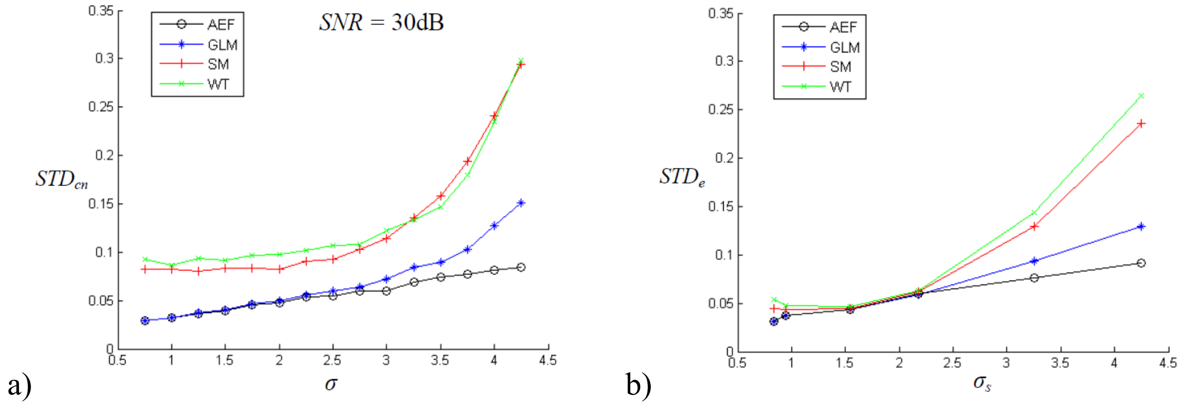

Fig. 2. Standard deviation of edge location error (in px): a) simulations, b) experiments (σ and σ_s are in px).

4. Discussion

We can conclude on the basis of simulations and experiments that for well-focused images (σ < 1) AEF and GLM methods offer highest accuracy. GLM can be preferred because it has the lowest computation time among all the compared methods. For slightly unfocused real images (1 ≤ σ < 2.5) the accuracy of all methods is roughly equal. However, GLM method can be preferred because of its lowest computation time. For strongly unfocused real images (2.5 ≤ σ) AEF method is most accurate. But the computational time of this method is significantly larger than computational time of the GLM method, so it can be used in the case, if computational time is not important.

Acknowledgements

This work was supported by grant No. VEGA 1/0987/12.

References

[1] Hagara M, Kulla P. Edge detection with sub-pixel accuracy based on approximation of edge with erf function. *Radioengineering*, 20 (2): 516-524, 2011.

[2] Tabatabai AJ, Mitchel OR. Edge location to subpixel values in digital imagery. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 6 (2): 188-201, 1984.

[3] Lyvers EP, Mitchel OR, Akey ML, Reevs AP. Subpixel measurements using a moment-based edge operator. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 11 (12): 1293-1309, 1989.

[4] Xu GS. Study on a fast method for sub-pixel edge detection. In proceedings of the 3rd International Conference on Information and Computing, 2010, 141-144.

[5] Zhang W, Bergholm F. Multi-scale blur estimation and edge type classification for scene analysis. *International Journal of Computer Vision*, 24 (3): 219-250, 1997.