Error analysis of subpixel edge localisation

Patrick Mikulastik, Raphael Höver and Onay Urfalioglu

Information Technology Laboratory (LFI), Leibniz University of Hannover

Abstract: In this work we show analytically and in real world experiments that an often used method for estimating subpixel edge positions in digital camera images generates a biased estimate of the edge position. The influence of this bias is as great as the uncertainty of edge positions due to camera noise. Many algorithms in computer vision rely on edge positions as input data. Some consider an uncertainty of the position due to camera noise. These algorithms can benefit from our calculation by adding our bias to their uncertainty.

Keywords: subpixel accuracy, edge detection, parabolic, regression

1 Introduction

The low level task of precise edge detection is a basis for many applications in image processing and computer vision.

In this work we show analytically and in real world experiments that an often used method for estimating subpixel edge positions in digital images generates a biased estimate of the edge position. We show analytically that common algorithms similar to the well known Canny [1] edge detector, which use parabolic functions for subpixel refinement of edge positions exhibit a bias of the estimage.

So far extensive work has been done on edge localisation, as it is a crucial task in computer vision and image processing. Nevertheless we think that we can make some additions to this toppic. The effect of the bias revealed by our calculation is in the same range as the uncertainty of edge positions introduced by camera noise. Our theoretical results are verified in easily reproducible experiments with real world data.

Some past approaches to edge localisation [2,3,4,5] have shown good or even optimal approaches for continuous signals. Nevertheless, they are not easily portable to the case of digitised images because it is the interpolation process that introduces the bias to the edges positions. Other approaches [1,6,7] use digitised images but apply no subpixel interpolation of edge positions. Kisworo et. al. [8] apply a local energy approach to localise subpixel edges. However their experiments don't show if there is a systematic bias for their approach. Rockett et. al. [9] and Mikulastik [10] examine parabolic interpolation for subpixel edge

2-9525435-1 © SITIS 2006

localisation. Mikulastik focuses on dealing with camera noise and Rockett et. al. [9] find that there is no bias for edges parallel to the sampling raster. This is because they use synthetic images for fitting their edge model and no real world images. However, none of the researches mentioned above has measured or described the error that leads to a systematic bias, that we are introducing here.

In the next section we introduce our signal model and describe a sample edge detector similar to the detector by introduced by Canny [1]. Afterwards in section 3 we show in a calculation that a parabolic interpolation generates a bias, which we also measure in CCD camera images in section 4. Finally we give some conclusions and a summary in section 5.

2 Edge Detection

The following steps lead to an edge detector similar to the Canny [1] edge detector. We assume that the pulse response of the image acquisition system, in our case a CCD camera, can be modelled by a Gaussian with variance σ_{cam}. This is widely accepted and applied in the literature [11,2]. We model the edges to be localised by a Gaussian with variance σ_{edge} convolved with an ideal step. An ideal step edge modelled this way would have a variance of $\sigma_{edge} = 0$. With $I(x)$ being the intensity at a coordinate x, the signal for an edge at position x_{max} can be modelled with

$$I(x) = \int_x \left(h_{edge}(x - x_{max}) * h_{cam}(x)\right) dx$$

with

$$h(x) = -\frac{x}{\sqrt{2\pi}\sigma^3} \cdot e^{-\frac{1}{2}\left(\frac{x}{\sigma}\right)^2} \quad . \tag{1}$$

The following approach describes a 1D filter applied in horizontal direction, for detection of mostly vertical edges. It can be applied in two passes in horizontal and vertical direction, so that all edges in a 2D image can be detected.

To be able to detect edges as maxima we have to generate a gradient image. Since an ideal gradient operator would introduce aliasing to the image we need a lowpass filter combined with a gradient. We choose the first derivate of a Gaussian. The Gaussian acts as lowpass filter. Combined with the first derivate we get the desired gradient. Choosing an *impulse invariance design* approach we sample the analog derivate of the Gaussian to get our digital filter. We get minimal distortion through aliasing and windowing with $\sigma_{grad} = 1.0062$ for a five tab filter.

Because of the gradient operation our edges now have the form of sampled Gaussian functions. The sampled signal $g(x)$ we get after application of the gradient filter described above is:

$$\begin{aligned}\frac{\partial}{\partial x} I(x) = g(x) &= (h_{edge}(x - x_{max}) * h_{cam}(x) * h_{grad}(x)) \\ &= (h_{all}(x - x_{max})) \qquad ,\end{aligned}$$

where the functions $h(x)$ have the same form as seen in equation 1. The variance of the Gaussians is:

$$\begin{aligned}\sigma_{all} &= \sqrt{\tilde{\sigma}_{edge}^2 + \tilde{\sigma}_{cam}^2 + \sigma_{grad}^2} \\ &= \sqrt{\left(\frac{\sigma_{edge}}{cos(\varphi)}\right)^2 + \left(\frac{\sigma_{cam}}{cos(\varphi)}\right)^2 + \sigma_{grad}^2} \quad .\end{aligned} \tag{2}$$

The angle φ is zero for edges orthogonal to the filter and greater for rotated edges.

The generation of the gradient signal is very similar to that used in the well known detector by Canny [1]. Many approaches now use regression with a parabolic function to find the subpixel peak points in the gradient image that give us the exact edge positions. The following calculation shows that this leads to a systematic bias.

3 Calculation of bias

We can write the following equation for a parabola

$$\hat{g}_{par}(x) = a \cdot x^2 + b \cdot x + c \qquad . \tag{3}$$

The hat on $\hat{g}_{par}(x)$ indicates an estimated value for the real value $g(x)$, which is not available. The maximum of the parabola lies at

$$\hat{x}_{max} = -\frac{b}{2a} \qquad . \tag{4}$$

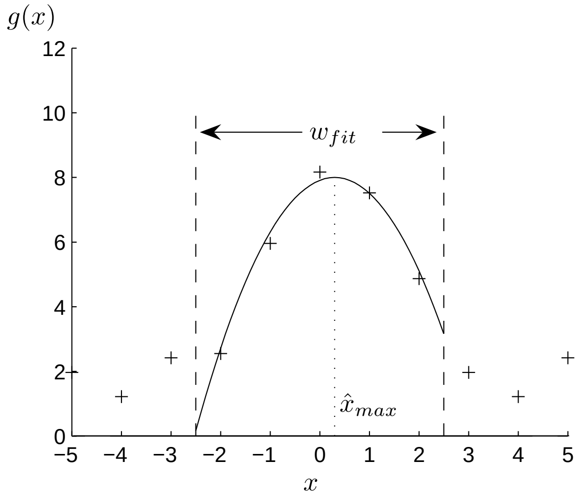

Figure 1. A parabola fitted to the signal $g(x)$. It is fitted to $w_{fit} = 5$ samples. The further away the sample positions are from $\hat{x}_{max}$ the more the samples of the gradient signal differ from a parabolic interpolation.

Figure 1 shows an example of a parabola fitted to the signal $g(x)$. From the figure we can deduce that the parabola approximates the signal $g(x)$ only in a window with the width w_{fit}. Outside this window the differences between parabola and $g(x)$ become quite big. Furthermore the signal $g(x)$ does not have the form of a parabola. It is just approximated by it. Therefore the maximum value x_{max} exhibits a systematic bias to the real maximum of $g(x)$. In the following section we show the derivation the systematic error $e_{sys} = \hat{x}_{max} - x_{max}$.

3.1 Systematic error

The systematic error is a function of the window width w_{fit} and the parameters σ_{all} and x_{max}.

$$e_{sys} = e_{sys}\left(w_{fit}, \sigma_{all}, x_{max}\right) \tag{5}$$

It is the difference between the estimated value $\hat{x}_{max}$ and the real value x_{max}.

$$\begin{aligned} e_{sys}\left(w_{fit}, \sigma_{all}, x_{max}\right) &= \hat{x}_{max} - x_{max} \\ &= -\frac{b}{2a} - x_{max} \end{aligned} \tag{6}$$

For regression with a parabolic function, as described in eq. 3, the following equation is valid and can be solved for a, b and c:

$$\begin{pmatrix} x_1^2 & x_1 & 1 \\ \vdots & \vdots & \vdots \\ x_{wfit}^2 & x_{wfit} & 1 \end{pmatrix} \cdot \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} g_{par}(x_1) \\ \vdots \\ g_{par}(x_{wfit}) \end{pmatrix} \tag{7}$$

$$\underline{\mathrm{A}} \cdot \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} g_{par}(x_1) \\ \vdots \\ g_{par}(x_{wfit}) \end{pmatrix} \tag{8}$$

$$\begin{aligned} \begin{pmatrix} a \\ b \\ c \end{pmatrix} &= \left(\underline{\mathrm{A}}^T \underline{\mathrm{A}}\right)^{-1} \underline{\mathrm{A}}^T \begin{pmatrix} g_{par}(x_1) \\ \vdots \\ g_{par}(x_{wfit}) \end{pmatrix} \\ &= \underline{\mathrm{M}} \cdot \begin{pmatrix} h_{all}(x_1 - x_{max}) \\ \vdots \\ h_{all}(x_{wfit} - x_{max}) \end{pmatrix} \\ &= \underline{\mathrm{M}} \cdot \boldsymbol{h}_{all}(\boldsymbol{x}_{wfit} - \boldsymbol{x}_{max}) \end{aligned} \tag{9}$$

with the matrix

$$\underline{\mathrm{M}} = \begin{pmatrix} \boldsymbol{m}_1^T \\ \boldsymbol{m}_2^T \\ \boldsymbol{m}_3^T \end{pmatrix} = \left(\underline{\mathrm{A}}^T \underline{\mathrm{A}}\right)^{-1} \underline{\mathrm{A}}^T \tag{10}$$

and the vectors

$$\boldsymbol{h}_{all}(\boldsymbol{x}) = \begin{pmatrix} h_{all}(x_1) \\ \vdots \\ h_{all}(x_N) \end{pmatrix} \qquad \boldsymbol{x}_{wfit} = \begin{pmatrix} x_1 \\ \vdots \\ x_{wfit} \end{pmatrix} \qquad \boldsymbol{x}_{max} = \begin{pmatrix} x_{max} \\ \vdots \\ x_{max} \end{pmatrix} \tag{11}$$

With these expressions eq. 6 can be written as

$$\begin{aligned} e_{sys}\left(w_{fit}, \sigma_{all}, x_{max}\right) &= -\frac{b}{2a} - x_{max} \\ &= -\frac{1}{2}\frac{\boldsymbol{m}_2^T \cdot \boldsymbol{h}_{G,Ges}(\boldsymbol{x}_{wfit} - \boldsymbol{x}_{max})}{\boldsymbol{m}_1^T \cdot \boldsymbol{h}_{G,Ges}(\boldsymbol{x}_{wfit} - \boldsymbol{x}_{max})} - x_{max} \end{aligned} \tag{12}$$

For the following calculation it is assumed that the maximal sample of $g(x)$ is located at position $x = 0$. This doesn't limit generality, since it can be achieved by a simple coordinate transformation. Therefore we can write $\underline{\mathrm{A}}$ as:

$$\underline{\mathrm{A}} = \begin{pmatrix} \left(-\frac{w_{fit}-1}{2}\right)^2 & -\frac{w_{fit}-1}{2} & 1 \\ \vdots & \vdots & \vdots \\ \left(\frac{w_{fit}-1}{2}\right)^2 & \frac{w_{fit}-1}{2} & 1 \end{pmatrix} \quad . \tag{13}$$

The elements $d_{i,j}$ of a matrix $\underline{\mathrm{D}} = \underline{\mathrm{A}}^T\underline{\mathrm{A}}$ are:

$$\begin{aligned} d_{1,1} &= \left(-\frac{w_{fit}-1}{2}\right)^4 + \left(-\frac{w_{fit}-1}{2}+1\right)^4 + \cdots + \left(-\frac{w_{fit}-1}{2}+w_{fit}-1\right)^4 \\ &= \sum_{k=0}^{w_{fit}-1}\left(-\frac{w_{fit}-1-2k}{2}\right)^4 = 2\cdot\sum_{k=0}^{\frac{w_{fit}-1}{2}} k^4 \end{aligned} \tag{14}$$

$$\begin{aligned} d_{1,2} &= \left(-\frac{w_{fit}-1}{2}\right)^3 + \left(-\frac{w_{fit}-1}{2}+1\right)^3 + \cdots + \left(-\frac{w_{fit}-1}{2}+w_{fit}-1\right)^3 \\ &= \sum_{k=0}^{w_{fit}-1}\left(-\frac{w_{fit}-1-2k}{2}\right)^3 = 0 \end{aligned} \tag{15}$$

$$\begin{aligned} d_{1,3} &= \left(-\frac{w_{fit}-1}{2}\right)^2 + \left(-\frac{w_{fit}-1}{2}+1\right)^2 + \cdots + \left(-\frac{w_{fit}-1}{2}+w_{fit}-1\right)^2 \\ &= \sum_{k=0}^{w_{fit}-1}\left(-\frac{w_{fit}-1-2k}{2}\right)^2 = 2\cdot\sum_{k=0}^{\frac{w_{fit}-1}{2}} k^2 \end{aligned} \tag{16}$$

and

$$d_{2,1} = 0 \qquad d_{2,2} = d_{1,3} \qquad d_{2,3} = 0 \tag{17}$$

$$d_{3,1} = d_{1,3} \qquad d_{3,2} = 0 \qquad d_{3,3} = w_{fit} \tag{18}$$

and the abbreviations μ and λ defined as:

$$\mu = 2 \cdot \sum_{k=0}^{\frac{w_{fit}-1}{2}} k^4 \tag{19}$$

$$\lambda = 2 \cdot \sum_{k=0}^{\frac{w_{fit}-1}{2}} k^2 \qquad , \tag{20}$$

the matrix $\underline{\mathrm{D}} = \underline{\mathrm{A}}^T \underline{\mathrm{A}}$ can be written as:

$$\underline{\mathrm{D}} = \begin{pmatrix} \mu \; 0 & \lambda \\ 0 \; \lambda & 0 \\ \lambda \; 0 & w_{fit} \end{pmatrix} \qquad . \tag{21}$$

The inverse $\underline{\mathrm{D}}^{-1}$ is:

$$\underline{\mathrm{D}}^{-1} = \begin{pmatrix} \frac{1}{\mu} - \frac{\lambda^2}{\mu(\lambda^2-\mu w_{fit})} & 0 & \frac{\lambda}{\lambda^2-\mu w_{fit}} \\ 0 & \frac{1}{\lambda} & 0 \\ \frac{\lambda}{\lambda^2-\mu w_{fit}} & 0 & -\frac{\mu}{\lambda^2-\mu w_{fit}} \end{pmatrix} \qquad . \tag{22}$$

Now, the matrix $\underline{\mathrm{M}} = (\underline{\mathrm{A}}^T \underline{\mathrm{A}})^{-1} \underline{\mathrm{A}}^T$ becomes:

$$\begin{aligned} \underline{\mathrm{M}} &= \begin{pmatrix} \frac{1}{\mu} - \frac{\lambda^2}{\mu(\lambda^2-\mu w_{fit})} & 0 & \frac{\lambda}{\lambda^2-\mu w_{fit}} \\ 0 & \frac{1}{\lambda} & 0 \\ \frac{\lambda}{\lambda^2-\mu w_{fit}} & 0 & -\frac{\mu}{\lambda^2-\mu w_{fit}} \end{pmatrix} \cdot \begin{pmatrix} \left(-\frac{w_{fit}-1}{2}\right)^2 & \cdots & \left(\frac{w_{fit}-1}{2}\right)^2 \\ -\frac{w_{fit}-1}{2} & \cdots & \frac{w_{fit}-1}{2} \\ 1 & \cdots & 1 \end{pmatrix} \\ &= \begin{pmatrix} \frac{\lambda-\frac{w_{fit}}{4}(w_{fit}-1)^2}{\lambda^2-\mu w_{fit}} & \frac{\lambda-\frac{w_{fit}}{4}(w_{fit}-3)^2}{\lambda^2-\mu w_{fit}} & \cdots & \frac{\lambda-\frac{w_{fit}}{4}(w_{fit}-2w_{fit}+1)^2}{\lambda^2-\mu w_{fit}} \\ \frac{1}{\lambda} \cdot \left(-\frac{w_{fit}-1}{2}\right) & \cdots & \cdots & \frac{1}{\lambda} \cdot \left(\frac{w_{fit}-1}{2}\right) \\ \frac{\frac{\lambda}{4}(w_{fit}-1)^2-\mu}{\lambda^2-\mu w_{fit}} & \frac{\frac{\lambda}{4}(w_{fit}-3)^2-\mu}{\lambda^2-\mu w_{fit}} & \cdots & \frac{\frac{\lambda}{4}(w_{fit}-2w_{fit}+1)^2-\mu}{\lambda^2-\mu w_{fit}} \end{pmatrix} \end{aligned} \tag{23}$$

with the vectors $\boldsymbol{m}_1^T$ and $\boldsymbol{m}_2^T$:

$$\boldsymbol{m}_1 = \frac{1}{\lambda^2 - \mu w_{fit}} \cdot \begin{pmatrix} \lambda - w_{fit}\left(\frac{w_{fit}-1}{2} - 0\right)^2 \\ \lambda - w_{fit}\left(\frac{w_{fit}-1}{2} - 1\right)^2 \\ \vdots \\ \lambda - w_{fit}\left(\frac{w_{fit}-1}{2} - (w_{fit}-1)\right)^2 \end{pmatrix} \tag{24}$$

$$\boldsymbol{m}_2 = \frac{1}{\lambda} \cdot \begin{pmatrix} -\frac{w_{fit}-1}{2} \\ \vdots \\ \frac{w_{fit}-1}{2} \end{pmatrix} \qquad . \tag{25}$$

As an example we consider an ideal edge with $\sigma_{edge} = 0$ and $\varphi = 0$. For the camera we choose $\sigma_{cam} = 1$, and for the gradient filter we set $\sigma_{grad} = 1,0062$ as chosen in section 2. For this example σ_{all} is

$$\sigma_{all} = \sqrt{\left(\frac{0}{cos(0)}\right)^2 + \left(\frac{1}{cos(0)}\right)^2 + 1,0062^2} = 1.4186 \qquad . \tag{26}$$

Figure 2 shows the systematic error e_{sys} as calculated in equation 12 for the values considered in this example. There are three curves for different values of w_{fit}. For greater values of w_{fit} the systematic error becomes greater. For $x_{max} = 0$ the systematic error is zero. In this case the samples of our gradient function are symmetric around the maximum sample and every symmetric function fitted through them results in the right estimated value $\hat{x}_{max}$. The systematic bias left and right of the coordinate $x_{max} = 0$ is due to the fact that the gradient of the image signal, which can be approximated by a Gaussian function, is interpolated with a parabolic function. This also explains why the bias is greater at greater values of w_{fit}. The further away the sample positions are from x_{max} the more the samples of the gradient signal differ from a parabolic interpolation. This can be seen in figure 1.

4 Experimental results

In order to verify the calculation from the last section, real world experiments are performed. For the comparison of calculated and measured values, ground truth data is needed. We recorded a scene containing one exact black and white edge. This was achieved by capturing a high resolution LCD showing a black and white edge. The edge lies vertically in the image so that in every row of the image there is one edge location. The camera is slightly rotated around its optical axis to achieve that the filmed edge appears slightly slanted. This way it has the whole range of possible subpixel edge positions from $x_{max} = -0.5$ to $x_{max} = 0.5$. Groundtruth edge positions are determined by fitting a line through all estimated edge positions. To be sure that camera noise is not affecting the analysis an average image was used. 1000 frames of the test image were taken

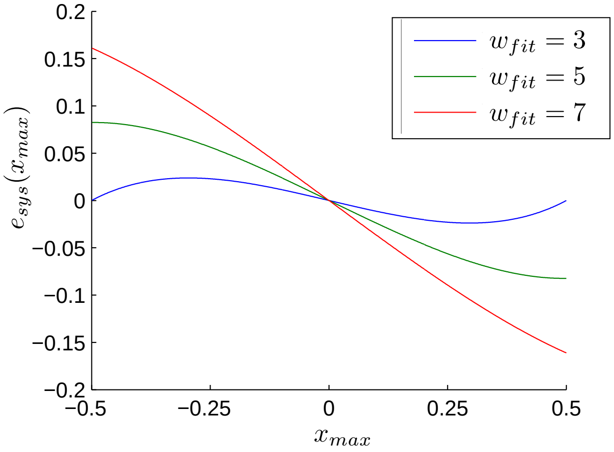

Figure 2. Systematic error $e_{sys}(x_{max})$ for the edge position when a parabolic function is used for interpolation of subpixel values. σ_{all} is set to $\sigma_{all} = 1,4186$.

with the camera installed on a tripod. The average value for each pixel position is used for the test image. Figure 3 shows a zoomed in view of our test image taken with a Sony DXC-D30WSP 3CCD video camera.

Figure 4 shows the measured bias e_m which is given through the difference of ground truth edge positions and estimated edge positions for each line of the test image.

Since the bias e_m repeats periodically for subpixel coordinates of edges from -0.5 to 0.5, it is sufficient to discuss one period of the signal, as can be seen in figure 5. Additional to the measured bias e_m, also the estimated bias e_{sys} is shown, for comparison. Measured and estimated values show only small differences, that can be explained by small differences in our signal model to that of the real camera.

In [10] a similar edge detector is examined in respect for the uncertainty of edge localisation due to camera noise. A Gaussian distributed localisation error variance of less than $0.002\,\mathrm{pel}^2$ for most edges was found in images with a high PSNR of 42. This corresponds to a standard deviation of $0.044\,\mathrm{pel}$. The maximum bias e_{sys} for a parabolic function through three sample values $w_{fit} = 3$ is about $0.025\,\mathrm{pel}$. This shows that the error introduced by the bias described here has almost the same size as the uncertainty due to camera noise and has to be considered in high level tasks that use edge positions as input data.

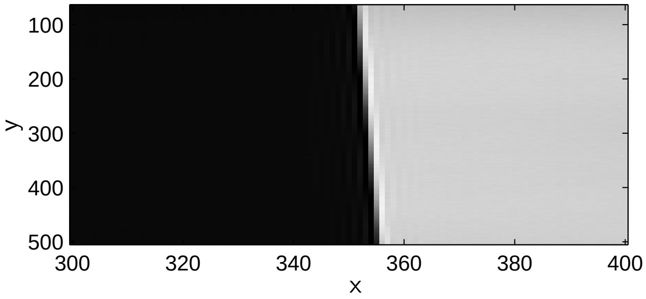

Figure 3. Test image for comparison of measured an calculated values. The edge is slightly slanted.

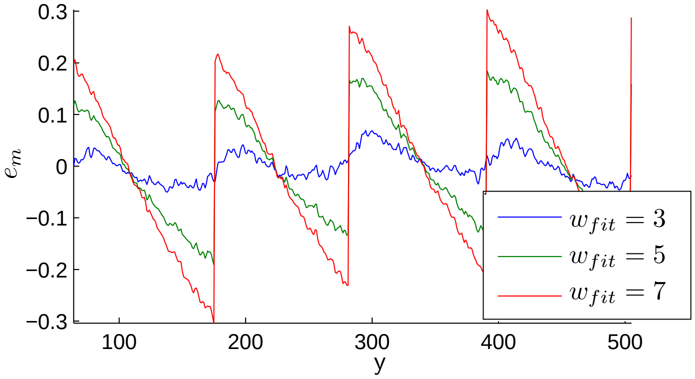

Figure 4. Measured bias e_m for each line of the test image.

5 Conclusions

This work has shown that edge detectors using parabolic functions for subpixel edge localisation estimate biased values for the edge positions. This is due to the fact that the gradient of the image signal, which can be approximated by a Gaussian function, is interpolated with a parabolic function. This bias is in the same range as errors of the edge positions due to camera noise. Therefore it is advisable to consider this bias in high level tasks that build upon precise edge localisation.

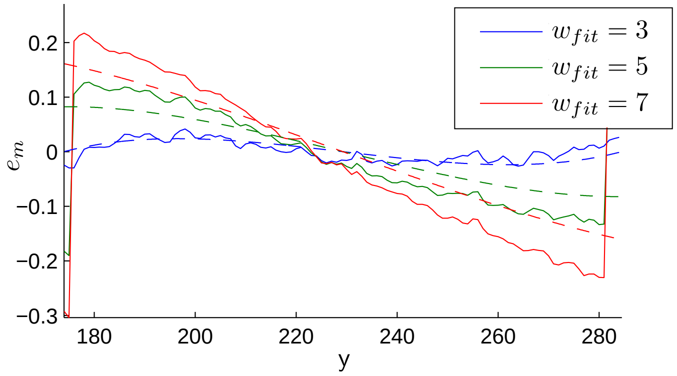

Figure 5. Measured bias e_m for lines 170 to 285 of the test image. The dashed lines show the the estimated bias e_{sys} for comparison.

References

1. Canny, J.F.: A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence **8** (1986) 679–698
2. R. Kakarala, A.H.: On achievable accuracy in edge localization. In: IEEE International Conference on Acoustics, Speech, and Signal Processing. Volume 4. (1991) 2545–2548
3. Jack Koplowitz, V.G.: On the edge location error for local maximum and zero-crossing edge detectors. IEEE Transactions on Pattern Analysis and Machine Intelligence **16** (1994)
4. Hemant D. Tagare, R.J.d.: On the localization performance measure and optimal edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence **12** (1990)
5. Piet W. Verbeek, L.J.v.V.: On the location error of curved edges in low-pass filtered 2-d and 3-d images. IEEE Transactions on Pattern Analysis and Machine Intelligence **16** (1994)
6. Kenji Suzuki, Isao Horiba, N.S.: Neural edge enhancer for supervised edge enhancement from noisy images. IEEE Transactions on Pattern Analysis and Machine Intelligence **25** (2003)
7. Richard J. Qian, T.S.H.: Optimal edge detection in two-dimensional images. IEEE Transactions on Image Processing **5** (1996)
8. M, Kisworo, S.V.G.W.: Modeling edges at subpixel accuracy using the local energy approach. IEEE Transactions on Pattern Analysis and Machine Intelligence **16** (1994)

9. Rockett, P.: The accuracy of sub-pixel localisation in the canny edge detector. Proc. British Machine Vision Conference (1999)
10. Mikulastik, P.: Genauigkeitsanalyse von Kantenpunkten fuer Stereokamerasysteme. In: Niccimon, Das Niedersaechsische Kompetenzzentrum fuer die Mobile Nutzung. Shaker Verlag (2006) 135–143 ISBN 3-8322-5225-2.
11. Mech, R.: Schaetzung der 2D-Form bewegter Objekte in Bildfolgen unter Nutzung von Vorwissen und einer Aperturkompensation. PhD thesis, University of Hannover, Germany (2003) ISBN 3-18-373210-6.