A Comparative Analysis of Algorithms for Determining the Peak Position of a Stripe to Sub-pixel Accuracy

D.K.Naidu
R.B.Fisher
Department of Artificial Intelligence, University of Edinburgh
5 Forrest Hill, Edinburgh EH1 2QL, UK.

Abstract

This paper provides a comparative analysis of five algorithms for determining the peak position in a laser ranging system to sub-pixel accuracy. The exercise was carried out in order to determine the most suitable method in terms of accuracy and robustness, for determining the location of the peak of an observed laser stripe to sub-pixel resolution.

1 Introduction

This paper describes a comparative analysis of the efficacy of five algorithms in determining the peak position of a laser stripe to sub-pixel accuracy. It is necessary in many applications to make detailed measurements which are outwith the scope of a system which relies on locational accuracy to the nearest pixel. It is, therefore, worthwhile to use algorithms that estimate feature positions to sub-pixel accuracy by interpolating the sensor response function (e.g.[1], [2]). In an imaging system which relies on accuracies to the nearest pixel while translating from 2-D camera coordinates to 3-D world coordinates, the accuracy of determination of the 3-D coordinates of a point in space will be limited by the resolution of the image from the camera. In our range sensor (working volume 20cm on a side), each pixel images about 1mm of the scene. In order to determine the stripe to sub-pixel accuracy, the image of the stripe must be blurred. This is almost always the case because although it is possible optically to focus the stripe to less than a single pixel width, the operative response of individual sensor elements often leads to a measurement that is several pixels wide.

The y-coordinate of the observed pixel is determined by the vertical distance of the scan line from the top of the image. The x-coordinate is determined by the location of the pixel across a particular scan line. Therefore, when we refer to the sub-pixel position of the peak of the stripe, we are discussing the x-coordinate of the pixel. Because of triangulation calculation, greater accuracy in determination of peak position in 2D will automatically result in a more accurate determination of the location of the peak in 3D coordinates.

BMVC 1991 doi:10.5244/C.5.28

2 Description of Algorithms

In all algorithms discussed below, the calculations use intensity values that have had the background intensity subtracted.

2.1 Gaussian approximation

The computation uses the three highest, contiguous intensity values around the observed peak of the stripe and assumes that the observed peak shape fits a Gaussian profile. This assumption is superficially reasonable as the light incident on the sensor is known to be nearly Gaussian distributed and the pixels have an overlapping, nearly Gaussian response. However, because each pixel integrates light over its field of view and because the physical sensor pads of the solid-state cameras we use have a gap between them, the distribution will not be exactly Gaussian. None the less, while we do not know the exact form of the distribution, we assume that the composition of all these effects can be modelled by a Gaussian distribution. If a, b and c are the intensity values observed at pixel positions $x-1$, x and $x+1$ with b having the highest value, then the sub-pixel location $(\hat{X})$ of the peak is given by :

$$\hat{X} = x - \frac{1}{2}\left(\frac{\ln(c) - \ln(a)}{\ln(a) + \ln(c) - 2\ln(b)}\right)$$

where x is the x-coordinate of the centre of the pixel with intensity value b. As a, b and c are integers in the range 0-255, the log calculation can be performed by table lookup.

2.2 Centre of Mass

The centre-of-mass(CoM3) algorithm also assumes that the spread of intensity values across the stripe conforms to a Gaussian distribution. Thus, the location of the peak can be computed by a simple weighted-average method. Suppose again that the three highest intensity values are given by a, b and c. Also suppose the x-coordinate of b is x, then the sub-pixel location of the peak is given by :

$$\hat{X} = \frac{a(x-1) + bx + c(x+1)}{a+b+c} = x + \frac{c-a}{a+b+c}$$

The extension of the algorithm for 5 and 7 points(called CoM5 and CoM7) is obvious. Algorithms to use all points along the raster scan[3] also exist.

2.3 Linear Interpolation

This method assumes that a simple, linear relationship defines the spread of intensity values across the stripe. Thus, if the three highest intensity values are identified as before, then :

If $c > a$, $\hat{X} = x - \frac{(a-c)}{2(b-a)}$ else, $\hat{X} = x - \frac{(a-c)}{2(b-c)}$

2.4 Parabolic Estimator

A continuous version of the peak finder is derivable from the Taylor series expansion of the signal intensity near the peak. If the peak is at $f(x+\delta)$ and we observe the signal at $f(x)$, then we have:

$$f'(x+\delta) = 0 = f'(x) + \delta f''(x) + O(\delta^2)$$

Neglecting the higher order terms, and estimating the derivatives discretely :

$$\delta \doteq -\frac{f'(x)}{f''(x)} \doteq \frac{f(x+1)-f(x-1)}{2(f(x+1)-2f(x)+f(x-1))}$$

This estimator is also that found by fitting a parabolic function to the points $f(x-1)$, $f(x)$ and $f(x+1)$. In the experiments below, we call this the **parabolic** estimator.

2.5 Blais and Rioux Detectors

Blais and Rioux[4] introduced fourth and eighth order linear filters:

$$\begin{aligned} g_4(x) &= f(x-2)+f(x-1)-f(x+1)-f(x+2) \\ g_8(x) &= f(x-4)+f(x-3)+f(x-2)+f(x-1)-f(x+1) \\ &\quad -f(x+2)-f(x+3)-f(x+4) \end{aligned}$$

to which we also add a second order filter:

$$g_2(x) = f(x-1) - f(x+1)$$

These operators act like a form of numerical derivative operator. The peak position is estimated as above by:

$$\delta = \frac{g(x)}{g(x)-g(x+1)}$$

Their results showed that the 4^{th} order operator had better performance than the 8^{th} order operator over the stripe widths that we are interested in here, so we only analyze it (called **BR4** below) and the simplified 2^{nd} order operator (called **BR2** below). The 8^{th} order operator has better performance for stripe widths with gaussian width parameter larger than 2.

3 Maximum Error of Estimators

Assuming that the observed stripe has Gaussian form and the true peak position is near to an observed pixel, we determine the relationship between the estimated and true peak positions (ie. offsets from that pixel), for each of the peak detectors. Assume that the continuous stripe is modeled by:

$$f(n) = e^{-\frac{(n-\delta)^2}{2\sigma^2}}$$

where $-\frac{1}{2} \leq \delta \leq \frac{1}{2}$ is the true peak position and f is sampled at $n = -2, -1, 0, 1, 2, \ldots$. We ignore the problems of pixels integrating their inputs over their spatial extent, as well as any shaping functions the camera and digitizer may apply.

We might ask what is the maximum deviation $\mid \delta - \hat{\delta} \mid$ over the range $-\frac{1}{2} \leq \delta \leq \frac{1}{2}$ for each estimator. We generated sampled stripes for values of δ over this interval and calculated the estimated $\hat{\delta}$. Figure 1(left) shows the error versus δ for the CoM7 estimator for $\sigma = 1.0$. By weighting the estimator ($\hat{\delta}' = \alpha_{estimator}\hat{\delta}$) we can, for a given σ, reduce the maximum error by spreading the error across the full range. Figure 1(right) shows the error for the resulting CoM7 estimator when $\alpha = 1.006$. When using the modified estimators, we can reduce the maximum errors to (using an α chosen to maximally reduce the error when $\sigma = 1.0$):

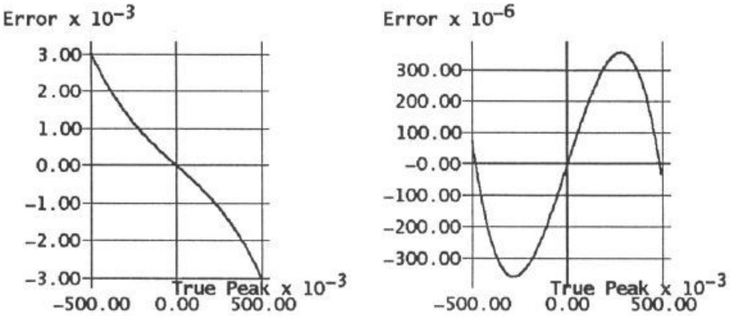

Figure 1: Error vs δ For Unbiased (left) and Biased (right) COM7 Estimator

σ	Gaussian	CoM3	CoM5	CoM7	Linear	Parabolic	BR2	BR4
0.5	0.0	0.380	0.041	0.021	0.103	0.156	0.026	0.023
1.0	0.0	0.005	0.002	0.000	0.030	0.029	0.024	0.013
1.5	0.0	0.239	0.150	0.057	0.049	0.034	0.022	0.011
α	1.0	1.85	1.093	1.006	0.93	1.08	0.95	0.975

This shows that, in at least the case of $\sigma = 1.0$, we can tune the estimator to have a very low error; however, setting the α values for one σ may produce reduced performance at other σs.

4 Non-Linearity of Estimators

Using the Gaussian stripe model in Section 3, we can determine an analytic model of the estimated peak offset $\hat{\delta}$ for a small, real offset, δ. Our analysis assumes first-order approximations, so:

$$f(n) \doteq e^{-\frac{n^2}{2\sigma^2}}(1 + \frac{n\delta}{\sigma^2})$$

We can now determine the form of $\hat{\delta}$ for each peak estimator:

Estimator	Local Estimate	Estimator	Local Estimate
Gaussian	δ	CoM3	$\frac{2\delta}{\sigma^2}\frac{e^{-\frac{1}{2\sigma^2}}}{(1+2e^{-\frac{4}{2\sigma^2}})}$
Linear	$\frac{\delta}{\sigma^2}\frac{e^{-\frac{1}{2\sigma^2}}}{(1-e^{-\frac{1}{2\sigma^2}})}$	CoM5	$\frac{2\delta}{\sigma^2}\frac{e^{-\frac{1}{2\sigma^2}}+4e^{-\frac{4}{2\sigma^2}}}{(1+2e^{-\frac{4}{2\sigma^2}}+2e^{-\frac{4}{2\sigma^2}})}$
Parabolic	$\frac{\delta}{2\sigma^2}\frac{e^{-\frac{1}{2\sigma^2}}}{(1-e^{-\frac{1}{2\sigma^2}})}$	CoM7	$\frac{2\delta}{\sigma^2}\frac{e^{-\frac{1}{2\sigma^2}}+4e^{-\frac{4}{2\sigma^2}}+9e^{-\frac{9}{2\sigma^2}}}{(1+2e^{-\frac{1}{2\sigma^2}}+2e^{-\frac{4}{2\sigma^2}}+2e^{-\frac{9}{2\sigma^2}})}$
BR2	$\frac{2\delta}{\sigma^2}\frac{e^{-\frac{1}{2\sigma^2}}}{(1-e^{-\frac{4}{2\sigma^2}})}$	BR4	$\frac{2\delta}{\sigma^2}\frac{e^{-\frac{1}{2\sigma^2}}+2e^{-\frac{4}{2\sigma^2}}}{(1+e^{-\frac{1}{2\sigma^2}}-e^{-\frac{4}{2\sigma^2}}-e^{-\frac{9}{2\sigma^2}})}$

The Gaussian estimator has the ideal form for small δ. From these results, we see that the parabolic operator gives one half the results of the linear operator. However, in light of the results from Section 3, we use the α estimator bias to change the linearity according to the algorithm. When $\sigma = 1.0$ (as approximately in our case), the resulting $\hat{\delta}$ is:

Estimator	Gauss	CoM3	CoM5	CoM7	Linear	Parabolic	BR2	BR4
$\hat{\delta}$	1.00δ	1.01δ	1.0δ	1.0δ	1.40δ	0.83δ	1.33δ	1.17δ

Hence, only the Linear and BR2 estimators are reasonably non-linear. Overall, this noise-free theoretical and empirical analysis suggests that the Linear, and BR2 estimators are not particularly good. However, given typical sensor substructure, pixel spatial integration and cross-talk, non-gaussian stripe formation and non-linear sensor transfer functions, errors of less than 5% seem unlikely in any case. Hence, the Gauss, CoM5, CoM7, Parabolic and BR4 estimators still seem like good candidates.

5 Errors in the Presence of Noise

In line with the experiments of Blais and Rioux[4], we investigated how error in the stripe data affected the estimated stripe position. These experiments were conducted by generating synthetic stripe data with a known, but randomly chosen stripe offset about an exact pixel position, and then corrupting the observed stripe intensity with noise. The main controlled variable was the stripe width. Uniform noise was added (following the model of Blais and Rioux). Point measurements were generated by:

$$s(m, x, \sigma, \beta) = e^{-\frac{(m-x)^2}{2\sigma^2}} + \beta n$$

where:

$x \in U[-0.5, +0.5]$ is the stripe position.
$m \in \{-3, -2, -1, 0, 1, 2, 3\}$ are the measured pixel positions.
$n \in U[0, 1]$ is the noise variable.
σ is the stripe width parameter (range 0.8 to 1.8).
β was the magnitude of the noise, and was considered for
$\beta = 0.0, 0.1, 0.25$, which bounded our observed noise level.

We measured both RMS error ($\sqrt{\frac{1}{N}\sum(x_i - \hat{x}_i)^2}$) and maximum deviation ($max_i \mid x_i - \hat{x}_i \mid$) as a function of σ for $N = 10,000$ samples. Figure 2 shows the RMS error for $\beta = 0.1$. Immediately, we see that the CoM3 and CoM5 estimators are problematic. What is surprising is the error of the CoM7 estimator at low stripe widths. However, this is understandable as, when the stripe width is low, the stripe intensities fall quickly at non-central pixels, causing the noise to more quickly dominate the signal and have a greater effect.

To compare the algorithms, we also summed the RMS error for $\sigma = 0.8 - 1.8$ (by 0.05) for the three values of β.

β	Gaussian	CoM3	CoM5	CoM7	Linear	Parabolic	BR2	BR4
0.00	0.00	3.71	1.36	0.31	0.87	0.49	0.39	0.24
0.10	1.07	3.90	1.86	1.32	1.36	1.23	0.93	0.77
0.25	2.49	4.25	2.67	2.63	2.62	2.61	2.12	1.86

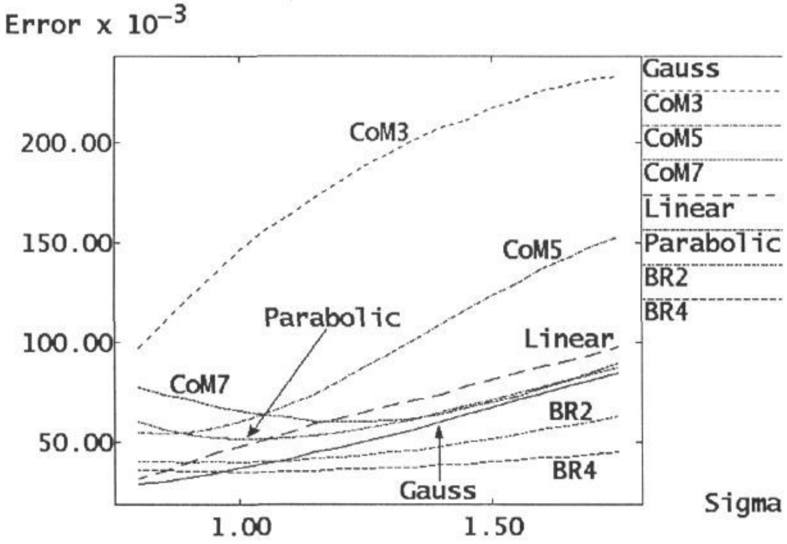

Figure 2: RMS Error Versus σ For The Estimators, Noise = 0.1

6 Empirical Testing

The experiments used three different test objects. These were a cube, a trapezoid with its top surface at an angle of 10° to the horizontal, and an equilateral prism (see Figure 3) to test the effect of surface orientation on peak location. In all cases, the object was oriented so that all range values along the stripe were equal. The experiments obtained a series of range images comprising 100 range stripes each. A single data point was chosen from each stripe, such that all the data points chosen lay along a line parallel to the x-axis. Secondly, the stripes were taken with a very small micro-stepper step size (0.2mm for the prism and 0.3mm for trapezoid and the cube), thereby leading to high data density. Also, the depth resolution was kept high (0.03mm), so that the errors being measured would be of larger magnitude than the quantisation errors.

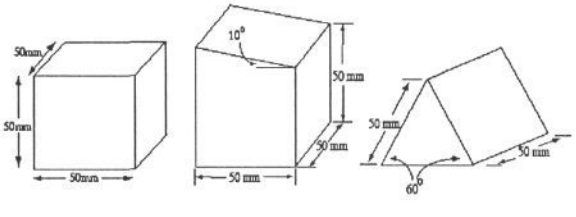

Figure 3: Detail of experimental objects

With each of the surfaces, each algorithm was used to detect the peak, and then the depth calculated from each camera was noted and their average computed. In the case of the trapezoid and the prism, the direction of the slope was alternated to eliminate the effect of the image of the stripe creeping across from one pixel to the next on the imaging surface of the cameras. The experiments with the prism were performed with the normal of the prism in the $x-z$ plane. A linear least-squares fit was computed for each set of data points. The slope of this fitted line ($z = \hat{a} + \hat{b}x$) was computed, and the minimum and maximum values of the errors from the fitted line were recorded. The variance of the errors was also computed. The comparative statistics are shown below.

Surface				Error(mm)		
	Algorithm	$\hat{a}$	$\hat{b}$	Min.	Max.	Variance
Flat Surface	Gaussian	49.7680	0.000112	-0.12762	0.09815	0.00249
	CoM - 3 pt	49.6466	0.000006	-0.06152	0.08812	0.00122
	CoM - 5 pt	49.7025	0.000130	-0.10121	0.09972	0.00156
	CoM - 7 pt	49.6239	0.000036	-0.09555	0.09658	0.00156
	Linear	49.7878	0.000090	-0.07676	0.08499	0.00120
	Parabolic	49.7397	0.000297	-0.14026	0.14496	0.00271
	BR - 2nd	49.6804	-0.000171	-0.21498	0.11769	0.00323
	BR - 4th	49.6808	-0.000411	-0.13912	0.16080	0.00343

Surface				Error(mm)		
	Algorithm	$\hat{a}$	$\hat{b}$	Min.	Max.	Variance
10° slope, left to right	Gaussian	57.2598	-0.05306	-0.24558	0.38329	0.02463
	CoM - 3 pt	57.7386	-0.05443	-0.45631	0.56070	0.05423
	CoM - 5 pt	57.6771	-0.05466	-0.37467	0.31389	0.03110
	CoM - 7 pt	57.6287	-0.05478	-0.29864	0.34434	0.02416
	Linear	57.6665	-0.05391	-0.30376	0.31999	0.01737
	Parabolic	57.7211	-0.05436	-0.23738	0.37868	0.01899
	BR - 2nd	57.8916	-0.05389	-0.27106	0.28737	0.01836
	BR - 4th	57.8261	-0.05382	-0.28686	0.33492	0.02110
10° slope, right to left	Gaussian	51.3929	0.05424	-0.31441	0.51424	0.03381
	CoM - 3 pt	51.8246	0.05257	-0.43193	0.47276	0.04573
	CoM - 5 pt	51.7356	0.05321	-0.34414	0.41777	0.03457
	CoM - 7 pt	51.6648	0.05277	-0.32581	0.42064	0.03128
	Linear	51.7305	0.05263	-0.35943	0.40299	0.03645
	Parabolic	51.7699	0.05281	-0.35278	0.41456	0.03496
	BR - 2nd	50.9346	0.05460	-0.33865	0.48187	0.03299
	BR - 4th	50.9251	0.05439	-0.30189	0.47461	0.03184

Surface				Error(mm)		
	Algorithm	$\hat{a}$	$\hat{b}$	Min.	Max.	Variance
60° slope, left to right	Gaussian	70.2352	-0.34806	-0.43505	0.45969	0.04737
	CoM - 3 pt	70.3217	-0.34726	-0.42918	0.51835	0.05024
	CoM - 5 pt	70.3418	-0.34816	-0.37187	0.55610	0.04146
	CoM - 7 pt	70.1715	-0.34771	-0.39819	0.51866	0.04106
	Linear	70.2851	-0.34810	-0.43443	0.42978	0.04897
	Parabolic	70.3154	-0.34841	-0.39460	0.50010	0.04764
	BR - 2nd	70.1493	-0.34781	-0.46431	0.51234	0.05624
	BR - 4th	70.1202	-0.34813	-0.47256	0.47195	0.04401
60° slope, right to left	Gaussian	23.5725	0.34834	-0.34344	0.44277	0.03918
	CoM - 3 pt	23.5071	0.34925	-0.33353	0.41938	0.03397
	CoM - 5 pt	23.5518	0.34852	-0.31148	0.35788	0.03213
	CoM - 7 pt	23.3997	0.34865	-0.40910	0.48459	0.03624
	Linear	23.5029	0.34894	-0.38680	0.47643	0.03994
	Parabolic	23.5249	0.34873	-0.39773	0.44770	0.03675
	BR - 2nd	23.5494	0.34841	-0.31710	0.46174	0.03628
	BR - 4th	23.5244	0.34870	-0.34116	0.52795	0.02985

The measured values for $\hat{a}$ are not particularly relevant. The values of $\hat{b}$ are of interest, because they specify the slope of the surface as measured by the algorithms used. The true value of $\hat{b}$ is 0 in the case of the cube, ± 0.0538 in the case of the trapezoid, and ± 0.3464 for the prism. These values were derived by careful, physical measurements of the objects, but are still subject to the usual measurement errors.

Figure 4 shows the variation of the residuals from the line of best fit applied to the measured data. The plots have each been offset by a different amount so that they are appear together on one graph. The maximum absolute variation of the plots from the line of best fit is about 0.5mm. Note the periodic structure of the residuals resulting from aliasing between the pixel spacing, inter-pixel gaps and the stripe width.

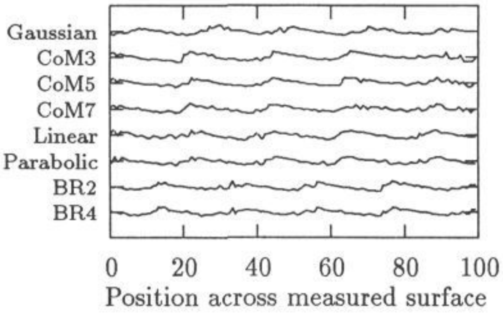

Figure 4: Comparative depth profiles across a 10°, right-to-left slope

The results from the trapezoid clearly show systematic errors in the system, caused by the image of the peak creeping from one pixel to the next, across the inter-pixel gap. The performance with the prism is even worse. This is because the acute angle of the object, which is very close to the angle of the camera axis to the horizontal, causes the stripe to move across the imaging sensors more quickly. In effect, the cameras are virtually sighting along the slope of the prism. The Gaussian and BR4 algorithms appear to show the least amount of perturbation. We estimate the inter-pixel gap to be almost as wide as a pixel.

7 Conclusions

The empirical results show that the CoM3 algorithm has poor performance. The other methods display performance within the same range probably because of factors such as sensor structure, inter-pixel gaps, cross-talk, and integration of the sensor response over the width of the pixel. The Linear and BR2 methods have been shown to possess high non-linearity (Section 4). When we consider the errors produced, the sum of the RMS errors are highest for the CoM3 and CoM5 algorithms. They are joined by the Parabolic algorithms when we consider the maximum errors. This leaves us with only the Gaussian, CoM7 and BR4 algorithms as suitable candidates.

The aliasing gives a periodic structure to the estimators. This could be estimated and an appropriate model used to correct for the effects. In the case of algorithms like the CoM7 and BR4, which rely on a large number of points around the peak, we notice that specular reflections and transparency may cause problems since the outlying pixels have a substantial effect on the computation of the location of the peak. Also, in the case where the object has holes in it, causing internal reflections and mutual illumination, the weighted average method of the CoM algorithms will deliver a skewed estimate of the peak position.

We can see good performance over a range of σ and β for the BR4 estimator. This is also clear in Figure 2; however, the Gaussian estimator has obvious benefits as the noise level or stripe width decreases. It is also interesting that the figures show to what extent the choice of estimator is linked to the specific stripe width and noise level. For our striper, the noise seems to be about 2-3 quanta, or about 1-5% of the peak intensity.

When comparing the speeds of the algorithms, The Gaussian is the slowest by about a factor of 2 over the Linear algorithm. However, the peak detection sub-process takes up only a small percentage of the total range image acquisition and peak detection time, so the speed of the algorithms is not a factor in their comparison.

These results apply to digitised video signals, whereas some algorithms, namely the centre-of-mass algorithm, can be applied directly to the video signal[2], [3]. An alternative approach to producing real-time stripe detection and sub-pixel location is to scan for the peak as the digitized intensity is calculated. Both approaches have been implemented in hardware and thus remove computational expense as a consideration, because one has to wait for the complete video scan anyway, when using standard video equipment.

8 Acknowledgements

The authors gratefully acknowledge the generous assistance of the University of Edinburgh and the European Institute of Technology (Grant EIT-061) for funding the research described in this paper.

References

[1] P.J.MacVicar-Whelan & T.O.Binford, "Intensity Discontinuity Location to Sub-pixel Precision", Proc. 7th Int. Conf. on A.I., pp 752 - 754, 1981.

[2] D.Braggins, "Achieving sub-pixel precision", Sensor Review, 1990.

[3] S.J.White, "Method and Apparatus for Locating Center of Reference Pulse in a Measurement System", U.S.Patent No. 4,628,469, December 1986.

[4] F.Blais & M.Rioux, "Real-Time Numerical Peak Detector", Signal Processing 11, pp 145-155, 1986.