A fast and Efficient Subpixelic Edge Detector

Frédéric Devernay

A Fast and Efficient Subpixelic Edge Detector

Frédéric Devernay
INRIA Sophia-Antipolis

Abstract

In this article we present a two dimensional edge extractor which gives the edge position in an image with a subpixelic precision. The method presented here gives a good accuracy with a low computational cost, and its implementation is very simple since it is derivated from the well-known Non-Maxima Suppression method [1, 3]. We also justify the method by showing that it gives the exact result in a theoretical one dimensional example. We have tested the edge extractor on several synthetical and real images and the results are reported in this paper.

1 Introduction

Edge detectors that are commonly used give the edge position up to within a pixel, but we really need more precision when we want to compute some differential properties of a curve such as orientation and euclidean, affine, or projective curvatures. Some attempts were made in subpixelic edge detection a few years ago, for example A. Huertas and G. Medioni [8] used a refinement of the zero-crossing of Laplacian but they did not give any results on the accuracy of the edge detection. A. J. Tababai and O. R. Mitchell [9] did some interesting work in the one-dimensional case which was extended to two-dimensional images and seemed to work properly. An edge relocation mechanism is also given in [10] but the required implementation is rather complex.

For these reasons we have made a very simple enhancement of the classical local non-maxima suppression, that gives a much better estimate of the curve position (up to within a tenth of a pixel) or the curve orientation without regularization. Using this edge detector, we can also calculate higher order differential properties of curves with much less regularization than when using older methods.

We present and interpret a wide variety of results to compare this method with existing edge detection methods. We also calculated the local edge orientation to show that the result of our method can also be used to calculate differential properties of the edges.

2 On edge extraction

The method we present here for subpixelic edge exctraction is based on the suppression of the local non-maxima of the magnitude of the gradient of image intensity in the direction of this gradient (also called NMS) [5], the other one being to consider edges as the zero-crossings of the Laplacian of image intensity [7, 6].

The NMS roughly consists of calculating values of the gradient norm in the direction of the gradient (Figure 1) by interpolation and then eliminating the pixel from the image if it is not a contour point, i.e. if it is not a maximum of the gradient norm. The interpolation can be either linear (between two near points) or quadratic (between three near points). To calculate the value of the gradient norm in B in Figure 1 we would use values at A_3 and A_4 for linear interpolation or at A_3, A_4, and A_5 for quadratic interpolation. After this edge detection process one usually does hysteresis thresholding [1] on the gradient norm and linking to get chains of pixels.

Our main improvement of the method is very simple and consists of only adding one single step to the NMS process: if (x, y) is a local maximum then estimate the position of the edge point in the direction of the gradient as the maximum of a quadratic interpolation on the values of gradient norm at (x, y) and the neighboring points. This maximum is simply the maximum of the parabola passing through the points $(-1, I(C))$, $(0, I(A))$, $(1, I(B))$ where A, B, and C are the points of Figure 1.

This is the principle, but we still have to find

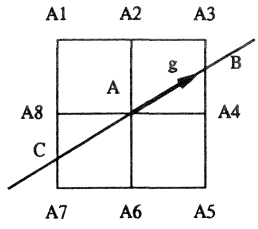

Figure 1: Checking whether pixel A is a local maximum of the magnitude of the gradient in the direction of the gradient.

the interpolation between gradient norm values we should apply to find the best position of the edge. We will consider a simple quadratic interpolation of the values of the gradient norm between the 3 values we have in the gradient direction. One could also try to locally fit a simple surface (e.g. biquadratic) on the neighborhood of the considered point but we want to keep the computations as simple as possible so that the implementation is fast and easy.

The choice of the quadratic interpolation to find the maximum can be justified because it gives the exact result in the one-dimensional case. Let $L = \{l(i)|i \in \aleph\}$ an infinite line of pixels with a step edge at position $i - \frac{1}{2}$, coordinate 0 corresponding to the middle of pixel $l(0)$. The continuous intensity function for this step edge is:

$$I(t) = \begin{cases} 0 & \text{if } t > \alpha - \frac{1}{2} \\ 1 & \text{otherwise} \end{cases}$$

so that

$$\begin{aligned} l(i) = & \int_{-\frac{1}{2}}^{\frac{1}{2}} I(t)dt \\ l(-\infty) = & \cdots = l(-2) = l(-1) = 1 \\ l(0) = & \alpha, 0 \leq \alpha \leq 1 \\ l(1) = & l(2) = \cdots = l(+\infty) = 0 \end{aligned}$$

And let ∇ be a general derivation operator. ∇ must be antisymmetric so that it can be written:

$$\begin{aligned} \nabla l(i) &= \sum_{k=1}^{+\infty} g_k(l(i+k) - l(i-k)) \\ \nabla l(i) &= \cdots - g_2 l(i-2) - g_1 l(i-1) \\ &\quad + g_1 l(i+1) + g_2 l(i+2) + \cdots \end{aligned}$$

Let us apply this general derivation operator on locations $-1, 0, 1$:

$$\nabla l(-1) = \cdots - g_2 - g_1 + g_1\,\alpha = C - (1-\alpha)\,g_1$$
$$\nabla l(0) = \cdots - g_2 - g_1 = C - g_1$$
$$\nabla l(1) = \cdots - g_2 - g_1\,\alpha = C - \alpha\,g_1$$

where $C = -\sum_{k=2}^{\infty} g_k$.

Since shifting and rescaling the values of the three points used to find a maximum of the quadratic interpolation do not affect the position of this maximum, we can simplify things by using $C = 0$ and $g_1 = 1$:

$$\begin{aligned} a = & \quad |\nabla l(-1)| = 1 - \alpha \\ b = & \quad |\nabla l(0)| = 1 \\ c = & \quad |\nabla l(1)| = \alpha \end{aligned}$$

Considering that the *center* of the pixels correspond to integer x coordinates, one can find that the x position of the maximum of the parabola passing through $(-1, a)$, $(0, b)$, and $(1, c)$ is:

$$m = \frac{a - c}{2(a - 2b + c)}. \qquad (1)$$

It can be easily seen that when $b \geq a$ and $b \geq c$ this value is bounded, and $-0.5 \leq m \leq 0.5$ Since integer coordinates correspond to pixel centers and pixel width is 1, this means that the subpixelic position of an edge point will always be *inside* the pixel.

That gives in this case $m = \alpha - \frac{1}{2}$ which is exactly the theoretical position of the edge! The justification of the choice of the quadratic interpolation in the two dimensional case has not been done because it involves too many parameters (including the way the values a and c are calculated). Equation 1 shows that the edge position is invariant to additive and multiplicative changes in the data. Moreover, this result can be extended to any kind of smooth edge, since a smooth edge is the result of the convolution of a step edge with a symmetric blurring operator s. The blurring operator s is symmetric so the action of the derivation on the ramp edge is the same as the action of the convolution of this operator with the blurring operator, which can be considered as another derivating operator, on the corresponding step edge:

$$\nabla(s \circ l)(i) = (\nabla \circ s)l(i) = \nabla' l(i)$$

Some people may want to use the squared gradient norm instead of the gradient norm in this process but in this case we find the maximum at $m = \frac{1-2\alpha}{4}$ which introduces a bias on the edge position. The maximum of the bias is $\delta m = 0.073$ pixels at $\alpha = 0.19$ and its standard

deviation is $\sigma(\delta m) = 0.052$, that is more than $\frac{1}{20}$ pixel (the same magnitude order as the precision we would like to get). In conclusion, using the squared gradient norm may reduce significantly the precision of the edge extraction.

Before seeing some results let us see how this edge detector can be used in a classical image processing chain.

3 Results

It seems to us that the qualities that should have a good edge detector are first a good estimate on the position of the edge, whatever the edge position, orientation and curvature; then good differential properties of the edge data, i.e. the edge orientation should not be biased and have a small variance, and higher order differential properties should be calculated accurately with not too much regularization; and finally all these properties should be robust to noise.

To verify these we used test data consisting of two series of images, each one consisting of a single edge. The first is a collection of lines, with light gray on one side and dark gray on the other side, with a wide variety of orientations. The second is a collection of filled circles with radiuses going from 3 to 100 pixels. These were 256 x 256 byte images generated with antialiasing, the dynamic of the edge was given, and Gaussian noise could be added on the image intensity data. We calculated the gradient of image intensity using a Deriche fourth order Gaussian recursive filter [4, 2].

We tested many configurations of the edge detector on these images including the classic NMS method on squared gradient norm using linear interpolation with no subpixelic approximation and our method on either *real* or *squared* gradient norm using either *linear* or *quadratic* interpolation to find the values of the norm in the gradient direction.

Using the result of edge detection, we calculated the distance from the calculated pixels to the theoretical edge and the difference between the theoretical edge orientation and the orientation of the line joining two consecutive edge pixels. For each of these measurements and for a given configuration of the edge detector we calculated its mean, standard deviation, and maximum for different edge orientations, calculated over all the edge pixels in the image (i.e. between 256 and 350 pixels).

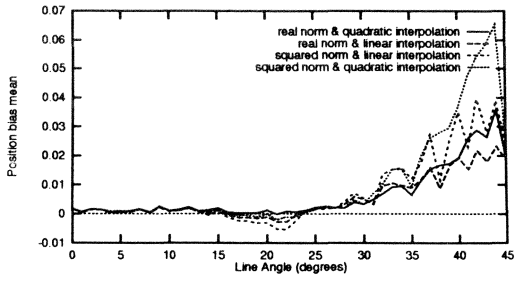

Figure 2: Position bias mean as a function of edge orientation.

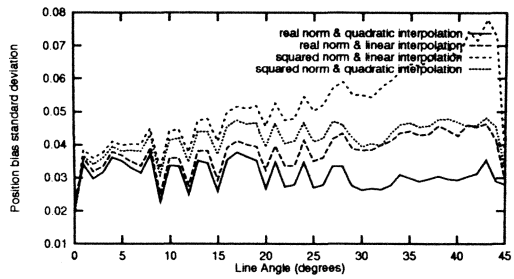

Figure 3: Standard deviation of the position bias as a function of edge orientation.

3.1 Edge position

The mean and standard deviation of the position bias for the different subpixelic edge detection methods are presented in Figures 2 and 3. The Gaussian derivative filter used for preprocessing had a σ of 1.5, which seemed to give the best results for edge position. For the classic NMS method, the maximum standard deviation of the position bias is 0.38 pixels at 41^0, and the mean bias varies between -0.08 and 0.16 pixels, far over the subpixelic methods. We can see with these figures that the best method for subpixelic edge detection is to use the real gradient norm as input and to estimate the gradient norm in the direction of the gradient using quadratic interpolation between the three neighboring points in the direction of the gradient. For a smaller computational cost, using the squared gradient norm with quadratic interpolation gives a rather good estimate.

We tested the robustness of the best method (real norm and quadratic approximation) by applying Gaussian noise to the image intensity for 512 x 512 images. The standard deviation of the edge position bias was calculated for different values of the intensity noise standard deviation and the Gaussian derivative filter standard de-

viation σ. $\sigma = 1.5$ gives the best results when there is not much noise but when the noise standard deviation is more than 40 the edge is almost always cut, so that bigger values of σ should be used. The same thing happens for $\sigma = 2.5$ when noise is more than 65.

3.2 Edge orientation

To prove that this edge detector gives excellent results we calculated the edge orientation in a very simple way, as the orientation of the line joining two consecutive edge points. This gives good results under zero noise conditions (the maximum angle bias mean for real norm and quadratic approximation is 0.35 degrees for a line orientation of 33 degrees and the maximum standard deviation is 3.2 degrees for an orientation of 17 degrees), but when there is noise in image intensity or when one needs more precision the data must be regularized. These results can be compared with figures 8.12 and 8.13 in [6].

To calculate a better value of the edge orientation we propose to take the orientation of the line which is the least squares approximation of n consecutive points ($n = 15$ in this case). This reduces significantly the standard deviation of the measures. We also noticed that using big values of σ can reduce the precision of angle estimation.

3.3 Other results

For the best edge configuration we calculated the edge position bias and orientation bias mean and standard deviation on the circle images. The edge position standard deviation is stable with the radius and equal to 0.035 pixels whereas the bias mean is always negative (-0.14 for $r = 4$, -0.02 for r=20 and -0.006 for $r = 100$) because of the shrinking due to Gaussian filtering and maybe also because of the edge detection method (it is well known that the NMS displaces the corners' positions), but this bias is still very low. We can notice that the standard deviation is in general smaller than those found for the straight edges. Curvatures can also be calculated with a good accuracy but the data needs to be regularized first.

We also present results on a real image of both the classic method and our method. One can see that some properties that were lost using the classic NMS method can be found using our method, such as the orientation of the letters and their shape, and the tangent along the two ellipses. The smooth edges are due again to the Gaussian derivative filtering.

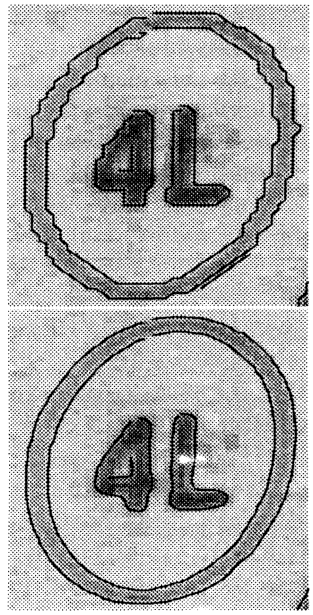

Figure 4: Results of the classical NMS algorithm (up) and of the subpixelic approximation (down) on a part of a real image.

4 Conclusion

In this article we presented an enhancement of the Non-Maxima Suppression edge detection method which gives us the edge position at a subpixelic precision. Since this method is very simple it can be implemented in a real-time vision system and should be used to increase the precision and reliability of the vision algorithms which use edges as input.

The result we got on a wide variety of synthetic and real images are very promising since they show that the precision of this edge detector is less than $\frac{1}{10}$ of a pixel. In the future we plan to use this edge detector to calculate some high degree differential properties of the curves such as euclidean, affine, or even projective curvature, and we will try to use it to enhance the precision of existing algorithms of stereo on curves.

References

[1] J.F. Canny. Finding Edges and Lines in Images. Technical Report AI-TR-720, Massachusets Institute of Technology, Artificial Intelligence Laboratory, June 1983.

[2] R. Deriche. Using Canny's Criteria to Derive a Recursively Implemented Optimal Edge Detector. *International Journal of Computer Vision*, 1(2):167–187, May 1987.

[3] R. Deriche. Using Canny's Criteria to Derive an Optimal Edge Detector Recursively Implemented. *The International Journal of Computer Vision*, 2:167–187, April 1987.

[4] R. Deriche. Recursively Implementing the Gaussian and Its Derivatives. Technical Report 1893, INRIA, Unité de Recherche Sophia-Antipolis, April 1993.

[5] O. Faugeras. *Three-dimensional Computer Vision: a geometric viewpoint*, chapter 5. MIT Press, 1993.

[6] R. M. Haralick and L. G. Shapiro. *Computer and Robot Vision*, volume 1. Addison-Wesley, 1992.

[7] Robert Haralick. Digital step edges from zero crossing of second directional derivatives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(1):58–68, January 1984.

[8] A. Huertas and G. Medioni. Detection of intensity changes with subpixel accuracy using laplacian-gaussian masks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(5):651–664, September 1986.

[9] Ali J. Tababai and O. Robert Mitchell. Edge location to subpixel values in digital imagery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(2):188–201, March 1984.

[10] T. Viéville and O.D. Faugeras. Robust and fast computation of edge characteristics in image sequences. *International Journal of Computer Vision*, 1993. in press.