

E-SPLINE SAMPLING FOR PRECISE AND ROBUST LINE-EDGE EXTRACTION

Akira Hirabayashi¹ and Pier-Luigi Dragotti²

¹ Dept. of Information Science and Engineering, Yamaguchi University
2-16-1, Tokiwadai, Ube 755-8611, Japan

² Dept. Electrical and Electronic Engineering, Imperial College London
Exhibition Road, London SW7-2AZ, United Kingdom

ABSTRACT

We propose a line-edge extraction algorithm using E-spline functions as a sampling kernel. Our method is capable of extracting line-edge parameters, including amplitude, orientation, and offset, not only at sub-pixel level but also exactly provided noiseless pixel values. Even in noisy scenario, simulation results show that the proposed method outperforms a similar one based around B-spline functions with gains in standard deviation of 1.86dB for the orientation and 9.64dB for the offset when SNR is 10dB. We also show by simulations that our method performs more precise than the Hough transform with faster computations.

Index Terms— Edge detection, edge extraction, E-spline functions, finite rate of innovation signals

1. INTRODUCTION

Line edge extraction is one of the most fundamental tasks in image processing and is used in many applications like object segmentation, registration, and super-resolution. The standard technique is that, first a method like Canny edge detector is applied at the pixel level [1, 2], and then line edges may be extracted by, for example, the Hough transform. This method is, however, neither precise nor computationally efficient.

To overcome this problem, Baboulaz *et al.* proposed a method which can exactly extract line-edges using B-spline functions as a sampling kernel, and they applied the extracted results to super-resolution [3]. Mathematical techniques used there are based on the sampling theory for signals with finite rate of innovations [4]. Since the method uses B-spline functions, however, it may suffer the following problem: B-spline is able to reproduce polynomials, and coefficients in the reproduction formulas grow in approximately polynomial orders. Since the coefficients play an important role in the method, the growth makes the method sensitive to noise.

Hence, in this paper, we propose a line-edge extraction algorithm based around E-spline functions [5]. An E-spline

of order P can reproduce exponential functions $e^{\alpha_p t}$, $p = 0, \dots, P$, by linear combinations of its shifted versions. Generally, coefficients used in the linear combinations grow in exponential orders. If the numbers α_p are purely imaginary, however, then their absolute values are constant, which is desirable for noise robustness. Such E-splines are especially called *trigonometric E-spline functions*. By using this function as sampling kernel, we propose a method which can extract line-edge parameters precisely and stably even in the presence of noise.

This paper is organized as follows. Section 2 provides mathematical preliminaries about the E-spline functions. Section 3 proposes the line-edge extraction method. We first describe a continuous line-edge image using amplitude, orientation, and offset parameters. This implies that the image is a signal with a finite rate of innovation. Then, the continuous image is sampled by E-spline functions. We derive formulas to obtain the three parameters from these samples. Further, by combining with Canny edge detector, we propose a line-edge extraction algorithm which uses only local area of an input image. Section 4 shows noise robustness of the proposed method compared to the counterpart which uses B-spline sampling. Section 5 concludes the paper.

2. E-SPLINE FUNCTIONS

Let $\vec{\alpha} = (\alpha_0, \alpha_1, \dots, \alpha_P)$ be a $(P+1)$ -dimensional complex vector, and $\beta_{\alpha_p}(t)$ be a function defined by

$$\beta_{\alpha_p}(t) = \begin{cases} e^{\alpha_p t} & (-0.5 \leq t < 0.5), \\ 0 & (t < -0.5, t \geq 0.5). \end{cases} \quad (1)$$

Then, an E-spline function $\beta_{\vec{\alpha}}(t)$ of order P is defined by the convolution of the $P+1$ functions $\beta_{\alpha_p}(t)$ as

$$\beta_{\vec{\alpha}}(t) = (\beta_{\alpha_0} * \beta_{\alpha_1} * \dots * \beta_{\alpha_P})(t). \quad (2)$$

Note that here we have used a centered one for the sake of image processing applications. If causality is necessary, one-sided E-spline should be considered [5].

Thanks to the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Young Scientists (B), 20700164, 2008.

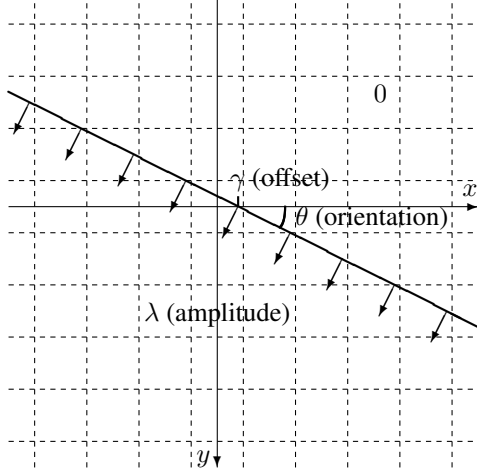


Fig. 1. Description parameters for a step line-edge. The xy coordinates are local ones whose origin is the center of the pixel detected as an edge. The grid shows sampled pixels.

The crucial characteristic of $\beta_{\vec{\alpha}}(t)$ is the ability to reproduce exponential functions by its integer-shifted versions as

$$\sum_{n=-\infty}^{\infty} c_n^{(p)} \beta_{\vec{\alpha}}(x - n) = e^{\alpha_p t}. \quad (3)$$

The coefficients $c_n^{(p)}$ are easily computed by

$$c_n^{(p)} = e^{n\alpha_p} / \left\{ \sum_{k=-P'}^{P'} e^{k\alpha_p} \beta_{\vec{\alpha}}(k) \right\}, \quad (4)$$

where P' is the maximum integer not exceeding $(P + 1)/2$. Eq. (4) implies that $c_n^{(p)}$ increases in an exponential order in general.

To overcome this problem one can choose the exponents α_p to be purely imaginary and equally spaced around the origin: $\vec{\alpha} = (\dots, -2i\omega_0, -i\omega_0, i\omega_0, 2\omega_0, \dots)$. This leads to the trigonometric E-spline. In this case, Eq. (4) implies that the absolute value of $c_n^{(p)}$ is $|c_0^{(p)}|$ irrespective of n . This is a desirable characteristic of the trigonometric E-spline from the viewpoints of stable computation and noise robustness. For example, the first order trigonometric E-spline with $\vec{\alpha} = (i\omega_0, -i\omega_0)$ is given by

$$\beta_{\vec{\alpha}}(t) = \begin{cases} \sin \omega_0(t + 1)/\omega_0 & (-1 \leq t < 0), \\ -\sin \omega_0(t - 1)/\omega_0 & (0 \leq t < 1), \\ 0 & (t \leq -1, t > 1). \end{cases}$$

As we can see from this example, trigonometric E-spline is a real valued function, whilst $c_n^{(p)}$ is a complex number.

3. LINE EDGE EXTRACTION

By using amplitude λ , orientation θ , and offset γ , which are defined as in Fig. 1, a step line-edge can be expressed as

$$f(x, y) = \lambda H(-x \sin \theta + y \cos \theta + \gamma \sin \theta), \quad (5)$$

where $H(t)$ is the unit step function whose value is 1 if $t \geq 0$ and 0 if $t < 0$. This continuous image is sampled by the integer-shifted version of $\psi(x)\psi(y)$ as

$$g[m, n] = \langle f(x, y), \psi(x - m)\psi(y - n) \rangle,$$

where $\psi(x)$ is a trigonometric E-spline function $\beta_{\vec{\alpha}}(t)$ with $P \geq 1$. Our task is to retrieve the three parameters λ , θ , and γ from the set of samples $g[m, n]$.

To this end, we track the techniques used in [3]. First, we compute a differentiated sample $d[m, n]$ which is given by $g[m + 1, n] - g[m, n]$. We then compute product-sum of $d[m, n]$ and coefficients $C_m^{(p)}$:

$$\tau_n^{(p)} = \sum_{m=-\infty}^{\infty} C_m^{(p)} d[m, n]. \quad (6)$$

The coefficients $C_m^{(p)}$ are determined so that they satisfy

$$\sum_{m=-\infty}^{\infty} C_m^{(p)} (\beta_{P+1} * \psi)(t - m) = e^{\alpha_p t} \quad (p = 0, 1, \dots, P+1),$$

where $\beta_{P+1}(t)$ is defined by Eq. (1) with $\alpha_{P+1} = 0$. The sampling kernel $\psi(t) = \beta_{\vec{\alpha}}(t)$ can originally produce $e^{\alpha_p t}$ with $p = 0$ up to P . The function, which is further convolved with $\beta_{P+1}(t)$, can produce $e^{\alpha_p t}$ up to $p = P + 1$. It should be noted that $C_m^{(p)}$ is different from $c_n^{(p)}$ in Eq. (3).

Let $\Psi(p, \theta)$ be

$$\Psi(p, \theta) = \int_{-\infty}^{\infty} \psi(t) e^{\alpha_p t / \tan \theta} dt.$$

Since $\alpha_{P+1} = 0$, $\Psi(P + 1, \theta)$ does not depend on θ . Hence, we denote it by $\Psi(P + 1) = \int_{-\infty}^{\infty} \psi(t) dt$.

By using $\tau_n^{(p)}$ and $\Psi(p, \theta)$, we can obtain λ , θ , and γ as follows:

$$\lambda = \frac{|\tau_n^{(P+1)}|}{\Psi(P + 1)}, \quad \tan \theta = \frac{\omega_0}{\angle(\tau_{n+1}^{(0)}/\tau_n^{(0)})},$$

$$\gamma = \frac{1}{\omega_0} \angle \left(\frac{\tau_n^{(0)} \text{sgn}(\tau_n^{(P+1)})}{\lambda \Psi(0, \theta)} \right) - \frac{n}{\tan \theta} + \frac{1}{2}, \quad (7)$$

where $\angle(z)$ is the phase angle of complex number z and $\text{sgn}(t)$ is the function whose value is 1 if $t > 0$, 0 if $t = 0$, and -1 if $t < 0$.

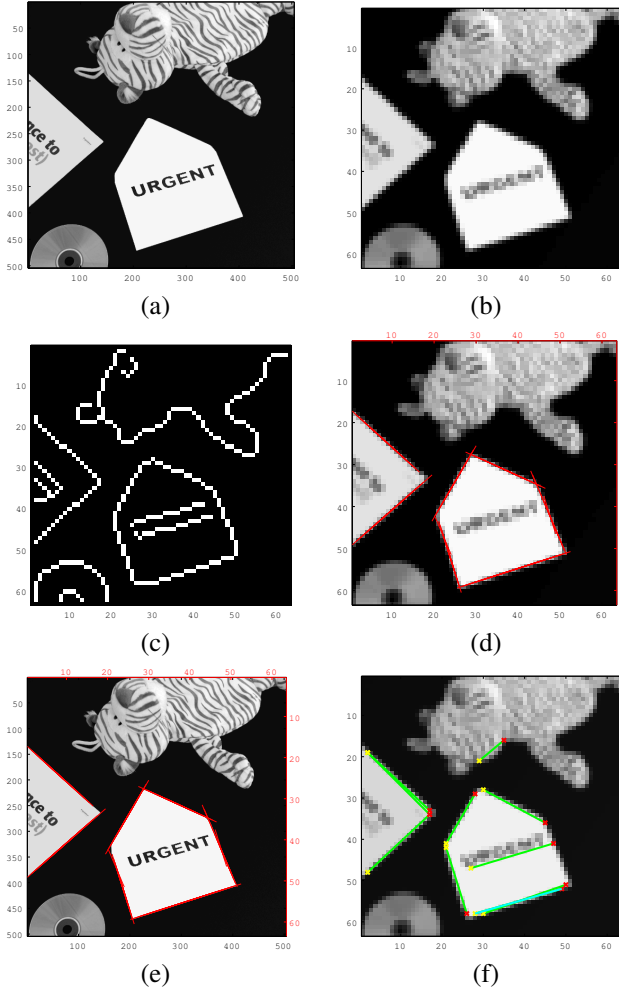


Fig. 2. Edge extraction results by the proposed algorithm: (a) original image, (b) downsampled image, (c) Canny edge detection results, (d) results by the proposed method on the downsampled image, (e) results by the proposed method on the original image, (f) results by the Hough transform.

Proof. It is shown in [6] that the differentiated samples $d[m, n]$ can be expressed as

$$d[m, n] = \left\langle \frac{\partial f(x, y)}{\partial x}, (\beta_{P+1} * \psi)(x - m - 0.5)\psi(y - n) \right\rangle.$$

This implies that $d[m, n]$ are the samples of the derivative of $f(x, y)$ along x with the modified kernel $(\beta_{P+1} * \psi)(x - 0.5)\psi(y)$. Eq. (5) implies that

$$\frac{\partial f(x, y)}{\partial x} = -\lambda \delta(-x \sin \theta + y \cos \theta + \gamma \sin \theta) \sin \theta,$$

where $\delta(t)$ is Dirac's delta function. Substituting these relations into Eq. (6) yields

$$\tau_n^{(p)} = \begin{cases} -\lambda e^{\alpha_p(\gamma+n/\tan\theta-0.5)}\Psi(p, \theta) & (\sin \theta > 0), \\ 0 & (\sin \theta = 0), \\ \lambda e^{\alpha_p(\gamma+n/\tan\theta-0.5)}\Psi(p, \theta) & (\sin \theta < 0). \end{cases}$$

Since $\text{sgn}(\sin \theta) = -\text{sgn}(\tau_n^{(P+1)})$, these relations are combined into

$$\tau_n^{(p)} = \text{sgn}(\tau_n^{(P+1)})\lambda e^{\alpha_p(\gamma+n/\tan\theta-0.5)}\Psi(p, \theta). \quad (8)$$

By setting $p = P + 1$ in Eq. (8), we have the first equation of Eq. (7) because $\alpha_{P+1} = 0$ and $\tau_n^{(P+1)}/\text{sgn}(\tau_n^{(P+1)}) = |\tau_n^{(P+1)}|$. Since $\alpha_0 = i\omega_0$, Eq. (8) implies that dividing $\tau_{n+1}^{(0)}$ by $\tau_n^{(0)}$ yields $\tau_{n+1}^{(0)}/\tau_n^{(0)} = e^{i\omega_0/\tan\theta}$, which results in the second equation in Eq. (7). Finally, solving Eq. (8) to γ yields the third equation in Eq. (7). \square

Eq. (7) uses $\tau_n^{(p)}$ and $\tau_{n+1}^{(p)}$, which are computed from pixels in lines of n and $n + 1$ in the sampled image, respectively. If these values are exact, then so are the three parameters. Since $f(x, y)$ is a step edge, $d[m, n]$ vanishes when pixels are away from the edge because $g[m, n]$ gets constant. Hence, it is sufficient to compute $\tau_n^{(p)}$ with pixels which are affected by the edge. For example, when the sampling kernel $\psi(t)$ is of order 1 and $\pi/4 \leq \theta < \pi/2$, only seven pixels around the edge are affected. Since we need the two lines n and $n + 1$, which may be one-pixel shifted each other because edge is normally slanted, eight pixels are necessary for $\tau_n^{(p)}$ and $\tau_{n+1}^{(p)}$ to be exact. When $0 < \theta < \pi/4$, the same argument can be applied by using Eq. (7) vertically. Hence, an 8×8 area, which is shown in Fig. 1, is suitable for exact extraction of the parameters. After the horizontal and vertical computations, their appropriate average should be the final result of the algorithm.

Based on the idea mentioned above, we arrive at the following algorithm: First, edge pixels are detected by a conventional detector like Canny operator. Then, for each pixel detected as an edge, the surrounding 8×8 pixel area is extracted, and Eq. (7) is applied to the area. To suppress extraction errors, similar edges are merged, while other edges are discarded.

Fig. 2 shows simulation results. In these simulations, $\omega_0 = \pi/M$, where $M = 32$ is the number of horizontal pixels. Figure (a) shows original image, and its downsampled image is shown in (b). Figure (c) shows the Canny edge detection results. Figures (d) and (e) show the extracted edges by the proposed method on the downsampled and original images, respectively. Figure (f) shows the results obtained by the Hough transform. We can see that the proposed method is more precise than the Hough transform. The computation of the proposed method was much faster than the Hough transform.

Let us note that, since the phase angle is $-\pi$ to π , Eq. (7) shows that $\tan \theta < -\omega_0/\pi$ or $\tan \theta > \omega_0/\pi$ and $-\pi/\omega_0 - n/\tan \theta + 0.5 < \gamma < \pi/\omega_0 - n/\tan \theta + 0.5$. The first restriction can be got rid of by applying Eq. (7) horizontally and vertically. The second restriction does not cause any problem because γ should stay a small value between -0.5 to 0.5 as shown in Fig. 1.

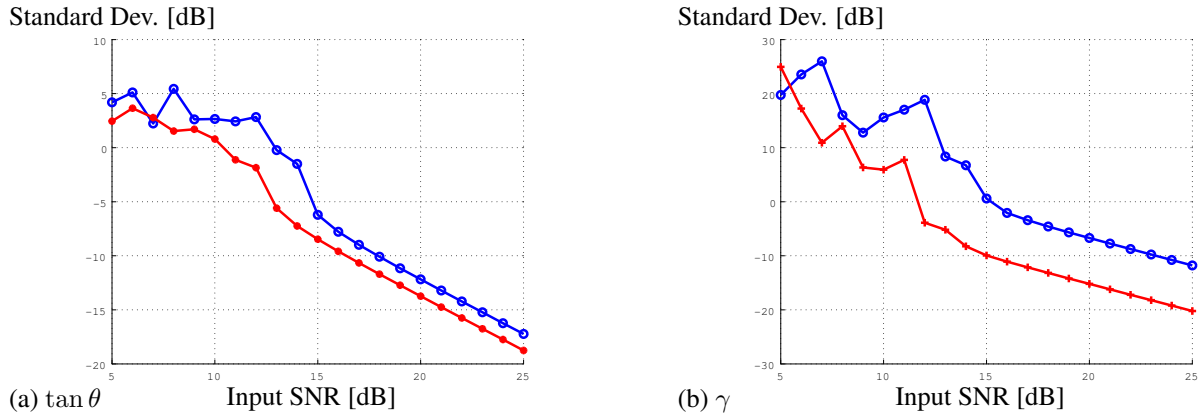


Fig. 3. Standard deviations [dB] of estimated results for (a) $\tan \theta$ and (b) γ in terms of signal-noise ratio from 5[dB] to 25[dB].

4. ERROR ROBUSTNESS EVALUATION

We evaluated robustness of the proposed line-edge extraction method when pixel values $g[m, n]$ are corrupted by noise as

$$y[m, n] = g[m, n] + \varepsilon[m, n],$$

where $\varepsilon[m, n]$ is independently and identically distributed Gaussian with standard deviation σ . We computed $\tau_n^{(p)}$ in Eq. (6) with $y[m, n]$ instead of $g[m, n]$, and obtained λ , θ , and γ by Eq. (7). The key advantage of the proposed method is that the coefficient $C_m^{(p)}$ is given by Eq. (4) with $\beta_{\bar{\alpha}}(t) = (\beta_{P+1} * \psi)(t)$, and therefore its absolute value stays constant. On the other hand, the coefficient for the similar method based around B-spline grows approximately linearly when $p = 1$. This makes difference in the robustness against the noise.

We used the standard deviation σ such that a signal-noise ratio (SNR) defined by $10 \log_{10}(\lambda/\sigma)$ was from 5dB to 25dB with step size 1dB. The same computations were executed for the method using B-spline functions. The results are shown in Fig. 3, in which (a) and (b) indicate the standard deviations of estimated values for $\tan \theta$ and γ , respectively. They are shown in dB in order to clearly show the difference for the large SNR values. In each figure, the lines with asterisks (red) and circles (blue) show the results obtained by the proposed method and the B-spline one, respectively. We can see that the proposed method outperformed the B-spline method on $\tan \theta$ except for SNR of 7[dB]. The gain was 0.92[dB] at least and more than 1.51[dB] after 15[dB] of SNR. As for γ , the proposed method outperformed the B-spline method except for SNR of 5[dB]. The gain was 2.04[dB] at least and more than 8.44[dB] after 15[dB] of SNR. These results show the robustness of the proposed method based around E-spline sampling.

5. CONCLUSION

We proposed a line-edge extraction algorithm using E-spline functions as a sampling kernel, and showed that line edge parameters are exactly extracted from noiseless pixel values. Even in noisy scenario, we showed by computer simulations that the proposed method is more robust than the similar method using B-spline with gains in standard deviation of 1.86dB for the orientation and 9.64dB for the offset when SNR is 10dB. The proposed method also outperformed the Hough transform with faster computations.

6. REFERENCES

- [1] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.
- [2] H. Trichili, M.-S. Bouhleb, N. Derbel, and L. Kamoun, "A survey and evaluation of edge detection operators application to medical images," in *2002 IEEE Int. Conf. Systems, Man and Cybernetics*, Oct. 2002.
- [3] L. Baboulaz and P.L. Dragotti, "Exact feature extraction using finite rate of innovation principles with an application to image super-resolution," *IEEE Trans. Image Processing*, vol. 18, no. 2, pp. 281–298, Feb. 2009.
- [4] M. Vetterli, P. Marziliano, and T. Blu, "Sampling signals with finite rate of innovation," *IEEE Trans. Signal Processing*, vol. 50, no. 6, pp. 1417–1428, June 2002.
- [5] M. Unser and T. Blu, "Cardinal exponential splines: Part I—Theory and filtering algorithms," *IEEE Trans. Signal Processing*, vol. 53, no. 4, pp. 1425–1438, April 2005.
- [6] P. L. Dragotti, M. Vetterli, and T. Blu, "Sampling moments and reconstructing signals of finite rate of innovation: Shannon meets Strang-Fix," *IEEE Trans. Signal Processing*, vol. 55, no. 5, pp. 1741–1757, May 2007.