Fast Defect Detection in Cloths with B-splines

PeiFeng Zeng and Tomio Hirata

***Abstract*--In this paper, an algorithm for fast defect detection in cloths based on fast B-spline transform is proposed. The algorithm can be applied to analyzing images at any integer scale for defect information. It can also be extended to a parallel algorithm that decomposes images at all desired scales simultaneously.**

A threshold parameter δ_{th} is used to adjust the defect detection reliability. A desired compromise between fault recognition and continuity of edge pixels can be reached by adjusting the value of δ_{th} properly.

***Index Terms*--B-spline transform, computer vision, fabric inspection, edge detection, defect detection**

I. Introduction

IN defect detection in cloths, images of cloths are sent to computer where they are processed to depress interference of texture of cloths and noise occurring in image capturing. Then features of defects are extracted for defect analysis. The defect detection algorithm discussed in this paper is designed for real-time processing, thus it is needed to execute the whole task in a short time period.

Defects in cloths correspond to singular pixels in their images. Exactly speaking, singular pixels exist in the edge areas of defects where the intensity of image changes sharply. Information about location and size of defects is obtained from their edge information.

Intensive research has been dedicated to edge detection in pattern recognition [1]-[3]. With three criteria in numerical optimizations of edge detection, Canny [1] derived an optimal edge operator that detects edge pixels from noisy data. Canny's edge operator extracts those pixels at which the convolution of the image and the first derivative of Gaussian function reach local maxima. Much calculation is necessary for the convolution, so Canny's edge operator is time-consuming especially when the width of the operator becomes wide. On the other hand, Sobel edge operator extracts edge information of pixels in an image by examining values of eight neighboring pixels. So the Sobel edge operator performs edge detection quickly while its noise performance is usually poor.

The edge operator discussed in this paper is used for defect detection in cloths. It is actually a Canny-like edge detector, but it is derived from B-splines instead of Gaussian function. Because B-splines converge to Gaussian function as the order of spline tends to infinity, and the cubic B-spline is near optimal in terms of time/frequency localization in the sense that its variance product is within 2% of the limit specified by the uncertainty principle [4], the output of our algorithm based on cubic B-spline well approximates signals derived by means of Gaussian function which is widely used in traditional scale-space analysis. Discrete wavelet transforms have been widely used in data compression and communications because there exist fast algorithms for discrete wavelet transform. To our knowledge, since no real symmetrical orthogonal wavelet exists except Haar wavelet, satisfying applications have not been reported for discrete wavelet transforms in pattern recognition. To meet the requirement of shift-invariance in pattern recognition, symmetrical wavelets are generally selected and dyadic wavelet transform or continuous wavelet transform is used for edge detection.

In this paper, B-splines are selected as the base functions for defect detection since their waveforms are symmetrical. We propose a fast algorithm for defect detection in cloths based on B-splines. The high efficiency of our algorithm is due to the fact that fast continuous wavelet transform can be implemented based on fast B-spline transforms.

Edge pixels are those pixels where the intensity of images varies sharply. So there exist large amount of high frequency components near edge pixels. When the first derivatives exist and are continuous for all pixels of an image, they reach local maxima at edge pixels.

Generally, images used for defect detection are contaminated with noise especially spike noise. Their derivatives are not continuous or even do not exist at some pixels. To obtain correct edge information, image should be smoothed to guarantee the existence of continuous derivatives at all pixels in the image.

Defect detection is implemented through multi-scale edge detection. Let m denote the scale at which images are decomposed. When m increases, more and more portion of signal that is contributed by texture of cloths and noise in images is smoothed away while singular information remains. Thus less interference remains in results at large scale m while detailed information is also smoothed away. We can determine the existence of defects and obtain their rough information through large-scale edge detection. When defects exist, their detailed information is obtained by small scale processing.

Fast B-spline transform is used to reduce the time for defect detection. Unlike discrete wavelet transform in which scale m changes from small values to large values orderly for signal decomposition, our algorithm can start defect detection at any scale. Furthermore, it can execute defect detection even at all useful scales simultaneously. The speed of defect detection will be improved greatly with such parallel processing.

The rest of the paper is organized as follows. In Section

Manuscript received April 24, 2001; revised December 9, 2001. This work was supported in part by “The Hori Information Science Promotion Foundation”.

The authors are with the School of Engineering, Nagoya University, Nagoya, Japan (e-mail: zengpf@hirata.nuee.nagoya-u.ac.jp; hirata@nuee.nagoya-u.ac.jp).

Publisher Item Identifier S 1682-0053(02)0108

II, some basic theories about B-splines are provided. In Section III, fast 1-D edge detection algorithm is discussed. This algorithm is expanded to 2-D space in Section IV. In Section V, the algorithm is applied to a sample cloth with defects in its image. Finally, some conclusions are given in Section VI.

II. Preliminaries

A. B-splines and B-spline transform

B-splines are defined as piecewise polynomials that satisfy some specific continuity constraints. The function

$$\beta^0(x) = \begin{cases} 1, & -\frac{1}{2} \le x \le \frac{1}{2} \\ 0, & elsewhere \end{cases} \tag{1}$$

is called as 0th-order B-spline. β^n denotes nth order B-spline and it is defined as

$$\beta^n(x) = \beta^0(x) * \beta^{n-1}(x) = \overbrace{\beta^0(x) * \cdots * \beta^0(x)}^{n+1}, \tag{2}$$

when $n \ge 1$. $\beta^n(x)$ is a function of class C^{n-1} (i.e., continuous function with continuous derivatives up to order $n-1$) and is equal to polynomials of degree n on each interval $[k, k+1), k \in Z$ when n is odd, and on each interval $[k - \frac{1}{2}, k + \frac{1}{2}), k \in Z$ when n is even. $\beta^n(x)$ can also be represented explicitly as follows [5].

$$\beta^n(x) := \sum_{j=0}^{n+1} \frac{(-1)^j}{n!} \binom{n+1}{j} \left(x + \frac{n+1}{2} - j \right)^n \cdot \mu\left(x + \frac{n+1}{2} - j \right), \quad (x \in R), \tag{3}$$

where $\mu(x)$ is the unit step function

$$\mu(x) = \begin{cases} 0, & x < 0 \\ 1, & x \ge 0 \end{cases}.$$

The first-derivative of n th-order B-spline

$$\frac{\partial \beta^n(x)}{\partial x} = \beta^{n-1}\left(x + \frac{1}{2} \right) - \beta^{n-1}\left(x - \frac{1}{2} \right) \tag{4}$$

equals to linear sum of two shifted $(n-1)$ th-order B-splines [6], [7].

$\beta_m^n(x)$ represents expanded B-spline by a factor of m, i.e., $\beta_m^n(x) = \beta^n(x/m)$. The function satisfies

$$\beta_m^n(x) = \frac{1}{m} \beta_m^{n-1}(x) * \beta_m^0(x) = \frac{1}{m^n} \overbrace{\beta_m^0(x) * \cdots * \beta_m^0(x)}^{n+1}. \tag{5}$$

In the case of discrete signal processing, $b^n(k)$ are used to represent discrete B-splines which meet with $b^n(k) = \beta^n(k), k \in Z$, i.e., $b^n(k)$ equal to sampling values of $\beta^n(x)$ at their knots when n is odd.

$b_m^n(k)$ are defined as the sampling values of expanded B-splines $\beta^n(x/m)$ divided by a factor of m, that is

$$b_m^n(k) = \tfrac{1}{m} \beta^n(k/m), \quad k \in Z. \tag{6}$$

Because $b_m^0(k) = \frac{1}{m}\{\overbrace{1,1,\cdots,1}^{m}\}$, we obtain

$$b_m^n(k) = (\overbrace{b_m^0 * b_m^0 * \cdots * b_m^0}^{n+1}) * b^n(k) \tag{7}$$

as m -scale equation for discrete B-splines.

Signals in computer processing are in discrete form. When discrete signals $\{f(k), k \in Z\}$ are interpolated with discrete B-splines b^n, we have

$$f(k) = \phi^n(k) = \sum_{i=-\infty}^{\infty} c(i) b^n(k-i). \tag{8}$$

In (8), $\{c(i)\}$ are called as spline coefficients, $f(k)$ is represented as the convolution of $\{c(i)\}$ and b^n. By taking the *z*-transform, we have

$$F(z) = B^n(z) C(z). \tag{9}$$

Equation (8) can be considered as a finite impulse response (FIR) operator $b^n(k)$ applied to $\{c(i)\}$ produces $\phi^n(k)$. So the spline coefficients $\{c(i)\}$ can be determined simply by inverse filtering, called direct B-spline transform

$$S^n(z) = B(z)^{-1} = \frac{1}{\sum b^n(k) z^{-k}}. \tag{10}$$

$S^n(z)$ is a recursive infinite impulse response (IIR) filter. It has been proved that it is stable in case n is odd for B-splines [8].

Indirect B-spline transform is used in the process of reconstructing discrete signal values from B-spline representation. When spline coefficients $\{c(i)\}$ are given, $f(k)$ can be recovered with indirect spline transform of (9), which is a finite impulse response (FIR) filter and stable.

III. 1-D Edge Detection

Edge detection is executed in x and y direction respectively, then the results are used to derive 2-D edge information. So we discuss 1-D edge detection before going into 2-D implementations.

In 1-D signals, edge points are those points where signals' intensity changes sharply. If the first derivatives of signals exist and are continuous for all points of the signals, they reach local maxima at edge points.

It has been proven that edges will not disappear while scale m increases in multi-scale edge detection [9]. At the same time, more and more noise is removed and the texture of cloths is smoothed away when m increases. For different purposes, edge should be detected at different scales. That is to say, if we just like to check the existence of edge points then detection at large scale is sufficient to obtain rough information of edges. On the other hand, small scale will be used to obtain detailed information about the existing edges. Therefore, multi-scale edge information is important for pattern recognition.

A. 1-D Multi-scale Edge Operator

Let $\phi(x)$ denote the smoothing function and $\phi_m(x)$ its dilation with factor m, i.e., $\phi_m(x) = \phi(x/m)$. Smoothing an input signal $f(x)$ at scale m with ϕ_m produces

$$g_m(x) = f * \phi_m(x). \tag{11}$$

Edge points at scale m are those points where $\partial g_m(x)/\partial x$ reaches local maxima. Because

$$\frac{\partial g_m(x)}{\partial x} = \frac{\partial f}{\partial x} * \phi_m(x) = f * \frac{\partial \phi_m(x)}{\partial x}, \tag{12}$$

edge detection is carried out by taking the convolution of an input signal $f(x)$ with edge operator $\varphi_m(x) = \partial \phi_m(x)/\partial x$.

B-splines are regarded as a kind of smoothing functions for the reason that their integrations are non-zero values. Let $\beta_m^n(x)$ be smoothing functions at scale m, i.e., $\phi_m(x) = \beta_m^n(x)$, we obtain

$$\varphi_m(x) = \partial \beta_m^n(x)/\partial x = \beta_m^{n-1}(x + \tfrac{m}{2}) - \beta_m^{n-1}(x - \tfrac{m}{2}) \tag{13}$$

as an edge operator derived from B-splines. It acts like the well-known Canny edge operator. Because splines are symmetrical, the edge operator is an anti-symmetrical one that finds edges in various scales at the same locations as edge points in input images.

B. Fast Multi-scale 1-D Edge Detecting Algorithm

To analyze functions $f(x)$ with an edge operator based on B-splines, $f(x)$ must be projected into B-spline space first. (8) gives the relation of discrete B-splines b^n with discrete signal $f(k)$. Because $f(k)$ equals to the sampling values of $f(x)$, the analog counterpart of (8) will be

$$f(x) = \sum_{i=-\infty}^{+\infty} c(i) \beta^{n_1}(x - i) \tag{14}$$

when $f(x)$ is projected to B-spline space of order n_1. With (14), (12) can be re-written

$$\frac{\partial g_m(x)}{\partial x} = \sum_{i=-\infty}^{+\infty} c(i) \beta^{n_1}(x - i) * \left(\beta_m^{n_2-1}(x + m/2) - \beta_m^{n_2-1}(x - m/2)\right) \tag{15}$$

when the edge operator is based on B-splines of order n_2. Using m-scale equation (7), we have

$$\beta_m^{n_2-1}(x \pm m/2) = m \sum_{j=-\infty}^{+\infty} b_m^{n_2-1}(j) \beta^{n_2-1}(x - j \pm m/2) \tag{16}$$

as the analog form of m-scale equation.

$$\begin{aligned} &\beta_m^{n_2-1}(x \pm m/2) * \beta^{n_1}(x - i) \\ &= \left(m \sum_{j=-\infty}^{+\infty} b_m^{n_2-1}(j) \beta^{n_2-1}(x - j \pm m/2) \right) * \beta^{n_1}(x - i) \\ &= m \sum_{j=-\infty}^{+\infty} b_m^{n_2-1}(j) \beta^{n_1+n_2}(x - j \pm m/2). \end{aligned} \tag{17}$$

Substituting (17) into (15) gives

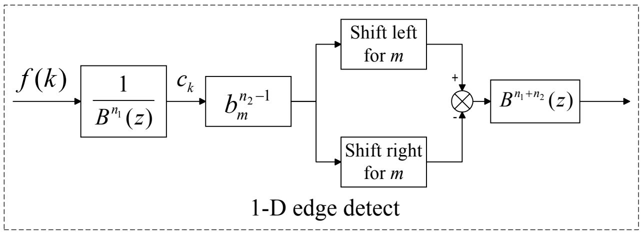

Fig. 1. Fast 1-D Edge Detection Algorithm

$$\begin{aligned} \frac{\partial g_m(x)}{\partial x} &= m \sum_{i=-\infty}^{+\infty} c(i) \sum_{j=-\infty}^{+\infty} b_m^{n_2-1}(j) \\ &\quad \beta^{n_1+n_2}(x - i - j + m/2) - \\ &\quad \beta^{n_1+n_2}(x - i - j - m/2) \\ &= m\left(b_m^{n_2-1} * c * \beta^{n_1+n_2}(x + m/2) \right. \\ &\quad \left. - b_m^{n_2-1} * c * \beta^{n_1+n_2}(x - m/2)\right) \end{aligned} \tag{18}$$

When the input signal $f(x)$ is sampled as $f(k)$, (18) is re-written as

$$\begin{aligned} g'_m(k) &= m\left(b_m^{n_2-1} * c * b^{n_1+n_2}(k + m/2) \right. \\ &\quad \left. - b_m^{n_2-1} * c * b^{n_1+n_2}(k - m/2)\right). \end{aligned} \tag{19}$$

The algorithm implied by (19) is shown in Fig. 1 as fast 1-D edge detection algorithm.

In Fig. 1, spline coefficients $c(k)$ are derived from the input signals $f(k)$ passing through direct B-spline transform $1/B^{n_1}(z)$. Then $c(k)$ is processed by the convolution with $b_m^{n_2-1}$ which is realized by repeating moving average of width m for n_2 times. The results are shifted left and right for m dots. The two shifted results are subtracted and the $(n_1 + n_2)$ th order indirect B-spline transform is performed to get the result of 1-D edge detection.

In the above processing, when both $n_1 = 3$ and $n_2 = 3$, there are 5 operations (multiple or divide) per-dots in direct B-spline transform and 6 operations in indirect B-spline transform. So the algorithm executes edge detection quickly.

IV. 2-D Edge Detection

With tensor-product [10], we can get the smoothing function in 2-D space at scale m

$$\phi_m(x, y) = \phi_m(x)\phi_m(y), \tag{20}$$

when 1-D smoothing function $\phi_m(x) = \beta_m^n(x)$. Edge pixels in an image are defined as pixels where the intensity changes sharply along some direction. If the first derivatives exist, they reach to local maxima at edge pixels along one direction in an image.

Canny's edge operator detects edge pixels in an image $f(x, y)$ by calculating the module of the gradient vector

$$\vec{\nabla} f = \begin{pmatrix} \partial f/\partial x \\ \partial f/\partial y \end{pmatrix}. \tag{21}$$

Let $\vec{n} = (a, b)$ be an unit vector in plane (x, y), the partial derivatives of $f(x, y)$ in the direction of $\vec{n}$ are calculated with an inner product of $\vec{\nabla} f$ and $\vec{n}$,

$$\frac{\partial f}{\partial \vec{n}} = \vec{\nabla} f \cdot \vec{n} = \frac{\partial f}{\partial x} a + \frac{\partial f}{\partial y} b \,. \tag{22}$$

when $\vec{n}$ is parallel to the direction of maximum change of $f(x, y)$, the absolute value of (22) reaches local maxima.

A pixel (x_0, y_0) is defined as an edge pixel if the module of $\vec{\nabla} f$ is locally maximum at (x_0, y_0) when (x, y) varies in a one-dimensional neighborhood of (x_0, y_0) that is parallel to $\vec{\nabla} f$.

Local maxima at scale m along any direction can be determined when derivatives of $f(x, y)$ at scale m along directions x and y

$$\begin{cases} \dfrac{\partial f}{\partial x}\Big|_m = f * \left(\dfrac{\partial \beta_m^n(x)}{\partial x} \beta_m^n(y) \right) \\ \dfrac{\partial f}{\partial y}\Big|_m = f * \left(\beta_m^n(x) \dfrac{\partial \beta_m^n(y)}{\partial y} \right) \end{cases} \tag{23}$$

are given. In Fig. 2, to speed up 2-D edge detection, we search local maxima along direction x, direction y and diagonal direction only instead of calculating module of $\vec{\nabla} f$ for local maxima in all directions.

In Fig. 2, Results from edge detection along x and y directions are sent into "Magnitude and phase detection" processing unit, where their magnitude are compared to determine the gradient vector for maximum changing of module of edge detection outputs.

V. Experimental Results

Images of cloths to be inspected are taken by cameras. To speedup the inspection, each image should cover as large the area as possible. In our experiments, the resolution of images of cloths is set such that the width of every yarn takes about two pixels.

There are two phases in defect detection; one is to determine the location of defects, the other is to get the detailed information of defects such as area, form, etc.

When multi-scale edge detection is carried out with the algorithm discussed in this paper, there is no pre-determined order for the changing of scale m. On the other hand, we know from practice that most edge information can be obtained at scales $m \le 3$ when the algorithm is based on cubic B-spline. At scale $m = 3$, the background texture is attenuated while the defects are accentuated well. So edge detection is performed at scale $m = 3$ directly to get the existence information of defect. If only exist defects, edge detection at small m is needed to obtain detailed information of existing defects.

The sample image of cloth is a 256-grayscale image with the size of 512×512 pixels as shown in Fig. 3. There are defects in the left part of the sample image. They are caused be machine oil spoils.

Pixels with local maxima of first derivatives in images of cloths largely come from texture output statistically. The distribution, which is represented as $k(d)$, of the value of

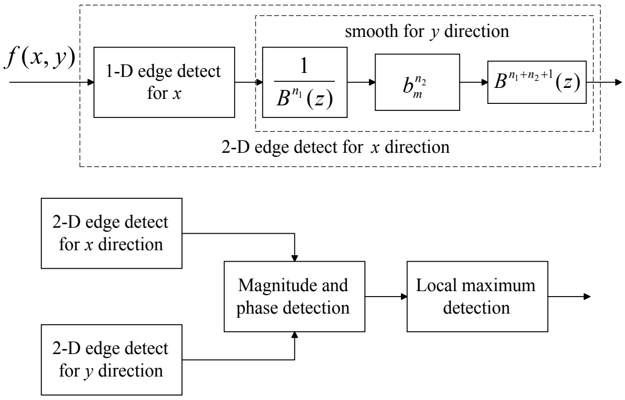

Fig. 2. 2-D Edge Detecting Algorithm

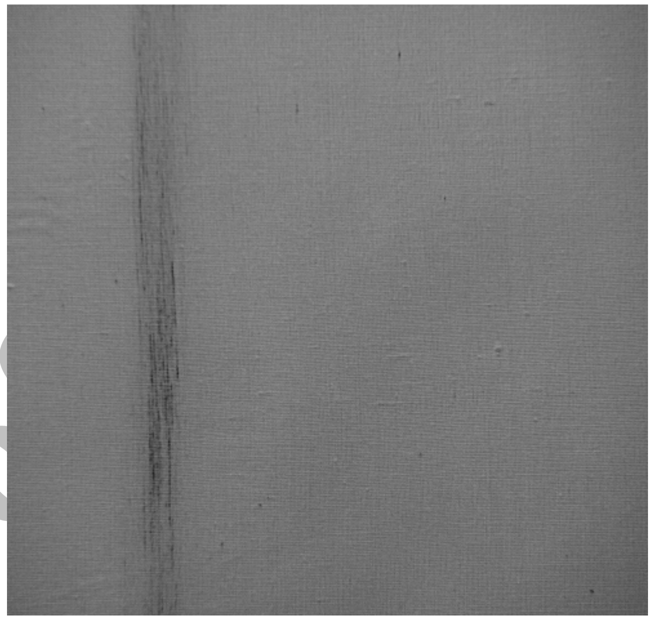

Fig. 3. Sample image of cloth with defects and handwriting characters.

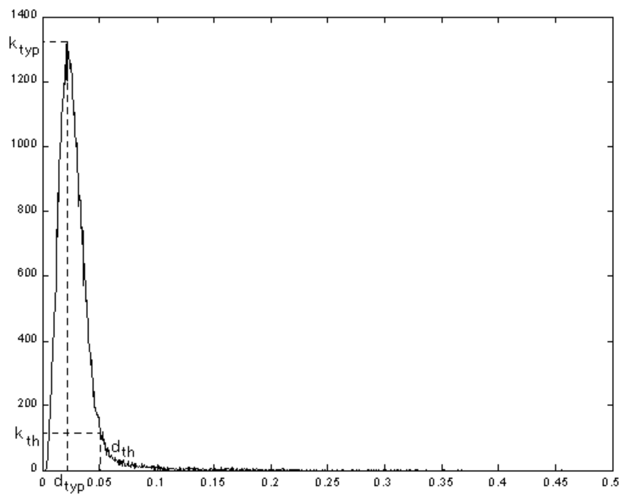

Fig. 4. The distribution of absolute values for local maxima of first derivatives of sample image at scale $m = 3$.

first derivatives of these pixels in sample image at scale $m = 3$ is shown in Fig. 4. k_{typ} is the maximum value of the distribution curve, its corresponding edge value equals to d_{typ}, the typical output of texture. If we set the reliability threshold as δ_{th}, which gives

$$k(d_{th}) = k_{th} = k_{typ} \cdot \delta_{th} \tag{24}$$

and $d \le d_{th}$ as the range in which the outputs of edge detector are regarded as phantom edges produced from texture of cloths.

Defect detection is processed at scale $m = 3$ for the sample image in Fig. 3. When $\delta_{th} = 0.05$ is selected, the result of defect detection is shown in Fig. 5; Fig. 6 is the result with $\delta_{th} = 0.02$. The number of phantom edge pixels in the result of defect detection reduces when the threshold parameter δ_{th} becomes smaller, while the continuity of obtained edge pixels of defects gets worse. Changing the threshold parameter δ_{th} we can set an appropriate compromise between the number of phantom edge pixels and the continuity of edge pixels.

Figs. 7 and 8 are the results of the edge operator applied to the sample image at scale $m = 1$ with $\delta_{th} = 0.05$ and $\delta_{th} = 0.02$, respectively. We have a conclusion that it is difficult to distinguish edge pixels from noise when m takes smaller values. To depress interference that comes from texture of cloths small δ_{th} is preferred. That will deteriorate the continuity of edge pixels of defects (see Fig. 8).

In production stage the algorithm needs to be trained before fabric inspection. A defect-free sample is used and the parameter δ_{th} is adjusted such that no defect is detected. The inspection is applied upon the non-overlapping frames of images that are taken from the same fabric. Because the texture is repeating and the lightening is unchanged, the value of δ_{th} is needless to be adjusted in the frames of images for inspection.

Figs. 9 and 10 are results obtained by applying Sobel edge operator to the sample image. Because the edge information of a pixel merely rely on surrounding eight pixels, its noise performance is even worse compared to the results in Figs. 7 and 8. Furthermore, the distribution of output intensity of edge pixels from Sobel edge operator that is shown in Fig. 10 is more separated and irregular. This makes it difficult to determine the value of δ_{th} to distinguish the output of edge pixels from noise.

VI. Conclusions and Future Work

We have proposed a fast defect detection algorithm based on fast B-spline transform.

The main objective of defect detection is to acquire the existence, location, and size information about defects in images. Existence and location information at large scale and detailed information at small scale can be obtained with multi-scale edge detection. Unlike discrete wavelet transform in which scales for signal decomposition must change from small values to large values in pre-determined order, our algorithm runs at large scales firstly. It is also possible that our algorithm executes at all necessary scales simultaneously on a parallel computing model.

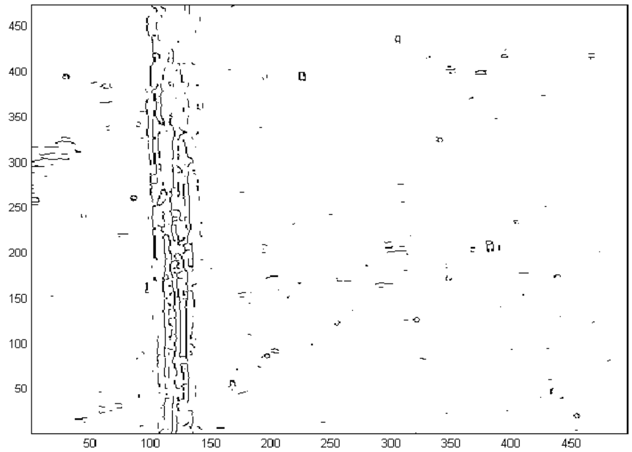

Fig. 5. The Result of edge detecting at scale $m = 3$, with $\delta_{th} = 0.05$.

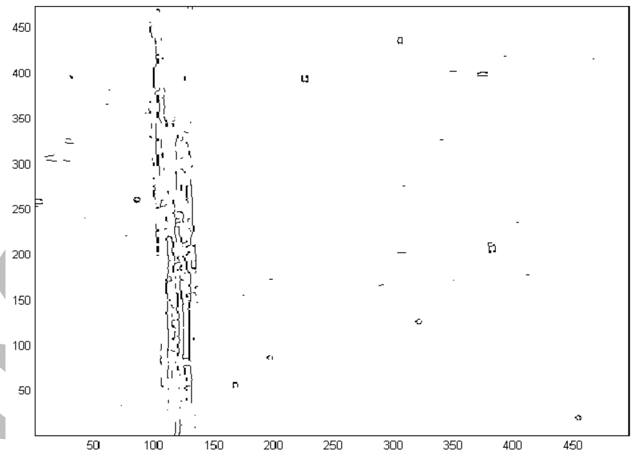

Fig. 6. The Result of edge detecting at scale $m = 3$, with $\delta_{th} = 0.02$.

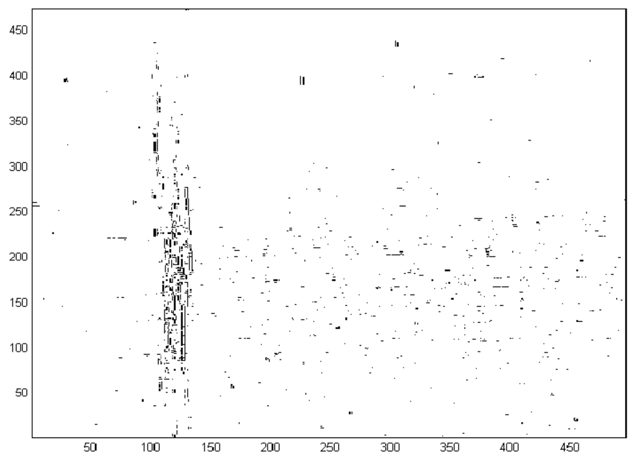

Fig. 7. Result of edge detecting at scale $m = 1$, with $\delta_{th} = 0.05$.

Changing the adjusting parameter δ_{th}, desired compromise can be reached between the minimization of erroneous recognition and the continuity of edge pixels of defects.

In our future developments, the automatic selection of δ_{th} will be studied. As we expected, the detect result will be deteriorated when the image noise level is high. So multi-scale detection and correlative de-noising should be studied to reduce the false alarm rate.

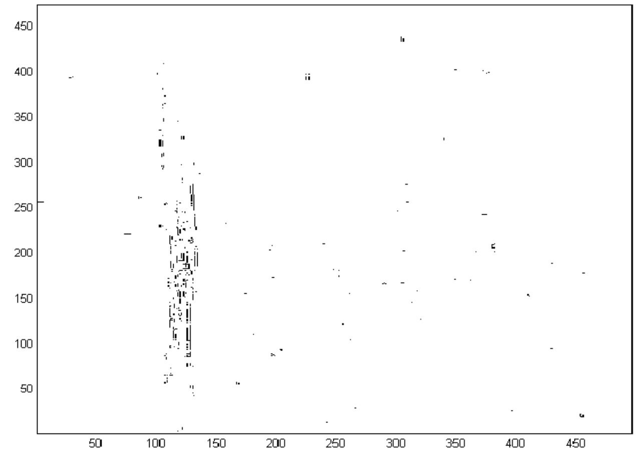

Fig. 8. Result of edge detecting at scale $m = 1$, with $\delta_{th} = 0.02$.

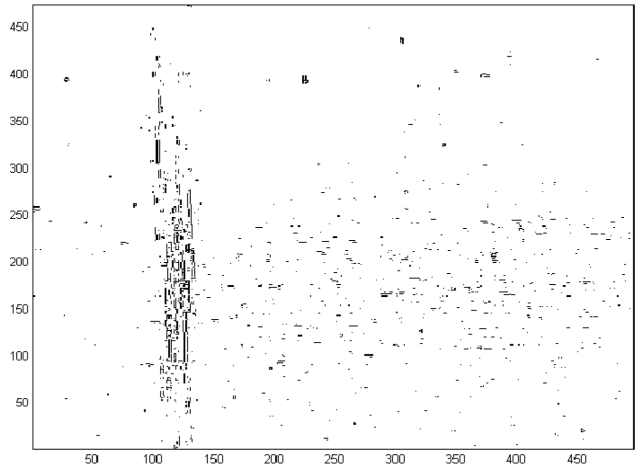

Fig. 9. Result of Sobel edge operator at $\delta_{th} = 0.05$.

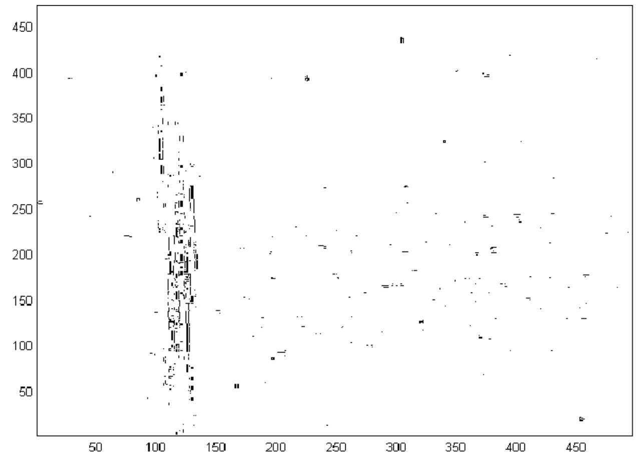

Fig. 10. Result of Sobel edge operator at $\delta_{th} = 0.02$.

ACKNOWLEDGMENT

The authors thank editors for their insightful suggestions and anonymous reviewers for their critical comments, which help to improve the presentation of this paper.

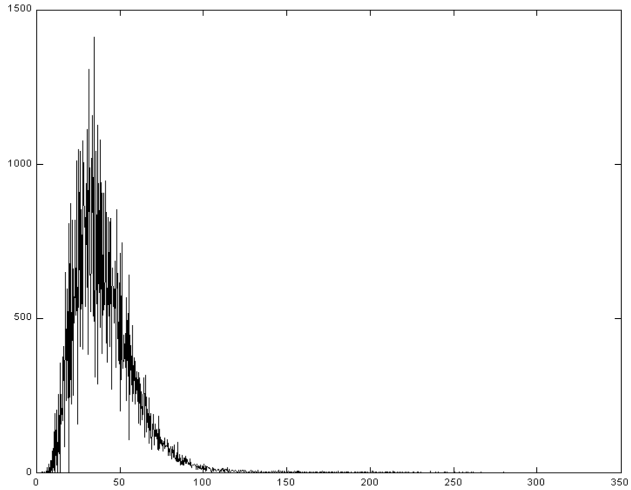

Fig. 11. The distribution of absolute values for local maximum first derivatives of the output signals of Sobel edge operator.

REFERENCES

[1] J. Canny, "A computational approach to edge detection," *IEEE Trans. PAMI*, vol. 8, no. 6, pp. 679-698, November 1986.

[2] J. J. Clark, "Authenticating edges produced by zero-crossing algorithms," *IEEE Trans. PAMI*, vol. 11, no. 1, pp. 43-57, November 1986.

[3] V. S. Nalwa and T. O. Binford, "On detecting edges," *IEEE Trans. PAMI*, vol. 8, no. 6, pp. 699-714, November 1986.

0 M. Unser, A. Aldroubi, and M. Eden, "On the asymptotic convergence of B-spline Wavelets to Gabor functions," *IEEE Trans. Inform. Theory*, vol. 38, no. 2, pp. 864-872, March 1992.

[4] I. J. Schoenberg, "Cardinal interpolation and spline functions," *J. Approximation Theory*, vol. 2, pp. 167-206, 1969.

[5] M. Unser, A. Aldroubi, and M. Eden, "B-spline signal processing: Part I-theory," *IEEE Trans. Signal Processing*, vol. 41, no. 2, pp. 821-833, February 1993.

[6] M. Unser, A. Aldroubi, and M. Eden, "B-spline signal processing: Part II-efficient design and applications," *IEEE Trans. Signal Processing*, vol. 41, no. 2, pp. 834-848, February 1993.

[7] M. Unser, A. Aldroubi, and M. Eden, "Fast B-spline transforms for continuous image representation and interpolation," *IEEE Trans. PAMI*, vol. 13, no. 3, pp. 277-285, March 1991.

[8] Stephone Mallat, *A Wavelet Tour of Signal Processing*. Academic Press, 1998.

[9] L. L. Schumaker, *Spline Functions: Basic Theory*. New York: Wiley, 1981.

PeiFeng Zeng received B.E. and M.E. from Southeast University, in China in 1984 and 1990, respectively. He worked at DongHua University from 1984 to 1987 and from 1990 to 1998. He is currently a Ph.D. candidate at the Graduate School of Electronic Engineering, Nagoya University.

Tomio Hirata received B.Sc., M.Sc. and Ph.D. in Computer Science, all from Tohoku University in 1976, 1978, and 1981, respectively. He is currently a Professor in the Department of Electronics, Nagoya University. His research interests include graph algorithms, approximation algorithms, and image processing.