

<http://ansinet.com/itj>

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSI*net*

Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Object Edge Smoothing by Wavelet Transformation

Zheng Xiaodong, Huang Xinghan and Wang Ming
Department of Control Science and Engineering,
Huazhong University of Science and Technology, Wuhan, 430074, China

Abstract: In order to solve problem that most 2D (Two-dimensional) shape representations don't fit for wavelet transformation processing, a new method of 2D shape representation has been put forward. It stores object borderline pixels coordinate value (x_i, y_i) into two arrays $x = X[i]$; $y = Y[i]$ by the sequence of object borderline tracking result, object boundary has been represented by this two array. It changes a 2D image problem into two single dimension arrays' problem and can be processed by wavelet transformation. Then make full use of the ability of time and frequency localization of wavelet transformation and find characters both in time and frequency. The boundaries leaps can be distinguish from noise. This technique has been applied in workpiece boundary noise reducing. And it shows that noise can be eliminated and boundaries corners are remained at same time.

Key words: Shape representation, binary wavelet, image processing, boundary tracking

INTRODUCTION

After edge detection or threshold segmentation we obtain the outline or the region shape of the object. It is very important to process the shape of 2D images for machinery recognize. Wavelet transforms is a kind of powerful tool that can analyze data both in time and frequency. It can adjust time and frequency window according to the characters of frequency and it is a self-adaptation method of time and frequency localization^[1,2]. But it can not process 2D shape directly. It is obvious that if object boundaries are intent to be processed by wavelet transform a new method of 2D shape representation is needed which can be processed by wavelet transform.

Two-dimensional shape representation includes two aspects, one is region-based representation another is boundary representation^[3,4]. A classic region-based representation method is to present a finitude measure collections about the object shape by invariant moments^[4]. Boundary representations include chain code, Fourier shape representation operators, chord distributes^[3], B-spline fitting methods etc.^[4,5]. And basic element (simple functions) fitting^[4,6] is a important boundary representation method. It segments object borderline first and then fit it by given functions which usually are arc and beeline. But all these representations can not be processed by wavelet transform directly. In order to facilitate the processing of object boundary such as filtering etc by wavelet transformation, a new representation of 2D shape is presented. The basic technique is to tract the boundary first, second stores the object borderline pixels' coordinates (x_i, y_i) into two arrays $x = X[i]$; $y = Y[i]$ by the sequence of object borderline tracking result. $X[i]$ is the abscissa value,

$Y[i]$ is the ordinate value, i represent array sequence number.

$X[i]$, $Y[i]$ can be regard as two functions with independent variable i : $x = X[i]$, $y = Y[i]$. Due to the various and simple processing methods of functions of one variable, we can easily manipulate object boundary by processing $x = X[i]$, $y = Y[i]$. For example smoothing functions $x = X[i]$ and $y = Y[i]$ can also smooth object borderline. We can enhance the high frequency to strengthen the corner. We can also process borderline by other filters.

For reasons of object borderline image being changed into two functions, it needn't segment borderline when fitting this two functions so long as enough parameters were given.

The property of this representation and the common processing of object boundary image by this representation method are introduced.

BASIC IDEA

To apply wavelet to 2D shape processing the processing object should be translated into function form: $y = f(x)$. It means that the function value y is fixed while the x is fixed. That is to say every x has only one function value. Condition (1) is Defined for a given point array $D = \{d_i(x_i, y_i), i = 1 \dots n\}$: The projection on x-axis of curve which been made of all connections of the points in sequence never wraps. If point array satisfied condition (1) then we can represent it by function form: $y = f(x)$. That means methods relevant to function processing including wavelet transform can be applied to object borderline processing.

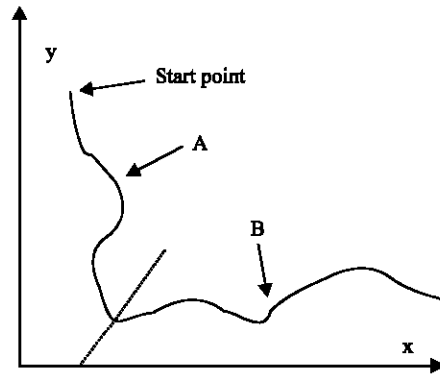
2D object borderlines are composed of curves and all these curves are made up of pixels bordering upon each other in original object borderline images. The biggest difference between function and object borderlines is that the functions satisfy condition (1), but the object borderlines do not in most occasions. Traditional way to represent object borderlines by functions is to regard borderlines as the combination of some basic element (simple functions). So the object borderline needs to be segmented in order to satisfy condition (1) in every part. But this study doesn't research into what kind of basic elements should be applied to describe the borderline; it tries to solve the problem that the object borderlines cannot be represent by functions. The method that has been put forward regards the object borderline as a course of tracking it. It begin with $t = 1$ and end in $t = n$. As independent variable t increasing each step, every value of x and y are stored. Each t corresponds to only one x and one y . So both $x = X(t)$, $y = Y(t)$ satisfy condition (1) and they contain all the information about object borderline. This representation method not only greatly lower down the store capacity but also make it convenient for analysis and processing.

If continuation is needed when analyzing, both of the two functions can be regard as functions with an n (the pixels number of borderline) cycle.

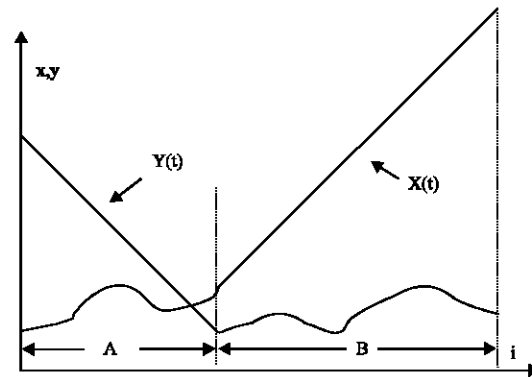
PROPERTY OF $X[n]$, $Y[n]$

From the tracking course we know that when next pixel is 4 neighborhood of current pixel the distance between two pixels is 1, if next pixel is a 8 neighborhood but not a 4 neighborhood the distance is $\sqrt{2}$ and the abscissa x and ordinate y increase 1 decrease 1 or not change in each step. Suppose that (P_l, P_m) , is part of a borderline, l, m is sequence number, if the slop k satisfies $|k| \leq 1$ at every pixel then borderline extends mainly in x -axis direction. This indicates that the moving speed in x -axis direction is faster than in y -axis direction. So the value x changes every step without any stop during the tracking course, but value y has the possibility of stop. It shows that $x = X[t]$ is a beeline with slop of 1 or -1 and $y = Y[t]$ is just the translation of the original borderline. The same, when $|k| \geq 1$, $Y = r[t]$ is a beeline with slop of 1 or -1 and $x = X[t]$ is just the translation and 90 degrees rotation of the original borderline.

Figure 1a is part of object borderline, Fig. 1b is $x = X(t)$, $y = Y(t)$. Dashed in Fig. 1a separates object borderline into two parts, $|k| \geq 1$ in part A, from Fig. 1b we can see that $Y(t)$ is a beeline in part A, $X(t)$ is the translation and 90 degrees rotation of part A. $|k| \leq 1$ in Fig. 1a in part B, the $X(t)$ in Fig. 1b in part B is a beeline and $Y(t)$ is the translation of part B in Fig. 1a. So the representation can be considered that when borderline extend mainly in x -axis direction, the borderline is described by $y = f(x)$ x is independent,



(a) 2D curve



(b) $x = X(t)$, $y = Y(t)$ functions

Fig. 1: 2D curve and its functions of $x = X(t)$, $y = Y(t)$

when borderline extend mainly in y -axis direction, the borderline is described by $x = f(y)$ y is independent.

Another important property is that the slop of $x = X[t]$, $y = Y[t]$ satisfies $|k'| \leq 1$. That because number $i+1$ pixel is a 8 neighborhood of number i pixel, the value of $X[i+1]-X[i]$ equals $-1, 1$ or 0 , so $|X[i+1]-X[i]| \leq 1$, viz., $|dx/dt| \leq 1$. And $|dy/dt| \leq 1$ for the same reason.

PROCESSING STEP

The whole image process by this 2D shape representation consists of 4 stages:

- Detect the object borderline;
- Create arrays $X[n]$, $Y[n]$, track and record x , y coordinate into arrays;
- Process and analyses $X[n]$, $Y[n]$. For example noise reduction;
- Format $X[n]$, $Y[n]$.

Application of this 2D shape representation is introduced and a example of Boundary noise reduction is given.

Edge detection: A fundamental condition of 2D shape representation is to obtain precision object borderline. The borderline can be tracked should satisfy conditions bellows: If S is pixels collection of edge detection; the pixels collection that can be tracked named T is a subclass of S and it is also a connected component of S, all pixels except end pixels have and only have two neighboring pixels (end pixels have and only have one neighboring pixel). One of common ways of edge detection is to image threshold segment first and then find out all the pixels in object boundary. Another is edge enhancement first then threshold segmentation. In general the result of edge detection can't be tracked directly, so the image after edge detection needs additional processing. A skeletonizing algorithm should be applied. A method presented by Naccache and Shinghal^[7], Xia^[3] satisfies condition to avoid boundary disconnection caused by noise, some noise reduction algorithms such as morphology operations and connection strategies should be applied.

Tracking curve: Create a pair of arrays X[n], Y[n], then track the curve, store every pixel coordinates into X[i], Y[i], i is the sequence number. If there are more than one object in one image, a series of arrays X_m[n], Y_m[n] need to be constructed, each pair for one borderline. After tracking process, all the boundary information has been stored in arrays X[n], Y[n].

Noise reducing: X[n], Y[n] are the boundary pixel projections on x-axis and y-axis. when tracking borderline, X(t), Y(t) move along the x-axis and y-axis respectively. If object boundary isn't smooth, the X(t), Y(t) will move to-and-fro frequently. Reflection on the functions x = X(t), y = Y(t) is the mightiness of high-frequency components.

Object boundary noise reducing can be implemented by restraining the high-frequency components of functions x = X(t), y = Y(t). By wavelet analysis we can reduce x = X(t), y = Y(t) 's noise and keep the sharp inflections of object boundary at the same time.

After wavelet decomposition the information of boundary approximate shape is mainly included in low-frequency components, Noise and burrs are included in high-frequency components. But the object boundary leaps are reflected in high-frequency components. To reduce noise and conserve boundary leaps should find out the difference between them. Now suppose that noise amplitude is much lower than boundary leaps. That supposal is acceptable, because higher amplitude part can be treat as part of boundary. The process of noise reducing is divided into three steps: 1) Wavelet decomposition. 2) Get rid of noise. First revert high-frequency components, second threshold segment the result and third restrain or eliminate the part lower than threshold. 3) Revert

low-frequency components and then revert whole image with processed high-frequency part. Wavelet decomposition algorithm as follows:

$$W_f(a, b) = \int f(t) \overline{\psi_{ab}}(t) dt \quad (1)$$

Wavelet function:

$$\psi_{ab}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right),$$

Reversion algorithm:

$$f(t) = \frac{1}{C_\psi} \iint_{\mathbb{R}^2} W_f(a, b) \psi_{ab}(t) \frac{da}{a^2} db \quad (2)$$

There into
$$C_\psi = \int \frac{|\widehat{\psi}(\omega)|^2}{\omega} d\omega < \infty$$

In order to make wavelet transform algorithm more efficient, binary wavelet transform is applied. The flex step factor a in Eq. 1 and 2 is changed into 2^j. Binary wavelet transform algorithm only disperses the scale factor and the shift factor keeps continuity. Binary wavelet transform has zoom lens ability and continuity in time domain at the same time. Binary wavelet function is:

$$\psi_{jb}(t) = 2^{j/2} \psi(2^j(t-b)) \quad (3)$$

Final wavelet decomposition algorithm:

$$W_f(j, b) = 2^{j/2} \int f(t) \overline{\psi}(2^j(t-b)) dt \quad (4)$$

Reversion algorithm:

$$f(t) = \sum_{j \in \mathbb{Z}} 2^j \int W_f(j, b) \overline{\psi}(2^j(t-b)) db \quad (5)$$

The $\widehat{\psi}(x)$ is called rebuild wavelet and it satisfies:

$$\widehat{\widehat{\psi}}(\omega) = \frac{\widehat{\psi}(\omega)}{\sum_{j \in \mathbb{Z}} |\widehat{\psi}(2^j \omega)|^2}$$

Mexico strawhat function is selected as wavelet function. And the calculation result shows that its rebuild function is very similar to its-self except for amplitude.

Format X[n], Y[n]: The data after processing named X'ⁿ[n] and Y'ⁿ[n] doesn't satisfy properties presented

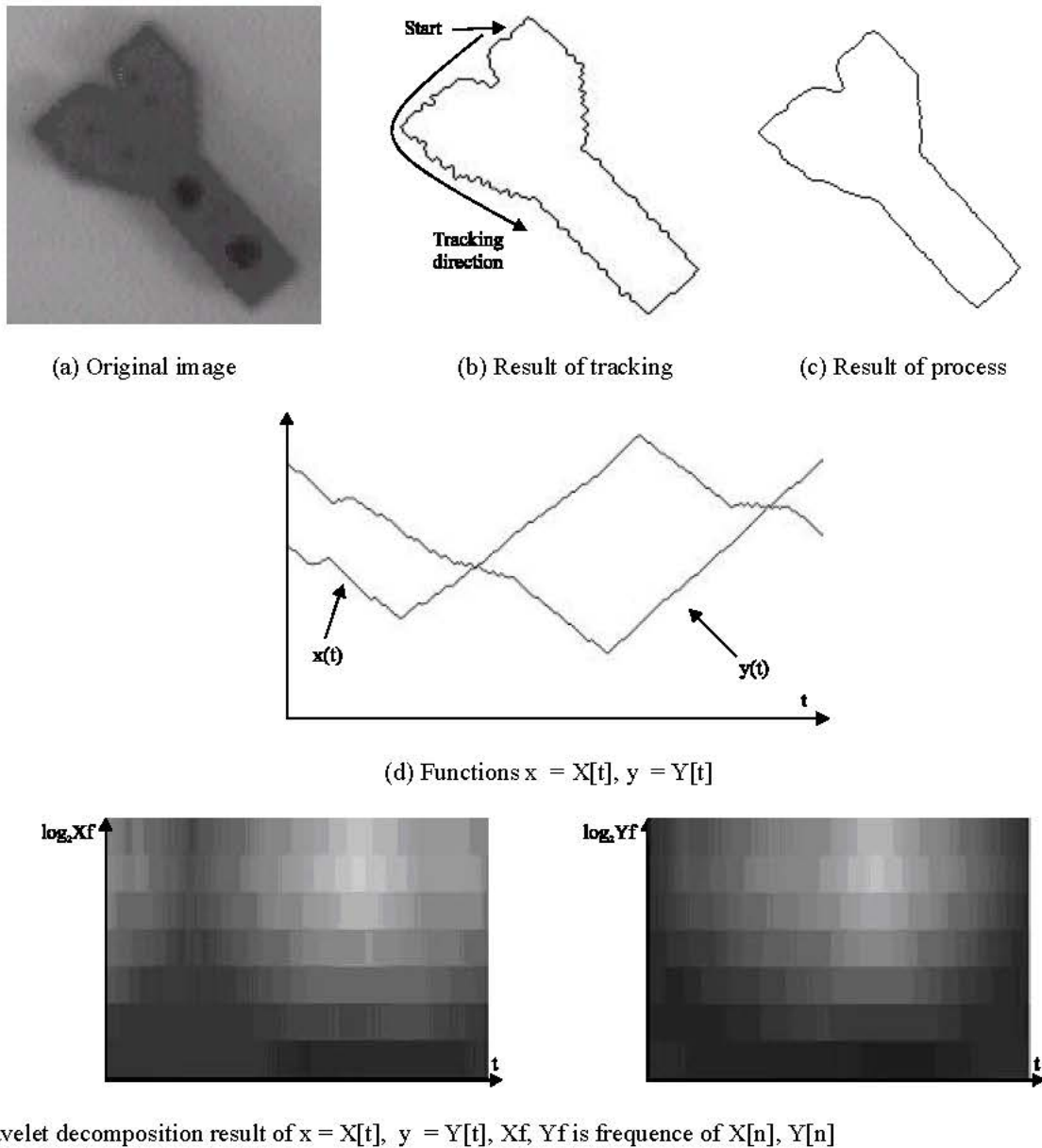


Fig. 2: The whole process of object edge smoothing

above. The continuity of boundary is required before tracking, that is to say every pixel is 8 neighborhood of last and next pixel. There are three cases would happen after processing: 1) current pixel does not connect with next pixel. 2) current pixel and next pixel are same position. 3) More than two pixels are connected with current pixel. So if case 1 happened, just connect the incontinuity part by beeline. If case 2 happened should delete superabundance pixels. If case 3 happened, skeletonizing algorithm should be applied.

RESULTS AND DISCUSSION

Figure 2a is original image. Figure 2b is tracking result. It tracks from start point with anticlockwise direction. The tracking result reflects approximately shape of workpiece. But the boundary isn't smooth and the noise is obvious. Figure 2c is the final image after processing. It shows that the noise has been reduced effectively. Figure 2d is the two functions of boundary. Figure 2e is the wavelet decomposition result of $x = X[t]$, $y = Y[t]$.

CONCLUSIONS

In order to apply wavelet to 2D shape processing the method presented translate object boundary into two one-dimensional arrays. It is convenient for data process and lower down the storage greatly. Same with the chain code and Fourier shape representation operators, they all need boundary tracking. Chain code can be regarded as the difference form of representation put forward and this representation method is more intuitionistic. Fourier shape representation operators method describe every pixel by complex number form and transform the pixel sequence by Fourier transformation. The Fourier coefficients of the transformed expression are called Fourier representation operators. The Fourier shape representation operators method has already changed the original data from time domain to frequency domain. It can reflect frequency status clearly, but cannot localize in time domain. And it is inconvenience to apply other processing to boundary. All algorithms realized by Visual C++ and achieve well effect in workpieces recognition.

REFERENCES

1. Charkes, K.C., 1994. An Introduction to Wavelets. (Chinese Edn.), Xi-an: Xi'an Jiaotong Univ. Publishing Company.
2. Xu, C.F. and G.K. Li, 2001. Application of Wavelet Method. Wuhan: Huazhong University of Science and Technology Publishing Company.
3. Xia, L.Z., 2002, Digital Image Process. Nanjing: South east University Publishing Company.
4. Zheng, N.N., 1998, Computational Vision and Pattern Recognition. Beijing: National Defence Industry Publishing Company.
5. Rosenfeld, R.F., 1973. Application of B-spline approximation to geometric problem of CAD. Ph.D Thesis, Syracuse University.
6. Roberge, J., 1985. A data reduction algorithm for planar curves. *Comput. Vision Graphics Image Process.*, 29: 168-195.
7. Naccache, N.J. and R. Shinghal, 1986. In response to: A comment on an investigation into the skeletonization approach to hilditch. *Patt. Recogn.*, 19: 111.