Journal of Zhejiang University SCIENCE A
ISSN 1009-3095 (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Denoising of Chinese calligraphy tablet images based on run-length statistics and structure characteristic of character strokes*

ZHANG Jun-song[†1], YU Jin-hui[1], MAO Guo-hong[2], YE Xiu-zi[1]
(*[1]State Key Lab of CAD & CG, Zhejiang University, Hangzhou 310027, China*)
(*[2]School of Computer Science, China University of Geosciences, Wuhan 430074, China*)
[†]E-mail: jszhang@cad.zju.edu.cn
Received Apr. 6, 2006; revision accepted Apr. 19, 2006

Abstract: In this paper, a novel approach is proposed for denoising of Chinese calligraphy tablet documents. The method includes two phases: First, a partial differential equations (PDE) based the total variation model and Otsu thresholding method are used to preprocess the calligraphy document image. Second, a new method based on run-length statistics and structure characteristics of Chinese characters is proposed to remove some random and ant-like noises. This includes the optimal threshold selection from histogram of run-length probability density, and improved Hough transform algorithm for line shape noise detection and removal. Examples are given in the paper to demonstrate the proposed method.

Key words: Denoising, Tablet images, Structure characteristics, Character strokes
doi:10.1631/jzus.2006.A1178 **Document code:** A **CLC number:** TP391

INTRODUCTION

As an art form, Chinese calligraphy has a long history of several thousand years. Many ancient calligraphy work created by former famous calligraphers were carved on stone tablets, and calligraphy documents were produced by rubbings. The most immemorial tablets we could look with reverence are "Shigu" stone tablets (left of Fig.1) which appeared about 2500 years ago in Warring States Period. The rubbed calligraphy document images however may look noisy, due to natural weathering of the tablets and artificial noises during the making rubbings, as shown in three examples of rubbed calligraphy documents in Fig.1.

Fig.1 Noisy calligraphy tablet documents

Denoising of the rubbed calligraphy document image is crucial for research on the digital calligraphy document, calligraphy character recognition, calligraphy document press, and calligraphy education. In recent years, many researchers did some research on computer calligraphy, which could mainly be divided into two categories, model based and image based. The model based research emphasizes modelling of basic elements of Chinese calligraphy, such as simulating of hairy brush (Strassmann, 1986; Wong and Ip, 2000; Xu *et al*., 2002; Mi *et al*., 2002; Girshick, 2004) and ink diffusion (Guo and Kunii, 1991; Lee, 1999; 2001; Way *et al*., 2003; Chu and Tai, 2005), while image based research mainly is concerned with how to reconstruct or analyze the calligraphy characters

* Project supported by the National Basic Research Program (973) of China (No. 2002-CB-312101) and the National Natural Science Foundation of China (No. 60773037)

from images, for instance, Xu *et al*.(2005) presented a novel approach to automatically generate artistic Chinese calligraphy from calligraphy images, Wong *et al*.(2005) proposed a method to achieve hairy brush parameters based on calligraphy images, Yu and Peng (2005) proposed a framework to synthesize 'Cao' style of Chinese calligraphy using brush texture patches from calligraphy images, and Zhuang *et al*.(2004) presented an approximate correspondence point algorithm to retrieve Chinese calligraphic character images. Those techniques however only work well for clear calligraphy images. Wang and Lee (2001) proposed an approach for thresholding of Chinese calligraphy documents, which could alleviate degree of noise to a certain extent, the approach is however not very effective for removing random ant-like noises.

Denoising of the rubbed calligraphy document images is in general a big problem, because the noises are randomly distributed in size and shape, and denoising sometimes may destroy the characteristic parts of strokes simultaneously, such as the stroke tips and corners. In this paper, we propose a novel and effective approach for denoising of calligraphy tablet images based on statistics and structure characteristics of Chinese characters. Our approach can remove random ant-like noises and preserve characteristic parts better than existing methods.

This paper is organized as follows. Section 2 describes the partial differential equations (PDE) based the total variation model and Otsu thresholding method used in the preprocessing phase in our system. In Section 3, we propose an approach based on run-length statistics and structure characteristics of Chinese characters to remove some random ant-like noises that are left after the preprocessing. Section 4 presents the experimental results and Section 5 gives a simple conclusion.

IMAGE SMOOTHING AND BINARIZATION

Characters on the tablet documents appear brighter, and the background appears darker. Simply applying thresholding to the documents may produce irregular holes on the character strokes because of the wide range of intensity on tablet images, as shown on the upper right of Fig.2. To overcome this drawback associated with the simple thresholding technique, we adopt the total variation model as well as Otsu binarization method for character image preprocessing, as described below.

Fig.2 Comparison of direct binarization and total variation before binarization is applied to a Chinese character "左"

Smoothing using total variation method

Total variation based smoothing and denoising approaches have achieved great success in the past decade, first suggested by Rudin *et al*.(1992). Compared with other image smoothing and denoising approaches, it is predominant in both theory and computation (Chan *et al*., 2004). We applied it to tablet image because it can preserve well the boundaries of characters which is crucial for calligraphy document image denoising. Besides, it can smooth the image and decrease broken strokes after thresholding.

The total variational PDE method is based on the linear degradation model, as described in (Chan *et al*., 2004):

$$u_0(x,y)-u(x,y)=n(x,y), \tag{1}$$

where $u_0(x,y)$ is the noisy image, $u(x,y)$ denotes the original clear image, and $n(x,y)$ refers to the noises which should be removed from the noisy image. Let $D\subset\mathbb{R}^N$ denote the image domain. By using the standard deviation of noise and integrating both sides of

Eq.(1), we obtain:

$$\int_{\mathbb{R}^N} (u_0 - u)^2 = \int_{\mathbb{R}^N} n^2 = \sigma^2. \tag{2}$$

The smoothing is produced after taking the local averaging around the image except at the boundary. In general, the definition of averaging leads to:

$$u(t,x,y)=u(0,x,y)=u_0(t,x,y), \text{ when } t=0, \tag{3}$$
$$\mathrm{d}u/\mathrm{d}t=\Delta u=u_{xx}+u_{yy}, \tag{4}$$

where $u_{xx}+u_{yy}$ is the Laplacian operator, which has strong smoothing effect. To gain better control over the smoothing effect, an upper limit for the average is used. Let M be a bound, we take the local average only where |derivative of $u_o(x)|\leq M$. In fact, it is transformed into a minimizing problem:

$$\min F(u) = \int_{\mathbb{R}^N} (u_x^2 + u_y^2)^{1/2} \mathrm{d}x\mathrm{d}y. \tag{5}$$

Combining the linear method constraint Eq.(2) with the nonlinear constraint Eq.(5), we obtain the standard total variation model:

$$\min F(u) = \lambda\int_{\mathbb{R}^N} (u_0 - u)^2 \mathrm{d}x\mathrm{d}y + \int_{\mathbb{R}^N} (u_x^2 + u_y^2)^{1/2} \mathrm{d}x\mathrm{d}y, \tag{6}$$

where $\lambda>0$ is a parameter to be chosen by the user. The total variation model does not address how to automatically set the parameters λ and the iterative times *iter*. Based on our experiments, $\lambda=0.01$ and *iter*=50 are good choices. The result of applying the total variation model is shown on the bottom left of Fig.2.

Otsu thresholding method to enhance the characters

The image processed with the total variation model is still characterized by gray value. For further image processing we need a binary image, so that thresholding techniques are required to process the gray image. Thresholding techniques are generally divided into global and local thresholding (Ye *et al*., 2001). The Otsu thresholding method (Otsu, 1979) is a global method that minimizes the weighted sum of within-class variances of the foreground and background pixels to achieve an optimum threshold. The Otsu method gives better results when the numbers of pixels in each class are close to each other, which is the case in the calligraphy tablet image. The result of the Otsu thresholding technique applied to the gray image derived from the total variation method is shown on the bottom right of Fig.2 showing that some dark parts superposed with character are smoothed and that the character in the tablet image is enhanced from the background.

RUN-LENGTH BASED NOISES REMOVAL

Strokes of Chinese characters generally include five primitive strokes: point, horizontal stroke, vertical stroke, left slant and right slant (as shown in Fig.3). All other complex strokes are just morphings or combinations of those primitive strokes, sometimes with local variations in shapes. Fig.3 shows that the stroke width varies in a continuous manner, and that strokes have distinct orientations. While noises may distribute randomly in the image in isolation and their shapes and sizes can be irregular, it is safe to assume that the sizes of noises are smaller than those of stroke widths in most cases. This is the fundamental assumption we will use in the paper.

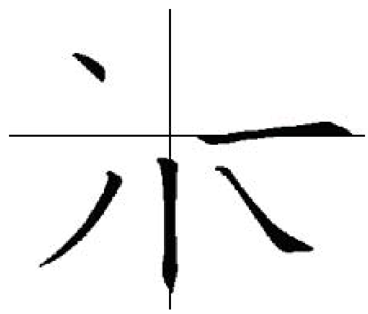


Fig.3 Five primitive strokes and their tendency

Our method first employs the run-length coding technique to get the statistics on the run-length of noise and character strokes to achieve the probability density histogram of run-length. We compute an optimal threshold T to distinguish the run-length of noises from character strokes and remove noises from the calligraphy tablet images based on the characteristics of stroke widths and orientations. We then adopt an improved Hough transform to remove remaining line shape noises and isolated points. We will describe our algorithm in detail next.

Statistics of run-length of noises and strokes

Fan and Wu (2000) used the run-length coding to extract strokes for Chinese character recognition, we adopt here the run-length coding in order to obtain an estimation of stroke widths. In the binary images (right bottom in Fig.2 where characters are white, and the background is black), we define the directional run as a set of white points along the given direction, with the run-length being the number of white points in the set.

The directional run can be defined in different directions. We choose horizontal run and vertical run in our implementation. Based on the statistical result of Chinese character strokes presented by Zhang (2003), horizontals, verticals, left slants, and dots appear in Chinese characters with probability of 18.21%, 16.54%, 16.51% and 13.2%, respectively; followed by the Zhuan-zhe (a horizontal stroke joining two left slants), Zhe (a horizontal run joined with a vertical run at the right side of the horizontal run) and left slants, with probability of 9.81%, 5.71% and 3.9%, respectively. Thus we can obtain a good estimation of stroke widths with the horizontal run and vertical run in most cases.

Fig.4 illustrates how the horizontal run-lengths are obtained, and the detailed procedures for calculating the horizontal and vertical run-lengths are given below:

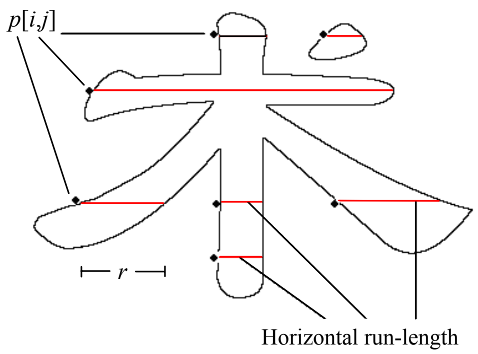


Fig.4 Horizontal run-lengths and points of left boundary of the character in horizontal scans

(1) Find all pixels $p[i, j]$ in sets A and B, which correspond to the left and upper boundaries of both characters and noises in the tablet image.

Horizontal scan:
$A=\{p[i,j]|p[i,j]=0 \text{ and } p[i+1,j]=1 \text{ and } p[i-1,j]=0\}$.

Vertical scan:
$B=\{p[i,j]|p[i,j]=0 \text{ and } p[i,j+1]=0 \text{ and } p[i,j-1]=1\}$,

where $p[i, j]$ denotes the location of the point in the calligraphy tablet image, corresponding to the ith row and jth column (see Fig.4 for horizontal scans).

(2) For every pixel $p[i, j]$ in sets A and B, we calculate the horizontal and vertical run-length r. To avoid the influence of horizontal or vertical strokes on the statistics of run-length during the horizontal or vertical scan, we fix three pixels to divide the run-length into four equal parts in length (Fig.5). As a result we get the corresponding pixel's vertical or horizontal run-length l. Through experiments we found that, if one of the three vertical or horizontal run-lengths is less than $r/3$, it is a horizontal stroke, otherwise it is stored in a one-dimension array Q.

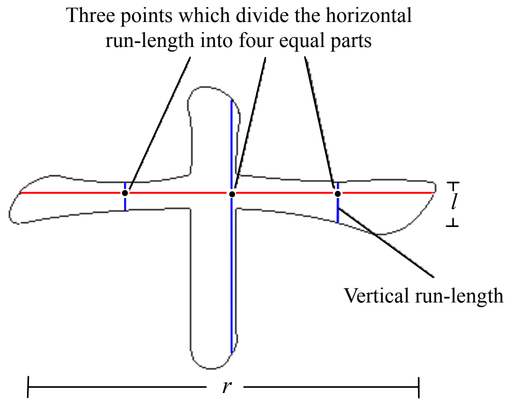


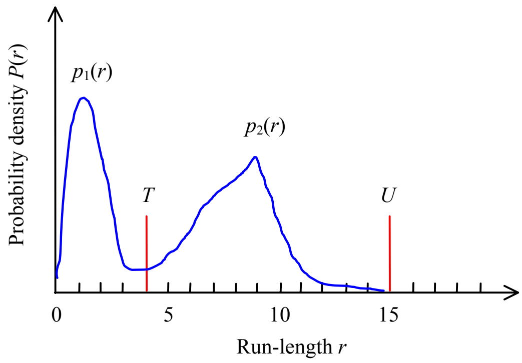
Fig.5 Method to avoid influence of horizontal stroke while horizontal scan

(3) Calculate the probability density of every run-length in array Q, and store it in a one-dimension array P.

$$P[r]=\frac{r\_runlength}{points\_sum}, \tag{7}$$

where *r_runlength* denotes the number of points with run-length r, *points_sum* is the sum of all the *r_runlengths* in set A and set B.

(4) Plot the histogram of run-length probability density. Fig.6 shows the histogram of probability density of run-length (Fig.6a) derived from a character “陽” (Fig.6b).



(a)

(b)

Fig.6 The run-length probability density histogram of Chinese character “陽”

Optimal threshold selection from histogram of run-length probability density

Through horizontal and vertical scanning of tablet images, the statistics of run-length from both upper and left boundaries of characters and noises can be obtained. By taking the run-length as a random variable, we can then get the histogram of mixed probability density distribution (Fig.6). There are two peaks in Fig.6, the first one corresponding to the run-lengths of most of the noises is close in position to the vertical axis, and the second corresponding to the run-lengths of stroke widths is far in position from the vertical axis. There is an overlap between them, and we need to find an optimal threshold T to partition them with the minimum error. Similar to the optimal threshold selection in image segmentation (Gonzalez and Woods, 2003), we presume the mixed PDF of run-length of noise and stroke width as:

$$p(r)=P_1p_1(r)+P_2p_2(r), \tag{8}$$

where P_1 and P_2 represent the probability of noise run-length and stroke run-length respectively, and that:

$$P_1+P_2=1. \tag{9}$$

According to the probability theory, the probability of random variable in a closed interval $[a, b]$ can be computed by integrating its probability density function from a to b, namely the area surrounded by the curve of probability density function in $[a, b]$. So if the stroke run-length is classified as noise, the probability of its error is:

$$E_1(T)=\int_{-\infty}^{T} p_2(r)\mathrm{d}r. \tag{10}$$

This is the area below the curve $p_2(r)$ on the left of the threshold T. Similarly, the probability of error of run-length of noise being classified as stroke is

$$E_2(T)=\int_{T}^{+\infty} p_1(r)\mathrm{d}r. \tag{11}$$

This is the area lying below curve $p_1(r)$ on the right of threshold T. Therefore, the total probability of error is

$$E(T)=P_2E_1(T)+P_1E_2(T). \tag{12}$$

We can obtain an optimal threshold T which makes the error minimum by differentiating $E(T)$ to T, i.e.,

$$P_1p_1(T)=P_2p_2(T), \tag{13}$$

and the solution of the above equation is the optimal threshold. We use Gaussian distribution to approximate $p_1(T)$ and $p_2(T)$, and obtain the mixed probability density function

$$p(r)=\frac{p_1}{\sqrt{2\pi}\sigma_1}\mathrm{e}^{-\frac{(r-\mu_1)^2}{2\sigma_1^2}}+\frac{p_2}{\sqrt{2\pi}\sigma_2}\mathrm{e}^{-\frac{(r-\mu_2)^2}{2\sigma_2^2}}, \tag{14}$$

where, μ_1 and σ_1^2 are the average and variance of

Gaussian density of noise run-length, respectively; μ_2 and σ_2^2 are the average and variance of Gaussian density of stroke run-length, respectively. Assuming $\sigma^2=\sigma_1^2=\sigma_2^2$, we use $p(r)$ in Eq.(14) to solve Eq.(13), and obtain the optimal threshold T as

$$T=\frac{\mu_1+\mu_2}{2}+\frac{\sigma^2}{\mu_1-\mu_2}\ln(P_2/P_1). \tag{15}$$

Based on the optimal threshold T and upper limit of stroke width U (Fig.6), we can get the bound of estimated stroke width $Sw\_b[T, U]$, the expectation value $E(r)$, so that the estimated stroke width in the tablet image, can be calculated through the following equation:

$$E(r)=\sum_{r=T}^{U}x_r p[r], \tag{16}$$

where x_r is the run-length, and $p[r]$ the corresponding probability density of x_r.

Noises removing by estimated stroke width

As mentioned above, Chinese character strokes have structural features in continuity and directions which can be used to distinguish strokes from most types of noises. An effective way to make sure a white pixel belongs to the stroke or noise in the calligraphy image is to compute its run-length in four directions: 0°, 45°, 90°, 135°, if all of them are smaller than the estimated stroke width T, then this pixel probably belongs to noise, and should be removed by setting it to black. This method can also protect the knife-edged stroke tips from being removed erroneously, though the stroke tips' run-length in three of the four directions may be smaller than the estimated stroke width, the run-length at least in one direction can be larger than the estimated stroke width (Fig.7).

However, after most of the noises are removed by the above method, there may remain some isolated points and line shape noises (Fig.8). For isolated points, we can simply remove them by detecting every white pixel in the image; we set those pixels black in the case of its neighboring eight pixels are all black.

After isolated point noises are removed, some line noises may still be left on the image, because their sizes are bigger in four directions (0°, 45°, 90°, 135°) than the stroke width. We adopt an improved Hough transform to remove line noises.

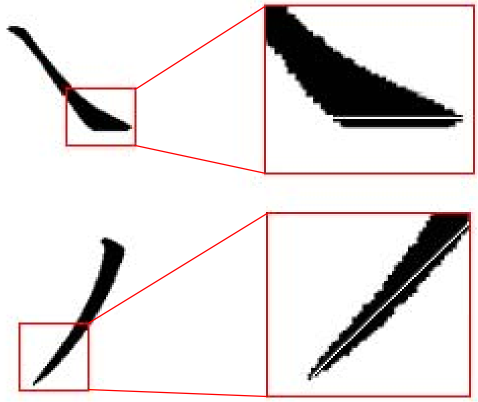


Fig.7 Micro-structure of stroke tips

Fig.8 Result of noise removal by estimated stroke width (Sw_b[4, 16], $E(r)$=7. 95)

Removing line noises by modified Hough transform

Hough transform can be used to isolate features of a particular shape within an image. Its main advantage is that it is relatively unaffected by image noise. For line detection, the classical Hough transform considers a point (x_i,y_i) in the Cartesian space. For any (x_i,y_i), there is a family of lines through this point, given by:

$$y_i=kx_i+b, \tag{17}$$

where k is the slope, and b is the intercept of the lines.

Changing the equation to

$$b=-x_ik+y_i, \tag{18}$$

which represents a line in parameter (k,b) space, and only confirmed by (x_i,y_i), then the parameter space is further segmented into accumulator elements (k_i,b_i). Initially, the values of the accumulator elements are set to 0. For each point (x_i,y_i) in the Cartesian space, given the parameter k, we can get the parameter b from Eq.(18). After all the points are processed in the Cartesian space, we can make sure which set of points form lines through a preset threshold value of accumulator elements. To solve the problem that line slopes approach infinity when lines are vertical to x axis in the Cartesian space, the line equation in Eq.(17) is replaced by the following polar form:

$$x_i\cos\theta+y_i\sin\theta=\rho, \tag{19}$$

then, the parameter space (k,b) is changed to (ρ,θ), and the computation process remains the same as described above.

The heavy computations involved in the generalized Hough transform restrict its application in our case. Since line noises are only left in four directions (0°, 45°, 90°, 135°) after the isolated point noises are removed, and most of line noises left are either single pixel or double pixels in width, or two more lines intersect, we modify the classical Hough transform to the following form to speed up the calculation in our application:

(1) We just detect lines in four directions (0°, 45°, 90°, 135°) rather than many more directions that the ordinary Hough transform performs, this reduces the calculation significantly.

(2) Eq.(17) is still used as the line equation, which avoids more computation involved in the trigonometric function in polar form. As for the lines that are vertical to x axis, we use the line equation $x=b$ when their slopes approach infinity.

(3) For intersection lines, we set a parameter n to count the white points around the detected lines. For example, $n=2$ indicates that there is a single line intersecting the current line, so this line should be removed.

Fig.9 shows the result when the lines and isolated points are removed.

Fig.9 Removing lines by modified Hough transform from Fig.8

EXPERIMENTAL RESULT

The proposed algorithm was implemented on a PC platform with Microsoft Visual C++ 6.0. All testing calligraphy tablet images were obtained from the Internet or through scanning of tablet image books in 300 dpi with 256 gray scales, which totaled more than 100 documents. We have integrated all the steps described above together, including the Otsu method from the original tablet images; run-length statistics of character strokes and noises; optimal threshold selection to distinguish noise from character strokes; as well as noise removal by estimated stroke width and line shape noise removal by modified Hough transform.

The whole process of denoising runs automatically, in which the total variation smoothing is optional. The smoothing process may cause discrete noise points to be connected, thus forming bigger noises that are difficult to be distinguished from character strokes. We can get better results of denoising without performing the smoothing operation in such cases.

Fig.10 shows the results of the denoising compared with the method proposed by Wang and Lee (2001), where anisotropic diffusion and binarization are integrated to remove the noises. We can see that though the thresholding before anisotropic diffusion could alleviate the level of noises, there are still many noises left un-removed.

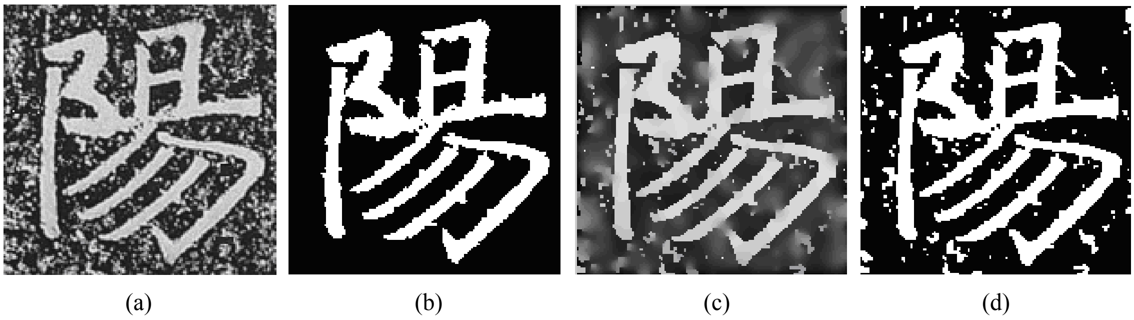


Fig.10 The results of denoising the tablet image with Chinese characters "陽". (a) The original image; (b) Denoising with our method (*Sw_b*[4, 16], *E*(*r*)=7. 95); (c) 50 times anisotropic diffusion from the original image; (d) Binarization result with Otsu from (c)

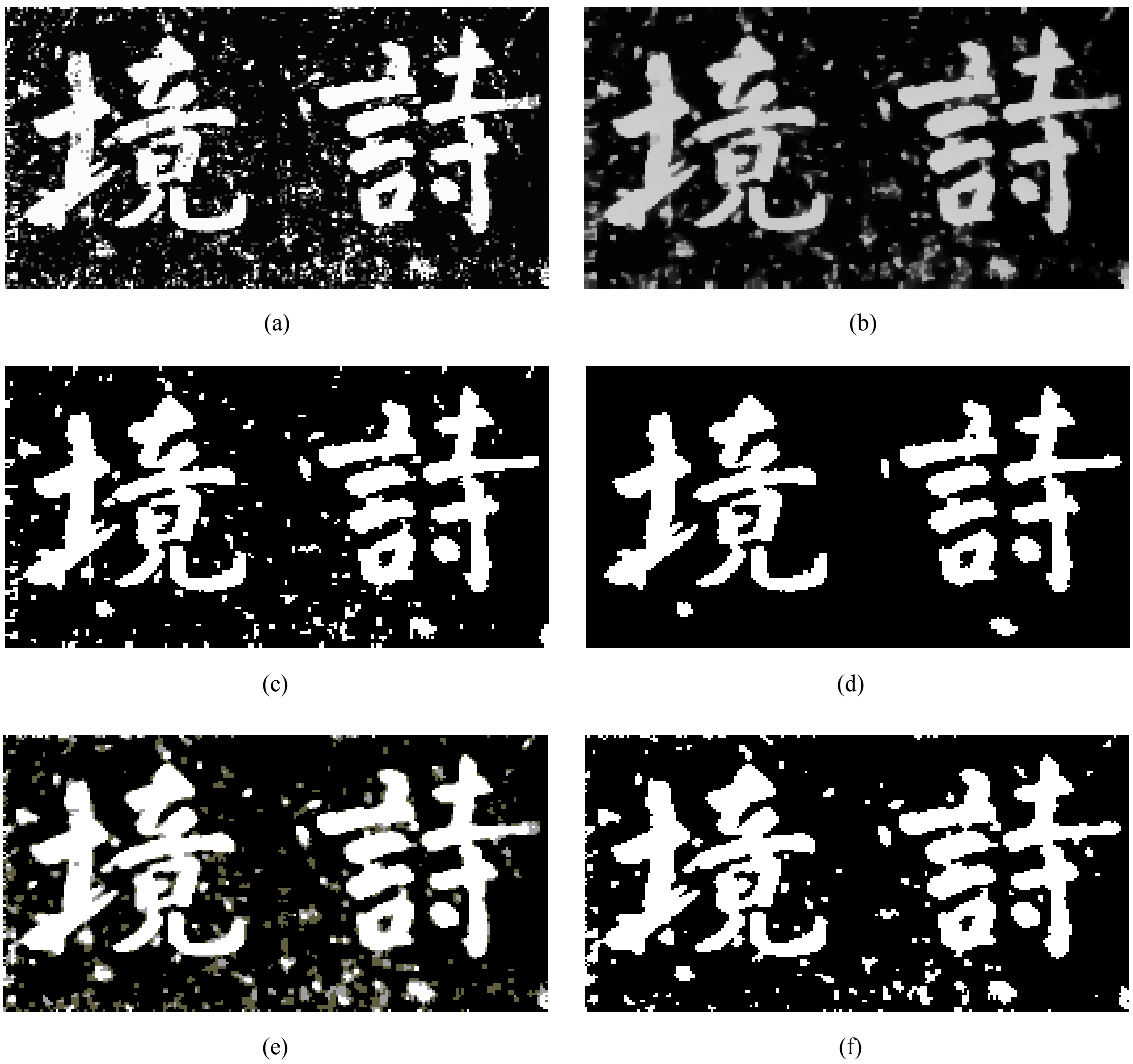


Fig.11 The results of denoising with Chinese character "詩境". (a) The original image; (b) Smoothing with total variation; (c) Otsu from (b); (d) Denoising with estimated stroke width (*Sw_b*[3, 9], *E*(*r*)=5.62) as well as isolation points and lines removal by modified Hough transform; (e) Median filter from the original image; (f) Otsu from (e)

Fig.11 (see page 1185) shows another example of denoising compared with median filter normally used to reduce spike or speckling noise from a gray image. Our method is more effective than median filter for denoising of tablet images. However, there is still some noise left (Fig.11d), which are bigger in size so that it is very hard to distinguish them from the primitive stroke ‘Point’.

CONCLUSION

In this paper, we presented a novel approach for denoising tablet documents of Chinese calligraphy. The proposed method utilizes the total variation model, Otsu thresholding technique with run-length statistics and structure characteristics of Chinese characters. Examples given in the paper demonstrate the power of our method with the results being better than those achieved by other methods.

The boundary of strokes after denoising processing still looks rough, we intend to smooth the stroke boundary in our future work.

ACKNOWLEDGEMENTS

We would like to appreciate Mr. Tang Yonghui and Mr. Tong Zhou for their helpful discussion in this paper.

References

Chan, T.F., Esedoglu, S., Nikolova, M., 2004. Algorithms for Finding Global Minimizes of Image Segmentation and Denoising Models. UCLA CAM Report 04-54.

Chu, N.S.H., Tai, C.L., 2005. MoXi: Real-time Ink Dispersion in Absorbent Paper. *ACM Trans. on Graphics (SIGGRAPH’05)*, **24**(3):504-511. [doi:10.1145/1073204.1073221]

Fan, K.C., Wu, W.H., 2000. A Run-length Coding Based Approach to Stroke Extraction of Chinese Characters. 15th International Conference on Pattern Recognition (ICPR’00), **2**:565-568. [doi:10.1109/ICPR.2000.906137]

Girshick, R.B., 2004. Simulating Chinese Brush Painting: The Parametric Hairy Brush. Proc. of SIGGRAPH’04.

Gonzalez, R.C., Woods, R.E., 2003. Digital Image Processing, 2nd Edition. Publishing House of Electronics Industry, Beijing.

Guo, Q., Kunii, T., 1991. Modelling the diffuse painting of “Sumi-e”. Modelling in Computer Graphics. Springer-Verlag, Tokyo, p.329-338.

Lee, J., 1999. Simulating oriental black-ink painting. *IEEE Computer Graphics and Applications*, **19**(3):74-81. [doi:10.1109/38.761553]

Lee, J., 2001. Diffusion rendering of black ink paintings using new paper and ink models. *Computers & Graphics*, **25**(2):295-308. [doi:10.1016/S0097-8493(00)00132-1]

Mi, X.F., Xu, J., Tang, M., Dong, J.X., 2002. The Droplet Virtual Brush for Chinese Calligraphic Character Modelling. Sixth IEEE Workshop on Applications of Computer Vision. Orlando, Florida, p.330-338.

Otsu, N., 1979. A threshold selection method from gray level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, **9**(1):62-66.

Rudin, L.I., Osher, S., Fatemi, E., 1992. Nonlinear total variation based noises removal algorithms. *Physica D Nonlinear Phenomena*, **60**(1-4):259-268. [doi:10.1016/0167-2789(92)90242-F]

Sam, T.S., Wong, H.L., Ip, H.H.S., 2005. Model-based Analysis of Chinese Calligraphy Images. 9th International Conference on Information Visualization. London, England, p.221-226.

Strassmann, S., 1986. Hairy brushes. *Computer Graphics (SIGGRAPH’86)*, **20**(4):225-232. [doi:10.1145/15886.15911]

Wang, S.Z., Lee, H.J., 2001. Dual-Binarization and Anisotropic Diffusion of Chinese Characters in Calligraphy Documents. Proceedings of the Sixth International Conference on Document Analysis and Recognition, p.271-275. [doi:10.1109/ICDAR.2001.953797]

Way, D.L., Huang, S.W., Shih, Z.C., 2003. Physical-based Model of Ink Diffusion in Chinese Ink Paintings. WSCG, 2003.

Wong, H., Ip, H., 2000. Virtual brush: a model-based synthesis of Chinese calligraphy. *Computer & Graphics*, **24**(1):99-113. [doi:10.1016/S0097-8493(99)00141-7]

Xu, S.H., Tang, M., Lau, F., Pan, Y.H., 2002. A solid model based virtual hairy brush. *Computer Graphics Forum*, **21**(3):299-308. [doi:10.1111/1467-8659.00589]

Xu, S.H., Lau, F.C.M., Cheung, W.K., Pan, Y.H., 2005. Automatic generation of artistic Chinese calligraphy. *IEEE Intelligent Systems*, **20**(3):32-39. [doi:10.1109/MIS.2005.41]

Ye, X.Y., Cheriet, M., Suen, C.Y., 2001. Stroke-model-based character extraction from gray-level document images. *IEEE Trans. Image Processing*, **10**(8):1152-1161. [doi:10.1109/83.935031]

Yu, J.H., Peng, Q.S., 2005. Realistic synthesis of Cao shu of Chinese calligraphy. *Computers & Graphics*, **29**(1):145-153. [doi:10.1016/j.cag.2004.11.013]

Zhang, S.Z., 2003. Statistic Characteristics of Strokes of Chinese Characters. Http://www.chancezoo.net/hz/hzbhtjtx.htm.

Zhuang, Y.T., Zhang, X.F., Wu, J.Q., Lu, X.Q., 2004. Retrieval of Chinese calligraphic character image. *PCM*, (1):17-24.