ISOLATED CHARACTERS EXTRACTION USING DIFFERENCE-OF-GAUSSIAN FUNCTION

YON-PING CHEN(陳永平)
Dept. of Electrical and Control Engineering,
National Chiao-Tung University
Email:ypchen@cc.nctu.edu.tw

TIEN-DER YEH(葉天德)
Dept. of Electrical and Control Engineering,
National Chiao-Tung University
Email:tainder.ece91g@nctu.edu.tw

***Abstract*—A method to extract isolated characters is proposed by using Difference-of-Gaussian(DOG) function. Isolated characters, especially English alphabet or numerical characters, can be seen everywhere in our daily life. How to extract these characters from a digital image efficiently and robustly has been a popular topic for researchers. The DOG function, similar to Laplacian of Gaussian function, was proven to produce the most stable image features compared to a range of other possible image functions. The method incrementally convolves the input image with different scale Gaussian functions and minimizes the computations in high scale images by means of sub-sampling. The candidates of characters are found by connected components analysis in the DOG image and then filtered by group sizes to ignore the unmatched groups. Finally, the experimental results demonstrate the success of isolated characters extraction and robustness against noise and illumination change.**

***Index Terms*— Difference of Gaussian function, isolated character, character recognition, scale space,**

I. INTRODUCTION

Isolated English alphabet and numerical characters are closely connected with our daily lives nowadays. They can be seen everywhere no matter in texts, home appliances, traffic vehicles, commercial products, or factory facilitates. Due to wide spread of these characters, how to locate and identify them by a computer is getting more and more important for not only living convenience but also working essentials.

Characters recognition, one of the most popular topics of computer vision, has been studied by a lot of investigators for many decades. Some character recognition systems are already applied onto real world applications and play important roles around us. For example, optical character recognition (OCR) systems [1] [2] are mature today to recognize the characters on texts. It helps to translate printed texts into digital data automatically and saves much time of boring typing. Today's OCRs are widely used and well performed in not only large devices such as desktop personal computers but also small portable devices such like PDA(Personal Digital Assistant) or mobile phone. License plate recognition (LPR) systems recognize vehicles automatically so that it is useful for many applications such as portal controlling, traffic monitoring, stolen car detection, and etc. Traffic sign recognition (TSR) systems improve road safety by recognizing traffic signs and informing the driver.

The procedures for characters recognition, take LPR systems for example, often include several stages. Stage one is detection of possible areas in which the characters may exist. This task is challenge since images may contain far more information than just only expected characters. Until now new methods are continuously discovered to deal with special pattern detection in an image. Stage two is segmentation, in which the detected areas are divided into several regions and each of them may contain one character candidate or more. Stage three is normalization, the character candidates are re-sampled with some parameters, e.g., size or orientation, aligned to fixed values for robust recognition in later stage. Stage four is recognition stage; the segmented characters can be recognized by technologies such as vector quantization[9] or neural networks[4][5]. Most researchers propose to recognize characters in binary forms so that they find thresholds[6][7] to depict the regions of interest in the detected areas.

Up to now, there are some limitations in the character recognition systems mentioned above and make them difficult to be directly adopted into isolated character recognition. For OCR system nowadays, even though they are applicable to help people on many applications, most of them still face the limitation of recognizing only scanned documents. Camera based documents analysis [8] [9] is still a problem due to many complicated imaging factors it may involve. Without pre-defined parameters, typical imaging factors such as noise, illumination, focus error, camera view angle,…, are much more complicated than scanned documents and difficult to be handled by an OCR system. On the other hand, LPRs and TSRs rely greatly on some specific patterns of license plates and traffic signs. For example, LPR need a white and rectangular background with the characters aligned tidy on a baseline [6] before the background. TSR detects the special shapes, triangle, circle or rectangle, of the traffic signs before recognizing the characters so that it cannot be utilized to recognize characters without the special shapes. In conclusion, LPRs and TSRs are not suitable to be applied into isolated characters recognition.

There are few literatures discussing about isolated characters recognition due to several difficulties it has. First, it is difficult to extract the exact position and size of an isolated character. Due to lack of special patterns, isolated characters are not easily detected as those of license plates. Traditional methods take threshold(on intensity or gradient magnitude) first then apply connected component analysis is not suitable for the isolated characters because it needs the exact information of the target character and background under different camera or environmental factors. Second, it is difficult to know orientation of an isolated character. In OCR [1] [2] or LPR [6] [10], the orientations of characters can be detected by searching for baseline and spacing with adjacent characters; however, this method is not suitable for isolated characters because baseline of a single character is not significant and often leads to unstable orientation. Third, the unfixed camera view angle often introduces large deformation on the character shapes or stroke directions. It makes the detection and normalization process difficult to be applied. Fourth, the unknown orientations and shapes exposed under unknown light condition and environment makes it more difficult for the characters to be correctly detected and recognized.

This paper targets on the first problem of isolated character recognition: to extract the profiles of isolated characters by three steps. First, the scale-space differences are derived by difference of Gaussian functions [11], which make the result stable against noise and illumination change. Second, the pixels of positive or negative differences are gathered into groups by means of connected components analysis and form candidates of target characters. Third, the candidates of unexpected sizes are removed and the remainders are proceeded to the next recognition stage. The experiment results prove the proposed method is robust against noise and illumination change.

II. Theory of Scale Space

The concept of scale space [13] starts from the basic observation that real-world objects are composed of different structures at different scales. In other words, real-world objects may appear in different ways depending on the scale of observation. For a computer designed to detect the existence of an object in an image, it is necessary to consider all the possible scales the object may appear in order to capture it in the correct scale.

Earlier works such as [14] and [15] have suggested that Gaussian function is the best choice for scale-space kernel. Also, in [15], the author showed that the difference-of-Gaussian function provides a close approximation to the scale-normalized Laplacian of Gaussian, $\sigma^2\nabla^2 G$, which was proven by detail experiment in [16] that it produces the most stable image features compared to a range of other possible image functions.

In mathematical representation, let $\boldsymbol{I}(x, y)$ denote pixels of the input image, the scale space of an image is defined as a function $\boldsymbol{L}(x,y,\sigma)$ generated by convolving the input image with variable-scale

Gaussian function $G(x,y,\sigma)$, denoted as

$$\boldsymbol{L}(x,y,\sigma)=G(x,y,\sigma)*\boldsymbol{I}(x,y), \tag{1}$$

where * is the convolution operator and $G(x,y,\sigma)$ is expressed as

$$G(x,y,\sigma)=\frac{1}{2\pi\sigma^2}\mathrm{e}^{-\left(x^2+y^2\right)/2\sigma^2}, \tag{2}$$

In the field of computer vision, smoothing of images before all the other processes is often necessary for noise reduction. The Difference-of-Gaussian functions, unlike most edge sharpening filters utilized in computer vision having strong responses to noise, play roles of not only enhancing the edges but also reducing noise [17] in a digital image. Moreover, there are two advantages to use Gaussian functions as smoothing kernel. First, the symmetric property of it makes it practical to decompose the two-dimensional convolution into two independent single dimensional equations.

$$\begin{aligned}G(x,y,\sigma)&=G_x(x,\sigma)\times G_y(y,\sigma)\\&=\frac{1}{\sqrt{2\pi}\sigma}\mathrm{e}^{-\left(x^2\right)/2\sigma^2}\times\frac{1}{\sqrt{2\pi}\sigma}\mathrm{e}^{-\left(y^2\right)/2\sigma^2}\end{aligned}, \tag{3}$$

This greatly reduces the computation and shortens the process time in finding different scale images. Second, taking the Fourier transform of a Gaussian function yields another Gaussian function [18], i.e.,

$$F\left(\frac{1}{\sqrt{2\pi}\sigma}\mathrm{e}^{-\left(x^2\right)/2\sigma^2}\right)=\mathrm{e}^{-\left(\omega^2\right)/2(1/\sigma)^2}, \tag{4}$$

where $F(g)$ denote the Fourier transform of a function g. Based on (4) we can derive that the succeeded convolution with $G(\sigma_2)$ after $G(\sigma_1)$ is equivalent to convolution with $G(\sigma_1+\sigma_2)$

$$\begin{aligned}F(I*G(\sigma_1)*G(\sigma_2))&=F(I)\cdot\mathrm{e}^{-\left(\omega^2\right)/2(1/\sigma_1)^2}\cdot\mathrm{e}^{-\left(\omega^2\right)/2(1/\sigma_2)^2}\\&=F(I)\cdot\mathrm{e}^{-\left(\omega^2\right)/2(1/(\sigma_1+\sigma_2))^2}\\&=F(I*G(\sigma_1+\sigma_2))\end{aligned}, \tag{5}$$

Eq.(5) is helpful to derive the smoothed images on different sizes to save a quite amount of computations.

III. Extraction of Isolated Characters

It is difficult to extract something from an image if no prior information is known on the imaging factors. Typical imaging factors are known in four categories: camera, environment, background, and target. Environment factors such as light condition, noise, occlusion, are difficult to be controlled so that only cancellation or compensation can be used. Camera factors such as focus, view angle, saturation …, are possible to be controlled to fit the condition of best successful rate. Background color, texture, target's color, shape, size, orientation …, are closely related to the final successful rate and all are important for extraction. The more information is known about these factors before processing, the easier and better result is possible to be achieved in final results.

Before the process starts, there are four assumptions made for extracting isolated characters by the proposed methods:

1. The color (or intensities for gray scale images) of a character is monotonic, i.e., the character is composed of single color without texture on it.
2. Same as 1, the color of background around the character is monotonic, too.
3. The color of the character is always different from that of the background;
4. Characters must be isolated between each other and no overlap in the input image.

The scale space theory introduced in Chapter.2 will be used to extract the characters of interest from an image. In following sections the procedures of extraction will be detail described by two parts: produce the Difference-of-Gaussian images and make group of them.

A. Produce the Difference-of-Gaussian Images

As mentioned before that the isolated characters must be captured in the correct scale, we proposed to search them in all the scales. Taking advantage of the scale-space theories[11]-[15], the extraction of characters become systematic and effective. As introduced previously, the scale-space images are generated by convolving input image with different scale Gaussian functions. And the 2D Gaussian functions are decomposed into two 1D filters with equal parameters but orthogonal direction. Two parameters, filter width λ and smoothing factor σ, required for Gaussian filters are not fully independent yet some constraints between them need to be discussed.

The range of smoothing factor σ is determined from experiments that a better choice of it is from 1 to 16, for the input image sizes ranged from 640×480 to 2048×1536. Two factors are needed to be considered about the sampling frequency of σ: the resolution of the target characters and the available computational resources(including processing time). These two factors play roles of trade-off and often determined case by case. We choose to set σ of a scale double of that of the previous scale for convenient computation, i.e., $\sigma_2=2\sigma_1$, $\sigma_3=2\sigma_2$ …, where σ_1, σ_2, σ_3…, are the corresponding smoothing factors of the scale numbered 1,2,3,…. As a result, the choice of smoothing factors in our case is, $\sigma_1=1$, $\sigma_2=2$, $\sigma_3=4$, $\sigma_4=8$, and $\sigma_5=16$. Consider factors of noise and sampling frequency in the spatial domain, the larger size the character is, the more stable and better result can be achieved by a larger σ in the extraction. Ideally the width λ of a Gaussian filter is infinity, while in real case it is reasonably to be integers to match the requirements of digital computers. Furthermore, the integers cannot be large due to limited computation resources and only odd integers are chosen such that each convolution output can be aligned to the center of a pixel. The smoothing factor σ is in other words the standard deviation of the Gaussian distribution. Smaller σ has better response on edges but yet more sensitive to noise. When σ is small, there is no need to define a large λ because the filter decays to a very small value when it reaches the boundary. In this paper we propose to choose the two parameters satisfying the following inequality,

$$\lambda \geq \sigma \times 7 \quad \& \quad \lambda = 2n+1, \forall n \in N. \tag{6}$$

Consider to the limit, when λ equals 7σ, the filter magnitude at the boundary point x_b =$(7\sigma$-1)/2 (center point x_c=0) becomes

$$G_x(x_b,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\left(\frac{(7\sigma-1)^2}{2}\right)/2\sigma^2} \Big|_{\sigma=1} < 0.001 G_x(x_c,\sigma) \tag{7}$$

It states that the filter magnitude falls to less than 0.1% of that of the center and is reasonably to be ignored.

An efficient way to generate the smoothed images is taking sub-sampling. As explained above that the filter width is better chosen $\lambda \geq 7\sigma$, it makes the filter width grows to 112 if the smoothing factor reaches 16 in scale number 5. This leads to a large amount of computation in real case if the filters are implemented in such a long size. To avoid expanding the filter width directly, we take use of sub-sampling on images of smoothing factors $\sigma>1$ based on the truth that the information in images are decreased as the smoothing factors increase. The smoothed images are divided into several octaves that the length and width are one half of those of the previous octave. There are two images in each octave and their differences are computed to produce the desired Difference-of-Gaussian(DOG) image for later processing.

Let the length and width of the input image $\boldsymbol{I}(x,y)$ be L and W respectively. In the beginning, $\boldsymbol{I}(x,y)$ is convolved with Gaussian filter $G(x,y,\sigma_a)$ to generate the first smoothed image, $\boldsymbol{I}_1(x,y)$ for the first octave. σ_a is the smoothing factor of the initial scale and selected as 1 ($\sigma_a=\sigma_1$) in our experiments. The smoothed image $\boldsymbol{I}_1(x,y)$ is used to convolve with Gaussian filter $G(x,y,\sigma_b)$ to generate the second smoothed image $\boldsymbol{I}_2(x,y)$, which will be used to subtract from $\boldsymbol{I}_1(x,y)$ to get the first DOG image $\boldsymbol{D}_1(x,y)$ on the octave and sub-sampled by every two pixels in each row and column to produce the image $\boldsymbol{I}_2'(x,y)$ for the next octave. It is worth to note that an image sub-sampled from a source image has smoothing factor equal to one half of that of the source image. The length and width of image $\boldsymbol{I}_2'(x,y)$ are $L/2$ and $W/2$, and the equivalent smoothing factor is $(\sigma_a+\sigma_b)/2$. As the σ_b is selected to be same as the smoothing factor σ_a of the initial scale, the image $\boldsymbol{I}_2'(x,y)$ therefore has the equivalent smoothing factor $\sigma=\sigma_a$, and is served as the initial scale of the second octave. The image $\boldsymbol{I}_2'(x,y)$ is convolved with $G(x,y,\sigma_b)$ again to generate the third smoothed image, $\boldsymbol{I}_3(x,y)$, which can be subtracted from $\boldsymbol{I}_2'(x,y)$ to produce the second DOG image $\boldsymbol{D}_2(x,y)$. The same procedure can be applied to the remaining octaves to generate the required smoothed images $\boldsymbol{I}_4$ and $\boldsymbol{I}_5$, and difference-of-Gaussian images $\boldsymbol{D}_3$ and $\boldsymbol{D}_4$. Fig.1 illustrates the procedure to produce the smoothed and DOG images.

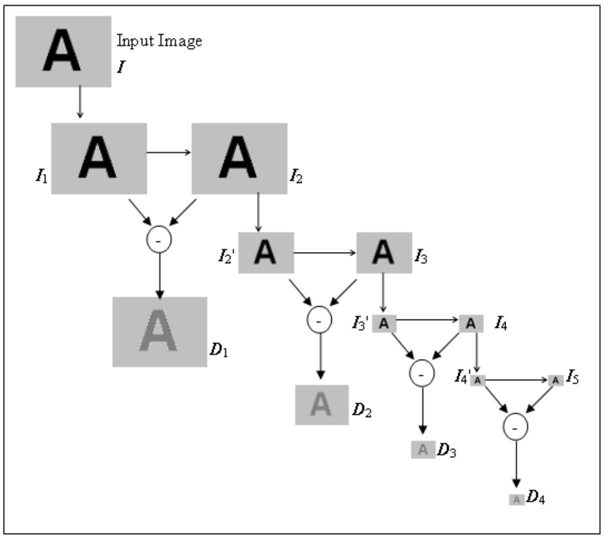


Fig.1: The procedure to produce Difference-of-Gaussian Images

B. Grouping of the Difference-of-Gaussian Images

As shown in Fig.2, an ideal edge (unit step) can be separated into two regions by convolving with a DOG function, one region has positive response(x>0) and the other has negative response(x<0). Besides, it tends to generate two peaks (positive peak **A** and negative peak **B**) near the edge which are opposite in sign and are robust to represent the existence of the edge. This property is useful to separate the target characters from background while minimizing the impact of noise and illumination change.

To find the interested characters in the DOG image, the first step is to apply connected components analysis to connect pixels of positive (or negative) responses into groups. Black characters (compare to background) can be extracted by negative responses while white ones can be extracted by positive responses. The analysis tends to concentrate on the pixels of peaks i.e. the set of **A** or **B** in Fig.2 and often forms groups along the edges to generate the boundaries segmenting target characters from background. After connected components analysis, all the groups are filtered by their sizes and eliminated if their sizes are not falling into the desired range. The most stable sizes for extracting general characters on each octave are ranged from 32×32 to 64×64. Characters sizes smaller than 32×32 are easily disturbed by noise and result in incorrect results. Note that characters sizes larger than 64×64 can be extracted on the next octave.

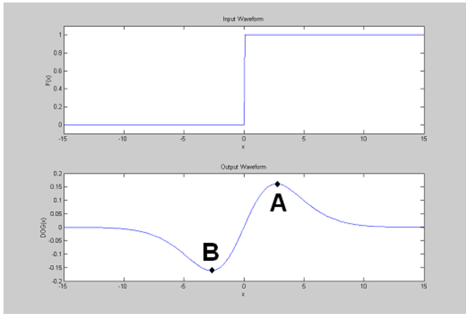


Fig.2: An ideal edge(upper graph) and the response(lower graph) of convolution with a DOG function(σ_1 =2;σ_2 =4)

It is worth highlight that the DOG response, behaving like band-pass filtering, is robust to desired frequency bands (like edges) but maybe unstable to DC (like flat surfaces). Fig.4 shows an image of a license plate and its DOG image shows only pixels of negative response in white pixels. It is apparent that the white pixels in Fig.4(b) is stable along the edges but irregular on large flat surfaces(some areas inside characters **N** and **W**). Although these pixels won't affect the extraction result of the characters, they will occupy a lot of computation resources and slower down the processing speed. A best way to deal with this characteristic is taking threshold since it is often made of small perturbations near zero responses. Fig.4(c) shows the binary image after taking threshold.

(a).

(b).

(c).

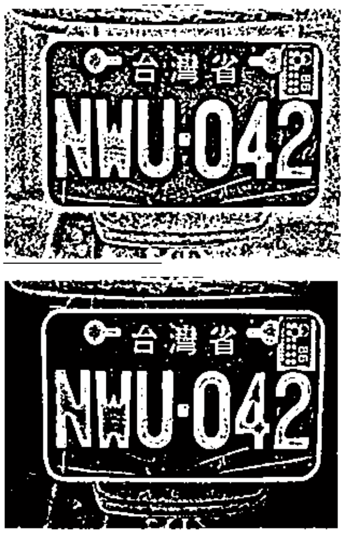


Fig.4 (a) An image of License plate and (b) binary image, white pixels are formed by negative responses of DOG image. (c) Thresholded image.

IV. EXPERIMENTS

The test images in the experiments are captured around our campus and the characters inside are very common in our daily life. For examples, license plates, elevators, telephones, mobile phones, clocks, calculators, computer keyboards… During the experiments, all the test images are converted into 8-bit gray-scale images and the sizes of them are re-sampled to 640×480. The experiments results are collected into three bins by different characters sizes (size1: 16×16, size2: 32×32, size3: 64×64) and measured by the true positive rate (TPR), i.e., the rate that the true characters are extracted successfully in the test image. The discussion is focused on the stability to several imaging factors, such as noise and illumination change. A character is considered as successfully extracted if it is isolated from external objects and the grouped pixels can be recognized by human eyes.

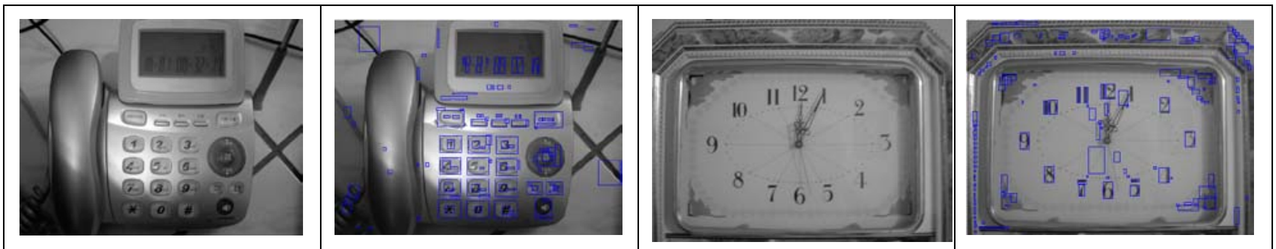
Due to page size limitation, only six test images and the results of extraction are shown in Fig.5. We use blue rectangles to show the character candidates by negative DOG responses and red rectangles by positive DOG responses.

A. Stability to Noise

There are two different kinds of noise used in the experiments, one is pepper and salt noise and the other is quantization noise.

Five levels [1% ~ 5%] of pepper and salt noise are added in the test images; wherein 5% stands for that one pepper or salt appears in every 20 pixels. From the simulation result in Fig.6, we can see that the characters with larger sizes are more robust against pepper and salt noise. If an image contains characters of sizes smaller than 16×16, it is suggested to extend the width of Gaussian filter to increase the stability against noise.

For quantization noise, we added 5 levels[0.8%, 1.6%, 2.4%, 3.2%, 4.0%] of noise into each pixel. The 0.8% noise level is equivalent to add 1 or -1 randomly into the 8-bit gray-level image. From the simulation result in Fig.7, the same conclusion can be obtained as previous result that bigger characters tend to be more stable against quantization noise. In addition, to reduce the impact of quantization noise, an input image of higher gray-level resolution is preferred, e.g., 12-bit or 16-bit will be much better than 8-bit resolution for unknown light condition and environment.



(a) (b)

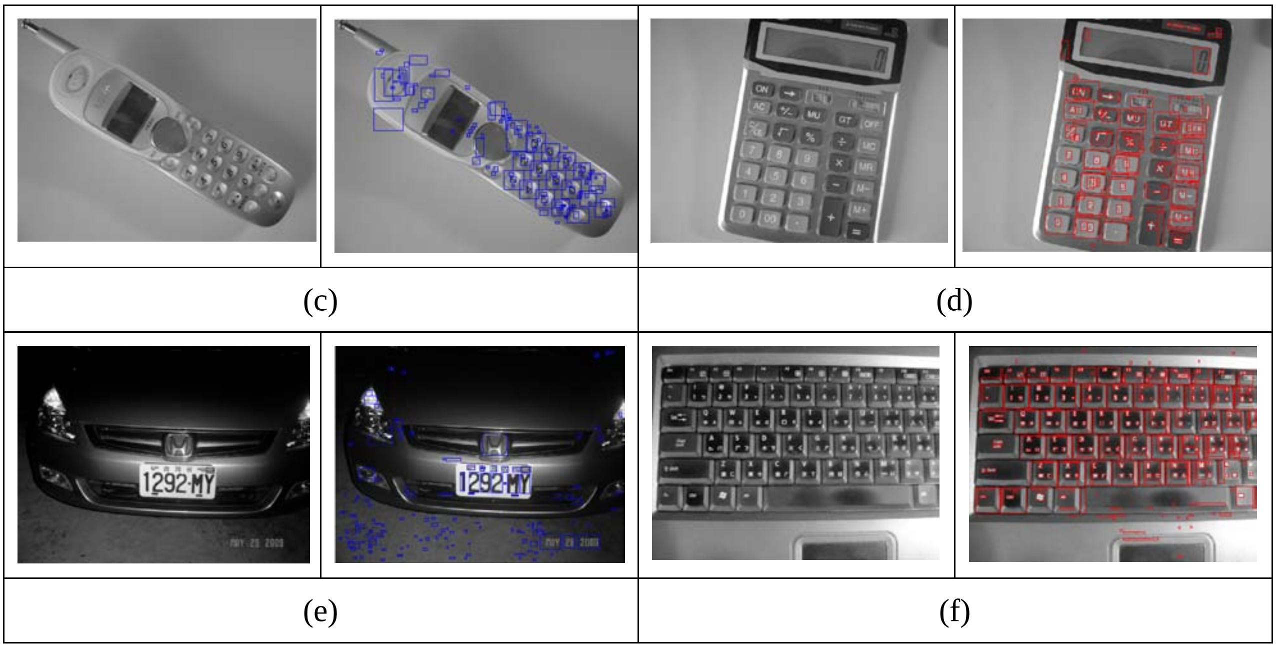


Fig.5. Several test images in the experiments and the extraction results

B. Stability to Illumination change

It is found that the proposed method is very robust and almost no change on extraction rate when the illumination is changed by constant factors,

$$\boldsymbol{I}'(x,y) = k \cdot \boldsymbol{I}(x,y) \tag{8}$$

where k is a constant and must be a reasonable value to prevent the images fall into conditions of saturation or zero intensity. It also means that the rate of success extraction will not be decreased by a DC change in illumination.

For the non-DC change of illumination, we multiply each image by four directional light sources L_1 to L_4 to imitate the responses under different illumination

$$L_1(x,y) = (x + y + 10)/(L + W + 10)$$
$$L_2(x,y) = ((W - x) + y + 10)/(L + W + 10)$$
$$L_3(x,y) = (x + (L - y) + 10)/(L + W + 10)$$
$$L_4(x,y) = ((W - x) + (L - y) + 10)/(L + W + 10)$$

where the W and L are respectively the width and length of the test image. After the light sources are added, the experiments of noise analysis in section 4.1 are repeated to simulate the effect of illumination change.

Fig.8 and Fig.9 show the result of noise analysis after illumination change by adding the light sources L_1 to L_4. We can see that the illumination change has very little influence to the result of noise analysis. It causes only about 3% drop on the TPR.

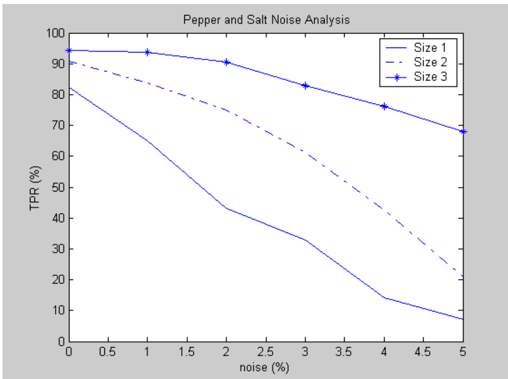


Fig.6: Pepper and salt noise analysis

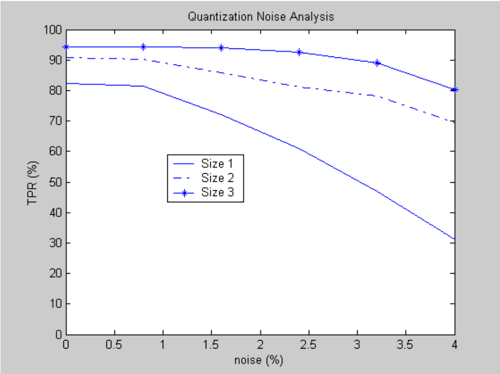


Fig.7: Quantization noise analysis

A detail analysis indicates that insufficient gray-level depth makes some edges disappeared under illumination change and becomes the major reason for TPR drop. In other words, if the gray-level depth is enough, the proposed method is very robust against illumination change.

V. Conclusions

A method to extract isolated characters is presented using Difference-of-Gaussian function. The method incrementally convolves the input image with different scale Gaussian functions and minimizes the computations in high scale images by means of sub-sampling. The characters are found by connected components analysis on the Difference-of-Gaussian image and filtered by expected sizes. The experimental results demonstrate the success of the proposed method and robustness against noise and illumination change.

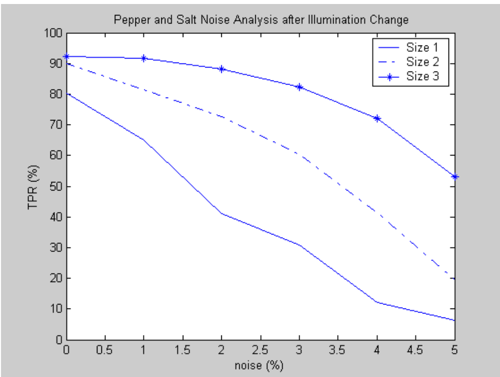


Fig.8: Pepper and salt noise analysis after illumination change

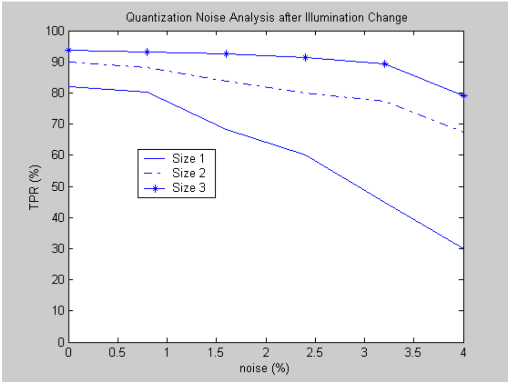


Fig.9: Quantization noise analysis after illumination change

The DOG images contain many features that are robust and distinctive for isolated characters. How to normalize the extracted characters and recognize them quickly and robustly based on the DOG image is a direction for future research.

ACKNOWLEDGEMENT

This research was supported by a grant provided by National Science Council, Taiwan, R.O.C.(NSC 93-2213-E-009-133-)

REFERENCES

[1]. Atallah AL-Shatnawi and Khairuddin Omar., "Methods of Arabic Language Baseline Detection – The State of Art," *IJCSNS*, vol. 8, no.10, Oct 2008

[2]. Liana, M & Venu, G., "Offline Arabic Handwriting Recognition: A Survey." *Transactions on Pattern Analysis and Machine Intelligence, IEEE*, vol. 28, no.5, pp712-724, May 2006

[3]. Zunino, R.; Rovetta, S., "Vector quantization for license-plate location and image coding." *Transactions on Industrial Electronics, IEEE*, vol. 47, no.1, pp159 - 167, Feb 2000, DOI: 10.1109/41.824138

[4]. Kwan, H.K., "Multilayer recurrent neural networks [character recognition application exam-

ple].” *The 2002 45th Midwest Symposium on*, vol. 3, pp 97-100, 4-7 Aug. 2002, ISBN: 0-7803-7523-8, INSPEC: 7736581

[5]. Wu-Jun Li; Chong-Jun Wang; Dian-Xiang Xu; Shi-Fu Chen., “Illumination invariant face recognition based on neural network ensemble.” *ICTAI 2004*., pp 486 - 490, 15-17 Nov. 2004, DOI: 10.1109/ICTAI.2004.71

[6]. Wang, S.-Z.; Lee, H.-J., “A Cascade Framework for a Real-Time Statistical Plate Recognition System,” *Transactions on Information Forensics and Security, IEEE*, vol. 2, no.2, pp. 267 - 282, DOI: 10.1109/TIFS.2007.897251, June 2007

[7]. N. Otsu, “A Threshold Selection Method from Gray-Level Histograms.” *Transactions on Systems, Man and Cybernetics, IEEE*, vol. 9, no.1, pp62-66, Jan. 1979, ISSN: 0018-9472, DOI: 10.1109/TSMC.1979.4310076

[8]. M.J. Taylor, A. Zappala, W.M. Newman, and C.R. Dance, “Documents through Cameras,” *Image and Vision Computing*, vol. 17, no. 11, pp. 831-844, 1999

[9]. M.J. Taylor, A. Zappala, W.M. Newman, and C.R. Dance, “Geometric Rectification of Camera-Captured Document Images,” *Pattern Analysis and Machine Intelligence, IEEE*, vol. 30, no. 4, pp. 591-605, ISSN: 0162-8828, INSPEC: 9902752, DOI: 10.1109/TPAMI.2007.70724, Apr. 2008

[10]. S. Kim, D. Kim, Y. Ryu, and G. Kim, “A Robust License Plate Extraction Method Under Complex Image Conditions,” *ICPR’02*, Quebec City, Canada, vol. 3, pp. 216-219, Aug. 2002.

[11]. David G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, 60, 2 , pp. 91-110, 2004

[12]. David G. Lowe, "Object recognition from local scale-invariant features," *International Conference on Computer Vision*, Corfu, Greece (September 1999), pp. 1150-1157.

[13]. Witkin, A.P. "Scale-space filtering," *International Joint Conference on Artificial Intelligence*, Karlsruhe, Germany, pp. 1019-1022, 1983

[14]. Koenderink, J.J. "The structure of images," *Biological Cybernetics*, 50:363-396, 1984

[15]. Lindeberg, T. "Scale space theory: A basic tool for analyzing structures at different scales." *Journal of Applied Statistics*, 21(2):224-270, 1984

[16]. Mikolajczyk, K. "Detection of local features invariant to affine transformations." *Ph.D thesis, Institut National Polytechnique de Grenoble, France*, 2002

[17]. Michael W., Davidson, Mortimer Abramowitz, "Molecular Expressions Microscopy Primer: Digital Image Processing - Difference of Gaussians Edge Enhancement Algorithm", *Olympus America Inc., and Florida State University*

[18]. Smith, Julius O. Spectral Audio Signal Processing, October 2008 Draft, http://ccrma.stanford.edu/~jos/sasp/, online book, accessed 22 June 2009.