Journal of Computer and Communications, 2015, 3, 25-32
Published Online November 2015 in SciRes. http://www.scirp.org/journal/jcc
http://dx.doi.org/10.4236/jcc.2015.311005

An Improved Histogram Equalization Method in the Traffic Monitoring Image Processing Field

Jia Shi, Kejian Yang

Department of Computer Science and Technology, Wuhan University of Technology, Wuhan, China
Email: 838357982@qq.com, ykjcgs@whut.edu.cn

Received September 2015

Abstract

To process the traffic monitoring image, a local Histogram Equalization method based on fuzzy mathematics was proposed in this paper. In this paper, firstly, we define a function to measure the similarity degree of two images. Then, a suitable Gaussian fuzzy distribution function was chose to generate a 3 × 3 matrix of influential factors. In order to reduce the artificial boundaries, we combined the 3 × 3 influential matrix with a 3 × 3 smooth filter matrix to get the final smooth-influence matrix. Finally, the smooth-influence matrix was used to process the center block image. The simulation results demonstrated that the proposed method can reduce time consumption while improving the image contrast and can get satisfactory results.

Keywords

Image Process, Histogram Equalization, Fuzzy Mathematics, Traffic Monitoring Image

1. Introduction

With the rapid development of IoT (Internet of thing), traffic monitoring technology has been more and more widely used. But due to various reasons, the traffic monitoring image cannot be applied directly. Among them, the device, environments, and the intensity of light are all the factors that influence the image quality. In order to make secondary use, we have to process the image and enhance the image contrast. As an important step in image processing, contrast enhancement has becoming more widely used.

Currently, there are many methods of contrast enhancement; among them, histogram equalization is a well-known method because of its less complexity and high efficiency [1] [2]. Generally, histogram equalization can be divided into two categories: one is global histogram equalization; the other is local histogram equalization. In global histogram equalization, the transformation function is the cumulative distribution function of the whole input image. However, in the global histogram, because of the transformation process of the image information application in each pixel, the processed image may lose a lot of local details. Especially when the gray level of the input image gathers in the left part, the gray level of the output image will gather in the right part. And the processed image shows a blenching effect and has an unnatural look. In contrast, the local histogram equaliza-

How to cite this paper: Shi, J. and Yang, K.J. (2015) An Improved Histogram Equalization Method in the Traffic Monitoring Image Processing Field. *Journal of Computer and Communications*, **3**, 25-32. http://dx.doi.org/10.4236/jcc.2015.311005

tion used a sliding window method to produce an enhanced contrast effect but the process is complex and time-consuming.

In recent years, many researchers have tried different ways to improve the traditional histogram equalization algorithm, and have acquired some great achievements. Among them, BHE (bi-histogram equalization) method firstly divided the input image gray level into two parts by the average gray value [3], and then use HE algorithm in each part independently. The results showed that BHE method substantially eliminates the excessive bleaching phenomenon of the image, and retains an average luminance; however some minor details still have not been enhanced. To compensate for this defect, the local histogram equalization came into being. AHE (adaptive histogram equalization) method divides the image into a series of sub-blocks, then HE method is implemented in each sub-block [1], meanwhile the bi-linear interpolation is used to eliminate the artificial boundaries between sub-blocks. Like AHE method [4], BOHE (over-lapped sub-block histogram equalization) method divides the image into a series of rectangular sub-blocks, and then moves the central sub-block one pixel by one pixel. This method can acquire a perfect processed image, but it is too complex to be applied in reality. Therefore, POSHE (partially over-lapped sub-block histogram equalization) method was proposed to improve the BOHE method [4], this method uses a low pass filter to generate the transformation function of central block, the simulation results show that it has increased the time efficiency by 100 times than that of the BOHE method.

To process the traffic monitoring image [5], there are three aspects need to be taken into consideration. Firstly, most parts of the image are black because of the asphalt pavement. Therefore, it is necessary to prevent the bleaching phenomenon of the processed image. Secondly, in order to ensure the time efficiency, the proposed method should not be too complex. Thirdly, in order to ensure the method to be processed automatically, it should not have too many arguments. From the above three aspects, we proposed our method. Firstly, the image was divided into a series of sub-blocks, and then according to the similarity function and Gaussian fuzzy distribution function, a 3 × 3 influential factors matrix was generated to show the influence degree of the peripheral sub-blocks on the central block. Moreover, the 3 × 3 influential factors matrix was combined with a 3 × 3 smoothing filter matrix to reduce the artificial boundaries. Finally, the combined smooth-influence matrix was used to process the center sub-block.

The rest of this paper is organized as follows: Section 2 is to describe our method, Section 3 is to show the simulation and comparison results, Section 4 is to analyze our method, and Section 5 is to conclude this whole paper.

2. Proposed Methodology

2.1. Histogram Equalization

During the process of Histogram Equalization, the discrete transformation function is as follows [1]:

$$s_k = T(r_k) = \sum_{j=0}^{k} p_r(r_j) \qquad (k = 0,1,2,\cdots,L-1) \tag{1}$$

where,

$p_r(r_j) = \frac{n_j}{n}$; n is the total number of pixels of the input image;

n_j is the number of pixels of r_j gray level; L is the total gray levels of input image;

r_k is the gray level of input image; s_k is the gray level of output image.

In order to simplify the analysis the continuous form of formula (1) is figured out as follows [1]:

$$s = T(r) = \int_0^r p_r(w)dw \tag{2}$$

As the Formula (2) shows, there are differential relationships between $T(r)$ and $P(r)$. When the value of $P(r)$ is large, the growth rate of $T(r)$ is high, so the intensive gray level will be stretched. When the value of $P(r)$ is small, the growth rate of $T(r)$ is low, so the sparse gray level will be compressed. In this way, the gray level of the whole image will be distributed uniformly, and the image will be enhanced.

2.2. Similarity Function

We use the difference of the pixels amount of the same gray level between two sub-block images to measure the similarity degree of them. In order to simplify the process of the measurement, firstly, the normalization preprocessing of each gray level was implemented. The similarity degree varies between 0 and 1, especially, when the two sub-block images are the same, the similarity degree is 1. Therefore, the similarity function is defined as follows:

$$S(image_a, image_b) = 1 - \sum_{i=0}^{L-1} \left| \frac{n_{ai}}{n_a} - \frac{n_{bi}}{n_b} \right| = 1 - \sum_{i=0}^{L-1} \left| p_{ar}(r_i) - p_{br}(r_i) \right| \quad (3)$$

where,

$image_a, image_b$ represent the two sub-block images;

n_a, n_b are the total pixels number of the two sub-block images;

L is the total gray level of the two sub-block images;

n_{ai}, n_{bi} represent the *i*th gray level pixel of sub-block $image_a$ and $image_b$ separately;

$P_{ar}(r_i), P_{br}(r_i)$ represent the ratio of gray level r_i of sub-block $image_a$ and $image_b$ separately after the normalization preprocessing.

Defined in this way, the similarity function not only can decrease the computational complexity but also is suitable for sub-blocks of different sizes [6].

2.3. The Calculation of Influential Factor Matrix

The domain of the similarity degree is defined as $X = [0,1]$. Fuzzy set A represents “low-influence degree” and the influential factor is 1/16; Fuzzy set B represents “mid-influence degree” and the influential factor is 1/8; Fuzzy set C represents “high-influence degree” and the influential factor is 1/4.Because only when the similarity degree is suitable, the value of the influence degree can be maximum. Therefore, Fuzzy distribution function of the intermediate type should be assigned as the membership function. In this paper, we choose Gaussian function as the membership function of Fuzzy set A, B and C. The formulas of these three functions are listed as follows [5] [7]:

$$A(x) = \exp\left\{ -\left(\frac{x - a_1}{\sigma_1} \right)^2 \right\} \quad x \in [0,1] \quad (4)$$

$$B(x) = \exp\left\{ -\left(\frac{x - a_2}{\sigma_2} \right)^2 \right\} \quad x \in [0,1] \quad (5)$$

$$C(x) = \exp\left\{ -\left(\frac{x - a_3}{\sigma_3} \right)^2 \right\} \quad x \in [0,1] \quad (6)$$

where,

x is the similarity of the two sub-block images;

a_1, σ_1 are the arguments of membership function $A(x)$;

a_2, σ_2 are the arguments of membership function $B(x)$;

a_3, σ_3 are the arguments of membership function $C(x)$.

After several tests, the arguments of the three membership functions are determined, on the condition that $a_1 = 0.15$, $a_2 = 0.45$, $a_3 = 0.75$, and $\sigma_1 = \sigma_2 = \sigma_3 = 1$, the processed image is the best.

In order to simplify the description, we number the sub-blocks of the split image, which can be shown in **Figure 1**. According to the three membership functions and the principle of maximizing membership, the influential factors from the around sub-blocks to the central block is calculated (that is to calculate the influential factors form sub-blocks A, B, C, D, E, F, G, H to the center block), and then the influential factors matrix $M_{\text{influence}}$ can be acquired. Because the similarity degree of the center block is 1, the center value of $M_{\text{influence}}$ is always 1/4 of “high-influence degree”.

2.4. The Combination of Smooth-Influence Matrix

Because we have divided the input image into a series of sub-blocks before the processing, there are some in

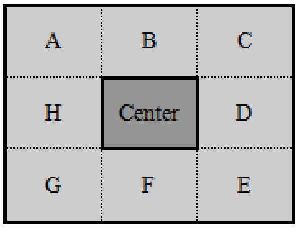


Figure 1. The graph of the split image.

evitable artificial boundaries in the processed image, In order to decrease the computational complexity and reduce the quantity of artificial boundaries, the 3 × 3 smoothing filter matrix is combined with $M_{\text{influence}}$ to acquire the final smooth-influence matrix to process the image. In this way, the artificial boundaries will not be processed independently.

Define operator $\oplus$ to represent the combination of matrix, which can be shown as follows:

$$M_A \oplus M_B = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \oplus \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} = \begin{bmatrix} \frac{a_{11}+b_{11}}{sum} & \frac{a_{12}+b_{12}}{sum} & \frac{a_{13}+b_{13}}{sum} \\ \frac{a_{21}+b_{21}}{sum} & \frac{a_{22}+b_{22}}{sum} & \frac{a_{23}+b_{23}}{sum} \\ \frac{a_{31}+b_{31}}{sum} & \frac{a_{32}+b_{32}}{sum} & \frac{a_{33}+b_{33}}{sum} \end{bmatrix} \tag{7}$$

where,

$sum = \sum_{i=1}^{3} \sum_{j=1}^{3} (a_{ij} + b_{ij});$

$a_{11} \cdots a_{33}$ are the elements of M_A;

$b_{11} \cdots b_{33}$ are the elements of M_B.

According to the operator defined in Formula (7), the 3 × 3 smoothing filter matrix is combined $M_{\text{influence}}$ with to acquire the final smooth-influence matrix.

$$M_{\text{smooth-influence}} = M_{\text{influence}} \oplus M_{\text{smooth-fliter}} \tag{8}$$

where,

$M_{\text{smooth-fliter}} = \frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$ is the 3 × 3 smoothing filter matrix;

$M_{\text{influence}}$ is the influential factor matrix;

$M_{\text{smooth-influence}}$ is the smooth-influence matrix.

According to the smooth-influence matrix, we compute the histogram of the center block and get the transformation function to process the center block image. In this way, the process can retain the information of the center block and peripheral sub-blocks, meanwhile reducing the artificial boundaries.

3. Simulation and Comparison

The proposed method can be divided into the following steps [8] [9]:

Step 1: Color transformation (RGB- > HSV);

Step 2: Extract the V component, and divide it into a series of sub-blocks;

Step 3: Compute $M_{\text{smooth-influence}}$, and get the transformation function of the center block;

Step 4: Implement the equalization process of the center block;

Step 5: Revert the V component and color transformation (HSV- > RGB), then output the image.

In the proposed method, the processing of each sub-block needs the auxiliary processing of the 8 peripheral blocks. So we have to do some additional processing on the edge blocks. The results of additional processing is showed in **Figure 2**.

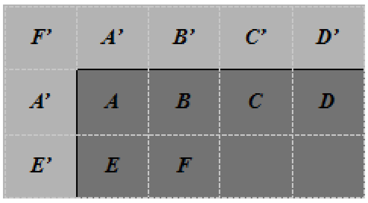


Figure 2. The graph of additional process.

For the edge blocks, we copy their reflectional symmetrical blocks as the filled blocks (as the **Figure 3** shows, block A and block A', block B and block B' and so on); for the corner blocks, we copy their rotational symmetrical blocks as the filled blocks (As the **Figure 3** shows, block F and block F').

Then two traffic monitoring images of low contrast were chose (such as a dawn image and a fog image) to do experiments:

We have tested HE, BHE, AHE and POSHE methods respectively. The results of the experiment are showed in **Figure 3**. And the corresponding histograms are showed in **Figure 4**. Form **Figure 3**, we can know that the results of local histogram equalization get by AHE and POSHE methods are better than the results of global histogram equalization get by HE and BHE methods. As the **Figure 3** shows, the gray level of the original image gathers in the left part, so the result of HE method shows a bleaching effect and the histogram of the processed image gathers in the right part. The BHE method doesn't do enough contrast enhancements on the image. The processed images of AHE and POSHE methods have a great enhancement, and the gray level of the whole image distributes uniformly, meanwhile the details are retained (e.g. the wire in the top right corner of image, and the lamp of vehicle etc.). Our proposed method produced satisfactory results. On the one hand, the contrast of the processed image have been enhanced (e.g. road, vehicle and the roadside trees); on the other hand, as the histogram shows, our proposed method have stretched the contrast, and the gray level of the whole image distributes uniformly.

In order to have a more objective understanding of the proposed method, we compared HE, BHE, AHE, POSHE and the proposed method from following three aspect: pre-processing, artificial boundaries process and average time. The results of the comparison are shown in **Table 1**.

As **Table 1** shows, the local methods such as AHE and POSHE need to eliminate artificial boundaries because they divide the image before processing .So the local methods are slower than the global methods such as HE and BHE. The execution time of the proposed method is between BHE and POSHE, which is about 0.2705 seconds. Its execution speed is very fast. That is to say it can be applied to embed device and real-time processing.

4. Performance Evaluation

4.1. Analysis of Time Efficiency

In the process of our method, the peripheral sub-blocks A, B, C, D, E, F, G and H affect the histogram equalization of the center block. The main time consumption concentrates on the computation of $M_{\text{smooth-influence}}$ because the computation of the three membership functions includes exponential operation and square operation. For each block, the transformation function can be written as:

$$s_k = T_{center}(r_k) = \sum_{i=1}^{3}\sum_{j=1}^{3} M_{\text{smooth-influence}}(i,j) \times T(r_k)_{i,j} \tag{9}$$

where,

$M_{\text{smooth-influence}}(i,j)$ is the corresponding element of $M_{\text{smooth-influence}}$

$T(r_k)_{i,j}$ is the corresponding transformation function of each peripheral sub-block.

If the image is divided into $n \times m$ sub-blocks and each block have $size$ pixels, we can suppose the execution time of similarity function as $T_a(size)$, the execution time of membership function as T_b and the execution time of HE for each block as $T_c(size)$. According to Formula (9), for each block, the similarity function needs to compute 8 times, the membership function needs to compute 24 times, and the transformation function needs to compute 1 time.

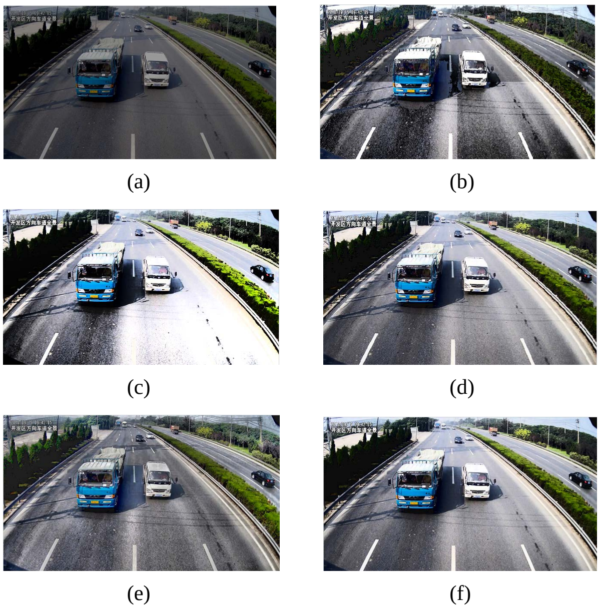

(a) (b) (c) (d) (e) (f)

Figure 3. Simulation results. (a) Original; (b) Proposed; (c) Heed; (d) BHEed; (e) AHEed; (f) POSHEed.

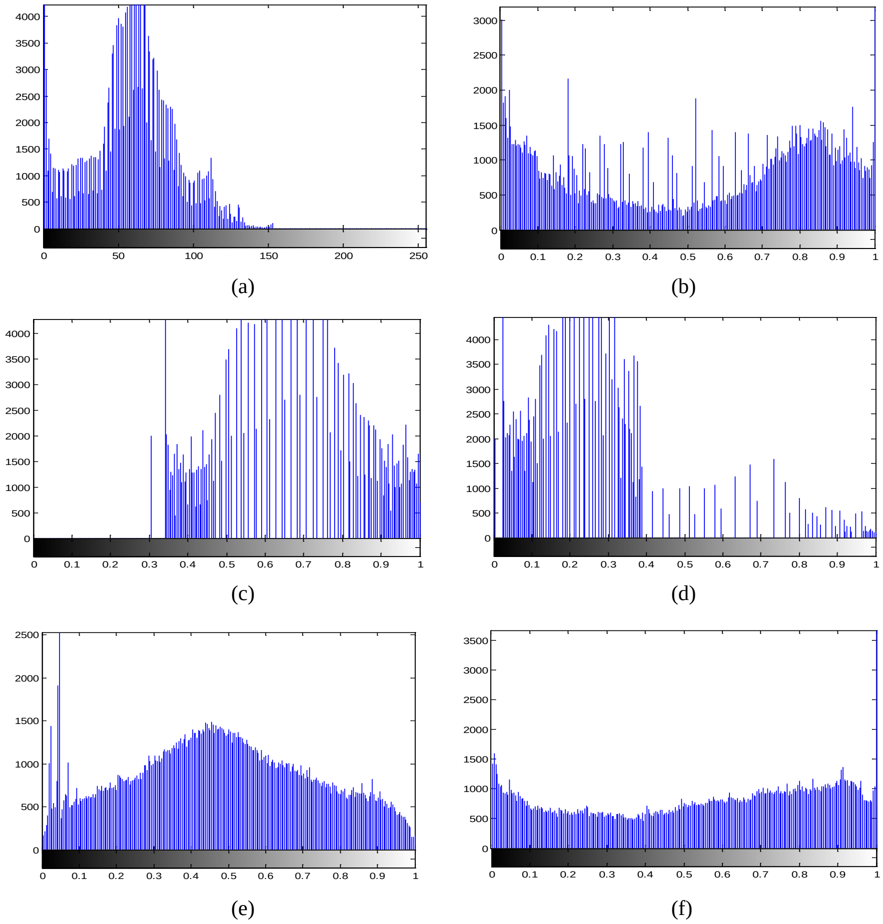


Figure 4. Histogram of **Figure 3**. (a) Original; (b) Proposed; (c) Heed; (d) BHEed; (e) AHEed; (f) POSHEed.

Table 1. The result of comparison.

Method	Pre-processing	Artificial boundaries process	Average time
Proposed	Divide	Using smooth filter	0.270552
HE	NULL	NULL	0.256914
BHE	NULL	NULL	0.256914
AHE	Divide	Bi-linear interpolation	0.293260
POSHE	Divide	Detect and eliminate	0.289349

a. Experimental environment: Matlab 7.06, Intel (R) Celeron (R) CPU, 4.00 GB RAM; b. Experimental data: 720 P traffic monitoring images.

So the total execution time of processing an image can be written as:

$$T = n \times m \times \left(8 \times T_a(size) + 24 \times T_b + T_c(size)\right) \tag{10}$$

From the Formula (10), we know the total execution time is proportional to the number of sub-blocks, that is, when the value of $n \times m$ is increased, the total execution time is also increased; meanwhile the total execution time is also proportional to the size of sub-blocks, that is, when the size of a sub-block is increased, the total execution time is also increased; However, the number of sub-blocks is inversely proportional to the size of sub-blocks, because when a fixed size image is divided into $n \times m$ sub-blocks, the more sub-blocks you divided, the less size of sub-blocks you will get. So in order to reduce the total execution time, the main problem is to balance the number of sub-blocks and the size of sub-blocks. Firstly, an image was divided into different sizes, then our method was tested and the total execution time was recorded.

As the **Figure 5** shows, we know when the number of sub-blocks is between 4 × 6 and 8 × 8, the total execution time is relative short; when the number of sub-blocks is small, the total execution time is a little shorter than that of HE method; when the number of sub-blocks is large, the total execution time is very long, because the computation of membership function is very time-consumed. For the monitoring image of 1280 × 720 pixel image, the most suitable segmentation approach is dividing the image into 4 × 6 or 6 × 6 sub-blocks.

4.2. Analysis of Membership Function

In this paper, there are two reasons why do we choose Gaussian function as the membership function. On one hand, the Gaussian function is suitable for describing the phenomenon of nature, human society, psychology and education; On the other hand, the Gaussian function's arguments are relatively simple compared to the other intermediate types of distribution fuzzy function, it only needs two arguments to control the position and shape of the function.

Form the subject point, on the condition that the shape controlling parameter σ is static, when a_1, a_2, a_3 are small and concentrated, the influences of the peripheral sub-blocks on the central sub-block are great, and the processing results of the central sub-block will loss too much details. When a_1, a_2, a_3 are large and concentrated, the influences of the peripheral sub-blocks on the central sub-block are small, and the image processing results will show obvious artificial boundaries. Therefore, arguments a_1, a_2, a_3 should distribute on the coordinate axis uniformly one by one, then the influences of the around sub-blocks on the central sub-block can be better measured.

As the **Figure 6** shows, three different groups of arguments was chose to experiment. The arguments of (b) distribute on the axis uniformly, the result looks better. The arguments of (c) are small and concentrated, the results lost a lot of details (e.g. the goods of truck). On contrast, the arguments of (d) are large and concentrated, there are obvious artificial boundaries in the results.

From the experimental results, we know the more suitable arguments are $a_1 = 0.15, a_2 = 0.45, a_3 = 0.75$, and $\sigma_1 = \sigma_2 = \sigma_3 = 1$.

5. Conclusion

With the rapid development of traffic monitoring technology, the conventional image processing method is no longer suitable for the traffic monitoring images. In this paper, three features of traffic monitoring images have

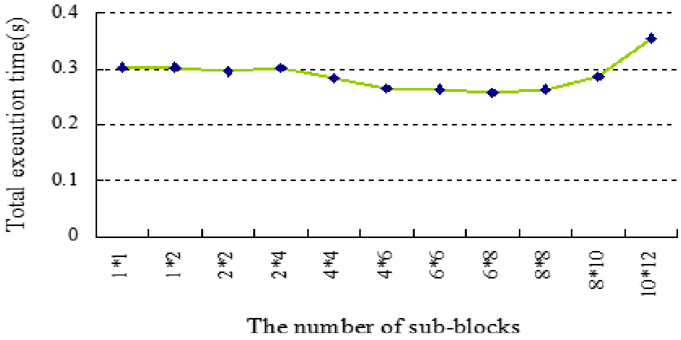
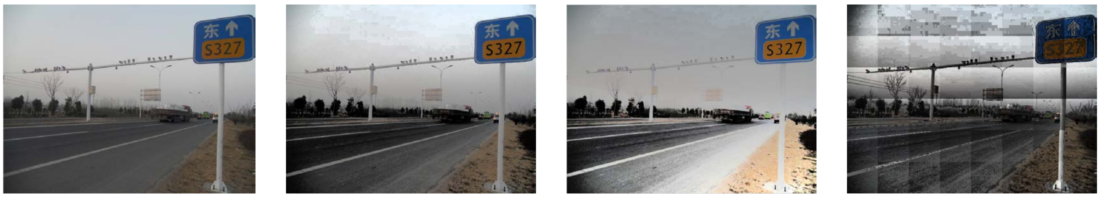


Figure 5. Execution time graph.



(a) (b) (c) (d)

Figure 6. Experimental results with different arguments. (a) Original image; (b) $a_1 = 0.15, a_2 = 0.45, a_3 = 0.75$; (c) $a_1 = 0.75, a_2 = 0.85, a_3 = 0.95$; (d) $a_1 = 0.15, a_2 = 0.20, a_3 = 0.25$.

been analyzed. On the basis of these features, a local histogram equalization method based on fuzzy mathematics was proposed. This method is suitable for contrast enhancement of traffic monitoring images. The contrast of the processed images is stretched, meanwhile the time consumption of our method is relatively low. In order to reduce artificial boundaries, we use a 3 × 3 smooth filter. But when observed carefully, some slight artificial boundaries will be found. Because the artificial boundaries and the image content are closely related, the method of reducing artificial boundaries completely and quickly is more challenging. So one of the focuses of future research is to eliminate the artificial boundaries.

References

[1] Conzalez, R.C. and Woods, R.E. (2001) Digial Image Processing. Prentice-Hall, Inc., New Jersey.

[2] Gonzalez, R.C., Woods, R.E. and Eddins, S.L. (2004) Digital Image Processing Using MATLAB. Prentice Hall.

[3] Kim, Y. (1997) Contrast Enhancement Using Brightness Preserving Bi-Histogram Equalization. *IEEE Transactions on Consumer Electronics*, **43**, 1-8. http://dx.doi.org/10.1109/30.580378

[4] Youn, J., Sup, K.L. and Hwang, K.S. (2001) An Advanced Contrast Enhancement Using Partially Overlapped Sub-Block Histogram Equalization. *IEEE Transactions Circuits and Systems for Video Technology*, **11**, 475-484. http://dx.doi.org/10.1109/76.915354

[5] Zadeh, L.A. (2005) Fuzzy Set and Fuzzy Information-Granulation Theory. Beijing Normal University.

[6] Tan, K.K. and Oakley, J.P. (2005) Physics-Based Approach to Color Image Enhancement in Poor Visibility Conditions. *Journal of the Optical Society of America A*, **18**, 2460-2467. http://dx.doi.org/10.1364/JOSAA.18.002460

[7] Wang, J.-H. and Wang, X.-Z. (2004) An Initial Analysis of Parametric Sensitivity to Fuzzy Extension Matrix. *Proceedings of* 2004 *International Conference on Machine Learning and Cybernetics*, **3**, 1884-1888. http://dx.doi.org/10.1109/ICMLC.2004.1382085

[8] Meylan, L. and Süsstrunk, S. (2004) Color Image Enhancement Using a Retinex-Based Adaptive Filter. *Proceedings of IS&T Second European Conference on Color in Graphics*, *Image*, *and Vision* (*CGIV* 2004), **2**, 359-363.

[9] Terai, Y., Goto, T., Hirano, S. and Sakurai, M. (2009) Color Image Contrast Enhancement by Retinex Model. *The* 13*th IEEE International Symposium on Consumer Electronics* (*ISCE*), 392-393.