Recognition of Flow Readings for Float Flowmeters Based on Digital Image Technique

Xiao-xuan Cao, Yu-ming Shen
University of Shanghai for Science and Technology, Shanghai, P.R.China
516, Jungong Road, Shanghai, China, 200093,
Phone:0086-21-55274825,FAX:0086-21-55274825,E-mail:ym-shen@usst.edu.cn

Abstract: Put forward a way to use MATLAB when processing the image of the glass flowmeter in order to automatically get its float lever. In the process of image collection, we study the phenomenon of uneven background brightness that caused by illumination, and decide to use diffuse. First, we use MATLAB image processing toolbox, such as edge detection, filtering, thresholding, expansion, thinning, etc, and then add our own algorithms according to the specific image feature, to detect the float position.

Keywords: image processing, MATLAB, flowmeter

1. Introduction

In industrial production, we often need to know various fluid flow such as air and water, and use it as the process control parameters, as a result, flow measurement devices have been invented. Flowmeters play a very important role in low-velocity flow measurement, especially in the case of tiny flows, and low Reynolds number.

eople spend a lot of time researching on the flow equations of flowmeters, and the correction of flow measurement error in non-standard state. However, there is seldom someone concerned about the problem of the automatic measurement of float position. Traditional flowmeters that tested by hand, increase not only work time, but also labor intensity of surveyors. With the rapid development of science and technology, computer technology has been gradually integrated into manufacture, and its combination with measurement technology and instrument makes many intelligent automatic meter to come true. Most of these instruments increase the automatic reading function, and improve the efficiency. The advantages of digital image such as containing large amount of information, high transmission speed and long distance, become main method for people to obtain and make use of information. Image processing technology is relatively mature in recognition of the scale and the pointer, so its feasible to use Image processing technology on flowmeter identification.

2. System Components

The system's hardware component is shown in Figure 1. First, we use strip light treated by diffuse to evenly illuminate the glass rotor flowmeter. By this means, we can prevent the effect of uneven light and reflective glass, and effectively inhibit the generation of spot. Then, we collect the flowmeter images by the MVC - II - 1MM camera, and the image information converted to computer signals will be sent to a computer through the USB interface. After that, high resolution(1280 * 1024) BMP files are formed. Finally, we complete the process of image processing, float lever recognition and calculation, and display the measuring results. In the image, nearer the distance of the camera and the flowmeter is, higher accuracy the pictures are. But if their distance is closer than the focus, we can only get a blurred image, and it will affect the subsequent processing. So, we must make sure both to shooting the whole flowmeter and that the shooting distance is larger than the focus.

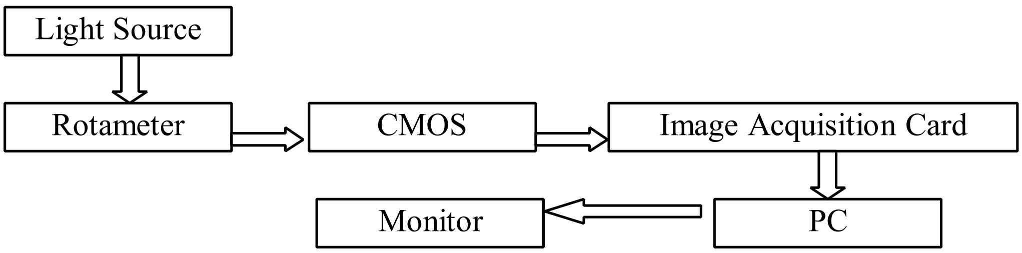


Fig.1 Measuring system diagram

3. Flowmeter Image Processing and Float Lever Recognition

Because we want to identify the volatile float in the glass, the camera should shoot the whole flowmeter. The image is shown in Fig2. The computer runs the entire system's core-recognition algorithm. It contains four parts of pretreatment, scale recognition, float recognition and reading.

3.1 Automatic Interception of Meter's Glass Image

In the experiment, the position of the flowmeter is not constant in the original image, so we first need to locate glass. After the operation of canny edge detection, expansion, filling and smooth, binary image of Fig3 is got. In this issue, because the scale lines are large and small, Canny edge detector is used.

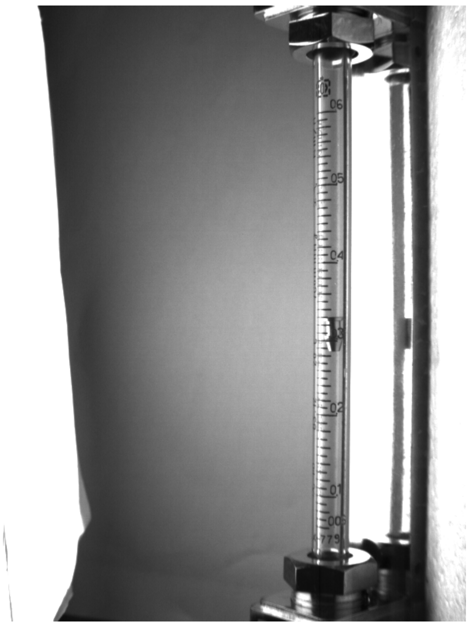


Fig.2 Original flowmeter image

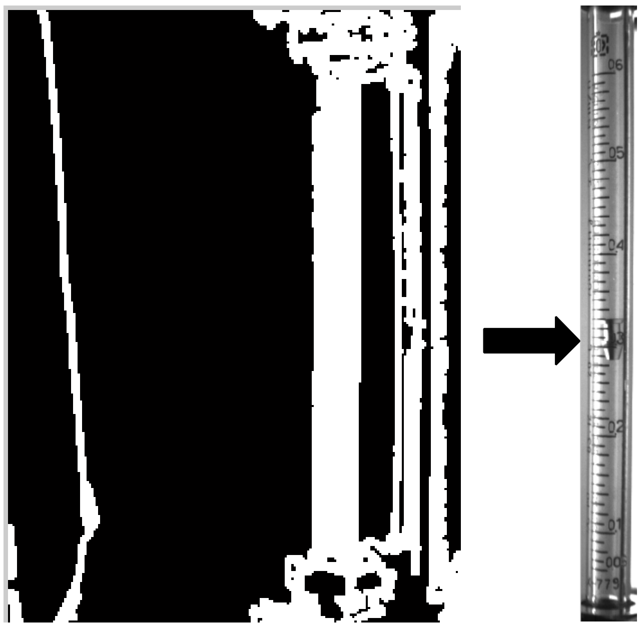


Fig. 3 Experimental images

3.1.1 Canny Edge Detection Operators

Canny basic ideas:1, Use gaussian filter to reduce smooth the image, and reduce noise. 2, Calculate the image gradient by gauss filter's derivatives. 3, Look for local maxima of the image gradient. That is to say, we get $I(i,j)*G_a(i,j)$ by gaussian filtering. The image edge points in the direction of A_a is to obtain the maximum points M_a locally.

$$M_a = \| I(i,j) * \nabla G_a(i,j) \| \quad (1)$$

$$A_a = \frac{I(i,j) * \nabla G_a(i,j)}{\| I(i,j) * \nabla G_a(i,j) \|} \tag{2}$$

where

M_a: modulus of gradient vector; A_a: direction of gradient vector

3.1.2 Automatic Interception Algorithm

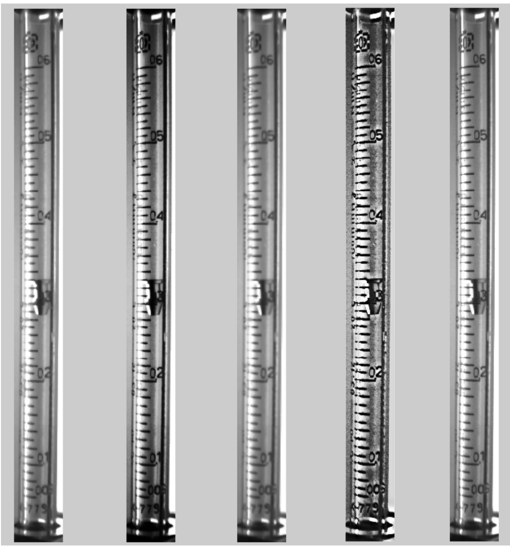
We know from the graph that most of the pixels representing the glass are '1'. Cut the middle 1/3 to calculate. Suppose that the image's size is M*N pixels, I(i,j) stands for each pixel. Calculate every column's sum, and get the array called SUM1 of 1*N structure.

$$SUM1(j) = \sum_{i=1}^{M} I(i,j) \tag{3}$$

$$SUM2(j,i) = \sum_{j=i}^{i+N/3-1} SUM1(j)/j \tag{4}$$

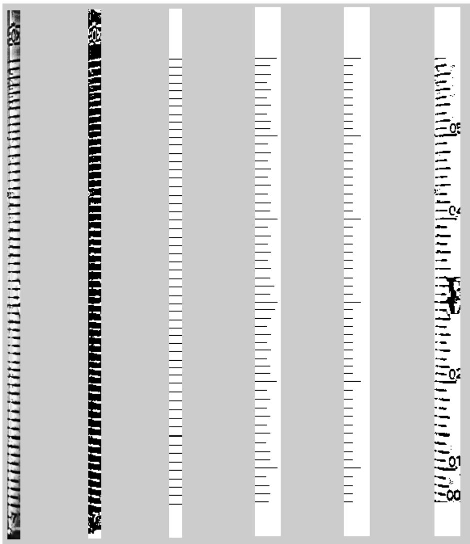
According to the Fig3(a), when i is the left side of the glass, and j is the glass's width, the value that divides sum of all the pixels in the matrix by j has the most maximums. That means column i of SUM2 has most maximums, and its number is j. According to these parameters, original image is intercepted, and redundant data is removed. This is how to get flowmeter pictures Fig3(b). This method seldom requires the relative position of camera and flowmeter, therefore, it is very practical.

3.2. High-frequency Emphasis Filter



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

Fig.4 Filter results comparison



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

Fig.5 Calibration analysis

As the low-pass filtering makes the image become blurred, high-pass filtering makes it clear (sharp), by weakening the low-frequency of Fourier transform when keeping the high-frequency. After this filtering, both the low-frequency and high frequency components have been enhanced, but the low component increases a lesser extent, which makes the edges more clearly. The transfer function of this filter is as follows.

$$H_{hfe}(u,v) = a + b^* H_{hp}(u,v) \tag{5}$$

where

a: offset; b: multiplier; $H_{hp}(u,v)$: transfer function

Fig4(a) uses the median filter. Fig4(b) uses a histogram equalization, Fig4(c) uses the adaptive filter, Fig4(d) uses the high frequency emphasize filter, Fig4(e) is the original figure

.

3.3. Scale Analysis

In practice, since we indicate the float lever by the relative position between the float and scale, recognition of the scale location is needed. To detect the edge of Fig4(d), we can find that there are more relatively inerratic edges in scale location, therefore a variance comparison is undertaken. Suppose that the scale begins from column j of the glass, and its width is a quarter.

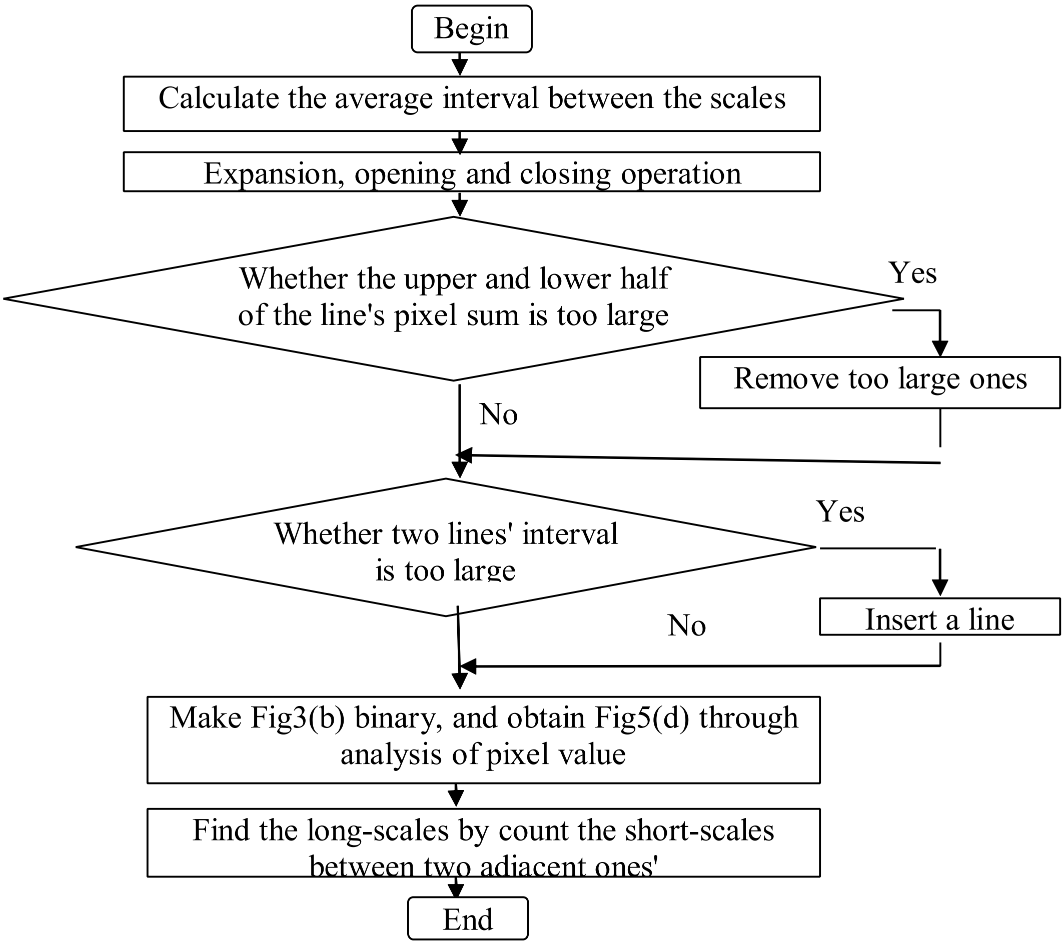


Fig.6 Scale identification flowchart

$$SUM3(j) = \sum_{i=1}^{M} I(i,j) \tag{6}$$

$$\overline{SUM3}(j) = \frac{\sum_{i=j}^{j+N/4} SUM(i)}{N/4+1} \tag{7}$$

$$VAR(j)=\sqrt{\frac{\sum_{i=j}^{j+N/4}(SUM(i)-\overline{SUM3}(i))}{N/4}} \qquad (8)$$

where

M: the glass image's height; N: the glass image's width

When VAR is minimum, j is the scale's beginning. Fig5(a) is the scale figure. Transform the picture into binary ones. Because image contrast after filtering is enhanced, we find in experiment that the threshold demand is not high. So when regular light is provided, we choose 0.2 as the threshold, and get Fig5(b). According to each line's pixel sum, we get each scale location. As each line sometimes owns more than one pixel width, we should look for the largest gray one before the next step calculation. Through analysis of the texture characteristics, Fig5(c) is got, and the process is shown in Fig6. Fig5(e) is the final scale line image.

3.4. Float Recognition

If we want to get the height of the float, we should find its edge. The gray contrast between the float of other parts is greater. Fig5(f) becomes from Fig4(d) after texture removing and binarization. Because we ask for the relative position of float lever and scales, the float height should in the scale context.

$$SUM4(i)=\sum_{j=1}^{N}\sum_{i1=i}^{L+i-1}I(i,j) \qquad (9)$$

where

L: five times of the scale interval

$$SUM5(i)=\sum_{j=1}^{N}I(i,j) \qquad (10)$$

When SUM4(i) is the minimum, row i to i+L is the float location. Suppose I1 is the lowest point of SUM4, and calculate the sum from row I1 to I1+L to find the peak values. After comparing all the peak ones, we can see that the value of float edge gray changes in the form of gradient, that is, the pixels of float edge grow out of nothing. With one peak in this condition, float height is got.

As shown in Fig 7, there are 5 peaks that may be the position of the float edge. Then, according to the edge of the float features, we search the front of the peaks. If we find zero in SUM5, the peak behind it will be the height, otherwise, the first peak point is the float height. We set this point as I2.

We get the float lever by comparing the scale and I2. This method is especially useful when we need to replace float flowmeter, because it can excuse calibration work.

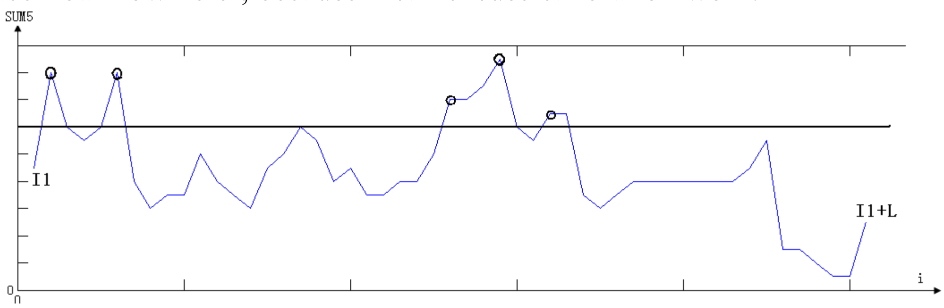


Fig.7 Analysis diagram of the pixel sum of matrix from row I1 to I1+L

4. Results

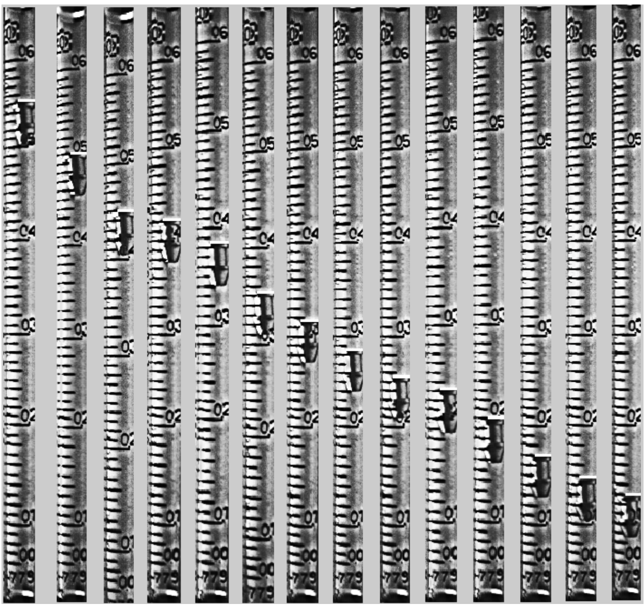


Fig. 7 Float edge detection

By using MATLB image processing toolbox function, this system automatically recognize the float lever of the flowmeter.

5. Conclusion

This paper mainly introduces a device of flowmeter image recognition. The cardinal principle is to analyze relative position of the float and scales through glass rotor flowmeter image. Experiments have proved that this method reach a 10% minimum scale line resolution. We not only reduce the workload and duty cycle, but also improve the degree of automation. The proposed algorithm in this paper can also used for other recognition of pointer and scale.

Acknowledgment

The authors are grateful to the Bureau of Standards2010, Metrology and Inspection (BSMI/MOEA), Taiwan R.O.C., for support of this research under project code 93-1403-31-H-00-00-00-24.

References

[1]Jia Yonghong. Digital Image Processing[M], Wuhan University Press, 2003-09
[2]Rafael C.Gonzalez Richard E.Woods Steven L.Eddins. Digital Image Processing Using MATLAB, Electronic Industry Press, 2005-09
[3].Pelin Gorgel, Ahmet Sertbas, Osman N. Ucan. Journal of Medical Systems[SpringLink], 2009-06
[4]Gonzalez, R.C. and Woods, R.E. [2002].Digital Image Processing, 2nd ed., Prentice Hall, Upper Saddle River, NJ.
[5]Zhang Yujin, Image Engineering, Tsinghua University Press, 2007