Autofocus

Wei-Sheng Liao (廖偉勝) and Chiou-Shann Fuh (傅楸善)

Department of Computer Science and Information Engineering

National Taiwan University, Taipei, Taiwan

r91049@csie.ntu.edu.tw and fuh@csie.ntu.edu.tw

Abstract

This paper introduces some autofocus methods. We do some experiments and implement these methods to understand their performances. Then, we discuss our experiments and the performances.

I. Introduction

The lens is the main tool which we get images. If we take a blurred picture through the lens, whatever post-processing we can not get a sharp image. Lens is like the eyes of a human, so we should adjust the lens to focus on objective precisely. What distance should we focus on? This is subjective. Some people feel this picture acceptably sharp, but some feel blurred somewhere on the picture. This is decided by what result you want, so the manual focus is a little troublesome. Most people focus because they want an acceptably clear picture. Autofocus can help them focus more easily, and they do not want to focus manually. Users just press the button to take pictures.

II. Background

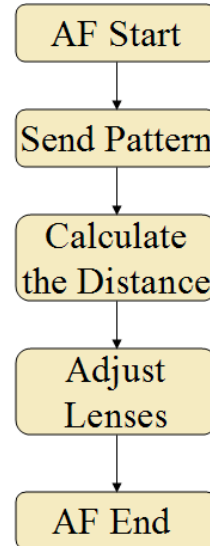Autofocus method has two types mainly. One is the active type, and the other is passive type.

Figure 1 The flow chart of the active autofocus method.

The flow chart of the active autofocus method is shown in Figure 1. When the active autofocus method starts, the camera would send patterns, for instance visible light or infrared rays. After receiving the reflected patterns, the camera would calculate the distance between the camera and the objective. According to the calculated distance, the camera adjusts the position of its lens. This type needs sensors to send patterns and receive them, so the camera needs space to place sensors. The cost of this camera would increase because of sensors. When you picture through glass, some patterns may be reflected. At this situation, the active method would calculate the wrong distance. The solution is to change the pattern that could go through glass. The choice of patterns is important. When there are many things in front of the main objective,

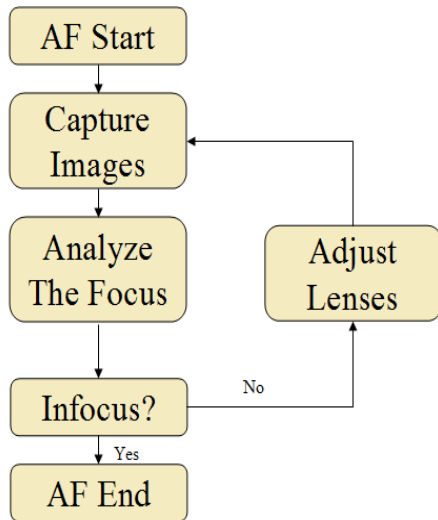this method would also calculate the wrong distance.

Figure 2 The flow chart of the passive autofocus method.

The flow chart of the passive autofocus method is shown in Figure 2. This method needs information to analyze if the position of lens is right. After the passive autofocus method starts, the camera first captures an image. Second, the camera uses some algorithms to calculate the sharpness value. (We often can not judge a picture sharpness easily. We define a value and use some algorithms to calculate sharpness of a picture. If the value is high, the picture is sharp.) According to this value, the camera uses another algorithm to judge if the lens focuses on the objective. If not, the camera goes back to the first step. The camera repeats those steps till the lens is in focus. This passive method would have trouble in lower light situation. In this situation, the sharpness value may be too low to focus all the time. The solution is to change the threshold at this situation. Some cameras would use light to assist in focusing, and some cameras would use an active autofocus method to assist, for

example Sony DSC F707. The choice of algorithms is important. Some algorithms may be fast, but they have worse focus. How do we find the balance between the speed and effect is a trade-off. People always want their cameras to work fast and have good focus. The main problem of the passive autofocus is to find an algorithm that needs less computation but has good focus.

The above two types of autofocus methods have advantages and disadvantages. The active type is often faster than the passive type because of the more computation of the passive type. The active type may require higher cost. Different levels of cameras have different cost, and the cameras can use different policies of the autofocus methods.

Here, we introduce a noun "depth of field". If a camera is focused at a specific distance, the object at the exact distance would be sharp. Things at a range near the object would be also represented sharply. The range is called depth of field. If your object is right at the exact distance, things at the 1/3 of the range in front of the object are sharp. Things at the 2/3 of the range behind the object are also sharp. Why is the depth of field produced? This uses the weakness of the eyes. When the diameter of a blurred image is smaller than 0.00357 (1/2800) inch, our human eyes would feel the blurred image sharp. The range of the depth of field is affected by some factors. We explain as follows:

1. Aperture

The larger aperture has shallower depth of field, and the smaller one has deeper depth of field. We consider an original blurred image of the object. When we change a smaller aperture, original light which would be through the lens is

blocked by the aperture. The measure of area of the blurred image would be reduced to a smaller one which eyes look as one dot. Thus the blurred image becomes sharp. The background of Figure 4 is clearer than that of Figure 3.

Figure 3 The aperture is F/2.8. (F number = focal length / aperture diameter)

Figure 4 The aperture is F/8 (smaller aperture diameter thus deeper depth of field).

2. Focal length

The longer focal length has a shallower depth of field. The shorter focal length has a deeper depth of field. We consider an original sharp image of the object. The image is formed at the focal point. When the focal length becomes longer, and the image plane is maintained, the measure of area of the original image becomes bigger. The original sharp image becomes blurred. We look at Figures 5 and 6. Figure 5 is clearer.
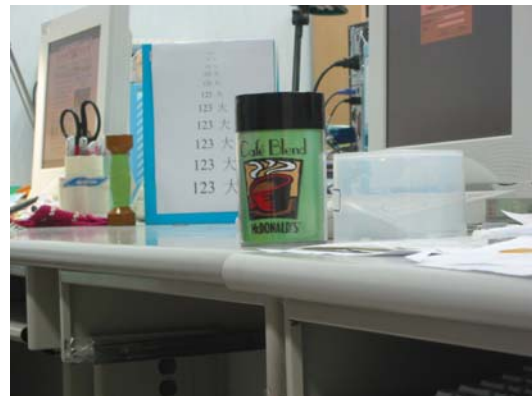

Figure 5 The focal length is 10 mm.

Figure 6 The focal length is 21 mm (longer focal length thus shallower depth of field).

3. Distance of the object

The longer object distance has a deeper depth of field. The shorter object distance has a shallower depth of field. We consider an original blurred image of the object. When we change the longer distance of the object, the distance of the light from the object is longer. The distance approximates infinite. We know that infinite light would get together at the focal point. Thus farther object has deeper depth of field. We look at Figures 7 and 8. Figure 7 is sharper.
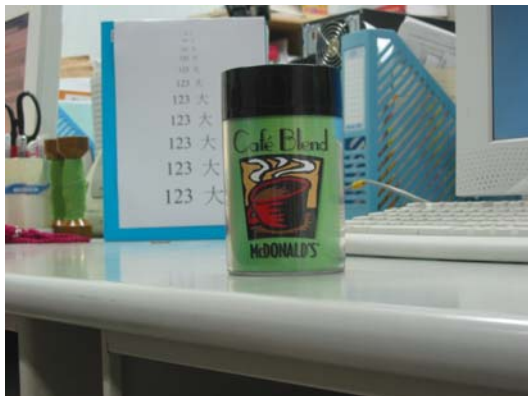
Figure 7 The distance of the object is 100 cm.

Figure 8 The distance of the object is 60 cm (shorter object distance thus shallower depth of field).

III. Approach

Here we introduce mainly some passive algorithms of autofocus methods. The passive autofocus methods apply edge detection methods. They detect edges mainly, and then decide if the camera is in focus. If the camera focuses, the autofocus ends. If not, the camera adjusts the position of the lens. Then, it takes a picture and detects edges. It goes iteratively till it focuses. Now, we introduce some edge detection methods.

1. Roberts operators [1]

This method was employed by Roberts in 1965. He used two 2*2 masks to calculate the edges in two diagonal directions. The masks are shown in Figure 9.

$$r_1 \qquad\qquad r_2$$

Figure 9 Masks used for the Roberts operators.

Value r_1 is calculated from the left mask, and r_2 is the value calculated from the right mask. Then, we calculate $\sqrt{r_1^2 + r_2^2}$. If this value is greater than a threshold θ, we label this pixel as edge. This method uses the differential:

$$f(x)' = \lim_{\Delta x \to 0} \frac{f(x) - f(x - \Delta x)}{\Delta x}$$

But $\Delta x \to 0$ is impossible in the digital world. We let $\Delta x = 1$, so the above differential is changed as follows:

$$f(x)' = f(x) - f(x-1)$$

The two masks use the differential in the two diagonal directions.

2. Prewitt operators [1]

This method was employed by Prewitt in 1970. He used two 3*3 masks oriented in the row and column direction. The masks are shown in Figure 10.

$$p_1 \qquad\qquad p_2$$

Figure 10 Prewitt edge detector masks.

Each column of the left mask looks one entry. Each entry calculates *(f(x)'+f(x+1)')/2*.

$$f(x)'+f(x+1)'=f(x)-f(x-1)$$
$$+f(x+1)-f(x)=f(x+1)-f(x-1)$$

Each row of the right mask is the same as the left one. Finally, we calculate $\sqrt{p_1^2+p_2^2}$.

3. Sobel operators [1]

This method is used by Sobel. It also uses two 3*3 masks. These masks are similar to the masks of Prewitt operators, but the center has more weight. The masks are shown in Figure 11.

$$S_1 \qquad\qquad S_2$$

Figure 11 Sobel edge detector masks.

This method is similar to Prewitt method to calculate $\sqrt{S_1^2+S_2^2}$.

4. Laplacian method [1]

This method uses second differential which is shown as follows:
$$f(x+1)''=f(x+1)'-f(x)'$$
$$=f(x+1)-f(x)-f(x)+f(x-1)$$
$$=f(x-1)-2*f(x)+f(x+1)$$

Thus, this method sums the absolute value of each pixel and finds the maximum.

5. Other operators

There are some edge detector operators. For instance, the Frei and Chen operator is similar to Sobel method, but it has other weighted coefficient. Some methods use more than two masks or lager masks. These methods may not be suitable for autofocus because they need more computing time and more memory space. They may take several seconds in the desktop

computer but more than one minute for the camera microcontroller. The user would not want to spend too much time taking pictures.

In fact, a passive autofocus method consists of the edge detection and the algorithm of the moving lens. How fast the lens moves to the best focal position also affects the total time for autofocus. If searching best position for focus is slow, users would not use this camera. We introduce two methods:

1. Global Search Method [2]:

This method has to decide a step, first. You should initialize your lens in the start position and take a picture. Then you calculate the edges with your edge detection method and record the value in the camera memory. You move your lens moves forward or backward one step and calculate the edge value again. Repeat above steps till your lens can not move forward or backward. Find the maximal value and put the lens in the position. This is the simplest method, but it wastes time and need more memory. The steps of moving lens are too many.

2. Mountain-Climbing Method:

This method is a little similar to global search method. Set the lens in a start position. This method also calculates the edge value when the lens moves. It defines a value Δx .
$$\Delta x=V(x)-V(x-1)$$
Value $V(x-1)$ indicates the past step edge value. Value $V(x)$ indicates the edge value in this position. If Δx is positive and over a threshold, it indicates the value goes up the mountain. The lens goes forward. If Δx is negative, it indicates it goes downhill. The lens should move backward, and it reduces the step. It repeats above method with the same

edge detection or other method till it finds the maximal value. With this method, the total distance of moving lens would not be longer than Global Search Method. This method does not reserve so many data, and it saves the memory. This method improves the Global Search Method. Many other methods are similar to this method.

IV. Experiments

Step:

1. We take pictures in lower light situation.

2. We use the differential, Roberts, Prewitt, Sobel, and Laplacian method to calculate the edges.

3. We normalize the results and plot it.

In this experiment, we use the Canon camera PowerShot S40 to take pictures. We fix the aperture in F/7.1. We adjust the focal length from 10 cm to infinite. But we can not control the step very well. We set the exposure time 1/80 second as lower light case. In Step 2, we use the following formula to change the R, G, and B value to the intensity value.

$$I(R,G,B) = 0.299*R + 0.587*G + 0.114*B$$

We use C++ to develop a program to calculate the edges. We run the program with CPU AMD XP 2500+ and 512 MB RAM. We use Matlab to plot the results.

In Step 1, we turn off the digital zoom and just use optical zoom. We set the focal length 10 cm initially and zoom out. When zooming each time, we take a picture. In Step 3, because we normalize the results, the differences between values would be reduced. The following is the lower light pictures:
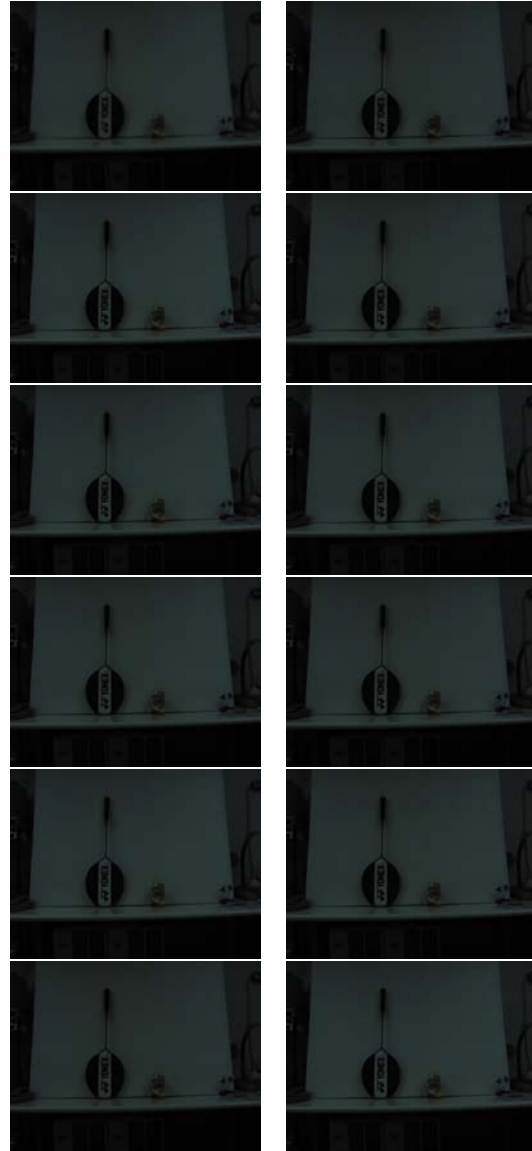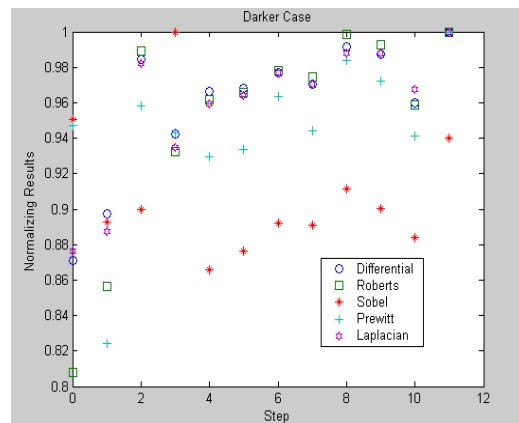

V. Results

Figure 12 The results in lower light case.

Figure 12 is the results in lower light case. The Roberts is the best of all, but it has three

peaks. The values of three peaks are similar in this plot, but the values of these peaks are not the same. This is caused by the normalization. The waveforms of the differential and the Laplacian are similar. The Sobel and Prewitt method are worse. They are affected by other things. Their peaks appear in other position. It seems to be affected by noise.

VI. Discussion

We also do the experiments in brighter light and normal light cases, but the results are not ideal. The Roberts method is the best in the lower light case. The differential and Laplacian are affected by the background. Their peaks appear in other position. We think this may be affected by some other less important things. But this may be a problem in the real world. Autofocus may make mistake. We think we may choose some areas in the picture to calculate focal distance. This may reduce the possibility of wrong focal length and the computing complexity.
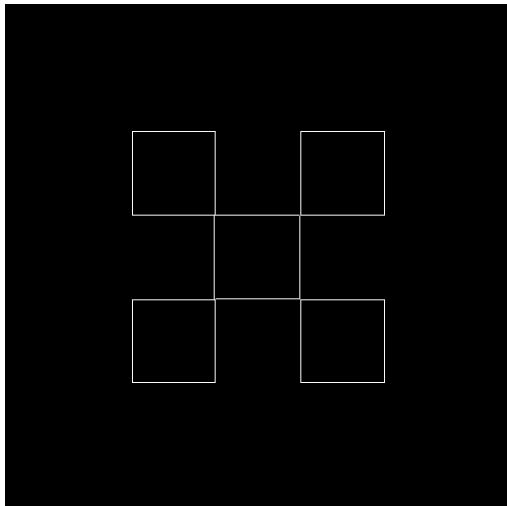

Figure 13 An example for areas computing

Figure 13 is an example for areas computing. Because it just computes the values of the five areas, it can save time. It also can reduce the influence of background. Why we choose these five areas? If we cut an image by four lines, we can get nine areas. People often put the object in the neighbor of the center area or just in the center area. We just compute the edge values of the object because it is the most important.

Understanding the property of your lens is important for the autofocus method testing, for instance the times of your lens and the step for adjusting the focus. Our camera just has three times optical zoom. We should let the pictures have large variance, but we did not do it. We should design the experiment with finer steps.

In this experiment, the Sobel and Prewitt methods take more one second. This may takes many seconds in the camera. Users do not want to wait. The Laplacian has more computing complexity than the differential. The best choice is the Laplacian. If you want to save time, you could choose the differential.

VII. Reference

[1] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision,* Vol. I, pp. 337-352. Addison-Wesley, Reading, MA, 1992.

[2] N. K. Chern, N. P. A. Neow, and M. H. Ang, Jr., "Practical Issues in Pixel-Based Autofocusing for Machine Vision," *Proceedings of the IEEE International Conference on Robotics and Automation*, Seoul, Vol. 3, pp. 2791-2796, 2001.