國 立 交 通 大 學

電子工程學系 電子研究所碩士班

碩 士 論 文

動態自動對焦演算法

On Dynamic Auto-Focusing Algorithms

研 究 生：林耀仚

指導教授：杭學鳴 博士

中 華 民 國 九 十 六 年 六 月

動態自動對焦演算法

On Dynamic Auto-Focusing Algorithms

研究生：林耀企　　Student: Yao-Hsuan Lin
指導教授：杭學鳴　　Advisor: Dr. Hsueh-Ming Hang

國 立 交 通 大 學
電 子 工 程 學 系　電 子 研 究 所 碩 士 班
碩 士 論 文

A Thesis
Submitted to Department of Electronics Engineering & Institute of Electronics
College of Electrical and Computer Engineering
National Chiao Tung University
in Partial Fulfillment of the Requirements
for the Degree of Master
in
Electronics Engineering

June 2007

HsinChu, Taiwan, Republic of China

中華民國九十六年六月

動態自動對焦演算法

研究生:林耀佡　　　　　　　　　　　　　　　　　　指導教授:杭學鳴

國立交通大學　電子工程學系　電子研究所碩士班

摘要

自動對焦，是數位影像加強技術中重要的議題之一。近年來數位相機/攝影機的普及,隨著科技的進步,影像品質逐漸受到更多的重視。影像加強是影像處理技術的核心,影像加強的目的是追求真實，修正影像與人眼感官相符。自動對焦和自動曝光、自動白平衡，並稱 3A 演算法，是目前影像加強的主要演算法。靜態影像的自動對焦演算法已經有很多的研究成果，唯動態影像的自動對焦演算法尚沒有顯著的討論。本篇論文改良整合現有的自動對焦技術，提出一個動態自動對焦的演算法。

動態自動對焦與靜態自動對焦的主要差異，在於搜尋演算法沒有所謂起始和終止，必須不斷地遞迴搜尋。論文先分析數位攝影機的基本架構，並討論其回饋系統的特性。討論自動對焦的結果，會受到系統本身哪些的限制。我們針對畫面轉換、物體晃動、焦距變動幾個現象，改良現有的爬山搜尋演算法，維持自動對焦系統的效能和穩定性。我們以有限狀態機器的形式，藉由設定專門的轉換參數，控制搜尋演算法的狀態轉換，來實現動態自動對焦演算法。我們利用亮度指標來判斷換面轉換的發生，利用門檻加強的搜尋演算法，處理物體晃動的問題。並提出一個線性的預測模型，來改善焦距變動的對焦效率。最後我們以軟體模擬的形式,來驗證我們的自動對焦演算法能對動態影像運作。

On Dynamic Auto-Focusing Algorithms

Student: Yao-Hsuan Lin Advisor: Dr.Hsueh-Ming Hang

Department of Electronics Engineering &

Institute of Electronics

National Chiao Tung University

Abstract

Auto-Focus is one of the most significant research issues in the recent digital image enhancement technology. Its importance increases due mainly to the widespread use of digital video/imaging devices in this new millennium. The stationary image auto-focusing system is a long-standing research topic. In this thesis, we modify and improve the conventional auto-focusing algorithms and integrate them into a dynamic auto-focusing algorithm.

The major difference between a still image and a video auto-focusing system is the search algorithm. Still image search algorithms often have specific start and end, but the video application urges a continuous searching routine. First, we study the digital camera system and the feedback control theory. Through these studies, we understand how the system structure limits the auto-focusing result and the principles of designing a good auto-focusing system, which is a special type of control system.

Since the search algorithm is critical, we improve the climbing search algorithm for particularly the dynamic environments such as scene change, local object motion, and zoom tracking. This proposed search algorithm working on the video is the core of our auto-focusing algorithm. Then, we develop our search algorithm using a finite state machine structure. We design specific transition conditions and state transition table to match the requirements of dynamic auto-focusing applications. A luminance-based metric helps to detect the scene change. We adopt a threshold climbing search algorithm to solve the local motion problem. And the zoom tracking processing is accelerated with the assistance of a well-designed linear prediction model. Finally, we show that this algorithm is reliable and efficient by a series of software simulations.

誌謝

感謝杭學鳴老師，在這一年半中給我諄諄教誨。由於提早入學的關係，我的時間比一般同學來的緊湊，研究過程和想法也比一般同學來的不成熟。在老師的包容和耐心教導下，讓我慢慢學會如何做研究以及寫出這篇論文。

再來要感謝我的家人。謝謝爸爸媽媽平時給我鼓勵，在我徬徨迷惘時，繼續支持我。

還要感謝實驗室的學長們。雖然我加入的時間比較晚，大家還是很幫忙我。在我學業和研究上遇到瓶頸時，很熱心的幫助我解決問題。實驗室這種溫馨的氣氛，真的讓我受惠良多。

最後僅將本論文獻給所有關心和幫助過我的人。

Contents

List of Table

List of Figures

Chapter 1

Introduction

A digital image capture device is an essential part of today's high-tech products. To see is to believe. Today, not only digital cameras but also mobile phones and laptops have a lens to make a shot or produce a live video conversation. People can store what they see and share images with the others. The most important aspect for everyone is the quality of the images or video sequence. Therefore, clarity is an essential requirement. In order to take a focused photograph, everyone traditionally needs to adjust the camera lens. Auto-focusing technique helps customers to take a clear picture without mastering a complex focusing process.

Auto-focusing algorithm has been a long-standing research subject. Since the 1970s, the traditional camera producers have made products with passive auto-focusing systems. In order to achieve accurate auto-focusing it varies necessary to measure the differences of reflected light between two CCD sensors. This technique has been used widely on instamatic devices until recently. The source of light is the main factor that controls the performance of passive auto-focusing system. Due to the dependence of passive auto-focus systems on luminance, image quality has rapidly deteriorated in the dim environments. Due to this major drawback, only inexpensive cameras now use only passive auto-focusing systems today.

Clearly, the development of a better auto-focusing system was urgently needed. As a result, camera producers developed a new active auto-focusing system in the early 1980s. An additional device that generates an infrared ray to assist in focusing in the dim light was added. In order to achieve accurate auto-focus it is necessary to measure the difference in time between when the infrared ray is sent and when the focused subject reflects it. While the development of technology to overcome insufficient light represents great progress, the focus performance still suffers from the uncertainty of the object reflection. Therefore, an external focal point of reference is not reliable and hard to improve. Perhaps the solution to this problem is to be found within the device itself, more specifically in the digital image data.

This idea began in the mid 1980s and was first used in the conventional camera; however, the core concept is same in today's digital cameras. Each lens has a control motor and sensor that allows the lens to change its focus until its sensor achieves maximum signal energy. The sharpness of the image object boundary is used to determine the quality of image focus. Unlike the previous auto-focus techniques, using the current image information makes the new auto-focus process much faster and more precise. It has been gradually applied to most cameras and is a basic function of modern digital camera products.

However, the auto-focusing algorithm still has a long way to go. First, the huge number of sensors in current digital imaging devices must make a large amount of computations in order to process the received signals which consume considerable time and power, and more importantly, the existing algorithm is unable to cope with this volume. As more accurate auto-focus results require more computation. To strike a balance between performance and efficiency is the top priority of the auto-focusing algorithm. However, at present there is still no generally acknowledged measurement to determine whether an algorithm is better or not. Second, the dynamic auto-focusing application is another vital issue. Consumers today are not satisfied with just taking a photograph, but they also want to be able to do video recording. This requirement urges the development of new compression techniques and reveals that the auto-focusing algorithm should put more emphasis on dynamic issue.

The thesis is organized as follows. In Chapter 2, basic background of general digital camera system will be described. It is in structure a feedback control system. In Chapter 3, we introduce the basic auto-focusing algorithms. The dynamic auto-focusing algorithm we develop is discussed in Chapter 4. We will show our simulation result in Chapter 5. Chapter 6 is the conclusion.

Chapter 2

Overview of Digital Camera Systems

Digital cameras dominate today's camera markets. A digital camera stores photographs in a digital format, instead of on conventional photographic film. The digital image format, which is easier to store and share with others, plays an important role in the modern E-society. Since the digital image format is becoming more and more popular, almost all high-tech electronic products contain a digital camera system for customs to capture images anytime they wish. Modern compact digital cameras are also multifunctional, with some devices capable of recording video and sound. Digital cameras can be classified into many categories: video cameras, webcams, live-preview digital cameras and compact digital cameras. Video cameras are defined as devices whose main purpose is recording moving images; all functions are dynamic and operate in real-time. Live-preview cameras are cameras that generate live-preview digital image on an electronic screen before taking the photograph. Many modern digital cameras have a movie mode, and a growing number of camcorders can take still pictures. Nevertheless, the cheapest digital cameras may take better still image quality than a mid-range video cameras, and mid-range video have much lower video quality than a mid range video cameras. Different preprocessing algorithms and processor commands leads to a great diversity of digital (video) cameras, but most of them have similar hardware architectures.

2.1 Architecture of General Digital Camera Systems

There are many types of digital cameras, but the basic architecture is common to all. A general digital camera system is shown in Figure 2.1[1].

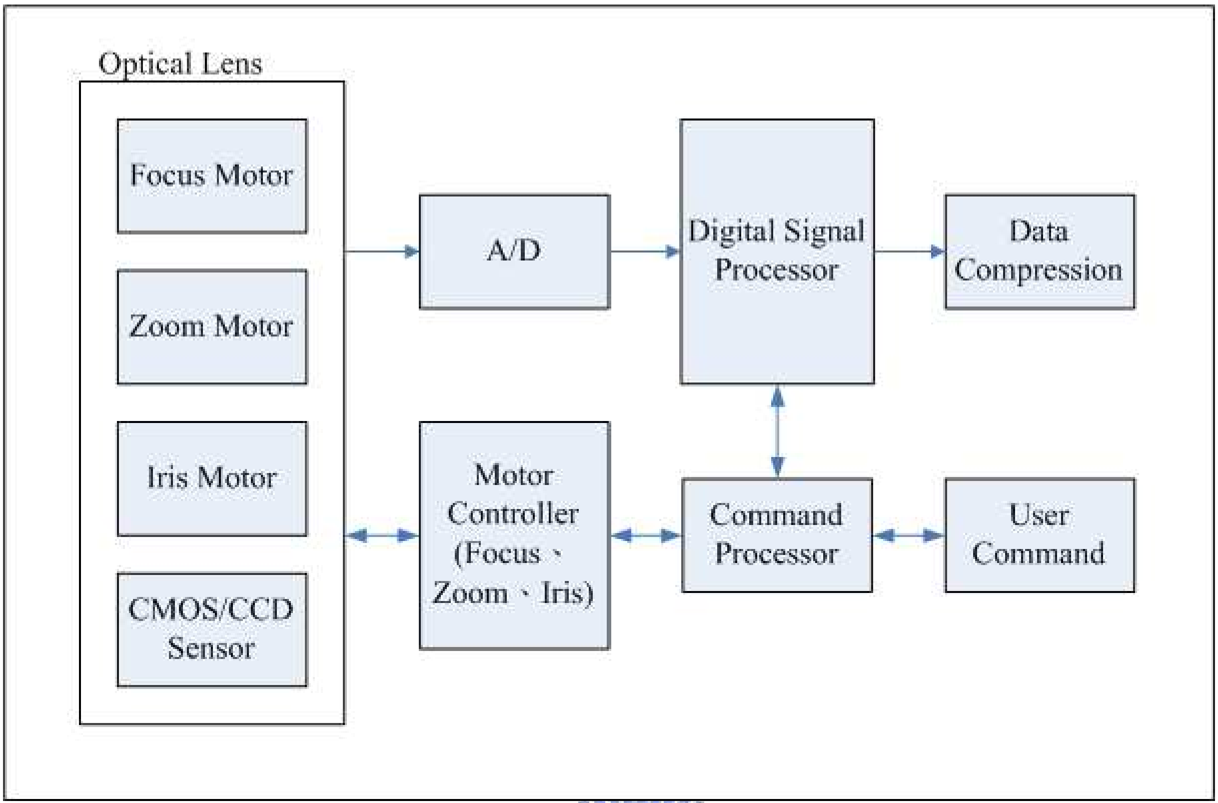


Figure 2.1 Block diagram of digital camera system.

The camera receives signal (light) gets via its optical lens motors and sensors [12]. A camera lens consists of a set of lenses that are adjusted with the use of motors. Motors can change the focal length of a lens set, which can directly impact on focus and zoom. Light coming through the lens is finally caught by the CCD/CMOS sensors that convert light into electronic pulse. Due to the fact that the camera system is digital, an A/D is used to convert analog signals into digital formats. However, this raw digital data needs further refinement. This digital raw data could not transform to standard image formats yet, because current consumer digital cameras use Bayer color filters to promote color image's resolutions and qualities which makes data spilt in discrete RGB domain. The process to refine raw data with hardware is called image pre-processing, a comparison with image post-preprocessing which is aimed at refining images with software. The well-known 3A algorithms are also parts of image pre-processing. Each digital camera has a digital signal processor to handle this complex pre-process and store raw data into standard image formats, such as JPEG. There is also a robust connection between digital signal processor and a control processor, which receives outside user instructions and inside system requirements. A protocol is like blood vessels cascade all control units together. Different camera systems have different protocol for internal system control to fulfill hardware requirement. Motor controllers inside camera's lens adjust specific function, like zoom value and focus value, precisely under control. The following sections may describe more detail for auto-focusing application.

2.1.1 Lens

A photograph lens is an optical mechanism to make images of objects. In principle, a lens used for a camera, a microscope, a telescope, or other apparatus is similar, but the detailed design and construction are different. A practical camera lens often incorporates an adjustable aperture mechanism to regulate the amount of lights that may exposure on a camera film. The main optical parameter of a camera lens is focal length, which determines the angle of view, and the magnification of the camera system. Another widely usable aperture of a lens is specified as the focal ratio or f- number, which is defined by the focal length divided by the effective aperture diameter in the same units. When the f-number gets lower, there are more lights delivered to the focal plane. The focal length of a lens is constant. In order to adjust suitable focal length, the simplest vari-focal lens, shown as Figure 2-2 [3], is a set of lenses that is composed of two converging lenses outside and one diverging lens inside.

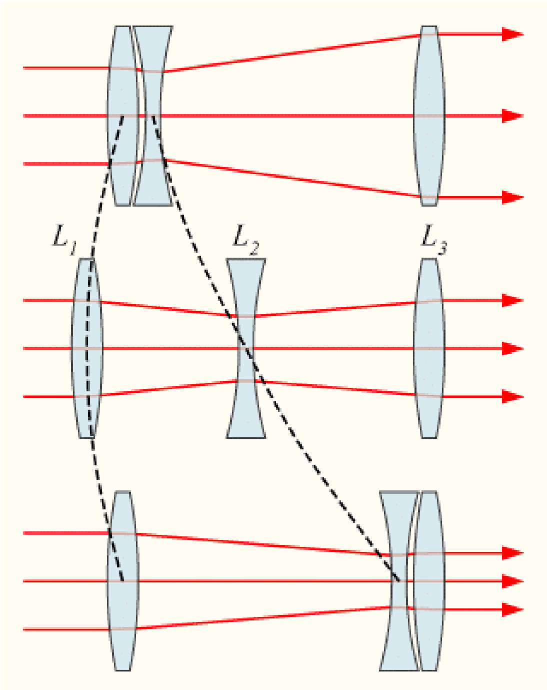


Figure 2.2 A simple vari-local lens set. [3]

For auto-focusing purpose, this set of lenses equips electric motor to vary its focal length, which is also called zoom lenses. A vari-focal lens may zoom from moderate wide angle to extreme telephoto. The zoom range is constrained by maximum aperture of a camera lens. In optical physics theory, zoom is closely connected with focus process and can be compute to a formula. Commercial camera lens has more detailed design than the simplest vari-focal lens, which is composed of more than 9 lenses and takes out a patent.

2.1.2 CCD/CMOS Sensors

A charged-coupled device (CCD) is an image sensor, which consists of an integrated circuit containing arrays of linked light-sensitive capacitors. The material of CCD image sensor is semi-conductor, mostly used silicon. Another popular image sensor is called CMOS sensor, which has similar manufacturing with the CCD image sensor. The main difference between a CCD sensor and a CMOS sensor is the process of image recording. The CCD image sensor stores the light signal into arrays of electrons, and the electrons can only be read out array-to-array that’s mean if an electron is lost, the data of the total array will be crushed.

In order to get correct data, each CCD sensor should be manufacture carefully. Therefore, the CCD manufacturing is more complex and expansive than typical semi-conductor. The CMOS image sensor benefits by its cheap manufacturing and regular process, which is important for integration of image sensors and other electronic device. The detail block diagrams of a CCD and a CMOS sensor are shown as followings.

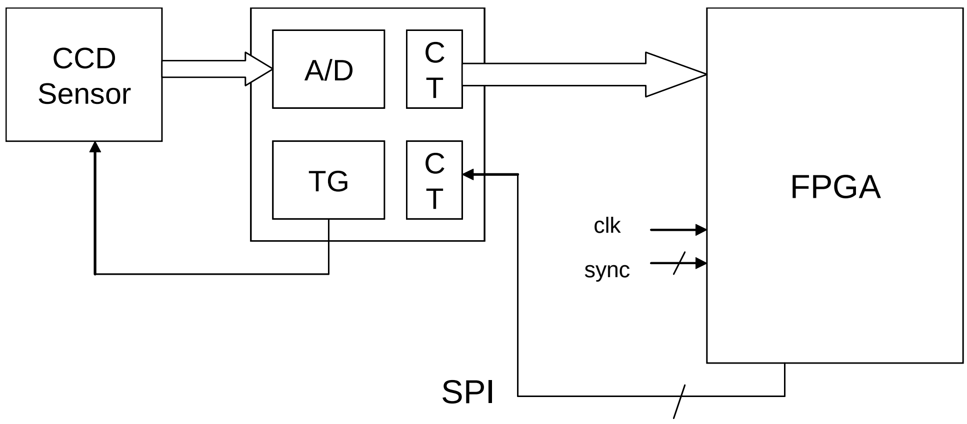


Figure 2.3 Block diagram of CCD sensor.

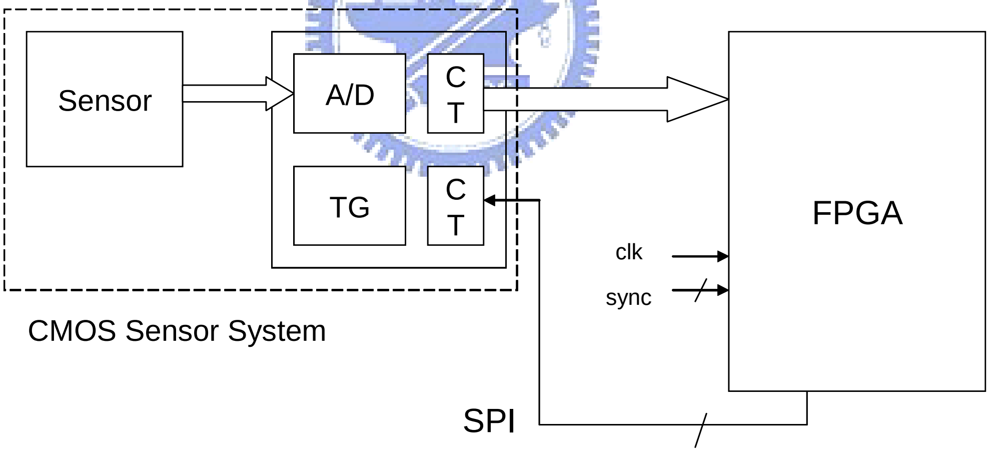


Figure 2.4 Block diagram of CMOS sensor.

Table 2.1 A comparison between CCD and CMOS sensor.

CCD Advantages	CCD Disadvantages
Low noise, high sensitivity, simple and mature process.	Large power, cannot randomly read any pixel and high requirement.
CMOS Advantages	CMOS Disadvantages

Low cost, low power, randomly read any pixel and SoC implementation.	High noise, weak sensitivity and complex circuits.

2.1.3 Mosaics, Interpolation, and Aliasing

The standard RGB color model requires three intensity values for each pixel: one each for red, green, and blue. A single sensor element cannot record these three intensities at the same time, and so a color filter array is usually used to selectively filter a particular color for each pixel. Most consumer digital cameras use Bayer filter mosaic (Figure2-2) in combination with an optical anti-aliasing filter to decrease the aliasing effect due to up sampling of different primary-color images. The Bayer filter pattern is a repeating 2*2 mosaic pattern of RGB color filters, with a red one at right-up corner, a blue one at left-down side, and green ones in the other two positions. The high proportion of green filters takes advantages of properties of human visual system, which determines brightness mostly from green and is much more sensitive to brightness than to hue or saturation. Sometimes a 4-color filter pattern is used, often involving two different hues of green. This provides potentially more accurate color, but requires a slightly more complicated interpolation process.

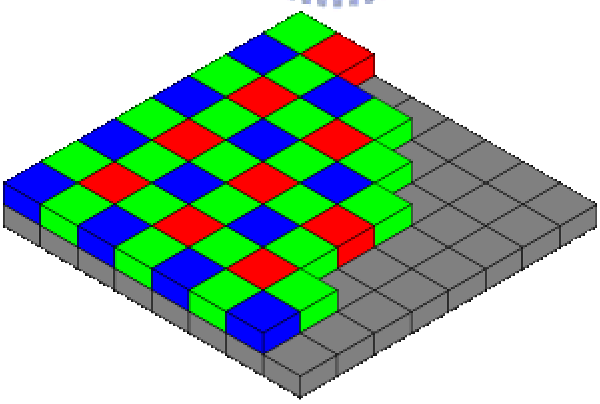


Figure 2.5 Bayer filter. [3]

A demosaicing algorithm is used to interpolate color information to create a full array of RGB image data. The simplest is the bilinear interpolation method. In this method, the red value of non-red pixel is computed as the average of the adjacent green pixels, and similar for green and blue. Some algorithms not only compute linearly but also adapt their method of estimation on features of the area surrounding the pixels of interests. These up sampling algorithms all encounter the critical aliasing problem on high frequency detail, which is lost through the Bayer filter process. The huge numbers of pixels in today's digital cameras make the fast algorithm more and more urgent. Many commercial products develop its own algorithm about which is little publicly known and may be much different with known algorithm.

2.2 Feedback Control System

The camera system mentioned before is also a feedback control system [5], which can be simplified as the following graph Figure 2.6. Concern with auto-focus application, the topic of this thesis, the system input signal, Y_{sp}, is the correct focus point of object in theory. A feedback controller gets this information and then sends control signals to system process, which is noised by disturbance, D. Under this feedback system, the system output Y will finally be as same as the input Y_{sp}. However, the feedback control in real world is not as perfect as mentioned.

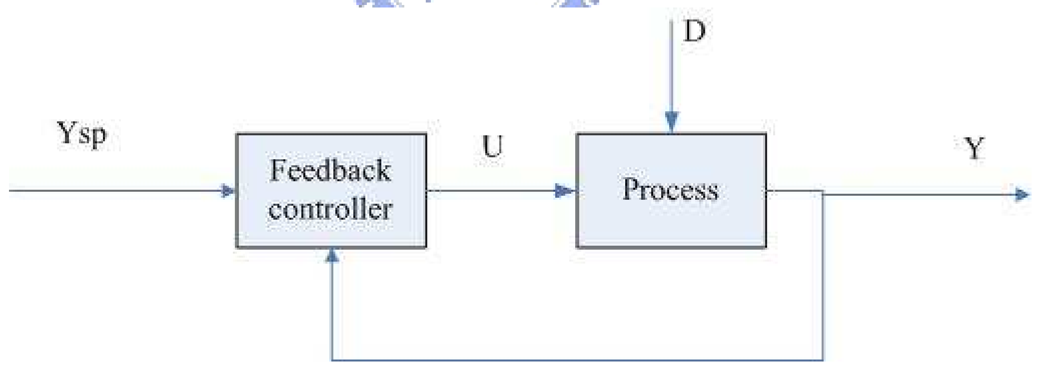


Figure 2.6 Block Diagram of feedback control system. [5]

Feedback control is an important technique that is widely used in the process industries. Its main advantages are as follows.

1. Corrective action occurs simultaneously as the controlled variable deviates form the set point, regardless of the noise and type of source.
2. The requirement of system knowledge is minimal. In particular, a mathematic model of

the process is not required.

3. The ubiquitous PID controller is both versatile and robust. Whether process conditions changes, re-tuning the controller usually produces satisfactory control.

However, feedback control also has certain intrinsic disadvantages:

1. Perfect control is theoretically impossible. No corrective action is taken until after a deviation in the controlled variable occurs, even the set point changes or is during disturbance.
2. No predictive control action is provided. It cannot compensate for the effects of known or measurable disturbances.
3. It may not be satisfactory for process with large time constants or large time delays. If large and frequent disturbances occur, the process may operate continuously in a transient state and never reach the desired steady state.
4. Sometimes, the controlled variable cannot be measured consequently, and feedback control is not feasible.

We will present some enhanced loop control strategies in this section.

2.2.1 Cascade Control

The first disadvantage of conventional feedback control is no corrective action is taken until after a deviation in the controlled variable occurs. The cascade control employs a secondary measurement point and a secondary feedback controller to improve the dynamic response to disturbances. The cascade controller is useful when the disturbances are associated with specific variables or when the final control elements are nonlinear.

The cascade control loop structure, shown in Figure 2.7, has two distinguish features:

1. The output signal of master controller (G_{c1}) determines the set point for the slave controller.
2. The two feedback control loops are nested, with the secondary control loop located inside the primary control loop.

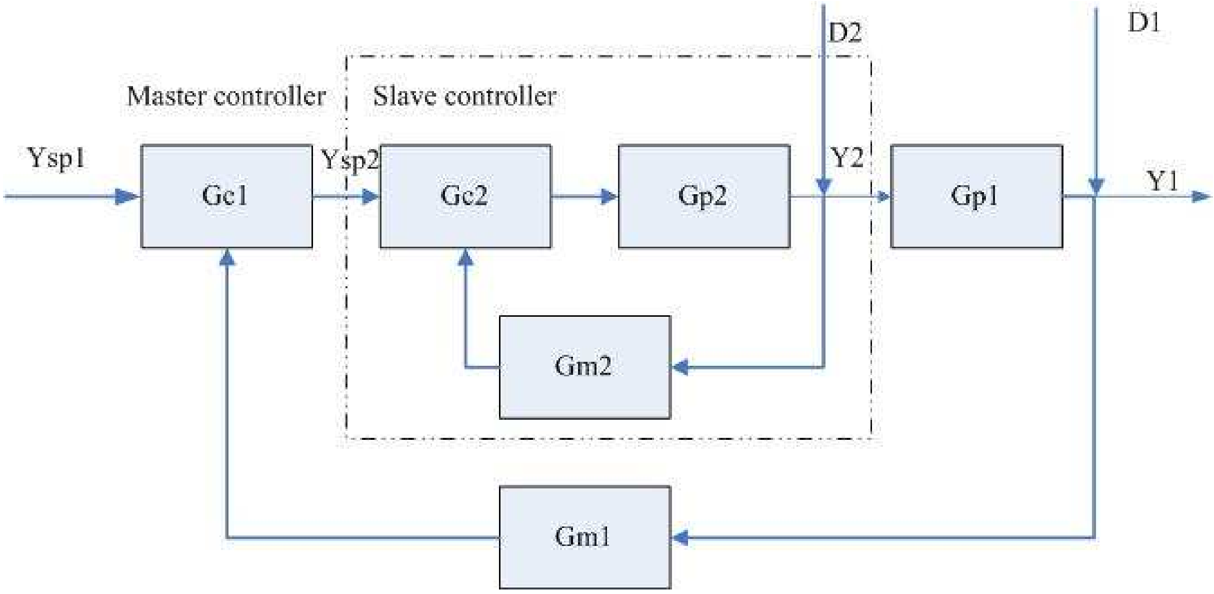


Figure 2.7 Block diagram of cascade control system. [5]

Cascade control can enhance the sensitivity to set point change. However, it should be designed properly to improve its performance in the presence of disturbance. Considering the block diagram algebra:

$$\frac{Y_1}{D_2} = \frac{G_{p1}}{1 + G_{c2}G_{p2} + G_{c1}G_{c2}G_{p1}G_{p2}} \tag{1}$$

By similar analysis, the set-point transfer functions for the inner and outer loops are:

$$\frac{Y_1}{Y_{sp1}} = \frac{G_{c1}G_{c2}G_{p1}G_{p2}}{1 + G_{c2}G_{p2} + G_{c1}G_{c2}G_{p1}G_{p2}} \tag{2}$$

$$\frac{Y_2}{Y_{sp2}} = \frac{G_{c2}G_{p2}}{1 + G_{c2}G_{p2}} \tag{3}$$

For disturbance in D1, the closed-loop transfer function is

$$\frac{Y_1}{D_1} = \frac{1 + G_{c2}G_{p2}}{1 + G_{c2}G_{p2} + G_{c1}G_{c2}G_{p1}G_{p2}} \tag{4}$$

Observing this equation realize when the slave loop responds faster than the master loop, the cascade control system will have improved stability. Cascade control also makes the closed-loop process less sensitive to disturbance or errors.

2.2.2 Time Delay Compensation

In this section we present an advanced control technique, time-delay compensation, which deals with a common problem in process control, namely, the occurrence of significant time delays. From a frequency response perspective, a time delay adds phase lag to the feedback loop, which adversely affects closed-loop stability. Therefore, the controller gain must be reduced below the value that could be used if no time delay were present, and the response of the closed-loop system will be inefficient compared to the control loop with no time delay.

In order to improve the performance of system with time delays, special control strategies have been developed that provide significant time delay compensation. The Smith predictor technique is the best strategy that is known. A block diagram of the smith predictor controller structure is shown in Figure 2.8.

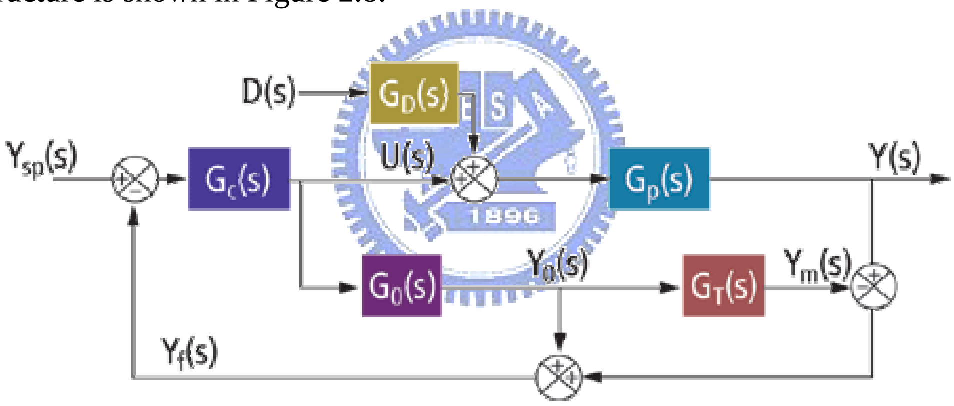


Figure 2.8 Block diagram of Smith predictor.

After some computation, the closed loop set point transfer function is:

$$\frac{Y}{Y_{sp}} = \frac{G_c G^* e^{-\theta s}}{1 + G_c G^*}, \tag{5}$$

where $G_T(S) = e^{-\theta s}$ is estimation of process delay. By contrast, for conventional feedback control

$$\frac{Y}{Y_{sp}} = \frac{G_c G^* e^{-\theta s}}{1 + G_c G^* e^{-\theta s}} \tag{6}$$

Compared with conventional feedback control, the Smith predictor has theoretical advantages of eliminating the time delay from the characteristic the time delay from the characteristic equation. But it has a serious disadvantage, the advantage is lost if the process model is inaccurate. Fortunately, the delay in camera system is usually constant and not large, which is caused by processor computation and CCD/CMOS capture. Another disadvantage of the Smith predictor approach is that it is model-based; that is a dynamic model of the process is required.

Chapter 3

Principles of Auto-Focusing System

This chapter describes the auto-focusing process from the perspectives of both digital image processing and signal processing. Auto-focus is an important pre-process of a digital camera system, which adjusts the focus position. It receives and processes the raw data obtained from the image sensors. The auto-focusing system, unlike the other pre-processes, does not directly alter the raw data, but it does provide feedback. In other words, an auto-focusing system cannot be an instant or real-time system. It needs a little bit of time to complete the auto-focusing process. How to extract the focus value from the image data and how to reduce the time of focusing process are the critical issues in the development of a robust and practical auto-focusing algorithm.

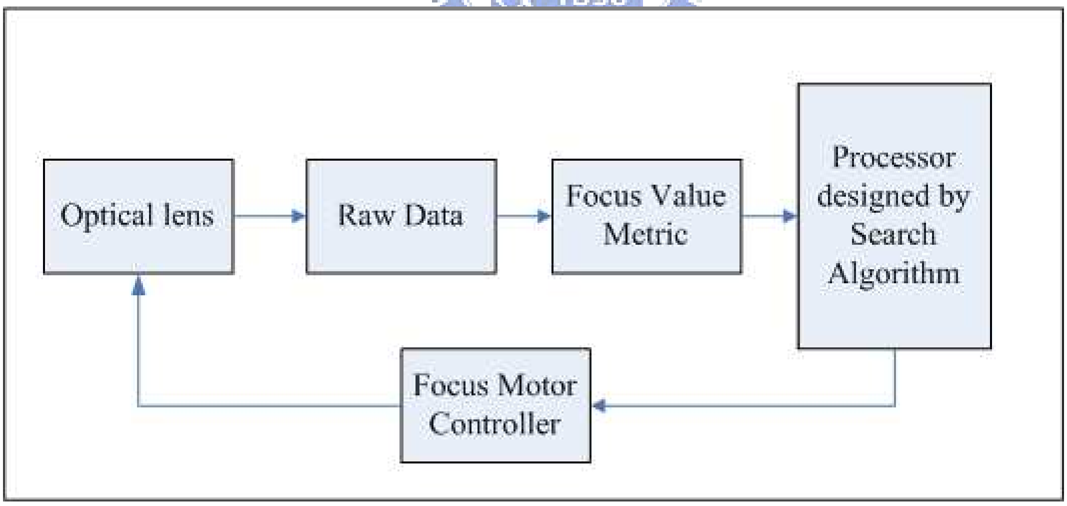


Figure 3.1 Block diagram of an auto-focusing system

Figure 3.1 shows a block diagram of an auto-focusing system, which is simplifies from the model of a digital camera system. The major parts of the auto-focusing algorithm are the focus value metric and the search algorithm. We also introduce the window selection, which is a common function in the modern auto-focusing system.

3.1 Focus value and Focus curve

The first step in the development of a robust auto-focusing system is to determine a good focus value metric. This step is classified as feature extraction, which compresses a large amount of data into a simple form with sufficient accuracy. So the focus value is defined as the focus measurement of an image data. A good focus value model should have the following properties: First, its process should be independent of the received data and its measurement is universal for all types of image structures. Second, it should be a decreasing function with respect to the true focus of the target. If the image is sharper, the focus value gets higher. Third, it can tolerate disturbance and noise. Even though the camera system is disturbed by noise and delay, its variation is small and predictable. In theory, a blurred image is the output of a focused image convolved with a band-limited low-pass filter. Therefore, a simple focus value model is detecting the high-frequency components of an image.

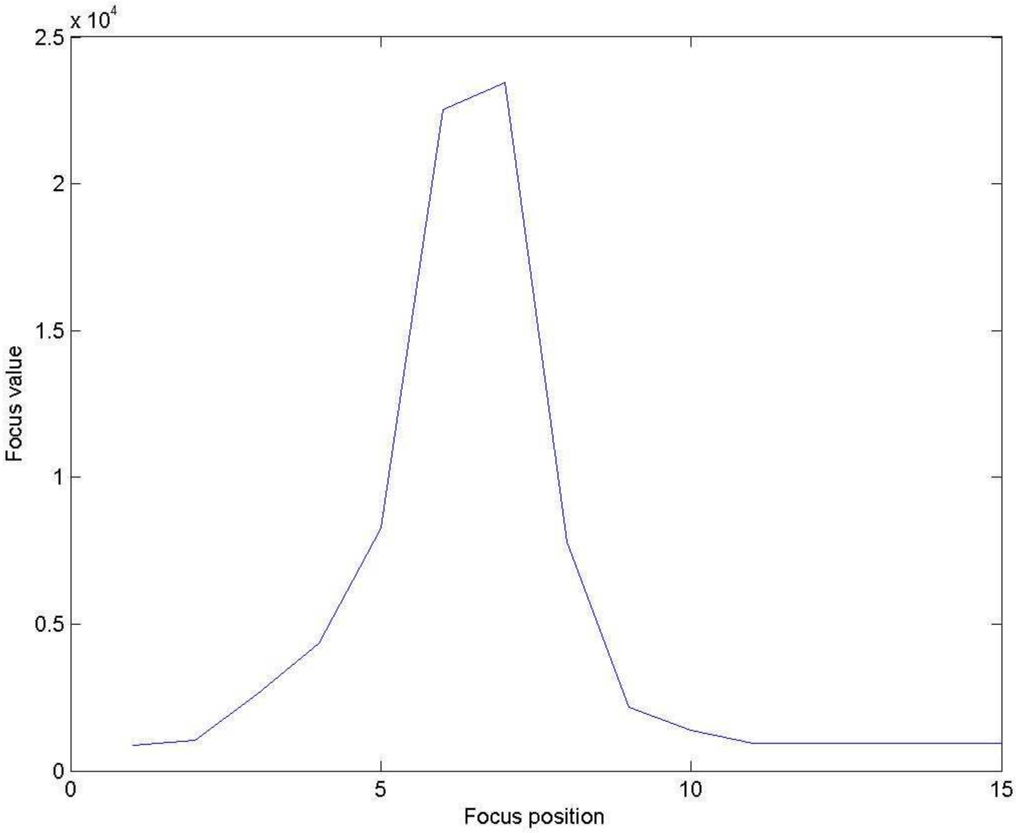


Figure 3.2 A general focus curve.

Figure 3.2 shows an example of focus curve, which plots the focus position against the focus value derived using a Sobel operator. A focus curve, which shows the focus values at different positions (locations), is a result of several factors. Its shape depends on the image, the optical system, the lighting condition, and the metrics (operator). The general shape of many published focus curves is similar. They often have a peak with a sharp tip and two steep slopes. The tip of the focus curve shows the optimal focus position of the scene. A close-up of such a focus curve reveals small magnitude disturbance that are due to thermal noises. Many research and published papers assume that focus curve is the basis in designing and evaluating an auto-focusing system. An auto-focusing system with stable focus curves indicates its system design is more robust, because the stable focus curves shows that exceptional condition is very rare in this auto-focusing system. Although many commercial devices have good auto-focusing systems, there still lacks a high performance focus value model in the literature. Certain types of frequency transforms or image operators are widely used as the focus value model as describe below.

3.1.1 Sobel Operator (First Derivative)

The Sobel operator is a first-derivative operator. First derivatives in image processing are implemented using the magnitude of the gradient. This is also an image enhancement in the spatial domain, which convolves a designed filter mask with the original image. Masks of even size are awkward to implement. The smallest filter mask is a mask of size 3*3.

$$\mathbf{G_x} = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G_y} = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A}$$

$$\mathbf{G} = \sqrt{\mathbf{G_x}^2 + \mathbf{G_y}^2}$$

Figure 3.3 Sobel masks and equations.

The mask of the sobel operators/filters shown in Figure 3.3 can be used for the edge enhancement. In Figure 3.3, A is the original image, and G_x and G_y are derivatives along the horizontal and the vertical axes. The resulting gradient approximation G is the square root of G_x and G_y. A weight value of 2 at the center is giving more importance to the center point. Note that the coefficients in the masks shown in Figure. 3.3 sum to 0, indicating that they would give a response of 0 in an area of constant level, as expected of a derivative operator. Mathematically, the Sobel operator is not a linear operator because of the square root calculation. Figure 3.4 illustrates the computation of the Sobel operators on the image of Lena.

Figure 3.4 Original image and Sobel gradient.

3.1.2 Laplacian of Gaussian (Second Derivative)

The Laplacian of Gaussian filter (LoG) is a convolution of a 2^{nd} derivative filter. The LoG value of an image region of rapid intensity is often used as an edge detector. This approach is a specific spatial filter with impulse response related to Gaussian signal shape.

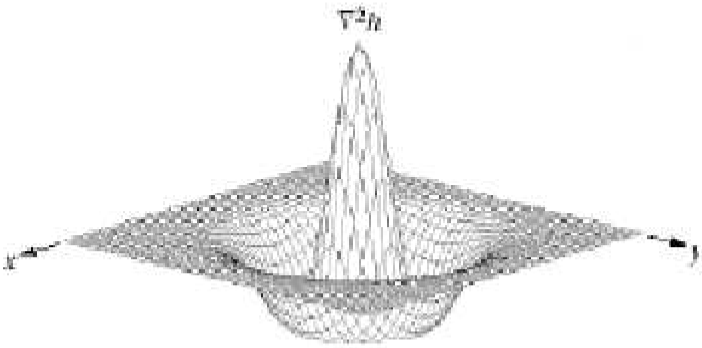
0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

Figure 3.5 Laplacian masks.

Because the Laplacian kernels are approximating a second derivative metrics on the image, they are very sensitive to the noise. To depress noise, the image often smoothed by a Gaussian low-pass filter that reduces the high frequency noise components prior to the differentiation step Laplacian filter. Therefore, since the convolution operation is associative, we can convolve the Gaussian smoothing filter with the Laplacian filter first of all, the LoG filter, and then convolve this hybrid filter with the image to achieve the required result. The function of Laplacian of Gaussian is

$$LoG(x,y) = -\frac{1}{\pi\sigma^4}\left(1-\frac{x^2+y^2}{2\sigma^2}\right)e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{7}$$



0	0	−1	0	0
0	−1	−2	−1	0
−1	−2	16	−2	−1
0	−1	−2	−1	0
0	0	−1	0	0

Figure 3.6 LoG 3-D plot and mask. [17]

Figure 3.6 shows a 3-D plot of the function of the LoG function. Also shown is a 5 * 5 mask that approximates Eq. (7), which is not the unique expression. The coefficients must sum to zero, so that the response of the mask is zero in taking gradient of a constant. Therefore, the LoG is a linear operation and this linear property gives its implementation more flexibility that it can be decomposed into separate linear filters. Employing the functions separately often has advantages over using a single composite mask. The size of masks can be smaller and the function can be easier to implement. Similar to the LoG, difference of Gaussian (DoG) approximates the LoG filter with combination of two different Gaussian filters. The result of the DoG filter is similar to that of the LoG filter, but it has the advantage of computation complexity.

Figureure 3.7 illustrates the output of the LoG filter on the image of Lena compared to the original image. Note that the noise components are much higher than Figure 3.4 shows, which is the serious side effect of 2^{nd} derivative based technique.

Figure 3.7 Original image and LoG result.

3.1.3 Wavelet Transform (Frequency Domain)

Wavelet transform [14] is a spatial decomposition that is done along the horizontal and the vertical directions. Wavelet transform also contains frequency information in each sub-band. Therefore, the transform coefficients reflect the energy distribution of the source image in both space and frequency domains. Wavelet transforms are broadly classified into the discrete wavelet transform (DWT) and the continuous wavelet transform (CWT) ,and we briefly describe the DWT.

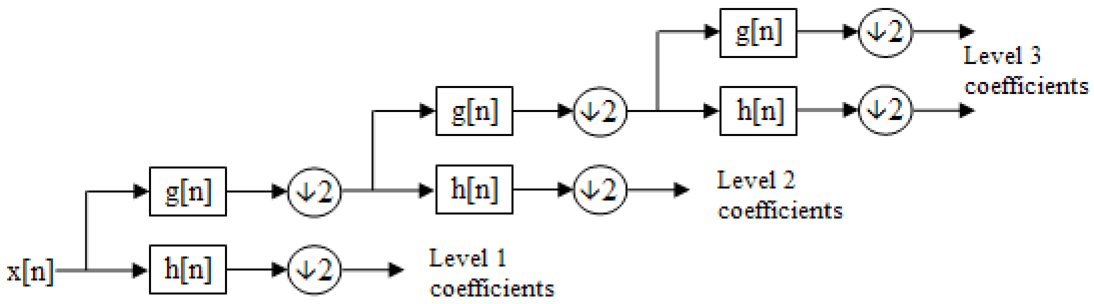


Figure 3.8 A 3 level filter bank. [3]

The decomposition of wavelet transform is repeated to further increase the frequency resolution and the wavelet coefficients are produced by the high-pass and low-pass filters and then down-sampled. This is represented as a binary tree with nodes representing a sub-space with a different time-frequency localization. Figure 3.8 [3] shows a 3-level filter bank The same process can generate an N-level DWT decomposition.

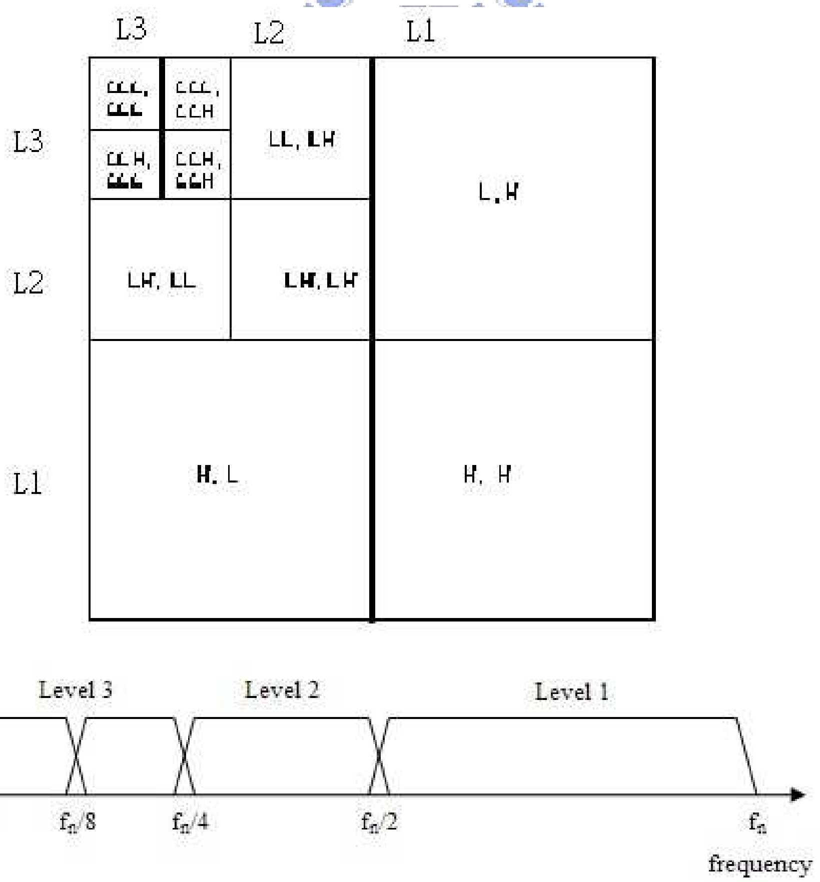


Figure 3.9 3 level decomposition in spatial and frequency domain. [3]

Figure 3.9 [3] shows a 3 level wavelet decomposition. Because the DWT's scalable kernels are used as lowpass and highpass filters, most Fourier-based filtering techniques have an equivalent wavelet domain approach. So the wavelet transform benefits greatly from the filtering techniques that reduces noise effect. Figure 3.10 illustrates the result of 2 level wavelet transform on the image of Lena. The high frequency bands may be use used for edge detection and thus representing focus value.

Figure 3.10 Original image and its wavelet transform.

3.2 Search Algorithms

In section 3.1, the definitions and usage of the focus value and the focus curve are introduced. It seems that an auto-focusing process is quite simple. It simply finds out the global maximum points of a focus curve. The only thing to do is to choose an efficient search or sort algorithm and implement it. However, a focus curve can only be acquired using offline simulation or a full scan operation. Typically, the focus curve is used to train the auto-focusing system in advance. An apparatus only receives or detects a single focus value at a time, not the entire focus curve at one time. Therefore, we need a search algorithm that identifies the best focus position with one by one sequential focus values.

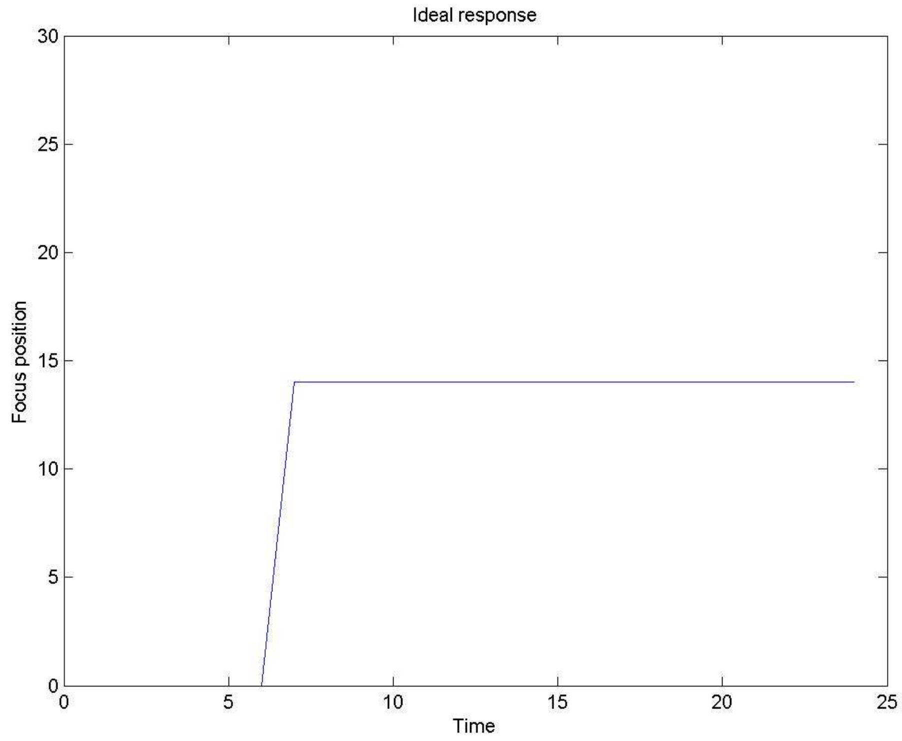


Figure 3.11 Ideal output response of an auto-focusing system.

Before introducing of practical algorithms, let us consider a fundamental problem first, the ideal system response of an auto-focusing system. What the auto-focusing system does is adjusting the focus position to the location where the image looks sharp. It is reasonable to conclude that the best response of an auto-focusing system is near a step function response, as shown in Figure 3.11. But this is impossible for a practical auto-focusing system because of the following reasons.

1. An auto-focusing system is a feedback control system. A feedback control system cannot produce an instantaneous response without processing delay. There must be a system delay that tilts the slope of response curve.
2. The input is not the correct focus position. The input of an auto-focusing system is not the correct focus position but the focus values, which are clued of the correct focus position indirectly. Since the input is indirectly, it is hard to design a linear auto-focusing system to match the optimal response.
3. There are always delays and noise in a practical system, which causes the system response

imperfect.

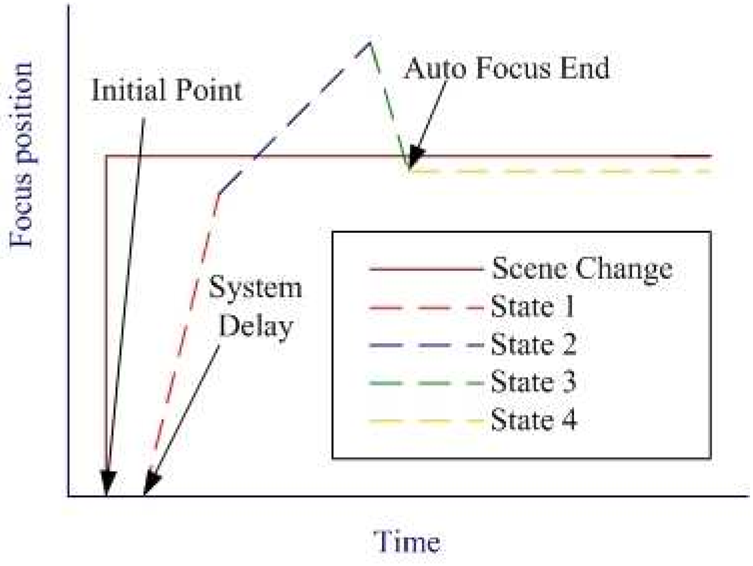


Figure 3.12 A general output response of an auto-focusing system.

Figure 3.12 contains a general model of output response, plots of focus position versus time, for a practical auto-focusing system. The solid line, a step function response, indicates the ideal auto-focusing result, and the dot line, the general output response, shows the various stages in the search algorithm. The dot line state 1 and the dot line state2 are the main search process. The state 1 limited by the speed of the system response is a process of moving current focus position quickly to the near optimal position or, in other words, to the position with a high focus value. The state 2 is a process of identifying the position with the maximum focus value, the optimal focus position. Usually, the system goes beyond the focus position due to delay and needs to come back. The state 3 is a process of re-adjusting the focus position from the overshoot position to the optimal focus position found at state 2. The state 4 is a process of locking the current focus position and detecting whether the image object is changing its location and then a new auto-focusing process needs to execute again. Compare to the ideal step function response that contains only two line segments or states, state 2 and state 3 represent the imperfection of a practical auto-focusing system. The state 2 and state 3, also called over-focus, is one main indicator to judge the efficiency of a search algorithm. Therefore, the correctness (for the given data) or the performance (in real environment) determines whether a search algorithm is useful or not. A robust auto-focusing algorithm is better than a fast one, although a fast algorithm is more and more desirable for today's digital camera system.

3.2.1 Full Search algorithm

The full search is the simplest and most robust search algorithm. Scan all the focus positions from far to near and choose the position with the maximum focus value for the correct solution. Full search algorithm provides a global maximum focus value, even though the system has serious noise and with a disturbed focus curve.

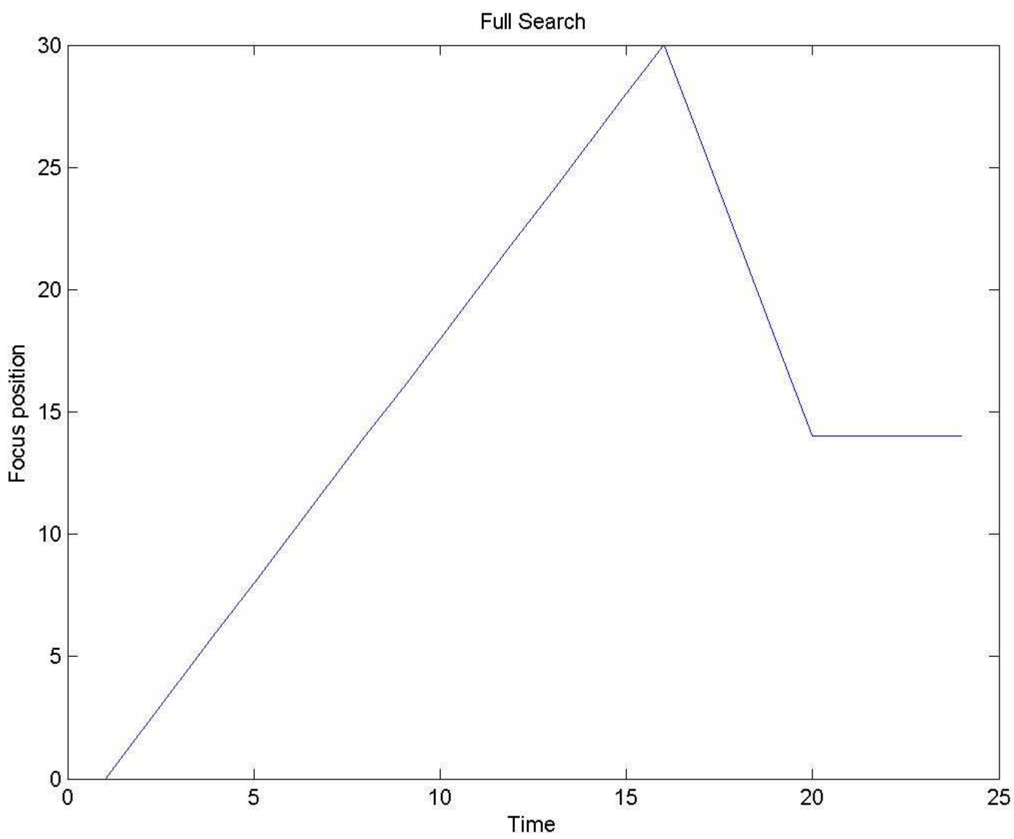


Figure 3.13 Output response of the full search algorithm. The best focus position is 14.

The full search algorithm is very robust but inefficient in time. The full search algorithm has a serious over-focus, the long line segments of state 2 and state 3. It is also difficult to implement a full search algorithm for the current camera apparatus because of the large amounts of data. For example, there are 33792 focus positions in the SONY EVI-D100 camera. The modified full search algorithms, such as fabonacci search, reduce the number of search points and accelerate search process and then the output response has a steeper slope. The full search algorithm is not a mainstream in the existing auto-focusing system, but it provides a reference for solving difficult cases.

3.3.2 Climbing Search Algorithm

The climbing search algorithm [18] is a fast search algorithm with an adaptive step size. This algorithm is developed based on the shape of the mountain-like focus curve, and finding out the optimal focus position at the maximum focus value is similar to climbing to the summit of a mountain. The climbing search algorithm reduces the necessary search steps to increase search efficiency that but suffers from serious noise and with a disturbed focus curve.

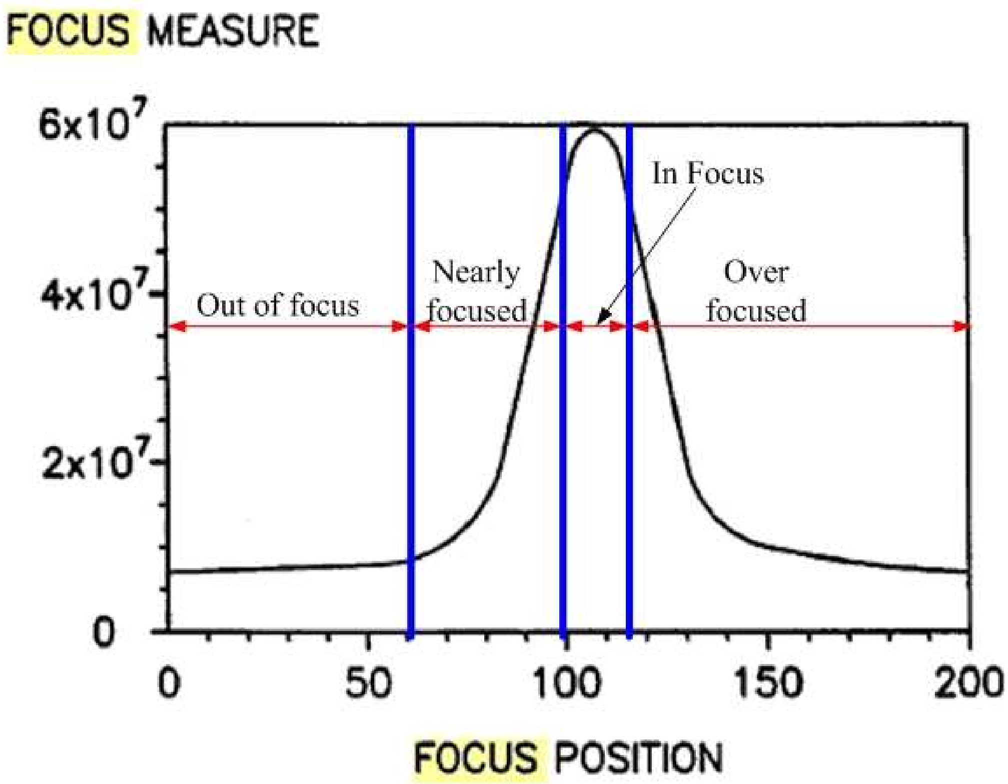


Figure 3.14 The four states of the climbing search algorithm.

A basic climbing search algorithm contains four stages [5]: out-of-focus, nearly focused, over-focused, and in-focus. Figure 3.14 shows each stage on the focus curve. The out-of-focus stage means the current focus position is far away from the optimal focus position. The nearly focused stage means the focus position is close to the optimal focus position. The over-focused stage means the focus position goes beyond the optimal position and becomes less focused. The in-focus case means the focus position is esteemed as the optimal focus position and the auto-focusing process terminates. Figure3.15 shows the state diagram model of the mountain climbing algorithm, which is widely adopted in electronic circuit design.

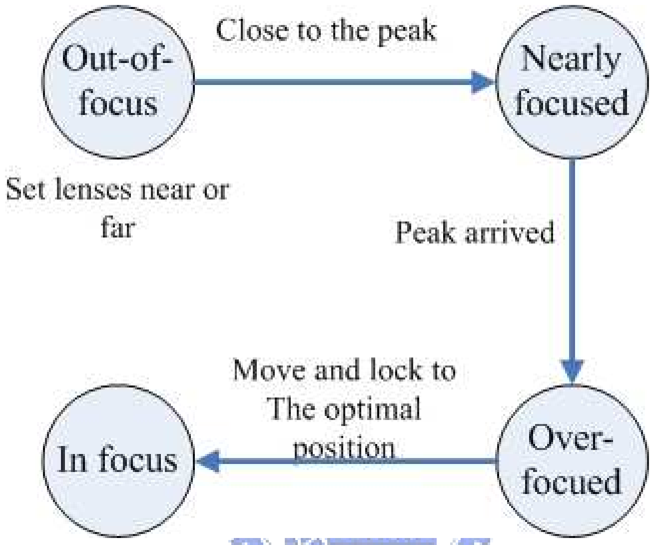


Figure 3.15 The state diagram of the climbing algorithm.

In order to identify the four stages, especially the out-of-focus and the nearly focused cases, we need a measurement typically uses the differences between previous and current focus values or equivalently calculates the slope of focus curve between two focus positions. A positive value indicates the out-of-focus state. When the value gradually moves to 1, it indicates the nearly focused state. A negative value, s the current focus value is lower than the previous focus value, indicates the out-of-focus value. The accuracy of this measurement correlates highly to the performance of the focus value. In other words, the performance of a mountain climbing algorithm depends on the focus value/curve and may be disturbed or corrupted by serious noise and delay.

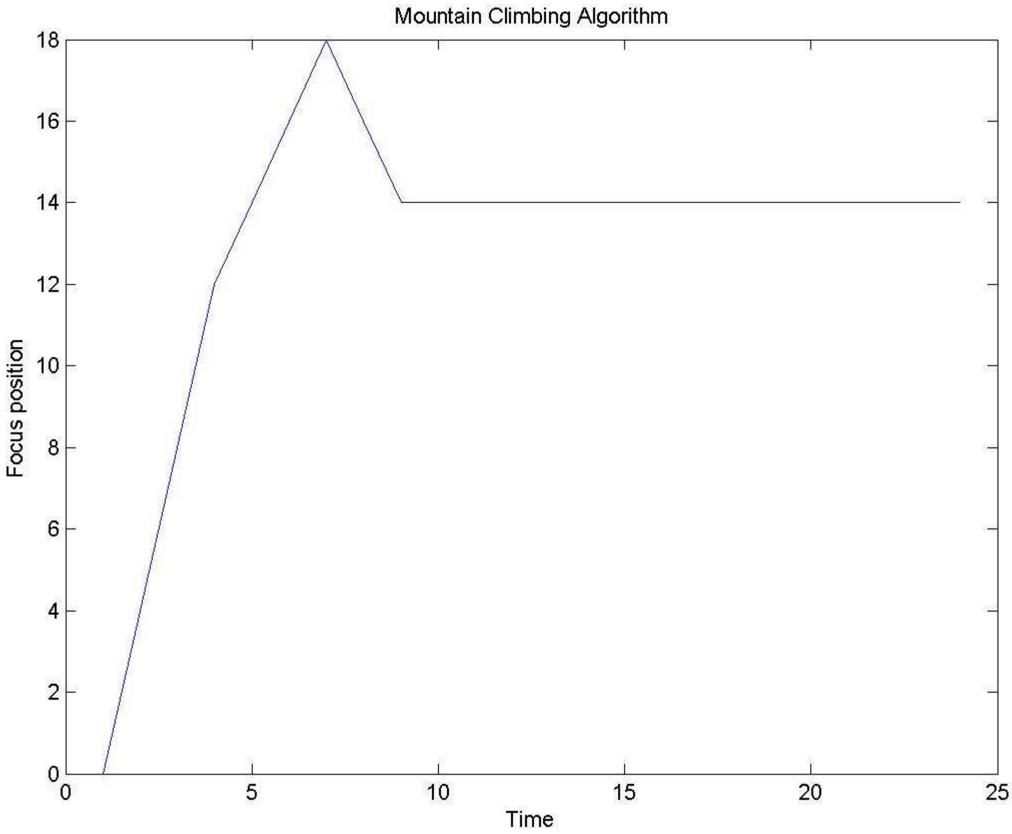


Figure 3.16 Output response of the mountain climbing algorithm. The best focus position is 14.

3.3 Window Selection

Window selection [4] is a common additional function, what is a spatial domain enhancement method in an auto-focusing system. It helps the auto-focusing system to locate the focusing target and improves the curve shape. Window selection has simple computation but powerful results.

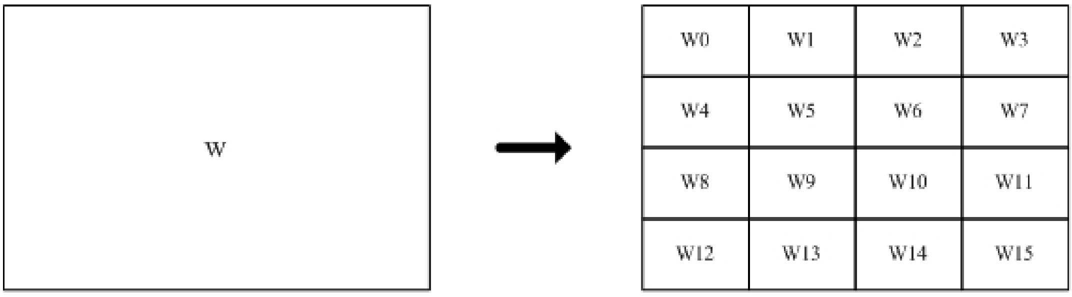


Figure 3.17 The original screen and the 16 focus windows.

Figure 3.17 shows the 16 focus windows model. We divide the original screen into 16 rectangles. The number of windows depends on the design of the auto-focusing system. Fewer windows require fewer memories but also lose some spatial information. Figure 3.18 shows the image “Scarecrow” and the Sobel operator result. It clearly shows that the edge feature (focus cue) is not uniformly distributed but clusters at certain regions. How to locate these hot regions?

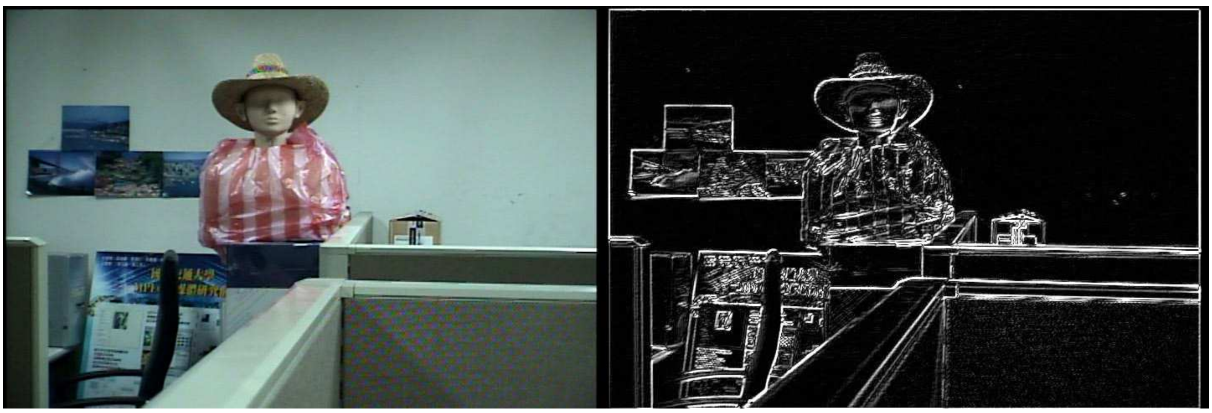


Figure 3.18 The picture and its edge feature. (focus cue)

Of course, the best solution is to identify the foreground or the targets, such as faces or people, and filter out the background that may confuse the auto-focusing system. This approach is an object recognition problem and still a challenge to the real time hardware implementation. Hence, simple window selection is a feasible solution from the viewpoint of hardware design. A good window selection adaptively adjusts the weightings of each window to locate the hot regions.

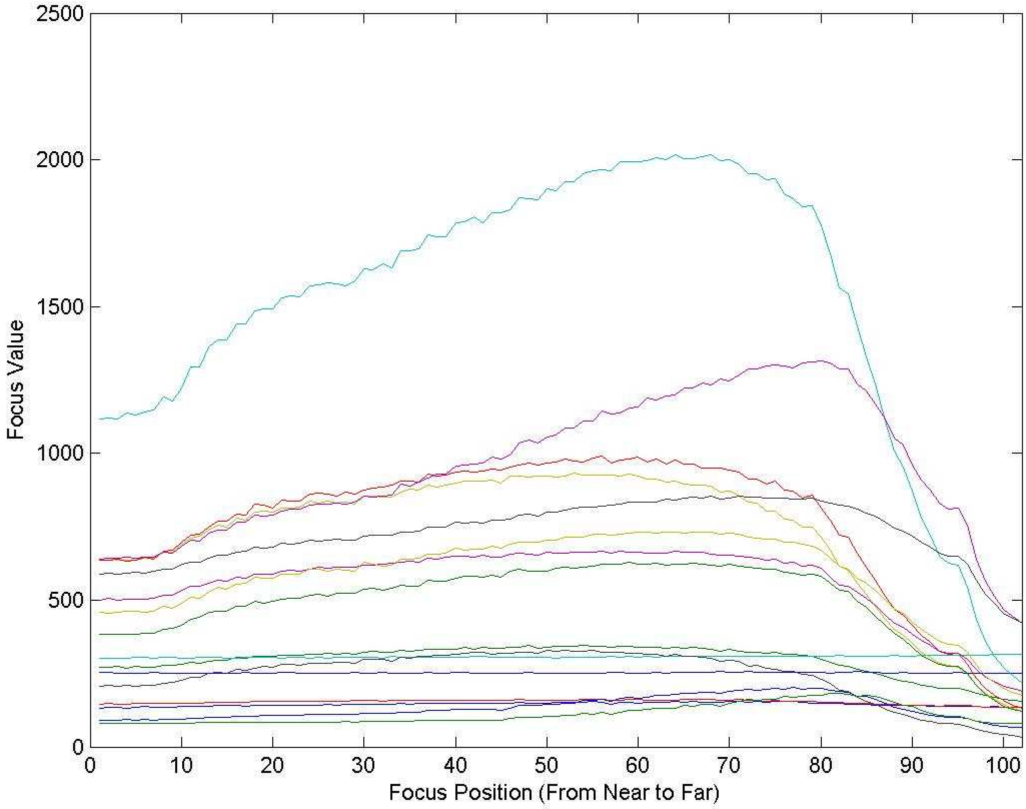


Figure 3.19 The focus curves of each window.

Figure 3.19 shows 16 different focus curves associated with 16 windows. It is apparent that the focus values of different windows behave differently at the same focus position. The overall focus value is a weighted sum of these window values.

$$wfv = \sum w(i)\, fv(i) \qquad (1)$$

The main purpose of window selection is to select the window weighting coefficient for each window.

In general, the center windows have higher weights because most people put their attention on the center of an image. Another approach is that the window with the maximum focus value has the highest weight or chooses it as the focus window. The overall focus value comes from the focus window only.

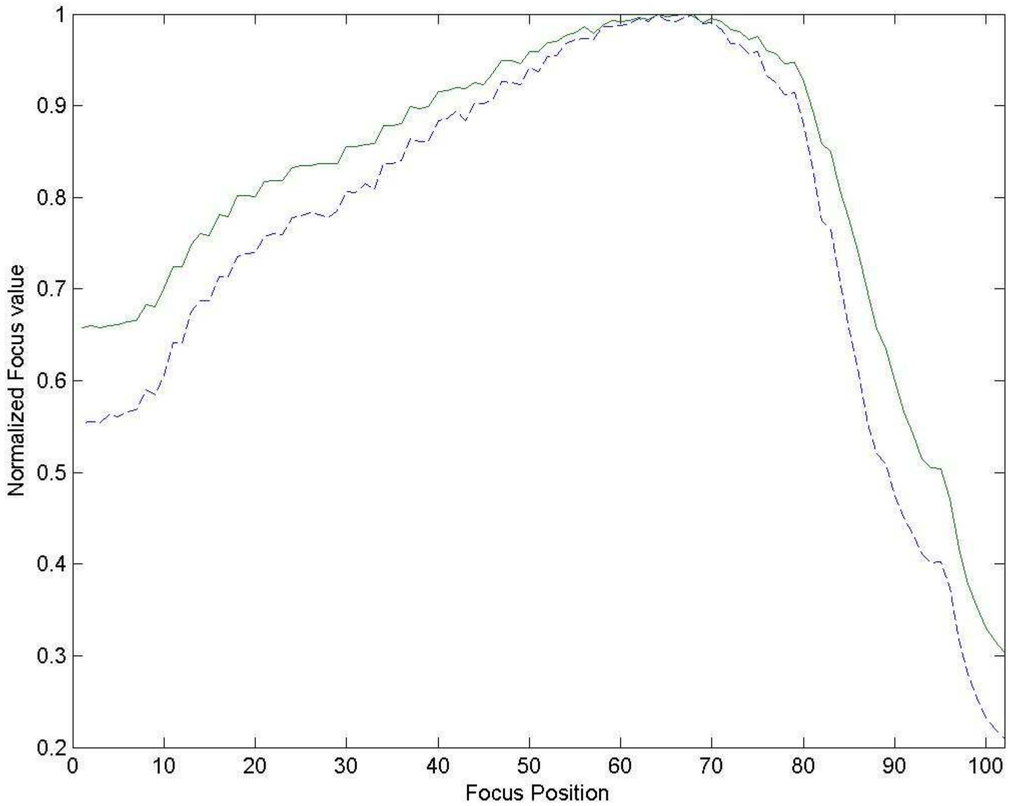


Figure 3.20 The normalized focus curve comparison between no window selection (solid line) and maximum value selection (dot line).

Figure 3.20 shows the normalized focus value comparison between no window selection (solid line) and maximum value selection (dot line) for scarecrow using sobel operator. The dot line expresses the maximum value selection, which has shaper curve.

3.4 Conclusion

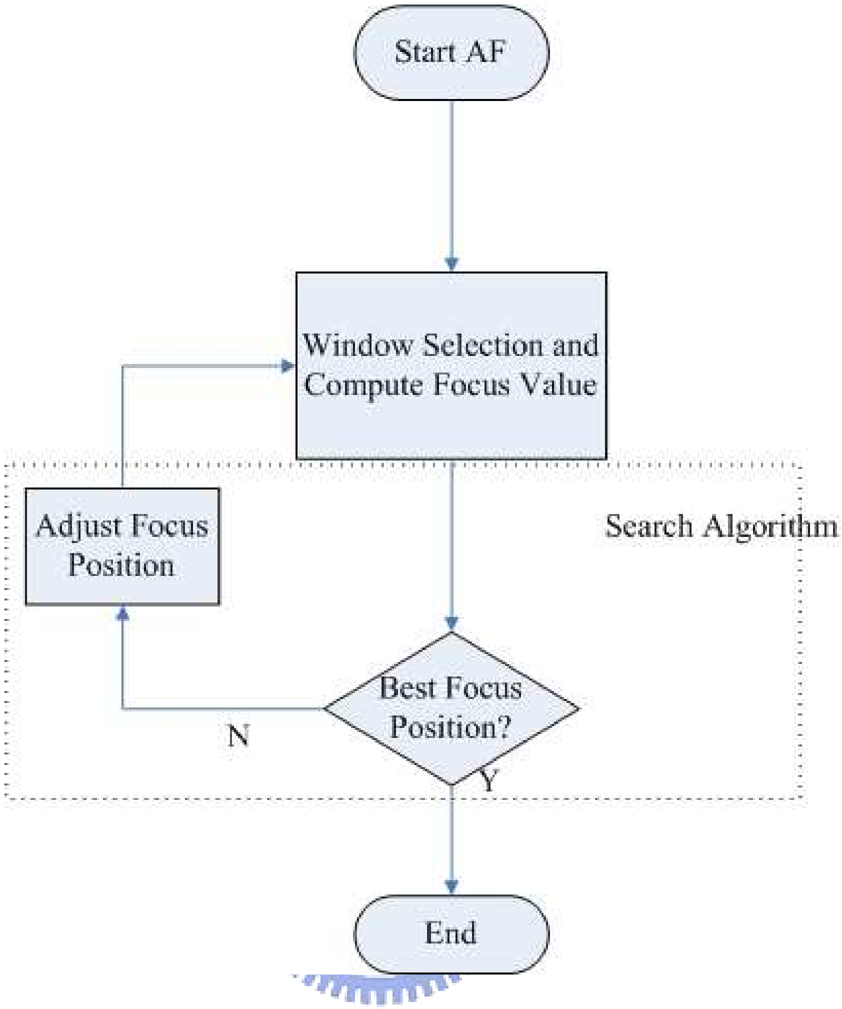


Figure 3.21 Flowchart of an auto-focusing algorithm for still image.

Figure 3.21 shows flowchart of an auto-focusing system for still image. The auto-focusing process begins when the start signal is received and the system gets the focus value with the metrics and window the selection method setting in advance. The auto-focusing will run the search loop continuously until the search algorithm finds the optimal focus position. Thus the system terminates the auto-focusing process and locks the focus position.

Although our purpose is to design a dynamic auto-focusing algorithm, there are parts of algorithms working well no matter for still image or video, such as the focus value metrics and the window selection method. We choose the Sobel operator as our focus value metrics because it is well known and is less sensitive to noise. We also divide our focus window into 16 rectangles and adapt the maximum value selection to enhance our focus curve. The only but core problem is that the search algorithm is designed for a single process and cannot support operating continuously. In the next chapter, we will introduce our dynamic auto-focusing algorithm that revises based on the climbing search algorithm, which is the popular and efficiency search algorithm today.

Chapter 4

Dynamic Auto-focusing System

This chapter proposes a few algorithms for the dynamic auto-focusing system. The dynamic system, such as a video camera, has some different properties comparing to the still image digital camera. One problem is the (video) scene change that the background and foreground are both changed significantly. When the scene change happens, the focus value and focus curve have a large variation. The auto-focusing system must restart to match the characteristic of the new scene. Another problem is object motion that parts of the scene are changing. This problem seems to be similar to the scene change but it is in fact different. The auto-focusing process should continue rather than restart. The last problem is zoom tracking that adjusting the focus position with the given zoom information. Although we can treat this problem as the scene change, finding out the physics relationship between zoom and focus can lead to a good focus prediction. There is one more section discussing the delay effects. Although it is not a dynamic issue, it is a common problem in a practical auto-focusing system. We will describe these problems in details. The basic auto-focusing algorithm of our system is the climbing search algorithm that described in Chapter 3.

4.1 Experiment Set-up

Our auto-focusing simulation environment is not a real-time system. In this section we verify that our experiment really carries conviction. The detail simulation steps are shown below:

Step 1. Control the focus position and the zoom position:

We write a program to control the focus/zoom position of the video cam EVI-D100. The EVI-D100 technical manual provides the command list for controlling the focus position and the zoom position. The control command is transmitted to the EVI-D100 by RS232 cable. This program helps us to assign the focus position as we wish.

Step 2. Capture the image/focus value:

We modify the StillCap sample code of the directshow filter provided by DirectX9.0. We combine the sample code with both the focus value metric and window selection method. This program can read image data from the EVI-D100 and write down the focus value information. We repeat Step1 and Step2 to construct a database that records useful information at each focus position.

Step 3. Matlab simulation:

We write a matlab program to verify our auto-focusing algorithm with the database.

A detail block diagram is shown as Figure 4.1.

Figure 4.1 Block diagram of our auto-focusing system.

4.2 Scene Change Detection

The existing auto-focusing algorithms usually put their emphasis on processing a still image. This section we will discuss a basic dynamic focusing problem: Scene change. Scene change means that the foreground and background are both changed.

Figure 4.2 Two different scenes. The left is “Scarecrow” and the right is “Bookshelf”.

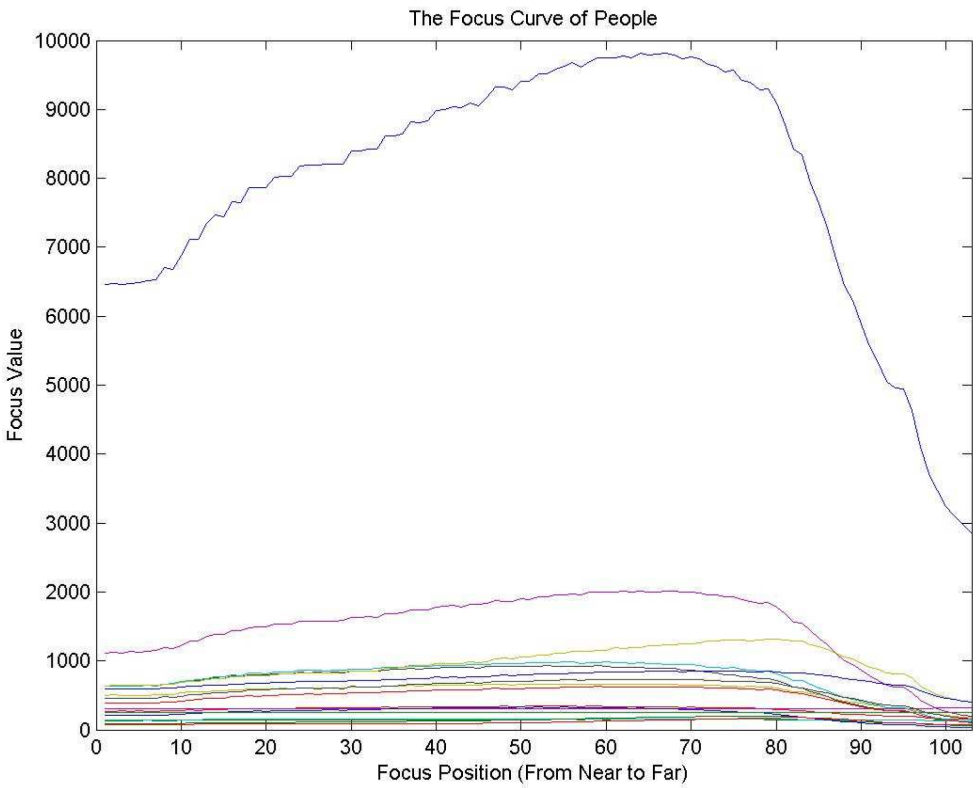


Figure 4.3 The Focus Curve of “Scarecrow”.

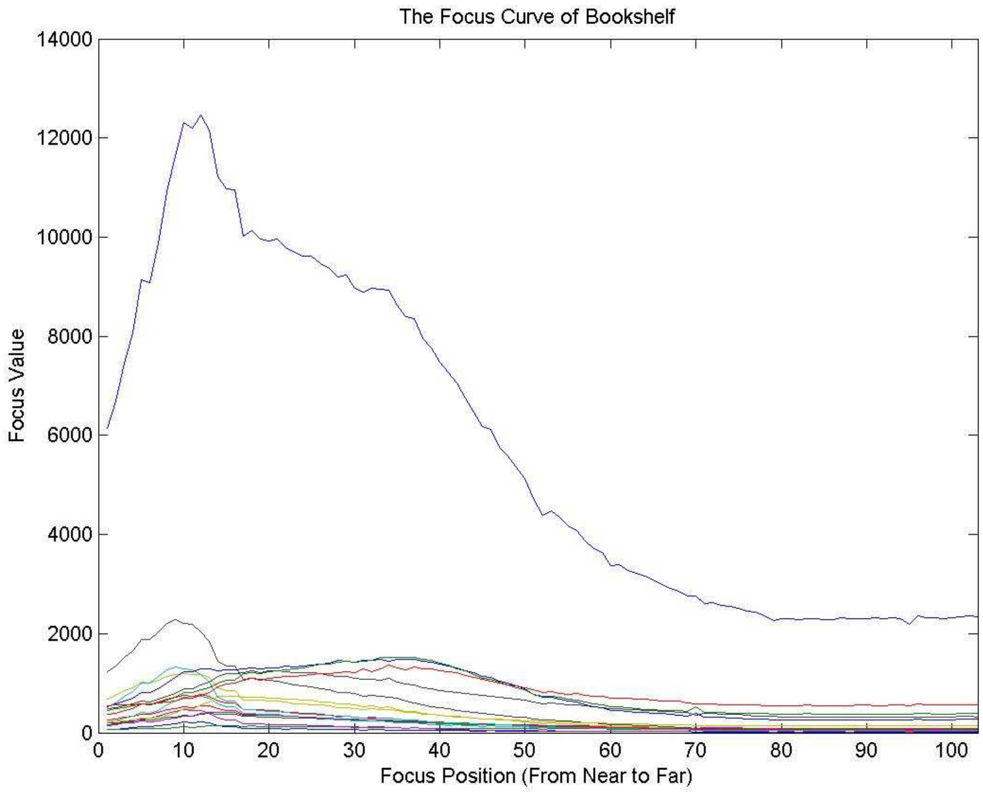


Figure 4.4 The Focus Curve of "Bookshelf".

Figure 4.2 shows two sample scenes. It is intuitive that the samples "Scarecrow" and "Bookshelf" are definitely different. But could the auto-focusing system detect their differences too and do the proper processing? Figure 4.3 and Figure 4.4 show the focus curves of "Scarecrow" and "Bookshelf". Considering a situation that the camera is focusing on the sample "Scarecrow" and the focus position is at 35. Suddenly, the camera makes a turn and the scene changes to the sample "Bookshelf". Although the scene is changed, the focus value of "Bookshelf" is similar with "Scarecrow" at focus position 35. Hence, the auto-focusing system cannot detect this scene change and may make a wrong decision judging the focus position of 35 is the best focus position. It seems that the focus value is not a good measurement for the scene change detection. The focus value varies at every focus position and cannot use to specify a scene. A good scene change measurement must have the following properties: First, it is consistent (has similar value) or nearly consistent at all focus positions of a scene. Second, when the scene change happens, it shows an impulse.

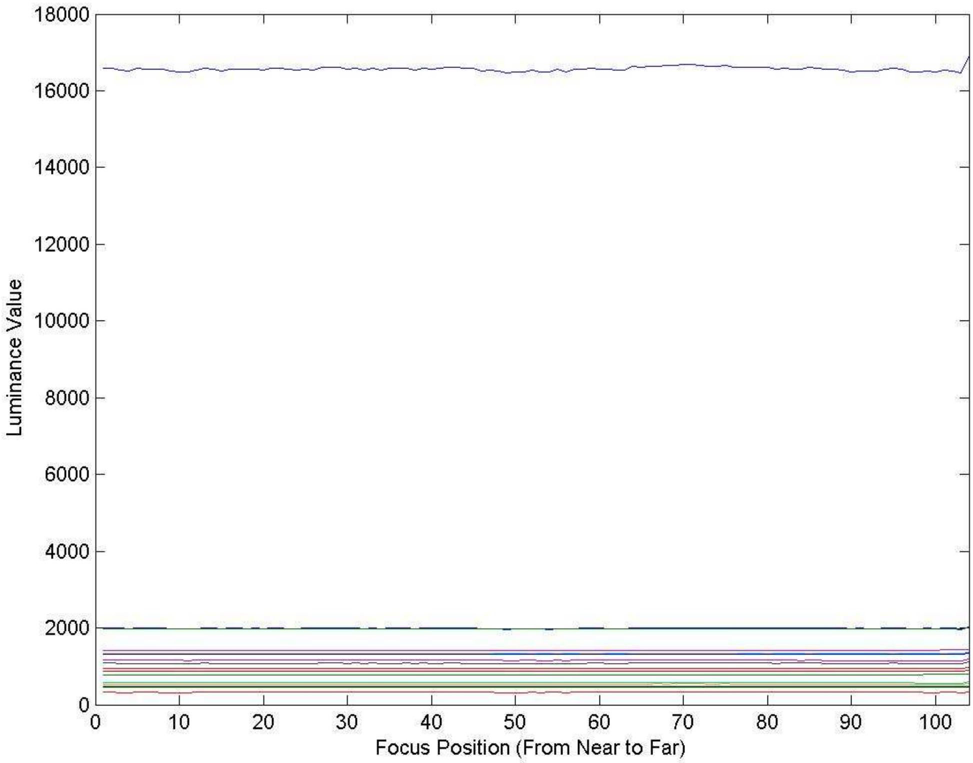


Figure 4.5 The Luminance Curve of “Scarecrow”.

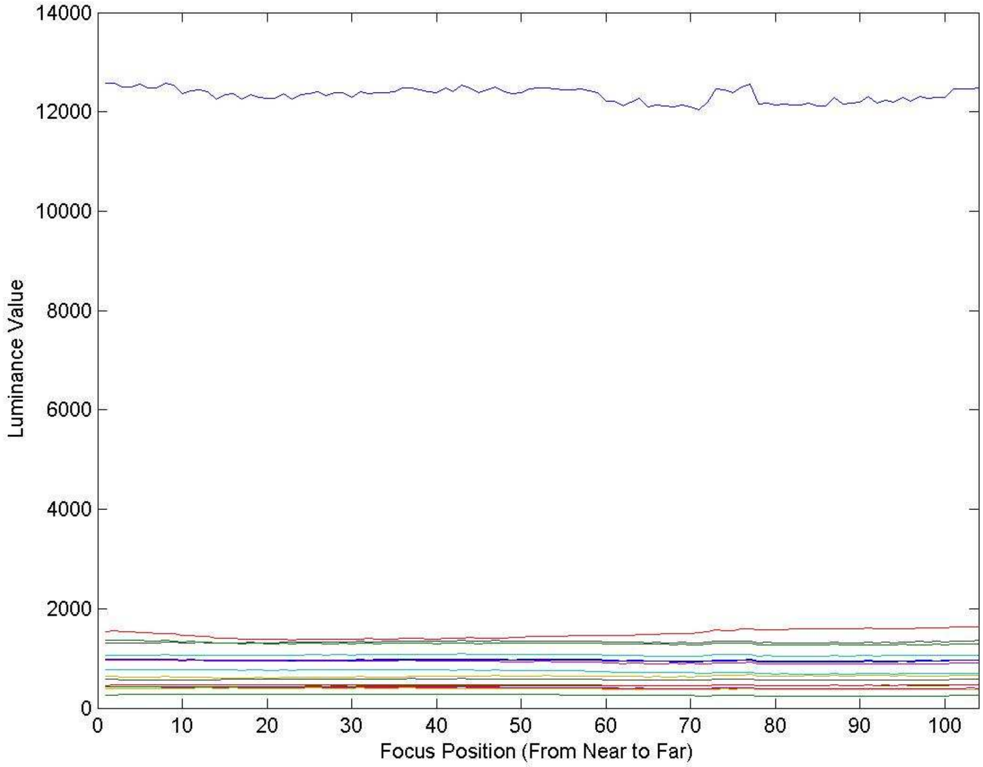


Figure 4.6 The Luminance Curve of “Bookshelf”.

In theory, the blur image is a clear image convolves with a low-pass filter. Hence, moving the focus position is equivalent to convolving the scene with different low-pass filters. The focus value is mostly composed of high frequency components, so it varies at different focus positions. However, the luminance value is mostly composed of low frequency components. The luminance-based method can be a good scene change measurement. Figures 4.5 and 4.6 show the luminance curves of “Scarecrow” and “Bookshelf”. The luminance curve plots the luminance value against the focus position. This metric can also adapts window selection methods in order to preserve the spatial information. The luminance curve is much more smooth than the focus curve.

Based on the luminance value, we proposed a scene change detector: the mean square error of current and previous luminance values. The mean square error is very sensitive to the difference and is also a popular estimator to check the difference of two images.

$$SC_n = E[(lum_n - lum_{n-1})^2], \tag{8}$$

where Lum_n means the luminance value at cycle n and Lum_{n-1} means the luminance value at cycle n-1.

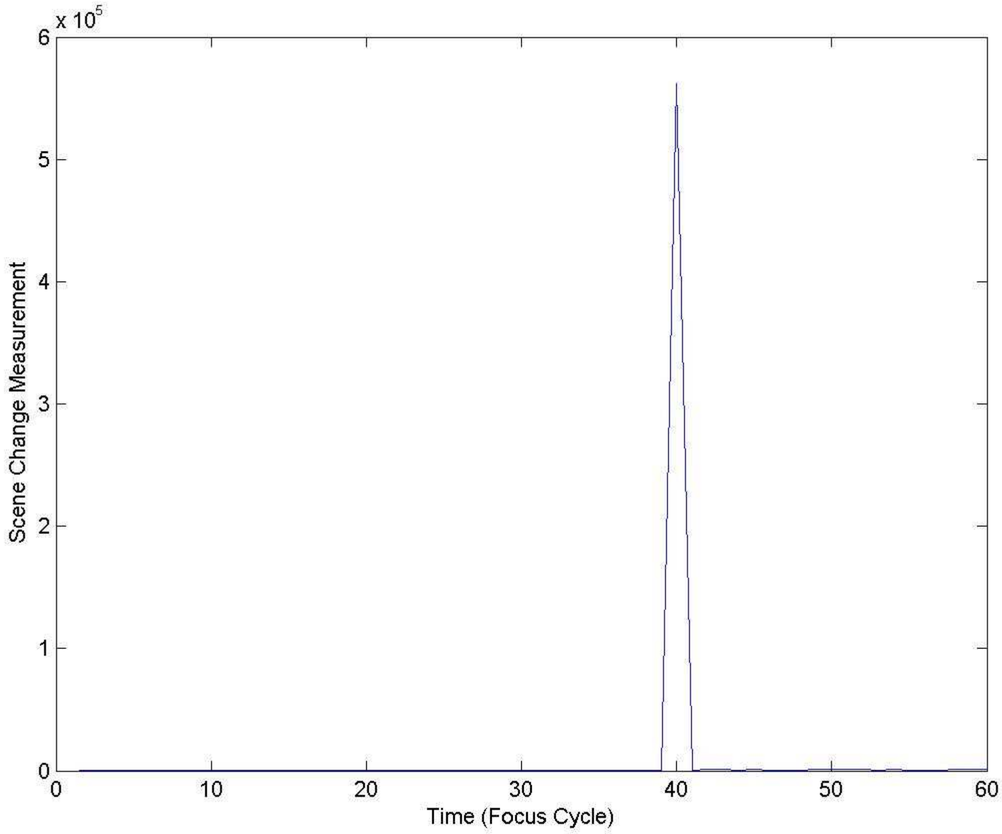


Figure 4.7 The scene change measurement.

Figure 4.7 shows a simulation result of the scene change detection. First, the system focuses on the scene “Scarecrow”. The scene change measurement has a small variation but can be treated as noise. At cycle 40, we change the scene “Scarecrow” to “Bookshelf”. There is an impulse at cycle 40, which means a scene change.

We adapt this scene change measurement into our auto-focusing algorithm. If a scene change happens, the auto-focusing process will be reactivated. Figure 4.8 shows the output response of our auto-focusing algorithm. First, the system focuses on the scene “Scarecrow” and then sets the focus position on 63 at time (cycle) 20. Compared with Figure 4.3, this focus position has the maximum focus value in the scene “Scarecrow”. At cycle 40, the scene “Scarecrow” is changed to “Bookshelf” and the focus position varies too. The auto-focusing algorithm starts to find the new focus position. At last the focus position arrives at 13. Compared with Figure 4.8, this focus position has the maximum focus value in the scene “Bookshelf”. This simulation confirms that our auto-focusing algorithm can detect the scene change well when a scene change happens.

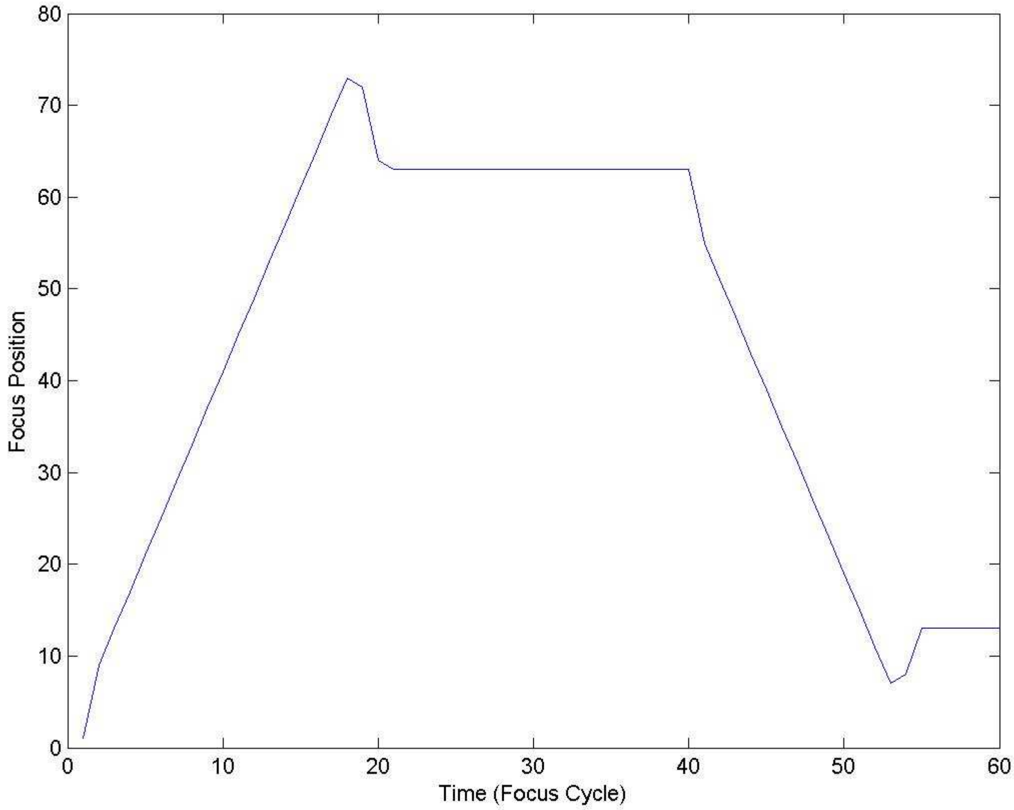


Figure 4.8 The focus result of scene change after focused.

The previous test proves that our auto-focusing system can be automatically activated when the scene is focused. But we also need to test if this measurement still works when the scene change happens during the auto-focusing process is in progress. Figure 4.8 shows that the auto-focusing process of the scene “Scarecrow” is more than 20 cycle. Hence, we change the scene “Scarecrow” to “Bookshelf” at cycle 18 when the focus position of the scene “Scarecrow” is still not found. Figure 4.9 shows the output response under this situation. The auto-focusing system finally sets the focus position at 10. Figure 4.10 shows the focused images at the focus positions 13 and 10 and it verifies our algorithm works well under scene change.

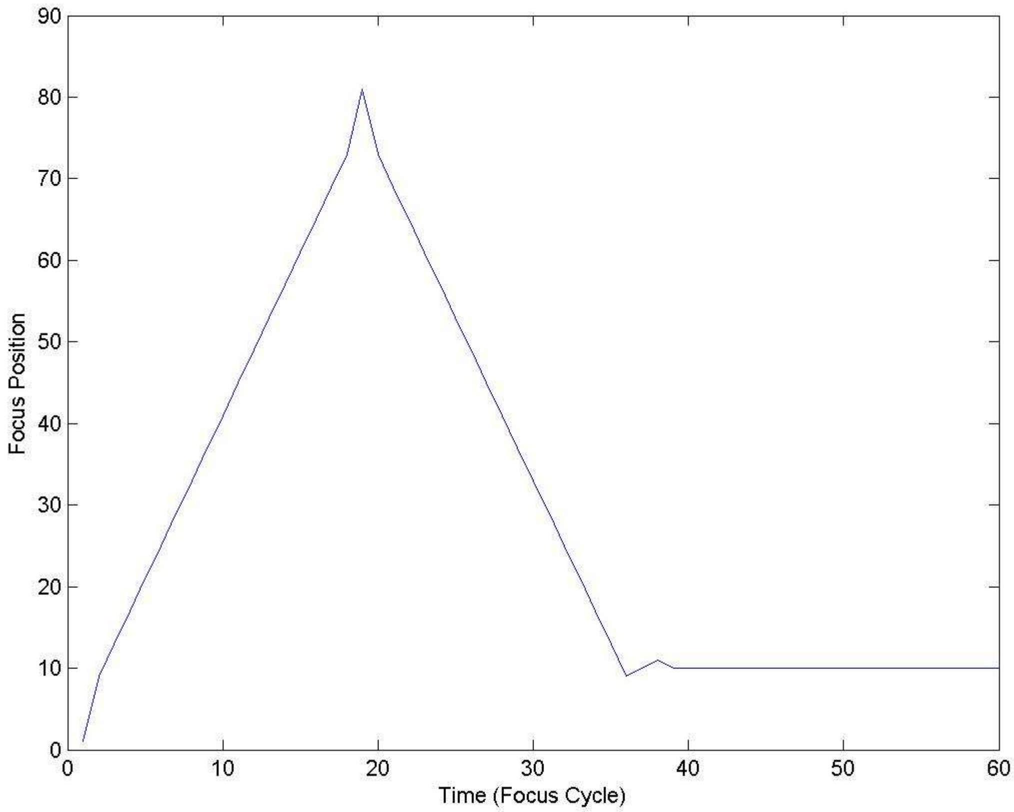


Figure 4.9 The focus result of the scene change appears in the middle of an auto-focusing process.

Figure 4.11 shows the block diagram of the scene change detection algorithm. There are two separate blocks to handle different scene change cases.

Figure 4.10 The auto-focused images.

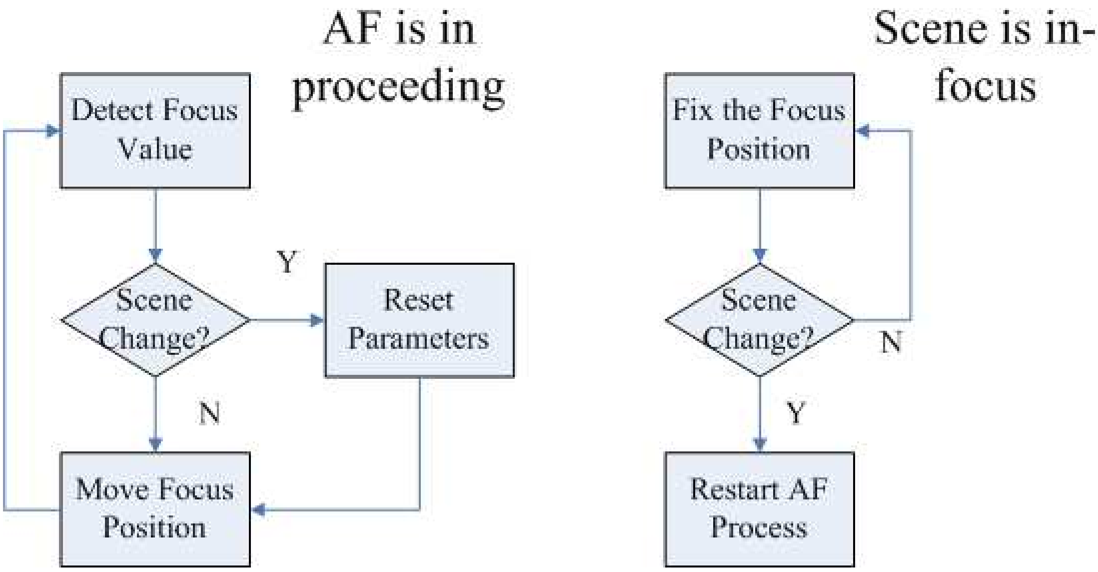


Figure 4.11 The block diagram of scene change detection.

4.3 Local Object Motion

Object motion is a typical dynamic auto-focusing problem. We handle a more restricted case here. The definition of motion here is that parts of a scene moves, rotates, or vibrates regularly and regionally. For example, a working fan rotates. The motion may change the focus value although the object location is not changed. The general auto-focusing system that relies on the difference of focus value usually fails under this situation. Most algorithms shift to a transient state and the focus position is drifting.

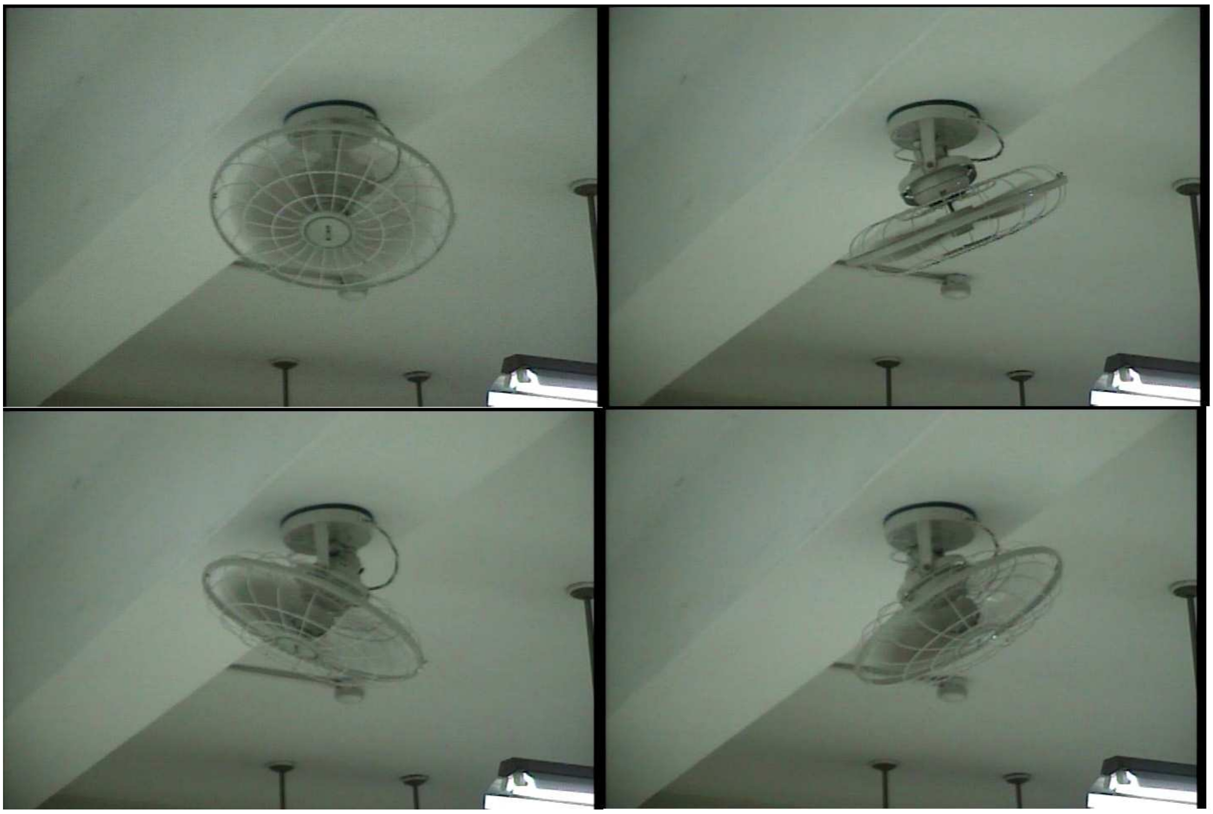


Figure 4.12 The rotating fan.

Figure 4.12 shows four frames of a rotating fan, which is fixed on the ceiling. The texture of the fan has a significant change under different angle. Therefore, the focus curve of each scene may be very different because the focus value metric is usually strongly related to the texture. When measuring on the motion object, there are noticeable changes on the focus value even the focus position is fixed. This phenomenon seems to bring new uncertainty that the focus curve is not sufficient in deciding the correct focus position.

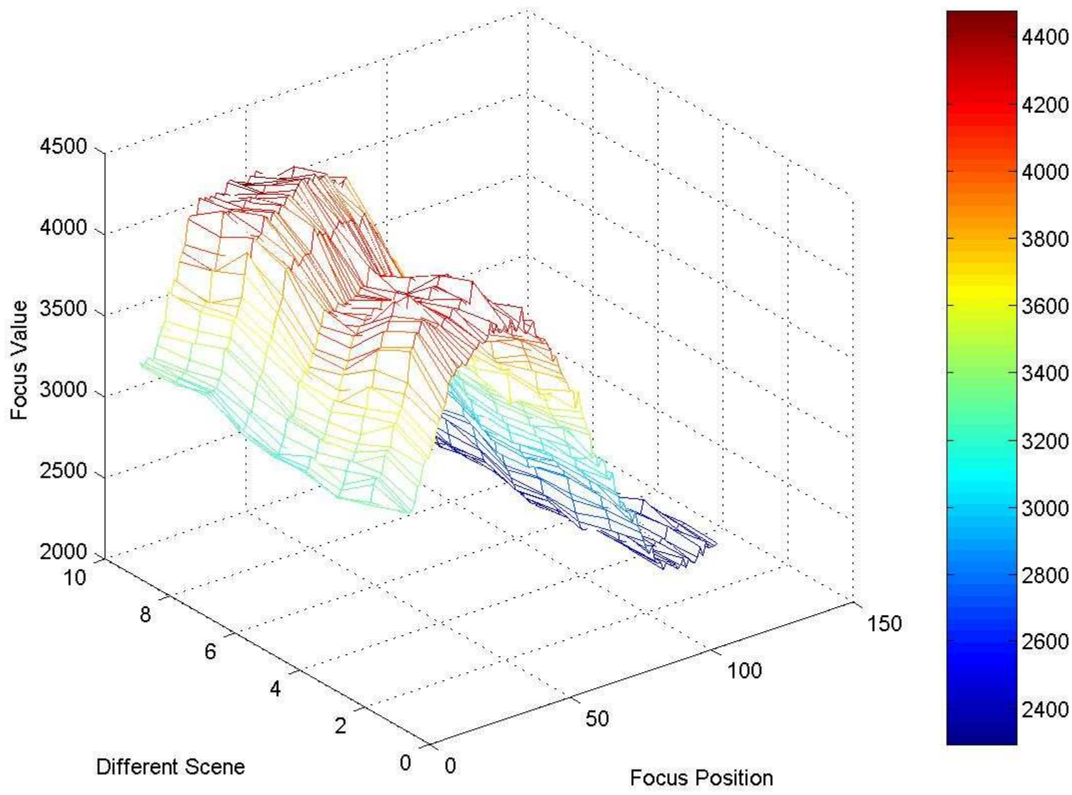


Figure 4.13 3D Focus surface.

Figure 4.13 shows the 3D focus surface that is composed of focus curves at different frame or time. We take nine different samples for the scene of a rotating fan. From the focus surface, we like to find a clue to solve the motion problem. Previously the focus position is the only parameter that changes the focus value, but now the motion object also changes its focus value along the time axis. However, the focus value change caused by motion is fixed but blind. In general, a focus position is associated with multiple a focus value that is one-to-one. But now a focus position corresponds to focus values, i.e. one-to-many. The motion auto-focusing is to find the optimal focus position (a line) on the 3D surface with 1D search algorithm.

There are several issues:

1. The ordinary search algorithm fails. The focus surface is not always convex but somewhere concave. The motion changes may trap the search algorithm into a local

maximum point.

2. Unstoppable auto-focusing process. The focus value changes at a fix focus position. Thus the auto-focusing system continuous the search process without ending. The auto-focusing system stays in a transient state and never stops.
3. False alarms of scene change. Sometimes the boundary between motion and scene change is blurred. In section 4.2, we introduce our scene change detection measurement, a MSE based method that allows a large margin of threshold for a static scene. But Figure 4.14 shows the luminance surface of the rotating fan has a much larger variation than the static scene.

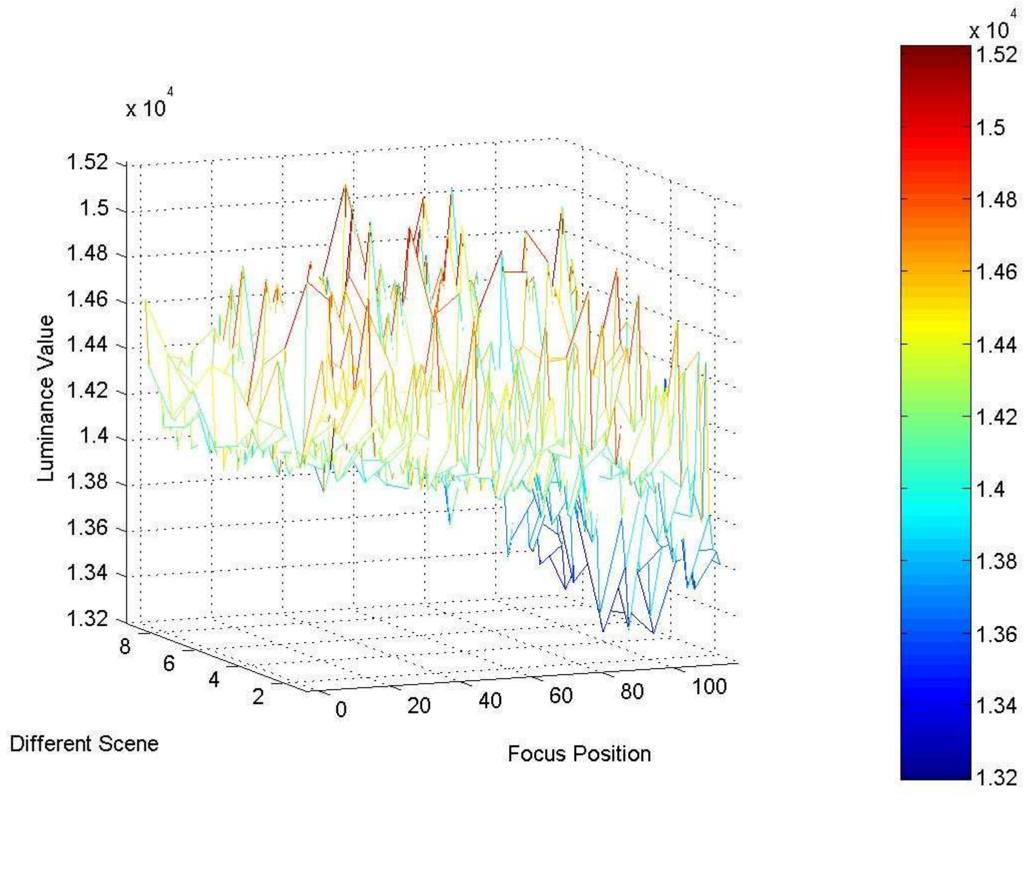


Figure 4.14 The luminance surface.

In order to solve the above problems, we modify the search algorithm. We propose a new focus decision criterion to solve the first fail issue caused by the uneven focus surface. The general focus difference based value is formulated as

$$fs\_slope = \frac{fv_{now} - fv_{before}}{| fp_{now} - fp_{before} |}, \tag{9}$$

where fv means focus value and fp means focus position.

The search generally ends when a negative fs_slope occurs after a series of continuous positive ones. But this accounts the failure because the focus surface has local maximum. Our motion search algorithm now has to endure the accidental negative cases before reaching the global maximum value. We introduce new parameters fs_now, fs_pos, and fs_neg that are accumulated metrics of fs_slope. Table 4.1 shows the definitions of these three parameters. Fs_now is an accumulation of fs_slope of the same sign. When the sign of fs_slope changes, we reset the fs_now value to zero. Fs_pos is set when fs_now is bigger than the threshold value T. fs_pos replaces fs_slope to determine whether the slope of the move is positive. Fs_neg is similarly to defined for negative slopes.

Table 4.1 The threshold parameter.

Symbol	Definition
fs_now	An accumulation of fs_slope. Reset to zero when the sign of fs_slope changes.
fs_pos	Set when fs_now > T
fs_neg	Set when fs_now < -T

Here, we use the values of fs_pos and fs_neg to carry out our search algorithm. The procedure is very similar to the general climbing search algorithm introduced in section 3.2. The new parameters can still use the hard threshold method that eliminates the large fluctuation on the focus surface/curve. The drawback of this approach is that it may not reach the optimal solution. The difference of focus values of the rotating fans at different angle magnifies when the focus position is close to the optimal focus position. This fluctuation difference is too big to be eliminated with a threshold. Although the result is not the best, the focus position determined by this search algorithm is usually good enough to produce a clear image.

Now consider the next two issues due to motion object. In order to reduce the transient problem, we design a sleep mode for our algorithm. The search algorithm enters into the sleep mode after finding the focus position. The auto-focusing function is frozen in the sleep mode. Scene change is the only event that can awake the auto-focusing system. Thus, the focus position is fixed in the sleep mode. Figure 4.15 shows the focus results of the rotating fan. The focus position will finally be fixed at a certain focus position and the focus process is as same as that for still images. The optimal focus position detected at the static image is 40, compared to the result 44.

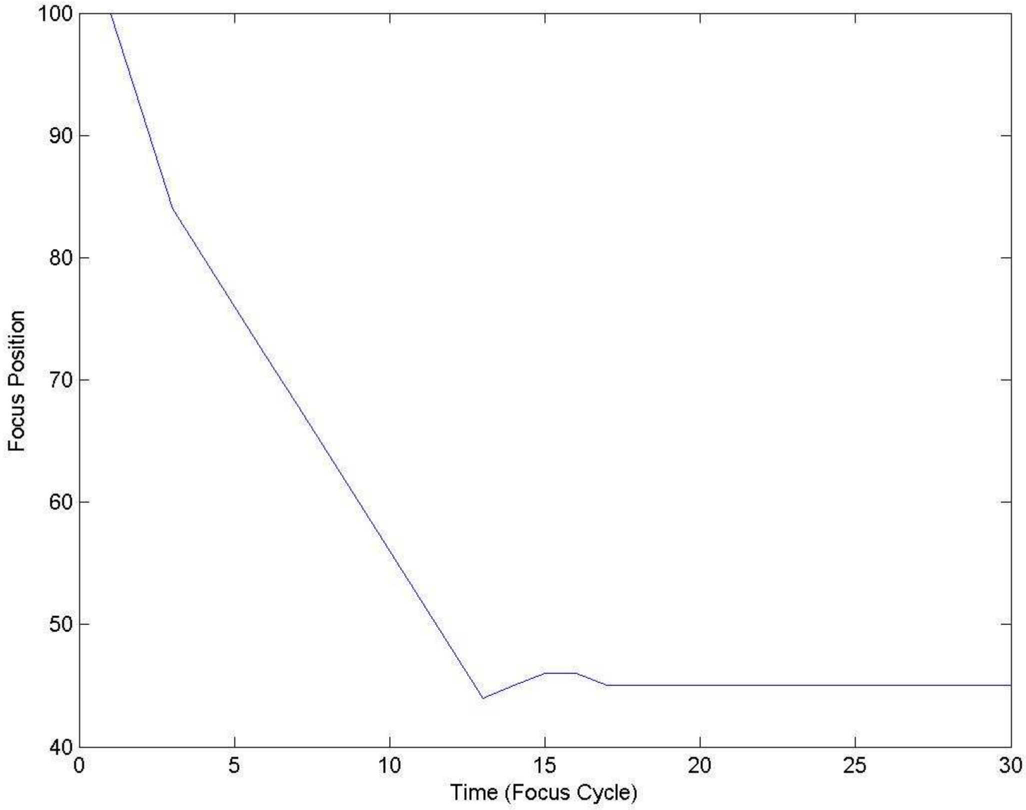


Figure 4.15 The focus result for the rotating fan from near to far.

This design relies on the accuracy of the search algorithm and scene change detection. Hence, the false alarm of scene change sometimes happens and the final focus position is not the optimal one. Our system cannot solve these two problems entirely but often the results are satisfactory. Figure 4.16 shows another focus result that we search for the focus position from the inverse side. The search algorithm awakens from the sleep mode because of a false alarm of scene change. The focus position moves close to the optimal focus position. Figure 4.17 show the image results at focus positions 33 and 44, separately the final focus position in Figures 4.15 and 4.16. Although our algorithm cannot solve the motion problem perfectly, its performance and stability are still acceptable.

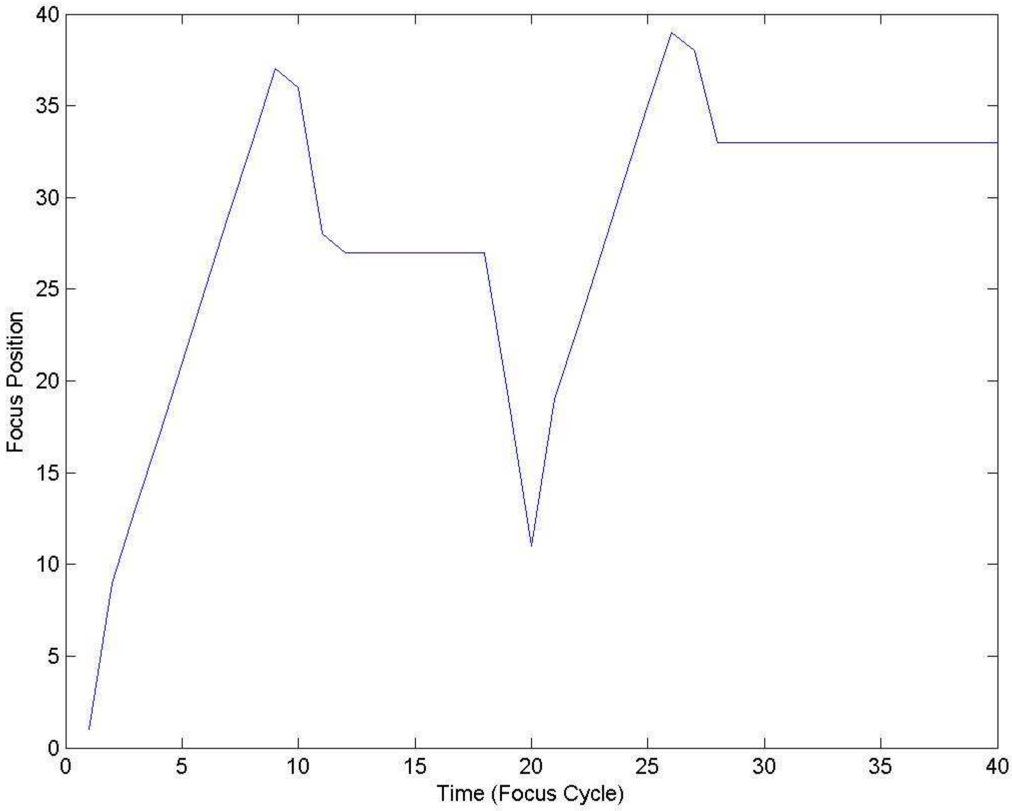


Figure 4.16 Another focus result of the rotating fan from far to near.

Figure 4.17 Focus results at focus position 33 and 44.

4.4 Zoom

Zooming is another common case in the dynamic auto-focusing system. Unlike the scene change detection, detecting the zoom change is an easy job because the system provides this information. Compared to the focus position, there is a parameter called zoom position that shows the lens is in tele or wide mode. The image also has great changes at different zoom position. Although it seems that we can solve the zooming problem and the scene change problem in the same way, the relationship between zoom and focus provides a clue in handling it.

Zooming unlike other dynamic cases is related to the focus position. From the viewpoint of physics, a tele lens magnifies the image like it is closer and a wide lens shrinks the image like it is farther. For example, when the zoom position becomes wider, the focus position should become farer. What we want to do is if we know a focus position under a specific zoom position, we can predict the focus position when the zoom position is changed.

Figure 4.18 The scenes at different zoom position.

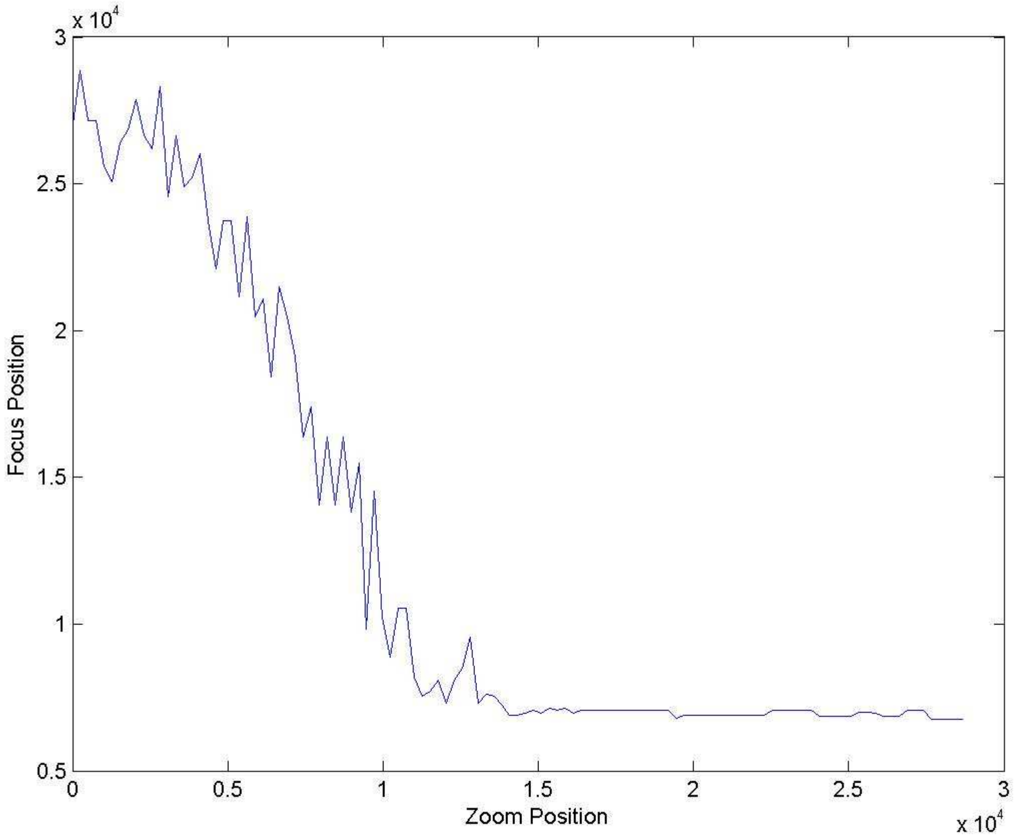


Figure 4.19 A trace curve.

Figure 4.19 shows the trace curve [7] obtained from the scene shown in Figure 4.18, which plots the zoom position from wide to tele versus the in-focus focus position from near to far. Unlike the other data, this set of data is processed by the SONY EVI-D100 built-in auto-focusing system, which means that we do not exactly know its auto-focusing algorithm in detail. Although the image looks like closer and closer, the in-focus focus position is not decreasing smoothly and uniformly.

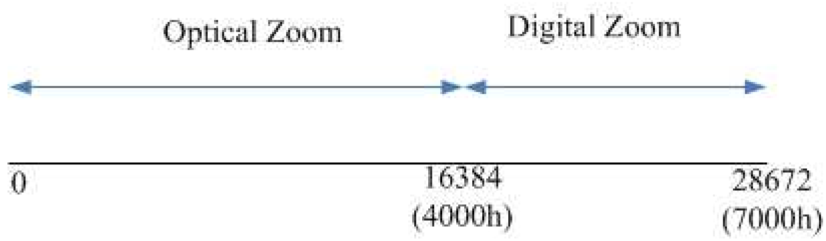


Figure 4.20 The EVI-D100 zoom zone.

One reason is that there are two types of zooms in this camera system. The optical zoom actually changes the focal length of the lenses that brings the image closer physically. Compared to the optical zoom, the digital zoom magnifies the image by the digital image-processing algorithm such as interpolation. In other words, the digital zoom cannot provide new focus information and it has little impact on the auto-focusing system. Figure 4.20 shows the zoom zone and boundary between optical zoom and digital zoom positions. This type of system operation is generally adopted by nearly every camera system, too. The digital zoom with no additional information has very little influence on the focus position and Figure 4.19 supports this assumption that the focus position is not changing much on the digital zoom zone. But, the focus position does not decreases smoothly even in the optical zoom zone. In order to predict the focus position, we need to find the relationship between zoom and focus. This is the so-called zoom tracking problem.

One solution is the table look-up method [8]. The trace curves at different object distance are all stored in the memory. The drawback of this approach is the expensive of large memory and sometimes it encounters the one-to-many mapping problem. Another approach is storing a few the trace curves and do interpolation such as geometric zoom tracking (GZT) [9] and adaptive zoom tracking (AZT) [10]. But our trace curves are not as smooth as their experimental results and the device is not the same. The complexity of the background and the percentage of the foreground limit the performance of zoom tracking. Our purpose is estimating the new focus position approximately that can speed up the auto-focusing process. Therefore, we collect some trace curves and then do curve fitting.

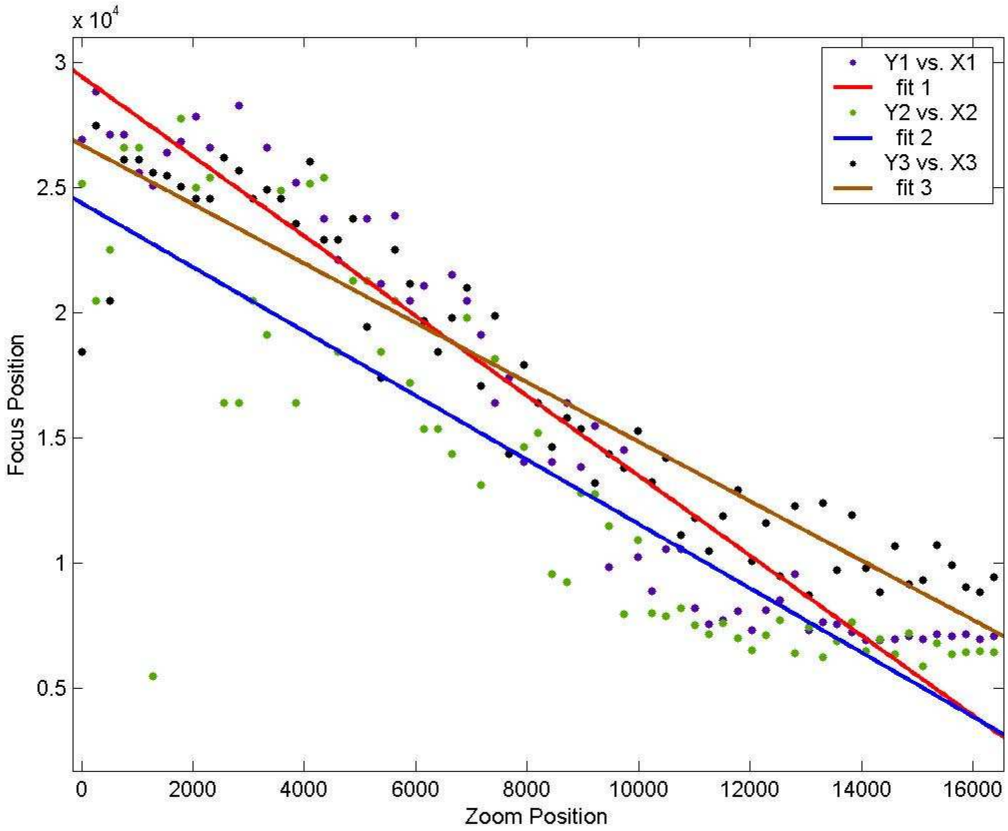


Figure 4.21 Data and the curve fitting.

The optical zoom actually changing the lenses farer or more near is the only predictable zone. Hence, we only select the changes in the optical zoom for the curve fitting. Figure 4.22 shows the fitting results for three different scenes. The fitting function we use is a linear one.

$$Y = aX + b \tag{10}$$

Then we can predict the new focus position as

$$Fp_{new} = a(Zp_{new} - Zp_{ini}) + Fp_{ini} \tag{11}$$

Table 4.2 The curve fitting results

Scene	a	B
1	-1.597	2.944E4
2	-1.284	2.438E4
3	-1.187	2.671E4

Table 4.2 shows our experimental result of the linear curve fitting. We add the zoom

tracking method into our auto-focusing system and compare with the system without this function. Figure 4.22 shows the in-focus scenes at two different zoom positions. We first focus on the closer image and zoom wide at cycle 30. The zoom tracking algorithm predicts the focus position. We also simulate the same auto-focusing algorithm without zoom tracking under the same condition.

Figure 4.22 Two Scenes with different zoom position.

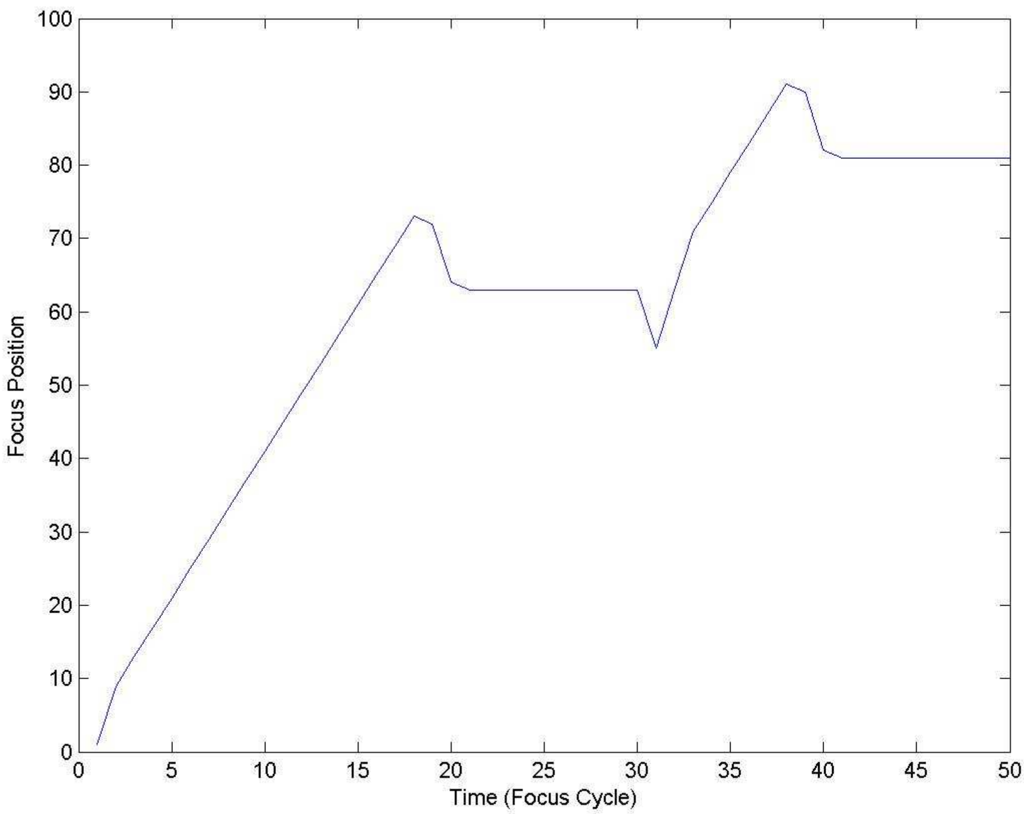


Figure 4.23 The focus result without zoom tracking

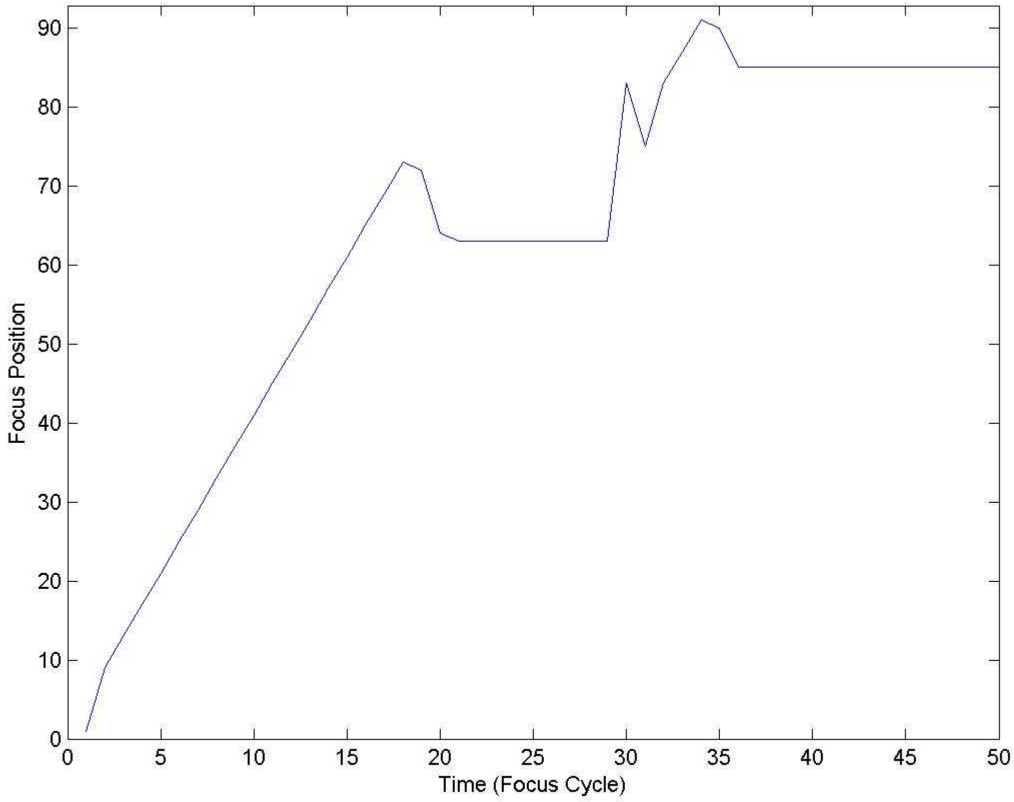


Figure 4.24 The focus result with zoom tracking.

Figure 4.24 shows the focus result without zoom tracking. The zoom position is changed at cycle 30 and it spends 12 cycles to arrive the new focus position. Figure 4.25 shows the focus result with zoom tracking. The zoom position is changed at cycle 30 and predicts the focus position first. The new algorithm spends only 8 cycles to arrive the new focus position, saving 1/3 cycle. These results verify that the system with zoom tracking spends fewer cycles than the system without zoom tracking in finding the focus position. Zoom tracking algorithm helps in saving the in-focus time. Figure 4.26 shows the block diagram of the zoom tracking algorithm.

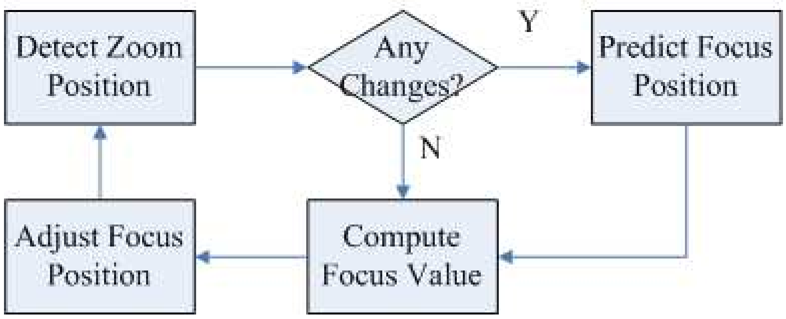


Figure 4.25 Block diagram of the zoom tracking algorithm.

4.5 Delay Compensation

Delay is a common phenomenon in the control system. The cause of delay is reading data or making a complex computation. Delay seems to produce a fixed time shift in the data stream. Hence, delay mainly degrades the search algorithm performance but has slight impact on the focus value metric. Fortunately, delay that occurs in the auto-focusing system is usually a fixed small value.

In general, the search algorithm moves the focus position depending on the difference of the focus value between the current and the previous focus positions. But it is impossible in the real world to obtain the new focus value immediately after moving the focus position. In other words, the search algorithm moves the focus position depending on the information related to the previous focus positions not the current position information. This delay effect is only a fixed latency that all system commands shift in most cases. However, the climbing search algorithm typically ends with detecting a negative slope and fixes the near focus position as the optimal focus position. Hence, the delay causes the climbing search algorithm to stop at an over-focused position that is away from the optimal focus position.

We add a linear prediction method to help our search algorithm in finding the optimal focus position. If there is no delay, locating the focus position is done as soon as the over-focused case is detected. However, the delay problem makes the over-focused case more serious and thus we design a linear prediction to locate the focus position. The prediction is formulated as:

$$fp = \frac{fp_1\,fv_1 + fp_2\,fv_2}{fv_1 + fv_2}, \tag{12}$$

where fp_1 and fp_2 are the two focus positions with two focus values, fv_1 and fv_2. This interpolation has a side effect that the predicted focus position does not exactly have the maximum focus value.

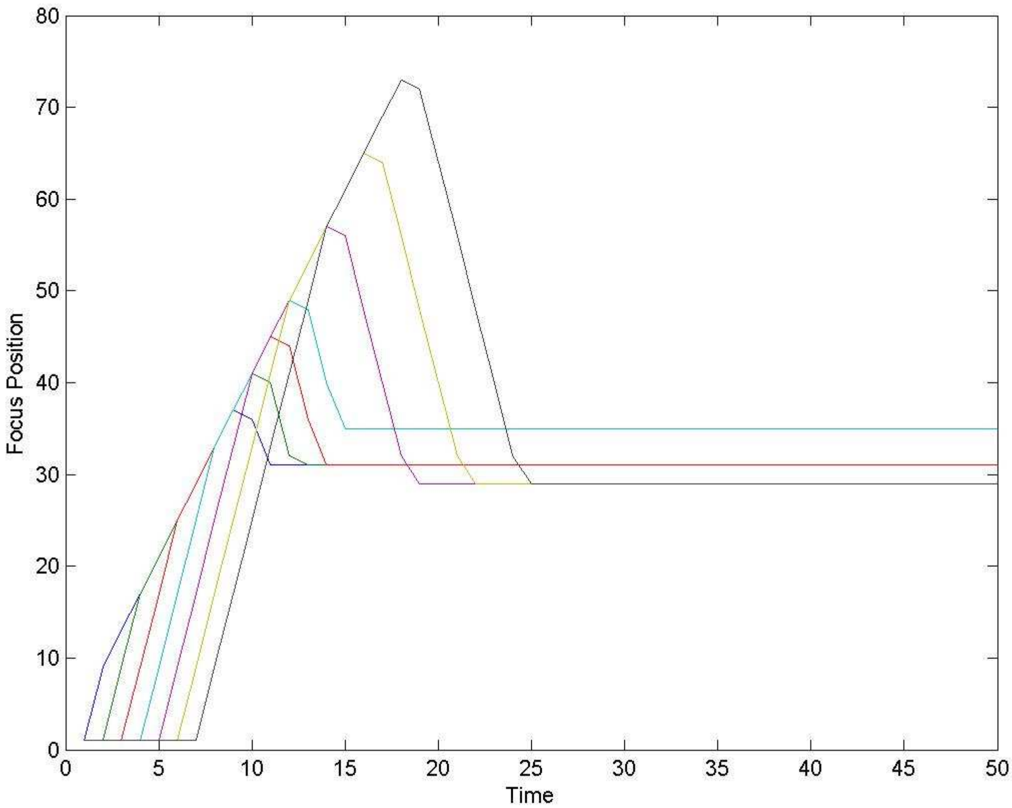


Figure 4.26 Simulated results under different delay values.

Figure 4.26 shows the simulation results of our search algorithm with delay compensation. The different initial time indicates the different curve has different delay value. The auto-focusing system with a longer delay has a more serious over-focus situation that takes additional time to focus. The linear prediction solves the error problem brought by delay and is independent to the length of delay. However, it still cannot reduce the extra time caused by serious over-focused phenomenon. Although our delay compensation algorithm can work under variable length of delay, the performance of our search algorithm suffers severely with long delay.

4.6 Finite State Machine Approach

In the previous section, we describe each element in our dynamic auto-focusing algorithms separately and then integrate all of them into a system. The finite state machine is a popular model for running a software simulation or a hardware implementation.

Table 4.3 State transition table

	1(F)	2	3(F)	4	5(F)	6	7(F)	8	S
1	00x0x	10x0x	11x0x	----	01x0x	----	----	xxx1x	----
2	10x0x	00x0x	----	11x0x	----	01x0x	xxx1x	----	----
3	xxxx1	----	0xx00	----	----	----	----	----	10x00
4	-----	xxxx1	----	0xx00	----	----	----	----	10x00
5	xxx01	----	----	1xx00	0xx00	----	----	xxx1x	----
6	-----	xxx01	1xx00	----	----	0xx00	xxx1x	----	----
7	xxxx1	----	----	----	xx1x0	----	xx0x0	----	----
8	----	xxxx1	----	----	----	xx1x0	----	xx0x0	----
S	xxxx1	-----	------	-----	-----	-----	-----	-----	xxxx0

Table 4.3 shows the state transition table of our auto-focusing system, which the row means the current state and the column means the next states. The transition condition is composed of six parameters: [fs_neg fs_pos fs_max fs_out fs_cha fs_idl]. 0 means the parameter is not set, 1 means set, and x means don't care. The parameter setting is as follow:

fs_neg: Set when the focus value is decreasing. The detail is discussed in section 4.3.

fs_pos: Set when the focus value is increasing. The detail is discussed in section 4.3.

fs_max: Set when the current focus value is the maximum.

fs_out: Set when the focus position is overflow.

fs_cha: Set when the scene change or zoom change happens. All the parameter and data will be reset when it is set. The detail is discussed in section 4.2.

The focus position increases (far) when the state number is odd. The focus position decreases when the state number is even. The state S is a special state that means the auto-focusing process is done tentatively. The initial state is determined to state 1.

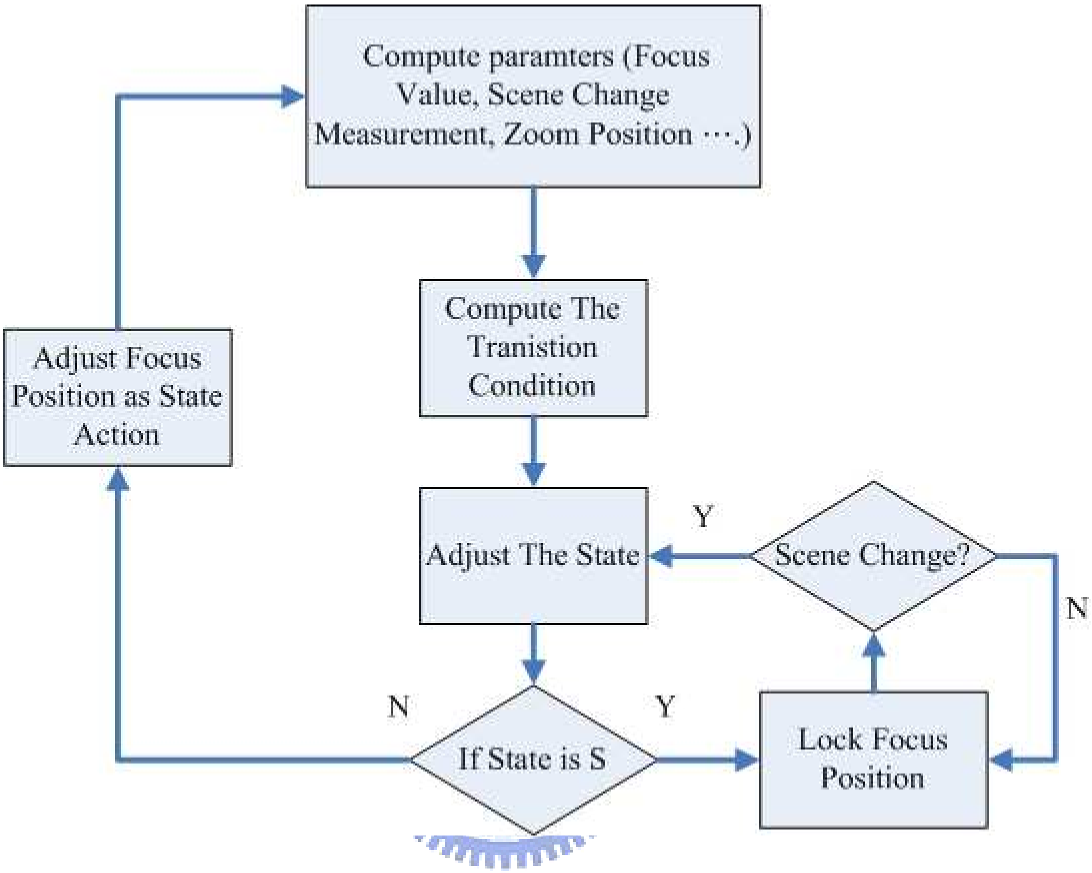


Figure 4.27 Flowchart of our auto-focusing system with a finite state machine approach.

Fig 4.27 shows flowchart of our auto-focusing system with a finite state machine approach. The original part of search algorithm is replaced with the state machines that each state has a unique action for adjusting the focus position. In order to handle the dynamic issues, the system computes much more parameters and translates these parameters into the state condition form. A new state S that locks the focus position helps the system to maintain the focus position temporarily, which allows our algorithm works as well as the traditional ones for the still image. If the algorithm cannot be satisfied, we can revise the state transition table with a more complex state condition form or add new states.

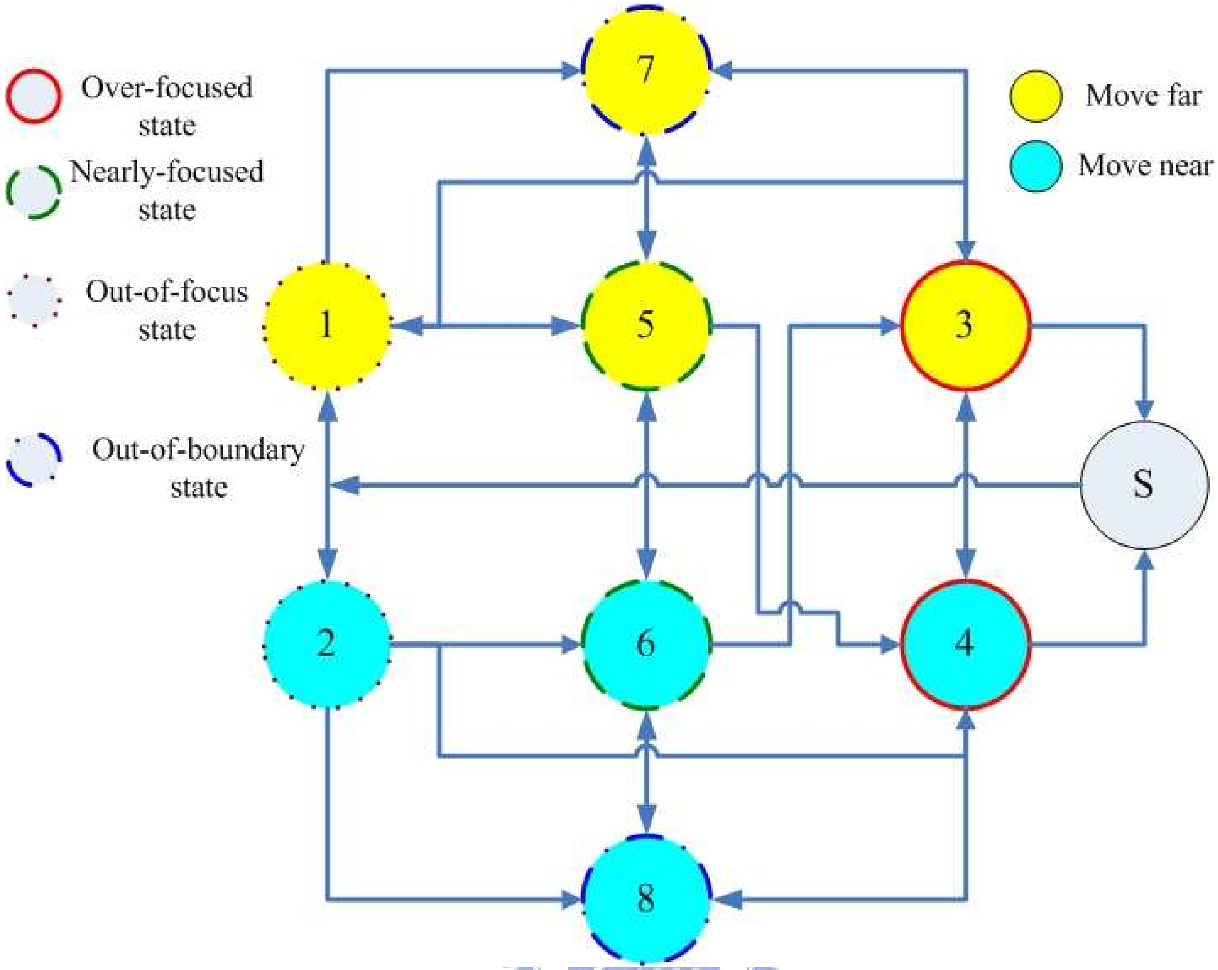


Figure 4.28 The state diagram for our auto-focusing system.

Figure 4.28 shows the state diagram derived from Table 4.6. The states are designed based on the basic climbing algorithm logics with the enhance transition conditions. Here we can understand our algorithm more clearly. For example, focusing a still image from a near initial position works as moving state 1 -> 5 -> 4 -> S sequentially. Considering a scene change case, the state moves as state 1 -> 5 -> 4 -> S ->(scene change happens) 1 -> 2 -> 6 -> 3 ->S. The robust state table maintains the search algorithm working under the dynamic situations.

Chapter 5

Simulation Results

In this chapter, we provide the experimental results of our proposed algorithm. We roughly examine our dynamic auto-focusing algorithm in the previous chapter. For checking the robustness of our algorithms, we give a series of experiments with our overall auto-focusing system. We choose the Sobel operator as our focus value metrics and divide our focus window into 16 rectangles with maximum value selection. The search algorithm is integrated with a finite state machine approach. The captured image data type is 24-bit BMP and the size is 720x480. Because we only compute our algorithm in gray level, we do a color transform in our algorithm first.

In the previous chapter, we mainly analyze the output response to show the efficiency of our auto-focusing algorithm. In order to help understanding our algorithm performance and results, the following section will show the captured images at certain process segment.

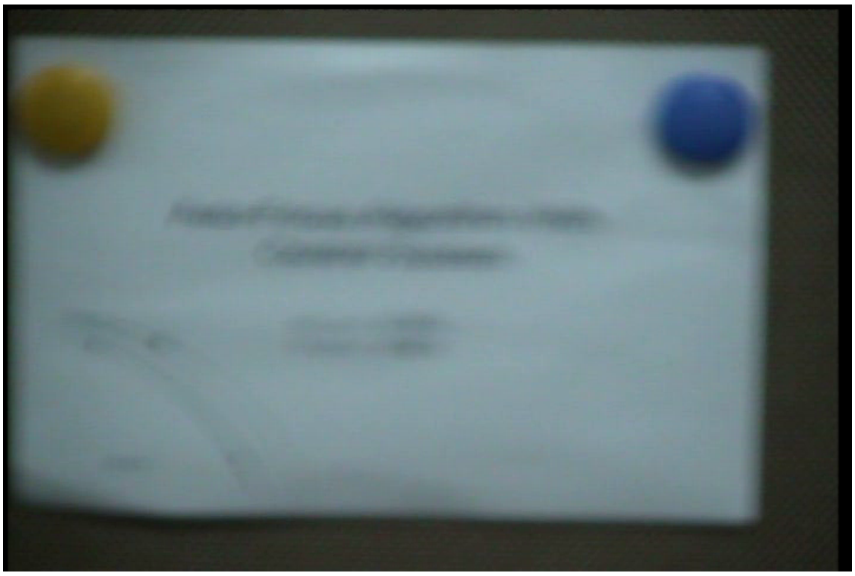
5.1 Simulation for Still Image

The still image is the basic and common auto-focusing case. It is merited that our auto-focusing system can handle this problem. Figures 5.1, 5.2, and 5.3 show our auto-focusing algorithm is capable of focusing the still image under different situation. The initial point is set at the most far focus position that makes the picture very blur. We can see that the initial pictures (a) are very blurred and the focused results (b) are quite clear. The time it costs is average 16 cycles.

(a) The initial picture.

(b) The focused picture after 13 cycles.

Figure 5.1 The test image “scarecrow.”



(a) The initial picture

(b) The focused picture after 17 cycles.

Figure 5.2 The test image “paper.”

(a) The initial picture

(b) The focused picture after 17 cycles.

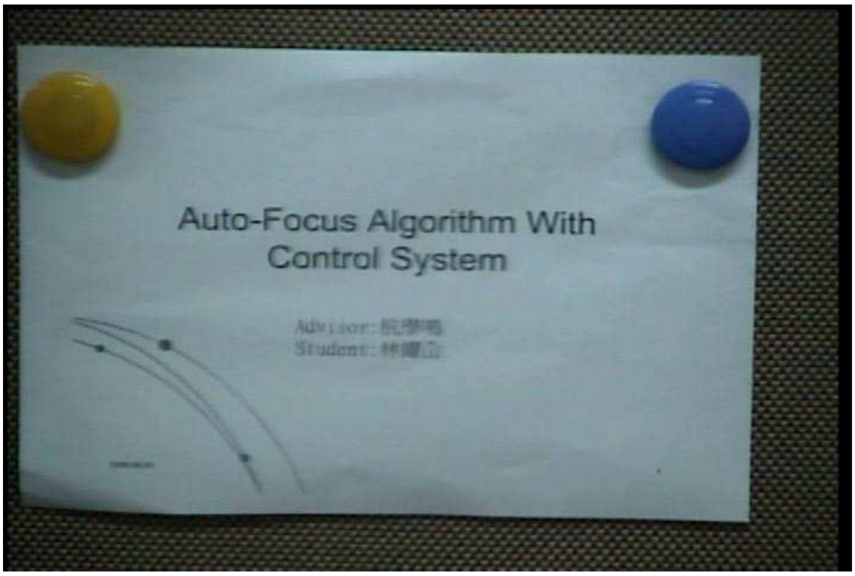
Figure 5.3 The test image “fan.”

5.2 Simulation for Scene Change:

The scene change is a dynamic auto-focusing case. Figure 5.4 shows a simulation procedure. We simulate the scene change by switching the reading database. It takes two kinds of image data to simulate the scene change case.

(a) The initial picture.

(b) The focused picture.



(c) The new picture with the same focus position.

(d) The new focused picture after 7 cycles.

Figure 5.4 The four test frames for a scene change simulation

The first two pictures (a) and (b) are the similar segment as the still image simulation. When the scene change happens, the new picture (c) seems a little blur. Our auto-focusing algorithm detects this change and corrects the picture to a focused one. The final picture (d) is still clear after the scene change occurs. It costs fewer cycles to focus the changed picture because the default initial focus position is the worst case.

5.3 Simulation for Local Object Motion

The object motion is a critical dynamic auto-focusing problem. Figure 5.5 shows a simulation procedure. We capture every motion separately and build them up into a motion database. We simulate the local object motion by switching the motion database with a regular routine.

(a) The initial picture.

(b) The picture at cycle 5.

(c) The picture at cycle 10.

(d) The focused picture at cycle 17

Figure 5.5 The four test frames for a local object motion simulation.

We can see the pictures shown in Figure 5.5 become more and more clear even the fan is rotating. Our algorithm does not take extra time for focusing a motion object but costs nearly the same time as focusing the still one. The reason is that the search algorithm handles the motion problem and the still image the same. Our threshold search algorithm is reliable and efficiency for a local object motion case.

5.4 Simulation for Zoom Tracking

The zoom tracking is a prediction method. Figure 5.6 shows a zoom tracking procedures. The simulation for zoom tracking case is very similar to the scene change case but it adds a prediction step when the zoom change occurs.

(a) The initial picture.

(b) The focused picture at cycle 22.

(c) The zoom-in picture with zoom tracking at cycle 25.

(d) The focused zoom-in picture at cycle 36.

Figure 5.6 The four test frames for zoom tracking.

Our algorithm predicts the new focus position simultaneously when the zoom position changes. Although it is not very accurate, the picture shown in figure 5.6 (c) is quite clear. The zoom tracking method spends less time on focusing the changed picture than the scene change approach. It also has better performance at the auto-focusing process. Figure 5.7 shows the comparison between the zoom tracking and no zoom tracking. The zoom tracking keeps the picture much clearer than without it.

Figure 5.7 The comparison of the zoom changed picture between no zoom tracking and zoom tracking.

Chapter 6

Conclusions

6.1 Conclusions

In this thesis, we design a dynamic auto-focusing algorithm for digital camera systems. Our algorithm is an extension of the existing auto-focusing algorithm for still image cameras. The mainly modify the search algorithm to match the dynamic requirements. Instead of calculating the focus value metric, the search algorithm plays a more important role in dynamic auto-focusing. A good focus value metric makes search algorithm easy but the research on focus value is quite extensive. Hence, our focus is to construct a robust search algorithm for a dynamic auto-focusing system.

We choose the Sobel operator as our focus value. The Sobel operator is well known and less sensitive to the noise. Although the Sobel operator is not a very robust focus value metrics, it is acceptable in our auto-focusing system. We also test a few window selection methods to enhance the focus value.

The search algorithm is the most important part of our dynamic auto-focusing algorithm. We implement our search algorithm using a finite state machine, which is flexible to insert new function states or update the current ones. We first propose a luminance-based measurement for detecting scene changes. It evaluates the mean value of the picture luminance and is quite reliable. Next, we consider the local object motion problem, where the pictures change the contents somewhat but not the focus position. We propose new metrics and thresholds to solve this local motion problem. The results are reasonably successful. Yet the third problem is the zoom tracking. We design the zoom tracking table that predicts the focus position with the zoom position. We also consider the delay effect in the system. Finally, we construct an overall system that integrates all the above into it. We take some experimental results to verify our algorithm. Our dynamic auto-focusing algorithm can handle these problems pretty well.

6.2 Future Work

There are still many research topics for future work.

Real-time implementation

Our experiment is only an offline simulation because of the simulation environment limit. A real-time system can easily verify the algorithm directly and helps adjusting the algorithms.

Robust focus value metrics

In chapter 3, we introduce a few metrics and this topic is quite popular in the current auto-focusing research literature. Although we do not put emphasis on this part, it still plays an important role in the auto-focusing system.

A complete 3A system

Auto-focus is a part of the 3A algorithm. Both Auto white balance and auto exposure have a certain amount of impact on auto-focusing. Therefore, it is desirable to design a complete 3A system for digital cameras.

References

[1] J. -S. Lee, Y. –Y. Jung, B. –S. Kim, and S. –J. Ko, "An advanced video camera system with robust AF, AE, and AWB control," *IEEE Transactions on Consumer Electronics*, Vol. 47, No. 3, pp.694-699, Aug. 2001.

[2] Y. Lee, S. –L Jang, K. Chung, D. Lee, W. Kim, and C. –W. Lee, "A fuzzy-control processor for automatic focusing," *IEEE Transactions on Consumer Electronics*, Vol. 40, No. 2, pp.138-144, May. 1994.

[3] http://www.wikipedia.com

[4] F. Li and H. Jin, "A fast auto-focusing method for digital still camera," *Proceedings of the Fourth International Conference on Machine Learning and Cybernetics*, Guangzhou, pp.5001-5005, Aug. 2005.

[5] D. –E. Seborg, T. –F. Edgar, and D. –A. Mellichamp, *Process Dynamics and Control*, second edition, Prentice Hall, 2002.

[6] C. -Y. Chen, C. -Y. Tseng, C. -H. Hung, I. -C. Yin, and S. -J. Wang, "Combinational AE-AF system with fuzzy climbing search servo," *Proceedings of SPIE-IS&T Electronic Imaging* Vol. 6069, pp.60690B1-7, 2004.

[7] T. Okajima and K. Sugimoto, "Automatic focusing device of image pickup device and method of correcting zoom tracking curve," U.S. Patent No. 6,624,851, Sanyo Electric Company, 2003.

[8] P. Hoad and J. Illingworth, "Automatic control of camera pan, zoom and focus for improving object recognition," *Fifth IEE International Conference on Image Processing and its Applications*, No. 410, pp. 291-295, July. 1995.

[9] J. –S. Lee, S. –J. Ko, Y. Kim, and A. Morales, "A video camera system with adaptive zoom tracking," *International Conference on Consumer Electronics*, pp. 56-57, June

2002.

[10] V. Peddogari and N. Kehtarnavaz, “A relational approach to zoom tracking for digital still cameras”, *IEEE Trans on Consumer Electronics*, Vol. 51, No. 4, Nov 2005.

[11] I. –S. Suh, “Apparatus and method for controlling a focus position for a digital still camera,” U.S. Patent No. 6,977,687, Samsung Techwin Co., 2005.

[12] W. Na, “Apparatus and method for controlling focus using adaptive filter,” U.S. Patent No. 6,545,715, Samsung Electronics Co., 2003.

[13] S. –E. Lee, J. –S. Park, and M. –K. Yeo, “Camera apparatus for detecting focus value engaged to output picture,” U.S. Patent No. 6,239,838, LG Electronics Inc., 2001.

[14] C. –M. Chen, C. –M. Hong, and H. –C. Chuang, ”Efficient auto-focus algorithm utilizing discrete difference equation prediction model for digital still cameras,” *IEEE Transactions on Consumer Electronics*, Vol. 52, No. 4, pp.1135-1143, Nov. 2006.

[15] J. –T. Huang, C. –H. Shen, S. –M. Phoong, and H. Chen, “Robust measure of image focus in the wavelet domain,” *Proceedings of 2005 International Symposium on Intelligent Signal Processing and Communication Systems*, pp.157-160, Dec. 2005.

[16] C. –H. Shen and H. Chen, “Robust focus measure for low-contrast images,” *International Conference on Consumer Electronics, 2006. ICCE '06. 2006 Digest of Technical Papers*, pp.69-70, Jan. 2006.

[17] R. –C. Gonzalez and R. –E. Woods, *Digital Imaging Processing*, second edition, Prentice Hall, 2001.

[18] http://msdn2.microsoft.com/default.aspx

[19] J. He, R. Zhou, and Z. Hong, “Modified fast climbing search auto-focus algorithm with adaptive step size searching technique for digital camera,” *IEEE Transactions on Consumer Electronics*, Vol. 49, No. 2, pp.257-262, May 2003.

[20] K. –Ooi etc. “An advanced autofocus system for video camera using quasi condition reasoning,” *IEEE Trans. an Consumer Electronics*, Vo1.36, No.3, pp526-530, Aug. 1990

作者簡歷

林耀合，男，民國七十三年八月二十四日出生於台北市。民國九十五年二月畢業於國立交通大學電子工程學系。同年進入國立交通大學電子研究所碩士班就讀。從事影像處理相關的研究。