

Auto-focusing System for Microscope Based on Computational Verb Controllers

Weibin Tang¹ Yinghao Liao*² Zhicong Chen³

^{1, 2, 3}Dept. of Electronic Engineering
Xiamen University
Xiamen, Fujian, China
tang102201@hotmail.com¹

Lihuan Cai⁴ Tao Yang⁵ Donghui Guo⁶

^{4, 5, 6}Dept. of Electronic Engineering
Xiamen University
Xiamen, Fujian, China
taoyang@xmu.edu.cn⁵

Abstract—In this paper, an auto-focusing system of microscope for integrated circuits (IC) analysis is presented. In this system, the Laplacian algorithm is used as the evaluation function, which provides a reference to the degree of defocus. The auto-focusing controlling algorithm based on computational verb theory consists of two controllers designed: The moving-speed controller and the moving-direction controller. Both controllers work well under the verb-control rules designed in this paper. It has shown that the system can focus accurately and quickly, and it can adjust itself when it is out of focus.

Keywords- Computational Verb; Controlling Algorithm; Evaluation Function;

I. INTRODUCTION

The technology of microscope auto-focusing plays an important role in the fields such as biology, medicine, IC design and etc. because it can provide high focusing precision and work with high efficiency. It consists of two stages, of which the first stage is the image definition analysis, and the second one is the motion control. There are usually three parts in such a system: the control center, e.g., a computer, the motion control part, such as a step motor, and the image capturing component such as a CCD camera.

Computation verb theory [1] is developed by Tao Yang in the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley in 1997. In computational verb theory, all dynamical experiences of human experts can be coded in mathematical models known as computational verb rules. Since computational verb rules can be in their natural language forms, they are extremely efficient to transform knowledge of know-how into mathematical models for engineering applications. In the last decade, computational verb theory had been applied to many engineering and commercial applications, such as: computational verb image processing, computational verb controllers, card counter, accurate CCTV flame detector, CCTV monitoring system for ITS, and many others [2] [3] [7]. A good control algorithm, which takes advantages of the knowledge from human experts, can reduce the time for auto-focusing. Our algorithm is based on computational verb controllers designed by using the accumulated experiences of tuning the microscopes manually. We design several verb-

control rules in our system to control the motion of the step motor. Experimental results have shown that our system works well under the control of these computational verb rules. Our computational verb auto-focusing algorithm can focus the microscope accurately and dramatically reduce the focusing time comparing with previous ones developed in the same laboratory. It can meet the requirements of precision and speed for auto-focusing.

The organization of this paper is as follows. In Section II, the configuration of the entire system will be presented. In Section III, the evaluation function will be constructed. In Section IV, the computational verb controlling algorithm will be designed. In Section V, conclusions will be included.

II. THE CONFIGURATION OF THE SYSTEM

Our system is made up of a computer, a CCD camera, a MCU Controller, a microscope and a step motor. The block diagram of our system is shown in Fig. 1.

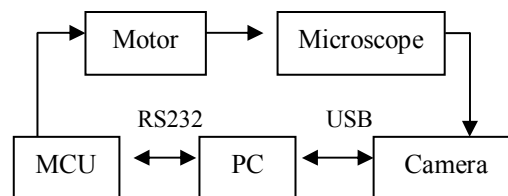


Fig.1 The block diagram of the auto-focusing system.

The auto-focusing process can be as follows:

- 1) The computer captures an image from the camera via a USB port.
- 2) The Laplacian algorithm is used to compute the definition of the captured image.
- 3) Then the value of definition at current step is compared with that of previous step. If the system is in focus, the value of definition will be at its maximal where the direction controller and the speed controller stop working. Otherwise, the computer will control the MCU controller via RS232 communication port in order to change the moving direction or speed of the

step motor until the camera on the microscope is in focus.

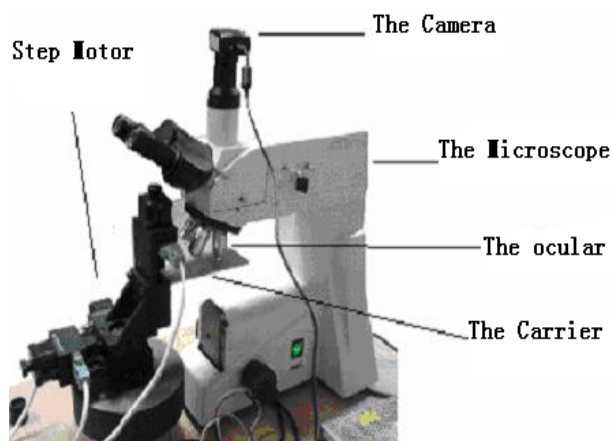


Fig.2 The photograph of the components in our system.

Theses controlling processes were implemented as computational verb-control rules.

III. THE EVALUATION FUNCTION

The evaluation function plays critical role in improving the control performance of auto-focusing. It has an immediate effect on the accuracy of the system. There are many evaluation functions for auto-focusing. But there are only a few are of good performances and in common uses. For example, Laplacian algorithm [4], Roberts Gradient algorithm [5], Sobel algorithm [6] and others. These algorithms share the same characteristic: When the camera is out of focus, the values of these functions will change slowly, but when it moves close to the focus point, they will increase sharply, as shown is Fig.3. The area within the dotted box indicates the region closed to the focus point, and at the peak the exact focusing point is located.

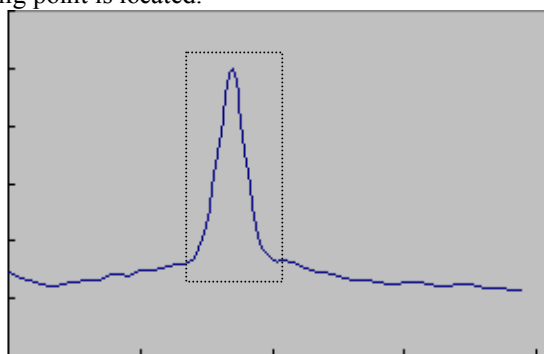


Fig.3 the curve of the evaluation function.

In this paper, the Laplacian algorithm is used as the evaluation function, because of its high accuracy and high speed, which have been discussed in details in Reference [4]. Here, we define $F(k)$ as the evaluation function, k is the control iteration, which ranges from 1 to N , $f(x,y)$ denotes the gray

value of the image, X and Y are the width and of the image, respectively. V is value of $F(k)$ at the focus point. Then we have the following equations:

$$F(k) = \sum_X \sum_Y \nabla^2 f(x,y)$$

$$= \sum_X \sum_Y \left\{ \left| \frac{\partial^2 f(x,y)}{\partial x^2} \right| + \left| \frac{\partial^2 f(x,y)}{\partial y^2} \right| \right\} \quad (1)$$

$$= \sum_X \sum_Y \left\{ |2f(x,y) - f(x-1,y) - f(x+1,y)| \right.$$

$$\left. + |2f(x,y) - f(x,y-1) - f(x,y+1)| \right\}$$

$$V = \max \{F(k)\}, k = 1, 2, 3, \dots, N \quad (2)$$

Experiments have shown that the Laplacian algorithm take a few iterations to find the focus point and it has a good sensitivity near the focus point under high SNR.

IV. THE CONTROLLING ALGORITHM

If a controlling algorithm is good enough, the auto-focusing quality and efficiency will be improved dramatically. The algorithm will be presented in this section meets the demands of quality and efficiency.

A. The motion controllers

In this paper, we assume that:

- μ stands for the action of the step motor.
- $F(k)$ stands for the definition of the current image.
- μ changes according to the value of $F(k)$. That is

$$\mu = f[F(k)] \quad (3)$$

If we assume F_i to be the input and μ_i to be the output, our algorithm to function as a filter, then the data stream of our system can be depicted as in Fig.4.

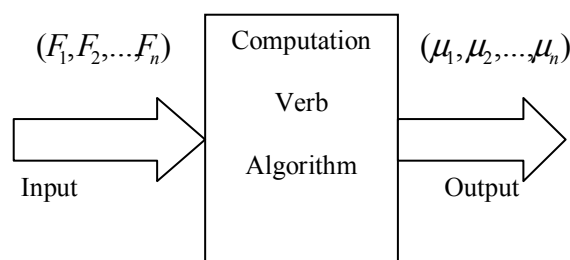


Fig.4 The structure of data stream.

As shown in Fig.3, the control goal is to drive the step motor to the peak value of the curve. However, the real curve won't be as smooth as shown in Fig.3, because of noises.

In order to describe the actions of the step motor clearly, we categorize these actions into two classes: The control steps and the moving directions. We design one controller for each of them: the speed controller and the direction controller.

According to the curve shown in Fig.3, we design several verb control rules for the speed controller. As we have referred above, $|\mu|$ stands for the speed, the verb control rules are listed as follows:

- 1) IF $F(k)$ changes too much, THEN $|\mu|$ decreases.
- 2) IF $F(k)$ changes slowly, THEN $|\mu|$ increases.
- 3) IF $F(k)$ is close to the maximum value, THEN $|\mu|$ decreases to a constant level.

Among them, the rule 3 is superior to the other two. It is designed to eliminate the influence of the noise near the peak point.

On the other hand, we design the following verb control rules for directions:

- 4) IF $F(k)$ increases, THEN the sign of μ keeps the same as μ
- 5) IF $F(k)$ decreases, THEN the sign of μ will be changed.

As we know, verbs won't be recognized by computer. In order to make them computable, we should model them as computational verbs. This is the most important step in our system design.

B. The modeling of computational verbs

Let us define the following variable:

$$\Delta F(k) = F(k) - F(k-1) \quad (4)$$

and choose the life span of each verbs in verb rules (1) to (5) as the current step and the previous step, then the rules in above can be converted into the following rules:

- 1) IF $|\Delta F(k)| > X_0$, THEN $|\mu|_{k+1} = |\mu|_k - \Delta\mu_0$.
- 2) IF $|\Delta F(k)| < X_1$, THEN $|\mu|_{k+1} = |\mu|_k + \Delta\mu_1$.
- 3) IF $F(k) > F_{near_max}$ THEN $|\mu|_{k+1} = \mu_{const}$.
- 4) IF $\Delta F(k) > 0$, THEN $Sign(\mu_{k+1}) = Sign(\mu_k)$
- 5) IF $\Delta F(k) < 0$, THEN $Sign(\mu_{k+1}) = -Sign(\mu_k)$

The values of X_0 , X_1 , $\Delta\mu_0$, $\Delta\mu_1$, μ_{const} , and

F_{near_max} depend on the real system involved. Observe that in this computational verb algorithm, the simplest evolving functions are chosen for all computational verbs in the antecedents and consequences. Also, the winner-take-all strategy is used to fire each rules. Therefore, it is not necessary to build a comprehensive computational algorithm by using computational verb similarities as presented in Reference [7,8]. Since the implementations of the five computational verb rules are the simplest, more advanced implementation of verb rules can be constructed in the similar way.

C. Experiments and Results

In our system, we choose $X_0 = 10000$, $X_1 = 2000$,

$$\Delta\mu_0 = \Delta\mu_1 = 16, \mu_{const} = 2, F_{near_max} = 120000 \quad . \quad \text{In}$$

order to validate the computational verb rules, we carry out some experiments with a chip layout. Here we choose two typical cases.

Case one is 142 steps off the focus point. It can be seen from Fig.5 that it only takes 16 actions to reach the focus point, with the precision of 2 steps. Otherwise, if you move step by step with the some precision without the computational verb rules, it will take $142/2=71$ actions to finish the same work, or we can decrease the precision in order to reduce the iteration steps needed for focusing. Figure 7(a) is a defocusing picture.

Case two is 222 steps off the focus point as shown in Fig.6. It only takes 28 actions to achieve focus. The same as above, if one moves step by step, it will take 111 actions to match the same precision, or one can decrease the precision to reduce the focusing time. Figure 7(c) and (d) are both taken in this case. Figure 7(c) is out of focus and Fig.7 (d) is in focus.

However, it is not advisable to spend much time in focusing in order to gain a high precision or to focus quickly by decreasing the precision. Fortunately, we have computational verb rules. It not only gains a high precision, but also focuses quickly. It is the best choice.

On the other hand, we can see from Fig.5 and Fig.6,

- When $F(k)$ changes dramatically, it slows down.
- When $F(k)$ changes smoothly, it speeds up.
- When $F(k)$ is very large, it keeps constant.
- When $F(k)$ increases, it keeps its direction.
- When $F(k)$ decreases, it changes its direction.

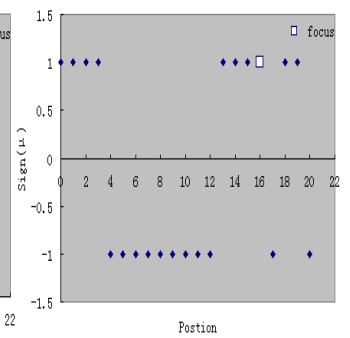
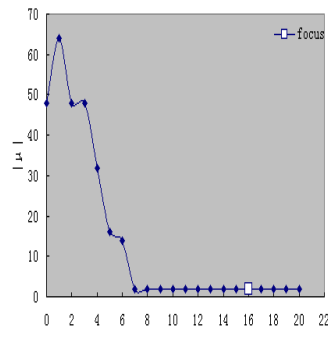
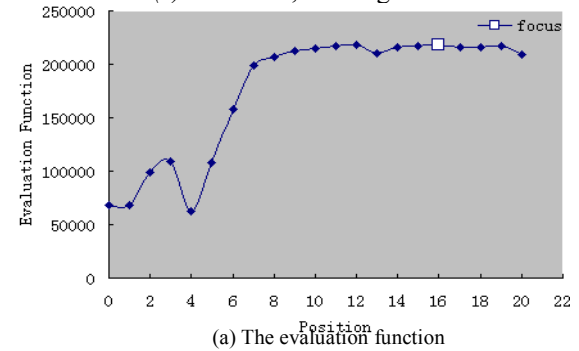


Fig.5 Case One

V. CONCLUSIONS

In this paper, we present the design of a real time auto-focusing system based on computational verb rules. These computational verb rules improve the performance dramatically both in precision and efficiency. We applied the experiences of human operators coded in natural language to control the system presented in this paper. In such applications, how to build rules and how to model the rules are the most important things. Good rules and good models will improve the system design dramatically. It has been shown that the computational verb rules working in our system were built properly. Compared with the traditional controlling algorithm, the computational verb rules make our system focused quickly and accurately. On the other hand, the computational verb control rules can efficiently suppress the effect of noises.

REFERENCES

- [1] Tao Yang. *Computational Verb Theory: From Engineering, Dynamic Systems to Physical Linguistics*, volume 2 of YangSky.com Monographs in Information Sciences. Tucson, AZ: Yang's Scientific Research Institute, Oct. 2002.
- [2] Tao Yang. "Applications of computational verbs to the design of P-controllers". *International Journal of Computational Cognition*, vol.3, no. 2, pp. 52-60, Jun 2005.
- [3] Sheng Zhu, Zhongjie Wang, Yong Liu, Baoliang Xia. "An Improvement of the Design of Computational Verb PID-Controllers". *System Simulation Technology*, vol.2, no.1, pp. 25-30, Jan 2006.
- [4] Yi Fang, Chongxun Zheng, Xiangguo Yan. "Research on an Auto-focusing Algorithm for Microscope". *Chinese Journal of Scientific Instrument*, vol.26, no.12, p.1276, Dec 2005.
- [5] Zan Gao, Wei Jiang, Kongfeng Zhu, Chao Wang. "Auto-focusing algorithm based on Roberts gradient". *Infrared and Laser Engineering*, vol.35, no.1, pp.117-121, Feb, 2006
- [6] Xian Zhou, Wei Jiang. "Auto-focus algorithm based on image edge energy". *Optical Technique*, vol. 32, no. 2, pp.213-215, Mar 2006.
- [7] Tao Yang, *The Mathematical Principles of Natural Languages: The First Course in Physical Linguistics*, volume 6 of YangSky.com Monographs in Information Sciences. Tucson, AZ: Yang's Scientific Press, Dec. 2007.
- [8] Tao Yang. "Computational verb theory: Ten years later". *International Journal of Computational Cognition*, vol.5, no.3, September 2007, pages 63-86.

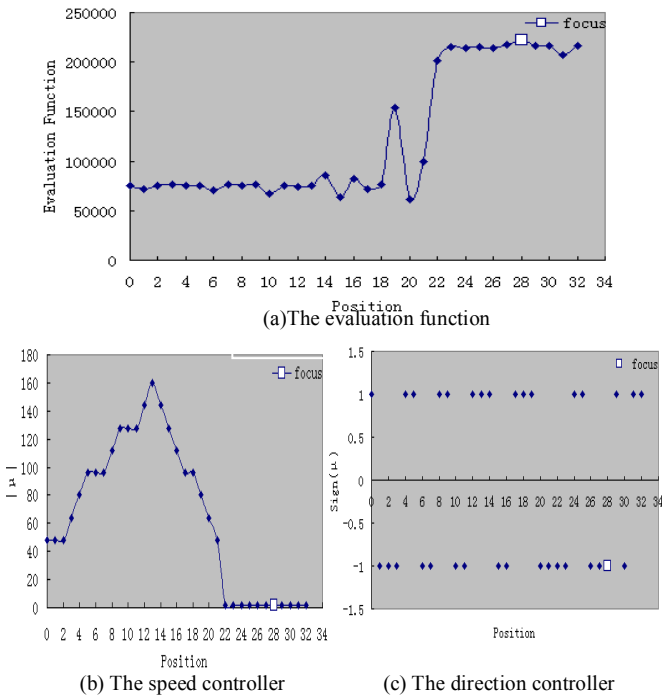


Fig.6 Case Two

The control process followed accurately the computational verb rules designed in our system. By computing the value of evaluation function, it can check whether the microscope is in focus or not. When it is out of focus, the system will adjust itself via these computational verb rules and keep it in the focus all the time.

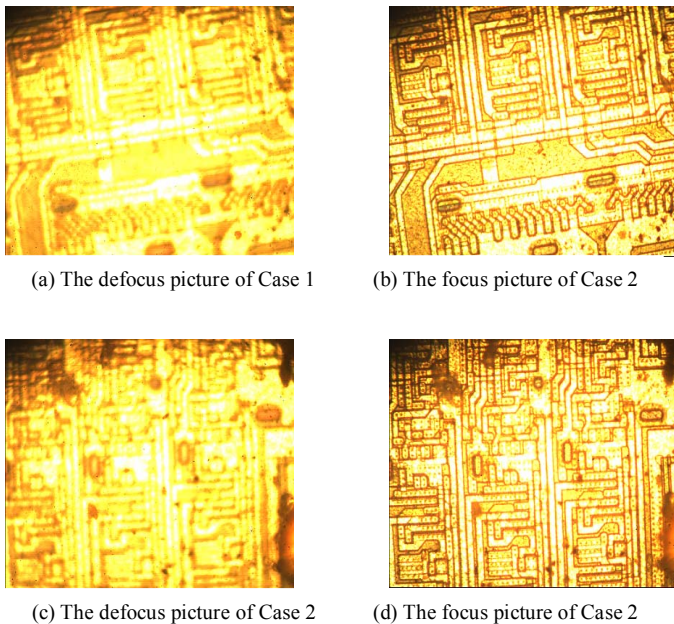


Fig.7 Images of the two cases studied in this paper.