Upper Computer design for active optics

Chen Li [a,b,c], Guomin Wang[a,b], Liang Gao[a,b,c]
[a]National Astronomical Observatories / Nanjing Institute of Astronomical Optics & Technology, Chinese Academy of Sciences, Nanjing 210042, China;
[b]Key Laboratory of Astronomical Optics & Technology, Nanjing Institute of Astronomical Optics & Technology, Chinese Academy of Sciences, Nanjing 210042, China;
[c]Graduate University of Chinese Academy of Sciences, Beijing 100049, China

ABSTRACT

China has joined the international global network SONG project and will build one 1-meter telescope as one node of SONG network. This paper shows the upper computer software system design under Linux operating system for active optics control system of Chinese SONG telescope. The upper computer software developed in this paper under Linux OS has three functions: detection of S-H wavefront, calculation of mirror correction force and communication with the controller of hardware. We will introduce the three modules developed under Linux environment: wavefront image processing module, communication module and GUI module.

Keywords: telescope, upper computer, Linux OS

1. INTRODUCTION

SONG (Stellar Oscillations Network Group) is a Danish initiative to design and build a global network of 1-m class telescopes. The scientific goals of SONG are to study the internal structure and evolution of stars and to hunt exoplant.

As to improve the surface precision of mirror in a large aperture telescope at different altitude and angles, the active optics technology is proposed in SONG project.

This paper focus on the upper computer design of active optics control system of SONG telescope and the main task of upper computer is to finish three function : image processing, communication with controller and GUI.

2. S-H wavefront detection

Shack-Hartmann wavefront sensor is an optical instrument used to characterize an imaging system. It is a wavefront sensor commonly used in active optics system. It consists of an array of lenses of the same focal length. Each is focused onto a CCD array, the local tilt of the wavafront across each lens can then be calculated from the position of the focal spot on the sensor[1]. Any phase aberration can be approximated to a set of discrete tilts, by sampling an array of lenslets all of these tilts can be measured and the whole wavefront approximated. The rate of deviation between each focus and center point in the x and y direction reflects the average tilt of the unit wavefront[2].

3. Function model of software design

SONG active optics electronic system consists mainly of upper computer, S-H wavefront sensor, controller, driven element and feedback element, which is shown as Figure 1.

In our project, we select Linux OS as upper computer OS, whose distribution version is Debian 6.0, and software development is qt 4 using as qt creator as its IDE. This paper focus on the upper computer software design and development of the whole control system under Linux OS, which is the important part of the whole system, and user eventually realize the mirror surface control with it, so as to accomplish the whole correction process. The software system main points three modules : image processing module, communication module and Human-Computer interface module. With this three modules working together we can reach our goal of observing the whole control system[3].

Software and Cyberinfrastructure for Astronomy II, edited by Nicole M. Radziwill, Gianluca Chiozzi,
Proc. of SPIE Vol. 8451, 845136 · © 2012 SPIE · CCC code: 0277-786X/12/$18 · doi: 10.1117/12.926070

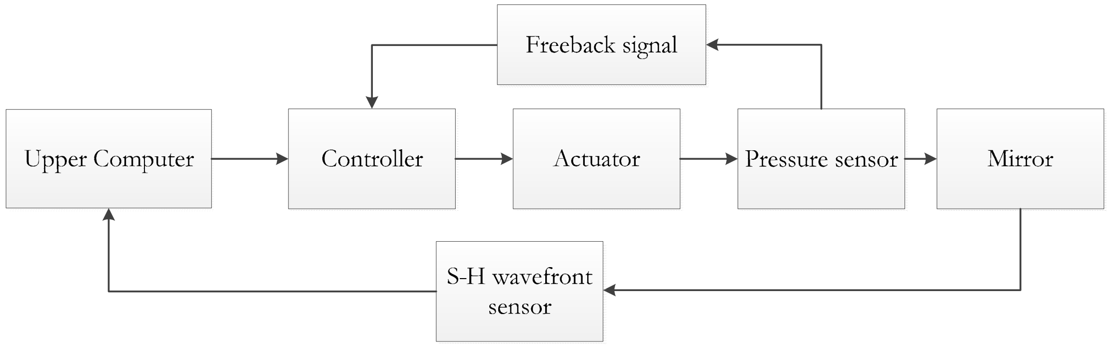

Figure 1: Active optics control system schematic diagram

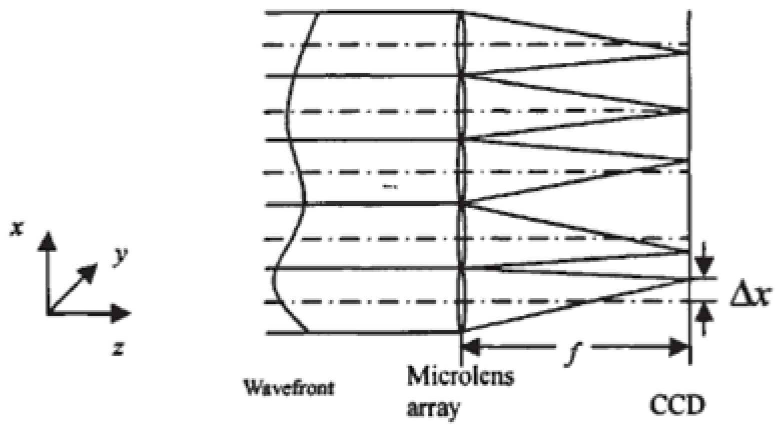

Figure 2 : Shack-Hartmann wavefront sensor schematic diagram

3.1 Image processing module

Our goal is controling the surface precision to a certain degree, so first the surface detection is needed. Our team uses the Shack-Hartmann wavefront sensor to do this, whose principle is shown in Figure 2.

S-H wavefront sensor is mainly consists of Mircolens arrays and CCD camera, which consists of an array of lenses of the same focal length. Each is focused onto a CCD array. We can get a sub aperture flare arrays in n × n dimensions which is shown as Figure 3. When the surface of the mirror changes, the centroid of each flare also generate center offset in x and y direction, we can get the wavefront reconstruction via a set of algorithm, and the formula are[4]:

$$\Delta x = f\frac{\partial w}{\partial x}, \Delta y = f\frac{\partial w}{\partial y}$$

$$W(r,\theta) = a_0 Z_0(r,\theta) + a_1 Z_1(r,\theta) + a_2 Z_2(r,\theta) + \cdots = \sum_{i=0}^{N} a_i Z_i(r,\theta)$$

Where, the f is focal length of the micro lens, $\frac{\partial w}{\partial x}, \frac{\partial w}{\partial y}$ are the wavefront slope of the aperture wavefront fitting in Zernike polynomial. To offset the $W(r,\theta)$, we need the force to generate W in the opposite direction. They meet the equation as below:

$$CF = -W$$

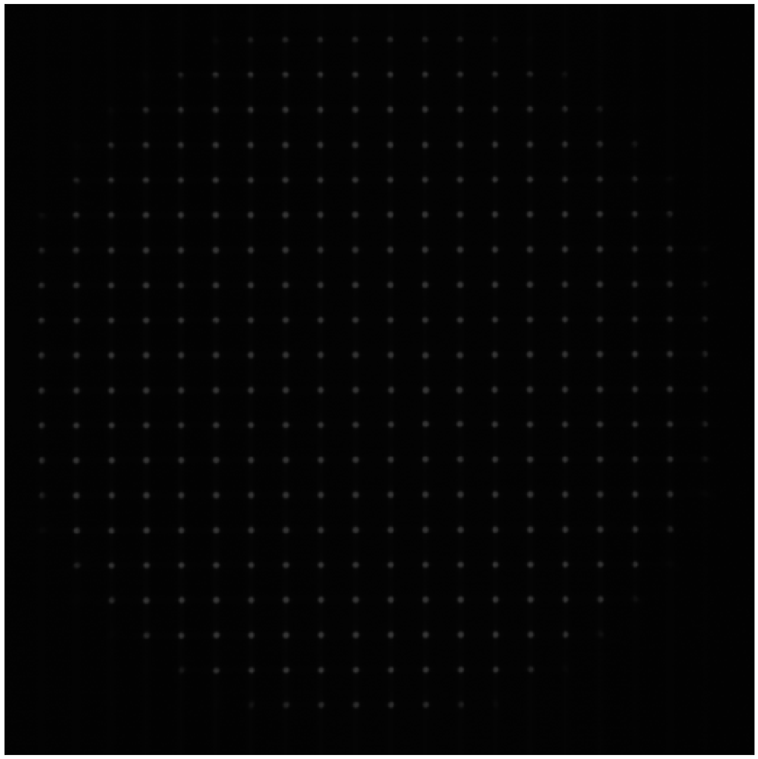

Figure 3 : CCD webcam image

We express it in another form:

$$\begin{bmatrix} \frac{\partial a_1}{\partial f_1} & \frac{\partial a_1}{\partial f_2} & \cdots & \frac{\partial a_1}{\partial f_n} \\ \frac{\partial a_2}{\partial f_1} & \frac{\partial a_2}{\partial f_2} & \cdots & \frac{\partial a_2}{\partial f_n} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial a_m}{\partial f_1} & \frac{\partial a_m}{\partial f_2} & \cdots & \frac{\partial a_m}{\partial f_n} \end{bmatrix} \times \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix}$$

Where, n is the numbers of the actuators, $a_i(i=1,2,\cdots,m)$ is the i item of Zernike polynomial's coefficient[5-7]. We can get $F=-(C^TC)^{-1}C^TW$ and the F is just we want to change the mirror surface. we will give the code realization. The steps are:

1. Image acquisition and the preservation using the SDK from the company who sells us the CCD camera.
2. Confirmation of position of centroid of flare using the Matlab build-in command graythresh, im2bw and the regionprops due to its very high efficiency.(Figure 4)
3. Zernike polynomial fitting (Figure 5) and calculating the correction force using the mothod of least square method.
4. Using the interface between the C++ and matlab to calls matlab build-in command in c++ environment.

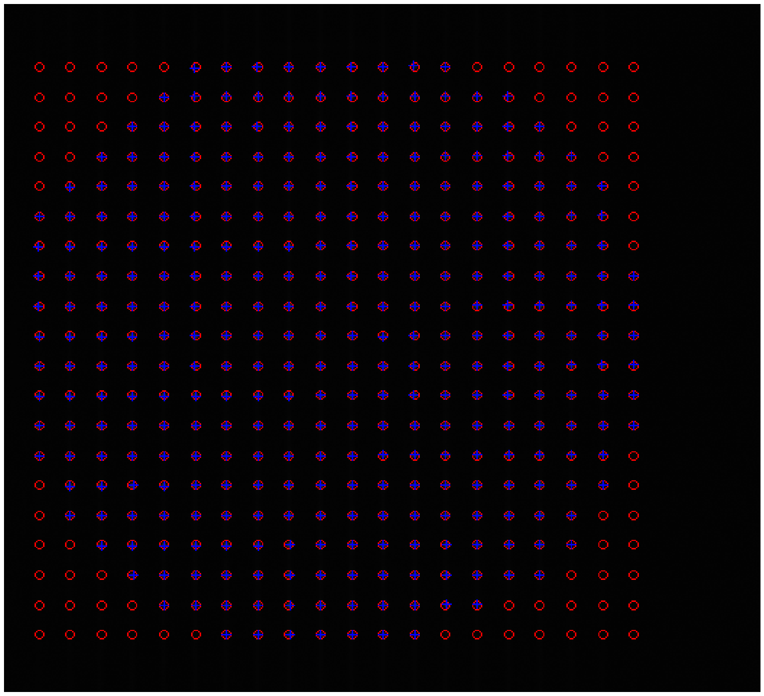

Figure 4 : the centroid detection

The realization of code is easy using Matlab build-in command, the key is how to using c++ to call matlab command. And the surprise is matlab supply the interface between the c++ and matlab. We use the command like this: “Mcc –W cpplib:liblens –T link:lib lens.m” to generate c++ shared libraries, so we can realize the communication between c++ and matlab. What we need to pay attention to is the function interface’s data type, which is mxArray. Every mwArray class contains a pointer to a MATLAB array structure. For this reason, we only use the mxArray date type to finish the interface between c++ and MATLAB.

3.2 Communication module

In order to realize the high speed and real-time communication between upper computer and controller, network communication is proposed in our project based on TCP/IP protocol, we use Linux socket to realize the communication using the C/S model.

Server code:

1. Create socket
2. Bind the socket to a local address and port
3. Set the socket listen model, ready for receiving the request from client
4. Return to the socket when the request is coming
5. Using the socket returned to communicate
6. Wait for the next request
7. Close socket

Client code:

1. Create socket
2. Request the server
3. Communicate with the server
4. Close socket

When the upper computer send the data, upper computer is the client and controller is the server. When the upper computer receive the data, upper computer is the server and the controller is the client. We create pairs of socket to realize the communication with any part send or receive data.

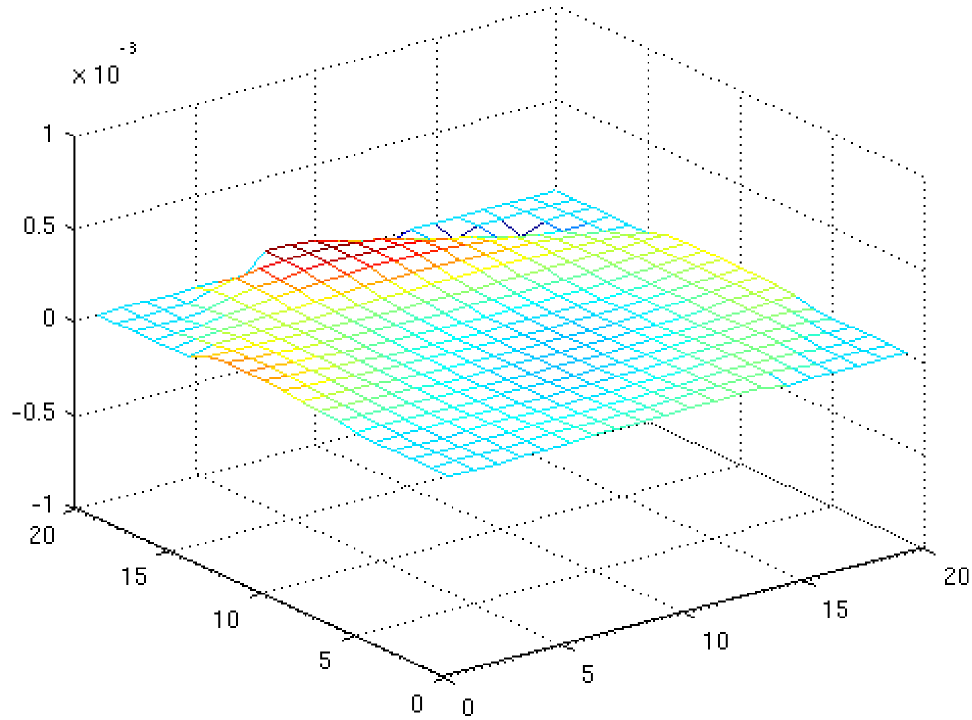

Figure 5 : Zernike polynomial fitting

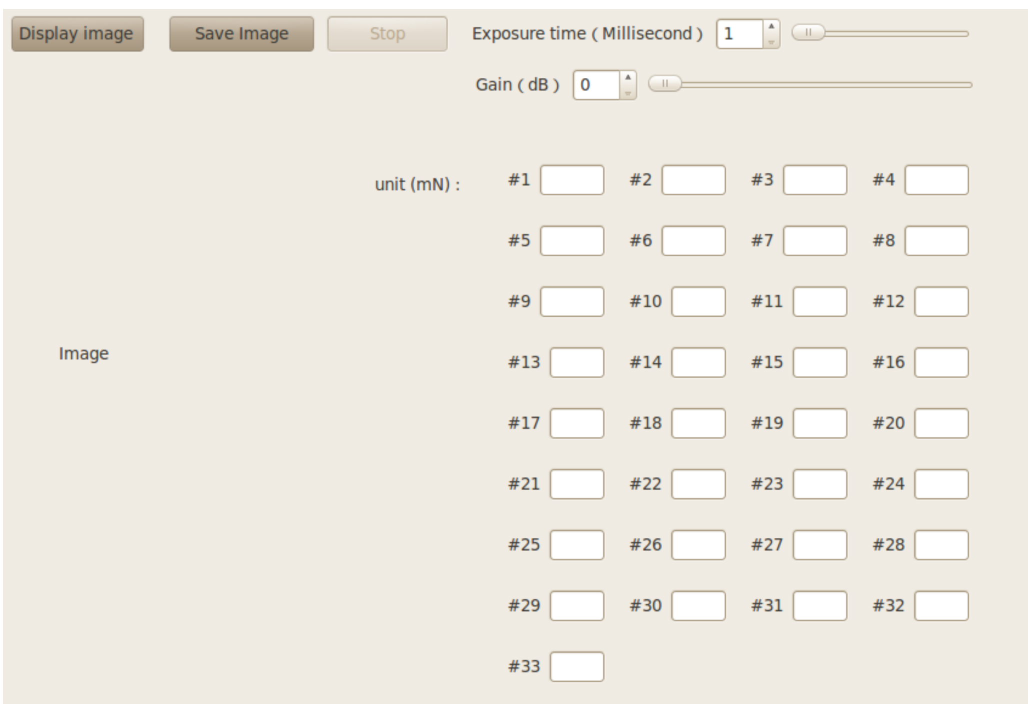

Figure 6: programmer interface

3.3 GUI module

Qt contains lots of graphics library, which is beneficial to our graphics programming, as shown in Figure 6, The left of UI is the region of CCD webcam image, and the right of UI is the real-time feedback of actuator.

4. CONCLUSION

This paper mainly discusses the main structure and function of upper computer, mainly including the image processing, module, communication module and the GUI module, and the key technology about the interface between C++ and MATLAB under Linux environment is mentioned also.

REFERENCES

[1] Noethe, L., Andrenoi, G., Granza, F., "Latest development of active optics of the ESO NTT and the implication of the ESO VLT" Proc. SPIE,1542, 293-296(1991).

[2] KNOHL E-D, "VLT primary support system," Proc. SPIE, 2199, 271-283(1994).

[3] Noeth, L., "Active optics in modern large optical telescopes," Progress in Optics, 43,1-69(2002).

[4]A.Preumont, R.Bastaits, G.Rodrigues. "Scale effects in active optics of large segmented mirrors," Mechatronic,19(8),1286-1293(2009).

[5]Noethe, L., Franza, F., Giordano, P,. "Active optics 2: results of an experiment with a thin 1m test mirror," Journal of Modern Optics, 35(9),1427-1457(1988).

[6] Noethe, L., Franza F., Wilson R., N., "Active optics: from the test set up to the NNT in the observatory," Proc. SPIE , 1114, 314-319(1989).

[7] SU Ding-qiang, JIANG Sheng-tao, ZOU Wei-yao. "Experiment system of the thin-mirror active optics," Proc. SPIE, 2199, 609-621(1994).