

The Design of An Optical Lens Simulation by OpenGL

Lei Sheng^{1*} Shanjun Zhang¹ Hongbing Zhu²

¹ Dept. information science, faculty of Science, Kanagawa University

² Faculty of information design, Hiroshima Kokusai Gakuin University

* Corresponding author's Email: lsheng_hustoe@hotmail.co.jp

Abstract: In this paper, a simulation method for lens design is presented considering the aberration effect. Traditionally, most optical path calculation algorithms are based on aberration theory and result in approximation. For eliminating this problem and avoiding the limitations from aberration theory, this paper considers paraxial rays and others as the same to trace their paths. For checking out the tracing algorithm, a programs based on OpenGL is created in three-dimension and with real time controlling. Another program is implemented to show how the digital photos should be changed when using different lens parameters.

Keywords: Optic Simulation, Optical Path, Lens, OpenGL, Computer Vision

1. Introduction

In optical system, parts such as lens and diaphragm are used for focusing rays on the screen. These optical devices can change the direction of rays that radiated from light source or others. In ideal optical system, all rays from a point should be focused on an image point; and all image points should be in a single plane of the screen at a proper position. In reality, however, it's impossible to design an ideal system; there exists some deviations from the ideal. The deviation is known as aberration. There are two types of aberrations, geometrical and chromatic. The former relies on the geometry of the lens, and the latter is on the refractive index function of wavelength. Traditionally, most of the real optical systems are designed based on ideal optical that helps to calculate some approximate values then approach to the precision one according to aberration theories. The aberrations can be reduced by combining lenses of different types. To do this, it is very important to grasp the features of different aberrations such as spherical aberration, astigmatism, distortion and coma. Though the traditional approximation method is good enough and can be calculated easily, in some special situations the exact

one can be essential.

For there is no way to get the exact result in ideal optical based on the actual surface geometry, it is desired to have a simulation system to get the result visually. As a method based on three dimensional geometry to calculate the optical path for each ray, the techniques of ray tracing can be exactly for all the rays. There is no more necessary to classify rays in different position to paraxial and others, also to the method for optical path calculation. That is, the non-paraxial that hardly to calculate can also have the same treatment as paraxial rays. This removes some traditional limitations that lead to difficult calculation such as lens' shape and light source position. So a result of cubic lens can be calculated in the same way to spherical lens, if only the geometries and the refraction features of the lenses are known.

Within a complex and dynamic manufacturing of optical lenses, gathering valid data is a difficult task, and to understand the data properly is furthermore difficult. Computer simulations by the techniques of computer graphic is a good choice [1]~[6]. In this paper, we will describe the theoretical basis of our tracing algorithm, and provide an assessment of its accuracy. Using CG to create a virtual reality environment can show the optical paths intuitively.

In addition, it can also get easier for the result checking. Real time response and dynamic controlling of the parameters are also helpful for finding best parameter sets.

The rest of the paper is arranged as follows. Section 2 explained the tracing algorithm in detail. Section 3 created a program that based on the algorithm discussed in section 2 with OpenGL. Section 4 created another program to simulate lens in some special situation and at last discussed some improvements and other possible applications.

2. Tracing Algorithm

As shown in figure 1, consider the incident ray is from for the point light source S. Then name the refractive point to P, the normal vector at point P to \overline{QO} . Refraction of the lens is set to N. There are at least two ways to calculate the outgoing ray \overline{PT} which is the refracted ray.

The traditional one uses $\angle SPQ$ and follows the refraction law to solve $\angle OPT$. It's perfect to determination ray's path just for it is in meridian plane. The assumption that paraxial rays not in the meridian plane also have the same behavior to ones in meridian plane leads to the reason of this trouble. Beside this, calculation for non-paraxial rays has to use some special methods determined by the ray's position. The result calculated in this way can be corrected by aberration to a more exactly result.

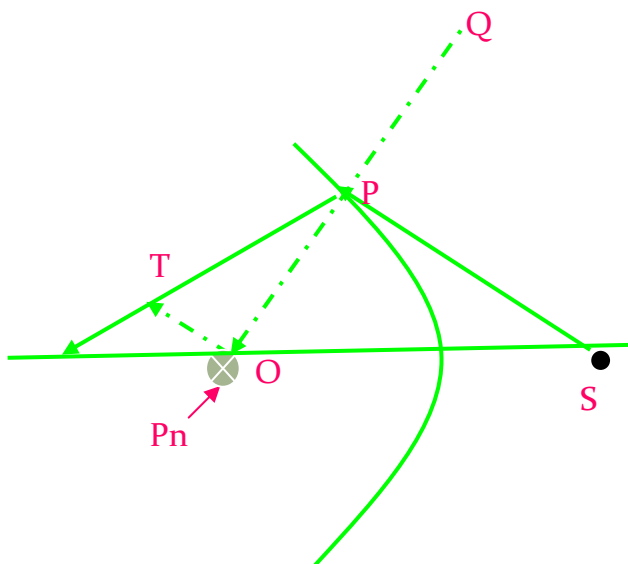


Figure.1 Optical System

The second way based on space geometry can

treat rays in all patterns in a same way. Assume that there is an outgoing ray \overline{PT} , the purpose is solving its direction. Here presents a method to calculate the direction as follow.

As the vector \overline{SP} and normal \overline{QO} is already known, $\angle SPQ$ should be:

$$\angle SPQ = \arccos\left(\frac{\overline{SP} * \overline{QO}}{|\overline{SP}| * |\overline{QO}|}\right) \quad (1)$$

All the points S, P, Q, O should at plane SPQ, then $\angle OPT$ is as:

$$\angle OPT = \arcsin(\angle SPQ * \frac{1}{N}) \quad (2)$$

(Assume that right of figure 1 is air with refraction of 1.0)

Find a point O at the normal of P with the length of PO is 1. So the vector \overline{PO} is a normalized vector. From O, make a new vector intersecting outgoing vector at point T and the vector is vertical to \overline{PO} . Model of \overline{OT} can be calculated as:

$$|\overline{OT}| = \tan * |\overline{PO}| \quad (3)$$

Consider all these vectors are in single plane, there should be a vector \overline{PN} which is the plane's normal. It can be derived by \overline{SP} and \overline{PO} :

$$\overline{PN} = \overline{SP} \times \overline{PO} \quad (4)$$

For \overline{OT} is vertical to \overline{PO} and \overline{PN} , \overline{OT} can be determined by normalized \overline{PO} and \overline{PN} as:

$$\overline{ot} = \overline{PO} \times \overline{PN} \quad (5)$$

With the result that the model and direction of \overline{OT} , it can be shown as:

$$\overline{OT} = \overline{ot} * |\overline{OT}| \quad (6)$$

Finally, just plus them as:

$$\overline{PT} = \overline{PO} + \overline{OT} \quad (7)$$

It's the path for single ray. No matter rays with different parameters, it can be calculated as previous method. This calculation process should be rapidly and exactly by using computer.

3. Real Time Simulation by OpenGL

3.1 Application Details

There are several kinds of optical simulation soft and methods for optical design [7] [8]. These can make simulations and manufacturing more

efficiency [9] [10]. These soft includes components that can produce a virtual optical system and calculate the system parameters. However, these numerical results can not show a system intuitively, so changing the result's form to graph which make the design more effective is necessary.

The cross-platform library OpenGL is easy to use and can performance graph with high quality. Based the OpenGL, a three-dimension program to simulate lens with a point light source is produced.

Figure 2 shows a simple optical system that only has a light source a spherical lens and a screen. Right of the system is the point light source from which rays radiated out. At the origin is the spherical lens whose parameters can be changed. To this lens, the variables are position, refraction, aperture, curvature radius for front surface and also for back surface. At the left there is the screen whose position can be changed to get different spots. From the source to screen, optical path is calculated and shown as the lines. Points on screen are the spot of the lens. At top right the spot is scaled to a suitable size for watching clearly. All the parameters motioned can be controlled by mouse and keyboard real time or can be set by application menu. And the perspective also can be set in real time. Parameters in this program are initialed to screen position -1, light source position 1, lens' position (0, 0, 0), the refraction to 1.7, aperture to 0.2, front curvature radius to 0.3, the back one to 0.6. In the follow, this program is used to simulate some optical aberration phenomenon and check the core algorithm which motioned in section 2 out.

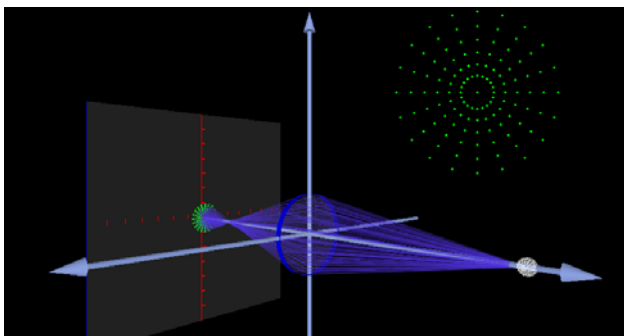


Figure.2 Default State

3.2 Spherical Aberration

Spherical aberration is the simplest case in aberrations. Ideal optical considers that parallel rays should be focused on one point with a suitable screen position. But as the spherical aberration, this should be never happened, whatever the spots are minimized, they can not become a point.

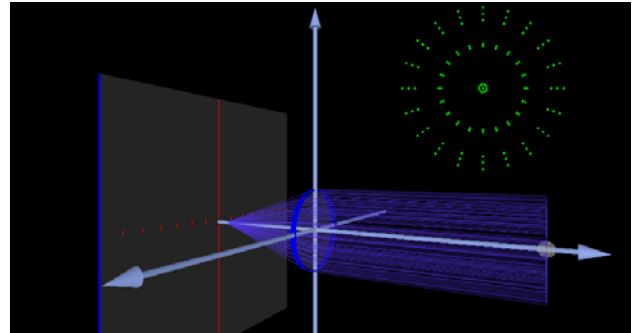


Figure.3 Spherical aberration

The parameters changed as: the lens center is at origin, front curvature radius 0.3 and back 0.6, refraction 1.7, aperture 0.2. Adjust the screen and at the position -0.730 found a minimized spot.

As the figure 3, with these parameters, by adjusting the screen to have minimized spot still can not focus the spots to a point.

3.3 Coma

Coma should be caused by non-paraxial wide beam. Keep almost the parameters as the previous experiment, and just change the lens position to a new one that far from axis to make a non-paraxial environment.

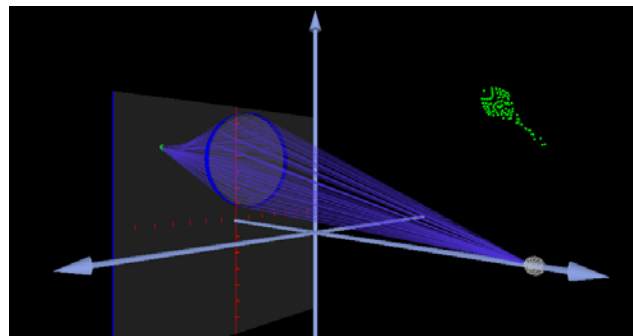


Figure.4 Coma

As the figure 4, lens' position is changed to (0, 0.35, 0.39), and the minimized spot is found when screen at -0.650. As its shape which is shown as figure 4 is like comet, this aberration is called coma.

3.4 Astigmatism

This aberration needs thin beam, it's necessary to set the aperture to a small value and need non-paraxial rays.

In figure 5, the aperture is set to 0.05 and screen center is at (0, 0.35, 0). These 6 pictures are phases

for spots changed by adjust the position of screen at -0.825, -0.780, -0.750, -0.740, -0.730, -0.695.

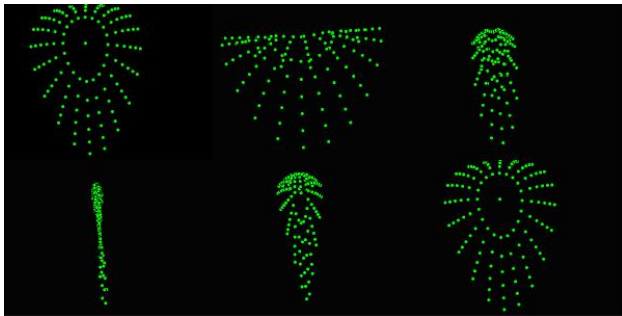


Figure.5 Astigmatism

3.5 Field Curvature

When mapping a surface picture to a hyperplane, the picture should be distorted. And, this distortion should be serious with the increasing of the incident angle.

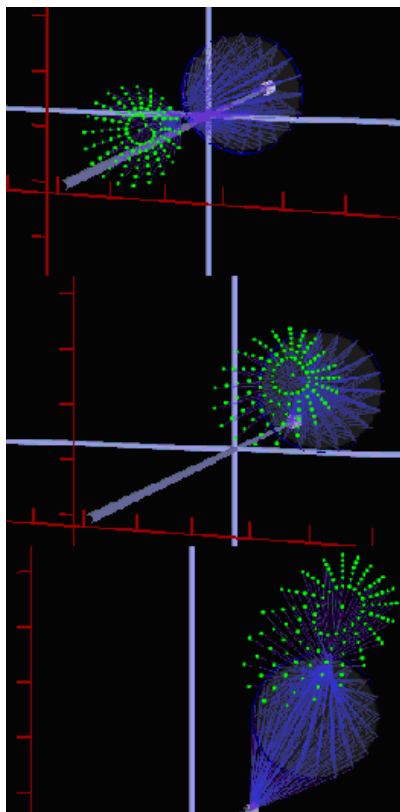


Figure.6 Field Curvature

This experiment uses default parameter set except the position of lens. Figure 6 has three steps with lens position changed to (0, 0.8, 0.11), (0, 0.2, 0.28),

(0, 0.4, 0.4). As the figure 6, at first step, it's almost a perfect circle, then in step 2 and 3 the distortion becomes terrible. The spot becomes bigger when incident angle increases this can make photo's fuzzily at the areas that far from center.

3.6 Analysis

In geometry optical books, there are many photos shows the aberrations and can use to ensure these pictures have the same result. And if it's necessary, numerical data also can be output, this proof the algorithm's correctness.

In this program, light source is considered as a point and set parameters only for aberrations. But the parameters can be set to any value that not invalid, by this the program can simulate all situations for this type optical system. In addition, just add some new lenses or light sources in this optical system, more complex simulation will become possible. And there is no limitation to the shape of lens, by defining normal, cubic even polygon lens can be simulated. Using this program in geometry optic lessons should make the optical knowledge understood more easily because of that it shows not only spots but also the path and can be controlled in real time.

4. Surface Tracing for Lens Design

4.1 Really Optical System

Actually, point light source is rarely at reality optical system. To approximate system as this, consider all things that we can see as a light source which there is no matter they can shine or not, even is just reflecting.

For we can not focus all the rays perfectly, it's common that only let the target region image best. Because optical systems like microscope system, telescope system and camera have different purpose, the targets are also different. Lens always designed for the making the interest target clearly. But, how can designers know it's suitable or not through that parameters? Producing of lens is difficulty and taking a long time, another way to simulate an optical design is necessary. The follow tries to take a photo by using a virtual optical system.

4.2 Point Tracing

As simulated in section 3, that program can get perfect optical path for one point light source. If add

amount of light source, that program can also simulate for surface light source. In real world, light sources that have different strength radiant rays to lens and by different angle to lens should have different color and contrast. In real world, rays can be considered as vectors. For simulating it, an environment is necessary.

For take it easy, consider rays as scalar first. When take a photo of a digital picture, assume that every point on the picture can only radiant forward. What these rays should be after refracted by a given lens? If using an ideal thin convex lens, the system should be as figure7.

In figure 7, the aperture of the lens is big enough, and R is the radius of lens, N is the refraction, L is the length from the optical main plane to screen. Here assume each pixel to be a light source that can only radiant at its forward. The rays should be focused on the screen as spots.

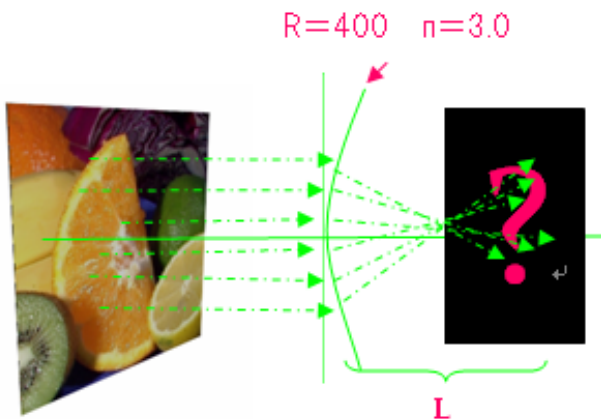


Figure.7 System Total View

Though it's not a real, we can consider it as a part of a real system. With the L adjusting, a series picture that changed by the lens should be output.

Figure 8 shows how the picture changed with the adjusting of L.

This changed can be explained by geometry. The focus of this lens can be calculated as:

$$f = \frac{NR}{N-1} = 600 \quad (8)$$

At the field that L is smaller than f, rays get closely more and more, thus the picture also became smaller and smaller. And it's twisted special at the corners. Follow this, when L is equating to the focus as 600, the picture almost be a point. At this length, almost points of the source picture focused to minimized spots.

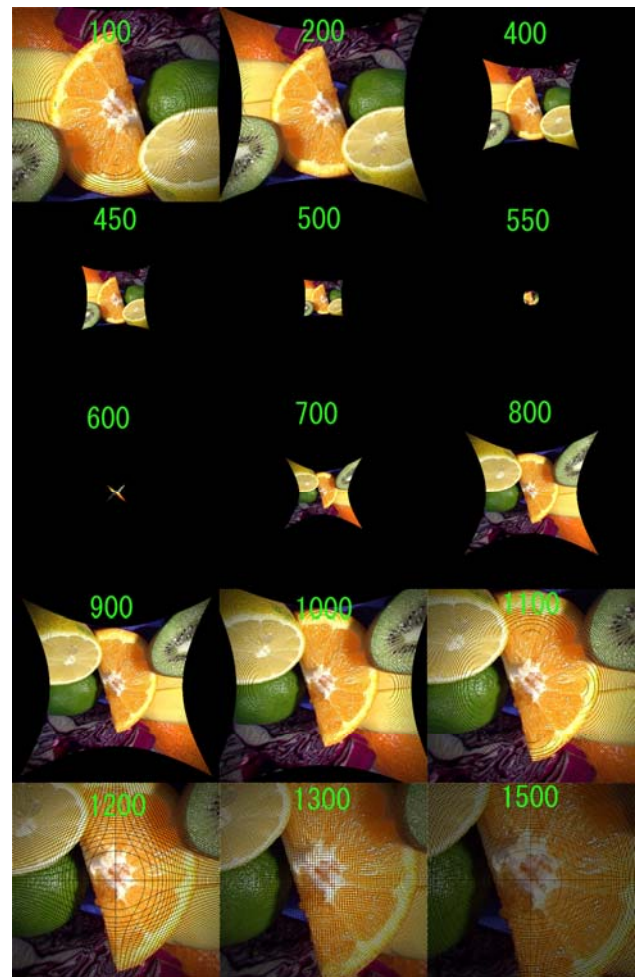


Figure.8 Convex Lens Process

Go on the trial in these pictures, when L is at the field f to 2*f, the result picture is transposed and still reduced. But it became bigger and bigger. Because of rays running to the opposite of its original position, the picture should be always transposed when L is greater than f. And there are so many black points at the corners some areas even black fields. The rectangle picture that has not enough samples at corners to support enough samples to the spherical lens leads to this.

Finally, as rays get the field greater than 2*f, the picture is scaled up. The length of L has no limitation, but with the L being bigger, black points also gain. Then the whole picture should be dark more and more. The reason of this is that a part of ray loss to the limited screen. Lacking of sample is also the reason of mesh in the results.

As another example, figure 9 show pictures for concave lens. Here using N to 3.5, R is 200, then its focus is -280.

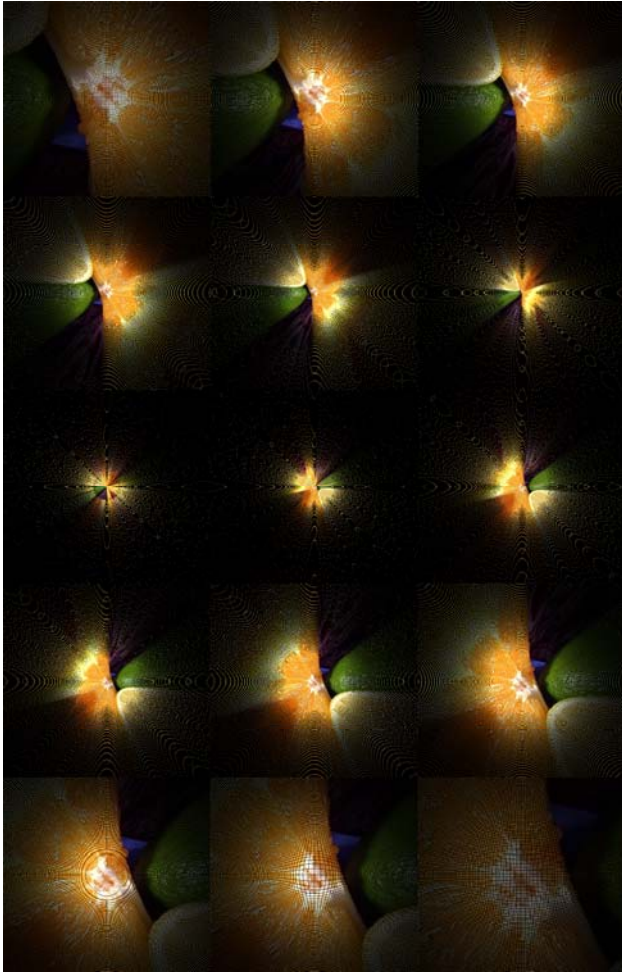


Figure.9 Concave Lens Process

4.3 Surface Tracing

As motioned in the previous sections, rays can be considered as vectors which have two features as direction and strength. Section 4.2 discussed a situation with limited light direction. Now removes this limitation and consider that each light source can radiant to all directions in space.

As shown in figure 10, without considering the intensity strength of ray, each light source should radiant to all the surface of the lens. It's impossible to trace optical path at that situation because there is too many paths to be calculated. Take some sample points at the lens' surface to calculate the refraction path way also can have an result that good enough. Though this optical system can not focus all the light sources on one screen plane to produce a photo

clearly, it can show the situation that how a plane should be when refracted by a lens.

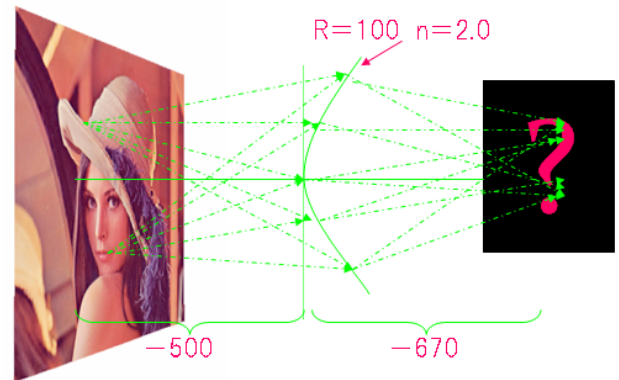


Figure.10 Surface Tracing System

This system sets the radius to 100 and refraction to 2.0, the focus can be calculated as 200 according to formula 8. Rays from one light source point should be focused as a scattered spots instead of a point, and the spots' size should be changed by adjusting the position of screen. Real optical system should keep the user interested region of result image clear, that is, it should minimize the scattered spots. For example, if one want to take a photo for some body, it's most important to make sure that the person in the photo is clear but not the background.

Figure 11 shows the result of the simulation when the sample density for one source is set to be 30.

We can see that for each of tracing sample (there are 30 samples for one source point), there is a twisted photo in the result image. Because of the field curvature, it looks like mapping the original picture to many hyperplanes. In lens design, it is required to map these hyperplane a common plane by tuning the parameters in the optical system for practical use. Our research provided a means to simulate the result of the optical system before the lenses are really produced actually. In figure 12, it shows the simulation result for the case that 600 samples for each of the source points.

In our simulation system, only the path ways of the rays are considered as vectors, the strengths for rays of different directions radiated from one source



Figure.11 Low Density Surface Tracing

are ignored right now. But it is possible to be extended in the future. For these complicated situations, where the strengths of the rays radiated from a source may vary for particular directions, it is difficult to use the traditional method to record the result for the direction variations by using film, CCD or CMOS, where only the final sum of the amount of luminous flux are recorded. So, the original direction information cannot be extracted from the photos taken by these ways. Though Laser hologram contains the directional information, it is preferred to be computed through simulation. To create a simulation environment like this, it should be helpful by using CG technology. In CG, it's easy to create a lens model and set the parameters for the directions, and the strengths of the rays, and set the parameters for the material features of the lens. We can get the simulation result of the optical system before the lenses are really manufactured. By using shading language, it can be dealt with even more effectively. Fortunately, algorithm doing this has been developed for many years, the only different is the viewpoint. As the word "point" in name, CG calculates rays from source to this point, of cause, possible for all directions. Thus, if we create a lens instead of the view point, the simulate environment

should be accomplished. That is, create a program as section 3, but tracing it as section 4 (multiply sources and surface tracing). Then, add some other parts or more lens with suitable values, this system would world as a real one.

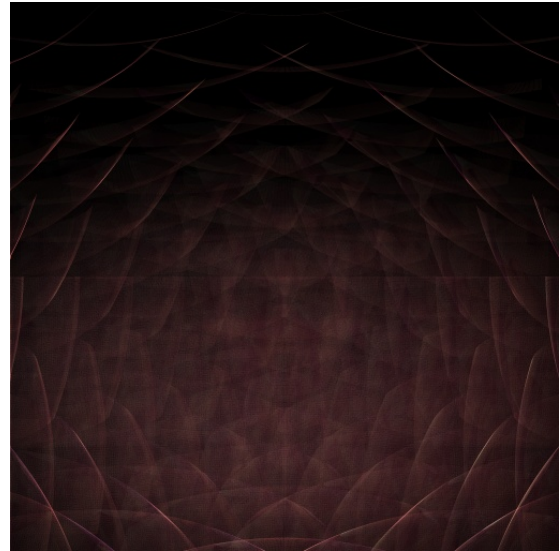


Figure.12 High Density Tracing

Simulation system can be used not only for designing the optical system, but also for certificating the existed optical system Assume that a system with all parameters known but unfortunately get some damage and can not be repaired soon, such as an exploration robot in some situations. If a inspection device is preinstalled, we can rectify the damaged images by updating parameters for the damaged system.

5. Conclusion

Simulation can be helpful to correct theory or test design of optical system. In this paper, a method leads to intuitive result is presented and some simulations results are provided by using tracing algorithm. As for the future work, we are planned to use GPU to calculate the vectors and get a better respond time.

References

- [1] Tong Gao, Xin Zheng, Qian Yin, "Software-Based

- Non-invasive Implementation of Binocular Vision”,
*Computer Science and Software Engineering, 2008
 International Conference on Volume 2*, 12-14 Dec.
 2008, pp:1102 - 1105
- [2] Fu Zhanping, Di Yangzhou, Shang Xiliang, Qiu Zhi,”Flight Course Replay And Simulation Base On OpenGL”, *Proceedings of TENCON’02, VOL 3*, Oct.2002, pp.1757-1760.
- [3] Moller.R, “State-of-the-Art 3D Graphics for Embedded Systems”, *Proceedings of the 6th International Caribbean Conference on Devices, Circuits and Systems, 10.1109*, Apr.2006, pp.339-343.
- [4] Tangyun Dai, Zemin Wang, Shouheng Xu,” Research of Creating and Fetching 3D Models of Virtual Reality Based on OpenGL”, *Proceedings of the 2006 IEEE International Conference on Mechatronics and Automation 25-28*, Jun.2006, pp.1991-1995.
- [5]Marino, G., Vercelli, D., Tecchia, F., Gasparello,P.S., ”Description and Performance Analysis of a Distributed Rendering Arcitecture for Vitual Environments”, *Artificail Reality and Telexistence,17th international Conference on 28-30* , Nov.2007, pp234-241.
- [7] Christopher W. Penney, “Rotman Lens Design and simulation in software”, *Microwave Magazine, IEEE Volume 9, Issue 6*, December 200,8 pp:138 - 149
- [8] C.W. Penney, R.J. Luebbers, and E. Lenzing, “Broad band Rotman Lens simulations in FDTD,” in *Proc. 2005 IEEE AP-S Int, Symp.*, July 2005, vol.2B, pp. 51–54.
- [9] Olugu, E.U.; Kuan Yew Wong, “Simulation Study on Lens Manufacturing Process Flow”, *Modeling & Simulation, 2008. AICMS 08. Second Asia International Conference on 13-15*, May 2008 pp:806 - 811
- [10] Korn, S., G. R. Burns, and D. K. Harrison, “The Application of Multiparadigm Simulation Techniques to Manufacturing Processes”, *International Journal of Advanced Manufacturing Technology, Springer-verlag*, London, 1999, pp:869–875.