DEVELOPMENT OF A SOFTWARE SIMULATION TOOL FOR MOVABLE OPTIC COMPONENTS

Luciano Selva Ginani, luciano.ginani@tu-ilmenau.de
René Theska, rene.thesk@tu-ilmenau.de
Technische Universität Ilmenau

Abstract. *Many optical simulators are available commercially, offering a wide range of features and covering not only geometric but also diffractive optics and many other fields. But most of those systems are aimed at lens design and many times inappropriate for the evaluation of mechanical movement between the different elements of the optical system. This article presents the first steps in the development of a software tool to address the particular field of the simulation of movable optical components, providing a tool for the analysis and optimization of those systems and their kinematics while taking in account the optical properties of the design. The simulator implements the three dimensional rotation and translation of plane, spherical and aspherical optical components while dealing with the propagation of the light through the optical system. The basics of the geometric optic theory behind the simulation engine is shortly presented together with the simulators interface and an example application, where a plane mirror is rotated in two axes in order to steer a light beam and generate a scanning pattern in a surface. The results of the simulation include the movement of the optical focus point of the system and the deformation of the light spot on the projection surface. Concluding the article, the future steps in the development of the system and the pretended features are presented.*

Keywords*: geometric optics, optical design, optical simulation, ray-tracing*

1. INTRODUCTION

Simulation and analysis software offer optical designers, engineers and researchers a valuable help in the design of optical systems. A wide range of such tools are commercially available and offer different features, but most of those systems are inappropriate for the evaluation of free movement between system components. In this article the first steps in the development of a tool to fulfill this deficiency is presented, aiming at offering an easy design and simulation environment while keeping the modeled elements unconstrained, therefore not forcing optical alignment or optical centralization.

The system is based on 3D ray-tracing and has two basic concept ideas: geometrical description of the elements involved and free simulation of geometrical and optical parameters. These concepts allow the simulation of surfaces and rays through the optical system while the ray-tracing solves the optical path and generates the simulation data for the analysis of the optical properties of the designed optical system.

In this first development phase the use of sequential ray-tracing was adopted because of its simplicity and efficiency in the simulation of geometric optics. Ray-tracing is a widespread technique for the design and simulation of optical elements and systems through the generation of an optical path in a system based on the laws of geometric optics. By successively applying the laws of reflection and refraction on the optical surfaces (lens, mirror, screen, etc.), a light ray can be propagated from its source to its target and the analysis of the optical properties carried out.

The main objective behind the development of the presented tool is the systematic analysis of different designs used to steer light beams in systems such as confocal-microscopes, laser engraving and cutting tools, etc.

The mathematical basis of the simulation tool is presented, followed by the developed software interface and a practical example of the analysis of the rotation of a plane mirror in two axes to steer a light beam and generate a scanning pattern.

2. MATHEMATICAL BACKGROUND

Optical design and simulation software are usually based on one of three categories: sequential ray-tracing, non-sequential ray tracing, and finite-difference time-domain (FDTD) simulation (Miller *et al.*, 2005).

Ray-tracing involves modeling the geometrical components of optical systems, defining the optical properties of these objects, approximating light sources with directional rays and then propagating these rays through system models (Spencer and Murty, 1962).

Sequential ray-tracing trace the rays created sequentially, intersecting optical elements one at a time in a pre-defined order. It is most often used to design and optimize systems of lenses.

Non-sequential ray-tracing permit rays to encounter surfaces in any order and any number of times with automatic ray splitting, allowing rays to scatter and interact with system components as they do in reality. It is often used to model complex optical systems in which scattering, stray light characteristics, coherent effects, and polarization effects must be known and controlled.

With the need to achieve an even greater accuracy and to deal with wavelength-scale structures, conventional ray-tracing starts to fail. FDTD solves Maxwell's equations to propagate electromagnetic fields through optical structures. This enables the consideration of wave-optics phenomena and the precise simulation of optical systems.

The developed tool utilizes the sequential ray-tracing technique to deal with the optic simulation, while using linear transformations for taking into account the mechanical movement between components. A more detailed description of this mathematical basis is presented.

2.1. Sequential ray-tracing

A basic optical system can be described through the reflection and refraction laws illustrated in Fig. 1 and to implement those principles in the macroscopic scale, ray-tracing is a powerful tool.

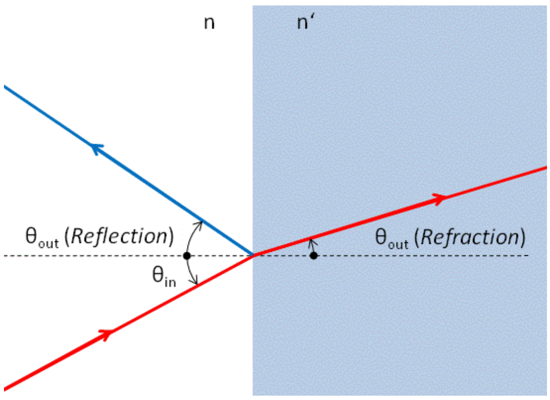

Figure 1. Reflection and refraction

Both sequential and non-sequential ray-tracing techniques can be summarized through the flowchart illustrated in Fig. 2. While non-sequential ray-tracing demands the determination of the next surface that the ray intersect, by sequential ray-tracing the surface sequence is predetermined.

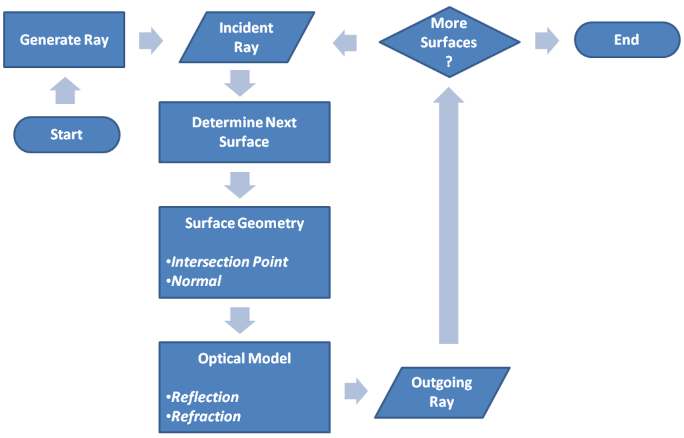

Figure 2. Ray-tracing flowchart

The ray-tracing tool is the basis of the simulation system and, together with the coordinate transformations, builds up the geometrical model of the system.

The first step implementing the ray-tracing tool is to define mathematically the involved structures: ray and surface. A ray is defined through the parametric equation shown in Eq. (1) while a surface can be defined through an arbitrary function $f(x, y)$ as in Eq. (2).

$$\vec{r}(t) = \begin{bmatrix} x(t) \\ y(t) \\ z(t) \end{bmatrix} = \begin{bmatrix} x_o \\ y_o \\ z_o \end{bmatrix} + \begin{bmatrix} x_d \\ y_d \\ z_d \end{bmatrix} t \qquad (1)$$

$$\vec{s}(x,y) = \begin{bmatrix} x \\ y \\ f(x,y) \end{bmatrix} \tag{2}$$

, where $[x_o \quad y_o \quad z_o]'$ is the origin of the ray and $[x_d \quad y_d \quad z_d]'$ its normalized direction.

In order to enable the surface $\vec{s}(x,y)$ to move freely through the space and simulate the mechanical movement of the system components, its function must be redefined as:

$$\vec{s}(x,y) = T(\alpha,\beta,\gamma,d_x,d_y,d_z)\begin{bmatrix} x \\ y \\ f(x,y) \end{bmatrix} \tag{3}$$

, where $T(\alpha,\beta,\gamma,d_x,d_y,d_z)$ is the linear transformation responsible for the surface's movement.

Once the basic mathematical structures are described, the next step is the definition of the ray-tracing operation illustrated in Fig. 2.

In sequential ray-tracing, the surface sequence is predetermined, so the determination of the next surface is automatic. For the evaluation of the surface geometry, the calculation of its intersection point can be defined as the solution of the vectorial equation:

$$T\vec{s}(x,y) - \vec{r}(t) = \vec{0} \tag{4}$$

As the function $f(x,y)$ can be any arbitrary function, the calculation is carried out numerically using the Levenberg-Marquardt algorithm (Lampton, 1997; Jacoby *et al.*, 1972) to determine the parameter t that minimizes the function:

$$\varepsilon(t) = \Big(z(t) - z\big(x(t),y(t)\big)\Big)^2 = \Big(z_o + z_d t - T^{-1}(\alpha,\beta,\gamma,d_x,d_y,d_z) f(x_o + x_d t, y_o + y_d t)\Big)^2 \tag{5}$$

, where $T^{-1}(\alpha,\beta,\gamma,d_x,d_y,d_z)$ is the inverse linear transformation.

Once the intersection point (x_p,y_p,z_p) is determined, the surface normal in this point can be calculated as:

$$\vec{N} = T\left[\begin{bmatrix} -\frac{\partial f(x,y)}{\partial x} \\ -\frac{\partial f(x,y)}{\partial y} \\ 1 \end{bmatrix}\right]_{(x_p,y_p,z_p)} \tag{6}$$

As the incident ray and the normal vector are now known, it is possible to calculate the angle between them as:

$$\begin{cases} \theta_{in} = \vec{N} \cdot \begin{bmatrix} x_d \\ y_d \\ z_d \end{bmatrix} \\ \vec{V} = \vec{N} \times \begin{bmatrix} x_d \\ y_d \\ z_d \end{bmatrix} \end{cases} \tag{7}$$

, where θ_{in} is the angle between both vectors, and $\vec{V}$ is the rotation normal.

The final step in determining the outgoing ray is the calculation of the exit angle using the optical model. In this first phase of the optical simulator, only reflection and refraction are addressed, without any considerations of wavelength, diffraction and other optical phenomena.

$$\begin{cases} \text{Reflection} \rightarrow \theta_{out} = -\theta_{in} \\ \text{Refraction} \rightarrow \theta_{out} = \sin^{-1}\left(\frac{n}{n'}\sin(\theta_{in})\right) \end{cases} \tag{8}$$

The outgoing ray can then be calculated as:

$$\vec{r}_{out}(t) = \begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix} + \mathrm{ROT}_{\vec{V}}(\theta_{out})\begin{bmatrix} x_d \\ y_d \\ z_d \end{bmatrix} t \tag{9}$$

, where $\mathrm{ROT}_{\vec{V}}(\theta_{out})$ is a rotation of θ_{out} around the vector $\vec{V}$.

The whole process is then recursively repeated until all surfaces are processed.

It is important to notice that through considering a ray as a mathematical abstraction with a point of origin and a direction, properties such as energy, polarization and the wave nature of light are not addressed by the simulator. The purpose of this first development is the creation of the 3D geometrical basis upon which future developments involving more advanced optic simulation topics will be founded.

2.2. Surface Modeling

There is a wide range of surfaces that are used in the production of optical components. When dealing with rotation symmetrical surfaces the most common are spheres, aspheres and planes. Those surfaces can be mathematically described through Eq. (10) (Schröder, 1990).

$$f(x,y) = z = \frac{C(x^2+y^2)}{1+\sqrt{1-(K+1)C^2(x^2+y^2)}} + K_2(x^2+y^2)^2 + K_3(x^2+y^2)^3 + K_4(x^2+y^2)^4 + K_5(x^2+y^2)^5 \quad (10)$$

So that those surfaces are fully defined through the parameters C, K, K_2, K_3, K_4, K_5.

The simulator presented deals, up to now, only with rotation symmetrical surfaces, so that Eq. (10) constitutes the basic model for surface representation in the optical simulation.

3. SYSTEM OVERVIEW

The developed simulator, although in its initial steps, offers already a powerful analysis tool for geometric optics. In Fig. 3 the software tool main screen is shown.

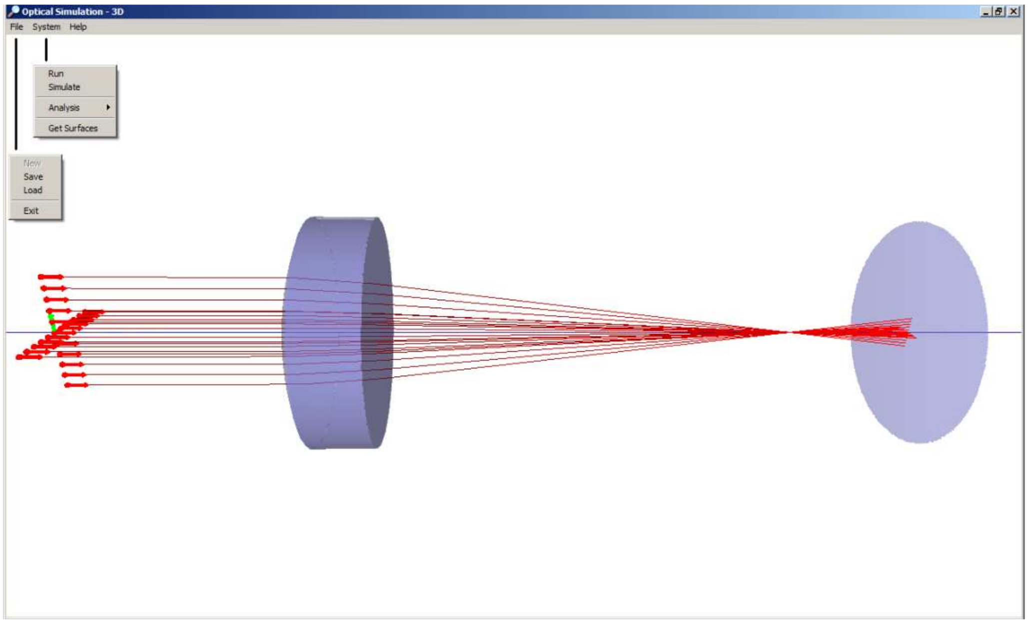

Figure 3. Simulator main screen

The ray-tracing tool is the basis of the simulation system and, together with the coordinate transformations, builds up the geometrical kernel of the system.

For the visualization of the simulation and plotting rays and surfaces a 3D interface based on the OPEN-GL programming library was implemented. The system was developed using the C/C++ programming language and the free compiler Bloodshed DEV C++.

4. PRACTICAL EXAMPLE

Optical beam scanners (Marshall, 1991) are used in a wide range of applications from data storage and barcode readers to confocal microscopy, industrial metrology and even image projectors. In each of those systems it is of extreme importance the ability to steer the light beam into a specific position, and for that task the coupling between mechanical and optical systems plays an important role.

A basic kinematic design for such a system is the use of a single laser, a two-axis rotational plane mirror and a simple aspheric objective as shown in Fig. 4.

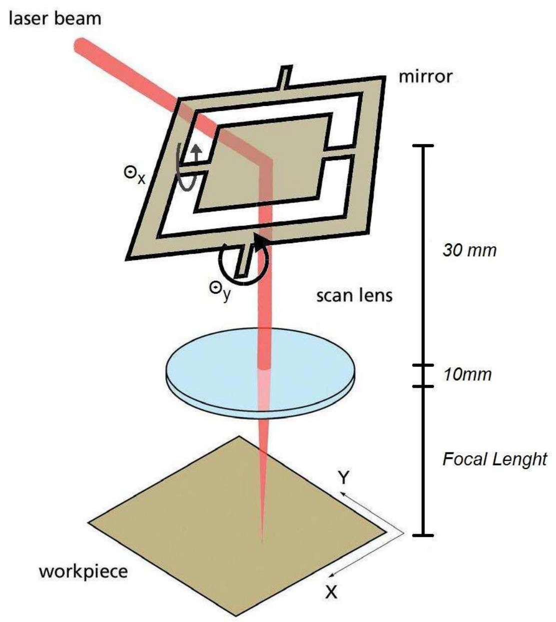

Figure 4. Simulation model of a 2D scanning system

As an application example of the developed tool, the analysis of this basic architecture is carried out using a simple objective. The first step is the description of all the involved surfaces through its parameters (Tab. 1) in order to build the virtual system shown in Fig. 4.

Table 1. Surface parameters for the application example.

Surface	**Parameters**						
	Type	C	K	K_2	K_3	K_4	K_5
1	Mirror	*0.0*	*0.0*	*0.0*	*0.0*	*0.0*	*0.0*
2	Lens	*0.02078511*	*0.0*	*-1.507097E-5*	*0.0*	*0.0*	*0.0*
3	Lens	*-0.014463401*	*0.0*	*-1.269052E-5*	*0.0*	*0.0*	*0.0*
4	Screen	*0.0*	*0.0*	*0.0*	*0.0*	*0.0*	*0.0*

The first analysis carried out is the dependence of the scanning point movement and the rotation angles of the scanning mirror. Figure 5 illustrates the simulated behavior of the system. Ideally, the independency of both axes should be expected, but, because of the optical aberration inherit from the lens geometry, that is not achieved.

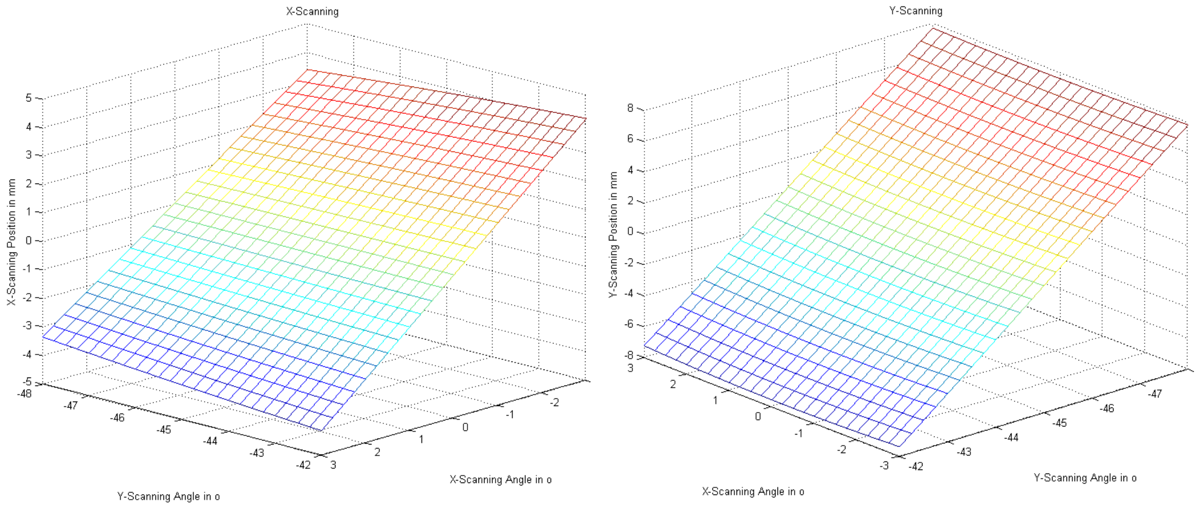

Figure 5. XY-scanning movement

The second analysis carried out is the analysis of the focal length of the system. Fig. 6 illustrates the dependence of the focal length and the angle of the scanning mirror. It can be observed, that, as expected, the focal length of the system changes as the scanning angles change.

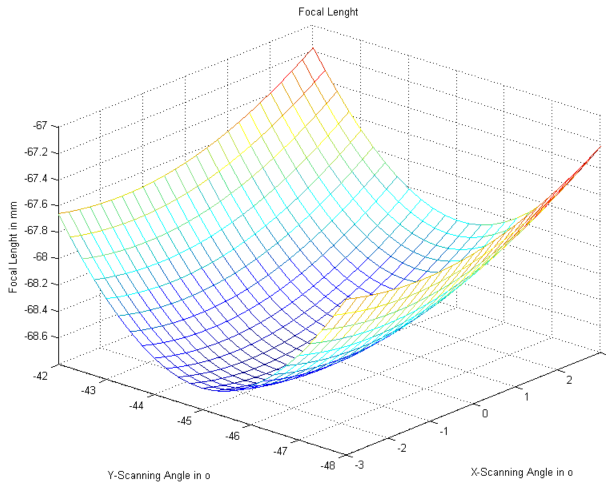

Figure 6. Focal length of the scanning system

With this data the optical system can be redesigned to compensate the observed aberrations, or, if possible, a computational correction can be carried out. Either way, the simulation data is of great importance for the evaluation of the designed system and the analysis of its performance.

5. CONCLUSION AND FUTURE WORK

The development of simulation tools for the design and analysis of optical systems is of great importance to opto-mechanical engineering as it facilitates the design process reducing development time and costs. The integration of optical and mechanical systems to produce even more complex systems demands the development of new tools that can address specific problems.

This paper presented the first steps in the development of an optical simulation tool capable of dealing with free unconstrained movement between system components, therefore not forcing optical alignment or optical centering.

In order to allow the free movement of surfaces through the 3D space a coupling between ray-tracing techniques and linear transformations was presented and implemented. Its mathematical background was presented together with a brief description of the sequential ray-tracing technique.

As an application example of the developed tool, the analysis of a two-axis rotational plane mirror in combination with a simple lens is carried out and the obtained results discussed.

Future work within the development of the software tool includes the automation of several analysis features and the introduction of non-sequential ray-tracing and wave optics, aiming at a more accurate analysis of optical systems and at the simulation of more complex systems. Another feature to be included in the system is the integration of an optical materials database.

7. REFERENCES

Jacoby, S.L.S., Kowalik, J.S., Pizzo, J.T., 1972, “Iterative Methods for Nonlinear Optimization Problems”, Ed. Prentice Hall, N.J., USA, 274p.

Lampton, M., 1997, “Damping-Undamping Strategies for the Levenberg-Marquardt Non-Linear Least-Squares Method”, Computers in Physics, Vol. 11, No. 1, pp. 110-115.

Marshall, G.F., 1991, “Optical Scanning”, Ed. Marcel Dekker Inc., New York, USA, 896p.

Miller, S., Pond, J., Michel , B., 2005, “Raytracing trifft Maxwell-Gleichung: Über die Integration von mikro- und makrooptischem Design”, Photonik 6/2005, pp. 40-43.

Schröder, G., 1990, “Technische Optik: Grundlagen und Anwendungen”, Ed. Vogel, Würzburg, Germany, 206p.

Spencer, G.H., Murty, M.V.R.K., 1962, “General Ray-Tracing Procedure”, J. of the Optical Soc. of America, Vol. 51, No. 6, pp. 672-678.

5. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.