

LENS FOR A COMPUTER MODEL OF A POLARIZING MICROSCOPE

VICTOR A. DEBELOV^{*}, KONSTANTIN G. KUSHNER[†] AND
LUDMILA F. VASILYEVA^{*}

^{*} Institute of Computational Mathematics and Mathematical Geophysics SB RAS
Novosibirsk, Russia
e-mail: debelov@oapmg.ssc.ru, ludvas@oapmg.ssc.ru, web page: <http://www.icmmg.nsc.ru>

[†] Novosibirsk State University, Novosibirsk, Russia
Email: kushner.k.g@gmail.com - Web page: <http://www.nsu.ru>

Summary. In this paper, the development of a computer model of a lens for ray tracing is considered. The model provides physically correct processing of linearly polarized light and traces some parameters, such as ray coherence and optical path (phase). The lens is assumed to be made of a homogeneous isotropic material with given refraction and absorption indices in the visible region of the spectrum. This is the case for lenses of a polarizing microscope. The software interface of the module makes it possible to consider the lens as a primitive of a 3D scene and use it as such in programs of photorealistic rendering.

1 INTRODUCTION

After the development and verification of an algorithm of interaction of light rays with optically anisotropic transparent media¹, some successful experiments with computer simulation² of well-known physical experiments on visualization of the phenomenon of interference (Young's experiment and that with amplitude division³) we decided to construct a computer model of a polarizing microscope (PM).

A polarizing (petrographic) microscope⁴ is an important tool for mineralogists and crystallographers. A polariscope, which is a simplified version of a polarizing microscope, is routinely used by jewelers and some other specialists. Placing a mineral sample in a polarizing microscope or polariscope, the researcher observes or photographs the interference image (Fig. 1) depending on the optical characteristics of the mineral and PM settings.

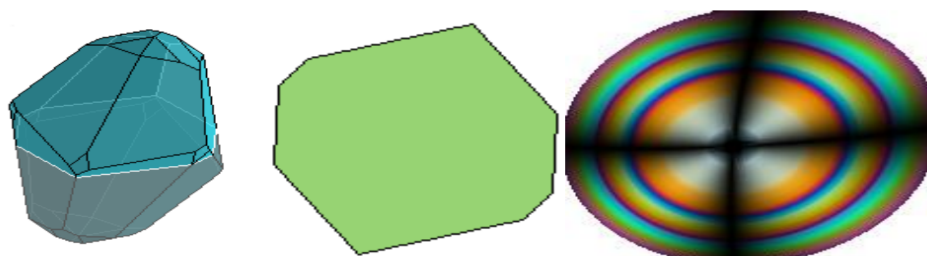


Figure 1. Left to right: 1) cross-section of crystal to prepare a specimen (thin plate); 2) orthoscopic interference image; 3) conoscopic interference image

The calculation of photorealistic images which can be observed in reality is a major

2010 Mathematics Subject Classification: 65Z05, 68-04, 78-04.

Key words and Phrases: Lens, Ray tracing, Polarization, Interference.

problem of photorealistic computer graphics. A renderer is a program providing calculation (rendering) of images of 3D scenes on the basis of descriptions of the camera, geometry, and optical properties of objects and sources.

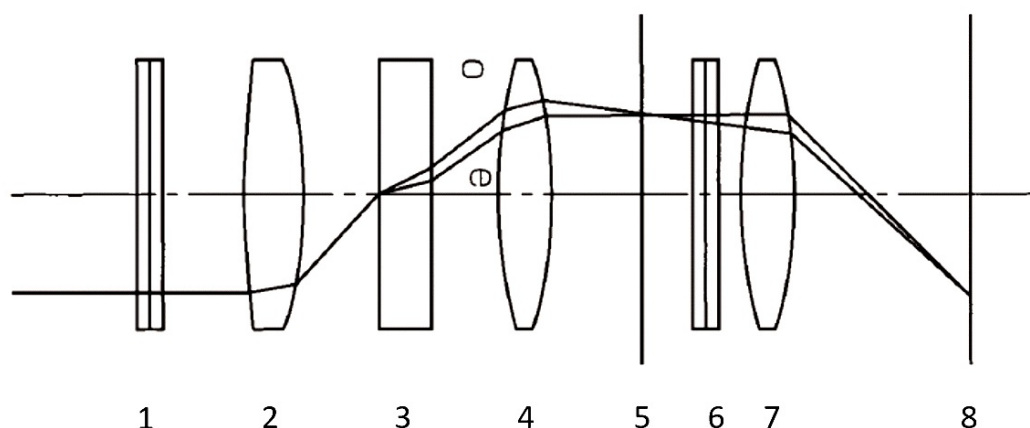


Figure 2. Optical scheme of a typical polarizing microscope

Fig. 2 shows a scheme of a polarizing (petrographic) microscope: 1) polarizer, 2) condenser lenses (not one lens), 3) the sample being investigated, 4) multiple objective lenses, 5) conoscopic image plane, 6) analyzer, 7) Bertrand lens and, possibly, ocular lenses, and 8) plane of the final picture. The solid bifurcating line shows linearly polarized light rays for an optically uniaxial crystal: o – ordinary ray, e – extraordinary ray. Bloss⁴ called the microscope an optical tube. The point is that the PM can be considered as a conveyor of lenses processing light rays: starting from the polarizer, linearly polarized light rays propagate along the conveyor. There is more than one lens, and all lenses are different. Note that PM lenses are made of good glass having no inclusions or scattering elements, which is optically homogeneous, without striae, bubbles, and birefringence (that is, without stresses)⁵. The lens operation is simulated in a *visible range* of electromagnetic waves, 380–780 nm. In the model, it is sufficient to take into account the following parameters determined by the glass type:

1. Spectrum of the refraction index.
2. Spectrum of the linear attenuation coefficient of glass.

Thus, our task is to develop or adapt a computer model of a lens that, in an optically isotropic medium, can provide physically correct processing of incident rays of linearly polarized light and provides a necessary and sufficient basis for the calculation of orthoscopic and conoscopic interference images.

Let us construct a computer model of a PM in the form of an application program (AP) in which interference images are calculated by a special photorealistic renderer.

2 OBJECT MATERIAL IN COMPUTER GRAPHICS

In systems of virtual and augmented reality, computer models of real world objects are widely used. In computer graphics, the parameters of a scene object describing its interaction with light are denoted by the concept of *object material*. Historically, renderers have been developed for specific classes of materials and based on a corresponding mathematical model

of interaction of the objects with light, and in the case of ray tracing, with light rays. It is evident that one and the same renderer can calculate an image on the basis of mathematical models of various physical correctness. As a rule, the choice of a renderer is determined by an application, since the more physically correct is the model, the more calculation time is needed. As examples, Figs. 3 and 4 present variants of calculated images versus the physical correctness of both the object material specified in the scene description and the mathematical model used by the renderer for the calculation.

Consider an RGB-model that originates from the so-called three-stimulus theory of psycho-physiological perception of light as a mixture of primary colors: R – red, G – green, and B – blue. The most important in this model is that even our household display devices (TV sets, computer monitors, and smartphone screens) are based on this theory. It should be noted that to date this model has been mostly used in games, design, virtual reality systems, etc.

It is well-known that light is the totality of electromagnetic radiation waves and is of spectral nature: any colors are spectra of the visible wave range. Let us create a simple scene with a colored plane, a lens, and a point light source. Fig. 3 (left) shows an image calculated on the basis of the RGB-model using the most popular Whitted algorithm^{6,7}. This image is rather realistic, although we have to use *the same refraction coefficient value* of glass for all three components of light. Assume now that we know the spectral characteristics of the light source and the lens glass. We select the CIE Standard Illuminant D65⁸ and the N-SF11 glass⁹. Then we can use spectral rendering⁷, when the refraction coefficients (or refraction indices) are different for different wavelengths, using the same Whitted algorithm, and obtain a mean image as in Fig. 3. The differences are visible by a naked eye. If the spectrum is represented by nSp samples, we need to construct nSp times more tracing trees. This will increase considerably the calculation time; in our case $nSp = 81$. Fig. 3 (right) shows the difference between the two previous images. One can see that the spectral model of light and color that is more correct physically describes more effects of the interaction of light with transparent surfaces. This is especially clear when going from one color to another. It is reasonable to conclude that the more correct model needs more calculation time. Nevertheless, the simpler model gives a sufficiently correct image. For instance, a black spot is present in all images, and it shows an effect of total internal reflection.

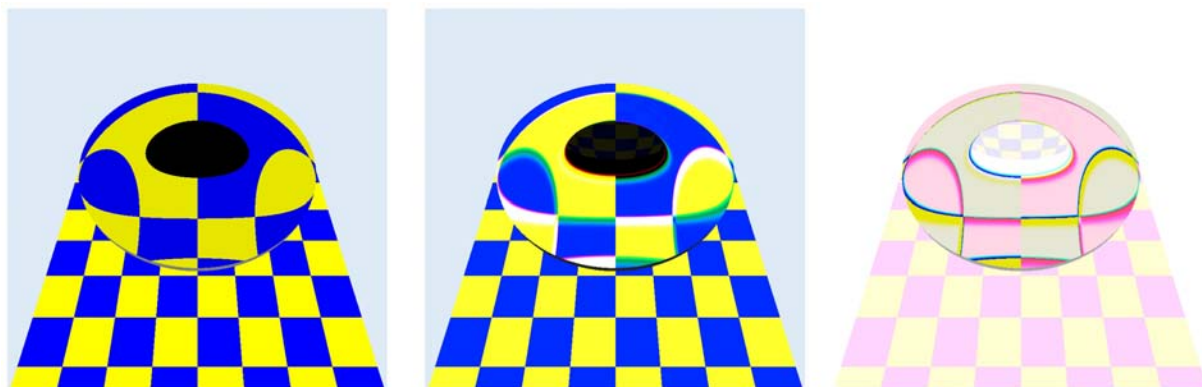


Figure 3. Calculated mages (from left to right): RGB-model, spectral model, difference

Let us consider in more detail the behavior of light at the interaction with boundaries of transparent media. Light can be natural (unpolarized), partially polarized, and polarized³. The well-known widely used programs of rendering are designed only for unpolarized light. It is physically correct to calculate the intensity change using the Fresnel formulas³, when at the interface between the media unpolarized light is divided into two linearly polarized components. Schlick¹⁰ proposed an approximation of the Fresnel approach to calculate the intensities of the generated (reflected and refracted) rays with respect to a ray incident on the boundary between two transparent media. Let us modernize the scene as follows: instead of a point source, a plane is used as a source of diffused unpolarized illumination in the scene. According to our experiments, the Schlick approximation allows a twofold acceleration of this calculation. Fig. 4 shows images obtained using the Schlick approximation and the Fresnel formulas. The negative difference between these images shows that the approximation has a certain error.

Although the four images were obtained with different degrees of physical correctness in specifying the lens material and the mathematical model of the interaction of light rays with objects, all of them show a recognizable scene. The question is to reduce the calculation time and obtain the needed degree of realism.

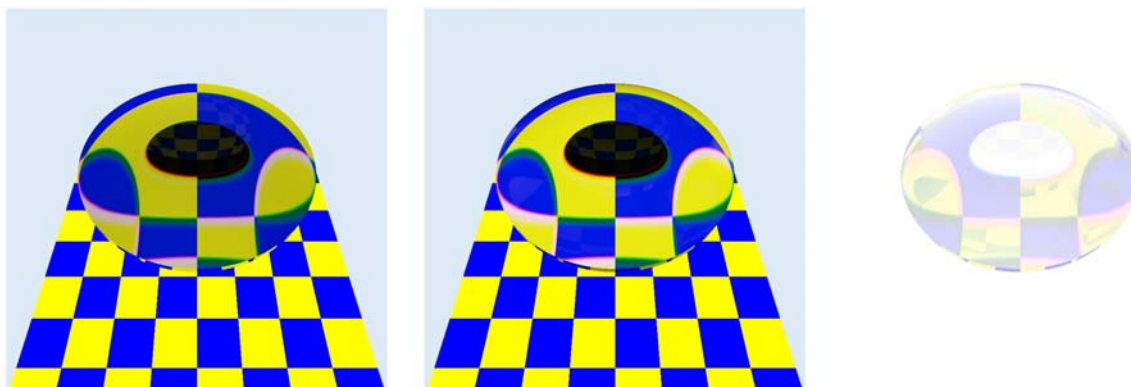


Figure 4. Images from left to right: calculation on the basis of Schlick approximation, calculation by Fresnel, negative difference

The available widely used renderers can rather realistically calculate images of the real world not taking into account light polarization. This is reasonable, since in real scenes a ray is, as a rule, in a polarized state only for a short time and the degree of its polarization decreases when it encounters the next object of the scene. Nevertheless, computer simulation of a photocamera with a polarization filter using these programs is impossible. For instance, images like those presented in Internet¹¹ cannot be calculated.

For further increase in the physical correctness of specifying the object material or the mathematical model used by a renderer for the calculation, it is necessary to take into account light polarization. In some papers, algorithms have been developed for the calculation of images of objects from anisotropic materials demonstrating birefringence^{1 12 13}. This model called for more detailed specification of the material. A more complicated mathematical model of the interaction of a light ray with objects will make it possible to take into account polarization. This will allow simulating the operation of a photocamera with a polarization filter, as well as calculating the ray optical path and taking into account the ray coherence:

visualizing the interference.

3 LIGHT RAY MODEL WITH POLARIZATION

An approach to the evolution of light representation, that is, *information loading* (in what follows we simply say loading) on a mathematical ray can be found in¹⁴. The approach presently being used in practice is as follows: The loading has one intensity value for an individual light wavelength (at spectral rendering) or three values at RGB-rendering. It is assumed that the ray transfers unpolarized light between the points of interaction of the ray with the scene surfaces. It was shown in the previous section that even a local calculation (by Schlick or Fresnel) on the basis of polarization provides a general refinement of the picture. All this reasoning about the calculation speed can be used in a situation when one must choose visual perception of calculated pictures for “ordinary” scenes. The situation is different in obtaining interference images by a polarization microscope, when one cannot use any approximations accelerating the calculation or phenomenological models of light reflection, such as the Phong model⁹. The rays must reach the camera when the state of polarization of light being transferred is exactly specified. This requirement is not very critical in everyday life. However, it is of primary importance in displaying optically anisotropic objects or media, for instance, observing crystals with a PM. Few studies have been devoted to this direction of research in computer graphics (major references can be found in^{2 12}). For the state of the art in computer graphics with light interference, see².

A lens is included in many optical devices, and a polarizing microscope can be represented as a conveyor for light flow processing with lenses, or an analog of a tube with lenses-partitions⁴. The problem of the major application cannot be solved at the level of a computer model of the lens. Hence, its major task is to correctly transfer incoming rays. That is, the intensity, phase, and the state of polarization must be taken into account. In our consideration of possible applications of the lens, we took into account the possibility of obtaining interference images in the main program, a computer model of a PM. At this stage, questions of diffraction have not been considered.

4 LENS MODELS

The following types of geometrical models of lenses have been developed to date: classical thin lenses¹⁵, thick lenses¹⁶, and other specific models. In these models, a lens is considered as an object that transforms the light passing through it mostly according to the laws of geometrical optics. Nevertheless, in solving an application-specific problem the needed information is, as a rule, incomplete and contained in various literature sources, or the required parameters of the lens are not specified exhaustively. Our major goal is to consider in computer graphics a geometrical model of a lens described by a set of surfaces bounding a medium with specific optical characteristics representing a sub-scene of a more general 3D scene.

A similar traditional representation is used in high-power design systems of optical devices, such as ASAP¹⁷, TracePro¹⁸, and others. Many of the questions we had to solve had already been solved within the framework of such systems, for instance, the interaction of a polarized light ray with the boundary of optically isotropic transparent media [6]. These systems cannot be adapted for the following reasons: *they are closed, very cumbersome, and expensive. They are designed for the construction (geometrical and optical simulation) and*

*calculation of ray paths and light rays in a complex optical system, and estimation of the ray parameters at one or another scene point, **but not for obtaining photorealistic images**.* For instance, TracePro has an operation mode of calculation of photorealistic images, but ignores specified polarization and light dispersion in this mode. Nevertheless, ASAP and TracePro in a simulation mode can be very useful in tuning-up of individual optical objects of a 3D scene owing to the physically correct algorithms for constructing specified rays and analyzing their optical characteristics.

In 3D graphical editors, one can create a lens as a scene object, and place it in a realistic visualization system, i.e. renderer. Note that in the description of a 3D scene passed to corresponding renderers the lens description will be presented as a set of its surfaces. That is, the surface primitives describing the lens will be included along with the primitives describing other objects of the scene. Paper¹⁹ presents an illustrative example of presenting lenses (and whole objectives) in computer graphics. Here the lens is simply a part of a scene that is not denoted specifically. It is described by a set of surfaces with specific optical properties. To obtain an image, ray tracing is used: a tree (or a path) is constructed whose nodes correspond to the points of scene surfaces where reflection and refraction of light take place. That is, the possible branching generates no more than two descendants. In another original approach²⁰, light polarization is also ignored, and a rather efficient simulation of the behavior of real lenses and optical systems to calculate images using Monte Carlo ray tracing is proposed.

A major argument against using any renderers (both free and commercial) is that they do not take into account light polarization at ray tracing.

5 PROBLEM STATEMENT

Since a PM is assumed to be an optical conveyor and our problem is to develop a software model of a lens (SML) such that its description be sufficiently traditional for computer graphics:

- Lens is considered as an individual indivisible object, that is, a primitive of a 3D scene. The construction of a photorealistic renderer must allow us to introduce new primitives, for instance, Nvidia OptiX Ray Tracing Engine^{21 22}.
- No special limitations are imposed on the geometrical shape of a lens. In the present paper, a lens is a simply connected object made of an optically isotropic material with whose boundary one can calculate the point of intersection with a ray. The normal is determined at each point of the boundary.
- SML must provide physically correct processing of incoming rays of completely linearly polarized light. At the SML output, rays of completely linearly polarized light are also generated. Processing of outgoing rays is a task of calling an application program AP. A wave of natural (unpolarized) light of intensity I is equivalent to two independent linearly polarized waves³. The intensity of each wave is $I/2$. The electric vectors oscillate in two mutually perpendicular planes that are normal to the direction of propagation – the direction of the ray, taking into account the fact that the medium is optically isotropic (e.g. glass, air, oil). Partially polarized light can be presented in the form of the sum of completely unpolarized and completely polarized parts that do not depend on each other. It follows from these assumptions that the lens module works only with rays of polarized light, and

the use is not limited. It is expected that these facts are taken into account in the AP, although an additional module (SML envelope) to solve problems of incomplete polarization of incoming light rays can be created.

- Owing to its simplicity, linear polarization is represented explicitly. For polarization, no presentations like coherence matrices, Jones or Stokes vectors, or Mueller matrices are used^{3 23}, and the AP developers have some freedom of choice.
- SML must correctly calculate the phase (optical path) and control the coherence of outgoing light rays. This is very important in the calculation of interference images.
- Both forward and backward ray tracing types are used in the renderers. SML must operate correctly in renderers of both types.

6 LIGHT, RAY, POLARIZATION, TRANSFORMATIONS

In our problem, we consider quasi-monochromatic waves represented by monochromatic light rays. It is the task of the AP to present light as a **set** of monochromatic rays. The SML works with one input ray of completely linearly polarized light. A *light ray* of any degree of polarization is presented¹⁴ as a structure called $Ray = \langle R, Payload \rangle$, where $R = \langle P, \mathbf{dir} \rangle$ is a mathematical ray with an initial point P and a direction \mathbf{dir} , $\|\mathbf{dir}\|=1$, and *Payload* (loading) is the information transferred by the ray. When developing the SML, the following **presentation of information loading** is used:

$$Payload = \langle \lambda, I(\lambda), \mathbf{v}_p, \delta, O_p, ch \rangle,$$

where λ is the wave length in vacuum, $I(\lambda)$ is the intensity transferred by the ray, \mathbf{v}_p is the vector of oscillations of the wave electric component in a plane perpendicular to the ray direction \mathbf{dir} , δ is the jumpwise phase shift (see below), O_p is the optical path, and ch is a source identifier used by the AP to determine the coherence of rays coming to some point. ch is given in the AP, and the SML only passes this value to the generated rays. The electric component of a light wave of linearly polarized light can be represented in the following form³:

$$\mathbf{E} = \mathbf{E}_0 \cdot e^{-i(\omega t - \varphi)},$$

where \mathbf{E}_0 is the amplitude, ω is the frequency, and φ is the phase. We rewrite

$$\mathbf{E} = \left(E_0 \cdot e^{-i(\omega t - \varphi)} \right) \mathbf{v}_p, \quad E_0 = \|\mathbf{E}_0\|, \quad \|\mathbf{v}_p\| = 1.$$

Then the system of coordinates for specification of linear polarization need not be stored. Since it is well-known that the intensity is proportional to the squared amplitude³, we believe that it is more practical to store the intensity in the ray loading.

Let a **transformation** G denote the use of the Fresnel formulas at branching points – intersections of a ray with the boundary between transparent media and a calculation of the phase change when the path is inside the lens. Since the needed formulas are in different parts of the various optics books^{3 15 24}, here we put them together.

Let a completely linearly polarized ray with electric vector \mathbf{E} fall on the interface of two transparent media. Here α is the angle of incidence, γ is the angle of refraction, β is the Brewster angle³, that is, $\operatorname{tg} \beta = n_2/n_1$, when n_1 is the refraction index of the medium from which the ray comes, and n_2 is the refraction index of the medium on whose boundary it falls, $n_{21} = n_2/n_1$. Let us decompose \mathbf{E} into two components: $\mathbf{E} = \mathbf{E}_{\parallel} + \mathbf{E}_{\perp}$. One component lies in the plane of incidence, and the other one is perpendicular to it. Let us use a similar decomposition for the reflected wave vectors, $\mathbf{E}_r = \mathbf{E}_{r\parallel} + \mathbf{E}_{r\perp}$, and refracted wave vectors, $\mathbf{E}_t = \mathbf{E}_{t\parallel} + \mathbf{E}_{t\perp}$. With the Fresnel formulas one can find the following relations between the *amplitudes* of the incident, reflected, and refracted waves:

$$\begin{aligned} \frac{E_{r\perp}}{E_{\perp}} &= -\frac{\sin(\alpha - \gamma)}{\sin(\alpha + \gamma)}, \\ \frac{E_{t\perp}}{E_{\perp}} &= \frac{2 \cos \alpha \sin \gamma}{\sin(\alpha + \gamma)}, \\ \frac{E_{r\parallel}}{E_{\parallel}} &= \frac{\operatorname{tg}(\alpha - \gamma)}{\operatorname{tg}(\alpha + \gamma)}, \\ \frac{E_{t\parallel}}{E_{\parallel}} &= \frac{2 \cos \alpha \sin \gamma}{\sin(\alpha + \gamma) \cos(\alpha - \gamma)}. \end{aligned}$$

We must note that the reflected and refracted rays are completely linearly polarized. Only their polarization planes, as a rule, are rotated with respect to the plane of polarization of the incident ray. In the case of normal incidence of a ray, there is no division into components, and the formulas are transformed as follows:

$$\begin{aligned} \frac{E_r}{E} &= \frac{n_1 - n_2}{n_1 + n_2}, \\ \frac{E_t}{E} &= \frac{2n_1}{n_1 + n_2}. \end{aligned}$$

One may consider the case of total internal reflection, which needs special investigation:

$$\begin{aligned} E_{r\parallel} &= E_{\parallel}, \\ E_{r\perp} &= E_{\perp}. \end{aligned}$$

Calculation of the optical path and phase jumps. The optical path for a distance of length S in a medium with refraction index n is calculated by the formula² $O_p = nS$. The continuous phase variation of a wave of length λ as it passes this path is $2\pi \cdot O_p / \lambda$. However, at branching points (at the boundaries) there may be jumpwise phase changes^{2 24}. The rules of such changes are as follows:

- under refraction, the incident and refracted rays are co-phased.
- E_{\perp} : under reflection, the phase changes by $-\pi$ if $n_1 < n_2$.

- E_{\parallel} : under reflection, the phase changes by $-\pi$, if the following condition is satisfied:

$$(n_1 < n_2 \wedge \alpha < \beta) \vee (n_1 > n_2 \wedge \alpha > \beta).$$

The phase shift, π , is taken into account by changing the direction of the vector \mathbf{v}_p for the required component. Then a linearly polarized ray is formed after reflection. The phase change of this ray will be fully determined by the optical path, and the parameter δ is zero in the representation *Payload*. For the case of total internal reflection, we use more complicated expressions for the phase jump²⁴:

$$\delta_{\parallel} = 2 \operatorname{arctg} \left(\frac{\sqrt{\sin^2 \alpha - n_{21}^2}}{n_{21}^2 \cos \alpha} \right) - \pi,$$

$$\delta_{\perp} = 2 \operatorname{arctg} \left(\frac{\sqrt{\sin^2 \alpha - n_{21}^2}}{\cos \alpha} \right).$$

Since in this situation the reflected ray is elliptically polarized, to use the principle “work only with linearly polarized light” we form two linearly polarized beams corresponding to the reflected beam components $E_{r\parallel}$ and $E_{r\perp}$ with the needed values of the parameter δ in the *Payload*.

The AP should calculate the full phase change in the lens by the formula

$$d\varphi = 2\pi \cdot \frac{O_p}{\lambda} - \delta.$$

Absorption. The Beer–Lambert–Bouguer law¹⁵: the intensity of monochromatic light decreases exponentially with a distance in the material, that is

$$I(x) = I_0 e^{-\mu x},$$

where I_0 is an initial intensity, μ is a linear attenuation coefficient, x is a distance.

7 FUNCTIONING OF THE LENS MODULE

Consider the process of interaction of a lens with the surrounding medium under ray tracing. Fig. 5 shows how a ray falling on the lens generates a tree of rays. One of the requirements on the development of the software module was that it must be universal: that is, it must be a module used in the programs of both *forward* and *backward* ray tracing. Thus, an SML in the form of an independent software module with specified interface is controlled by an application program which “decides” what outgoing rays to return and what rays to ignore.

An important parameter of the backward tracing algorithms is the rendering tree depth: the smaller is the depth, the faster is the calculation. It is intuitively obvious that the farther is a leaf from the tree root, the less is the contribution of this leaf to the total energy, since it decreases at each reflection or refraction. On the other hand, if we decrease the depth in order to increase the calculation speed, that is, if we cut off the branches, starting from some level

we can lose a leaf corresponding to a very powerful source.

The AP gives the SML a limiting depth of tracing inside the lens, dT . In the most general case the SML, using the data on the incident ray and the given depth, processes all outgoing rays. In Fig. 5 $dT = 5$. The SML calls the functions specified in the AP for each of the generated rays and requests information about all the rays. The tracing depth parameter, dT , is important: its value already obtained in the tree construction, can be used to limit branching in the lens. Some peculiarities of applications are: in the polariscope, in the case of forward tracing the reflected ray directed outward, as well as the rays leaving the lens and moving towards the incident ray can be ignored, since they will not participate in the formation of the target image. Proper control modes are specified for the lens module from an AP solving a global task.

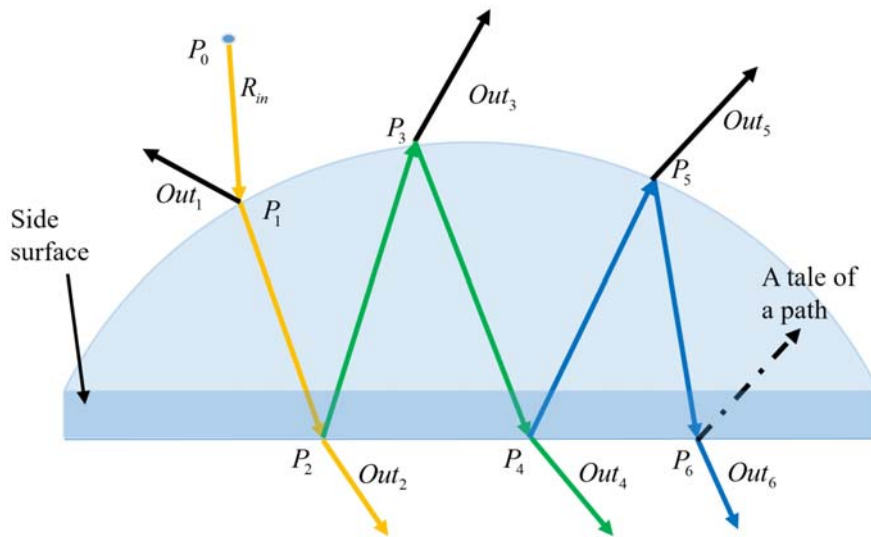


Figure 5. Subtree of rays of depth 5 in the lens. Notation: R_{in} – incident ray, P_* – treetops (nodes), Out_* – rays coming from the lens

At *forward tracing* a ray in a *Payload* representation with all available parameters is supplied to the SML inlet. At all tracing tree nodes, a single-type transformation G (see above) will be used. It generates two rays in the same representation, except for the case of total internal reflection with no refracted ray. Up to the lens inlet point P_1 , the phase and the optical path are controlled by the AP, and from this point to any of the outlets, they are controlled by the SML.

At *backward tracing*, at the tree construction pass, a ray is also supplied to the SML inlet for which only one wavelength, λ , is specified in the *Payload* representation. After all outgoing rays Out_* are transferred from the SML to the AP, the lens module waits so long as the AP completes the construction of the tracing tree, starts the second pass of energy collection, and returns to the SML the response information *about every ray* generated by the lens. There may be not one but several completely linearly polarized rays depending on the value of dT – “a bundle of rays”. At its backward passage, the SML processes each of these rays at the internal lens nodes using the transformation G .

Forward tracing (rays from the source to the scene). Forward ray tracing in the calculation of scene images is as follows: initial rays are generated on light sources, and their paths resulting from reflection and refraction from the scene surfaces are traced until they reach the image plane (screen) or another recording surface. Forward tracing, because of its complexity, is almost not used to obtain images in computer graphics. It is often used as an auxiliary stage for primary distribution of illumination along the scene surfaces, for instance, in the form of illumination maps²⁵. The microscope presents a specific scene. All paths from the source outside the optical tube can often be ignored. This makes it possible to gather the rays reaching the ocular in a reasonable time. The problem of estimating the influence of ambient light (if it exists) incident from outside onto the specimen must be solved additionally. On the whole, forward tracing can be used to estimate the various effects and parameters of the optical elements. For instance, by forward tracing one can calculate the degree and zone of vignetting of the objective.

Backward tracing (from the observer to the scene). A classic example is the Whitted backward recursive ray tracing algorithm⁶. This algorithm is most often used to calculate realistic images. It consists of two stages. At the first stage, the camera issues a ray through a screen pixel. A rendering tree whose leaves are the sources is constructed. A part of the branches is cut off when the ray goes beyond the scene or when a given tree depth is reached. At the second stage, the energy going to the camera with the initial ray is collected. For this, the tree is passed in the opposite direction (i.e. from the leaves to the root), and the energy of the descendants is combined at every node and passed to the parent.

The operation of the tracer (a renderer on the basis of ray tracing) has been described so far similarly to that of the Whitted tracer in the calculation of photorealistic images of 3D scenes. In the Whitted algorithm, the resulting generalized energy collected from all descendants of this node (i.e. one or three numbers in case of color) is passed to the parent level. In our case, the state of polarization of each portion of energy obtained from the corresponding leaf-source is traced. In backward tracing, as the lens module ignores the energy gathering phase, it provides a whole “bundle”, that is, a list of the resulting “returned” rays: minimum one for each of the depths, $1, 2, \dots, dT$. Why not combine this in one ray? We used the principle of separation of functions: the AP solves the main problem, and the SML transforms polarized rays passing through the lens. For instance, by solving a problem of calculation of an interference image, the AP determines the ray coherence and calculates the intensity in a pixel using the coming intensities, phases, and coherence of the sources of the entire family of rays at backward passing of the tracing tree.

Special cases of processing. As mentioned above, the SML being developed is oriented to the processing of linearly polarized light rays. Under *total internal reflection* (TIR), both components of decomposition of incident radiation \mathbf{E} acquire, when reflected, different phase jumps that are not multiples of π . That is, the reflected ray becomes elliptically polarized. The operation modes of the SML are as follows:

1. Termination of further path tracing. The TIR situation is reported to the AP. This agrees with the major function of a lens in an optical tube⁴.
2. The AP under the SML initialization sets a special procedure of response to the TIR `TIR_callback`. The SML passes both rays and all their characteristics to `TIR_callback`. Once the information is processed by the AP, it returns the control to the SML, which continues tracing these linearly polarized rays individually. Thus, at a large tracing

depth dT and with several successive situations of TIR, the number of rays being traced increases as a power of two. This mode can be implemented in practice. For example, one can implement *Fresnel rhomb*³ transforming linearly polarized light into light with circular polarization using TIR.

Lens ledge. The above description is valid for any geometrical shape of a lens. Some surfaces of the boundary are used to mount the lens on a microscope. These are called lateral surfaces or ledges. The optical properties of the ledges are typically unknown. It can be assumed with high probability that the light incident on lateral surfaces is strongly depolarized and diffusively scattered. The SML processes a ledge as in the case of TIR: it terminates further tracing of a given ray or calls the procedure `ledge_callback` for the AP to process the ray incidence of the ledge.

Absorption. If the attenuation coefficient of a lens glass differs from 0 the SML for each path calculates the intensity decrease along a passed distance inside a lens using the Beer–Lambert–Bouguer law.

8 EXPERIMENTS

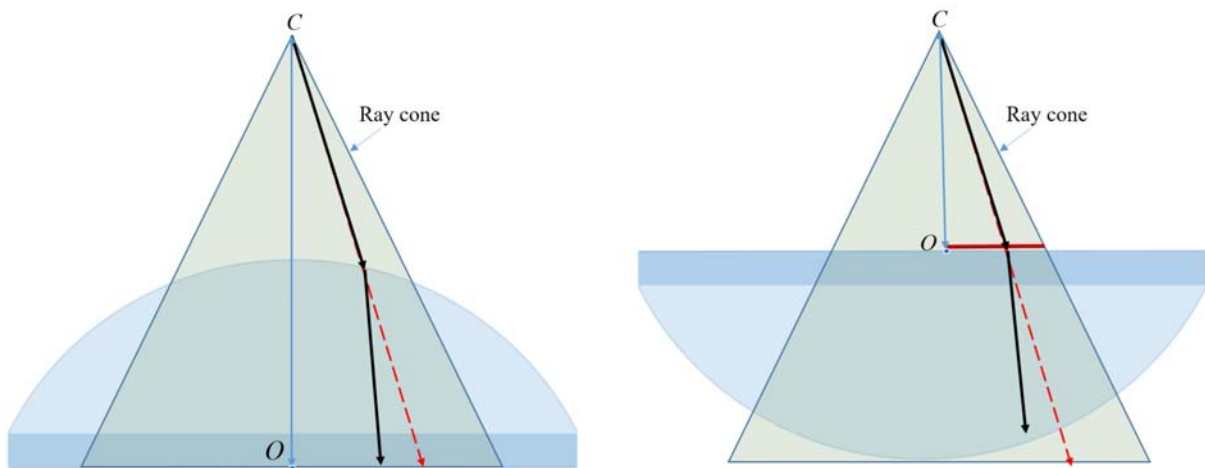


Figure 6. Cone of rays

From a geometrical point of view, computer models of lenses or objectives can be considered to be bodies of revolution. Therefore, many characteristics of lenses can be calculated when rays fall only on a half of a cross-section. Fig. 6 shows two locations of a lens relative to a camera C . In the following experiments we define a *cone of rays* specifying an initial direction of tracing for the further path not to go through the lens ledge (the cylindrical part of the boundary), at least to the point P_6 in Fig. 5. It is assumed that when a polarized ray hits the ledge it ceases to be polarized. At least we have not found any standards for the optical parameters of the mounting zones. Hence, such a ray is not suited for our purposes of constructing interference images. Note that the observer sees usually only a central part of the image. In Fig. 6, the red dashed vector shows one of the directions inside the cone, the black one denotes the path starting from this direction, O is the reference point, OC is the camera height, R_C is the radius of the cone at level O (red segment), R_S is the radius of the sphere, and R_L is the radius of the lens.

From Table 1 one can estimate the internal radius in the lens in which the light paths being traced do not have parasitic reflections from the lateral surfaces. After the point P_6 in Fig. 5, the cutoff tracing tree branches make an insignificant contribution to the resulting intensity and polarization, as expected. The user, making a similar experiment, can choose a lens geometry for his purposes.

Left scene			Right scene		
R_s	η at $OC = 12.5$	η at $OC = \infty$	R_s	η at $OC = 8.5$	η at $OC = \infty$
8	100%	100%	8	91.5%	100%
6	93.5%	86%	6	100%	94.5%
4.3	100%	100%	4.3	39.5%	40%

Table 1. The ratio $\eta = \frac{R_C}{R_L}$, $R_L = 4.25$, ledge height: 0.3

Left scene, $OC = 12.5$			Right scene, $OC = 8.5$		
R_s	η_4	η_6	R_s	η_4	η_6
8	0.00597603	8.25118e-05	8	0.0060276	0.00129178
6	0.00585356	0.0620637	6	0.052925	0.0496469
4.3	0.00552146	3.12734e-05	4.3	0.0711181	0.00488492
Left scene, $OC = \infty$			Right scene, $OC = \infty$		
R_s	η_4	η_6	R_s	η_4	η_6
8	0.00597478	0.000178105	8	0.00756504	0.0104476
6	0.00585355	0.033183	6	0.0648151	0.028809
4.3	0.00630298	3.12795e-05	4.3	0.0727728	0.00506117

Table 2. Intensity parts Out_4 and Out_6 .

Consider Fig. 5 and estimate the contributions I_4 and I_6 to a screen pixel (point P_0) by tree branches denoted by Out_4 and Out_6 . These contributions will be reduced in the lens until they get into the pixel. Note that in the test AP it is assumed that in all three rays (Out_2 , Out_4 , and Out_6) at backward tracing the lens has the same intensity, I_0 . We consider only the paths starting in the cone calculated before and not hitting the ledge up to the point P_6 . Let the total intensity I come from the three branches, calculate $\eta_4 = \frac{I_4}{I}$, $\eta_6 = \frac{I_6}{I}$, and determine their **maximum values** for all paths being considered in the cone, and summarize them in Table 2. Analysis of values in Table 2 shows that the branches Out_4 and Out_6 make the greatest contribution, up to 8%, for the right scene at $\eta = 39.5\%$, $OC = 8.5$, and $R_s = 4.3$. Recalculating the experimental data for a cone with $\eta = 20\%$, we obtain $\eta_4 = 0.00744892$, $\eta_6 = 0.000552532$, which is less than 1%. The AP developer, performing similar experiments, can use another criterion for limiting the ray cone, in order to get the accuracy needed for a

specific lens or choose another geometry of the lens.

9 CONCLUSIONS

In this paper, a software implementation of a computer model of an SML designed only for linearly polarized light is presented. This limitation can be easily removed, since an unpolarized or partially polarized light ray can be represented as the sum of linearly polarized rays with corresponding intensities.

The software model of a lens has been implemented to meet the following major requirements:

- The SML can be included into renderers based on ray tracing. This module can operate both at forward and backward recursive tracings of rays like the Whitted algorithm.
- The calculation is made in a spectral format when all parameters are calculated for each wavelength of the spectrum separately in a visible range of electromagnetic waves of 380–780 nm.
- Every incoming polarized ray is characterized by initial values of the state of polarization, phase, intensity, and light source identification.
- During processing, the SML can generate several outgoing rays. This is due to the fact that a ray getting into the lens can be re-reflected several times, and each time this can result in an outgoing (refracted) ray. The AP can specify a limiting number of internal re-reflections in the lens, which allows controlling the calculation speed and accuracy.
- Every outgoing ray has an identifier of the generating ray source, as well as recalculated values of the state of polarization, phase, and intensity.
- Every outgoing ray is characterized by complete linear polarization. If there are several polarized rays in a given direction, they are not combined and are generated in the form of a “bundle”.
- To describe linear polarization, it is sufficient to have only one vector in addition to a ray direction. This does not impose any requirements on the presentation of polarized light. Thus, an application using the SML decides how to treat the information about every such “bundle”.
- When a ray hits a lens lateral surfaces or in the case of total internal reflection further ray tracing is terminated, but the application program can process these cases itself.

Our nearest plans are to develop a computer model of a simplified model of a polarizing microscope including lenses built on the base of the SML.

This work was supported by the RFBR (project no. 16-07-00762).

REFERENCES

- [1] V.A. Debelov, D.S. Kozlov, “A Local Model of Light Interaction with Transparent Crystalline Media”, *IEEE Transactions on Visualization and Computer Graphics*, **19**(8), 1274–1287 (2013).
- [2] V.A. Debelov, “Interferentsiya sveta, izotropnye prozrachnye ob’ekty, trassirovka luchej”, *Proceedings of 25-th International Conference on Computer Graphics and Vision, September 22-25, Protvino, Moscow region, Russia*, 168-173 (2015). (in Russian)
- [3] M. Born, E. Wolf, *Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light*, Cambridge Univ. Press, (1980).

- [4] F.D. Bloss, *Introduction to the Methods of Optical Crystallography*, NY: Holt, Rinehart and Winston, (1961).
- [5] S.M. Latyev, G.V. Egorov, et al, “Konstruirivanie tipovykh opticheskikh detalei i sborochnykh edinits opticheskikh priborov”, URL: http://de.ifmo.ru/bk_netra/page.php?tutindex=46&index=5. (in Russian)
- [6] T. Whitted, “An Improved Illumination Model for Shaded Display”, *Commun. ACM*, **23**(6), 343-349, (1980).
- [7] J.D. Foley, A. van Dam, S.K. Feiner, J. Hughes, *Computer Graphics: Principles and Practice (2nd ed.)*, Addison-Wesley, (1990).
- [8] Colorimetry, CIE Publication, 2nd ed., **15**, (1986).
- [8] Optical Glass, URL: <https://refractiveindex.info/download/data/2015/schott-optical-glass-collection-datasheets-july-2015-us.pdf>.
- [9] C. Schlick, “An Inexpensive BRDF Model for Physically-Based Rendering”, *Computer Graphics Forum*, **13**(3), 233-246 (1994).
- [11] https://en.wikipedia.org/wiki/Brewster%27s_angle.
- [12] A. Weidlich, A. Wilkie, “Realistic Rendering of Birefringency in Uniaxial Crystals”, *ACM Transactions on Graphics*, **27**(1), 6(1–12) (2008).
- [13] S.V. Ershov, A.A. Garbul, S.G. Pozdnyakov, V.G. Sokolov, A.G. Voloboy, “Lighting simulation in anisotropic media”, *Mathematica Montisnigri*, **XXXIX**, 42-56 (2017).
- [14] L.F. Vasilyeva, V.A. Debelov, “Evolutsiya modeli lucha sveta dlya renderinga”, *Proceedings of International Conference “Situational centers and the information-analytical system of class 4i for the tasks of monitoring and security SCVRT 1516”*, November 21-25, Puschino, Moscow region, Russia, 178-184 (2016). (in Russian)
- [15] G.S. Landsberg, *Optika*, Moscow: Fizmatlit, (2003). (in Russian)
- [16] W. Heidrich, P. Slusallek, H.-P. Siedel, “An Image-Based Model for Realistic Lens Systems in Interactive Computer Graphics”, *Graphics Interface, Canadian Information Processing Society*, 68-75 (1997).
- [17] ASAP, http://www.breault.com/sites/default/files/knowledge_base/broman0955_primer.pdf.
- [18] TracePro, URL: https://www.lambdare.com/wp-content/uploads/TraceProDownload/TracePro_User_Manual.pdf.
- [19] D.D. Zhdanov, I.S. Potemin, et al, “Metody stohasticheskoi trassirovki luchej v zadache postroeniya izobrazhenii, formiruemykh real’nymi opticheskimi systemami”, *Proceedings of 26-th International Conference on Computer Graphics and Vision, September 19-23, Nizhny Novgorod, Russia*, 29-33 (2016). (in Russian)
- [20] J. Hanika, C. Dachsbacher, “Efficient Monte Carlo rendering with realistic lenses”, *Computer Graphics Forum*, **33**(2), 323–332 (2014).
- [21] NVIDIA® OptiX™ Ray Tracing Engine, URL: <https://developer.nvidia.com/optix>
- [22] V.A. Debelov, “Primenenie NVidia OptiX dlya provedeniya chislennykh eksperimentov”, *Scientific Visualization*, **6**(4), 1-10 (2014). (In Russian)
- [23] E.L. O’Neil, *Introduction to statistical optics*, Addiso-Wesley, (1963).
- [24] V.A. Kizel, *Otrazhenie sveta*, “Nauka”, Moscow, (1973). (in Russian)
- [25] J. Arvo, “Backward Ray Tracing”, *ACM SIGGRAPH’86 Course Notes – Developments in Ray Tracing*, 259-263 (1986).

This paper is an extended version of a report presented at 27-th International Conference on Computer Graphics and Vision (24–28 September 2017, Perm, Russia).

Received November 30, 2017