

Low-Cost Medical Simulation: a Retinal Laser Photocoagulation Simulator

Meseure P.¹, Karpf S.¹, Chaillou C.¹, Dubois P.², Rouland J.F.³

¹ Laboratoire d'Informatique Fondamentale de Lille
Université des Sciences et Technologies de Lille
59655 Villeneuve d'Ascq Cedex, FRANCE
Phone: (33) 20-43-65-57 Fax: (33) 20-43-65-66
E-mail: {meseure, karpf, chaillou}@lil.fr

² CLARC-Laboratoire de Biophysique, CH&U Lille, FRANCE

³ Clinique ophtalmologique, Hopital Huriez, CHU Lille FRANCE

Abstract

Retinal laser photocoagulation has been used for a decade, but no real improvement in results has been observed. The simulator described in this paper, by separating actual practice from apprenticeship, allows one both to learn how to manipulate the equipment and to train oneself to make a diagnosis and to operate. An actual slit-lamp is supplied with sensors measuring all the degrees of freedom of the system. The images are calculated by a PC and are displayed on a miniature screen located inside the binocular. The purpose of this article is to describe the graphical part of the simulator, i.e. the modeling of the lens and the eye, the visualization algorithm and a real-time depth of field simulation. The goal is to synthesize images as close as possible to the actual view.

Key Words: Medical Graphics, Graphics in simulation, Interactive Techniques, Graphics in Education.

Introduction

Retinal laser photocoagulation is a widely spread surgical technique. It has become one of the most important part of the ophthalmic area. The therapeutic effects of photocoagulation are no more to be demonstrated in retinal detachment prevention, diabetic retinopathy, A.R.M.D. (Age Related Macular Degeneration), etc... However, despite the number of installed equipments, no real improvement of complications which should have been solved has been observed. These can, in worst cases, lead to a complete blindness. One has the right to wonder if the apprenticeship technique is not to be reviewed.

A retinal laser photocoagulation simulator provides a good answer to the apprenticeship problem. The interest of such a simulator is the opportunity for the

student to train as often as he wants and to deal with very rare pathologies that he would have had little chance to meet during his relatively short training course. Moreover, it allows ophthalmologists to improve their skill during their continuous education or to prepare critical operations. Finally, it gives the ability to separate actual practice from apprenticeship, and therefore to lower the risks for the patients.

The simulator has been designed as close as possible to the actual operating conditions. The slit-lamp is kept, all the other tools and the patient are simulated. Sensors measure the positions of the different degrees of freedom and send the data to a PC computer which calculates the image to be displayed. Images are displayed in the binocular by means of a miniature screen. In order to synthesize realistic images, a texture mapping of an actual eye and an actual iris is used. Realism is increased by adding a lighting model of the slit-lamp and a blurring generation that simulates the focus.

After a quick state of the art in medical simulators, we present a description of laser photocoagulation, the configuration of the simulator and the list of all the sensors. Next, the model of the eye, the visualization algorithms and a real-time depth of field simulation will be presented.

1 Previous work

Medical simulators have been a recent research area. Compared to driving simulators, the use of medical simulators increases quite slowly. One of the reasons might be the reticence of the surgeons. Another reason is undoubtedly the complexity of the required algorithms within the medical area. Indeed, the surgeon not only visualizes the environment, but also interacts with it. That is why most simulators demand powerful



workstations. [CEB93] study a laparoscopic gall-bladder surgery simulator, and try to generalize their model to other organs. [SBM94] have designed a complete immersive simulator (including sensors, modeling, real-time deformations, visualization and force feedback) allowing operations on the anterior segment of the eye to be simulated. Their goal is to find new operation techniques. Other teams (Chapel-Hill, Georgia) are working on complete virtual environments. New domains are visited like childbirth [BG93], cardiology [SCA94]... Cinquin's team in Grenoble is working on topics like data acquisition, simulation, critical operation preparation and computer assisted intervention in areas like stereostatic neurosurgery, knee surgery, plastic and maxillo-facial surgery, rigid endoscopy [BCCT94]. Some other teams are working on 2D-data acquisition and 3D reconstitution which can be used for teleoperation, for instance [CGRB91] on stereostatic MRI.

Since the equipment required for complete environments is costly, some teams are working on cheaper simulators. For instance, [BDM92] uses images recorded on CD-ROM. [DSCP94] have designed a craniofacial surgery simulator, which intends to predict the result of an operation.

Some areas like observation, laser or endoscopy do not require very costly hardware [GW87][HGW92], because the surgeon watches the organ through a binocular or a screen, and because there is not or few mechanical interactions between the surgeon and the organ. Our simulator belongs to this category, since it can be implemented on a general purpose PC, the vision of the surgeon is limited by the binocular and only laser is used for the therapy (no physical interaction).

2 Retinal Laser Photocoagulation

The goal of laser photocoagulation is to heal retinal diseases. The ophthalmologist beams a high power laser in order to create a barring around the lesion. The equipment includes a binocular, a slit-lamp which lights the fundus, and the laser outfit.

A 3-mirror lens is placed on the patient's eye. It is made of three mirrors, spaced at 120 degree intervals, around a central lens. These mirrors have different inclinations so that the surgeon can observe different areas of the retina.

Figure 1 shows the actual operation and the principle of the simulation. The surgeon looks at the eye through the binocular. The eye is lit by means of a slit located at the back of the apparatus. A control lever enables the user to move the slit-lamp horizontally and vertically. A

micro-manipulator allows the user to choose the location of the laser target, and a control panel controls the settings of the laser (power, fire duration).

The surgeon makes the 3-mirror lens slip around the eye, and looks at the fundus through the central lens or through one of the mirrors. In the same way, the laser beam can reflect onto a mirror before passing through the central lens. A low power laser sets the target site which will be hit by the high power laser beam when the surgeon pushes the fire pedal. About one hundred shots can be fired during an actual operation.

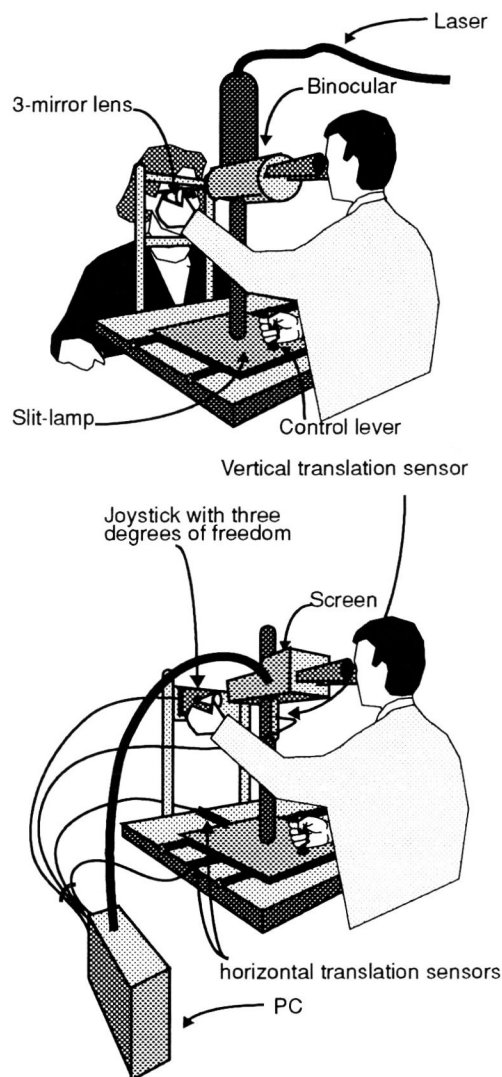


Figure 1: Actual operation vs. simulation



3 Description of the simulator

3.1 Overall design

Two PC computers are used for the simulator. The first one, called the simulation PC, runs the software that synthesizes the images. The algorithms have been implemented in C. This PC includes a conversion interface card which converts all the sensors data and sends them to the PC. Once created, the images are displayed both on a control monitor and on a miniature monitor located in the binocular. At every step, the PC gets the sensors data, calculates the image, and displays it on the two screens. The whole database (fundus, pathologies and fluorographies) is located in this PC.

The second PC, called the master PC, runs the user interface under the Windows environment. A quick and non-exhaustive description of this interface is given in the next section. The master PC sends orders to the simulation PC (also called slave-PC) by means of a serial link. Figure 2 shows the complete configuration.

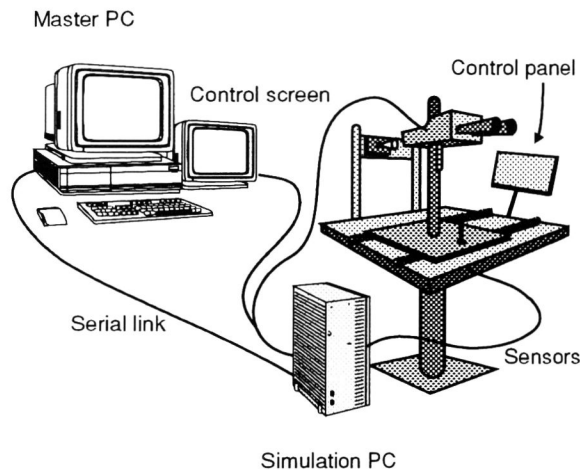


Figure 2: Simulator configuration

The miniature screen is a critical part of the simulator. We first experimented a 320x200 LCD monitor, but the quality of the LCD technology is very far from what ophthalmologists consider as acceptable. We now use a NuVision 1" field sequential display (NuColor technology) built from a monochrome CRT screen (up to 800x600) and color shutters. Such a technology, although much more expensive than the LCD one, gives excellent results, and makes us believe that medical applications cannot be satisfied with low quality displays often found in VR applications.

3.2 User Interface

The user interface [MRDKC95] consists in a succession of explicit menus, which are easy to use, even for people with little knowledge of computers. By typing his name and password, the user lets the software know what kind of user he is, among the three implemented classes. The *system manager* is responsible for the well-running of the simulator and for the database management. The *experts* can create photocoagulation exercises for their students, evaluate them, or prepare critical operations. The *students* can train on their own or inspect their improvement since their last operation simulations.

A simulation of an operation is called a session. Every sessions are recorded and can be reviewed (by means of the simulator) if required by the user, with the help of a video tape-like interface. Only experts can evaluate a student, with the help of some piece of information (number of impacts, location of impacts...) provided by the simulator.

4 Mechanical Features

4.1 Sensors

In our simulator, only the slit-lamp is kept. The patient and the lens are completely simulated. So, what the surgeon sees in the binocular has to be synthesized. The image depends upon the degrees of freedom of the system, which are measured by sensors [DRMKC95]. Here is the list of the degrees of freedom:

- 3 translations (2 horizontal, 1 vertical) of the slit-lamp, which are measured by linear potentiometers.
- 3 rotations (2 around the patient's eye and 1 around the lens axis), which are measured by an analogical 3-axis joystick.
- 2 rotations of the laser mirror, which are measured by an analogical joystick.
- The slit width and spot size are controlled by potentiometers.
- The power of the laser and the firing duration are modified with the help of +/- interrupters located on the control panel.
- The slit height, the magnifying level and the power of the lamp are controlled by means of multi-position interrupters.
- Firing is controlled by a pedal.

The data coming from the analogical and digital sensors are sent to the computer by means of a conversion interface. This card encodes values in 12-bit format. All the controls of the simulator are strictly



identical to those on the actual photocoagulator. This means that location and tactile sensation are the same. The modeling of the 3-mirror lens depends on the joystick, and is described in the next section.

4.2 Lens modeling

A 3-mirror lens is simulated since it is a good generalization of all ophthalmic lenses. It allows the eye to be observed directly or indirectly. The proposed model can therefore be adapted to other types of lenses. The three mirrors enables the surgeon to observe different zones of the eye. It is represented in Figure 3.

- Mirror 2 (low inclination) allows the zones located around the central fundus to be observed.
- Mirror 3 (average inclination) allows the peripheral retina to be observed.
- Mirror 4 (high inclination) allows the regions located just behind the iris to be observed.

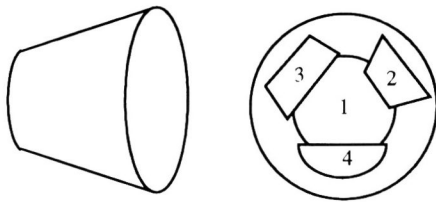


Figure 3: The 3-mirror lens

The 3-mirror lens is modeled by a cone, and the (spherical) central lens is approximated by a planar circle. Each mirror is modeled by a plane (an equation) and four points belonging to the plane, which delimit the mirror edges.

The model must be parameterized by the joystick data, in order to define the orientation of the virtual lens. These data are converted into angles: the 2 rotations around the eye are defined just as the spherical system coordinates (a latitude and a longitude). This is made possible because of the joystick technology, where one of the rotations depends upon the other (another technology, where the two rotations are independent would require more complex computing).

A space transformation matrix (from the local to the global coordinates) is calculated from the angles and is used to calculate the position of the lens elements. This matrix also allows the new equations of the mirrors planes to be determined and therefore, the image of space, seen through the mirrors, to be calculated by an orthogonal symmetry.

5 Eye modeling

5.1 Simplified model of the eye

The eye is modeled by a sphere, the cornea by a half-sphere (see Figure 4). The center of the eye and of the cornea are not the same: the cornea center is translated, along the visual axis, toward the pupil. The iris is modeled by a flat ring, orthogonal to the visual axis. A transformation matrix is calculated, to simulate any movement of the eye. But, this time, the angles are chosen by the computer, depending upon how "nervous" the virtual patient is: random numbers with Gaussian distribution give quite realistic effects.

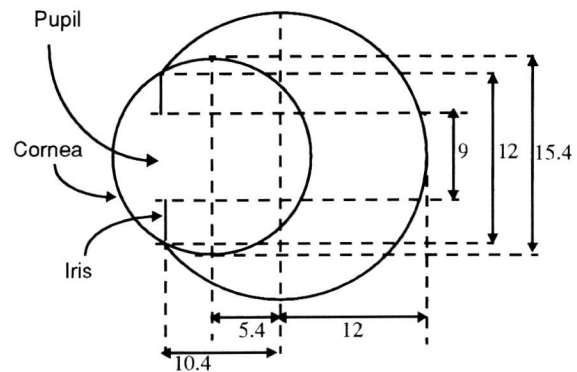


Figure 4: Model of the eye

As far as visualization was concerned, it was very important to let the user see a synthetic eye as realistic as possible. Obviously, the best method to get such a realism was to use photomapping (fundus and iris).

5.2 Fundus texture generation

A complete texture map of an eye has to be created (See Figure 5). At least 20 photographs of an actual fundus are taken through a special lens and placed side by side with a specific software. This software is easy to use: an empty 1024x1024xRGB TGA map is created, which represents a 2D map of the eye, by means of spherical coordinates. Each photograph is digitized with the KODAK photo-CD process. A null color is used in order to mask both photograph edges and badly lit zones. The photograph is then put into the map: it is orthogonally projected onto the eye sphere in order to get the spherical coordinates of each point and then placed into the map. The continuity of the blood vessel is used to find the right place of the photograph in the map. Four parameters enable the user to assume this continuity: the dilatation coefficient, the position in



angles of the center of the photograph and the rotation around it. After placing the 20 photographs, a complete map of the posterior part of the eye is obtained. Unfortunately, because it is almost impossible to take good photographs of the ora serata (area located just behind the iris), a part of the texture has to be drawn (see Figure 5).

All this process is not automated yet. Automatic positioning of the photographs would take a long time because of the number of parameters. A non-linear optimization method could perhaps accelerate the process, provided that the lighting difference between two photographs could be lowered.

The iris is textured too. As it is modeled by a flat ring, one front photograph of an actual eye is enough to build the texture map.

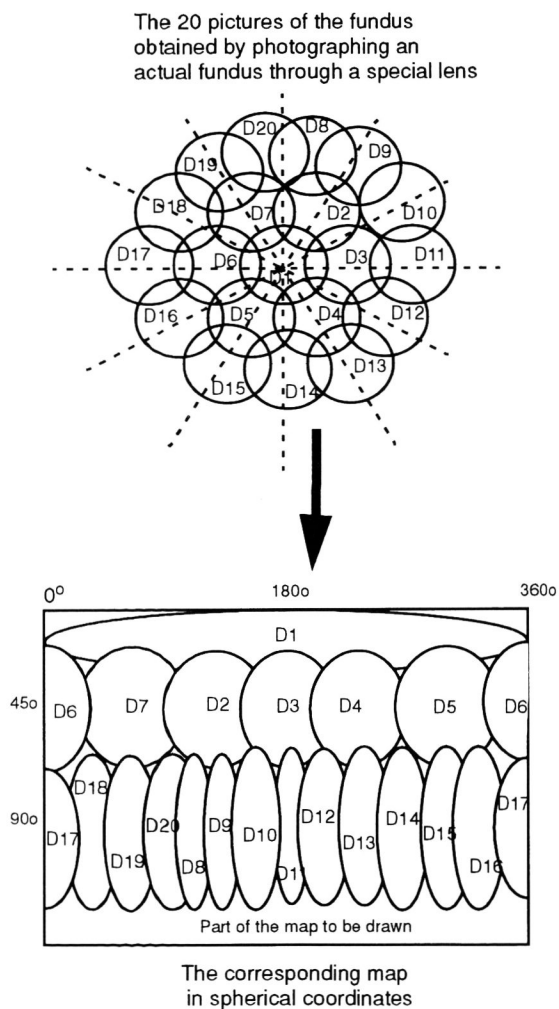


Figure 5: Creation of the fundus texture

5.3 Computer model of the eye

In order to visualize a textured eye, we decided to tessellate the eye sphere and associate a texture map to each face. The tessellation triangles do not need to be small: the smaller the triangles, the more faces will have to be processed, so processing time will increase, but display time will not change (the same number of pixels will be drawn). The tessellation must embody the sphere. Indeed, the surgeon looks at the interior of the eye, and some parts of the eye will not be displayed if the surgeon looks along a skimming direction.

The tessellation we choose is a regular icosahedra, which is a polyhedron with 20 equilateral triangles. The texture map of each triangle is put into an array, the color of each point being obtained by projecting the icosahedra onto the sphere (the center of the projection is the central point of the sphere).

Only one resolution of the texture is used for the map, for it is optimized for the usually used magnifying level. If the surgeon increases the magnifying level, several pixels on the screen will be displayed with the color of the same texel. If he decreases the magnifying level, several texels will not be used.

In an actual operation, only the surgeon's eye moves whereas in our model, the patient's one does. The whole laser path is computed (reflection onto the different mirrors) in order to get the location of the impact point on the tessellated eye. The impact is then drawn into the texture map. Its color depends on the pigmentation of the fundus, the firing duration, the power of the laser and the level of focus. The main advantage of integrating the spots into the texture map is that they are automatically displayed by the texture mapping algorithm without any specific algorithm.

6 Visualization

6.1 Visible surfaces determination

All the objects are projected onto the screen plane with a perspective projection. According to depth order, the objects are always arranged the same way. So, no sorting is needed. Objects are displayed sequentially, first the central lens, the iris, the fundus, then each mirror and the image of what is seen through it.

However, some objects have to be drawn only inside other objects. For example, the iris is seen through the central lens, therefore only the part of the iris in the lens must be drawn. In the same way, it is clear that the image through a mirror has to be clipped by the mirror. So it is interesting to exploit the fact that all the objects



are polygons on the screen. These polygons can define areas in which other objects will be drawn: the Sutherland-Hodgman algorithm is used (it calculates the intersection of two convex polygons, see [FDFH90]). Figure 6 shows the use of clipping polygons to obtain the polygon where the fundus and the iris will be drawn.

A polygon is defined as the intersection of the pupil, the internal edge of the lens, the edge of a mirror, and the edge of the front side of the lens. This polygon - now called clipping polygon- defines the zone where the fundus is seen: this zone has to be textured. Every well-oriented face of the eye is clipped by the clipping polygon and the result is displayed on the screen with its texture.

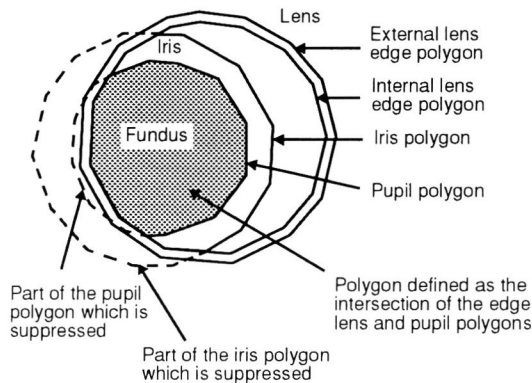


Figure 6: Clipping

6.2 Texture mapping

An efficient algorithm for texture mapping had to be implemented on the PC. The two-pass method described in [BS86] is interesting but does not allow the parameters to be interpolated. So, a bilinear interpolation is used [Hec86]. This method is fast since it can use integers. For each pixel, two additions and two tests are used. However, the method does not care of the perspective deformation (which requires the interpolation of a third variable and a division for each pixel). However, this is not very important in our case, because the faces that the surgeon observes are quite perpendicular to the visual axis, thus quite parallel to the screen plane: perspective deformation is thus negligible.

The tessellation of the sphere could also cause distortion, because of the non C1-continuity between two faces. However, this kind of defaults are hardly visible, even when a blood vessel passes from one face to another.

6.3 Rendering

Only the part of the eye which is lit by the slit is visible, the rest is too dark to be seen. The light from the slit looks like a rectangle on the screen. In order to suppress the discontinuous transition between the lit part and the dark one, a gradation on the edge of the rectangle has been implemented (the distance between the point and the edge determines the loss of brightness).

A linear gradation, although easy to implement, does not give realistic effects. Instead, we use another model, which is closer to physics, where the lighting subduing goes from 1 to 0 with a squared cosine. This seems very realistic and takes almost no time, because the function is precomputed (more exactly, given color and subduing, the resulting subdued color is read in a table).

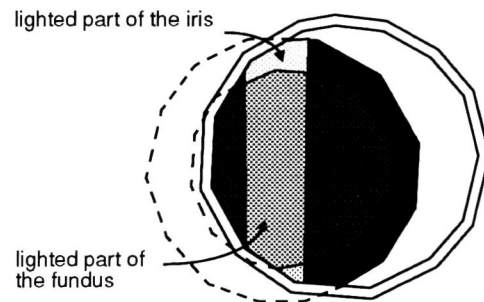


Figure 7: Slit lighting

Thanks to texture mapping and realism improvement, the resulting images are satisfying. Naturally, more special effects are still to be implemented (for example dazzling caused by light reflection). However, this is not sufficient for a perfect realism. Indeed, in an actual operation, sharpness of the image is controlled by the depth translation which focuses the different lenses. A good focus is required for a good photocoagulation, because the laser is efficient only when in focus. So it was important to simulate the depth of field of the binocular.

7 Depth of field simulation

Several solutions exist for simulating depth of field (e.g [PC81]). An accurate depth of field model can be implemented with a multi pass algorithm. In high end graphics systems, such algorithms can be efficiently implemented with a built-in accumulation buffer [HA90]. However, for low cost PC-based simulators,



multi pass algorithms are to be avoided because they are quite time consuming.

We propose here a solution for generating real-time 2D blurring effect for texture-based applications. This solution can be compared to the mip-mapping algorithm, where the texture map is defined at several levels of detail, the final color of a pixel being computed by interpolating between two levels. Our algorithm defines each texture map at several levels of bluriness, the final color of a pixel being also computed by interpolating between two levels.

In our application, the fundus or the iris seen through a mirror or the central lens can be completely out of focus, thus resulting in a heavily blurred image. In order to simulate any level of bluriness, we had to carefully choose how many basic levels had to be pre-computed and stored in memory. We found that four levels (including the original non-blurred image) are a good trade-off between accuracy and cost, as they provide smooth transitions from the first level (original image) to the last one (heavily blurred image).

7.1 Building blurred texture maps

Blurred texture maps are built by first filtering the original map (in spherical coordinates), and then generating the corresponding maps for each triangle of the eye. In our application, we chose 7x7, 13x13 and 21x21 filters for the three levels of bluriness. Although computing those maps is a pre-processing step only performed when building the texture map of a new eye, we implemented a Summed Area Table algorithm [Cro84], which allows any blurred image to be computed in a linear time whatever the size of the filter. The four texture maps (spherical coordinates) are then projected in order to build the actual maps associated to each triangle of the tessellated sphere. Once pre-computed, the maps are stored in the simulation PC database, ready to be used in an actual simulation.

All these (time-consuming) computations are done once for all in a pre-processing step. However, during the real-time simulation, 'dynamic' textures are used, since laser impacts are incorporated into the texture maps of the triangles they belong to (see section 5.3.). This means that blurred impacts are to be computed and incorporated in real-time in the three blurred texture maps. This is done by first drawing the impact in the non-blurred texture map, and then modifying in the three blurred texture maps only those texels around the impact that are affected by the filter.

7.2 The texture mapping algorithm

In our first texture mapping implementation, a linear interpolation across a triangle was used to find the texel coordinates corresponding to the current pixel. The color of the texel was then modulated to simulate the slit effect (see section 6.3.). To implement real-time blurring, the texel coordinates now define two texels from the closest levels of bluriness, the color of the pixel being computed by first interpolating between the two texels, and then modulating the color to simulate the slit effect. The exact interpolation between the two blurred colors normally requires at least two multiplications and one addition (per color component), which can be quite time-consuming on a PC. These computations can be efficiently replaced by a Look Up Table access (one per color component). In the current implementation, ten interpolation LUTs (64K each) are used (for $\alpha=0, 0.1, 0.2, \dots, 0.9$), providing 10 levels of bluriness between two pre-computed levels, and thus 30 levels of bluriness between the first one (non-blurred image) and the last one (heavily blurred image).

The result is a smooth real-time blurring effect that very accurately models the depth of field of the binocular, at the cost of storing four texture maps instead of one (which is no more a problem on a PC computer). Moreover, texture-mapped elements in an image can have various levels of bluriness depending on their distance to the binocular lens. For example, when the fundus is in focus in the central lens, the images in the mirrors are a bit out of focus, while the iris is completely out of focus (and thus heavily blurred). Such effects dramatically add realism to the computer generated images. However, the method does not accurately render the blurred boundary between the fundus and the iris. We are currently investigating transparency-based methods to solve this problem.

Results - Perspectives

The prototype currently runs at about 13 images per second (640x480, 32768 colors) on a Pentium 90-based PC. This rate will keep on increasing as the PC technology evolves (Pentium 150 MHz, P6, new PCI chipsets...). According to the ophtalmologists who have tried the simulator, it is very close to an actual operation. We are now working on adding new pathologies to the current database, and on automating the map creation process. Long-term project is the inclusion of the entire eye semiology, in order to simulate other diseases (for instance pathologies located in the anterior segment).



A prototype is being validated by the students at the hospital of Lille. Some other ophthalmologists team will be contacted to criticize and complete the data base. This project has been patented [CDKMR94] and is aimed to be industrialized.

Acknowledgments

The simulator is supported by DRED and CHR&U credits and by a grant from ANVAR (Agence Nationale de la Valorisation de la Recherche).

References

- [BCCT94] Bainville, E., Chaffanjon, P., Cinquin, P., Troccaz, J., and Lavallée, S., "Computer Augmented Medecine". *Troisième séminaire du groupe de travail "Animation et Simulation"* Lille 1994, 163-170.
- [BDM92] Beer-Gabel, M., Delmotte, S., and Muntlak, L., "Computer Assisted Training in Endoscopy (CATE) - From a Simulator to a Training Station". *Endoscopy* vol 24 suppl 2, 1992, 534-538.
- [BG93] Boissonat, J.D., and Geiger, B., "3d Simulation of Delivery". *Proceedings of Visualization'93*, 416-419.
- [BS86] Bier, E.A., and Sloan, K.R., "Two part Texture Mappings". *IEEE Computer Graphics & Applications* 6,10 September 1986,40-53.
- [CEB93] Cover, S.A., Ezquerro, N.F., O'Brien, J., Rowe, R., Gadacz, T., Palm, E., "Interactively Deformable Models for Surgery Simulation". *IEEE Computer Graphics & Applications* 13,6 november 1993.
- [CDKMR94] Chaillou, C., Dubois, P., Karpf S., Meseure, P., and Rouland, J.F., "Dispositif et Procédé de simulation d'un examen ou d'une opération chirurgicale effectuée sur un organe simulé". Dépôt de brevet no 9405487.
- [CGRB91] Clarysse, P., Gibon, D., Rousseau, J., Blond, S., Vasseur, C., and Marchandise, X., "A Computer-Assisted System for 3D Frameless Localization in Stereostatic MRI". *IEEE Transactions on Medical Imaging* 10,4 (1991).
- [Cro84] Crow, F. C., "Summed-area Tables for Texture Mapping", *Siggraph'84 conference Proceedings* 18,3 (1984), 207-12.
- [DRMKC95] P. Dubois, J.F. Rouland, P. Meseure, S. Karpf and C. Chaillou "Simulator for Laser Photocoagulation in Ophtalmology", *IEEE Transactions on biomedical engineering (to be published)*.
- [DSCP94] Delingette, H., Subsol, G., Cotin, S. and Pignon, J., "A craniofacial Surgery Simulation Testbed". *Rapport de recherche INRIA no 2199* february 1994.
- [FDFH90] Foley, J.D., van Dam, A., Feiner, S.K., and Hughes, J.F., *Computer Graphics: Principles and Practice*, Addison Wesley 1990.
- [GW87] Gillies, D.F., and Williams, C.B., "An Interactive Graphic Simulator for Teaching of Fibrendoscopic Techniques". *Proceedings of the EUROGRAPHICS'87 Conference* 1987,127-138.
- [HA90] Haeberli, P., and Akeley, K., "The Accumulation Buffer: Hardware Support for High Quality Rendering, *SIGGRAPH'90 Conference Proceedings* 24, 4 (1990) 309-318.
- [Hec86] Heckbert, P.S., "Survey of Texture Mapping". *IEEE Computer Graphics & Applications* 6,11 November 1986. 56-67.
- [HGW92] Haritsis, A., Gillies, D.F., and Williams, C.B., "Realistic Generation and Real Time Animation of Images of the Human Colon". *Proceedings of the EUROGRAPHICS'92* 11,3 (1992) 367-379.
- [MRDKC95] P. Meseure, J.F. Rouland, P. Dubois, S. Karpf et C. Chaillou "Sophocle: a Retinal Laser Photocoagulation Simulator - Overview", *Proceedings of CVRMed'95, 3-5 Avril 95, Nice FRANCE, Springer Verlag (to be published)*.
- [PC81] Potmesil, M. and Chakravarty, I. "A Lens and Aperture Camera Model for Synthetic Image Generation". *SIGGRAPH'81 Conference Proceedings* 15, 3 (1981), pp 297-305.
- [SBM94] Sagar, M. A., Bullivant, D., Mallinson, G. D., Hunter, P. J, and Hunter, J. W., "A Virtual Environment and Model of the Eye for Surgical Simulation". *SIGGRAPH'94 Conference Proceedings* (1994).
- [SCA94] Schwartz, S.L., Cao, Q. L., Azevedo, J., and Pandian, N.G., "Simulation of Intraoperative Visualization of Cardiac Structures and Study of Dynamic Surgical Anatomy with Real-Time Tridimensional Echocardiography". *Am. J. Cardiol.*, vol 74(7), 1994, 501-7.

