A virtual training simulator for learning cataract surgery with phacoemulsification

Kup-Sze Choi[1,*], Sophia Soo[1], Fu-Lai Chung[2]

[1]*School of Nursing, Hong Kong Polytechnic University, Hong Kong*
[2]*Department of Computing, Hong Kong Polytechnic University, Hong Kong*

Abstract

This paper presents the development of a low-cost cataract surgery simulator for trainees to practise phacoemulsification procedures with computer-generated models in virtual environments. It focuses on the training of cornea incision, capsulorrhexis and phaco-sculpting, which are simulated interactively with computationally efficient algorithms developed for tissue deformation, surface cutting and volume sculpting. Intuitive two-handed human-computer interactions are achieved with six degrees-of-freedom haptic devices. Performance of trainees on manual dexterity is recorded with quantifiable metrics. The proposed virtual-reality system has the potential to serve as an alternative training tool to supplement conventional cataract surgery education.

Keywords: Virtual reality; Surgical simulation; Medical training; Cataract surgery; Phacoemulsification

* Corresponding author. Tel.: +852-3400-3214; fax: +852-2364-9663.
E-mail address: hskschoi@inet.polyu.edu.hk (K.S. Choi)

1. Introduction

Cataract refers to clouding of the lens in the eye that obstructs the passage of light. It is the major cause of blindness and visual impairment, and is related to ageing in most cases. According to the World Health Organization, age-related cataract accounts for 48% of world blindness. Surgery is an effective treatment of cataract. It has been performed with intracapsular cataract extraction (ICCE) which is then supplanted by extracapsular cataract extraction (ECCE). While lens removal and implantation are not involved in ICCE, the clouded lens is removed in ECCE and is replaced with an artificial lens known as intraocular lens (IOL) implant. ECCE can be further divided into conventional ECCE and ultrasound phacoemulsification. The latter is now the most common and favoured technique since the wounds involved are small and usually self-sealing without suturing, thus enabling fast recovery. However, cataract surgery service remains inadequate in many countries.

Eye surgery is a complex operation requiring good hand-eye coordination and dexterous microscopic manipulations of instruments with high accuracy. Slight collisions with delicate ocular tissues can lead to blindness. Many years of education, including internship and residency training, are required to become specialized in ophthalmology. Like other surgical procedures, the indispensable master-apprentice model is a major approach for phacoemulsification training. Initially, trainees observe or assist in surgical operations performed by experienced surgeons. They also practise in wet labs with enucleated animal eyes. When a certain level of competency is developed, the trainees then begin to operate on patients under teacher's guidance. In this process, it is important to gain experience by practising on enucleated animal eyes

in wet labs [1, 2], which is a good alternative when post-mortem human eyes are not available. However, wet-lab training is limited in ophthalmology curriculum due to high financial costs. The anatomy and mechanical properties of animal eyes are also not consistent with that of human eyes [1]. Besides, animal experimentation could lead to ethical issues. These problems can be avoided with the aid of virtual reality technology. First, anatomically correct geometric models can be built based on the anatomy of human eye. Patient-specific eye models can also be re-constructed from the medical scanning data by volume or surface rendering techniques to enable surgical rehearsal and planning. In addition, difficult and rare cases can be simulated and made available for repeated practice. With appropriate biomechanical models and simulation techniques, the tissue responses in real surgical scenarios can be replicated and the tactile feeling can be reproduced through haptic devices [3]. Availability is no longer a problem since the computer-generated virtual surgical environments are reusable and accessible whenever necessary, making the training process more convenient and effective. Animal rights are certainly not a concern with this virtual-reality phacoemulsification trainer.

Furthermore, virtual-reality trainer is advantageous in that user performance can be recorded with quantitative measures, thus providing objective ways to assess surgical competency. Risk-free self-practice in virtual environments can be performed without teacher's on-site supervision, which expedites the learning curve and enhances trainee's ability to make independent judgement. While the initial setup and development cost may be high, virtual-reality technology is becoming more affordable nowadays. It is a promising learning tool to complement conventional phacoemulsification training curriculum [4].

Research effort has been devoted to the simulation of laser photocoagulation [5], vitrectomy [6-8], redial keratotomy [3] and in particular, phacoemulsification cataract surgery [9-14]. In the aspects of physical simulation, finite element method (FEM) has been adopted in conjunction with the elasticity theory to model large tissue deformation involved in eye surgery [3]. Mass-spring model (MSM) is also employed to model deformation and incisions performed on eye models constructed with mass points and springs [6]. ChainMail, a descriptive model that constrains nodal displacement by chains, also enjoys success in eye surgery simulation [14]. Besides, meshless method using smoothed particle hydrodynamics [9] and a hybrid approach combining MSM and FEM [11] have also been proposed to simulate the tissue responses during eye surgery. Among these simulators, the EYESi ophthalmic surgery trainer, capable of simulating vitrectomy and cataract surgery, has been made available in the market for surgical training [14]. On the other hand, researchers have begun to shift their focus to the study of the effectiveness of using virtual simulators for learning ophthalmic surgery as well as its ability in the transfer of skills, which can evaluated with quantitative measures of proficiency [15, 16]. Promising results have been obtained from clinical trials conducted to evaluate and compare the skill level between experts and novices [17-20].

This paper concerns the development of a low-cost, two-handed virtual surgical training system for learning cataract surgery by phacoemulsification. It concerns interactive simulation of the procedures involved in corneal incision, the creation of circular continuous capsulorrhexis (CCC) and the emulsification of the clouded lens (i.e. phaco-sculpting). Tissue responses during the surgery are modelled and simulated. A pair of haptic devices is employed as user interface to enable intuitive manipulations. More importantly, user performance can be recorded on-the-fly with quantitative

metrics in order to keep track of the learning progress of trainees and to provide them with objective feedback.

In what follows, the operative procedures of phacoemulsification will first be reviewed briefly in Section 2. The overall framework of the virtual cataract surgery training system and the geometric models created for the simulator are then presented in Section 3 and 4 respectively. The techniques developed in the simulation engine of the system are presented in Section 5, followed by the discussion on the simulation of three major procedures in phacoemulsification in Section 6. Discussion and conclusion are given in Section 7 and 8 respectively.

2. Phacoemulsification Procedure

Before providing a detailed discussion on the development of the proposed system, the basic steps of phacoemulsification are summarized as follows [21, 22]. The operative procedure of phacoemulsification begins by properly everting the eyelids and drape with a stick in order to avoid eyelashes getting into the surgical field and thus minimizing the risk of inflammation. Refer to the anatomy of human eye in Fig. 1, the eye is then stabilized by grasping the limbal conjunctivum with the forceps, after which a clear incision is made with an angled keratome by entering the cornea at the limbus at a shallow angle. The keratome is manoeuvred to pass forward through the corneal stroma and penetrate into the anterior chamber, until the blade is parallel to the iris plane. Through the tunnel created by corneal incision, viscoelastic solution is injected into and fully filled the anterior chamber to prevent chamber collapse.

Next, a continuous circular tear is made in the anterior capsule, which is known as capsulorrhexis or CCC. Here, a cystotome is used to puncture the capsule at the centre, from which a small flap is produced. The flap is pulled and torn carefully to create CCC

around the desired circumference. Hydrodissection is performed to separate the capsule from the epinucleus and thus allowing its free rotation within the capsule. This is achieved by injecting balanced salt solution in the form of fluid wave under the epinucleus.

The clouded lens is then emulsified by the use of ultrasound. Among the available techniques, the “divide and conquer” approach is an easy and safe method. First, within the CCC, the lens nucleus is sculpted to produce a cross-shape trench with minimum ultrasound energy. This step is referred to as phaco-sculpting. After proper rotation and manoeuvre, the sculpted nucleus is cracked into four separate quadrants and emulsified one by one. Irrigation and aspiration are then performed to remove the remaining soft lens matter. With the capsular bag and anterior chamber inflated with viscoelastic solution, the wound is enlarged to allow for easy insertion of a foldable intraocular lens implant. The IOL implant is folded with forceps and pushed into the capsular bag within which the lens unfolds afterwards. By using a dialling hook, the implant is rotated, pushed forwards and downwards, for stabilization and centralization.

Finally, all viscoelastic solution is removed. Closure is usually not required as the wounds can be self-sealed although stitching is needed in some cases. To ensure self-seal, balanced salt solution is injected into corneal stroma at the edge of the wound. In the proposed system, interactive simulation of corneal incision, capsulorrhexis and phaco-sculpting are concerned in particular, which involves the deformation and cutting of triangulated surface and the sculpting of volumetric tetrahedral mesh.

3. System framework

The overall framework of the proposed virtual cataract surgery trainer is shown schematically in Fig. 2. It is implemented on a personal computer with an Intel Core 2

Duo 2.66 GHz CPU and 2GB RAM, and equipped with a pair of Phantom Omni haptic devices. The software is developed with C/C++ where the graphics are programmed with OpenGL and the haptics with OpenHaptics and osgHaptics. The core of the framework is the simulation engine which governs the dynamics of simulated virtual objects and the interactions during tissue deformation, cutting, tearing, sculpting, piercing and grasping. The model database stores geometric models of the virtual objects, including the eye and the surgical tools. Through a pair of 6 degrees-of-freedom (DOF) haptic devices, user can manipulate virtual tools with both hands and perform surgical operations in the virtual environment. The real-time position and orientation of the virtual tools are read from the haptic devices and fed into the simulation engine, where collisions among the objects are detected continuously. When a tool-eye collision occurs, the simulation engine computes the visual and haptic responses on the fly, so that the objects are rendered graphically on the screen whilst force data are computed simultaneously to drive the haptic devices. Information regarding practice drills for corneal incision, capsulorrhexis and phaco-sculpting, e.g. procedures, standards and metrics, are stored in the training drills database which is accessed to retrieve the data required for simulating the corresponding procedures. Guidelines such as text instructions and visual cues are provided when a trainee is practising the surgical procedures. The training results are recorded in the user performance database for progress tracking and analysis.

4. Geometric Models

The major geometric model in the system is the virtual eye ball model which is constructed with reference to the anatomy of human eye, as shown in Fig. 3(a). The geometric models of cornea, anterior capsule, lens and sclera are constructed with 3D

surface meshes, containing 105, 96, 113 and 680 vertices respectively. The first two models are built with triangulated mesh with 179 and 167 elements respectively, whereas the other two models contain 127 and 679 quadrilaterals respectively. Texture mapping is applied to render the surface meshes. Note that the lens is also modelled with volumetric mesh in addition to the triangulated surface mesh described above. That is, two geometric representations are employed to model the lens. To increase rendering speed, the surface-based lens model is invoked when the simulated operative procedures do not involve topology changes of the lens. In the simulation of phaco-sculpting, however, since the lens is removed bit by bit progressively to form trenches, a volumetric tetrahedral lens model is more appropriate so that the tetrahedrons can be removed one by one accordingly. Virtual eyelids, eyelash and face surrounding the eye are also included to improve overall realism. The tools used in the cataract surgery simulation, including forceps, phacoemulsifier (i.e. phaco-probe), keratome and scalpel, are modelled with surface-based polygonal models, which are shown in Fig. 3(b).

5. The Simulation Engine

The role of the simulation engine is to simulate the responses due to tissue deformation, surface cutting and volume sculpting involved in the phacoemulsification process and to execute the corresponding algorithms developed for the simulation.

5.1. Tissue deformation

In the proposed system, simulation of tissue deformation is achieved by using the mass-spring model, where the vertices and edges of the geometric meshes are regarded as mass points and elastic springs. Here, the force propagation approach [23] (a.k.a. the

force propagation model, FPM) is employed to simulate the interactions between the mass points and springs as a result of externally applied forces. Deformation is considered as a sequential force transmission process with forces propagating from the surface contact points to their neighbouring nodes layer by layer at increasing topological distance. Instead of employing linear springs and setting all of them with the same spring stiffness k_e, it is modelled in the proposed system as follows,

$$k_e(i, j) = \frac{a_1 + a_2}{l_{ij}^2}$$

where $k_e(i,j)$ is the stiffness of the spring used to model an edge e connecting vertex i and j; a_1 and a_2 are the area of the two triangles sharing the common edge e; and l_{ij} is the rest length of the spring. This approach is more accurate than the conventional methods in simulating uniformly elastic surface [24]. Furthermore, the spring force $\mathbf{F}_e(i,j)$ has a quadratic relationship with spring extension $|\mathbf{x}|$, where $\mathbf{x} = |\mathbf{r}_j - \mathbf{r}_i|$ and $\mathbf{r}_i$ and $\mathbf{r}_j$ are the position of vertex i and j respectively. That is,

$$\mathbf{F}_e(i, j) = k_e(i, j) \cdot |\mathbf{x}|^2 \hat{\mathbf{x}},$$

which models tissue deformation and the tearing of anterior capsule during capsulorrhexis in a more realistic manner.

5.2. Surface mesh cutting

In cornea incision and capsulorrhexis, the 3D triangulated surface meshes representing the cornea and the anterior capsule are cut interactively. The mesh topology has to be updated on the fly in response to user's action. In general, interactive mesh cutting results in many possible cases of topology changes since a user can cut the mesh arbitrarily. It is thus necessary to generalize the topology changes into canonical cases without loss of generality, so that arbitrary cutting can be modelled routinely as

transitions from one case to another, thereby enabling continuous cutting in a progressive fashion, i.e. progressive cuts. Based on this strategy, a cutting simulation approach is developed as follows. The possible changes in topology of the triangulated surface mesh are first generalized into canonical cases. The system monitors mesh-tool interactions continuously to determine the way that a user cuts the mesh and thus the corresponding canonical cases. The mesh is then subdivided on the fly according to the identified cases, followed by the refinement of the subdivided mesh to expose the cut.

Several terms are defined for the discussion of this approach. Refer to Fig. 3(c), *cut points* refers to mesh-tool intersections sampled at regular intervals; *cut segment* is a straight line joining two consecutive cut points; *cut path* is a concatenation of successive cut segments created over time, which extends incrementally until the cutting tool leaves the mesh. When a cut is being made, it is possible for the cut path to sweep across different number of triangles, depending on the length of the cut segment. A long incision is regarded as the concatenation of consecutive cut segments over time. This is analogous to piecewise linear approximation of a curve. The canonical cases of topology changes are obtained by considering the various ways that a cut segment intersects would a triangulated surface mesh. The cases are classified into two categories – *initial cut* (IC) and *subsequent cut* (SC). IC is the *first* cut segment created for a new cut path, whereas SCs refer to the cuts created after an initial cut has been made (see Fig. 3(c)). They are considered separately since IC involves the creation of 2 cut points while SC only produces one. In both categories, different mesh subdivision schemes are identified and applied depending on the number of triangles that a new cut segment spans across. As illustrated in Fig. 4(a), three cases of mesh subdivision are concerned when the cut segment span across (i) one triangle, (ii) two adjacent triangles,

or (iii) more than two triangles. Furthermore, the cut points of a cut segment may lie inside a triangle, at a vertex or on an edge, which are also considered separately as shown with Table 1 and Fig. 4(b). In the process of arbitrary cutting, it is always possible for a cut point, as well as the intersection points between a cut segment and the mesh, to be too close to the mesh vertices, resulting in tiny and degenerate elements which can reduce the stability of numerical simulation. In the developed approach, proximity check is conducted to identify these improper points. That is, if a point is too close to a vertex in the mesh, it is directly snapped to that vertex; if the perpendicular distance from the point to an edge is below a certain threshold, it is snapped to the edge. Otherwise, the point is considered as an ordinary point inside a triangle. Some examples of ICs with the cut segment lying within one triangle and spanning across two triangles are shown in Fig. 4(b), where special cases of mesh subdivision are also taken into account, including cut point coinciding with a vertex or lying on an edge. Mesh cutting involving more than 2 triangles is represented with a combination of the ICs involving one or two triangles in Table 1, and simulated simply by invoking the corresponding algorithms.

After an IC is performed, the incision can be extended by making a new cut segment, i.e. SC, to join the initial cut, which can then be followed by a series of successive SCs to produce a progressively extending cut path until the cutting tool leaves the mesh. Since SC always begins from a mesh vertex, i.e. the cut point newly created by the previous cut segment, the IC cases where the first cut point located at a mesh vertex can be applied directly to subdivide the triangles associated with SCs (see the last row in Fig. 4(b)). In other words, the corresponding algorithms developed for ICs can be re-used to subdivide the triangles due to SC. Hence, the three cases of mesh subdivision for SC are

indeed a subset of that for IC, as indicated in Table 1. Note that when an SC is performed, an additional step is required to duplicate the current cut point c_t to create an identical copy c'_t. Both of them are involved in mesh subdivision due to the new cut point c_{t+1} in order to further extend and expose the incision. The process is depicted in Fig. 4(c). After the generalization of the changes in mesh topology, cutting can be simulated interactively by focusing on how a cut segment intersects the triangles and determining the corresponding cut cases and transitions while progressive cutting is being performed.

As the mesh is subdivided to conform to the cut path, the number of triangles increases and their area decreases, resulting in degradation of mesh quality. To circumvent this situation, local remeshing is performed on the triangles surrounding the cut path. Based on a selective merging approach, the triangles are visited successively in a pairwise fashion where two adjacent triangles are considered at a time. As illustrated in Fig. 5(a), the triangle pairs surrounding the cut path, denoted by (a, b), (b, c) and so on, are to be taken into account. A pair of adjacent triangles merges when (i) both triangles have one of its edges on the cut path, and (ii) these two edges are almost co-linear, crossing at an angle smaller than a certain threshold. The second criterion is to ensure that merging does not introduce significant modification to the shape of the cut path. The triangle pairs (c, d) and (h, i) meet the requirements and merge to form a single triangle. The results shown on the right of Fig. 5(a) demonstrate that the remeshing scheme reduces the element count and avoids the creation of tiny elements.

In the proposed system, 3D surface mesh is modelled with pre-stretched springs as the edges. When an incision is made, the intersected springs are split and the vertices on the cut path are identified, serving as the starting points of force propagation in

deformation simulation using FPM. The cut is visually exposed due to the restoring forces in the pre-stretched springs, as shown in Fig. 5(b). This mesh cutting scheme is also applied to simulate capsulorrhexis by defining the cut points created due to tearing which will be discussed in Section 6.2.

5.3. Volume sculpting

In phacoemulsification, the clouded lens is sculpted bit by bit using ultrasound phaco-probe to produce a 3D cross-shape trench. To simulate this procedure, tetrahedral mesh is adopted to model the geometry of the clouded lens. The lens model is built by first constructing a triangulated surface mesh and then applying Delaunay tetrahedralization to yield a tetrahedral volumetric mesh. The tetrahedrons are created such that the centre of mass of each tetrahedron is inside the tetrahedron. The mesh is updated interactively by deleting the tetrahedrons being sculpted away using the tip of the virtual phaco-probe. This approach is convenient and suitable for simulating the trenching of the clouded lens, avoiding tedious topology update in interactive 3D volume cutting [25]. However, since a large number of tetrahedral elements are necessary to render the visual effect of lens sculpting and trenching (14,300 tetrahedrons are used in the current implementation), the detection of probe-lens collision becomes a time critical step and brute-force search of all the tetrahedrons is prohibitive. Instead, collision detection is achieved by employing bounding boxes arranged with hierarchical octree to partition the 3D space containing the lens. At initialization, the tetrahedrons inside the subspace bounded by a leaf box of the octree are collected by using a list in the data structure of that leaf box (see Fig. 6). In addition, the tetrahedrons enclosed within each leaf box are depth-sorted so that the tetrahedrons closer to the user's viewpoint are placed at the front of the list to expedite tetrahedron

search. Since a trench is created in the lens by hollowing it out progressively, the tetrahedrons to be removed are usually next to those tetrahedrons being removed. To this end, the data structure of a tetrahedron registers its four neighbours, if any, so that when a tetrahedron collides with the probe tip, its neighbours can be accessed directly to detect for subsequent collisions.

The collision detection scheme is a two-phase algorithm. Collision between the bounding boxes and the probe is first performed by searching the octree downward from the root to the children boxes through breadth-first traversal. If probe-lens collision occurs, the leaf bounding box that intersects the probe tip will be identified and the second phase of the algorithm will be activated. The problem is then reduced to finding out from the collided leaf bounding box the tetrahedrons that intersect with the probe tip. Here, the probe tip is modelled as a bounding sphere and the test for collisions is carried out simply by evaluating the distance between the centre of a tetrahedron and the tip. Collision is identified when the distance is less than the radius of the bounding sphere. By controlling the radius and depending on the resolution of the tetrahedral mesh, the probe tip may collide with multiple tetrahedrons which can be removed simultaneously. This also provides a means to simulate the amount of ultrasound energy applied with the radius of the bounding sphere. While precise edge-face intersections between the probe and the tetrahedrons can be computed to detect for collisions, it is relatively more time-consuming and the level of accuracy provided is not necessary here. Besides, only the tetrahedrons on the surface of the virtual lens are rendered in order to improve real-time performance. A list of surface tetrahedrons is maintained and updated interactively in response to the process of sculpting where surface tetrahedrons are removed and internal tetrahedrons are exposed to become surface tetrahedrons.

5.4. Nucleus rotation

The virtual lens can be rotated about the optical axis during phaco-sculpting. The situation is depicted schematically in Fig. 7. The moment of inertia I of the lens is obtained by considering each constituting tetrahedron as a point mass of equal weight. The initial moment of inertia I_0 is given by

$$I_0 = \sum_i mR_i^2 = m\sum_i R_i^2 ,$$

where R_i is the distance between the optical axis and the i-th tetrahedron. While the lens is being sculpted, I is updated on the fly to reflect the reduction in inertia due to interactive tetrahedron removal. That is,

$$I = I_0 - m\sum_j r_j^2 ,$$

where r_j is the distance between the optical axis and the j-th deleted tetrahedron. The rotational motion is modelled with the assumption that there exists a constant damping torque T_d about the optical axis. To rotate the lens, the applied torque T is required to be large enough to counteract T_d before the residual torque induces an angular acceleration α for the lens. That is, $T - T_d = I\alpha$. Quasi-static equilibrium is also assumed here such that the initial angular velocity of the lens at the beginning of each simulation time step Δt is small and negligible. The angular displacement θ is given by $\theta = \frac{1}{2}\alpha(\Delta t)^2$. By reading the coordinates of the haptic interface point from the Phantom Omni device, the previous and current position of the lens-probe contact points are obtained to calculate the distance between these two consecutive points, which is then used to estimate the force applied to the lens. The applied force is set to be linearly dependent on the distance.

With the modelling and simulation techniques discussed above, the proposed virtual cataract surgery trainer is able to achieve the frame rates of 17, 23 and 10 frames per seconds respectively in the simulation of corneal incision, capsulorrhexis and phaco-sculpting.

6. Simulation of Phacoemulsification

By using the methods discussed in the last section, the three phacoemulsification steps are simulated interactively in the virtual cataract surgery trainer with reference to the procedures in real surgery [21, 22]. User performance is recorded with quantitative metrics during the virtual training process.

6.1. Corneal incision

To perform corneal incision, the limbal conjuctivum is grasped with a forceps either at 180 degrees opposite to the entry site or within 30 degrees of the entry site, followed by using an angled keratome to enter the cornea at the limubs at a shallow angle and then penetrate into the anterior chamber. The simulated scenarios are shown in Fig. 8(a), where visual guides are superimposed to suggest the optimum positions of grasping and puncture. Virtual grasping is simulated by picking a point on the surface mesh of the limbal conjuctivum model. In general, the grasped point can be regarded as a mesh vertex (trivial case), a point within a triangle or on the edge of a triangle, depending on its proximity to mesh vertices or edges. To simulate the last two scenarios, especially necessary when the mesh is of low resolution, a vertex and the associated edges are created temporarily to represent the grasped point, as shown in Fig. 9. Forces propagate from the grasped point to vertices in the mesh to simulate deformation due to grasping by using FPM. The temporary vertex and edges are removed after the mesh is released.

Virtual incision of cornea is achieved by subdividing the mesh elements using the cutting simulation method discussed in Section 5.2. In corneal incision, a virtual forceps and a keratome are manipulated with the left and the right hand respectively via haptic devices. Manual dexterity of trainees can be assessed by measuring the distance from the optimal grasping location, the angle and size of insertion, the penetration depth and the completion time.

6.2. Capsulorrhexis

The first step of CCC is to pierce into the anterior capsule at the centre by using a cystotome, followed by making a radial cut which is eventually diverted to circumferential direction to produce a flap. The flap is torn gently and continuously along the circumference of the anterior capsule to produce capsulorrhexis, as depicted by the visual cue shown in Fig. 8(b). To model the tearing process by employing the cutting simulation method in Section 5.2, it is necessary to determine the cut points due to tearing, which can be obtained by the linear combination of the current cut segment vector $\mathbf{c}_t$ and the projected force vector exerted on the flap $\mathbf{f}_t$ at time t. As shown in Fig. 10, the position of the new cut point $\mathbf{c}_{t+1}$ at time (t+1) is given by

$$\mathbf{c}_{t+1} = a\,\mathbf{f}_t + b\,\hat{\mathbf{c}}_t\,.$$

A perquisite before the new cut point $\mathbf{c}_{t+1}$ can be sampled is that $\mathbf{f}_t$ should exceed a preset threshold force of tearing. Furthermore, the new cut point will be snapped to the nearest mesh primitives, i.e. vertices or edges, if it is close to either of them. While this modelling approach is not based strictly on the physics of tearing, it is able to render the real scenarios in CCC by tuning the parameters a and b. In CCC, a virtual forceps and a cystotome are manoeuvred respectively with the left and the right hand. The completion time, location of puncture made on the anterior capsule, the size of cut, and the

trajectory of cystotome-capsule contacts during CCC can be recorded to evaluate user performance.

6.3. Phaco-sculpting

In phaco-sculpting, the lens is sculpted and emulsified bit by bit to produce a cross-shape trench. The trenches should be wide and deep enough; and they need to cross at 90 degrees to each other, as shown in Fig. 8(c). This procedure is simulated by using the virtual phaco-probe to remove the tetrahedrons in the volumetric mesh representing the clouded lens. The completion time, number of tetrahedrons removed, trench width and depth, and the crossing angle can be measured during the virtual sculpting process.

6.4. User Performance Data

The proposed system measures user performance quantitatively for the three phacoemulsification steps as descried above. The data are converted to generic XML format for convenient storage in database and to facilitate data exchange with external applications. In addition, the kinematics of movement of both hands during the training can also be recorded. A typical example is illustrated in Fig. 11. The magnitude of contact force during virtual training and its distribution are shown in Fig. 11(a), from which the average, standard deviation and the maximum value can be obtained to study the performance of trainees. Similar graphs regarding the speed of the haptic stylus end-point during the training are shown Fig. 11(b). The contacts between the cystotome and the anterior capsule during capsulorrhexis are recorded and displayed graphically in Fig. 11(c) to visualize the trajectory. These data provide specific and quantitative information about the manual dexterity of trainees.

7. Discussion

Among the three phacoemulsification steps investigated in this research, it is relatively easy to simulate the real surgical situations during corneal incision and capsulorrhexis. Here, cornea and anterior chamber are modelled with deformable triangulated surface mesh. In corneal incision which involves the making of a small opening, the mesh topology is modified only slightly and the resulting visual effect is similar to that in real surgery. However, the change in mesh topology during continuous circular tearing is more substantial and the quality of the virtual image is limited by the resolution of the surface mesh. The edges of the peeled membrane therefore appear jaggy in the simulation unless the mesh resolution is increased. During interactive simulation, user is able to observe the simulated tissue deformation and feel the feedback forces upon manipulations using virtual surgical tools.

Simulation of phaco-sculpting is more demanding and the tissues involved are highly dynamic during the operation. In the developed simulator, the dynamic process is partially simulated. First, the cataractous lens is modelled as a volumetric tetrahedral mesh and sculpting is simulated by removing individual tetrahedral elements interactively. The sculpted trenches observed in the virtual images are jagged since the tetrahedrons are rendered as-is. During phaco-sculpting, the virtual lens is allowed to rotate about the optical axis and the moment of inertia is updated on the fly to reflect the reduction in inertia due to interactive tetrahedron removal.

The simulator is developed with emphasis on practising the creation of the cross-shaped trench during phaco-sculpting; hence the true dynamics of divided lens nucleus or posterior capsule occurred in real surgery is not completely simulated. That is, even though rotational motions about the optical axis are simulated, the relative position

between each tetrahedral element and the centre of the lens remains the same throughout the simulation, unless it is removed.

Except the tissues described above, the other parts of the virtual eye are modelled as static objects which remain stationary throughout the simulation. For example, when the cataract is being sculpted with the ultrasound probe, the induced movement of the entire eyeball that occurs in real surgical operations is not simulated.

While the focus of the current simulator implementation is to familiarize trainees with the procedures of how the three phaco operations (corneal incision, capsulorrhexis and phaco-sculpting) are performed and to measure their performance quantitatively, further endeavour to enhance the realism of the virtual environments is beneficial to the trainees. However, to exactly reproduce the phaco-dynamics in surgical scenarios [26] is technically a very challenging task. It depends on various factors, e.g. nucleus hardness, phaco power, vacuum power, viscoelastic injection, aspiration and irrigation, and requires to the use of physics-based methods such as computational fluid dynamics to simulate the divided nucleus and fluid waves.

The simulator considers the cataractous lens as a homogeneous and isotropic object with fixed hardness. It does not take into consideration of the degree of nucleus sclerosis. The current simulator is developed for training on making the cross-shaped trench by applying the divide-and-conquer technique. This technique is suitable for sculpting medium cataract and taught to most beginning cataract surgeons [27]. For soft and hard cataract, other phaco-sculpting techniques are needed, such as chip-and-flip or chopping. In the regard, the simulator can be further expanded to simulate the degree of nucleus sclerosis with appropriate model parameters so that comprehensive training of different phaco-sculpting techniques can be performed.

8. Conclusion

This paper presents a VR-based simulation system for learning cataract surgery with phacoemulsification. The current implementation emphasizes the procedural training of corneal incision, capsulorrhexis and phaco-sculpting, and the measurement of user performance on manual dexterity in terms of quantitative metrics. By studying the quantitative information, it is possible for mentors or trainees can spot specific flaws and weakness during the operative procedures and focused training can be performed to polish up the skills. The proposed simulator is developed using computationally efficient simulation methods which can be implemented on a generic personal computer equipped with a pair of standard haptic devices available commercially in the market. The system is thus relatively inexpensive and can be popularized for medical training even with sub-optimal computing resources.

In the current implementation, however, the proposed system only consider the simulation of three intermediate steps in phacoemulsification, other procedures like hydrodissection and the replacement of the clouded lens with IOL implant are yet to be included in the simulator to realize a more comprehensive system. Besides, to simulate the tearing of anterior capsule during capsulorrhexis, a descriptive model is employed instead of using the real physics. While this strategy simplifies the modelling, the system parameters have to be tuned interactively to produce the desired result. Phaco-sculpting is simulated in the proposed system by interactive removal of tetrahedral elements from the virtual lens model. This simulation approach is efficient and adequate for the procedural training of making a cross-shape trench in the lens, but the visual effect is not so appealing that jaggy edges are produced. More sophisticated volume sculpting approaches can be explored to improve the results [28]. Although the

binocular vision in real surgery is not considered in the proposed system, it can be incorporated into the system with minor software modification and at the expense of hardware cost.

A feature of the proposed system is the provision of quantitative information about the performance on procedural training after the practice is completed. The information can serve as objective feedback to suggest trainees the parts of procedures that requires extra attention and practice. Indeed, some of the performance metrics can also be displayed in the virtual environment in real time so that trainees are provided with instant and timely feedback, which could be more helpful to improve their performance.

Usability study will be conducted to evaluate and improve the system. Quantification of expert knowledge will be performed by inviting ophthalmic surgeons to conduct virtual phacoemulsification using the proposed system. The performance data of experts will be analyzed to establish a benchmark so that the skill level of novices can be compared to determine the standard they have reached.

Acknowledgment

This work was supported in part by the Research Grants Council of the HKSAR (Project No. PolyU5147/06E and PolyU5145/05E). The authors would like to thank Dr. K. Sylvain and Dr. A.W. Siu for their support to the projects. Special thanks are due to Dr. V. Woo who demonstrated and explained the procedures in real phaco surgery, and to the anonymous reviewers who provided helpful comments to our work.

References

[1] V. K. Dada and N. Sindhu, "Cataract in enucleated goat eyes: training model for phacoemulsification," *Journal of Cataract & Refractive Surgery*, vol. 26, pp. 1114-1116, 2000.

[2] C. Hashimoto, D. Kurosaka, and Y. Uetsuki, "Teaching continuous curvilinear capsulorhexis using a postmortem pig eye with simulated cataract," *Journal of Cataract & Refractive Surgery*, vol. 27, pp. 814-816, 2001.

[3] I. W. Hunter, L. A. Jones, M. A. Sagar, S. R. Lafontaine, and P. J. Hunter, "Ophthalmic microsurgical robot and associated virtual environment," *Computers in Biology and Medicine*, vol. 25, pp. 173-182, 1995.

[4] Y. M. Khalifa, D. Bogorad, V. Gibson, J. Peifer, and J. Nussbaum, "Virtual Reality in Ophthalmology Training," *Survey of Ophthalmology*, vol. 51, pp. 259-273, 2006.

[5] P. Dubois, J. F. Rouland, P. Meseure, S. Karpf, and C. Chaillou, "Simulator for laser photocoagulation in ophthalmology," *IEEE Transactions on Biomedical Engineering*, vol. 42, pp. 688-693, 1995.

[6] P. F. Neumann, L. L. Sadler, and J. Gieser, "Virtual Reality Vitrectomy Simulator," in *Medical Image Computing and Computer-Assisted Interventation — MICCAI'98*, 1998, pp. 910.

[7] T. Hikichi, A. Yoshida, S. Igarashi, N. Mukai, M. Harada, K. Muroi, and T. Terada, "Vitreous surgery simulator," *Archives of ophthalmology*, vol. 118, pp. 1679-1681, 2000.

[8] D. Verma, D. Wills, and M. Verma, "Virtual reality simulator for vitreoretinal surgery," *Eye*, vol. 17, pp. 71-73, 2003.

[9] M. Agus, E. Gobbetti, G. Pintore, G. Zanetti, and A. Zorcolo, "Real Time Simulation of Phaco-emulsification for Cataract Surgery Training," presented at Workshop in Virtual Reality Interactions and Physical Simulations (VRIPHYS 2006), Madrid, Spain, 2006.

[10] J. F. Perez, R. Barea, L. Boquete, M. A. Hidalgo, M. Dapena, G. Vilar, and I. Dapena, "Cataract Surgery Simulator for Medical Education & Finite Element/3D Human Eye Model," in *14th Nordic-Baltic Conference on Biomedical Engineering and Medical Physics*, 2008, pp. 429-432.

[11] N. Santerre, F. Blondel, F. Racoussot, G. Laverdure, S. Karpf, P. Dubois, and J.-F. Rouland, "A teaching medical simulator: phacoemulsification in virtual reality," *J Fr Ophtalmol.* , vol. 30, pp. 621-626, 2007.

[12] M. Sinclair, J. Peifer, R. Haleblian, M. Luxenberg, G. K, and H. DS, "Computer-simulated eye surgery," *Ophthalmology*, vol. 102, pp. 517-521, 1995.

[13] C.-G. Laurell, P. Söderberg, L. Nordh, E. Skarman, and P. Nordqvist, "Computer-simulated phacoemulsification," *Ophthalmology*, vol. 111, pp. 693-698, 2004.

[14] C. Wagner, M. Schill, and R. Männer, "Intraocular surgery on a virtual eye," *Commun. ACM*, vol. 45, pp. 45-49, 2002.

[15] P. Soderberg, C.-G. Laurell, W. Simawi, E. Skarman, P. Nordqvist, and L. Nordh, "Performance index for virtual reality phacoemulsification surgery," presented at Ophthalmic Technologies XVII, San Jose, CA, USA, 2007.

[16] P. G. Soderberg, C.-G. Laurell, W. Simawi, P. Nordqvist, E. Skarman, and L. Nordh, "Evaluation of response variables in computer-simulated virtual cataract surgery," presented at Ophthalmic Technologies XVI, San Jose, CA, USA, 2006.
[17] B. H. Feldman, J. M. Ake, and C. E. Geist, "Virtual Reality Simulation," *Ophthalmology*, vol. 114, pp. 828.e1-828.e4, 2007.
[18] P. Soderberg, C.-G. Laurell, W. Simawi, E. Skarman, L. Nordh, and P. Nordqvist, "Measuring performance in virtual reality phacoemulsification surgery," presented at Ophthalmic Technologies XVIII, San Jose, CA, USA, 2008.
[19] P. Soderberg, C.-G. Laurell, W. Simawi, P. Nordqvist, E. Skarman, and L. Nordh, "Virtual reality phacoemulsification: a comparison between skilled surgeons and students naive to cataract surgery," presented at Ophthalmic Technologies XV, San Jose, CA, USA, 2005.
[20] J. V. Rossi, D. Verma, G. Y. Fujii, R. R. Lakhanpal, S. L. Wu, M. S. Humayun, and E. J. De Juan, "Virtual vitreoretinal surgical simulator as a training tool," *Retina*, vol. 24, pp. 231-236, 2004.
[21] R. Caesar and L. Benjamin, *Phacoemulsification: step by step*. Edinburgh: Butterworth-Heinemann, 2003.
[22] A. Foss, *Essential ophthalmic surgery*. Oxford: Butterworth-Heinemann, 2001.
[23] K.-S. Choi, H. Sun, and P.-A. Heng, "Interactive deformation of soft tissues with haptic feedback for medical learning," *IEEE Transactions on Information Technology in Biomedicine*, vol. 7, pp. 358-363, 2003.
[24] S. Ferdi, E. P. Richard, E. C. Wayne, and F. M. Stephen, "Anatomy-based modeling of the human musculature," in *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*: ACM Press/Addison-Wesley Publishing Co., 1997.
[25] D. Bielser, V. A. Maiwald, and M. H. Gross, "Interactive Cuts through 3-Dimensional Soft Tissue," *Computer Graphics Forum*, vol. 18, pp. 31-38, 1999.
[26] B. S. Seibel, *Phacodynamics : mastering the tools and techniques of phacoemulsification surgery*, 4th ed. Thorofare, NJ: SLACK, 2005.
[27] L. Benjamin, *Cataract surgery*. Edinburgh: Elsevier Saunders, 2007.
[28] K. T. McDonnell, H. Qin, and R. A. Wlodarczyk, "Virtual clay: a real-time sculpting system with haptic toolkits," in *Proceedings of the 2001 symposium on Interactive 3D graphics*: ACM, 2001.

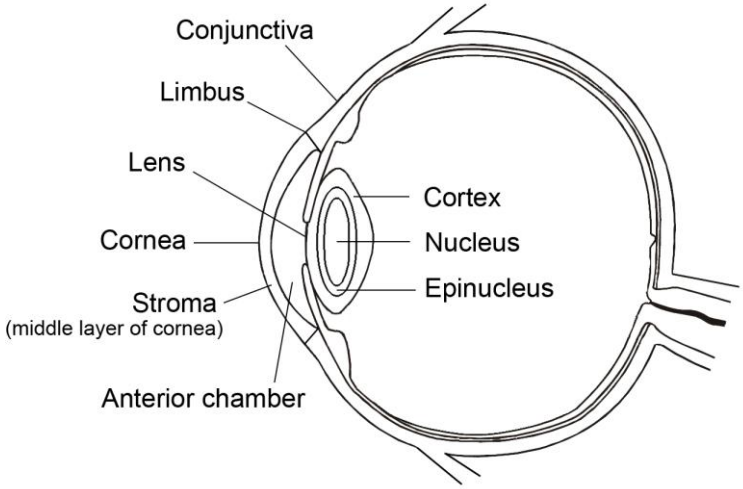


Fig. 1 Anatomy of the human eye.

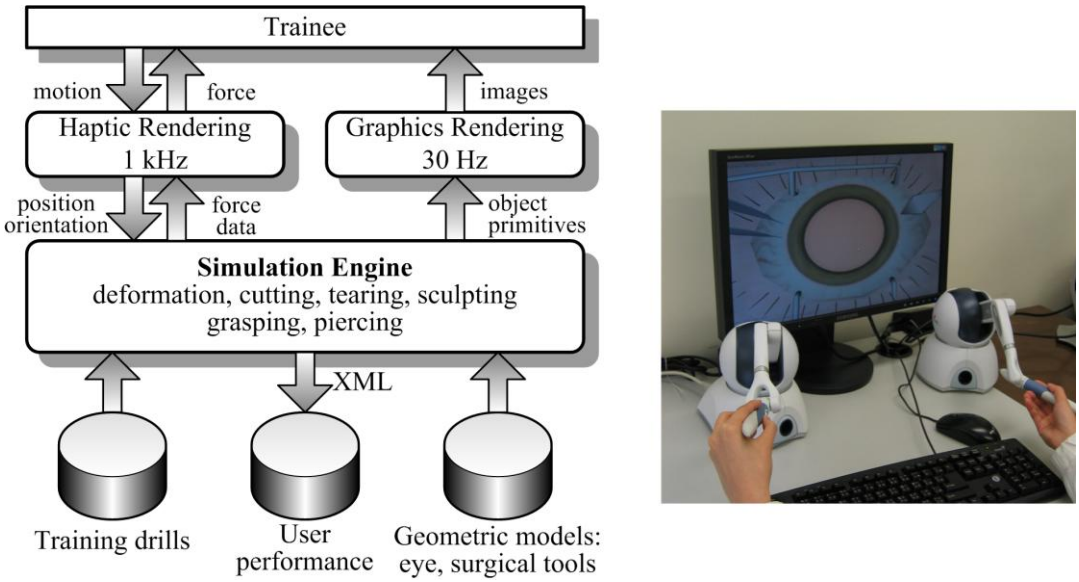
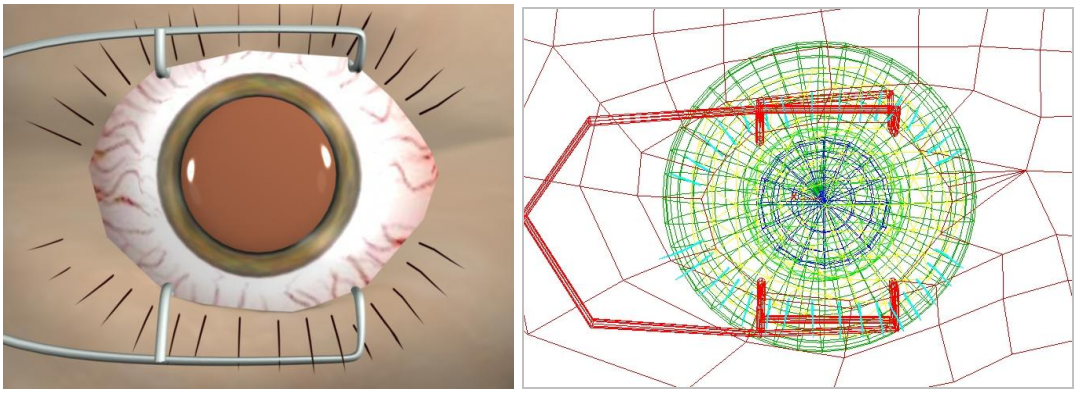
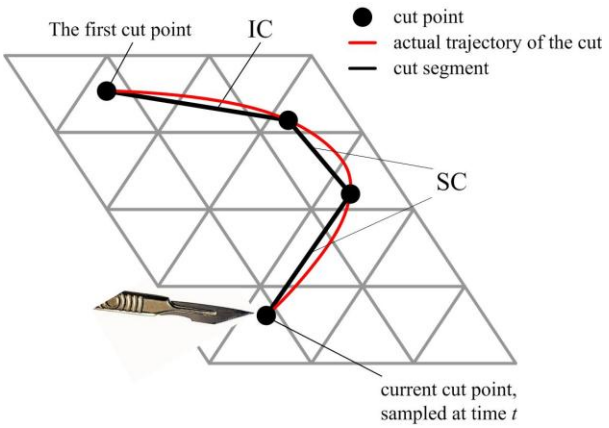


Fig. 2 The virtual cataract surgery training simulator: (a) system framework (b) a photo of the simulator.



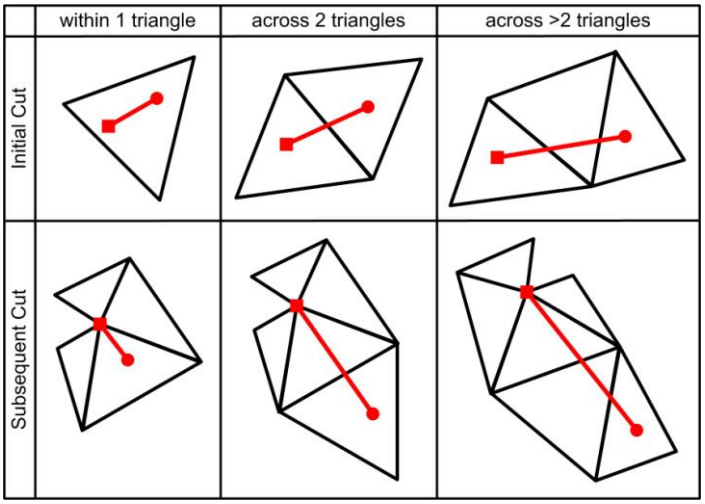
(a)

(b)

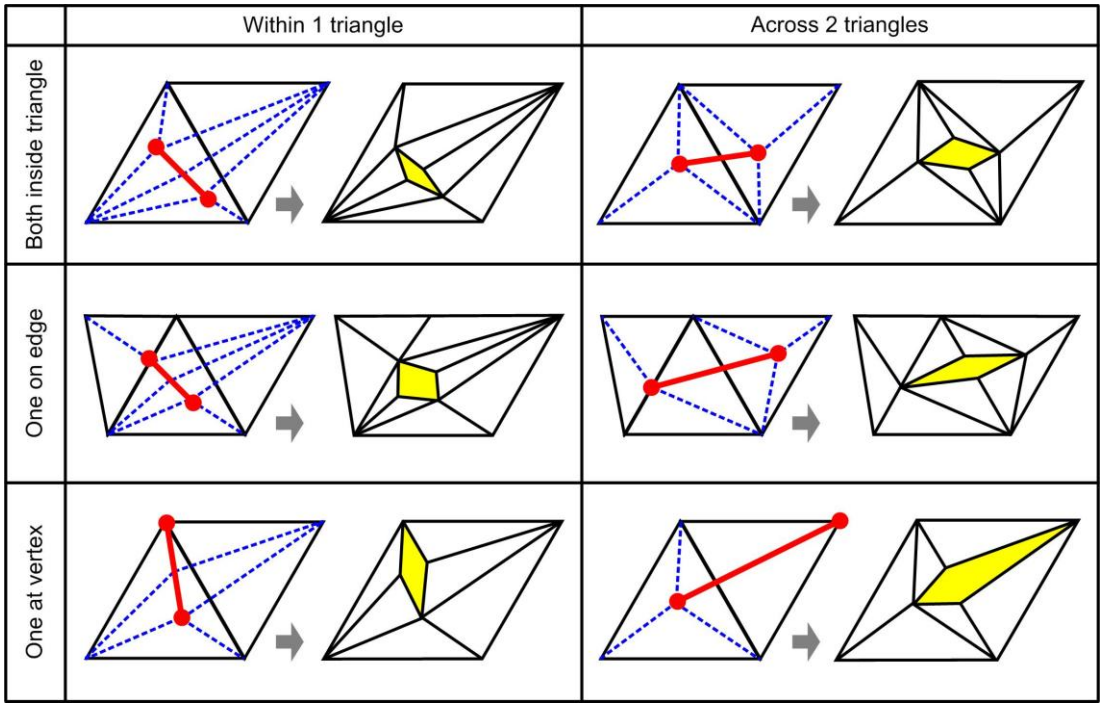


(c)

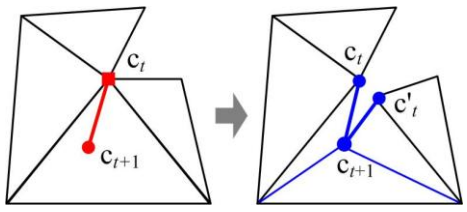
Fig. 3 (a) Virtual eyeball: surface rendering with texture mapping (left) wireframe (right); (b) virtual tools for simulating phacoemulsification; (c) terminology used in cutting simulation. The trajectory of the cut being made is sampled at regular time intervals and modelled as a piecewise linear path.



(a)

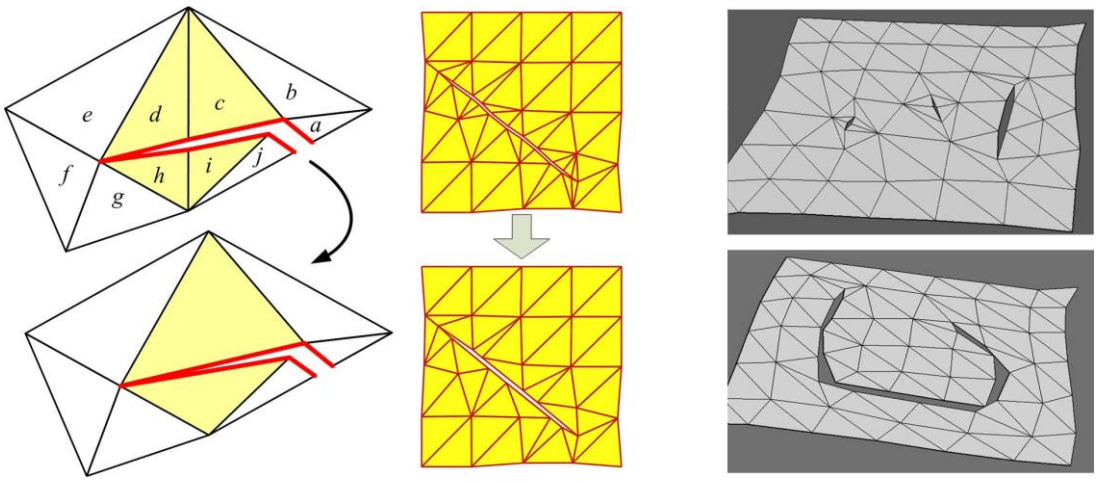


(b)



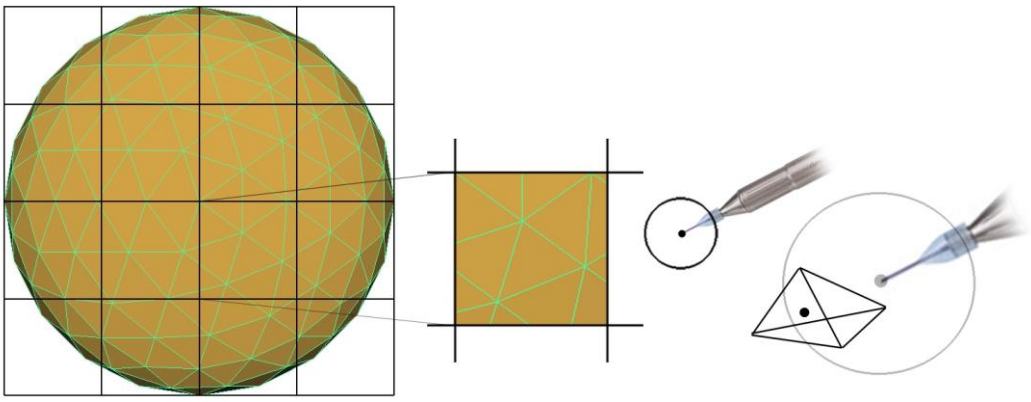
(c)

Fig. 4 (a) Mesh cutting involving one, two and more than two triangles. The square and round dots denote the start and end of a cut segment. (b) Examples of initial cuts with the cut points situated at three possible locations. (c) To create a subsequent cut, the current cut point c_t is duplicated to give an identical copy c'_t. Both of them are involved in mesh subdivision when the new cut point c_{t+1} is created.



(a) (b)

Fig. 5 (a) The local remeshing scheme: triangles along the cut path merge to form a single triangle; (b) Examples of ICs (top) and progressive cut (bottom).



(a) (b) (c)

Fig. 6 Phaco-sculpting simulation: (a) partition of the space containing the lens with octree, (b) each leaf box only stores a small number of tetrahedrons, (c) a tetrahedron is removed if its centre is within the bounding sphere attached to the tip of the phaco-probe.

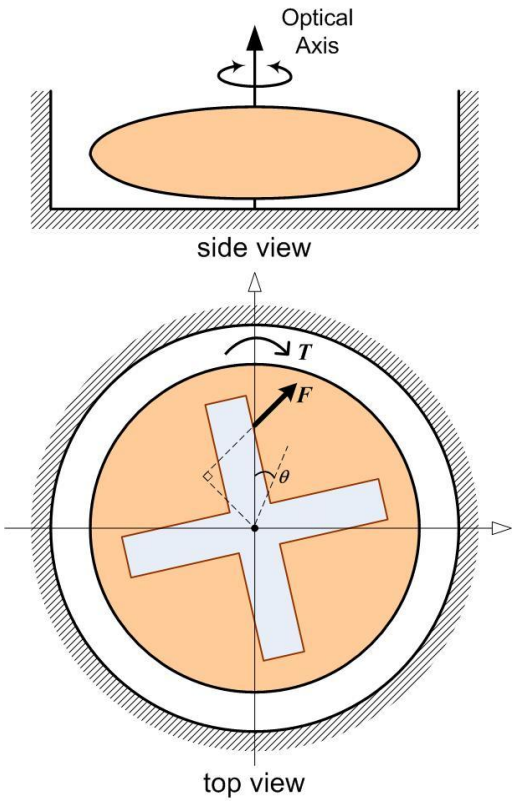
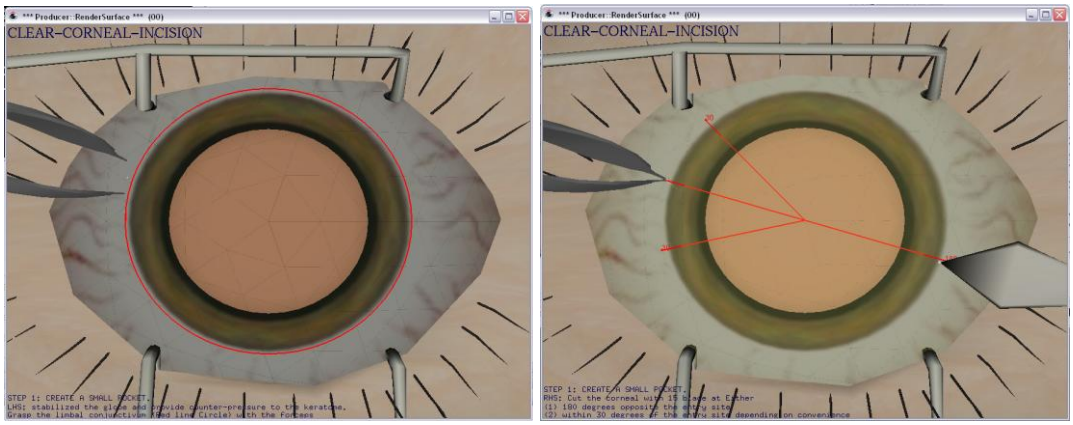
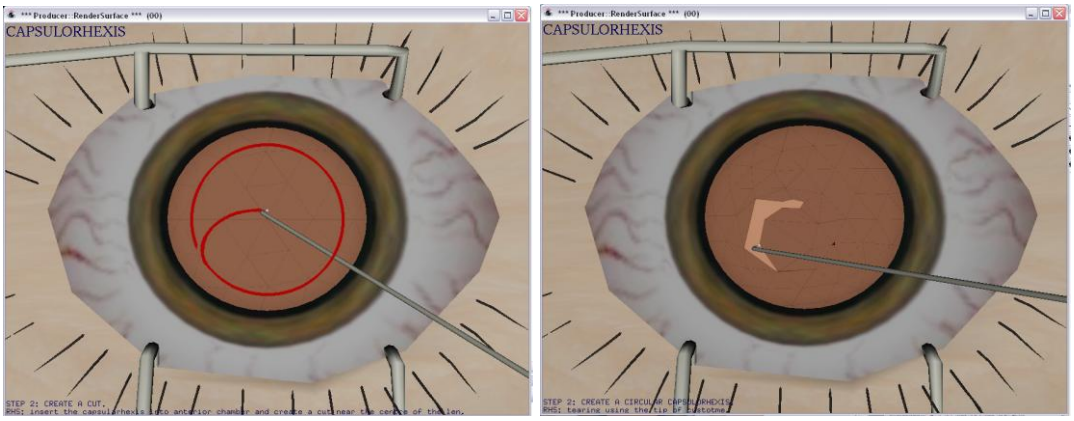


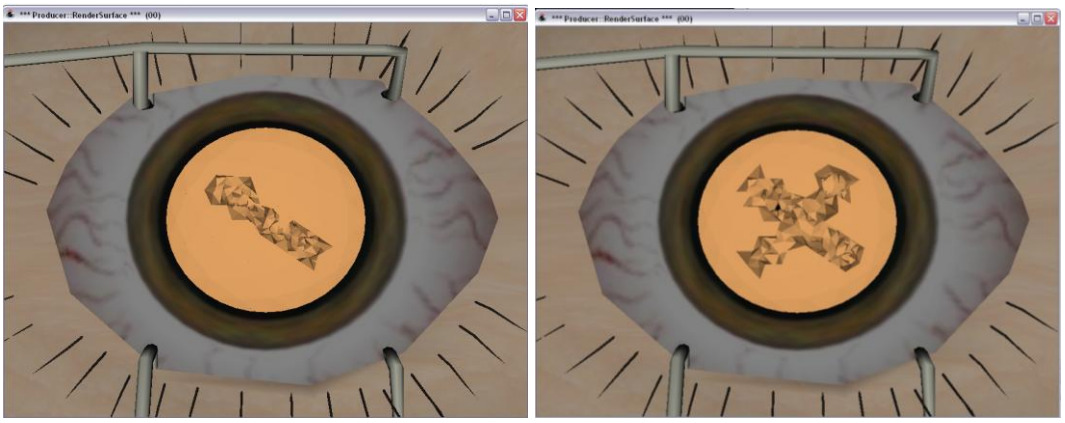
Fig. 7 Simulation of nucleus rotation.



(a)



(b)



(c)

Fig. 8 Simulation of phacoemulsification procedure: (a) corneal incision (b) capsulorrhexis (c) phaco-sculpting. The red lines are superimposed as visual cues to guide the trainees.

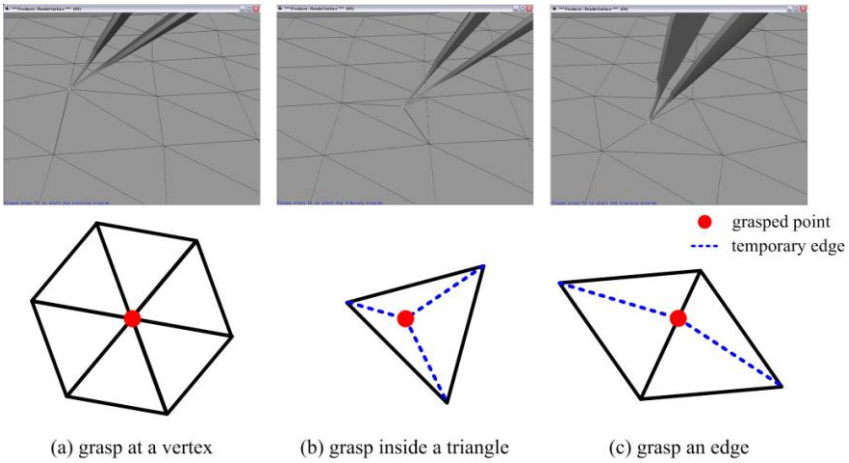


Fig. 9 Grasping surface mesh (a) at a vertex, (b) at a point inside a triangle, (c) at a point on an edge

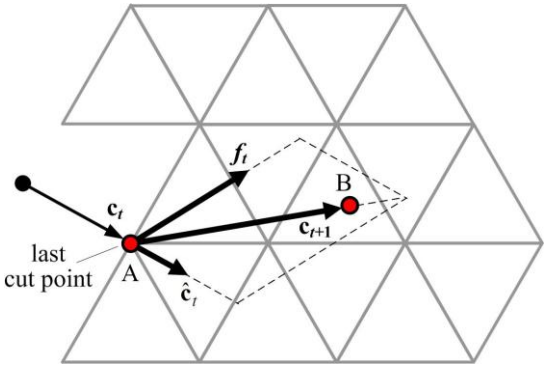
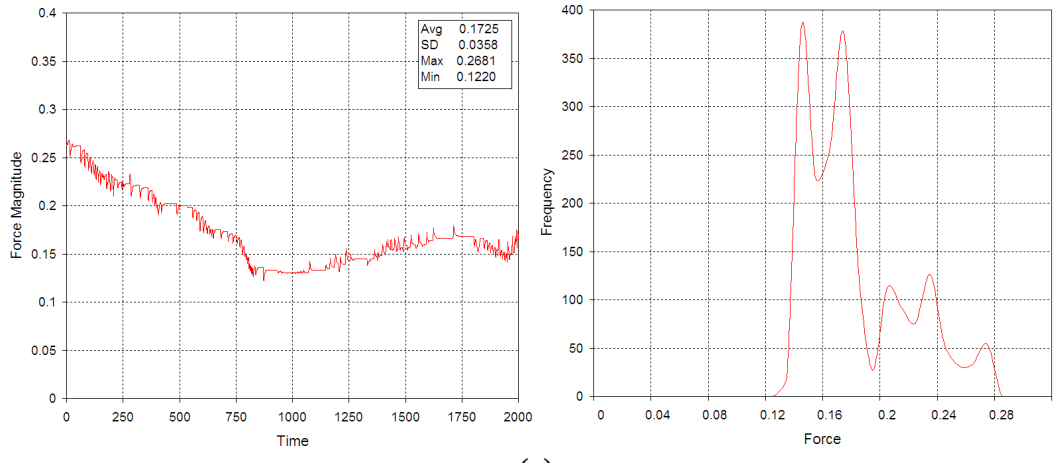
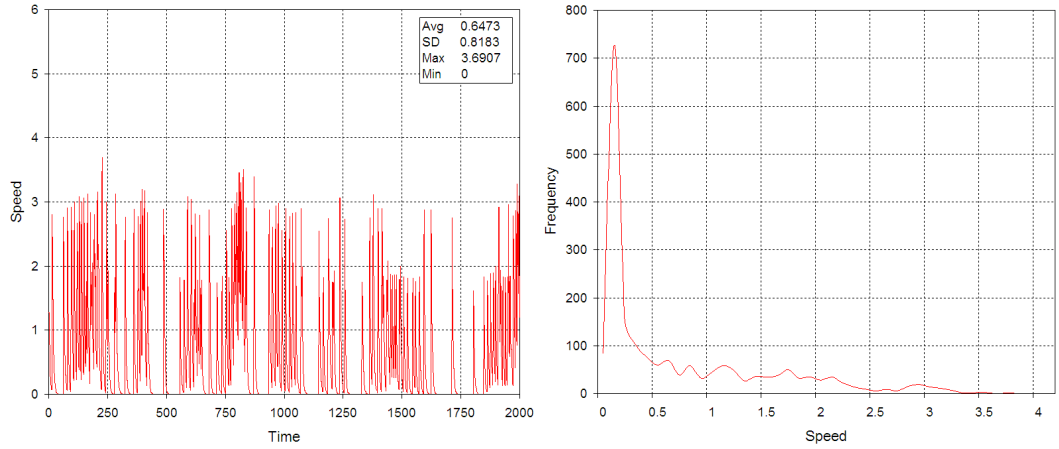


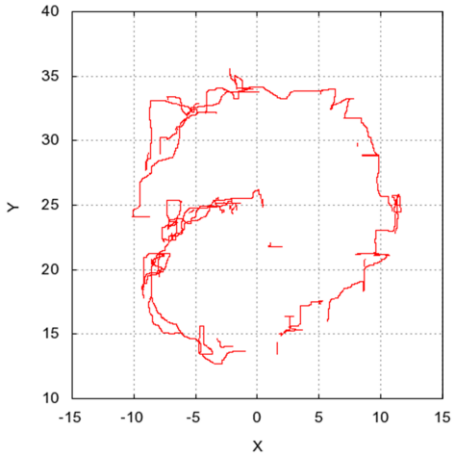
Fig. 10 Determine the location of the next cut point $\mathbf{c}_{t+1}$ due to tearing by using the cut segment vector $\mathbf{c}_t$ and the force $\mathbf{f}_t$. The next cut point B is snapped to the nearest mesh primitives, i.e. vertices or edges, if it is close to either of them.



(a)



(b)



(c)

Fig. 11 (a) Contact force magnitude (N) recorded over time and its distribution (b) Speed of the haptic stylus endpoint (mm/s) recorded over time and its distribution (c) Trajectory of tool-tissue contacts during capsulorrhexis.

c_t \ c_{t+1}	inside triangle	on edge	at vertex
inside triangle	IC_1	IC_4	IC_5
on edge	IC_4	IC_2	IC_6
at vertex	IC_5 (or SC_1)	IC_6 (or SC_2)	IC_3 (or SC_3)

IC_i and SC_i denote the cutting algorithms for initial cut and subsequent cut respectively

Table 1 Mapping of canonical cases of cutting involving one or two triangles, based on the location of the current cut point c_t and the new cut point c_{t+1}. Since SCs always begin from a vertex, the corresponding algorithms are the same as that of the ICs (last row). Cutting involving more than two triangles are decomposed into a combination of the cases in this table, and simulated by using the corresponding algorithms.

A summary of the paper

This paper presents the development of a virtual cataract surgery simulator for the training of the operative procedures in phacoemulsification, with attention to the procedural training corneal incision, capsulorrhexis and phaco-sculpting, and the collection of quantifiable information during the virtual training. Following a brief review of the procedures in real cataract surgery with phacoemulsification, the system framework of the virtual training simulator and the construction of the geometric models involved are described. Trainees are provided with haptic force feedback through a pair of generic, commercially available haptic devices. Both surface-based and volumetric meshes are used to model the objects in the virtual environment. The core of the training system is the simulation engine that simulates the response of virtual tissues interactively upon user's manipulation, e.g. deformation, cutting, tearing and sculpting. Computationally efficient methods are adopted to ensure real-time performance. Mass-spring model is employed to simulate tissue deformation by propagating the externally applied forces from the contact points to the internal mass points via interconnecting springs. Instead of using uniform springs for virtual tissue models, the stiffness of spring is set with a more accurate approach that takes the areas of triangles sharing a common spring into account. To simulate the subdivision of triangulated surface mesh during cutting and tearing, the various ways that the mesh topology is modified are generalized into canonical cases so that an arbitrary cut can be modelled routinely by transitions from one case to another. Phaco-sculpting is simulated by removing tetrahedrons progressively from the tetrahedral mesh constructed to model the clouded lens. Bounding boxes arranged with hierarchical octree are employed to

accelerate the simulation and to enable the use of high resolution tetrahedral mesh. User performance during the training is measured quantitatively in terms of such metrics as completion time and deviations from optimal positions, as well as contact force, speed and the trajectory of tool-tissue contact, which can serve as objective information to evaluate the manual dexterity of the trainees and their skill levels. As the proposed system does not involve intensive computation, hardware configuration involving a generic personal computer with standard haptic devices is sufficient for the implementation, which provides a feasible and inexpensive virtual training method that can be popularized for learning cataract surgery with phacoemulsification.