

Evangelos Papadopoulos · Alkiviadis Tsamis  
Kostas Vlachos

## Development of a real-time visual and force environment for a haptic medical training simulator

Received: November 1, 2006 / Accepted: April 20, 2007

**Abstract** In this article, a real-time, visual and force environment for a 5-dof haptic urological training simulator is presented that deals with a low-force, high-deformation environment. A real-time graphical representation of the male urethra during the insertion of an endoscope is developed. Smooth urethra deformations are produced by a mesh of piece-wise Bézier interpolations, while its inner wall is simulated by realistic tissue textures. Efficient real-time techniques are developed that introduce endoscope camera depth-of-field effects. A novel particle-based model computes in real-time the forces fed to the haptic device. A 13 fps refresh rate is achieved on a 2-GHz computer with the depth-of-field effect activated, while the rate is doubled to 26 fps with this feature disabled. It is expected that the simulator will contribute to ethical, efficient, and modern surgical training.

**Key words** Graphical training simulator · Force model · Haptics

### 1 Introduction

Minimally invasive surgical (MIS) operations are widely used as they offer significant advantages over established open procedures. However, the surgeons in MIS operations face several difficulties, such as limited depth perception and field of view, inaccurate tool position and orientation, and inadequate force feedback. To compensate for these, many hours of training, are needed, mostly with animals in vivo. However, such training is limited for ethical and economic reasons. These limitations call for medical training

simulators, which are less expensive, reduce animal use, and result in faster training.

In general, training simulators consist of a force feedback haptic device, a visual feedback system, and a coordinating control system. For example, the *VIRGY* is an endoscopic surgery simulator using the *PantoScope* haptic device.<sup>1</sup> The *Karlsruhe Endoscopic Surgery Trainer* and the *Virtual Endoscopic Surgery Trainer* are laparoscopic simulators using the software *KISMET*, the *Phantom* haptic device, or the *HIT Force Feedback Device*.<sup>2</sup> The few commercial urological simulators that exist, such as the *UroMentor* from Sionix or the *Bristol TURP Trainer* from Limbs and Things, are based on mannequins of the human anatomy, and have limited haptic feedback or even no simulated visual feedback.

While the ability to interact with a virtual environment through the feel of forces is important, a realistic visual representation of the human anatomy and tissue deformation is also of great importance. Soft-tissue modeling methods can be divided into the *geometry-based methods* (GbM) and the *physics-based methods* (PbM). The GbM are fast and result in smooth visuals using a separate *force-model* for computing forces. Here, the user displays deformations either by manipulating surface vertices directly (*vertex-based methods*),<sup>3</sup> or by manipulating a lattice of control points (*spline-based methods*) to produce smoother deformations.<sup>4</sup>

The PbMs include interaction dynamics for increased realism, but are computationally expensive and not always suitable for real-time applications. These PbMs are further classified as *particle-based*, *finite element (FEM)*, and *meshless* methods. The first model the soft tissue as a group of point masses connected to each other through a network of springs and dampers.<sup>5,6</sup> Such an approach was used for soft-tissue modeling, using known reference models.<sup>7</sup> Two 2-dimensional deformable mass-spring-damper models were employed for a combined graphic and haptic display of soft-tissue cutting.<sup>8</sup> A particle-based model was proposed to model the interactions of pulpal tissue with deformable tools in an endodontic simulation.<sup>9</sup> A mass-spring system has recently been used for real-time graphic simulations of

E. Papadopoulos (✉) · A. Tsamis · K. Vlachos  
Department of Mechanical Engineering, National Technical  
University of Athens, Heron Polytechniou 9, 15780 Athens, Greece  
Tel. +30-210-772-1440; Fax +30-210-772-1445  
e-mail: egpapado@central.ntua.gr

soft organs and tissues.<sup>10</sup> In FEM, a 3D object volume was discretized into finite elements, and several simplifications were employed to implement FEM in real-time.<sup>11–13</sup> To reduce the complexities introduced by mesh generation, the *meshless finite spheres* (MFS) method has been developed and applied in the cutting<sup>14</sup> and modeling of soft biological tissues.<sup>15</sup> Hybrid models that combine geometry and physics-based approaches give the best real-time results.<sup>16</sup>

In this article, a real-time visual and force environment for a 5-degree-of-freedom (dof) haptic urological training simulator is presented that deals with a low-force, high-deformation environment. Real-time graphics simulate the deformation of the male urethra due to endoscope insertion. A lattice of piece-wise Bézier interpolations is designed for the representation of smooth urethra deformations. The inner urethra wall is covered up by realistic tissue textures, while the camera depth-of-field effect, supported by some recently introduced endoscopes, is simulated by efficient real-time techniques. A novel particle-based model calculates in real-time the forces and torques applied to a trainee's hand through an endoscope attached to the haptic device. Real-time graphics achieve a 13 fps refresh rate on a 2-GHz computer with the depth-of-field effect activated, while disabling this feature doubles the rate. It is expected that such simulators will contribute to ethical, efficient, and modern surgical training.

## 2 Definition of requirements

The aim of this work is (a) to reproduce graphically the visual feedback that a surgeon receives during an actual urological operation, (b) to develop a mathematical model for calculating the forces and torques felt, and (c) to integrate the above in a training simulator. To this end, we observed in vivo urological operations on a human and on an animal (Fig. 1a).

A urological operation can be divided into two phases, the *insertion* and the *main* operation.<sup>17</sup> During insertion, the surgeon moves the endoscope tip on the patient's plane of symmetry,  $z$ - $y$ , along the urethra path from the insertion

point A to the final point C in the bladder, via an intermediate point B (Fig. 1b). At B, the endoscope is rotated, so as to align the urethra and avoid traumas. During the main operation, the endoscope rotates in three directions with minimal translation.

It was found that the haptic interface should have 5 dof, while it should carry the endoscope tip along the 20-cm ABC path in Fig. 1b, and achieve displacements of 10 cm along the  $y$  and  $z$  axes, rotations of  $\pm 180^\circ$  around the endoscope's  $z'$  axis, and rotations of  $\pm 30^\circ$  about the  $x'$  and  $y'$  axes.<sup>17</sup> The maximum endoscope forces and torques measured are quite small, i.e., 4.5 N along the  $z$  and  $y$  axes, 150 mNm about the endoscope's fixed  $y'$  and  $x'$  axes, and 10 mNm about the  $z'$  axis.

## 3 Virtual tissue

Surgeons have indicated that the quality of the graphics and real-time realism are of paramount importance. To select the proper methodology for modeling the highly deformable tissue (urethra) in real time, a number of techniques were evaluated. Due to limitations in computational power, the FEM methods are not yet mature enough for real-time applications. MFS has advantages when cutting is involved, but is relatively expensive. However, since cutting was not required in the operations considered here, while real-time execution was of paramount importance, the MFS was not considered. As a compromise between speed and realism, a hybrid methodology was developed in which smooth and large deformations are produced by parametric cubic interpolations, while the forces are computed separately.

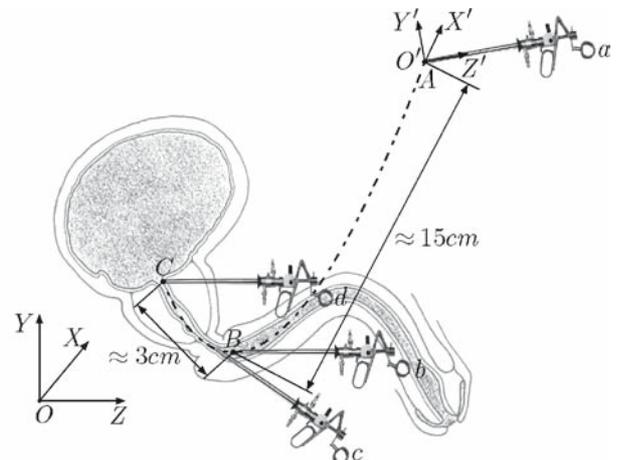
### 3.1 Geometry of nondeformed urethra

The study of typical endoscope paths revealed that the path's center line (CL) can be described by a parabola such as  $y = 0.13z^2$ . The urethra's cross section is assumed to be circular, with a constant radius in the undeformed state. The urethra's surface is constructed by parametric cubic curves,

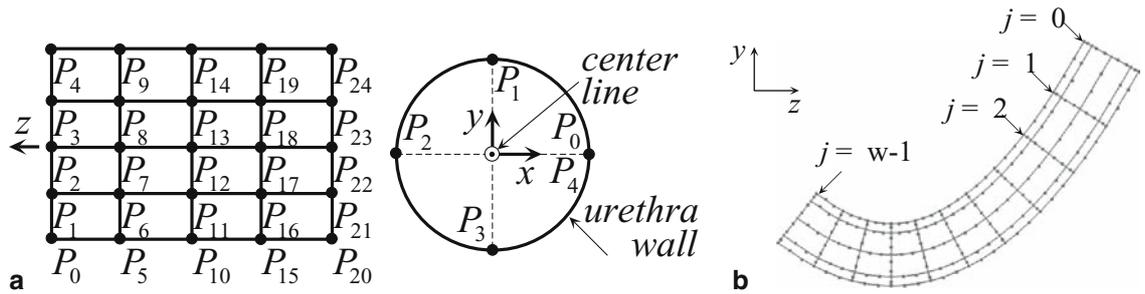
**Fig. 1.** **a** In vivo urological operation on an animal. **b** Endoscope path during a urological operation



**a**



**b**



**Fig. 2.** **a** Simplified representation of the surface discretization (*left*) and the 1st peripheral section (*right*), for  $n = 5$  and  $w = 5$ . **b** Nondeformed urethra for  $n = 9$ ,  $w = 11$ , and an arbitrary choice of radius

which pass through certain points. The points are chosen to be equidistant, and are obtained by discretizing an orthogonal surface in both the peripheral and axial direction (Fig. 2a). The urethra is produced by orthogonal surface bending about its  $z$  axis to form a cylinder. This results in points  $P_0$ ,  $P_5$ ,  $P_{10}$ ,  $P_{15}$ , and  $P_{20}$  coinciding with  $P_4$ ,  $P_9$ ,  $P_{14}$ ,  $P_{19}$ , and  $P_{24}$ , respectively. The points  $P_0, \dots, P_4$  define the 1st peripheral section, the next five points define the 2nd peripheral section, etc. (Fig. 2a). The number of points that define each section is assigned the parameter value  $n$ , while the number of sections is assigned the parameter value  $w$ . In Fig. 2a,  $n = 5$  and  $w = 5$ .

A parametric cubic curve must pass in real time through points  $P_i$  in both directions. To produce smooth deformations, we must maintain at least tangential continuity at the interpolating points. Out of the parametric cubic curves, only the piece-wise Bézier or Hermite interpolations satisfy the above requirements.<sup>18</sup> The Bézier form is chosen because the tangent direction and magnitude are better depicted by means of control points. Each quadrilateral surface in Fig. 2a, for example the one defined by points  $P_6$ ,  $P_{11}$ ,  $P_{12}$ , and  $P_7$ , is formed by four Bézier segments. The control points,  $CP_i$ , that define the shape of the Bézier curves are calculated by the Catmull–Rom method, which can represent closed curves with an appropriate choice of the first and last inner control points.<sup>19</sup> The knot sequence is created using the chord length method, which produces small curvature variations giving smooth deformations.<sup>19</sup>

To obtain the geometry of Fig. 1b, the centers of the peripheral sections must lie on the urethra CL. The section planes are rotated properly about the  $x$  axis, so that they are normal to the tangent of the parabola at the corresponding center position. Combining the piece-wise Bézier interpolation in both directions, the geometry in Fig. 2b is produced, where section centers are denominated as  $j=0, \dots, (w-1)$ , and are equidistant along the  $z$  axis.

### 3.2 Urethra deformation states

During endoscope insertion, the urethra is deformed for three reasons. (a) The endoscope diameter is greater than the urethra's, (b) the endoscope is rigid, and (c) the endoscope position and orientation are variable. To be able to represent (a) accurately, the peripheral sections are divided

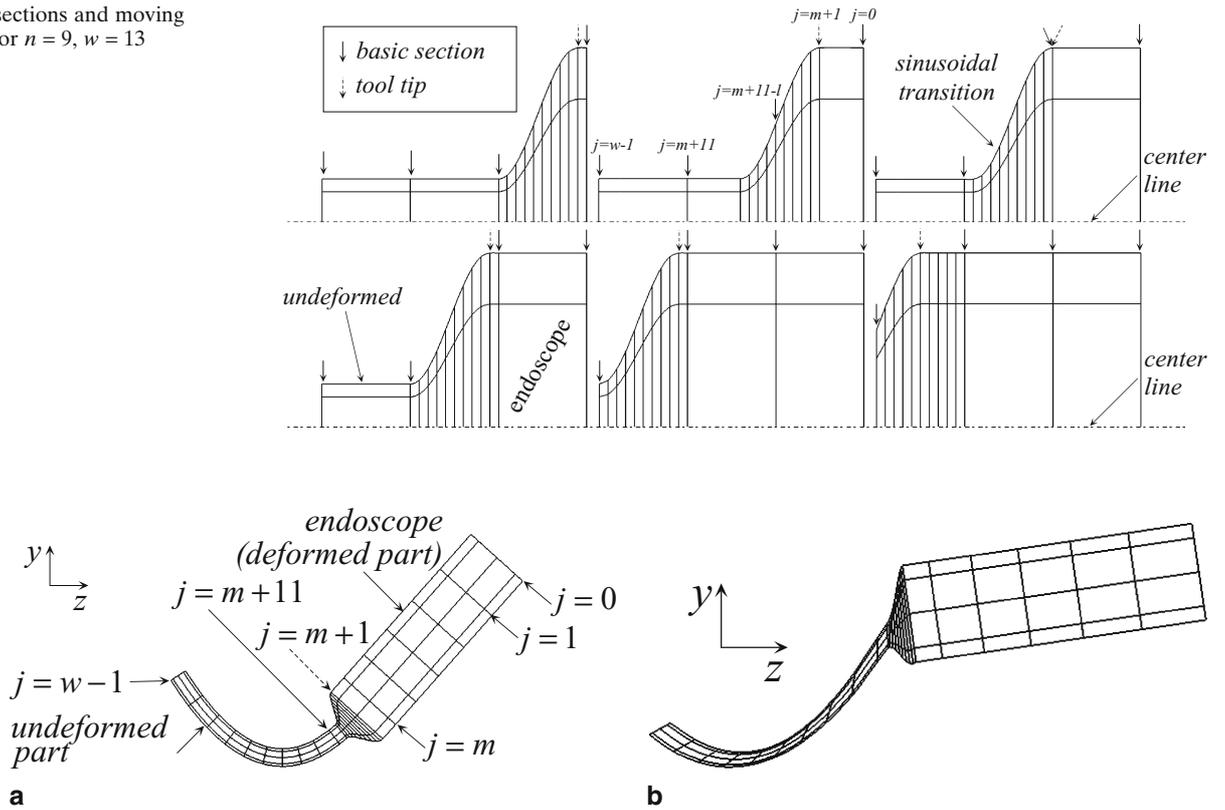
into two sets. The first set, called set of *basic sections*, defines the global urethra geometry. Sections in this set are fixed, and do not move along the urethra CL. The second set, called the *moving deformation*, represents the deformation of the urethra close to the endoscope tip. These sections are spatially dense, move along the urethra CL, contain a basic section without displacing it along the urethra CL, and define the transition from the endoscope diameter to the urethra diameter during insertion. The endpoints of the axial Bézier segments within the moving deformation are chosen to form a sinusoidal profile, allowing for smooth attachment of the deformation on the endoscope and the urethra. Note that second degree polynomials, such as those used by Basdogan et al.,<sup>3</sup> cannot be used here due to tangent discontinuities at one of the two ends. The moving deformation passes through the basic sections without displacing them along the urethra CL. Implementation results are shown in Fig. 3, where the propagation of the deformation of the urethra due to endoscope insertion with respect to the CL is shown.

The endoscope radius is assumed to be four times larger than that of the undeformed urethra. The moving deformation is described by ten subsections. Two integer parameters are used to define which snapshot is depicted at each time. Considering that only basic sections exist, as happens, for example, in Fig. 2b, the parameter  $m$  is employed, taking values in the range  $0, \dots, (w-1)$  and defining that the endoscope tip is between the basic sections  $j = m$  and  $j = m + 1$ . Another parameter,  $l$ , ( $l = 1, \dots, 10$ ), defines the specific position of the tip within the moving deformation for a given value of  $m$ . Globally now, when all peripheral sections coexist, the basic sections are denominated by  $j = 0, 1, \dots, m, m + 11 - l, m + 11, m + 12, \dots, w - 1$ , while the sections of the moving deformation are denominated by  $j = m + 1, m + 2, \dots, m + 10$ .

Due to (b), the part of the urethra that is expanded by the endoscope must acquire its geometry. At first, the centers of the sections that belong to the expanded part must lie on the endoscope's straight axis, while the planes of these sections must be normal to it. Combining the choices that satisfy (a) and (b) results in Fig. 4a, produced with arbitrary parameter values. All sections are denominated by  $j=0, \dots, (w-1)$ .

According to (c), the endoscope may not move along the parabolic CL or be tangential to it at the point where defor-

**Fig. 3.** Basic sections and moving deformation for  $n = 9$ ,  $w = 13$



**Fig. 4.** For  $n = 9$ ,  $w = 26$ ,  $m = 5$ ,  $l = 7$ . **a** The endoscope moves on a parabolic tangential orientation. **b** Arbitrary position and orientation

mation occurs. In this case, the sections ahead of the endoscope tip must be translated and rotated about the  $x$  axis to follow trainee manipulations, such that the sections close to the tip are affected more than the ones far from it. Section  $j = w - 1$  is rotated whereas its center is not translated, because this section represents the fixed bladder entrance. Also, the sections in the moving deformation should undergo the same translation and rotation, otherwise they may intersect with one another.

The deviation of the endoscope position  $dy$  and orientation  $d\theta_x$  with respect to the corresponding values on the undeformed CL are measured. These can be distributed linearly in the sections  $j = (m + 11), \dots, (w - 1)$  according to the above constraints. Translation of the last basic section  $j = w - 1$  is always zero. However, the rotation angle of this section is equal to  $d\theta_x$  divided by the number of downstream basic sections  $j = m + 11, m + 12, \dots, w - 1$ , when  $l = 1$ , and equal to  $d\theta_x$  divided by the number of downstream basic sections  $j = m + 12, m + 13, \dots, w - 1$ , when  $l = 10$ . The rotation angle increases linearly from one value to the other as  $l$  takes greater values in the range  $1, \dots, 10$ . The result is depicted in Fig. 4b for an arbitrary position and orientation of the endoscope.

The endoscope may also be rotated by an angle  $d\theta_y$  about the  $y$  axis. In this case, the centers of sections  $j = 0, \dots, (m + 1)$  must be translated and rotated by an angle  $d\theta_y$ . Also, the sections within the moving deformation should undergo the same amount of rotation  $d\theta_y$  to avoid intersecting with one another. Finally, angle  $d\theta_z$  is linearly distributed to

the remaining downstream sections, as happens with angle  $d\theta_x$ .

### 3.3 Texture mapping

The tissue view of the inner urethra wall is simulated by textures taken from medical databases. However, a texture may only cover flat quadrilateral surfaces. For this reason, each concave quadrilateral surface in Fig. 2b is linearly approximated by a finite number of flat quadrilateral surfaces using the Coons method.<sup>19</sup> Texture is then applied separately to each one of the finite flat quadrilateral surfaces.

The visual deformation of the urethra during endoscope insertion is implemented in OpenGL, which is a very portable and fast low-level 3D graphics and modeling API.<sup>20</sup> A few deformation states, as shown by the virtual endoscope camera with  $n = 17$  and  $w = 26$ , are depicted in Fig. 5a–c. In this figure, the endoscope tip is (a) at the urethra entrance, (b) rotated upwards about the  $x$  axis, and (c) rotated left about the  $y$  axis.

## 4 Depth-of-field effect

The depth-of-field effect, supported by some recently introduced endoscopes, reflects the lens model used in real-world cameras and is important for photo-realistic rendering,<sup>21</sup>

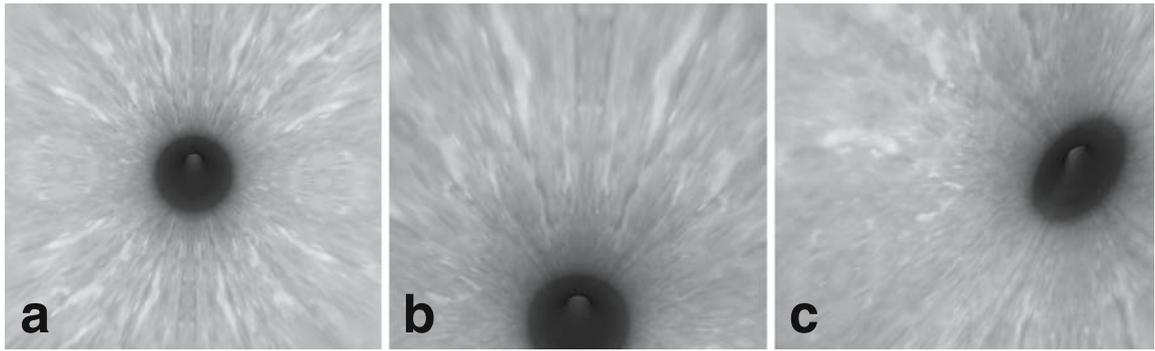


Fig. 5. Deformation states

providing a better sense of relative distances. Scenes viewed through a camera typically appear in focus only for a limited depth range. Outside this range, points appear blurry, with increasing blur for points farther in front or behind the focused area. On the other hand, virtual scenes viewed through a standard graphics pinhole camera model appear sharp at all depths. By failing to reproduce the depth-of-field effect, these applications omit a valuable visual depth cue. To restore the camera-focus effect, the approach by Potmesil and Chakravarty,<sup>21</sup> was evaluated. It was found that this approach is not suitable for real-time applications, due to computational power limitations. Hence, it was decided to develop a simplified technique using OpenGL library<sup>20</sup> tools.

#### 4.1 Development of scene blurriness

Looking at the problem optics, an out-of-focus point is projected onto the image plane as a circular disk (*circle of confusion*) rather than as a point. The circles of confusion are caused by the aperture size. A greater aperture leads to a smaller depth of field and more light. However, no apertures exist in computer graphics. So the blurriness had to be introduced artificially.

OpenGL has two appropriate tools to develop scene blurriness depending on the distance from the viewer,<sup>20</sup> *mipmapping* and *color blending functions*. Mipmapping uses textures with multiple images of different resolution, called *mipmapped textures*, which can be applied at points near or farther in front of and behind the focused area according to the desired blurriness. The larger the number of varying resolution textures, the better the visual result. However, the latter carries a severe price in speed, and reduces the available frames per second drastically. To avoid these problems, color blending functions are employed, which are used for transparency effects and are easily applicable without requiring high computational power. The color blending functions support effects such as transparency that help us add the degree of realism required. The blending functions are applied to the source color, i.e., the color of the object that is to appear transparent, and the destination color, i.e., the color of a nontransparent scene behind the transparent object. The results of the blending functions are overlapped to generate the new color value

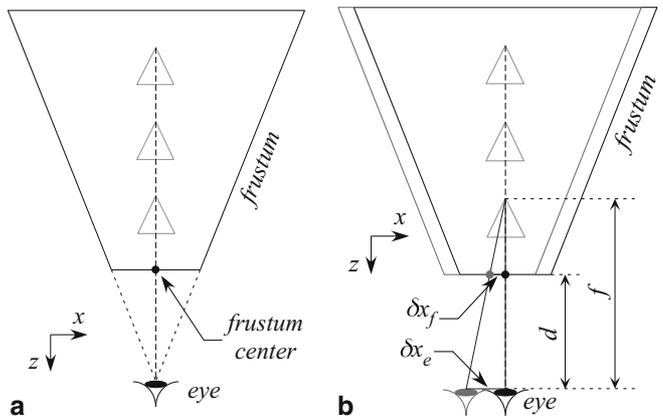


Fig. 6. **a** Central location of the eye and frustum. Object vertices project to the frustum center. **b** Frustum location adjustment

put on the screen. To produce the necessary blurriness, the source color is scaled by the alpha component,  $A$ , and added to the destination color, which is scaled by  $1-A$ :

$$R_N = R_S A + R_D (1 - A) \quad (1)$$

$$G_N = G_S A + G_D (1 - A) \quad (2)$$

$$B_N = B_S A + B_D (1 - A) \quad (3)$$

where  $R_N$ ,  $G_N$ , and  $B_N$  are the new color values put on screen,  $R_S$ ,  $G_S$ , and  $B_S$  are the source color values, and  $R_D$ ,  $G_D$ , and  $B_D$  are the destination color values. The alpha value ranges from 0 for the completely transparent to 1 for the completely opaque.

#### 4.2 Scene topology design

In order to develop depth-of-field blurriness using transparency effects, a proper scene topology must be designed. However, the OpenGL technique is not real-time. Therefore, the technique is modified here to increase its execution speed and achieve a real-time response. To do this, consider the simple scene geometry depicted in Fig. 6a. The viewing volume is the black trapezoid, called the *frustum*, which is a pyramid section viewed from the narrow end to the broad end. The frustum contains three triangular objects. The black dashed line denotes the projection of some vertices. In Fig. 6a, the vertices project to the center of the frustum.

According to theory, if we desire to focus on an object, e.g., the nearest one, for eye movement in the  $x$  direction, we adjust the location of the frustum so that the vertex of that object projects to the frustum center (see the red set-up in Fig. 6b). If  $f$  is the desired focus distance and  $dx_e$  is the eye displacement in the  $x$  direction (see Fig. 6b), then the frustum displacement in the  $x$  direction,  $dx_f$ , is given by

$$\delta x_f = (f - d)(\delta x_e / f) \quad (4)$$

where  $d$  is the distance from the eye to the near clipping plane. A similar thought applies for an eye displacement in the  $y$  direction,  $dy_e$ . In this case, the frustum displacement  $dy_f$  is

$$\delta y_f = (f - d)(\delta y_e / f) \quad (5)$$

Each set of  $(dx_f, dy_f)$  and  $(dx_e, dy_e)$  corresponds to the same scene rendered each time from a different point of view. The image for the *central eye location* (*cel*), for which  $dx_e = dy_e = 0$ , is called the *central image*. For  $dx_e \neq 0$  or  $dy_e \neq 0$ , the image coincides with the central image at the focus distance, but diverges from it in front of and behind the focus distance.

### 4.3 Eye locations

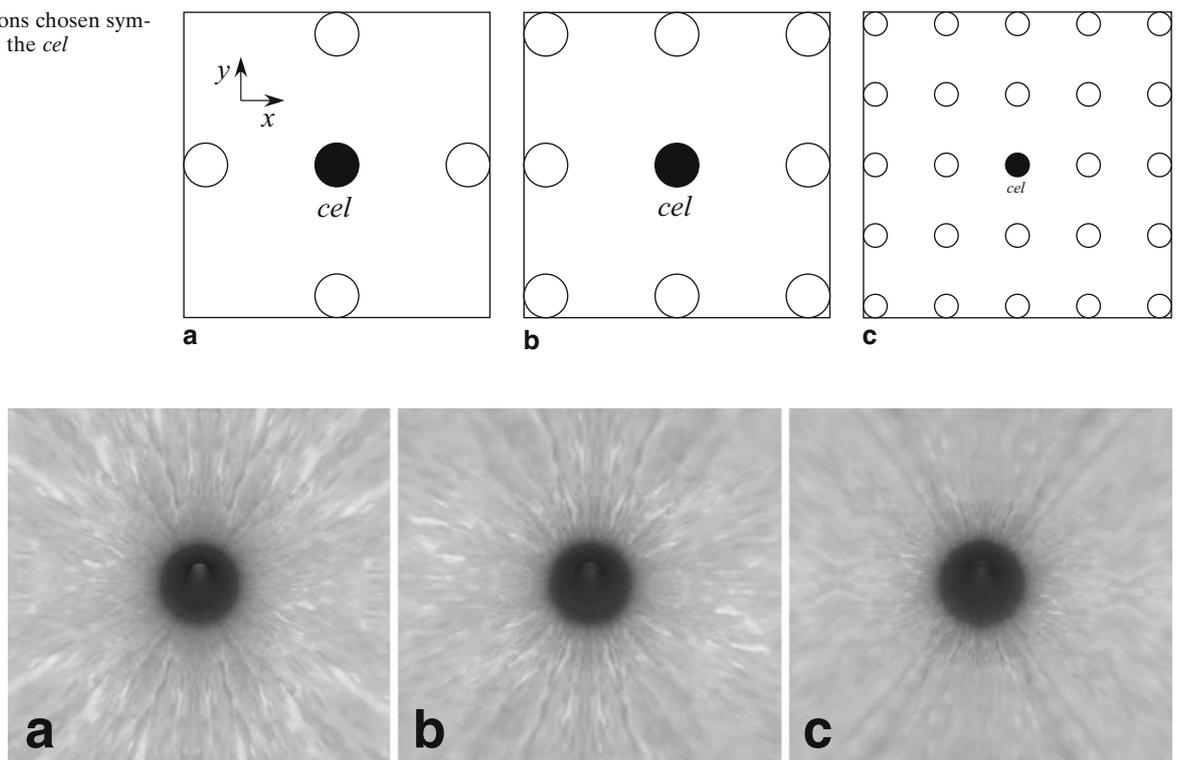
Next, we focus our attention on the following issues: (i) where to locate the eye with respect to the *cel*, (ii) how many eye locations (and images) to choose, and (iii) in what way should the images be combined in real time to produce the

desired blurriness. The existing constraint for interactive work does not allow for the rendering of many images. It was found that in this case, choosing four eye locations symmetrically around the *cel* provides the necessary degree of realism and still allows for a real-time response. A useful technique is to develop the symmetrical pattern depicted in Fig. 7. One may choose four, eight, or more eye locations to improve the visual result, at the expense of increased computational time.

To render the central along with the divergent images and produce the depth-of-field effect, one may use the *accumulation buffer* technique.<sup>20</sup> The accumulation buffer automatically renders the divergent images with an appropriate degree of transparency. However, using this buffer carries a price in speed, since the scene is drawn on the screen for each image separately, before the result is copied to the frame buffer for the final rendering. Therefore, this technique is not suitable for real-time applications.

Here, we develop a faster method, in which separate image drawing is avoided, and instead the result is copied directly to the frame buffer for one rendering per time only. To accomplish this, we draw one scene consisting of several different objects (central image and divergent images) in different positions. To produce the depth-of-field blurriness, the above-mentioned color-blending functions are employed. Setting the alpha component  $A = 1$  for the central image (completely opaque), choosing  $A = 0.3$  for the divergent images (transparent), and applying the symmetrical pattern of Fig. 7a or 7b, the depth-of-field effects shown in Fig. 8b and 8c, respectively, are produced for a focus distance  $f = 1.7$ .

**Fig. 7.** Eye positions chosen symmetrically around the *cel*



**Fig. 8.** **a** Pinhole camera model. The scene is sharp at all depths. **b** Depth-of-field effect after applying the symmetrical pattern of Fig. 7a. **c** Improved effect after applying the symmetrical pattern of Fig. 7b

## 5 Force model

In addition to the visual feedback described, the simulator must calculate the forces and torques that will be applied to a trainee's hand through an endoscope attached to a haptic device.<sup>17</sup> This is done using endoscope position feedback from the haptic device and a force model that represents the behavior of the tissues involved. Therefore, this model is developed with a focus on execution speed, and basic elements that account for tissue spring-damper-resistance effects.

### 5.1 Total resistance force along the endoscope axis

The total resistance force along the endoscope axis,  $F_{re}$ , is given by

$$F_{re} = F_1 + F_2 + F_3 \quad (6)$$

where  $F_1$  is Coulomb friction,  $F_2$  is viscous friction, and  $F_3$  is wall resistance force. Since the endoscope does not move out of the human plane of symmetry  $y$ - $z$ , only the projections of  $F_{re}$  on the plane  $y$ - $z$ , namely  $F_{re,y}$  and  $F_{re,z}$ , need to be calculated.

The Coulomb friction,  $F_1$ , is calculated by

$$F_1 = \mu N (V_p/V) \text{sgn}(V) \quad (7)$$

where  $V$  is the relative linear velocity between the endoscope and the tissue,  $\mu$  is the friction coefficient associated with the normal to the endoscope axis force  $N$ ,  $V_{\parallel}$  is the component of  $V$  that is parallel to the endoscope axis, and  $\text{sgn}(V)$  is the sign of the velocity  $V$ . To compute the normal force  $N$ , a virtual spring model is established on each basic section. To increase the speed of force computations, springs between the same and neighboring section points, such as those proposed by Brown et al.,<sup>5</sup> were ruled out, because the felt force here is mostly due to frictional effects and not to springiness along the urethra. Instead,  $n - 1$  springs connecting points  $P_i$  on each basic section with the urethra CL are established (Fig. 9a for  $n = 5$ ). The initial length of each spring is equal to the radius of the undeformed urethra. Springs are elongated during endoscope insertion, reaching its radius progressively. The normal force  $N$  is the sum of the forces due to the springs on sections behind the endoscope tip, and at some percentage of the basic section in the moving deformation. The latter increases with  $l$ .

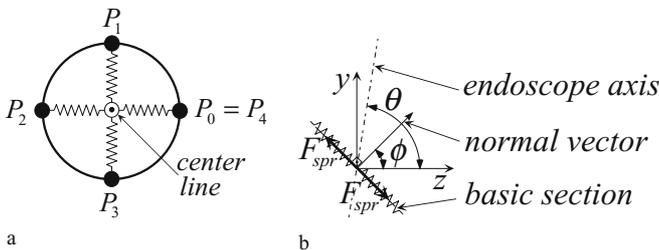


Fig. 9. **a** Coulomb friction springs. **b** Contribution to resistance

The viscous friction  $F_2$  is calculated by

$$F_2 = b_{V_{\parallel}} V_p \quad (8)$$

where  $b_{V_{\parallel}}$  is the viscous friction coefficient in the direction of  $V_{\parallel}$ , defined in Eq. 7. Force  $F_3$  is a pure spring-type resistance, produced by the Coulomb friction springs on the basic sections in front of endoscope tip, and calculated by summing the projections of spring forces,  $F_{spr}$ , along endoscope axis (Fig. 9b).

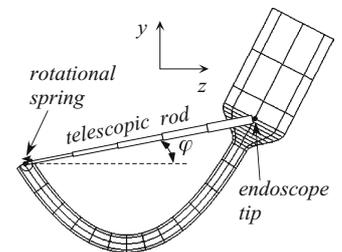
### 5.2 Resistance torque components

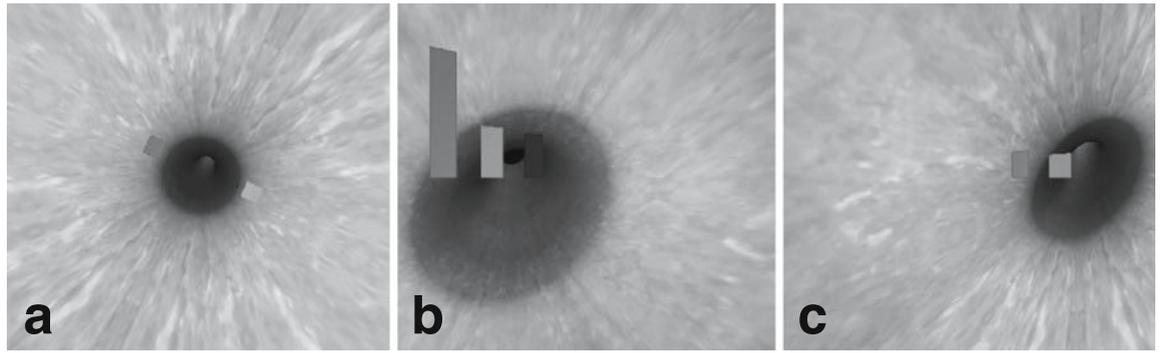
A torque with three components, namely  $T_x$ ,  $T_y$ , and  $T_z$ , acting about the  $x$ ,  $y$ , and  $z$  axes, respectively, is applied to the endoscope by the tissues. Dividing the urethra into two parts, of which the first is outside the body and free to move, while the second is inside and constrained, then  $T_x$  appears when the endoscope is (a) in the constrained part, and/or (b) off the parabolic CL (endoscope  $y$ - $z$  translation). To compute the first part of  $T_x$ , one could use virtual springs connecting each basic section center to a virtual wall point off the urethra,<sup>6</sup> in case of small basic section center displacements on its plane. However, here these sections undergo large rotations and displacements on the  $y$ - $z$  plane. To tackle this, a rotational spring is placed at the endoscope tip to produce a torque due to  $x$  rotations. The spring constant is set to zero when the endoscope is at the free part, and increases linearly according to the endoscope position at the constrained part. The spring constant is then multiplied by the angle of the endoscope orientation with respect to the tangent of the parabolic CL at the deformation point. To compute the part of  $T_x$  due to endoscope  $y$ - $z$  translation, a technique is developed that is faster and easier to implement than the one described in Papadopoulos et al.<sup>22</sup> A rotational spring accounting for the torque about the  $x$  axis is placed at the last basic section center, which represents the fixed bladder entrance, and a virtual telescopic rod connects the rotational spring with the endoscope tip (Fig. 10). The constant of this rotational spring is set to a positive value. If the endoscope tip moves on the parabolic CL, the angle  $\varphi_r$  of the rod direction with respect to the  $z$ -axis, called the *spring reference angle*, is calculated by

$$\varphi_r = \tan^{-1}[(0.13z_i^2 - y_b)/(z_i - z_b)] \quad (9)$$

where  $z_i$  is the endoscope tip coordinate on the  $z$ -axis and  $(y_b, z_b)$  are the  $y$ - $z$  coordinates of the last basic section

Fig. 10. Spring model for torque  $T_x$  due to  $y$ - $z$  endoscope translation





**Fig. 11.** Colored bars representing endoscope forces and torques

center. If the endoscope tip moves out of the CL, the angle  $\phi$  of the rod direction with respect to the  $z$ -axis is calculated by

$$\phi = \tan^{-1}[(y_t - y_b)/(z_t - z_b)] \quad (10)$$

where  $y_t$  is the endoscope tip coordinate on the  $y$ -axis. Multiplying the difference  $f - f_r$  by the constant of the rotational spring placed on the last basic section center yields the part of the torque  $T_x$  due to endoscope  $y$ - $z$  translation. The total torque  $T_x$  is the sum of the two parts described above.

To calculate the torque  $T_y$ , a rotational spring producing torque about the  $y$  axis is assumed to exist at the endoscope tip. The spring constant is set to zero when the endoscope is at the free part, and increases linearly according to the endoscope position at the constrained part. The spring constant is then multiplied by the angle of the endoscope orientation about the  $y$  axis with respect to the human plane of symmetry,  $y$ - $z$ .

The torque  $T_z$  is the sum of two components

$$T_z = T_1 + T_2 \quad (11)$$

$T_1$  is the Coulomb torque friction given by

$$T_1 = \mu N V_{\perp} (r/V) \text{sgn}(V), \quad V_{\perp} = \omega r \quad (12)$$

where  $r$  is the endoscope radius,  $V_{\perp}$  is the component of  $V$  (see Eq. 7) normal to the endoscope axis, and  $\omega$  is the angular velocity about the endoscope axis.  $T_2$  is viscous torque friction

$$T_2 = b_{\omega} \omega \quad (13)$$

where  $b_{\omega}$  is a viscous friction coefficient in the direction of  $\omega$ .

Apart from feeling the forces and torques developed through the haptic interface, one can also see their magnitude on paper or on screen by using colored bars. Warning messages are also shown when the force limits for breaking the tissue are approached. The green, blue, yellow, and red bars represents  $T_x$ ,  $T_y$ ,  $T_z$ , and  $F_{re}$ , respectively. Of those,  $T_x$ ,  $T_y$ ,  $T_z$ , and the two  $F_{re}$  components,  $F_{re,y}$  and  $F_{re,z}$ , are fed to the haptics hardware. For example, in Fig. 11, the endoscope (a) moves forward on the parabola at the free urethra part and is rotated counterclockwise about its axis, so  $F_{re}$  and  $T_z$  develop, (b) moves forward on the parabola at the constrained urethra part and is rotated right about the  $y$  axis,

so  $F_{re}$ ,  $T_x$ , and  $T_y$  develop, and (c) moves below the parabola at the free urethra part and is rotated left about the  $y$  axis, so only  $F_{re}$  and  $T_x$  exist.

## 6 Graphical user interface

To use the simulator effectively, a simple and user-friendly GUI is needed that must (a) be easy to use, (b) be extendable in future simulated operations, (c) allow a user to choose the input device (haptic, if available, or keyboard for fast testing), and (d) display all the necessary numerical data. The latter is an optional feature, useful for trainees and trainers during playback and the evaluation of a session.

The GUI developed (Fig. 12) consists of three windows: (a) the control, (b) the graphics, and (c) the info window. The control window is divided into three fields. The first is the “input device” field, where the user chooses a haptic device or a keyboard. The second is the “simulation type” field, where the user chooses the kind of training simulation in which to participate. Currently, only the “male urethra model” simulation is available, with two more, the “tissue model” and the “bone model” being under development. The third field is the “control buttons,” from which the user controls the session. A “reset” button, resetting the values in the info window and the graphics start position, is also available. The graphics window contains the simulated male urethra, its deformations, and visual force feedback. The info window displays important information about the training session and is also divided into three fields. The first field shows the time elapsed since the session start, the second shows the calculated forces/torques that are applied to a trainee’s hand by the haptic device (Fig. 13), and the third shows the endoscope tip coordinates. The GUI code was developed on a Macintosh computer running OSX 10.4, using the Xcode and Interface Builder applications.

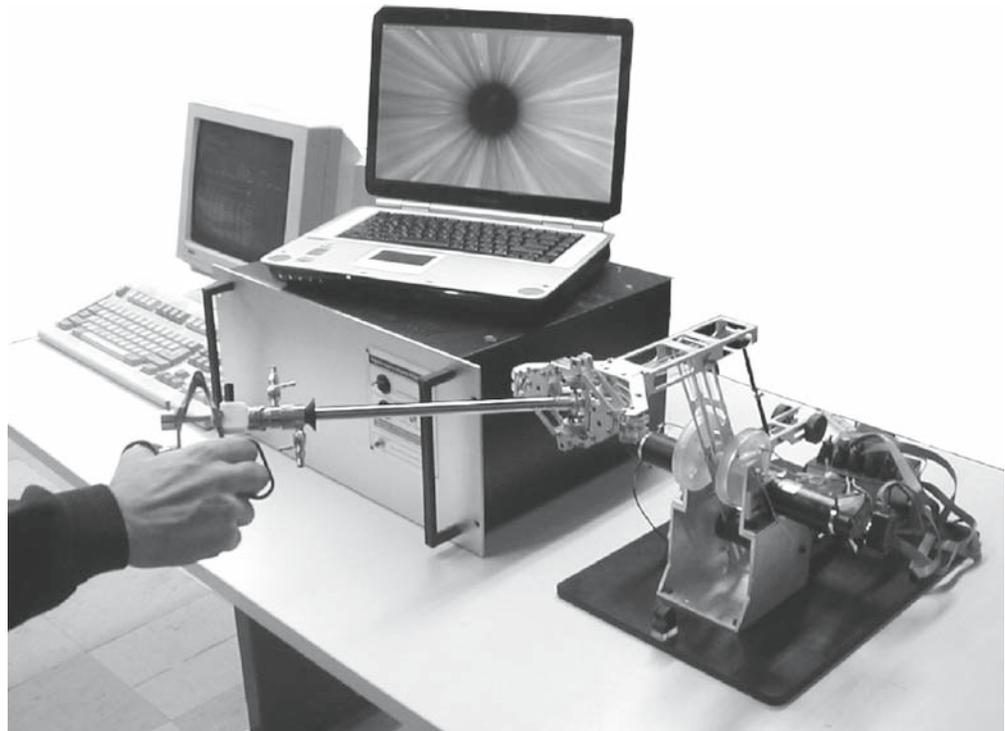
## 7 Conclusions

This article has presented a simulated visual and force system interfaced to a 5-dof haptic device, which are parts of a medical urological training simulator. The visual system



**Fig. 12.** The graphical user interface

**Fig. 13.** The training simulator shown during a testing session



includes a real-time graphical representation of the deformable male urethra during endoscope insertion, while a mesh of piece-wise Bézier interpolations was established for smooth deformations. An efficient real-time technique was developed to reproduce the camera depth-of-field effect, supported by some recently introduced endoscopes, while tissue textures from medical databases were applied for realistic tissue rendering. A novel particle-based model was developed to compute in real-time the forces and torques fed to the haptics. Real-time graphics are updated using the rendering thread method, and achieve a 13fps refresh rate on a 2-GHz computer with the depth-of-field effect activated, and twice that when this feature is deactivated. It is

expected that the environment developed will contribute to ethical, efficient, and modern surgeon training.

**Acknowledgments** This work is cofunded by the European Social Fund (75%) and National Resources (25%) – (EPEAEK II) – Pythagoras.

## References

1. Baur C, Guzzoni D, Georg O (1998) Virgy: a virtual reality and force feedback-based endoscopy surgery simulator. Proceedings of Medicine Meets Virtual Reality, San Diego, CA, IOS Press, Amsterdam, The Netherlands, pp 110–116

2. Kühnapfel U, Kuhn Ch, Hübner M, et al. (1997) The Karlsruhe endoscopic surgery trainer as an example for virtual reality in medical education. Minimally invasive therapy and allied technologies, Blackwell Science Ltd., Oxford, UK, pp 122–125
3. Basdogan C, Ho CH, Srinivasan MA, et al. (1998) Force interactions in laparoscopic simulations: haptic rendering of soft tissues. *Proceedings of Medicine Meets Virtual Reality*, San Diego, CA, IOS Press, Amsterdam, The Netherlands, pp 385–391
4. Edwards JC, Luecke GR (1996) Physically based models for use in a force feedback virtual environment. *Proceedings of the Japan/USA Symposium on Flexible Automation*, Boston, MA, ASCE, East Washington, DC, USA, pp 221–228
5. Brown J, Montgomery K, Latombe J, et al. (2001) A microsurgery simulation system. *Proceedings of Medical Image Computing and Computer-Assisted Intervention*, Utrecht, The Netherlands, Springer New York, NY, USA, pp 137–144
6. Ikuta K, Iritani K, Fukuyama J (2001) Mobile virtual endoscope system with haptic and visual information for non-invasive inspection training. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2001)*, Seoul, Korea, IEEE, Piscataway, NJ, USA, pp 2037–2044
7. Bianchi G, Solenthaler B, Szekely G, et al. (2004) Simultaneous topology and stiffness identification for mass–spring models based on fem reference deformations. *Proceedings of Medical Image Computing and Computer-Assisted Intervention (MICCAI 2004)*, St-Malo, Springer New York, NY, USA, pp 293–301
8. Weiss DJ, Okamura AM (2004) Haptic rendering of tissue cutting with scissors in a virtual environment. *Proceedings of Medicine Meets Virtual Reality Conference*, Newport Beach, IOS Press, Amsterdam, The Netherlands, pp 157–163
9. Li M, Liu Y-H (2005) Modeling interactions of pulpal tissue with deformable tools in endodontic simulation. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2005)*, Barcelona, Spain, IEEE, Piscataway, NJ, USA, pp 2637–2642
10. Chen P, Steiner KV, Barner KE (2006) Toward realistic soft tissue modeling for virtual surgery simulations. *University of Delaware Computational Science Day*, Newark, 2006, University of Delaware, Newark, DE, USA
11. DiMaio SP, Salcudean SE (2002) Needle insertion modelling and simulation. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2002)*, Washington, DC, IEEE, Piscataway, NJ, USA, pp 2098–2105
12. De S, Srinivasan MA (1999) Thin-walled models for haptic and graphical rendering of soft tissues in surgical simulations. In: Westwood JD, et al. (eds) *Medicine meets virtual reality*. IOS Press, Amsterdam, The Netherlands, pp 94–99
13. Basdogan C, Ho C, Srinivasan MA (2001) Virtual environments for medical training: graphical and haptic simulation of laparoscopic common bile duct exploration. *IEEE/ASME Trans Mechatron* 6:269–285
14. De S, Kim J, Manivannan M, et al. (2002) Multimodal simulation of laparoscopic Heller myotomy using a meshless technique. In: Westwood JD, et al. (eds) *Medicine meets virtual reality*. IOS Press, Amsterdam, The Netherlands, pp 127–132
15. Kim J, De S, Srinivasan MA (2002) Computationally efficient techniques for real-time surgical simulation with force feedback. *Proceedings of the 10th Symposium on Haptic Interfaces for Virtual Environments and Teleoperator Systems*, Orlando, FL, IEEE, Piscataway, NJ, USA, pp 51–57
16. Basdogan C, Srinivasan MA (2002) Haptic rendering in virtual environments. In: Stanney KM (eds) *Virtual environments handbook*. Lawrence Erlbaum, Mahwah, pp 117–134
17. Vlachos K, Papadopoulos E, Mitropoulos D (2003) Design and implementation of a haptic device for urological operations. *IEEE Trans Robotics Autom* 19:801–809
18. Foley JD, Dam A, Feiner SK, et al. (1997) *Computer graphics principles and practice*. Addison-Wesley, New York
19. Farin G (1997) *Curves and surfaces for computer-aided geometric design*. Academic Press, New York
20. Wright RS, Sweet M (2000) *Opengl Superbible*. Waite Group, Indianapolis, Indiana, USA
21. Potmesil M, Chakravarty I (1982) Synthetic image generation with a lens and aperture camera model. *ACM Trans Graphics*, vol 1, Issue 2, pp 85–108
22. Papadopoulos E, Tsamis A, Vlachos K (2004) A real-time graphic environment for a urological operation training simulator. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2004)*, New Orleans, LA, IEEE, Piscataway, NJ, USA, pp 1295–1300