

A Real-Time Graphic Environment for a Urological Operation Training Simulator^{*}

Evangelos Papadopoulos, Alkiviadis Tsamis, and Kostas Vlachos

Department of Mechanical Engineering, National Technical University of Athens, 15780 Athens, Greece

egpapado@central.ntua.gr

Abstract An OpenGL/C++ real-time graphic environment, part of a training simulator for urological operations, is presented. The graphic environment simulates endoscope insertion in a small diameter deformable tube and is used with a low-force 5-dof force-feedback haptic mechanism. Piecewise Bezier interpolations are used for smooth urethra deformations. A novel particle-based model computes the forces and torques fed to the haptics. Realistic textures from medical databases are employed and a 25 fps refresh rate is achieved using the Rendering Thread method. The overall simulator software is made of three processes running on two distinct platforms, communicating via Ethernet and TCP/IP.

Keywords Graphical training simulator, force model, haptics.

I. INTRODUCTION

Minimally invasive surgical operations (MIS) are widely used since they offer significant advantages over established open procedures. However, MIS surgeons face several difficulties not present in open procedures. These include limited depth perception and field of view, inaccurate tool position and orientation, and inadequate force feedback. Therefore, MIS surgeons need many hours of training, mostly *in vivo* with animals. However, such training is limited for ethical and economic reasons.

These problems call for training simulators with which both surgical residents and medical students may be trained. Simulators are less expensive, reduce animal use and result in faster training. Development of a useful and realistic surgical simulator requires the use of a virtual reality graphical environment and of haptic technologies. For example, the VIRGY is an endoscopic surgery simulator, using the PantoScope, [1]. The “Karlsruhe Endoscopic Surgery Trainer” and the “Virtual Endoscopic Surgery Trainer” are laparoscopic simulators that use the software system KISMET, [2].

While the ability to interact with a virtual environment through the feel of forces is important, a realistic visual representation of human anatomy and tissue deformation is also of great importance. Soft tissue modeling methods, employed in graphical environment models of soft tissues, can be divided into *Geometry-Based Methods* and *Physics-Based Methods*.

In the first, to display deformations, a user moves surface vertices or control points that surround a tissue. These methods are fast and result in smooth visuals but exclude the underlying deformation mechanics. To

compute forces, a separate *force-model*, is employed. Geometry-based methods are further subgrouped as *Vertex-Based* and *Spline-Based* methods. In the former, a user directly manipulates surface vertices to display deformations. Polynomial functions that fitted experimental data on finger pad deformation to model other organs were used in [3]. In spline-based methods, a user displaces indirectly surface vertices, manipulating a lattice of control points that surround the object, thus producing smoother deformations, for example see [4].

In physics-based methods, interactions dynamics is taken into account, increasing simulation realism. However, physics-based models are computationally expensive and not always suitable for real-time applications. These are further grouped as *Particle-based*, *Finite Element-based (FEM)* and *Meshless Methods*. The Particle-based consist of a group of point masses connected with each other through springs and dampers. Such methods were initially applied in soft tissue modeling, [5], [6]. Later, virtual suturing of blood vessels was modeled using a mass-spring model, [7]. A mass-spring-damper model was also used to model the large intestine during endoscopic inspection, [8]. Lately, such a method was used to model the gall bladder, [9]. In FEM, a 3D object volume is discretized into finite elements. Due to limitations in computational power, several simplifications have been applied to implement FEM in real-time, [10], [11], [12]. To reduce complexities introduced by mesh generation, the Meshless Finite Spheres Method has been developed and applied in cases with changes in topology. The method was used to simulate cutting, [13], and to model soft biological tissues, [14]. However, hybrid models that combine geometry and physics-based approaches seem to give the best real-time results, [15].

This paper describes the design and implementation of a urological operation, OpenGL/C++ real-time graphic environment working with a haptic device. The environment simulates endoscope insertion in a small diameter deformable tube that undergoes large displacements. Using piecewise Bezier interpolations, smooth urethra deformations are produced. A sinusoidal profile is used to describe the transition of diameter change from the urethra to the endoscope. A novel particle-based model computes forces and torques fed to the haptics. Realistic textures from medical databases are employed. The real-time graphics implementation is

^{*} Support of this work by the PENED and PRAXE programs of the Hellenic General Secretariat for Research and Technology is acknowledged.

based on the Rendering Thread method and is refreshed at 25 fps. The complete simulator software is made of three processes running on two distinct platforms communicating via Ethernet and TCP/IP.

II. WORKSPACE AND MECHANICAL REQUIREMENTS

A urological operation can be divided in two phases, the insertion and the main operation, [16], [17]. During insertion, the surgeon moves the tip of the endoscope along a path from the insertion point A to the final point C at the bladder, via an intermediate point B, see Fig. 1. At point B, the endoscope orientation changes without translation, so as to align the entire urethra and continue the insertion without traumas. The corresponding endoscope poses are labeled by a, b, c, and d, in Fig. 1. During the main operation, the endoscope rotates in all directions but its tip translates minimally. These observations point out that although the endoscope can have any orientation in a cone, its tip translations occur on the plane of symmetry of the patient, z - y , see Fig. 1. Therefore the graphics must represent five degrees-of-freedom (dof), i.e. two translational and three rotational.

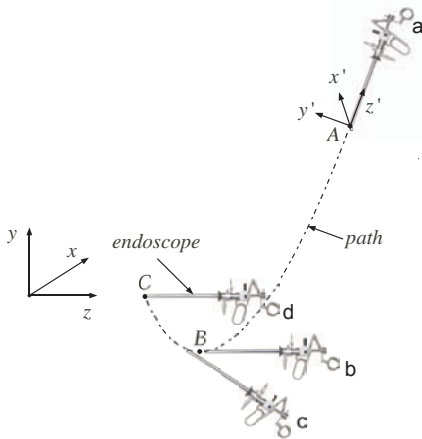


Fig. 1. Endoscope path during an operation.

The haptic interface kinematics are characterized by translations along the y and z fixed axes by 10 cm, rotations around the endoscope z' axis by $\pm 180^\circ$, and rotations about the remaining two axes by $\pm 30^\circ$. The translation along the endoscope path ABC in Fig. 1 is of the order of 20 cm. The measured maximum endoscope forces are 4.5 N along the z and y axes. The maximum measured torques are 150 mNm about the endoscope fixed y' and x' axes and 10 mNm about the z' axis.

III. GEOMETRY OF THE URETHRA

To select the proper methodology for modeling the deformable urethra, the various proposed techniques have been studied. FEM are not recommended for real-time applications due to the large computational load. Also, MFS are advantageous mostly in complex situations, such as those involving changes in topology during cutting procedures. Since this simulator does not require such topology changes, but rather various training scenarios, MFS was also not selected. Using elements from the remaining methods, a hybrid

methodology was employed that combines Geometry and Particle-Based techniques. In this approach, smooth deformations are produced by parametric cubic interpolations, while forces are computed separately.

The study of typical endoscope paths revealed that the path center-line can be approximated by the parabola $y = 0.13z^2$. The urethra's cross-section is assumed to be circular with a constant radius at the undeformed state.

The urethra's surface is constructed using parametric cubic curves and points through which these pass. These points are obtained by discretizing the surface in both the peripheral and axial directions. The points are obtained by using an orthogonal surface with equidistant points in both directions, see Fig. 2. The urethra is produced by folding the orthogonal surface about its z axis. 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 first peripheral section, the next five points define the second peripheral section etc., see Fig. 2. The number of points that define each section is assigned to the parameter value n while the number of sections is assigned to the parameter value w . In Fig. 2, $n = 5$ and $w = 5$.

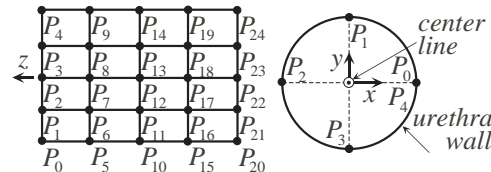


Fig. 2. Simplified representation of the surface discretization (left) and the first peripheral section (right), for $n = 5$ and $w = 5$.

A parametric cubic curve must pass in real-time through the points P_i in both directions. For example in Fig. 2 the first peripheral interpolation begins at P_0 and ends at P_4 , the second begins at P_5 and ends at P_9 , etc. Also the first axial interpolation begins at P_0 and ends at P_{20} , the second begins at P_1 and ends at P_{21} , etc. To produce smooth deformations, we must maintain at least tangent direction continuity at the interpolating points, i.e. continuity G^1 . Out of the parametric cubic curves, only the piecewise Bezier or Hermite interpolations satisfy the above requirements, [18]. The Bezier form is chosen, because the tangent direction and magnitude are better depicted using control points. Each quadrilateral surface in Fig. 2, for example the one defined by points P_6, P_{11}, P_{12} and P_7 , is formed by four Bezier segments, of which the 1st passes through points P_6 and P_7 , the 2nd through P_{11} and P_{12} , the 3rd through P_6 and P_{11} and the 4th through P_7 and P_{12} . The control points CP_i that define the shape of the Bezier curves, see Fig. 3, are calculated by the Catmull-Rom method, [19], which can represent closed curves with an appropriate choice of the first and last inner control points such as the CP_1 and CP_{25} in Fig. 3. For smoothness, the tangent magnitude is constant along the first and last segment of any peripheral or axial interpolation. To this end, the first and last inner control points are placed in the 1/3 of the distance between the respective pair of points [18]. The knot sequence is created using the Chord Length

method, which produces small curvature variations, thus giving smooth deformations, [19].

To obtain the geometry of Fig. 1, the centers of the peripheral sections must lie on the parabolic center-line. The section planes are rotated properly about the x axis, so that they are normal to the tangent vector of the parabola at the corresponding center position. Combining the piecewise Bezier interpolation in both directions, the geometry in Fig. 4 is produced, where the centers of the sections with $j = 0, \dots, (w-1)$ are equidistant along the z axis.

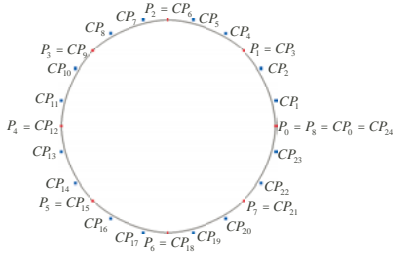


Fig. 3. First peripheral section with points P_i in red and control points CP_i in blue, for $n=9$.

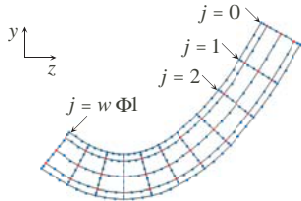


Fig. 4. Geometry of non-deformed urethra for $n=9$, $w=11$ and arbitrary choice of radius.

IV. DEFORMATION OF THE URETHRA

During the endoscope insertion, the urethra is deformed based on three criterions: (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 satisfy these requirements and to be able to implement an accurate force model, the peripheral sections are divided in two sets. The first set, called set of *basic sections*, defines at any time the global urethra geometry. Sections in this set are fixed and do not move along the urethra center-line. The second set, called the *moving deformation*, represents the deformation of the urethra close to the endoscope tip. These are spatially dense, move along the urethra center line and define the transition from the endoscope diameter to the urethra diameter during insertion. A sinusoidal profile was chosen for the moving deformation, since it allows for smooth attachment of the deformation on the endoscope and the urethra. Note that second degree polynomials, such as those used in [3], 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 center-line.

The above concept is implemented and its results are shown in Fig. 5, where the propagation of the deformation of the urethra due to endoscope insertion with respect to the center-line is shown. The endoscope

radius is assumed to be four times greater than the undeformed urethra's. Arrows in Fig. 5 indicate basic sections, while a dashed arrow indicates the endoscope tip section. The moving deformation is described by ten subsections. Two integer parameters are used to define which snapshot is depicted each time. The parameter m , taking values in the range $0, \dots, (w-1)$, defines whether the endoscope tip is between basic sections $j = m$ and $j = m+1$. The parameter l , taking values in the range $1, \dots, 10$, defines the specific position of the tip for a given value of m . For example, if the tip is between the basic sections $j = 0$ and $j = 1$, and is displaced five steps after the basic section $j = 0$, then $m = 0$ and $l = 5$, see upper center snapshot in Fig. 5.

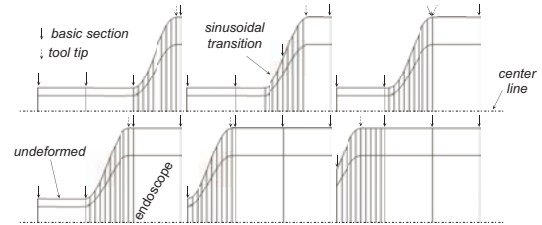


Fig. 5. Basic sections and moving deformation for $n=9$, $w=13$.

Embodying the criterion (b), the urethra's part on the endoscope must acquire the endoscope geometry. At first the centers of the sections that belong to that part of urethra must lie on the endoscope's straight axis. Then the planes of these sections must be normal to the endoscope axis of symmetry. Combining criteria (a) and (b) the result in Fig. 6 is produced for an arbitrary choice of parameter values. The sections are denominated $j = 0, \dots, (w-1)$.

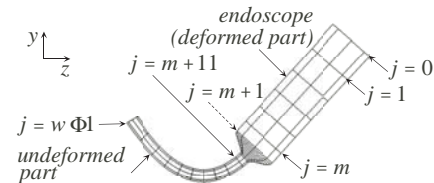


Fig. 6. The endoscope moves on the parabolic axis tangent orientated for $n=9$, $w=26$, $m=5$, $l=7$.

According to criterion (c) the endoscope may not move along the parabolic center-line or be tangent to it at the point where deformation occurs. In this case the sections ahead of the endoscope tip must be translated and rotated about x axis, to follow the user manipulations, in a way that the sections close to tip are affected more than the ones far from it. Section $j = w-1$ is not translated, because it represents the fixed plane for the bladder entrance. Also, sections close together, like these in the moving deformation, should undergo the same amount of translation and rotation, otherwise they may intersect one another. Finally to avoid intersection between the moving deformation and section $j = m+1$, the effect on this section should gradually increase reaching that of the moving deformation, as parameter l takes greater values in the range $1, \dots, 10$.

The deviation of the endoscope position δy and orientation $\delta \theta_x$ with respect to the corresponding

position and orientation on the undeformed center-line are measured. These can be distributed linearly in the sections $j = (m+1), \dots, (w-1)$ according to the above constraints. The result is depicted in Fig. 7 for an arbitrary position and orientation of the endoscope.

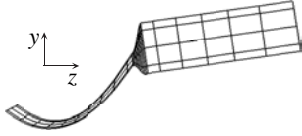


Fig. 7. Arbitrary position and orientation of endoscope for $n=9, w=26, m=5, l=7$.

The endoscope also may be rotated by an angle $\delta\theta_y$ about the y axis, on a circular orbit about its tip. In this case, the centers of sections $j = 0, \dots, (m+1)$ must be translated and rotated by an angle $\delta\theta_y$ to remain on the endoscope, and normal to its axis. The sections $j = (m+2), \dots, (w-1)$ downstream of the endoscope, must be rotated by the angle $\delta\theta_y$, to avoid overlaying each other.

V. TEXTURES

Texture images taken from medical databases, such as the one displayed in Fig. 8, are used for covering up the inner urethra wall. Additional textures can be used according to medical requirements. However, a texture may cover only flat quadrilateral surfaces. To apply this technique to the concave urethra surface, each concave quadrilateral surface in Fig. 4 is discretized into a group of flat quadrilateral surfaces using the Coons method, [19]. The parameter s is used to define the discretization density. For example when $s = 5$, the method produces a 5×5 mesh of flat surfaces, as depicted in Fig. 9.

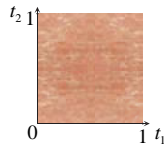


Fig. 8. Texture image with its coordinates t_1 and t_2 .

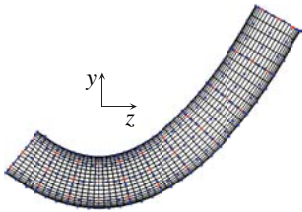


Fig. 9. Coons discretization for $n=9, w=11, s=5$.

To produce smooth shading during lighting, the normal vector of the vertex of each flat surface is used. To calculate this vector, the normal vector of each flat surface is initially calculated as the cross product of two independent surface vectors. Then, the vertex normal vector is calculated by the average of the normal vectors of the surfaces that bound the vertex.

If each flat surface is covered with the entire texture object, then the texture is compressed at the region of the moving deformation, where the sections are near-by, producing an unnatural visual feedback. Someone should better cover the entire surface with the entire

texture object, controlling the values of the coordinates t_1 and t_2 according to the values of the parameters m and l , thus taking into account the distance between the sections. The abscissa t_1 may take values in the axial direction and the ordinate t_2 in the peripheral direction. The coordinate correlation is depicted in Fig. 10.

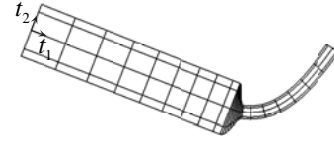


Fig. 10. Correlation of texture coordinates.

Several deformation states, as shown by the virtual endoscope camera with $n = 17, w = 26$ and $s = 5$ are depicted in Fig. 11.

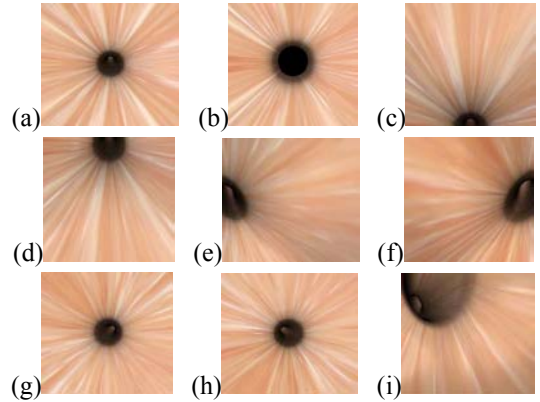


Fig. 11. Deformation states.

In this figure, the endoscope tip point camera is (a) at the urethra entrance, (b) when it reaches the bladder, (c) rotated upwards about x axis, (d) rotated downwards about x axis, (e) rotated right about the y axis, (f) rotated left about the y axis, (g) rotated counterclockwise about the z axis, (h) rotated clockwise about the z axis, (i) at an arbitrary position and orientation.

VI. FORCE MODEL

A force model is designed to calculate forces $F_{re,y}$ and $F_{re,z}$, acting on the human plane of symmetry, y - z , and torques T_x , T_y , and T_z acting about the x , y , and z axes respectively. These are fed back to the haptic device and are produced using linear or nonlinear spring equations. The parameters needed for the computation of the forces and moments are obtained experimentally.

Forces $F_{re,y}$ and $F_{re,z}$ are projections of the total resistance force acting along the endoscope axis, F_{re} , along the y and z axes. This force is given by

$$F_{re} = F_1 + F_2 + F_3 \quad (1)$$

where P_1 is Coulomb friction, P_2 is viscous friction and P_3 is wall resistance force. Force P_1 is calculated by

$$F_1 = \mu N (V_{\parallel} / V) \text{sgn}(V) \quad (2)$$

where V is the relative linear velocity between the endoscope and the tissue, μ is a friction coefficient, N is the force normal to the endoscope axis, 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 same and neighboring section points, such as those proposed in [7], were not used. This is reasonable, since the felt force is mostly due to frictional effects and not to springiness along the urethra. Instead here, $n-1$ springs connecting points P_i on each basic section with the urethra center line are employed. Fig. 12a depicts schematically these springs on the first basic section, 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 progressively its radius. Force N is calculated by summing the normal forces due to the springs on sections $j = 0, \dots, m$ behind the endoscope tip and in some percentage on the basic section $j = m+1$ in the moving deformation. The latter is increased linearly, as l takes values from 1 to 10.

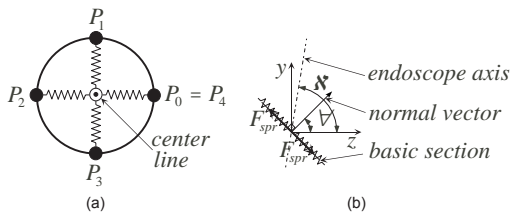


Fig. 12. (a) Springs for friction calculation, (b) contribution of spring forces to resistance.

Viscous friction P_2 is calculated by

$$F_2 = b_{V_{\parallel}} V_{\parallel} \quad (3)$$

where $b_{V_{\parallel}}$ is the viscous friction in the direction of V_{\parallel} . Force P_3 is pure spring-type resistance, produced by the springs on the basic sections $j = (m+1), \dots, (w-10)$ and is calculated by summing the projections of section spring forces, F_{spr} , along the endoscope axis, see Fig. 12b.

A urethra consists of two parts, of which the first is outside the body and free to move (free part) while the second is inside and constrained (constrained part). A torque around the x -axis, T_x , is mainly produced when the endoscope is in the constrained part and whenever the endoscope is off the parabolic center-line. To compute the torques due to urethra deformation during endoscope x rotation and y - z translation, T_x , one could use virtual springs connecting each basic section center to a virtual wall point off the urethra, see [8]. This method works well under the assumption 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 these, rotational springs are placed on basic section centers, see Fig 13a, to produce torques due to y - z translation. Also, a rotational spring is placed at the endoscope tip to produce a torque due to x rotations. The tip rotational spring constant is set to zero, when the endoscope is at the free part, and increases linearly according to endoscope position in the constrained part. The constants of the rotational springs

at the basic section centers are set to zero at the free part and to a positive value at the constrained part. During endoscope insertion, the rotational springs $j = 1, \dots, (w-11)$ are rotated by a relative angle Φ_j between adjacent springs, see Fig. 13b. The last spring $j = w-10$ is rotated by its absolute angle, Φ_{w-10} , about the x axis. The initial angle of the springs is defined at the non-deformed state. Torque T_x is the sum of the rotational spring torques.

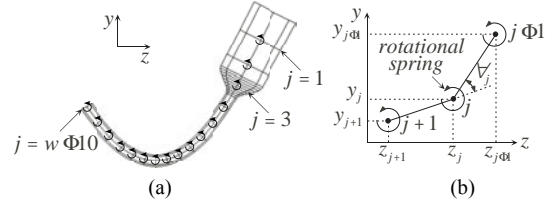


Fig. 13. (a) Rotational springs at the centers of basic sections and (b) relative rotation angles.

To calculate torque T_y , a rotational spring 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 endoscope position in the constrained part.

Torque T_z is produced by Coulomb torque friction, T_1 , and viscous torque friction, T_2 ,

$$T_z = T_1 + T_2 \quad (4)$$

with,

$$T_1 = \mu N V_{\parallel} (r/V) \text{sgn}(V) \quad , \quad V_{\parallel} = |r \dot{\theta}| \quad (5)$$

where r is the endoscope radius, V_{\parallel} is the normal to the endoscope axis component of V and $\dot{\theta}$ is the angular velocity about the endoscope axis.

Finally, the viscous friction torque T_2 is given by

$$T_2 = b \dot{\theta} \quad (6)$$

where b is the viscous friction in the direction of $\dot{\theta}$.

Colored bars have been used to depict the magnitude of the three torques and that of the resulting force. The green, blue, yellow and red bar represent T_x , T_y , T_z and F_{re} respectively. T_x , T_y , T_z and the two F_{re} components, $F_{re,y}$ and $F_{re,z}$, are fed to the haptics.

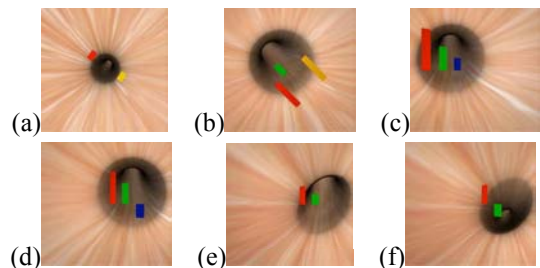


Fig. 14. Colored bars representing endoscope forces and torques.

In Fig. 14 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 are produced, (b) moves backward on the parabola at the constrained urethra part and is rotated clockwise about its axis, so converse F_{re} and T_z are produced and T_x also

appears, (c) moves forward on the parabola at constrained part and rotated right about y axis, so F_{re} , T_x and T_y are produced, (d) as before but rotated left about the y axis, so converse T_y is produced, (e) moves below the parabola at the free part and rotated left about the y axis, so only F_{re} and T_x exist, (f) as before but above the parabola, so converse T_x is produced.

VII. REAL-TIME IMPLEMENTATION

The real-time graphic environment is implemented in OpenGL, a very portable and fast low-level 3D graphics and modeling API. As an API, the OpenGL library follows the C or C++ calling convention. The final code was written using the GLUT library, which includes basic GUI features that facilitate the building of useful applications and easily provide joystick support.

The developed graphics is part of a training simulator that includes a 5-dof force-feedback haptic device. The haptic device has been developed by the same group at NTUA, [16], [17]. The haptic hardware communicates with the graphics via Ethernet cards and TCP/IP. The entire simulator software consists of three processes running on two different platforms.

The first process includes the control loop and the client side communication between the haptic device and the graphics, implemented in a PC/104 cards stack. This consists of a 233 MHz CPU, a 24 MB solid state flash disk, an Ethernet module, two encoder polling input modules, and an analog output module driving motor amplifiers. To ensure a constant force refresh rate of 1 KHz, the QNX[®], a real time operating system from QNX Software Systems Ltd., is employed.

The other two processes are running on a Pentium IV computer equipped with the OpenGL/C++ graphic library. The first includes the visual display of the endoscope view of the tissues and their deformations. The second includes the server side communication between the haptic device and the graphics as well as the force model yielding the forces and torques to be felt by the user. These processes are running in parallel, so the force calculations and communications are not delayed by the time consuming graphics.

The PC/104 stack reads from the encoders the endoscope position and orientation. It then calculates the endoscope velocity, and transmits to the PC data for force computation and displaying the graphics. Force commands are sent to the PC/104, which commands the motor amplifiers. Real-time graphics are updated on an average of 25 fps using the Rendering Thread Method [20], which is rarely delayed by Windows.

VIII. CONCLUSIONS

In this paper, an OpenGL/C++ real-time graphic environment for urological operations, working with a haptic device, was presented. The graphic environment simulates endoscope insertion in a small diameter deformable tube that undergoes large displacements. Piecewise Bezier interpolations produce smooth urethra deformations. A sinusoidal profile describes the urethra-

endoscope diameter change transition. A novel particle-based model was used to compute forces and torques fed to the haptics. Realistic textures were used and a 25 fps real-time graphics environment was achieved. The simulator includes three processes running on two platforms and communicating via Ethernet and TCP/IP.

REFERENCES

- [1] Baur, C., Guzzoni, D., Georg, O., (1998), "Virgy, A Virtual Reality and Force Feedback Based Endoscopy Surgery Simulator", *Proc. Medicine Meets Virtual Reality '98*, (MMVR'98), pp. 110–116.
- [2] Kühnapfel, U., et al., (1997), "The Karlsruhe Endoscopic Surgery Trainer as an example for Virtual Reality in Medical Education", *Minimally Invasive Therapy and Allied Technologies*, (MITAT), Blackwell Science Ltd., pp. 122-125.
- [3] Basdogan C., et al., (1998), "Force Interactions in Laparoscopic Simulations: Haptic Rendering of Soft Tissues", *Proc. of the Medicine Meets Virtual Reality VI Conference*, San Diego, CA, January 19-22, pp. 385-391.
- [4] Edwards J., Luecke G., (1996), "Physically Based Models for Use in a Force Feedback Virtual Environment", *Japan/USA Symposium on Flexible Automation*, ASME, pp. 221-228.
- [5] Cover S.A. et al., (1993), "Interactively Deformable Models for Surgery Simulation", *IEEE Comp. Graph. & Appl.*, pp. 65-78.
- [6] Lee, Y., Terzopoulos, D., Waters, K., (1995), "Realistic Modeling for Facial Animation", *Proc. SIGGRAPH*, pp. 55-62.
- [7] Brown J., et al., (2001), "A Microsurgery Simulation System", *MICCAI2001, LNCS 2208*, W. Niessen and M. Viergever (Eds.), Springer-Verlag, Berlin Heidelberg, pp. 137-144.
- [8] Ikuta K., Iritani K. & Fukuyama J., (2001), "Mobile Virtual Endoscope System With Haptic and Visual Information For Non-Invasive Inspection Training", *Proc. of the 2001 IEEE, International Conference on Robotics & Automation*, Seoul, Korea, May 21-26, pp. 2037-2044.
- [9] Webster R., et al., (2002), "Elastically Deformable 3D Organs for Haptic Surgical Simulation", *Medicine Meets Virtual Reality 2002*, J.D. Westwood et al. (Eds), IOS Press.
- [10] De S. & Srinivasan M. A., (1999), "Thin-Walled Models for Haptic and Graphical Rendering of Soft Tissues in Surgical Simulations", *Medicine Meets Virtual Reality*, Westwood et al. (Eds.), IOS Press, pp. 94-99.
- [11] DiMaio S. P. & Salcudean S. E., (2002), "Needle Insertion Modeling and Simulation", *Proc. 2002 Int. Conference on Robotics & Automation*, Washington, DC, pp. 2098-2105.
- [12] Basdogan C., Ho C. & Srinivasan M. A., (2001), "Virtual Environments for Medical Training: Graphical and Haptic Simulation of Laparoscopic Common Bile Duct Exploration", *IEEE/ASME Trans. on Mechatronics*, Vol. 6, No. 3, pp. 269-285.
- [13] De S., et al., (2002), "Multimodal Simulation of Laparoscopic Heller Myotomy Using a Meshless Technique", *Medicine Meets Virtual Reality*, Westwood et al. (Eds.), IOS Press, pp. 127-132.
- [14] Kim J., De S. and Srinivasan M. A., (2002), "Computationally Efficient Techniques for Real Time Surgical Simulation with Force Feedback", *Proc 10th Symp. on Haptic Interfaces for Virtual Environments & Teleoperator Systems*, (Haptics'02).
- [15] Basdogan C. & Srinivasan M. A., "Haptic Rendering in Virtual Environments", *Virtual Environments Handbook*, pp. 117-134.
- [16] Papadopoulos E., Vlachos K., and Mitropoulos D., (2002), "Design of a 5-dof Haptic Simulator for Urological Operations", *Proc. IEEE Int. Conference on Robotics and Automation, ICRA 2002*, Washington DC, May 11-15, pp. 2074-2079.
- [17] Vlachos K., Papadopoulos, E. and Mitropoulos, D., (2003), "Design and Implementation of a Haptic Device for Urological Operations", *IEEE Tr. Rob. & Aut.*, Vol. 19, No. 5, pp. 801-809.
- [18] Foley, J. D., et al. (1997), *Computer Graphics, Principles and Practice*, 2nd ed., Addison-Wesley Publishing Company.
- [19] Farin, Gerald, (1997), *Curves and Surfaces for CAGD, A Practical Guide*, 4th ed., Academic Press.
- [20] Wright & Sweet, (2000), *OpenGL SuperBible*, 2nd ed. Waite Group Press.