Computer Simulation of Prostate Surgery

M.A. Padilla, F. Altamirano & F. Arambula*

October 15, 2007

Abstract

In this work is presented the current state of a surgery simulation system for training Transurethral Resection of the Prostate (TURP). The interface emulates a real resectoscope and allows to perform the most important movements of the surgical tool during a TURP. The interface is able to work in conjunction with a virtual reality software with a deformable tissue model of the prostate, in order to simulate tissue resection and deformation. The current prototype has five degrees of freedom, which are enough to have a realistic simulation of the surgery movements. The results show that the interface is suitable for a real time surgery simulation training system of the prostate without force feedback.

1 INTRODUCTION

The prostate gland is located next to the bladder in human males, with the urethra running from the bladder neck through the prostate to the penile urethra (Fig. 1). A frequent condition in men above 50 years old is the benign enlargement of the prostate known as Benign Prostatic Hyperplasia (BHP), which in some cases results in significant blockage of the urinary flow. The standard surgical procedure to treat a hypertrophied prostate gland is the Transurethral Resection of the Prostate (TURP). It essentially consists of the removal of the inner lobes of the prostate in order to relieve urinary outflow obstruction. Mastering the TURP technique requires a highly developed hand-eye coordination which enables the surgeon to orientate inside the prostate, using only the monocular view of the lens of the resectoscope. Currently TURP is taught through example from an experienced surgeon. The resident of urology has very restricted opportunity to practice the procedure.

In [1, 3] we reported the development a 3D deformable volumetric model of the prostate for TURP simulation that involves tissue deformation and resection, considering the gland as a viscoelastic solid. In this work we describe the development of the virtual resectoscope interface for our simulator. Section 2 of this paper describes the development of the mechatronic interface that emulates the resectoscope; section 3 briefly describes the interaction scheme between the virtual resectoscope and the tissue model; finally in section 5 we present the conclusions and future perspectives of this work.

*M.A. Padilla & F. Arambula are with Image Analysis and Visualization Lab, CCADET, Universidad Nacional Autonoma de Mexico, P.O.Box 70-186, Mexico, D.F., 04510 {miguel.padilla, fernando.arambula}@ccadet.unam.mx
2 Mechatronic design

In order to obtain a realistic simulation of the most important movements of the surgeon during a TURP, a mechanism was designed based on a disk-ring array (Fig. 2). We decided to reproduce only the five most important degrees, corresponding to movement axes. Three of these axes are rotational and the other two are linear displacements of the resectoscope (Fig. 2.a). We ignore two additional translation degrees of the resectoscope sheath. In opinion of an expert urologist, the five movements chosen are enough for having a realistic reproduction of the real movements during cistoscopies and TURP procedures.

2.1 Signals sensing

Optical encoders are used in order to sense each of the three rotational movements. Each encoder is placed in each rotational axe; with this arrangement we can measure the direction
and the angle rotated by the user on each axis (Fig. 2.b): the output signals are a couple of TTL-level trains of pulses with varying width and phase, according to the direction of rotation and speed of the axis.

The linear displacement of the surgical tool is measured with a linear precision potentiometer. The output voltage of the potentiometer varies according to the position of the resectoscope: from 3.23 to 4.50 V, which corresponds to 23 cm of useful resectoscope displacement. The resecting loop has a linear movement range from 0 to 36 mm; this distance is measured with a two linear Hall effect sensors array and two permanent magnets, as shown in Fig. 2.c.

We used a set of microcontroller for real time sensing and monitoring movements. Since translational sensors produce analog signals, it is easy and fast to use one microcontroller for sensing and processing all translations: we used an LP3500 card [5], which is a low-power, single-board computer with a Rabbit 3000 8 bit microcontroller at 7.4 MHz and programmed in C language with the Dynamic C compiler [2]. Since the rotational sensors produce digital signals, the complexity of processing it is higher, so we used a set of PIC microcontrollers [6] for monitoring in parallel rotational movements.

Each optical sensor gives us three output signals: CH. A, CH. B and CH. I. A program developed on a PIC microcontroller calculates from the two out of phase signals CH. A and CH. B, the angular position. The program is based on storing the previous value given by the sensor to the microcontroller and comparing it with the present value, with the aim to detect any change in the phase. The angular distance is determined by counting the number of pulses of CH. A. CH. I is ignored because it is useful only when the optical sensor reaches a complete turn, which is not possible due to the mechanical design of the device.

The acquisition of the potentiometer output signal is made through one of the analog channels of the LP3500 microcontroller. The input signal is in the range [3.23,4.50]V that corresponds to the movements of the resectoscope from 0 to 23 cm. The mathematical model of this sensor corresponds to a linear device; for this reason, efficient linear interpolation of the linear device model was directly programmed as a routine of the LP3500.

The output signal of the Hall effect sensors array is measured through two analog channels of the LP3500: a program in C is in charge of signal acquisition, addition and final displacement calculation through interpolation of values in a fate-table. The fate-table stores 148 pre-computed values of the nonlinear model of the sensors array at off-line stage. For having more accurate displacements interpolated values of the table that corresponds to a displacement resolution of 0.5 mm are calculated in real-time.

3 Interaction with the virtual model

As we mention earlier, monitoring of the resectoscope movements are performed with an embedded electronic system consisting on five microcontrollers: four PIC microcontrollers for digital signal monitoring of the optical sensors, for rotations; the LP3500 card for the analog signals of the linear potentiometer and hall-effect sensors, for translations.

The five microcontrollers in the embedded system run together in parallel in order to monitoring the movements in real time; the system sends the movements information in the form of moving commands to the virtual model. The real-time resectoscope movements
consequently reflect the interaction between the surgeon and the tissue model. The interactions (tissue deformation and resection) between the virtual tool and the prostate model are consequence of the collision between them.

3.1 Resectoscope movements monitoring

The information flow in the electronic system is as follows (Fig. 3): The card LP3500 makes a request of digital data to the master PIC, in parallel while the data of the digital sensors are received in the reading bus of a serial port C, the signals of the analogical sensors are acquired with the DAC of the LP3500 card and stored in an shipment data array. Data request done by the LP3500 is received by the master PIC, which asks for the information of each of the digital sensors to the corresponding slave PIC, while the slave PIC continuously calculate in parallel the rotational axes position. Therefore when they received the data request the shipment is made immediately with the last sensed data; requests and data reception are made by the PIC serial port. After the master PIC receives the complete information of the digital sensors, sends it to the LP3500 by its serial port C, which stores the arriving information in the shipment data array. When the incoming array is full the LP3500 card sends byte by byte the information of the resectoscope position to the 3D model through its B serial port.

![Microcontrollers embedded system](image)

Figure 3: Microcontrollers embedded system.

3.2 Collision detection

Our approach uses a sphere-tree structure for objects representation. The sphere-three building starts by covering all triangles of the mesh $S$ with a set $L$ of small leaf spheres.
Next, the tree is constructed by recursively subdividing $L$ in two subsets of spheres, through the orthogonal plane placed at the middle of the principal component of the bounding-box containing the spheres. At each recursion the radius and center of the sphere needed to completely cover all its descendants leaf spheres is calculated; this criteria guarantees by induction that every node covers all the leaf spheres of its descendant nodes and allows efficiently updating the tree after mesh deformations or cuttings. The collision detection between two objects represented by the sphere-trees $ST_1$ and $ST_2$ are performed in two phases: a broad phase and a narrow phase. The broad phase consists on recursively traversing both trees in depth-first manner. At each recursion the algorithm evaluates for intersections between the corresponding spheres of $ST_1$ and $ST_2$. At any step, if intermediate spheres does not intersects, all its descendant leafs can not intersects either. On the other hand, if the algorithm detects an spheres intersection the algorithm must make finer evaluations of possible intersection between the children until two leafs intersects. In the narrow phase, if two leafs intersects, the algorithm explicitly checks if the two associated triangles collide and inserts collision if any, in a list $C$. In practice, we can avoid the narrow phase by setting a critical time such that the algorithm stops finding more possible collisions at the broad phase; as a consequence we defined a minimal distance between spheres as a collision criterion, avoiding explicit triangle to triangle collision testing. In fact, this approach works well for our purposes because the algorithm detects most collisions occurring near the resecting loop and ignoring the ones occurring near zones of the resectoscope with less probability of interaction. At the end, if the list $C$ of collisions is empty, there are no collisions detected; on the other hand, $C$ will provide at the post-collision stage, historic information for penetration field and collision response calculation with respect of the penetrating volume of the tool into the soft tissue body.

3.3 Tissue Interaction

The prostate model were implemented as a deformable mass-spring system. After a collision is detected the soft tissue must slightly deform before the tissue resection occurs. Tissue deformations result from the reacting forces due to collisions, where reacting forces are the forces needed to separate the penetrating objects and depends on the penetration depth and the mechanical properties of the soft model. The penetration depth map is in fact the signed distance field of the resectoscope submeshes penetrating the prostate submesh, which is calculated from the collision list $C$. For the moment, tissue resection is done without local mesh refinement, but we are currently working on locally subdividing in real-time the mesh at the contact zone by a recursive regular tetrahedra subdivision scheme in order to approximate as possible the rounded cuts produced by a real resectoscope. As a consequence tissue resections will performed as two operations: removing triangles from surface mesh at low resolution and adding the inner triangles exposed by the cut at higher resolution. More details of the prostate model and the collision detection mechanism are available in [3] and [4] respectively.
4 Conclusion and Future Work

We have presented the development of a virtual resectoscope interface for a surgery simulation system of the prostate, without force feedback. In Fig. 4 a view of the mechatronic device and in Fig. 5 a tissue deformation sequence of the prostate due to the interaction with the virtual resectoscope could be seen. We performed metrological tests by using a ZEISS MC850 Coordinates Measuring Machine (CMM) with an accuracy of $\pm 3 \mu m$, with 96% of confidence [7]: by comparing the data measured by our interface and the ones measured by the CMM for the same positions of the resectoscope head. The tests made reveals that as the resectoscope is moving inside the virtual urethra starting from the mechanical reference point, the displacement error increase, starting from 0.089 mm to an acceptable error of 2.14 mm when the resecting head is located 7cm from the reference point. This is an acceptable error rate for simulation, but more effort must be made for establishing the real approximate position of the prostate inside a patient with respect of an anatomic reference, in order to setup the virtual prostate position correctly in the simulator. For greater distances the error increase drastically until reaching 7mm, so mechanical adjustments must be made in order to reduce errors of movements when the resecting head is located farther from origin. The response rate of the movements monitoring is 28 Hz approximated, but the electronic system could be enhanced in order to reach higher rates of [60-80]Hz, because the minimnal response rate of the full system (including collision, deformation, cutting and rendering) must be 20 Hz (28 Hz for simulation and 60Hz for movements monitoring).

We are currently working in modelling tissue resection by removing and adding the triangles, produced by mesh refinement by subdivision, to the mesh. The mechatronic prototype has been evaluated by an urology specialists and in his opinion visual feedback is more important than haptic feedback and does not seem mandatory for TURP simulation, however we are also planning in the near future to include force feedback to the mechatronic device. We are also planning to evaluates the usefulness of the system with a group of residents in urology, as soon as the full virtual simulation system will be finished.
Figure 5: Tissue deformations due to interaction with the resectoscope.
References


