Computer-Aided Design and Applications

http://www.cadanda.com

Virtual Tele-robotic Operation and Applications

Yonghua Chen, Xuejian He and Ruihua Ye

The University of Hong Kong, yhchen@hkucc.hku.hk

ABSTRACT

Tele-operations have been widely used in many applications. However, it is very difficult to maneuver a device in an environment with hidden objects. Virtual tele-operation can not only provide a pre-operation path planning tool, it also provides a low cost method for pre-operation rehearsal and training. In this paper, a haptic device based virtual tele-robotic system for a 5-axis articulated robot is presented. Instead of one master configuration to one slave configuration, the proposed method presents a transformation of one master configuration to many slave configurations based on inverse kinematics. Using the proposed method, two application backgrounds will be illustrated: one is the robotic assembly path planning and the other is the femur bone reduction pre-operative planning. In both of the applications, the operation is made easy because any collision can be felt by the user who manipulates a virtual robot through a virtual environment by a haptic device.

Keywords: haptics, robot path planning, virtual tele-operation.
DOI: 10.3722/cadaps.2009.483-491

1. INTRODUCTION

In many applications, path planning is a key process before actual operations. Typical applications involving path planning can be found in robotics [2], assembly maintainability [3], computer animation [7] and computer numerical control[8].

In actual manufacturing environments, manual or semi-automatic path planning, such as on-line teaching, is widely practiced for robotic path planning. In the teaching process, an operator moves a robot to each desired position and records its motions, which are used to generate robot program that will move its arm through the same motions automatically. Though it is very simple, the most significant disadvantage of on-line teaching is that it occupies valuable production equipment and time.

To overcome these drawbacks, off-line manipulation of a virtual robot through keyboards and mouse is now commonly used in industry. However, using only keyboard and mouse devices, it is difficult to define robot orientations. He/she must input values of orientations manually, and these values are often obscure to an operator. In this paper, the main objective is to develop a convenient tool for intuitively manipulating a virtual robot arm based on a 6-DOF haptic interface Phantom [6]. Since the interaction between robots and obstacles involves object-object collisions, 3-*DOF* haptic rendering is not enough. Although there are some studies on using haptic interface for path planning [1], [5], [7], only very simple examples are given to illustrate their methods. In this paper, a physically modeled 6-*DOF* robot is developed for virtual tele-operations in complex virtual environments as will be shown in the sample studies.

Fig. 1 illustrates the tele-operation of a virtual robot by means of the virtual tele-operation system in comparison with traditional computer interfaces. The user moves the virtual robot arm as if he/she is holding a real robot arm. When collision between the virtual robot and the environment occurs, the user can feel it and the motion path can be changed immediately.

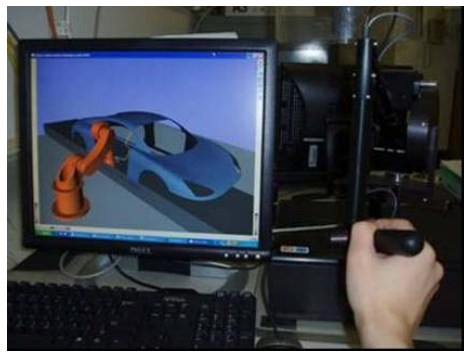


Fig. 1: Haptic manipulation of a virtual robot.

2. ROBOT MODELING AND VIRTUAL TELE-OPERATION

Since 6-*DOF* articulated robots are commonly used in industries, e.g. *ABB* robots (Global Robots Ltd.), a 6-*DOF* articulated robot is modeled in this research.

2.1 Robot Kinematics

First, 3D robot links, joints and end effector are modeled according to the real size of an ABB robot. Then the parts are assembled into a virtual robot as shown in Fig. 2. with all joint constraints information.

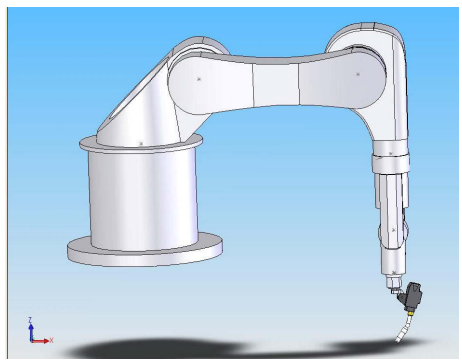


Fig. 2: Geometric modeling of a robot.

A commonly used convention for manipulator frames of reference in robotic applications is the Denavit-Hartenberg (D-H) notation [4]. In this convention, each homogeneous transformation ${}^{i-1}A_i$, relating frame i to frame i-1, is represented as:

$$ {}^{i-1}A_i = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.1} $$

where the four quantities, a_i, α_i, d_i, θ_i are link length, link twist, link offset, and joint angle respectively, associated with link i and joint, and c and s are abbreviations of trigonometric functions cosine and sine. If the robot has n links, then the homogeneous transformation relating the tool frame n to the base frame 0 is given by:

$$\begin{aligned} {}^0T_n &= {}^0A_1 \cdot {}^1A_2 \cdot \cdots \cdot {}^{n-1}A_n \\ &= G(q) \end{aligned} \tag{2.2}$$

Where $G(q)$ is the product of the coordinate frame transform matrices of all links. q is a vector of robot joint parameters.

2.2 Inverse Kinematics (IK)

The forward kinematics model defines the relation:

$$ {}^0T_n = G(q) \tag{2.3}$$

where ${}^0T_n$ is the homogeneous transform representing the positions and orientations of the manipulator tool (frame n) in the base frame 0. The IK model is defined by:

$$ q = G^{-1}({}^0T_n) \tag{2.4}$$

In general, this equation allows multiple solutions. The manipulator Jacobian defines the relation between the velocities in joint space $\dot{q}$ and in the Cartesian space $\dot{X}$:

$$ \dot{X} = J(q)\dot{q} \tag{2.5}$$

or the relation between small variations in joint space δq and small displacements in the Cartesian space δX

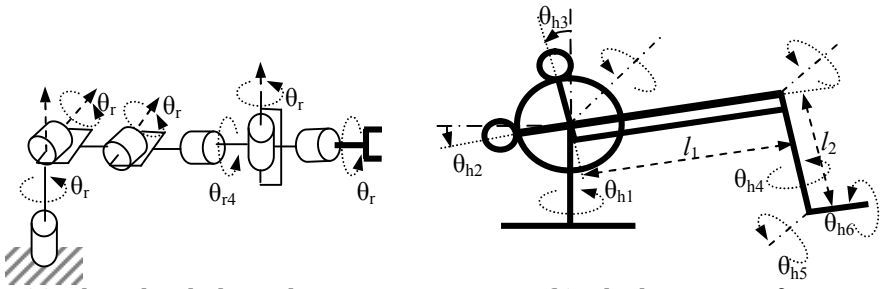
$$ \delta X \approx J(q)\delta q \tag{2.6}$$

Two IK models are compared. The first one (IK1) is based on Jacobian pseudo-inverse method. The second one (IK2) is based on the Taylor expansion. In order to evaluate the performances of these two methods, the IK problem of the 6-DOF robot is solved by IK1 and IK2. The accuracies of IK1 and IK2 are at the same order of 0.001 rad. However, IK1 runs much faster than IK2. On average, it costs IK2 1.54 milliseconds to solve the problem, while IK1 only needs 0.58 milliseconds. Hence, IK1 is adopted in our system since time is critical in real time virtual robot manipulation.

2.3 Virtual Tele-operation

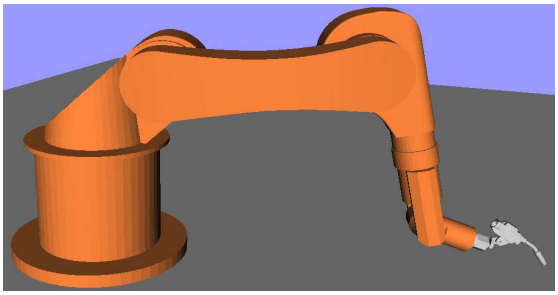
When a virtual robot is manipulated by a user using a haptic device, the tool frame at the robot end-effector must be kept consistent with that of the haptic device. When the virtual robot has 6 DOFs (Fig. 3(a).), there are two means to control the virtual robot through the 6-DOF haptic device (Fig. 3(b).). The first one is that the robot joints and the joints of the haptic interface are directly linked as shown in Fig. 4. (only three joints are illustrated). Because the ranges of their joints are different, a mapping method from the haptic configuration to the robotic configuration must be applied by the following equation.

$$ \theta_{ri} = (\theta_{ri-\max} - \theta_{ri-\min})(\theta_{hi} - \theta_{hi-\min}) / (\theta_{hi-\max} - \theta_{hi-\min}) + \theta_{ri-\min}, \quad (i = 1, \cdots 5) \tag{2.7}$$

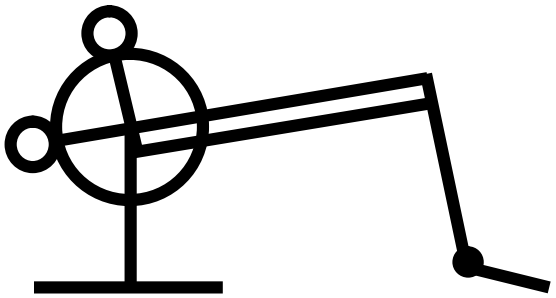


(a): The robot links and joints (b): The haptic interface

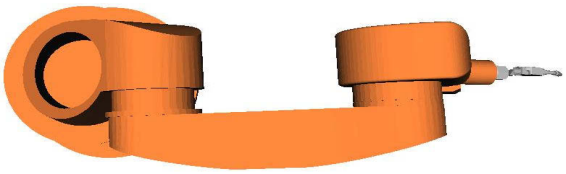
Fig. 3: Geometric structure of the robot and the haptic interface.



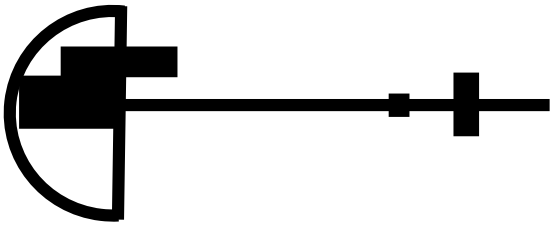
(a) Side view of the virtual robot.



(b) Side view of the haptic interface



(c) Top view of the virtual robot.



(d) Top view of the haptic interface

Fig. 4: Tele-operation method I: direct linking of joints.

whereθ_{ri} is the current *i*th joint value of the virtual robot, θ_{hi} is the current *i*th joint value of the haptic interface, $\theta_{hi\text{-}min}$ is its minimum joint value, $\theta_{hi\text{-}max}$ is its maximum joint value, $\theta_{ri\text{-}max}$ is the maximum *i*th joint value of the virtual robot, and $\theta_{ri\text{-}min}$ is its minimum joint value.

However, this method cannot be used to robots with different configurations or different DOFs. Therefore, a more general manipulation method based on inverse kinematics is proposed. In this mode, the tool center point (*TCP*) of a robot is manipulated by the haptic interface point (*HIP*) as shown in Fig. 5. In the simulation loop, firstly, the system traces the transformation of the haptic interface ($\boldsymbol{X}_{\mathrm{h}}$) in haptic workspace that is the physical space reachable by the haptic device. Then, it converts the transformation to graphics scene ($\boldsymbol{X}_{\mathrm{m}}$). Defining a mapping between the haptic workspace and the graphic scene will describe how movement of the haptic device translates to movement in the graphic scene. The transformation ${}^{m}T_{h}$ that transforms the haptic pose (position and orientation) to pose of graphic model in world coordinates is computed by:

$$ {}^{m}T_{h} = {}^{m}T_{c} \cdot {}^{c}T_{h} \tag{2.8} $$

where ${}^{c}T_{h}$ is the transformation that transforms haptic coordinates to view coordinates that are the local coordinates of the camera (eye coordinates), and ${}^{m}T_{c}$ is the transformation that transforms view coordinates to world coordinates.

Next, using inverse kinematics (IK), as explained in previous section, to calculate the robot configuration *q*.

$$ q = IK(X_{m}) \tag{2.9} $$

Then the graphic model and collision model of the virtual robot are updated according to the calculated robot configuration.

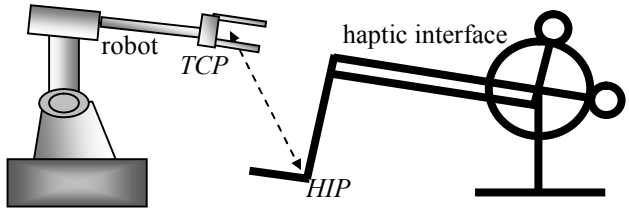
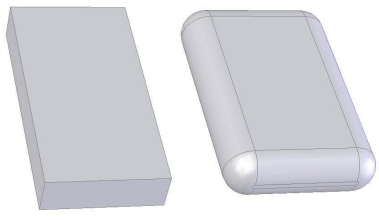


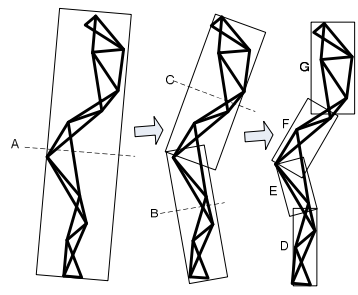
Fig. 5: Tele-operation method II: link the TCP to HIP.

3. COLLISION DETECTION

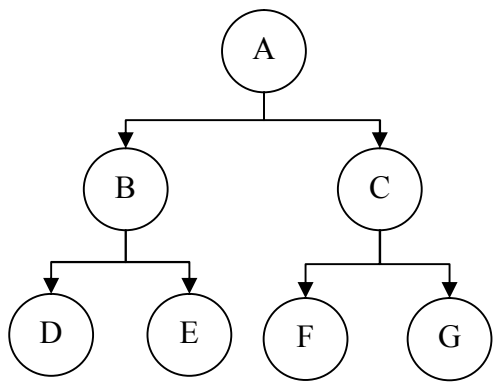
A proximity query package (*PQP*) based on *bounding volume hierarchies* (*BVH*s) is used for haptic rendering in this paper. It utilizes different swept spheres as *bounding volumes* (*BV*s) to construct a hybrid hierarchal data structure. *BV*s are used to enclose sets of primitives and *BVH* is a tree structure used to store *BV*s in nodes. The root of *BVH* contains all primitives of a model. Its children contain portions of the model and leaves contain a single primitive. Swept sphere volumes are generated by taking the Minkowski sum or convolution of a core primitive and a sphere. There are two types of *BV*s used in *PQP*: oriented bounding box (*OBB*) and rectangle swept sphere (*RSS*) as depicted in Fig. 6(a). Given a set of triangles, statistical techniques are used to calculate *BV*s, which encloses the underlying primitives. A top-down strategy is utilized to construct the *BVH* (Fig. 6(b). and Fig. 6(c).). The triangles in each node of the tree are recursively split into two subsets, until they contain only a single triangle. *PQP* uses bounding volume test tree (*BVTT*) to perform collision or distance checking. Each node in the *BVTT* represents a single collision or distance test between a pair of *BV*s as depicted in Fig. 6(d).



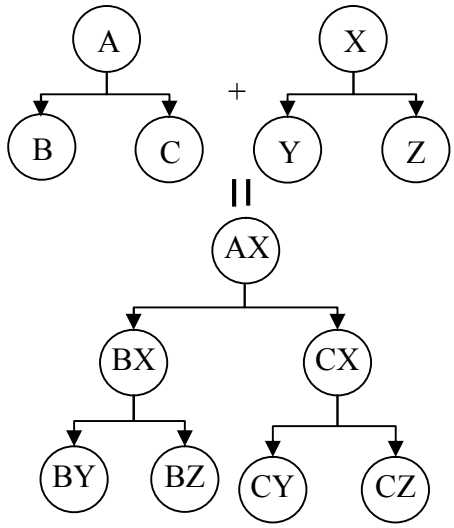
(a) OBB and RSS.



(b) Construction of a BVH.



(c) Tree of BVH.



(d) BVTT.

Fig. 6: Bounding volume test tree.

Two techniques, priority directed search or coherence are applied in *PQP* in order to speedup query. In priority directed search, a priority queue is used to schedule query order of *BVH* according to their distances between the queried pairs of *BV*s. The closest pair is given with the highest priority. In many applications, e.g. real-time simulation, the distances among objects to be queried only change slightly between successive frames. Therefore, it is useful to record the closest triangle pair from previous query for initialization of the minimum distance, ε.

3.1 Compute the Distance of *OBB*

Given two *OBB*s, *A* and *B* (Fig. 7(a).), and the half dimensions of A and B, a_i and b_i, where $i = 1, 2, 3$. The unit vectors, $\boldsymbol{A}_i$ and $\boldsymbol{B}_i$, denote the axes of A and B. $\boldsymbol{T}$ is a vector from the center of A to the center of B. If *A* and *B* are disjoint, a separating axis, $\boldsymbol{L}$, exists. Project the centers of *A* and *B* onto the axis, and compute the radii of the intervals, L_a and L_b, using the equation below.

$$L_a = \sum_{i=1}^{3} | a_i A_i \cdot L | \tag{3.1}$$

Similarly, L_b can be calculated. A and B is disjoint if and only if:

$$| T \cdot L | > L_a + L_b \tag{3.2}$$

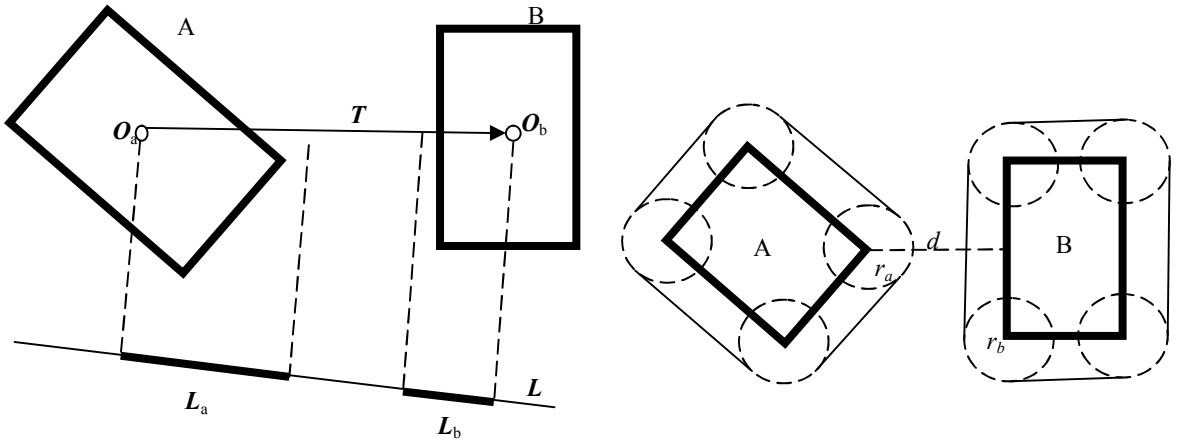
3.2 Compute the Distance of *RSS*

The distance between *RSS*, d_{rss}, can be obtained by computing the distance, *d*, between the core rectangles of *RSS*, and subtracting the sum of their radii, r_a and r_b, as explained in Fig. 7(b).

$$d_{rss} = d - r_a - r_b, \quad (d \geq r_a + r_b) \tag{3.3}$$

PQP can be used to perform four types of proximity queries on a pair of geometric models composed of triangle facets:

- *Distance computation (C1).*Compute the minimum distance between a pair of models, e.g., the distance between the closest pair of points.
- *Tolerance verification (C2).* Determine whether two models are closer or farther than a tolerance distance or not.
- *Collision verification (C3).* Detect whether the two models overlap or not.
- *Collision detection (C4).* Detect all of the overlapping triangles of the two models.



(a) Distance computation for *OBBs.* (b) Distance computation for *RSSs.*

Fig. 7: Distance computation.

In collision detection, the algorithm recursively traverses the *BVH*s of the geometric models and checks whether two *BV*s, e.g. A and B are overlapped for each recursive step. If *A* and *B* do not overlap, the recursive branch is terminated. Otherwise, the algorithm will be applied recursively to their children. In distance computation, the process of query is similar to that of collision detection.

4. HAPTIC RENDERING

When the virtual robot is manipulated by a user using the Phantom haptic device, the virtual robot follows the haptic device as described in previous section. When a user manipulates the robot to obstacles within a small distance, σ , forces and torques are computed and sent to the haptic controller. Therefore, forces and torques can be felt by the operator when the robot is close enough to an obstacle, reminding the user to move the robot more carefully. The haptic rendering method is based on the distance computation between a robot and obstacles (Fig. 8.).

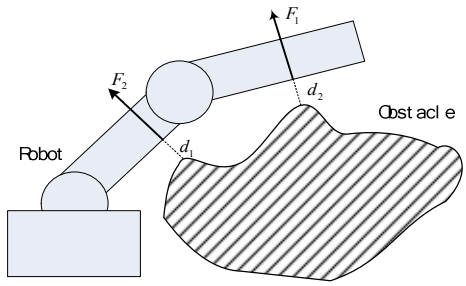


Fig. 8: Schematic of haptic rendering.

The force *F* and torque *M* are computed by:

$$F = \sum_i F_i = \sum_i k_f(|\, d_i \,| - \sigma) d_i \tag{3.4}$$

where k_f is the force stiffness constant, $|\boldsymbol{d}_i|$ is the minimum distance between robot and obstacle, and σ is the threshold.

$$M = \sum_i M_i = \sum_i k_t(F_i \times \overline{OP_i}) \tag{3.5}$$

where k_t is the torque stiffness constant, $\boldsymbol{O}$ is the center of obstacle, and $\boldsymbol{P}_i$ is the closest point to obstacle. Forces and torques can be recorded when the user manipulates the robot arm through the haptic interface.

5. SYSTEM IMPLEMENTATION AND SAMPLE APPLICATIONS

Our current experimental system is configured as follows: Dell® precision PWS670; a dual-processor Xeon (TM) 2.8 GHz with 1GB RAM; NVIDIA Quadro graphic card; Microsoft Windows XP operation system; a Phantom® Premium 1.5/6-DOF device and OpenHaptics (SensAble®) API for haptic rendering; Open Invent™ based on OpenGL® for graphic rendering.

An articulated robot is modeled based on the method mentioned above. A car frame as shown in Fig. 9. needs to be welded by the robot at some joints. An operator can manipulate the 6-*DOF* haptic interface to move the virtual robot to each desired configuration. At every time step, the system will automatically record the robot's configurations in the course of manipulation. Since the time step is set to be small (e.g. 1s), no interpolation method is needed. This is because many robot controllers adopt different interpolation methods, which will lead to different collision results in automatic movement. If the robot collides with the part during manipulation, the system will mark out visually which link of the robot arm is in collision. When collision occurs, forces and torques are sent to the operator through the stylus of the haptic device to remind him/her that the current configuration should not be selected as part of the path. Beside the visual and haptic hints, an alert sound is also used to remind the operator of the collision.

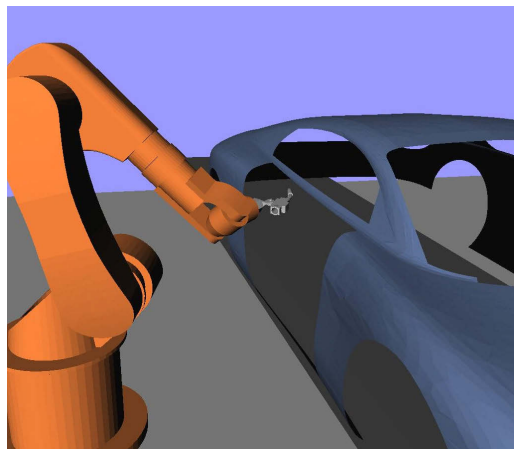


Fig. 9: Tele-operated virtual robot for assembly path planning.

Fig. 10 shows the application of the proposed system that is used for virtual tele-operation on robot-assisted femur bone reduction. Based on CT scans, a 3D model of the bone fracture section is constructed. The virtual tele-operation system allows a surgeon to practice the surgical operation and plan an optimal path in femur bone reduction.

6. CONCLUSIONS

A virtual tele-operation system for virtual robot manipulation has been presented in this paper. The system provides a convenient tool for intuitively manipulating a virtual robot arm in a complex virtual environment for various applications. Using the haptic interface, the selection and definition of critical robot configurations is intuitive, easy and fast for robot path planning. For femur bone reduction, the system can also provide some valuable practicing chances before the actual surgery takes place.
The robot as shown in the experimental study is a 6-*DOF* articulated configuration. In fact, robots with more general structures can be designed and modeled according to the methods described in this paper.

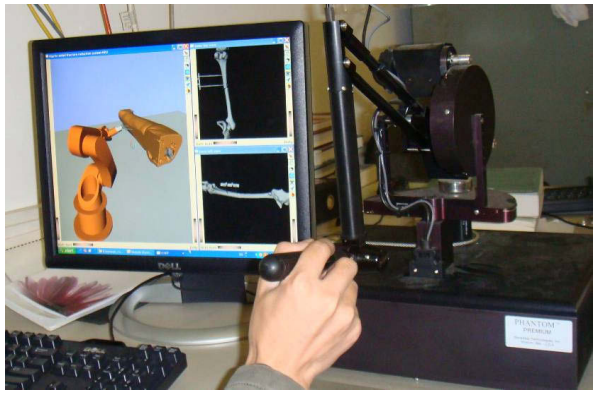


Fig. 10: Tele-operated virtual robot for femur bone reduction.

7. ACKNOWLEDGEMENT
This research is supported by a grant from Hong Kong Research Grants Council under the code HKU 7116/05E.

8. REFERENCES

[1] Amato, N.-M; Bayazit, O.-B; Song, G.: Providing haptic 'hints' to automatic motion planners, Proc. of the 4th PHANTOM User's Group Workshop (PUG), October 1999.
[2] Canny, J.: The Complexity of Robot Motion Planning, MIT Press, Cambridge, MA, 1988.
[3] Chang, H; Li, T.-Y.: Assembly Maintainability Study with Motion Planning, Proc. of the 1995 IEEE International Conference on Robotics and Automation, 1, 1995, 1012-1019.
[4] Denavit, J.; Hartenberg, R.-S.: A kinematic notation for lower pair mechanisms based on matrices, ASME J Appl Mech, 22, 1995, 215–221.
[5] Galeano, D.; Payandeh, S.: Artificial and natural force constraints in haptic-aided path planning, IEEE Int. Workshop on Haptic Audio Visual Environments and their Applications, Ottawa, Ontario, Canada, 1-2 October, 2005
[6] http://www.sensable.com/haptic-phantom-premium.htm
[7] Koga, Y.; Kondo, K.; Kuffner, J.; Latombe, J.-L.: Planning Motions with Intentions, Proc. of SIGGRAPH 94, ACM SIGGRAPH, 1994, 395-408.
[8] Zhu, W.-H.; Lee, Y.-S.: Five-axis pencil-cut planning and virtual prototyping with 5-DOF haptic interface, Computer-Aided Design, 36, 2004, 1295–1307.