The International Journal of **Medical Robotics** and **Computer Assisted Surgery**

THE INTERNATIONAL JOURNAL OF MEDICAL ROBOTICS AND COMPUTER ASSISTED SURGERY
Int J Med Robotics Comput Assist Surg (2013)
Published online in Wiley Online Library (wileyonlinelibrary.com) **DOI:** 10.1002/rcs.1551

ORIGINAL ARTICLE

Spatial motion constraints for a minimally invasive surgical robot using customizable virtual fixtures

Aolin Tang[1]*
Qixin Cao[1]
Tiewen Pan[2]

[1]*State Key Lab of Mechanical Systems and Vibration, School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai, China*

[2]*Division of Thoracic Surgery, Shanghai Changzhen Hospital, Shanghai, China*

*Correspondence to:
A. Tang, State Key Lab of Mechanical Systems and Vibration, School of Mechanical Engineering, Shanghai Jiao Tong University, No.800 Dongchuan Road, Minhang District, Shanghai, 200240, People's Republic of China.
E-mail: altang1988@gmail.com

Accepted: 8 October 2013

Abstract

Background A surgeon performs various surgical actions with the help of teleoperation technology during robot assisted minimally invasive surgery. The slave manipulators reproduce every movement of the surgeon's hands, completely and faithfully. However, in some cases, unexpected collisions between surgical instruments and fragile organs might occur around the lesion, causing further injury.

Method Marking critical areas near the surface of each vulnerable organ and generating virtual fixtures, based on which to constrain the slave manipulator motion to prevent any unexpected collision.

Result Virtual fixtures can be easily defined using the proposed method before or even during the surgery as the surgeon wishes. The experimental results show that the virtual fixtures can protect the organs very well during surgical manipulation.

Conclusion The proposed method can easily be applied to any master–slave minimally invasive surgical (MIS) robotic system to increase the safety of robot-assisted surgery. Copyright © 2013 John Wiley & Sons, Ltd.

Keywords minimally invasive surgical robotic system; customizable virtual fixtures; the minimum distance; spatial motion constraints

Introduction

Minimally invasive surgery is a procedure typically involving the use of arthroscopic or laparoscopic devices and the manipulation of instruments using indirect surgical field observation through an endoscope. The surgical procedure is carried out through a body cavity or an anatomical opening (1). This approach is less invasive than the traditional open surgery used for the same purpose, hence usually resulting in less post-operative pain, less scarring, less chance for infection and a quicker recovery. This approach is a significant breakthrough in surgical technology. However, there are still certain issues that remain to be addressed. First, a patient-side assistant is needed to operate the endoscope for the surgeon during the surgery. The position accuracy and stability of the hand-operated endoscope cannot be guaranteed. Hand-operated MIS instruments are also difficult to operate because the movement vectors are counter-intuitive. A surgeon requires extensive, long-term training to be skilled. Advanced robotic technology has been introduced to overcome these deficiencies. Various kinds of MIS robotic systems have been designed to help the surgeon with endoscope and surgical instrument manipulation (Figure 1). One of the most well known MIS robotic systems is the da Vinci surgical system (2) from

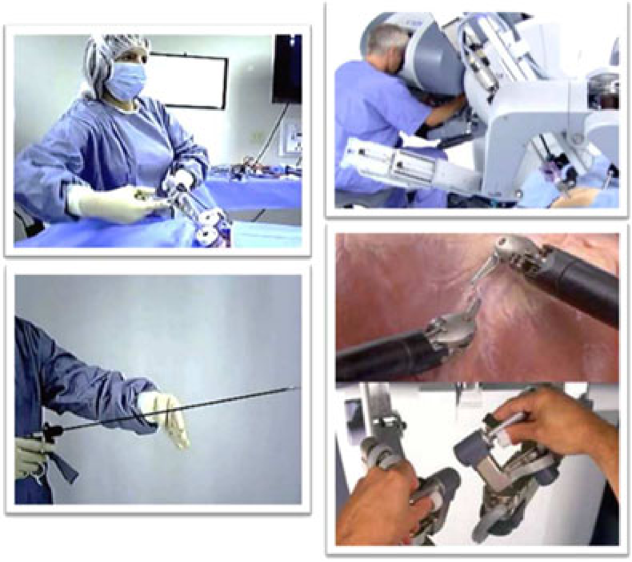

Figure 1. A comparison between the traditional minimally invasive surgery (left) and the robot-assisted minimally invasive surgery (right, using da Vinci surgical system (2))

Intuitive Surgical Inc. It consists of a master control console and four slave robot manipulators. Various surgical instruments can be mounted at the end of each manipulator. With the ability to move quickly, accurately and stably, the MIS robotic system significantly enhances the surgeon's ability, reduces their cognitive load and pressure, improves their performance and helps them carry out the surgery with higher quality and safety.

During robot-assisted minimally invasive surgery the surgeon manipulates the slave surgical manipulator inside the patient using teleoperation, which means the surgeon only needs to sit by a master console and perform the various operational actions with the input handles. The instruments on the slave manipulator will move in the same direction as the surgeon's hands. Every movement of the surgeon's hands will be completely and faithfully reproduced by the slave manipulator. This motion reproduction control strategy is simple, effective and identical to the user's intuition, and therefore has been adopted by many MIS robotic systems throughout the world. However, due to the lack of tactile feedback the surgeon might unconsciously move the surgical instrument excessively. Currently there is still no mature solution for sensing the force acting at the end of the surgical instrument. The severe requirements of minimally invasive surgery allow the surgeon only limited visual information of the pathological area. Especially when the operating volume area is very narrow, unexpected collisions might occur between the slave manipulator and vulnerable organs (such as nerves and arteries) around the lesion, causing inadvertent injury to them (even threats to the patient's life). Some MIS robotic systems utilize a surgical navigation system to handle this problem. A medical imaging system produces a 3D geometric model of the pathological area. The surgical navigation system then tracks the instrument relative to the accurate 3D model. Different algorithms can be used to constrain the slave manipulator motion to avoid inadvertent injury. Virtual fixtures technology (3) is currently one of the most commonly employed motion constraining methods. A virtual fixture is a computer-generated region in which the surgical instrument's degrees of freedom are reduced so that the instrument's movement can be influenced along a desired path (this is called guiding virtual fixture) or limited outside a restricted region (this is called forbidden regions virtual fixture or virtual wall). Visually a virtual fixture is to the surgical instrument like a ruler is to a pen. However, in prior research, nearly all virtual fixtures of interest had to be defined based on the 3D target pathological area model, which makes the preoperative phase very complex and time-consuming. The existing 3D imaging devices and surgical navigation systems are also very expensive. This paper presents a new virtual fixture generation method. The proposed method is low-cost and easy-to-use compared with traditional methods. Preoperative 3D modeling of the pathological area is no longer necessary. The surgeon only needs to manipulate the slave manipulator to mark critical points near the surface of each vulnerable organ in need of protection. Virtual fixtures are then automatically generated based on the marked points that completely cover the organ of interest. During the surgery the control system calculates and monitors the minimum distance between the slave manipulator and all generated virtual fixtures in real time. When the minimum distance is smaller than a threshold value, two types of virtual assistive forces will be generated to constrain the slave manipulator movement to prevent any possible collision.

This paper is organized as follows. After the introduction, additional details about the proposed virtual fixture generation method are introduced. A fast algorithm is then designed to calculate the minimum distance between the surgical instrument and all generated virtual fixtures in real time. Based on the minimum distance obtained the control system can accordingly constrain the spatial motion of the surgical instrument. A new motion

DOI: 10.1002/rcs

constraint method is then introduced that can prevent the surgical instrument from colliding with the virtual fixtures and also guide the surgical instrument to bypass the defined virtual fixtures and move towards its original goal. Experiments are then conducted to examine the validity of the proposed algorithm. The results and related discussions are given. Conclusions are drawn and future work is briefly described in the last section of this paper.

Materials and methods

A new virtual fixture generation method

The virtual fixture, as described above, is a technology that assists the human operator in safer, faster, and more accurate task completion. It can be a guiding virtual fixture or a forbidden regions virtual fixture, acting as a macro that assists a surgeon in carrying out a structured surgical action, or a safety constraint that prevents the surgical instrument from entering a potentially dangerous region of the workspace. The concept of virtual fixtures was first introduced by Rosenberg to improve telepresence in a telemanipulation task (4). Virtual fixtures have been widely used in robot-assisted surgery. Davies *et al*. (5) used it to constrain a robot in cutting the femur and tibia within a preprogrammed permitted region during prosthetic knee surgery. Park *et al*. (6) developed some virtual fixtures from preoperative CT scan data, to constrain the instrument's motions along appropriate paths for internal mammary artery (IMA) in harvesting a portion of the coronary artery in robot assisted bypass graft procedures. In Li *et al*.'s research (7,8) spatial motion constraints were designed to assist the surgeon in high-skilled manipulation tasks in ear, nose and throat (ENT) surgery. The virtual fixtures involved were automatically generated based on a 3D geometric model created from preoperative medical images. The research by Marayong *et al*. (9) provided a demonstration of control algorithms for general motion constraints with varying compliance. The virtual fixtures used were also created from sensor data. Abbott *et al*. (10,11) presented the design, analysis and implementation of two categories of virtual fixtures in their work. Kapoor *et al*. (12) introduced an algorithm for a stitching task in endoscopic surgery under robotic assistance. In Bettini *et al*.'s work (13), guiding virtual fixtures were implemented in a vision-based cooperative manipulation system to limit the motion of the surgical tool. Becker *et al*. (14) derived a virtual fixture framework for active handheld micromanipulators. The guiding virtual fixtures applied were generated in real time from microscope stereo video during the surgical procedure. Ryde *et al*. (15) utilized some forbidden-region virtual fixtures generated from streaming point clouds captured using a Kinect camera (Microsoft Corp.) to protect the beating heart. During the surgery the virtual fixtures were also updated in real time. Unfortunately, this method could not be applied to a minimally invasive surgical operation. In general, the virtual fixtures used to protect organs or tissues are generated from accurate 3D geometric data from the pathological area. Hence, the surgeon needs to scan the patient before the surgery using a medical imaging device such as CT or MRI (usually in another room) to obtain an accurate 3D model of the pathological area. Virtual fixtures are then defined and generated based on the 3D model. This process is time-consuming and wearisome. The required medical imaging devices are also very expensive.

This paper introduces a new virtual fixture generation method that is low-cost and very easy to use. To help us better illustrate this method a simple virtual surgical scene is built in the OpenHRP3 simulation environment, as shown in Figure 2. The virtual surgical scene consists of a virtual MIS robot (it has three slave manipulators, two of them are surgical tools carrying arms, and the third is an endoscope holding arm) and a surgical zone. The surgical zone contains a diseased organ (the red sphere), and some other vulnerable organs around the diseased organ (the green boxes). The objective of this virtual surgery is to remove part of the diseased organ (the red sphere). However, the diseased organ is surrounded by

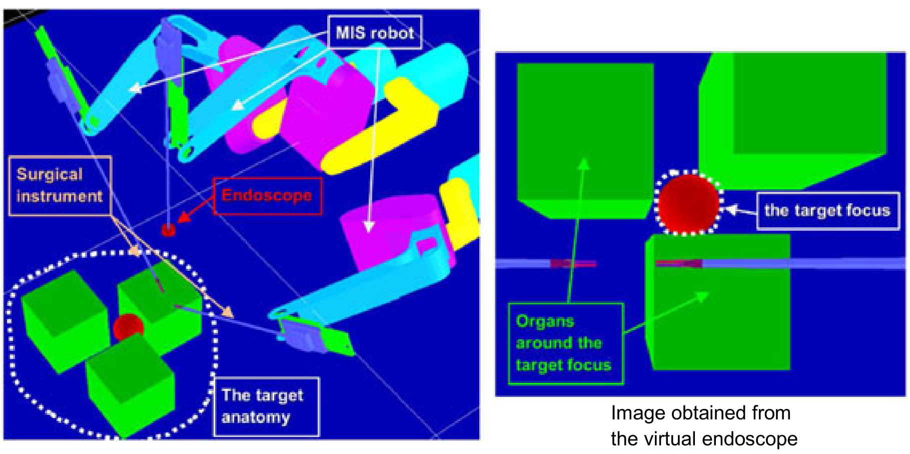

Figure 2. A simple virtual surgical scene in the OpenHRP3 simulation environment

DOI: 10.1002/rcs

other organs (the green boxes), which makes it difficult for the surgeon to finish the task without touching the surrounding vulnerable organs. Hence, some forbidden-region virtual fixtures are needed to assist with this manipulation task.

Unlike traditional virtual fixture generation methods, accurate 3D pathological area modeling is not required. We only need to manipulate the slave manipulator to mark critical points near the surface of each vulnerable organ to generate the virtual fixtures we need. As shown in Figure 3(a), first we command the slave manipulator to move along the organ shape features and mark a series of points at important locations. A blunt-ended instrument is mounted on the slave manipulator during this process. If this blunt instrument comes in contact with the organ surface it will not cause any injury. The spatial positions of the marked points can be easily obtained and recorded by solving the forward kinematics of the surgical manipulator. The robot's base coordinate system is selected as the coordinate reference system for the marked points to simplify the computation described later. We marked approximately 60 points in four key positions for each vulnerable organ (as shown in Figure 3(b)). This process took less than 2 min. Near every important position the surgeon can repeatedly mark several points to achieve a better result. After the marking of all points, the algorithm merged the relatively close points (closer than a threshold value, which can be adjusted in different surgical

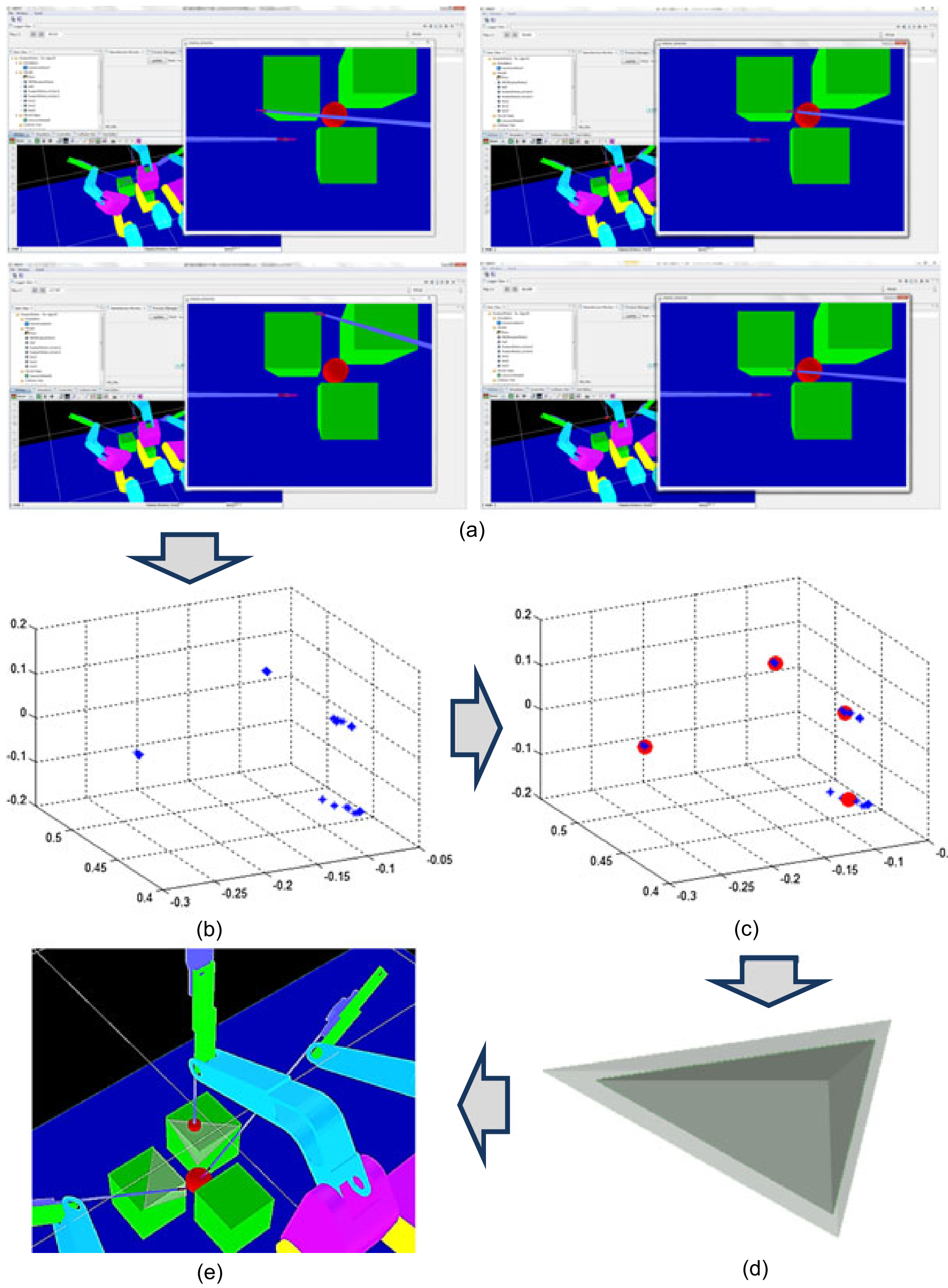

Figure 3. A demonstration of the new proposed virtual fixture generation method

DOI: 10.1002/rcs

operations) into one key point and its position stored in an array. We used four key points in the demonstration, as shown in Figure 3(c). The number of the key points needed to generate a virtual fixture is determined mainly by the geometrical shape of the organ or tissue to be protected and the precision requirement of the surgical manipulation. In practice, the surgeon can mark as many key points as he/she wishes to generate a more suitable virtual fixture for a complex-shaped organ/tissue. The more key points the surgeon marks, the more precise the virtual fixture's shape can be and naturally it costs more time and effort.

Based on the key design points, a convex hull can be automatically generated. Theoretically, this convex hull should be able to cover the organ (the green box) completely. However, it will be expanded moderately for safety's sake, usually by a factor of 1.1 or 1.2 to adapt to the organ's movement or deformation, as shown in Figure 3(d). The completed virtual fixture can be used to protect the organ (the green box) during the surgery (Figure 3(e)). After the first virtual fixture is generated, the procedure can be repeated to create more virtual fixtures until all fragile organs/tissues are well protected. Any number of virtual fixtures can be used at the same time during the surgery. A flow diagram of the proposed virtual fixture generation method is illustrated in Figure 4. As we can see this new method is simple and easy to use. This process can be conducted during the preoperative stage or even during the surgery, which makes its practical application very flexible and convenient. One thing to be noted is that the proposed method requires the movement or deformation of the protected organs to be relatively small or regular during the entire surgery. For easily deformable organs, such as the intestine, additional assistance tools are needed to restrain the organ positions first, then some virtual fixtures could be built to protect them.

Minimum distance calculation between the surgical instrument and all virtual fixtures

After virtual fixtures are generated we constrain the slave manipulator motion to ensure no inadvertent injury occurs during the surgery. The first step is to obtain the closest pair of points between the surgical instrument and all generated virtual fixtures where a collision is most likely to occur. This can be boiled down to the minimum distance problem among multiple convex polyhedrons, which is a problem commonly faced in many fields, especially in computer graphics. There are already numerous solutions to this problem, such as various kinds of bounding box methods (16–23), the Lin-Canny closest features algorithm and its derivative algorithms (the most famous one is the Voronoi-Clip algorithm) (24–27,35), the Enhanced GJK algorithm (28,29), fast solution algorithms using graphics hardware (30,31), the hierarchical presentation approach (32), the normal vector based calculation method (33), the particle swarm search algorithm (34), etc. In our research the generated virtual fixtures are simply-shaped convex polyhedrons. The surgical instrument mounted on the slave manipulator is usually slender rod-shaped, which could be treated as a line segment in the computation. The problem to be solved can be simplified to the minimum distance problem from a line segment to some simply-shaped convex polyhedrons.

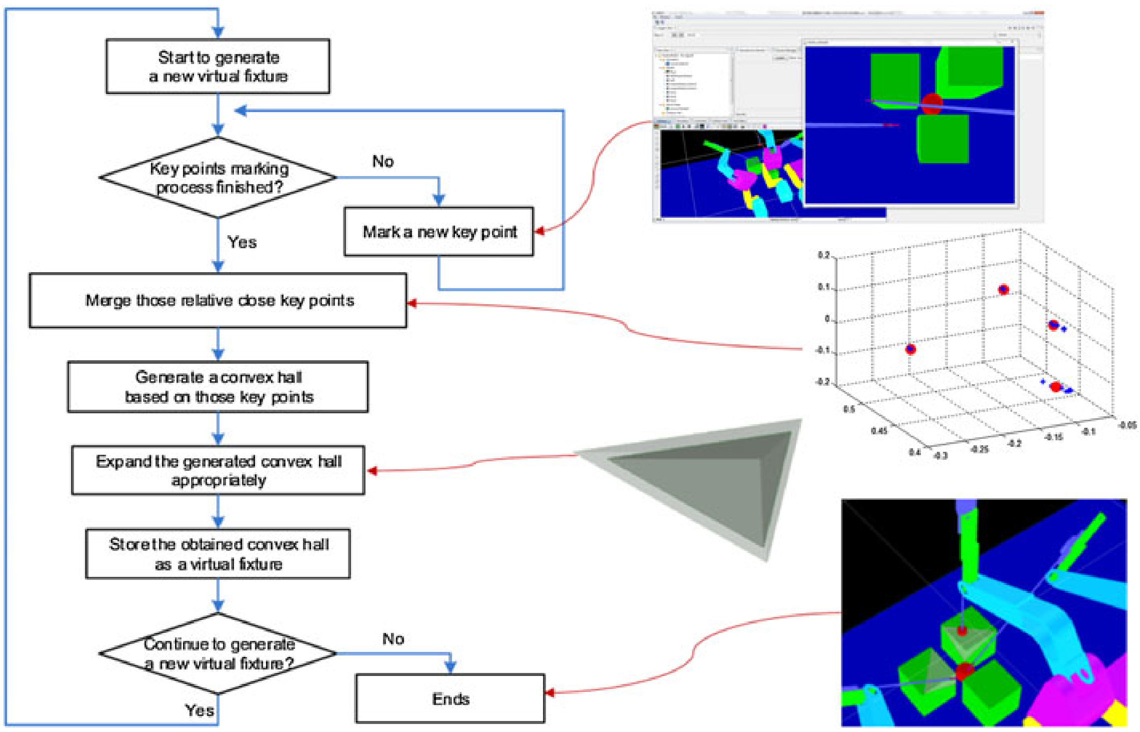

Figure 4. Flow diagram of the generation procedure of customizable virtual fixtures

DOI: 10.1002/rcs

The surgical robot control system is rigorous real-time demand, always requiring a high efficiency algorithm. Therefore, rather than using a traditional general algorithms, we propose a new fast solving algorithm that adheres to the characteristics of MIS robots. This new proposed algorithm is more efficient because it is specific to our application. More importantly, the proposed method has no specific requirement for convex polygon (virtual fixture) storage, which makes it easier to use.

To obtain the minimum distance between a line segment and a polyhedron in space, we need to calculate only the distance from this line segment to all faces of the polyhedron. The smallest value is the minimum distance (Figure 5). The minimum distance from a line segment PQ to a face (a triangle ABC) in space must be one of the following five distances (assuming that segment PQ does not intersect triangle ABC):

1. the minimum distance between segment PQ and segment AB;
2. the minimum distance between segment PQ and segment BC;
3. the minimum distance between segment PQ and segment AC;
4. the minimum distance from vertex P to triangle ABC;
5. the minimum distance from vertex Q to triangle ABC;

These five minimum distance candidates can be attributed to two cases: Case 1 ((1), (2) and (3)) involves solving the minimum distance between two segments; and case 2 ((4) and (5)) involves solving the minimum distance from a vertex to a triangle. For case 1 the minimum distance can be calculated as follows (assuming that there are two line segments PQ and AB in space):

1. If the feet of two line segments' common perpendicular are both located on segments PQ and AB, the distance between the two perpendicular feet is just the minimum distance between PQ and AB, as shown in Figure 6.
2. If any of the perpendicular feet are located outside PQ and AB, the minimum distance between the two segments should then be the smallest of the following four distances: the minimum distance from vertex P, Q to segment AB and the minimum distance from vertex A, B to segment PQ. The minimum distance calculation from a vertex to a line segment can be done based on plane geometry. As shown in Figure 7, through A, draw a line perpendicular to PQ. If the foot of the perpendicular lies on PQ, the distance between A and the perpendicular foot is the minimum distance between A and PQ. Otherwise, the minimum distance would be the smaller of the distances from A to P and from A to Q.

For case 2, the minimum distance between a vertex and a triangle can be calculated as follows (assuming that there are a vertex P and a triangle ABC in space):

1. If the foot of the perpendicular line from vertex P to triangle ABC lies within ABC, the distance from P to the foot of perpendicular is the minimum distance between P and ABC.
2. If the foot of the perpendicular line lies outside ABC, the minimum distance should then be the smallest of the minimum distances from vertex P to segments AB, BC and CA, as shown in Figure 8. The minimum distance calculation from a vertex to a line segment was already introduced above (Figure 7).

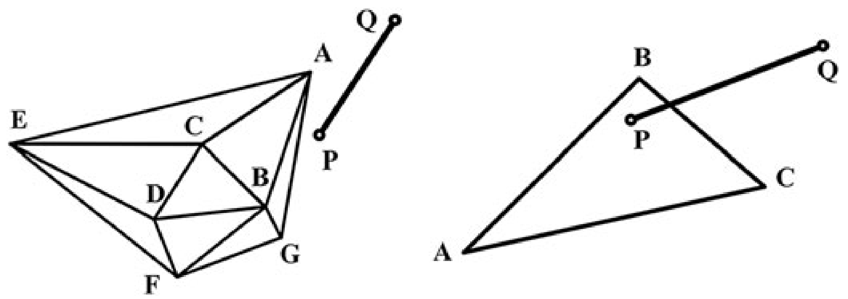

Figure 5. The smallest value among the minimum distances from the segment to all faces of the polyhedron is the minimum distance between the line segment and the polyhedron

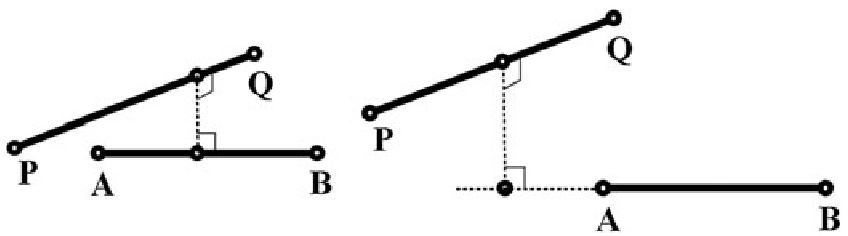

Figure 6. The minimum distance between two line segments in space

DOI: 10.1002/rcs

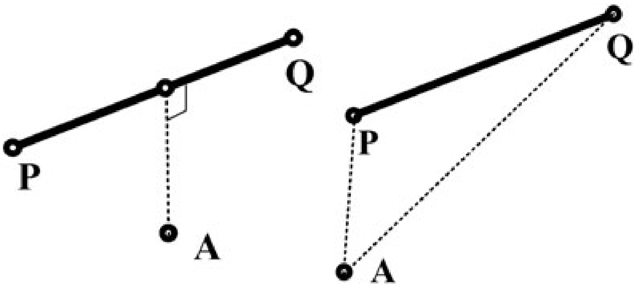

Figure 7. The minimum distance from a point to a line segment in space

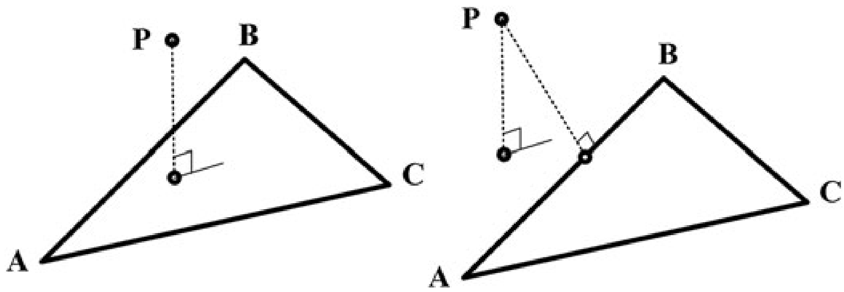

Figure 8. The minimum distance from a point to a triangle in space

In this way we can obtain the minimum distance between a line segment and a triangle in space, and further the minimum distance from this line segment to all polyhedrons in space. However, the traversal algorithm described above is ineffective. In each control interval the algorithm needs to refresh the minimum distance from the line segment to all triangles and then choose the smallest one. As the total number of triangles increases the time needed for calculation will also increase, which is undesirable for a MIS robotic system. Accelerating this algorithm is therefore necessary. For a MIS robot control system each control cycle usually costs less than 1 ms. In such a short period the displacement of the surgical instrument is relatively tiny. According to the continuity of motion the position of the surgical instrument in the next control interval should be somewhere close to its current position. The same thing also happens to the new location for the closest pair of points. The computation process can be accelerated based on this principle. In each control interval the area in which the closest pair of points might be located can be estimated based on its location in the previous control interval. We only need to calculate the minimum distance between the surgical instrument and those triangles within this possible area. In this way the computation in each control interval is greatly simplified.

As shown in Figure 9, for a convex polyhedron:

1. If the closest point S_1 on the convex polyhedron is located within a certain triangle in the previous

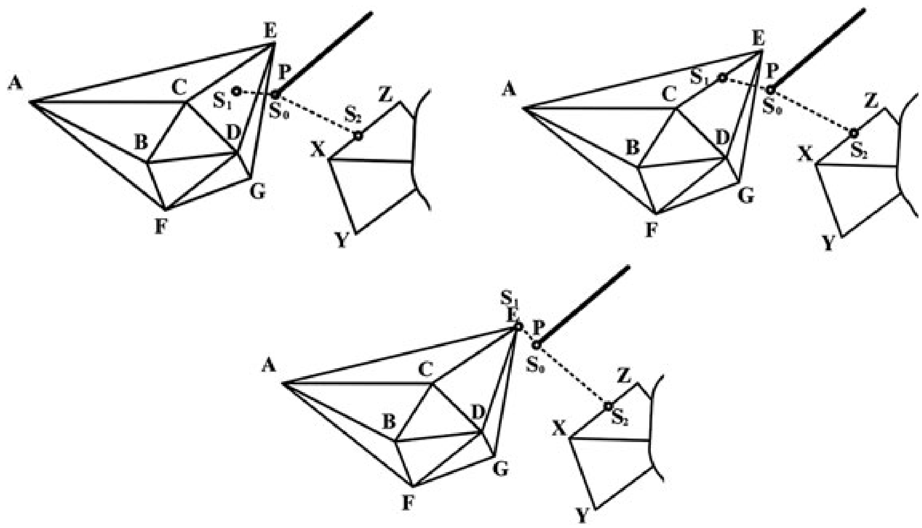

Figure 9. The relationship between the previous location and current location of the closest point on a convex polyhedron

DOI: 10.1002/rcs

control interval, it should still be there (within this triangle) in the current control interval.
2. If the closest point S_1 is located on a certain edge of the convex polyhedron in the previous control interval, then in the current control interval, S_1 might be located in one of those two triangles which contain this edge.
3. If the closest point S_1 is located on a certain vertex of the convex polyhedron in the previous control interval, then in the current control interval, S_1 might be located in one of the adjacent triangles which share this vertex.

Once we obtain the location of the closest point S_1 on a convex polyhedron, we can track it and predict its possible location in the future. In each control interval we only need to calculate the minimum distance between the surgical instrument and a few triangles instead of all of the triangles in space. In the above discussion only one virtual fixture is taken into consideration. When there is more than one virtual fixture, all of the candidate points (S_1, S_2, S_3 …… S_i) on each virtual fixture are tracked and the closest one is chosen. It must be pointed out that this method can only accelerate the computing process after obtaining the previous location of the closest point. During the first solution process the algorithm still needs to compute the minimum distances from the surgical instrument to all faces of the virtual fixtures. We can take some measures to accelerate the first solution process, such as using the Voronoi region to reduce the number of triangles to be computed. However, in practice, the virtual fixtures generated are usually simply-shaped and the total number of their faces would not be too large (usually smaller than 100). Besides, this time-consuming first solution process could be finished in the initialization phase as pre-computation, therefore it will not affect the real-time performance of the surgical teleoperation at all.

Spatial motion constrained method design for the MIS robot

Based on the minimum distance between the surgical instrument and all virtual fixtures a motion constrained algorithm is designed to prevent any inadvertent collision during the surgery. The artificial potential field (APF) method (36) is a commonly used method to constrain the spatial motion of a robot. Modeled after the potential fields' concept in physics, it generates a virtual potential field in the robot's workspace, usually consisting of a repulsive potential field used for obstacle avoidance and an attractive potential field for goal tracking purposes, as shown in Figure 10. The robot's motion can be constrained under the combined effect of the two types of virtual potential forces. The APF method is simple in principle and can be easily realized in the bottom control level, making it real-time. It is also very flexible, in that the generated artificial potential force can be easily modified according to the requirements of different applications. However, the local minimum trap is the biggest problem in this method. The local minimum trap might make the robot fall into a deadlock situation during autonomous movement (37). Other methods, such as the fuzzy logic control (38), the artificial neural network (ANN) (39), the vector field histogram (VFH) (40), the curvature-velocity method (41), the behavior optimization based method (42), were also widely used to constrain robot movement. The purpose of our research is to find a method to assist surgeons in a highly skilled manipulation task and provide haptic feedback to them. We therefore chose the relatively simple and efficient APF method. A repulsive potential force is utilized to prevent the surgical instrument from colliding with the virtual fixtures. We also designed another assistive force that can guide the surgical instrument to bypass the virtual fixtures and approach its original goal.

The definition of the artificial potential function varies in different applications. The repulsive potential function used in our research is defined as follows:

$$E_P\left(\overrightarrow{D}_{r2o}\right) = \begin{cases} \frac{1}{2}\lambda\left(\frac{1}{\left|\overrightarrow{D}_{r2o}\right|} - \frac{1}{d_{threshold}}\right)^2 & , \left|\overrightarrow{D}_{r2o}\right| \le d_{threshold} \\ 0 & , \left|\overrightarrow{D}_{r2o}\right| > d_{threshold} \end{cases}$$

where $\overrightarrow{D}_{r2o}$ denotes the minimum distance vector from the robot to the obstacle, and λ is a scaling factor. The repulsive potential force acting on the robot can be obtained as:

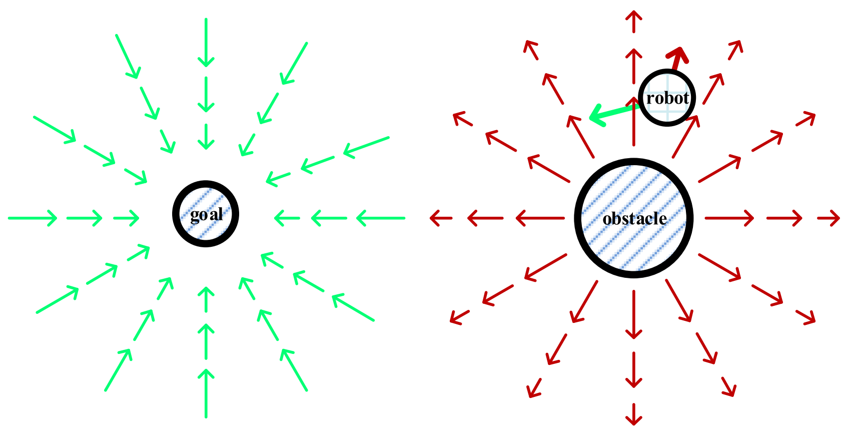

Figure 10. Two commonly used artificial potential fields: the attractive potential field and the repulsive potential field

DOI: 10.1002/rcs

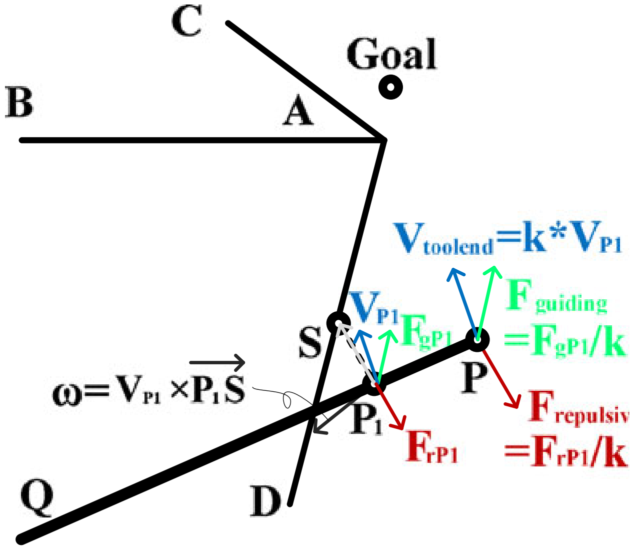

Figure 11. The repulsive potential force and the assistive guiding force acting on the instrument during teleoperation

$$\vec{F}_{repulsive}\left(\vec{D}_{r2o}\right) = \nabla E_P\left(\vec{D}_{r2o}\right) = \begin{cases} -\lambda\left(\frac{1}{\left|\vec{D}_{r2o}\right|} - \frac{1}{d_{threshold}}\right)\frac{1}{\left|\vec{D}_{r2o}\right|^2}\nabla\vec{D}_{r2o} & , \left|\vec{D}_{r2o}\right| \le d_{threshold} \\ 0 & , \left|\vec{D}_{r2o}\right| > d_{threshold} \end{cases}$$

The new assistive guiding force for assisting the surgical instrument in bypassing the virtual fixtures and moving towards its original goal can be calculated as follows: as shown in Figure 11, $\overrightarrow{P_1S}$ indicates the minimum distance vector from the instrument PQ to the virtual fixture at this moment, P_1 is the closest point on the surgical instrument while S is on the virtual fixture. The virtual repulsive force $\vec{F}_{rP1}$ acting on P_1 can be calculated based on the minimum distance vector $\overrightarrow{P_1S}\left(\vec{D}_{r2o} = \overrightarrow{P_1S}\right)$ as described before. The cross-product operation is used to generate the assistive guiding force. Note the velocity of the instrument's end P as $\vec{V}_{toolend}$, so we have the velocity of the closest point P_1 as $\vec{V}_{P1} = \vec{V}_{toolend}/k$, in which $k = \left|\overrightarrow{PQ}\right| / \left|\overrightarrow{P_1Q}\right|$, especially, when the tail end of the instrument P is the closest point P_1, $k = 1$. By employing a cross-product operation between the velocity $\vec{V}_{P1}$ and the minimum distance vector $\overrightarrow{P_1S}$, we get an auxiliary rotating vector $\vec{\omega}$ as $\vec{\omega} = \vec{V}_{P1} \times \overrightarrow{P_1S}$. Further, through another cross-product operation between the auxiliary rotating vector $\vec{\omega}$ and the minimum distance vector $\left(-\overrightarrow{P_1S}\right)$, we obtain another vector $\left(\vec{\omega} \times -\overrightarrow{P_1S}\right)$. The force along this new vector can guide the surgical instrument to bypass the virtual fixture (since the direction of this new vector is perpendicular to the minimum distance vector $\overrightarrow{P_1S}$) and move towards its original goal (due to the effect of the rotating vector $\vec{\omega}$). The magnitude of the guiding force can be defined differently according to the requirement of different surgical operations. This paper sets the guiding force in proportion to the velocity of the closest point $\left|\vec{V}_{P1}\right|$ and inversely proportional to the minimum distance $\left|\overrightarrow{P_1S}\right|$, as shown below:

$$\vec{F}_{guiding}\left(\vec{D}_{r2o}\right) = \begin{cases} -\gamma\left(\frac{1}{\left|\vec{D}_{r2o}\right|} - \frac{1}{d_{threshold}}\right) * \frac{1}{\left|\vec{D}_{r2o}\right|^3}\left(\vec{V}_{P1} \times \vec{D}_{r2o} \times -\vec{D}_{r2o}\right) & , \left|\vec{D}_{r2o}\right| \le d_{threshold} \\ 0 & , \left|\vec{D}_{r2o}\right| > d_{threshold} \end{cases} \quad \left(\vec{D}_{r2o} = \overrightarrow{P_1S}\right)$$

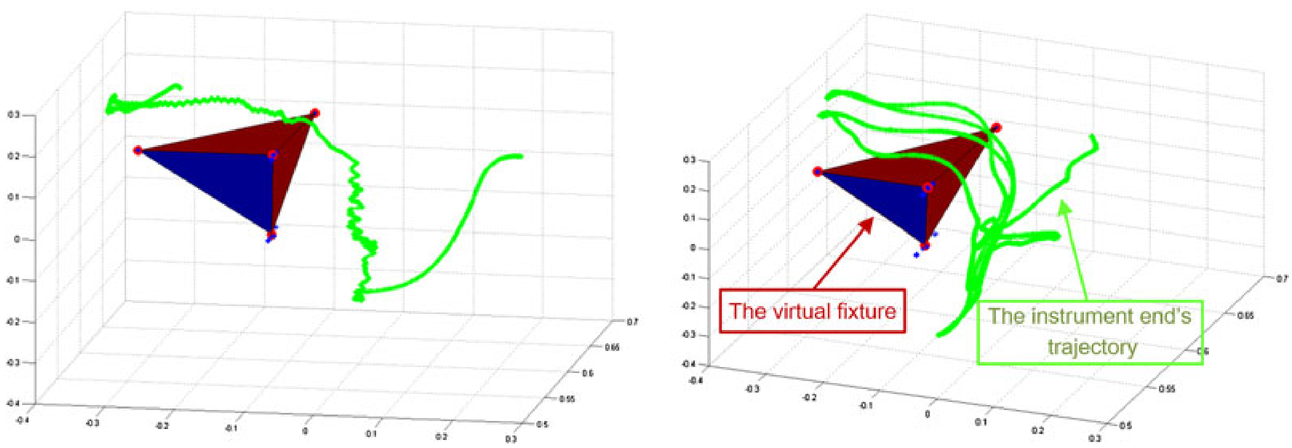

Figure 12. The trajectories of the tail end of the surgical instrument during the manipulations. (Left, only the repulsive potential force is applied; Right, both the repulsive potential force and the assistive guiding force are implemented)

DOI: 10.1002/rcs

where γ is also a scaling factor. After we get the repulsive potential force and the assistive guiding force acting on the closest point P_1, we still need to calculate the equivalent forces acting on the tail end of the instrument P, which will be used for the force feedback. The equivalent forces acting on P can be easily calculated as $\overrightarrow{F}_{guiding} = \overrightarrow{F}_{gP1}/k$ and $\overrightarrow{F}_{repulsive} = \overrightarrow{F}_{rP1}/k$, where $k = \left|\overrightarrow{PQ}\right| / \left|\overrightarrow{P_1Q}\right|$ as defined before. Using the combined effect of the repulsive potential force and the assistive guiding force, the constraint and guidance of the instrument's motion can be realized to assist the surgeon in the manipulation task.

The effectiveness of the proposed motion constrained method is examined using the virtual fixture generated in the previous section. Two manipulations are performed. During the first manipulation only the repulsive potential force is used to constrain the instrument's movement. Both the repulsive potential force and the assistive guiding force are implemented in the second manipulation. The distance threshold $d_{threshold}$ is set to 10 cm. The experiment results are shown in Figures 12, 13 and 14. Figure 12 shows the trajectories of the surgical instrument's tail end during the two manipulations. As we can see, with only the repulsive potential force implemented the trajectory becomes

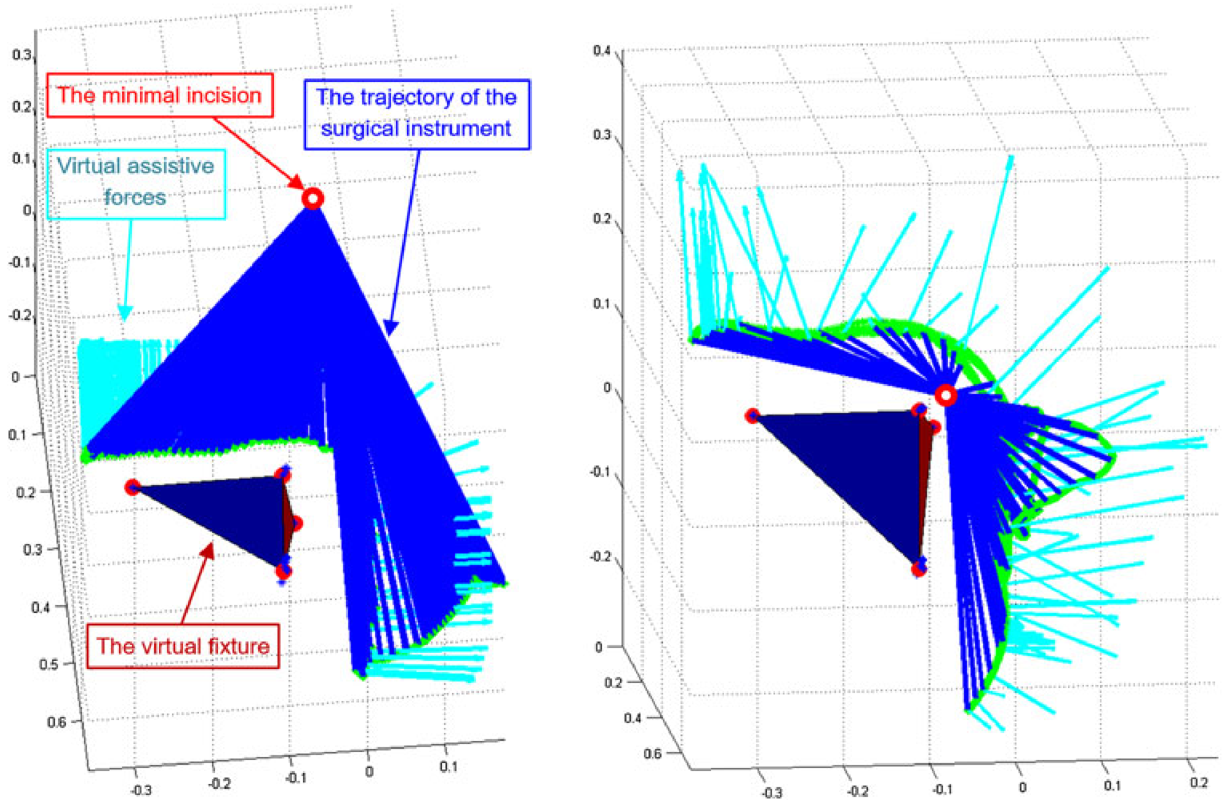

Figure 13. The virtual assistive force acting on the tail end of the surgical instrument. (Left, only the repulsive potential force is applied; Right, both the repulsive potential force and the assistive guiding force are implemented)

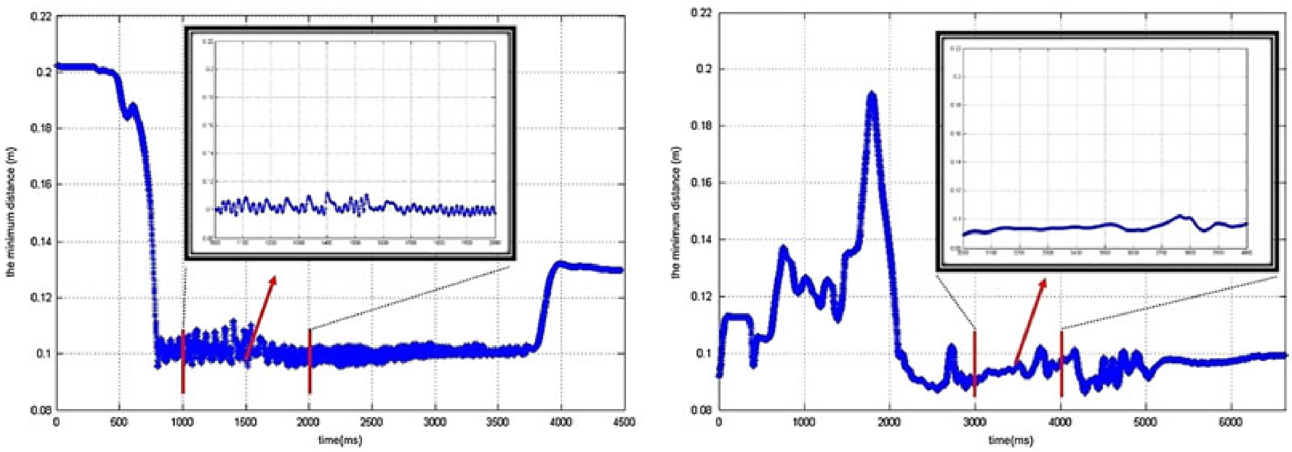

Figure 14. The minimum distance between the instrument and the virtual fixture during two manipulations. (Left, only the repulsive potential force is applied; Right, both the repulsive potential force and the assistive guiding force are implemented)

DOI: 10.1002/rcs

shaky when it moves close to the surface of the virtual fixture. That occurs because the repulsive potential force can only bounce the surgical instrument away from the fixture surface. After that the repulsive force will decrease. Due to the inertia of the user's hand the surgical instrument will overcome the reduced repulsive force and move towards the virtual fixture again. This procedure is repeated over and over, leading to shaky movement near the surface of the virtual fixture. During the second manipulation the assistive guiding force guided the surgical instrument past the virtual fixture quickly when the instrument moved towards the virtual fixture. The trajectory was relatively smooth. The virtual forces acting on the end of the surgical instrument during the two processes are illustrated in Figure 13. Figure 14 indicates the variation in the minimum distance between

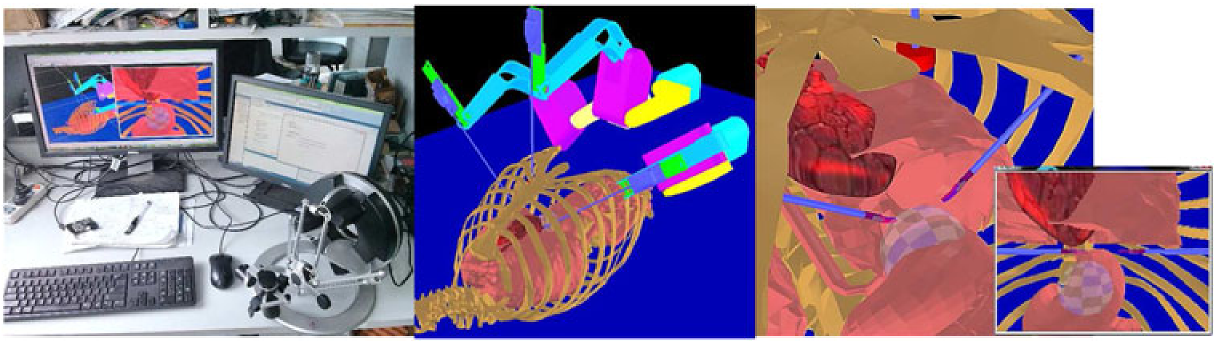

Figure 15. Our experiment platform: a 7-dof haptic interface Omega 7, a virtual MIS robot and a virtual surgical scene in the OpenHRP3 simulation environment

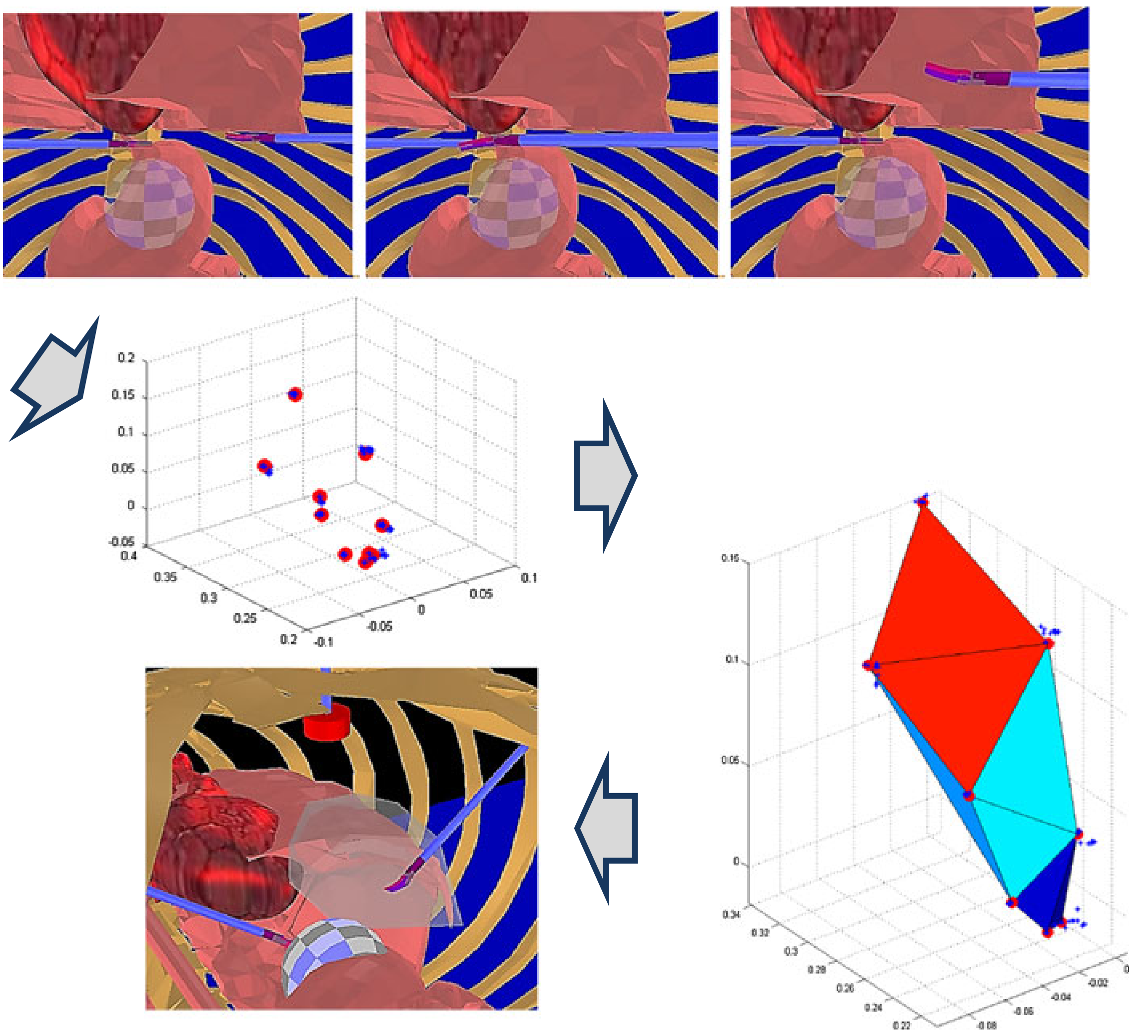

Figure 16. Generating a protection virtual fixture for the vulnerable organs

DOI: 10.1002/rcs

the instrument and the virtual fixture during two manipulation tasks. It can be seen clearly that the instrument movement was much smoother and more rapid when both the repulsive potential force and the assistive guiding force are implemented.

Results

Simulation experiments and results

To further illustrate the practical application of the proposed method we set up a more realistic virtual minimally invasive surgical scene in the open source simulation environment OpenHRP3, as shown in Figure 15. The 3D models of the organs used in our virtual surgery were obtained free from the SketchUp Model Library powered by Google. The layout of these organs is not technically accurate, but it is tolerable here since we only want to demonstrate how to use the proposed method and verify the effectiveness of the proposed motion constraint algorithm. The experimental platform consists of a 7-dof haptic device Omega 7 from Force Dimension Inc. as the input surgical handle, a virtual MIS robot and a virtual patient. We can use the Omega 7 to manipulate the virtual MIS robot to generate the required virtual fixtures and then perform various surgical actions.

During the experiment we manipulated the surgical instrument to carry out surgical actions on the stomach. The target area is indicated by a tessellated semitransparent sphere, as shown in Figures 15 and 16. Near the lesion there are important organs that require protection. We generated a virtual fixture to protect them from potential injury during surgery. The virtual fixture generation procedure is the same as before. We first manipulate the slave manipulator to mark some points near the surface of the organ. A virtual protection fixture is automatically generated and attached to this organ, as depicted in Figure 16. With the help of the generated virtual fixture we can confidently carry out various surgical actions. Whenever the surgical instrument gets too close to the virtual fixture (the distance threshold $d_{threshold}$ is set to 1 cm in this experiment), the virtual assistive forces prevent it from

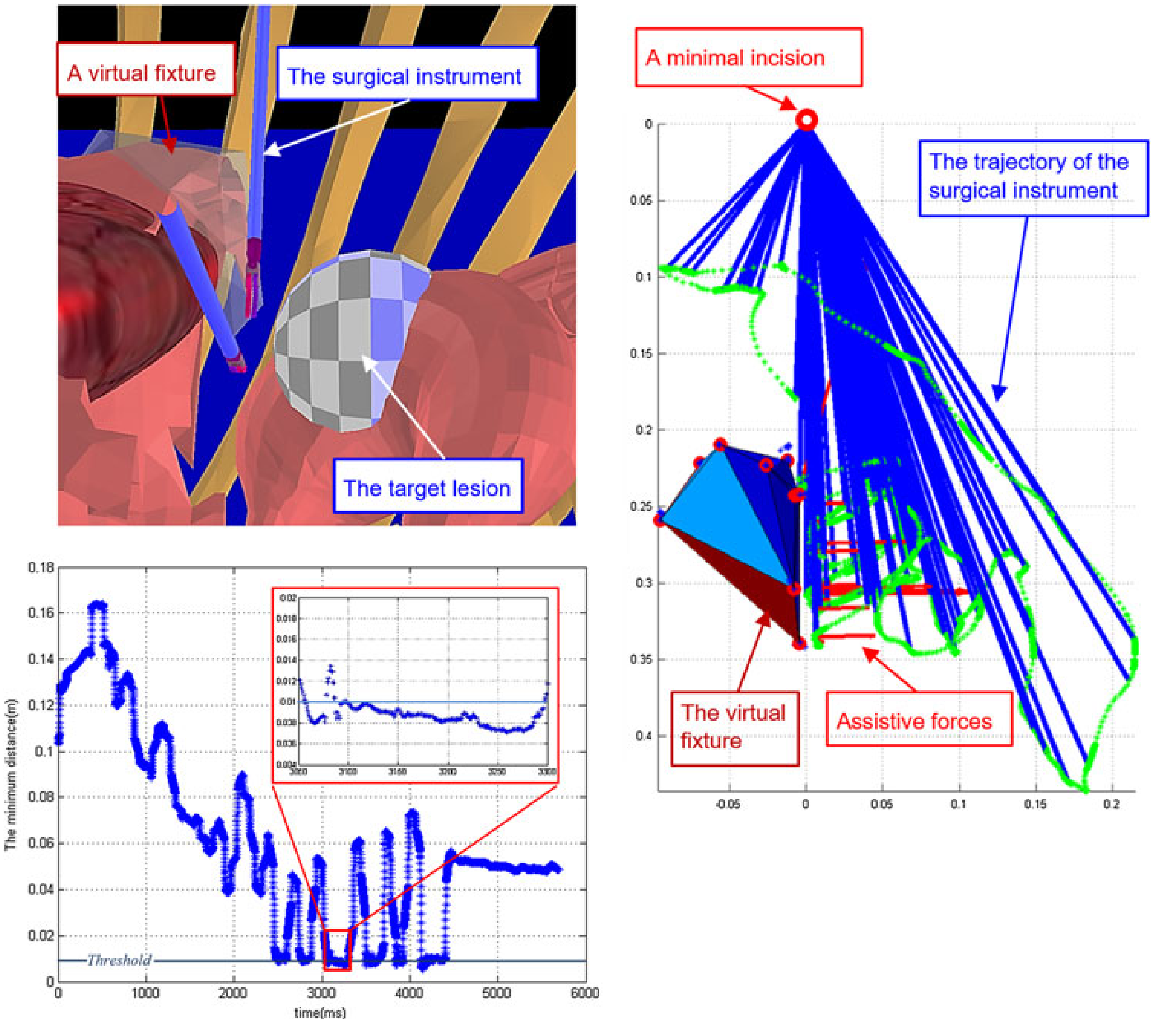

Figure 17. The generated virtual fixture can prevent the surgical instrument from colliding with the organ during the surgical manipulation

DOI: 10.1002/rcs

touching the organ and guide the surgical instrument around the virtual fixture (as shown in Figure 17). And the master device gives real-time feedback to the user.

Conclusion and discussion

This paper presented a novel forbidden region virtual fixture based motion constraint method proposed to assist surgeons in surgical teleoperations. The proposed method can prevent unexpected collisions between the surgical instrument and fragile organs around a target lesion, thus increasing the safety of robot-assisted minimally invasive surgery. First, a new simple and easy to use virtual fixture generation method was introduced, which can be easily applied to any master–slave MIS robot system. Unlike traditional methods, 3D modeling of the pathological area before surgery is not necessary. The surgeon only needs to manipulate the slave manipulator to mark a series of points near the surface of each vulnerable organ. Virtual fixtures that completely cover the vulnerable organs are automatically generated based on these marked points. In this way helpful virtual fixtures can be easily defined as the surgeon wishes before or even during the surgery. After generation of all needed virtual fixtures the control system computes and monitors the minimum distance between the surgical instrument and all virtual fixtures in real time. In light of the structural characteristics of our MIS robot a new fast solving algorithm was proposed. The proposed algorithm is efficient and has no specific requirements for virtual fixture storage. When the minimum distance is smaller than a threshold value, virtual assistive forces are generated to constrain the movement of the slave manipulator. Two types of assistive forces are utilized, a repulsive potential force that prevents the surgical instrument from colliding with the virtual fixtures and an assistive guiding force that guides the surgical instrument to bypass the defined virtual fixtures and move towards its original goal. Simulation experiments demonstrated the effectiveness of this algorithm. In the future we will develop the proposed method further and apply it to a real MIS robot developed in our laboratory. The ultimate goal of our research is to make this method available for minimally invasive surgery.

Acknowledgements

This work was supported by the National Natural Science Foundation (No. 81371650), the Biomedical Engineering Research Fund of Shanghai Jiao Tong University (No. YG2011MS03) and the Research Fund of the State Key Laboratory of Mechanical System and Vibration, China (No. MSV-MS-2010-01).

Conflict of interest

The authors have stated explicitly that there are no conflicts of interest in connection with this article.

Funding

No specific funding.

References

1. Invasiveness of surgical procedures. Wikipedia, the free encyclopedia. (updated 2013 March 12; cited 2013 March 23). Available from: http://en.wikipedia.org/wiki/Invasiveness_of_surgical_procedures#Minimally_invasive_procedure.
2. The da Vinci Surgical system (homepage on the Internet). Intuitive Surgical, Inc.; (cited 2013 March 23). Available from: http://www.intuitivesurgical.com/products/davinci_surgical_system/.
3. Virtual fixture. Wikipedia, the free encyclopedia. (updated 2013 January 24; cited 2013 March 23). Available from: http://en.wikipedia.org/wiki/Virtual_fixture
4. Rosenberg LB. Virtual fixtures: perceptual tools for telerobotic manipulation. IEEE Virtual Reality Annual International Symposium; Sep 18–22, 1993; Seattle, USA: 76–82.
5. Davies BL, Harris SJ, Lin WJ, *et al*. Active compliance in robotic surgery – the use of force control as a dynamic constraint. *Proc Inst Mech Eng H: J Eng Med* 1997; **211**(4): 285–292.
6. Park S, Howe RD, Torchiana DF. Virtual fixtures for robotic cardiac surgery. MICCAI '01: Proceedings of the 4th International Conference on Medical Image Computing and Computer - Assisted Intervention; October 14–17, 2001; Utrecht, the Netherlands: 1419–1420.
7. Li M, Taylor RH. Optimum robot control for 3D virtual fixture in constrained ENT surgery. MICCAI '03: Proceedings of the 6th International Conference on Medical Image Computing and Computer-Assisted Intervention; November 15–18, 2003; Montréal, Canada: 165–172.
8. Li M, Taylor RH. Spatial Motion Constraints in Medical Robot Using Virtual Fixtures Generated by Anatomy. ICRA '04: Proceedings of the 2004 IEEE International Conference on Robotics and Automation; 2004 26 April–1 May; New Orleans, LA, USA. 1270–1275.
9. Marayong P, Li M, Allison AM, Hager GD. Spatial motion constraints: theory and demonstrations for robot guidance using virtual fixtures. ICRA '03: Proceedings of the 2003 IEEE International Conference on Robotics and Automation; 30 June–3 July, 2003; Coimbra, Portugal: 1954–1959.
10. Abbott JJ, Marayong P, Okamura AM. Virtual fixture architectures for telemanipulation. ICRA '03: Proceedings of the 2003 IEEE International Conference on Robotics and Automation; June 30-July 3, 2003; Coimbra, Portugal: 2798–2805.
11. Abbott JJ, Marayong P, Okamura AM. Haptic virtual fixtures for robot-assisted manipulation. ISRR '05: 12th International Symposium of Robotics Research; October 12–15, 2005; San Francisco, CA, USA: 49–64.
12. Kapoor A, Li M, Taylor RH. Spatial motion constraints for robot assisted suturing using virtual fixtures. MICCAI '05: Proceedings of the 8th International Conference on Medical Image Computing and Computer-Assisted Intervention; Oct 26–30, 2005; Palm Springs, CA, USA: 89–96.
13. Bettini A, Marayong P, Lang S, *et al*. Vision assisted control for manipulation using virtual fixtures. *IEEE Trans Robot Automat* 2004; **20**(6): 953–966.
14. Becker BC, MacLachlan RA, Lobes JLA, *et al*. Vision-based control of a handheld surgical micromanipulator with virtual fixtures. *IEEE Trans Robot Automat* 2013; (99): 1–10.
15. Ryden F, Chizeck HJ. Forbidden-region virtual fixtures from streaming point clouds: remotely touching and protecting a beating heart. 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); Oct 7–12, 2012; Vilamoura-Algarve, Portugal: 3308–3313.
16. Van den Bergen G. Efficient collision detection of complex deformable models using AABB trees. *J Graphics Tools* 1997; **2**(4): 1–13.
17. Greß A, Zachmann G. Object-space interference detection on programmable graphics hardware. SIAM Conference on Geometric Modeling and Computing; November 9–13, 2004; Seattle, USA: 311–328.

DOI: 10.1002/rcs

18. Borrmann A, Schraufstetter S, van Treeck C, Rank E. An iterative, octree-based algorithm for distance computation between polyhedra with complex surfaces. Proceedings of the 2007 ASCE International Workshop on Computing in Civil Engineering; July 24–27, 2007; Pittsburgh, PA, USA: 103–110.
19. Quinlan S. Efficient distance computation between non-convex objects. ICRA '94: Proceedings of the 1994 IEEE International Conference on Robotics and Automation; May, 1994; San Diego, CA, USA: 3324–3329.
20. Larsen E, Gottschalk S, Lin MC, Manocha D. Fast proximity queries with swept sphere volumes. ICRA '00: Proceedings of the 2000 IEEE International Conference on Robotics and Automation; April 24–28, 2000; San Francisco, CA, USA: 3719–3726.
21. Gottschalk S, Lin MC, Manocha D. OBBTree: A hierarchical structure for rapid interference detection. Proceedings of the 1996 SIGGRAPH Conference; August 4–9, 1996; New Orleans, LA, USA: 171–179.
22. Klosowski JT. Efficient collision detection using bounding volume hierarchies of k-DOPs. *IEEE Trans Vis Comput Graphics* 1998; **4**(1): 21–36.
23. Jia Q, Zhang Q, Gao X, *et al*. Dynamic obstacle avoidance algorithm for redundant robots with pre-selected minimum distance index. *Robot* 2013; **35**(1): 17–22.
24. Ehmann SA, Lin MC. Accurate and fast proximity queries between polyhedra using convex surface decomposition. *Comput Graphics Forum* 2001; **20**(3): 500–511.
25. Lin MC, Canny JF. A fast algorithm for incremental distance calculation. ICRA '91: Proceedings of the 1991 IEEE International Conference on Robotics and Automation; April 9–12, 1991; Sacramento, CA, USA: 3719–3726.
26. Cohen JD, Lin MC, Manocha D, Ponamgi M. I-COLLIDE: an interactive and exact collision detection system for large-scale environments; Proceedings of the 1995 Interactive 3D Graphics Conference; April 9–12, 1995; Monterey, CA, USA: 189–196.
27. Mirtich B. V-Clip: Fast and robust polyhedral collision detection. *ACM Trans Graphics* 1998; **17**(3): 177–208.
28. Cameron S. Enhancing GJK: computing minimum and penetration distance between convex polyhedra. ICRA '97: Proceedings of the 1997 IEEE International Conference on Robotics and Automation; April 20–25, 1997; Albuquerque, New Mexico, USA: 3112–3117.
29. Cameron S, Stephen A. A comparison of two fast algorithms for computing the distance between convex polyhedral, Robotics and Automation. *IEEE Trans Robot Automat* 1997; **13**(6): 915–921.
30. Knott D, Pai DK. CInDeR: collision and interference detection in real–time using graphics hardware. Proceedings of the 2003 Conference on Graphics Interface; June 11–13, 2003; Halifax, Nova Scotia, Canada: 43–51.
31. Govindaraju NK, Redon S, Lin MC, Manocha D. CULLIDE: interactive collision detection between complex models in large environments using graphics hardware. HWWS '03: Proceedings of the 2003 ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware; July 26–27, 2003; San Diego, CA, USA: 25–32.
32. Fang X, Bao HJ, Wang PA, Peng QS. Algorithm for fast calculating the nearest distance between space point and arbitrary polyhedron. *J Comput-Aid Design Comput Graphics* 2001; **13**(9): 788–792.
33. Tan GY, Li GH, Chen D, Cui XM. Detection of collision and interference between 3D objects using normal vector, IVEC '99: Proceedings of the 1999 IEEE International Vehicle Electronics Conference; September 6–9, 1999; Changchun, China: 286–289.
34. Zhang ZH, Chou WS. Research on precise collision detection between deformable objects. *Comput Eng Applic* 2009; **45**(1): 176–178.
35. Lin MC, Gottschalk S. Collision detection between geometric models: a survey. Proceedings of the IMA Conference on Mathematics of Surfaces; August 31–September 2, 1998; Birmingham, UK: 37–56.
36. Khatib O. Real-time obstacle avoidance for manipulators and mobile robots. *Int J Robotics Res* 1986; **5**(1): 90–98.
37. RoboTurk 2011 Team Description - Small Size Robot League. (cited 2013 March 25). Available from: http://robocupssl.cpe.ku.ac.th/tdp/2011/Roboturk_tdp_2011.pdf
38. Héctor García DM, Fernando JP, José MGS. Path planning combined with Fuzzy control for autonomous ships. CIMSA '10: Proceedings of the 2010 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications; September 6–8, 2010; Taranto, Italy: 86–91.
39. Li CH, Li YB, Fan C. Dynamic obstacle avoidance Algorithm for the mobile robot. *J Shandong Univ (Eng Sci)* 2007; **37**(5): 1–5.
40. Borenstein J. The vector field histogram - fast obstacle avoidance for mobile robots. *IEEE Trans RobotAutomat* 1991; **7**(3): 278–288.
41. Reid GS. The curvature-velocity method for local obstacle avoidance, ICRA '96: Proceedings of the 1996 IEEE International Conference on Robotics and Automation; April 22–28, 1996; Minneapolis, Minnesota, USA: 3375–3382.
42. Yang JD, Cai ZS, Yang JH. Efficient method of obstacle avoidance based on behavior fusion for mobile robots. *J Mech Eng* 2012; **48**(5): 10–14.

DOI: 10.1002/rcs