

# Modeling of a Transfer Task in Computer Assisted Surgical Training

**Minsik Hong**

Dept. of Electrical and Computer Engineering  
University of Arizona, Tucson, Arizona  
mshong@email.arizona.edu

**Jerzy W. Rozenblit**

Dept. of Electrical and Computer Engineering  
Department of Surgery  
University of Arizona, Tucson, Arizona  
jr@ece.arizona.edu

## ABSTRACT

A realistic, model-based training scenario generation is proposed to support haptic guidance system for laparoscopic surgery skills training. In this paper, a simulation model of a peg transfer task (one of the hands-on exam topics in Fundamentals of Laparoscopic Surgery) is presented to provide a realistic challenge to a trainee. A simple specification based on an extensible markup language and Bezier curve based path generator are described to support the guidance system. Simulation results illustrate the feasibility of the proposed model.

## Author Keywords

Simulation-based surgical training, Medical Simulation, Laparoscopy, Haptic Guidance.

## 1. INTRODUCTION

As a simulation based training device, Computer Assisted Surgical Trainer (CAST) [1] has been developed for laparoscopic surgery skills training. Laparoscopic surgery is an advanced technology with many benefits including less blood loss and fast recovery time. However, becoming a skilled surgeon is rather challenging, thus trainees need training devices to practice in a simulated setting prior to operating on a patient. To master surgical skills, trainees can use several commercial training devices that range from low cost box trainers to expensive virtual reality trainers.

Unlike several other virtual reality surgical trainers, we designed and implemented an exchangeable task scenario and a haptic guidance system to provide a more realistic training environment [2]. Exchangeable cassettes enable a user to employ realistic materials such as 3D printed organ models or simple surgical practice setups. The haptic guidance system provides *force inputs* for a user so that he or she can get assistance in moving surgical instruments.

To support the guidance module (called *optGuide*) in CAST, the optimal and collision free path generator (*optMIS*) was designed and presented in [3]. The *optGuide* uses generated paths as reference inputs for the controller.

Also, the visual guidance module (*optViz*) overlays a reference path over the live camera image to provide visual, assistive information. Figure 1 illustrates an example of visual guidance for a simple “blocks world” training scenario.

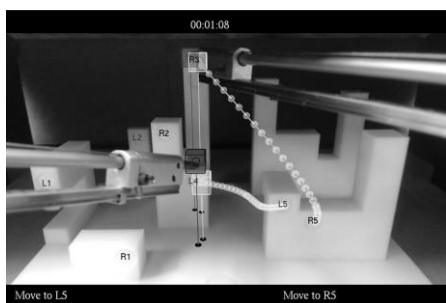
In this paper, we propose a modeling method of a transfer task to provide a realistic training scenario (such a task is a classic and fundamental training exercise in medical student and resident training; its model is also a good abstraction from which more complex exercises can be constructed.). This modeling method is intended as the basis for abstractions from which new task descriptions could be generated with relative ease. Such descriptions will then serve data for the haptic guidance system to provide various training scenarios.

The rest of this paper is organized as follows. A peg transfer task and its model are addressed in Section 2. In Section 3, task description and path generation methods are illustrated for the transfer task. Simulation and its results are presented in Section 4. Conclusions are given in Section 5.

## 2. REALISTIC TRAINING SCENARIO

The Fundamentals of Laparoscopic Surgery (FLS) program was designed by the Society of American Gastrointestinal and Endoscopic Surgeons (SAGES) for surgical residents and medical students to learn the basics of laparoscopic skills [4][5]. The FLS program includes a hands-on exam to evaluate trainee’s surgical skills. FLS exam includes five tasks: peg transfer, precision cutting, ligating loop, suture with extracorporeal knot, and suture with intracorporeal knot. For a visual explanation of the exam steps, we refer the reader to video demonstrations provided by VTi medical in [6].

Each task has its own maximum time limit and scoring formula. A trainee should perform efficiently without errors when he or she takes this exam. Exceeding the time limit is not allowed during the exam. According to written instructions [4], a proctor will ask a trainee to stop if the trainee does not complete a task within the time limit. However, completing a task within the time limit does not



**Figure 1.** Example of augmented reality overlay in *optViz*

guarantee a high score because the accuracy of performing a task is also important.

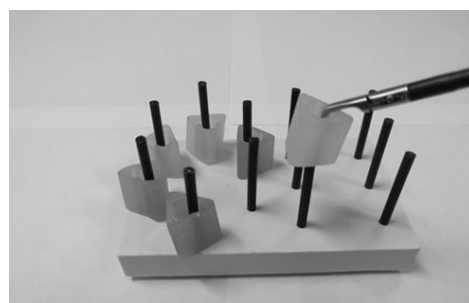
In this paper, we have selected the peg transfer task that serves as a case study, the basis from which simulation models can be developed to support more complex haptic guidance in CAST. In what follows, we describe the task in more detail and show how we model it so that more general abstractions can be developed for other surgical training steps.

### 2.1. Peg Transfer Task

Peg transfer is the first task of the FLS exam. It requires two surgical instruments (e.g., Maryland dissectors), one pegboard that has 12 pegs, and six rubber ring-like objects (triangles). Figure 2 illustrates the setup.

The instructions for this task are:

- 1) Six triangles are placed on the same side of the board. The recommendation is to place the triangles on the nearest side of a trainee's non-dominant hand. For example, if you are a right handed person, place triangles on the left side of the peg board.
- 2) A trainee grasps one of six triangles with the non-dominant hand.
- 3) A trainee transfers the triangle from the non-dominant hand to dominant hand in mid-air. During this transfer, it is not allowed to use pegs or anything else as assistive means (i.e., this transfer must be done in mid-air.).
- 4) A trainee places the triangle on a peg that is on the opposite side of the peg board.
- 5) Likewise, a trainee should transfer all remaining five triangles to the opposite side.
- 6) Once all six triangles are transferred by using the dominant hand, grasp a triangle, transfer it to the non-dominant hand, and place it on a peg that is on the original side of the peg board.
- 7) Likewise, a trainee should transfer all the remaining five triangles from the opposite side to the original side.



**Figure 2.** Peg transfer task



**Figure 3.** Grasper-type instruments

To complete the task, it is necessary to master how to manipulate grasper-type instruments. In the next section, basic actions for graspers are presented to support the guidance system.

### 2.2. Basic actions for grasper-type instruments

While performing a laparoscopic surgery, a surgeon generally uses different types of instruments such as dissectors, graspers, needle-holders, and scissors. Figure 3 depicts examples of grasper-type. Whenever a surgeon uses grasping instruments, he or she may perform particular actions such as moving an instrument, picking up an object, and carrying an object. To generalize the movements of grasper-type instruments, we model these actions as follows:

- 1) *Move* an instrument to a specific position.
- 2) *Grasp* (pick up) an object using a specific instrument.
- 3) *Carry* an object to a specific position using a specific instrument.
- 4) *Transfer* an object from a specific instrument to the other instrument.
- 5) *Place* (put down) an object on a certain location.

To better understand the above actions (and their potential sequencing), assume that a trainee is a right handed person and performs the peg transfer task. First, the trainee *moves* the left instrument toward a triangle and *grasps* the triangle. Second, the trainee will *carry* the triangle to somewhere in mid-air and *transfer* it to the right instrument. Finally, the trainee will *carry* the triangle and *place* it on a peg.

### 3. TASK GENERATION

In order to provide visual and haptic guidance in the CAST system, a task model is needed from which specific actions are generated. The *task generator* creates specific actions for a trainee to exercise and learn a particular laparoscopic surgery skill. In this section, we present the specification and implementation of the generator for the transfer task.

#### 3.1. Description of configuration and states for peg transfer task

Two grasper-type instruments, 12 pegs, and 6 rubber rings are modelled as follows.

- Instrument:  $i_k$  where  $k \in \{left, right\}$
- Peg:  $p_l$  where  $l \in \{1, 2, \dots, 12\}$
- Triangle:  $t_m$  where  $m \in \{1, 2, \dots, 6\}$
- Object:  $o \in \{i_k, p_l, t_m\}$

Object ( $o$ ) represents one of instruments, pegs, or triangles and has physical information such as height and width. One of the key physical information is the position of an object because each action is highly related to the location of an object. The position of an object is presented as follows.

- Instrument ( $i_k$ ) tip position:  $q_{i_k}$
- Position of the center of the peg ( $p_l$ ) bottom:  $q_{p_l}$
- Position of the center of the triangle ( $t_m$ ) bottom:  $q_{t_m}$
- “Mid-air” position between  $u$  and  $v$ :  $q_{uv}$ , where  $u, v \in \{i_k, p_l, t_m\}$
- Position  $q_x$  where  $x \in \{i_k, p_l, t_m, uv\}$

Based on a defined object ( $o$ ) and its position ( $q_x$ ), the state is described as follows:

- **grasperEmpty**( $i_k$ ) – The grasper of instrument  $i_k$  is empty.
- **holding**( $i_k, t_m$ ) – The grasper of instrument  $i_k$  holds an object  $t_m$ .
- **objAtLoc**( $o, q_x$ ) – the object  $o$  is located at the location  $q_x$
- **close**( $q_1, q_2$ ) – the location  $q_1$  is close to the location  $q_2$ , where “close” denotes a proximal to another object with a range established by the training scenario designer, for example, 2mm.

#### 3.2. Required actions for modeling

A task is decomposed task into basic, elementary actions. The concept of STRIPS (STanford Research Institute Problem Solver)-form forward rules (F-rules) [7] for robot planning is used to describe these actions. STRIPS-form F-rules were proposed for a “robot program-solving system” that find a solution given an initial state, a goal state, and a

world model. There are three components for STRIPS-form F-rules – precondition formula, delete list, and add formula. In [8], *precondition* and *effect* have been used to implement a STRIPS-like task planning for robot learning. Similarly, CAST’s task generator uses the precondition and effect to describe each action.

Precondition and effect are:

**Action1:** *move*  $i_k$  to  $q_x$

- Precondition: **grasperEmpty**( $i_k$ )
- Effect: **objAtLoc**( $i_k, q_x$ )

**Action2:** *grasp*  $t_m$  at  $q_x$  using  $i_k$

- Precondition: **grasperEmpty**( $i_k$ ) & **close**( $q_{i_k}, q_x$ ) & **objAtLoc**( $t_m, q_x$ )
- Effect: **holding**( $i_k, t_m$ )

**Action3:** *carry*  $t_m$  to  $q_x$  using  $i_k$

- Precondition: **holding**( $i_k, t_m$ )
- Effect: **objAtLoc**( $t_m, q_x$ ) & **objAtLoc**( $i_k, q_x$ )

**Action4:** *transfer*  $t_m$  from  $i_k$

- Precondition: **close**( $q_{i_k}, q_{t_m}$ ) & **close**( $q_{i_k}, q_{t_m}$ ) & **holding**( $i_k, t_m$ ) & **grasperEmpty**( $i_k$ )
- Effect: **grasperEmpty**( $i_k$ ) & **holding**( $i_k, t_m$ )

**Action5:** *place*  $t_m$  at  $q_x$  using  $i_k$

- Precondition: **holding**( $i_k, t_m$ ) & **close**( $q_{i_k}, q_x$ ) & **close**( $q_{t_m}, q_x$ )
- Effect: **grasperEmpty**( $i_k$ ) & **objAtLoc**( $t_m, q_x$ )

#### 3.3. Task Description based on Extensible Markup Language

Several task description methods have been proposed for robot programming. In [9], text-based and graphical robot programming methods are presented. As a text-based method, eXtensible Markup Language (XML) is used for internet-oriented robot motion description language, RoboML [10]. As a simple and flexible text based language, XML has several benefits [11]. It is not only a machine readable, but also a human readable language. Also, it is easy to use and it does not require a very complex editor for writing and editing. Because of these benefits, XML-based task description is proposed for CAST.

An XML document consists of tags to describe elements. Based on predefined tags, a desired task is described. Several predefined tags are listed below.

- `<peg name="" h="" d="" x="" y="" z="" />`

- `<triangle name="" h="" w="" x="" y="" z="" />`
- `<sub-task></sub-task>`
- `<action type="" instrument="" />`
- `<object name="" />`
- `<position type="" name="" />`
- `<mid-air></mid-air>`

where, `<sub-task>` tag contains at least one `<action>` tag to describe a sub task. Also, `<mid-air>` tag has two position tags to determine the interpolated location in mid-air. Using these tags, a user can create an XML document to describe a particular training task. The examples of using XML tags are illustrated in Section 4.

### 3.4. Path generation

A user can express a specific task for the peg transfer task using the task description XML document. For the haptic guidance system, it is necessary to provide a reference path. We now demonstrate a simple path generation method. Bezier curves have been widely used in computer graphics as parametric curves [12]. We use them for the peg transfer task to set up haptic guidance in CAST.

The shape of the Bezier curve is determined by control points. Unlike a cubic spline curve, this curve does not contain all the control points. In general, the curve starts at the first control point and ends at the last control point. Given  $n$  control points  $(\mathbf{p}_i \text{ in } \mathbf{R}^m, \text{ where } i = 1, 2, \dots, n)$ , Bezier curve  $(P(u))$  is defined as follows:

$$P(u) = \sum_{i=0}^n \beta_i^n(u) \mathbf{p}_i, 0 \leq u \leq 1 \quad (1)$$

where  $u$  is a parameter and  $\beta_i^n(u) = \binom{n}{i} u^i (1-u)^{n-i}$ .

Generally, a *move* or a *carry* action in the transfer task is a restricted motion because of the pegs. For example, after grasping a triangle on a specific peg, a trainee should lift up the triangle first, and carry it to the destination position. To represent this restricted motion, a Bezier curve with line segments is used. The workspace of the peg transfer task is a three dimensional space so that control point  $\mathbf{p}_i = (x_i, y_i, z_i)$  is in  $\mathbf{R}^3$ . Figure 4 illustrates an example of the generated path for *carry* action (*carry  $t_m$  to  $q_x$  using  $i_k$* ). There are three control points ( $\mathbf{p}_0$ ,  $\mathbf{p}_1$ , and  $\mathbf{p}_2$ ) for the Bezier curve ( $\widehat{\mathbf{p}_0 \mathbf{p}_1 \mathbf{p}_2}$ ) and two end points ( $\mathbf{p}_s$  and  $\mathbf{p}_e$ ) for line segments ( $\widehat{\mathbf{p}_s \mathbf{p}_0}$  and  $\widehat{\mathbf{p}_2 \mathbf{p}_e}$ ).  $\mathbf{p}_s$  is the current position of instrument  $i_k$  and  $\mathbf{p}_e$  is the destination position  $q_x$ . These start and end point are:

$$\mathbf{p}_s = (x_s, y_s, z_s) = q_{i_k}, \mathbf{p}_e = (x_e, y_e, z_e) = q_x.$$

To determine three control points, estimated velocity ( $v_z$ ) and desired velocity ( $v_{xy}$ ) are used based on the following method.

$$\mathbf{p}_0 = (x_0, y_0, z_0) = (x_s, y_s, z_s + h_{peg} + h_{triangle}),$$

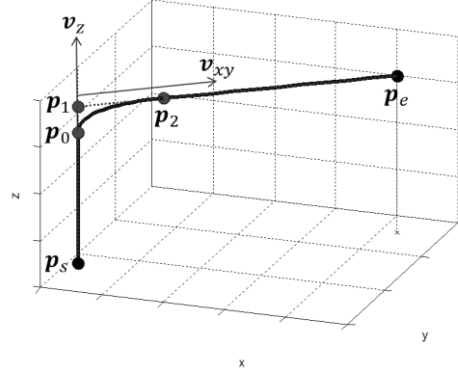


Figure 4. Bezier curve based path generation

$$\mathbf{p}_1 = (x_1, y_1, z_1) = (x_0, y_0, z_0 + d_z),$$

$$\mathbf{p}_2 = (x_2, y_2, z_2) = (x_1 + d_x, y_1 + d_y, z_1),$$

where  $h_{peg}$  is the height of peg and  $h_{triangle}$  is the height of a triangle. Also,  $d_x$ ,  $d_y$ , and  $d_z$  are determined as follows:

$$(d_x, d_y) = d_{xy} \cdot \vec{u}_{xy},$$

$$d_z = \frac{|v_z|^2}{a_d},$$

where  $a_d$  is the desired acceleration,  $\vec{u}_{xy} = v_{xy}/|v_{xy}|$ , and  $d_{xy} = |v_{xy}|^2/a_d$ .

## 4. SIMULATION RESULTS

The proposed task description and path generation are simulated using MATLAB before employing the model in the actual CAST system (The peg board is defined in a task description XML document.). Figure 5 depicts the description of peg board configuration based on `<peg>` and `<triangle>` tags. In the case of triangles, the locations are updated according to proper actions. Likewise, physical information of two instruments is defined in the configuration file of CAST as XML format.

The task description and the physical information are used as inputs of to MATLAB simulation. Also, the simulator updates the position of each instrument and the position of each triangle every 50ms. The updated position information is used to determine preconditions and effects for each action. In this simulation, the simulator generates ideal movements of instruments (i.e., traversing the guidance path) in order to verify the proposed model.

A surgical procedure is generally a sequential process that consists of a set of sub-tasks. Therefore, the XML document for the task description has also a set of sub-tasks represented as a sequential process. To specify each sub-task, `<action>` tag is used in `<sub-task>` tag. Figure 6 illustrates an example of XML based task description for the peg transfer task. This example has four sub tasks for the left instrument and each sub-task has a single action. Action tag in sub-task describes one of five actions.

```

<pegs>
  <peg name="peg1" h="3.3" d="0.3" x="-3.95" y="10.5" z="4.2"/>
  <peg name="peg2" h="3.3" d="0.3" x="-3.95" y="8.5" z="4.2"/>
  ...
  <peg name="peg12" h="3.3" d="0.3" x="4.15" y="7.6" z="4.2"/>
</pegs>

<triangles>
  <triangle name="triangle1" h="1.5" w="1.5" x="-3.95" y="10.5" z="4.2"/>
  ...
  <triangle name="triangle6" h="1.5" w="1.5" x="-0.65" y="8.5" z="4.2"/>
</triangles>

```

Figure 5. XML based peg board description

```

<sub-task>
  <action type="move" instrument="left">
    <position type="to" name="peg2"/>
  </action>
</sub-task>
<sub-task>
  <action type="grasp" instrument="left">
    <object name="triangle2"/>
    <position type="at" name="peg2"/>
  </action>
</sub-task>
<sub-task>
  <action type="carry" instrument="left">
    <object name="triangle2"/>
    <position type="to" name="mid-air"/>
    <mid-air>
      <position type="from" name="peg2"/>
      <position type="to" name="peg8"/>
    </mid-air>
  </action>
</sub-task>
<sub-task>
  <action type="transfer" instrument="left">
    <object name="triangle2"/>
    <position type="at" name="mid-air"/>
    <mid-air>
      <position type="from" name="peg2"/>
      <position type="to" name="peg8"/>
    </mid-air>
  </action>
</sub-task>

```

Figure 6. XML based task description for peg transfer task

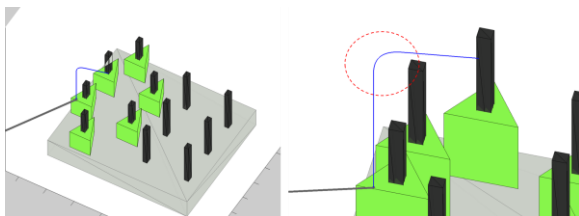


Figure 7. Bezier curve based guidance path for carry action

For example, the action, “carry triangle2 to mid-air location between peg2 and peg8 by using left instrument”, is presented in Figure 6. Also, the action, “transfer triangle 2 from the left instrument to the right instrument in mid-air location between peg2 and peg8” is presented in Figure 6.

The simulator verifies the sequence of actions for a training scenario that is described in the XML document and

evaluates validity of preconditions and effects for each action. Also, the simulator displays the corresponding Bezier curve based guidance paths for each sub-task and verifies these paths.

The model expressed in XML task description allows us to instantly generate instrument trajectories (this is done in the *optGuide* module of CAST). Thus, we can now support the training process by a) haptic assistance to the uses to navigate the instruments along these paths, and b) visual assistance by augmenting the camera image of the training task with rendition of optimal instrument paths.

Figure 7 illustrates the example of generating instant guidance path for *carry* actions – *carry* a triangle to a mid-air point using the left instrument.

## 5. CONCLUSION

In this work, modeling of peg transfer task for computer assisted surgical trainer has been presented. We have proposed the XML based task description and Bezier curve based guidance path generation methods. Also, we have simulated the proposed model in MATLAB before applying it to the CAST.

The peg transfer task serves as an instance to demonstrate the potential utility of a general task description model and its simulation. The key benefit of the proposed approach is the ability to relatively easily construct abstractions of more complex tasks and to simulate them, thus verifying their feasibility. Based on the simulation results, we then can fine tune the complex control algorithms of the haptic guidance module (*optGuide*) and the visualization assistance module (*optViz*) to provide highly realistic training scenarios.

## REFERENCES

1. Rozenblit, J. W., et al. "The Computer Assisted Surgical Trainer: Design, Models, and Implementation," Proc. of the 2014 Summer Simulation Conference, Modeling and Simulation in Medicine, pp. 211-220, Monterrey, CA July 2014.
2. Minsik Hong and Jerzy W. Rozenblit, “Haptic Guidance with Fuzzy Control in Simulation-Based Surgical Training.” Proc. of the 2015 Spring Simulation Conference, Modeling and Simulation in Medicine, vol. 47, no. 5, pp. 25-30, 2015.
3. Napalkova, Liana, et al. "An Optimal Motion Planning Method for Computer-Assisted Surgical Training." Applied Soft Computing, vol. 24, pp. 889-899, 2014.
4. Fundamentals of Laparoscopic Surgery, <http://www.flsprogram.org>.
5. Soper, Nathaniel J., and Gerald M. Fried. "The fundamentals of laparoscopic surgery: its time has come." Bull Am Coll Surg vol. 93, no. 9, pp.30-32, 2008.

6. FLS Trainer Skill Exercises, VTi medical,  
<https://www.youtube.com/watch?v=ROUGZ79Paxk>
7. Nilsson, Nils J. *Principles of artificial intelligence*. Morgan Kaufmann, 1986.
8. Ekvall, Staffan, and Danica Kragic. "Robot learning from demonstration: a task-level planning approach." *International Journal of Advanced Robotic Systems*, vol. 5, no. 3, pp. 223-234, 2008.
9. Biggs, Geoffrey, and Bruce MacDonald. "A survey of robot programming systems." In *Proceedings of the Australasian conference on robotics and automation*, pp. 1-3. 2003.
10. Makatchev, Maxim; Tso, S. K, "Human-robot interface using agents communicating in an xml-based markup language." *Robot and Human Interactive Communication, 2000 (RO-MAN 2000)*, pp. 270-275, 2000.
11. <http://www.w3.org/XML/>
12. Agoston, Max K. *Computer graphics and geometric modeling*. Springer, 2005.