Haptic Guidance with Fuzzy Control in Simulation-Based Surgical Training

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

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

ABSTRACT
This paper describes a haptic guidance system developed to assist in model-based training of minimally invasive, laparoscopic procedures. The key factor motivating the development of the device called CAST (Computer-Assisted Surgical Trainer) is the need to advance the state-of-the-art in teaching laparoscopy, and ultimately to improve surgical outcomes. In the paper, CAST's exchangeable task scenario and fuzzy logic controller with reference generator are presented. The system, while currently intended for off-line, laboratory use, has an excellent potential for real-time, assistive functions in the operating room.

Author Keywords
Surgical Training, Laparoscopy, Haptic Guidance, Fuzzy Logic Controller, Medical Simulation

INTRODUCTION
Our application domain is laparoscopic surgery, a procedure which is very effective when performed by a well-trained surgeon. Its benefits are short recovery time, low operative blood loss, and lesser post-operative pain. However, challenges such as restricted vision, hand-eye coordination problems, limited working space, and lack of tactile sensation are difficult to overcome for novices. These issues make laparoscopic surgery a more difficult technique for medical students and residents to master [1].

Therefore, simulation-based tools for teaching laparoscopy are now becoming a *sine qua non* condition for effective training environments. Over the past several years, we have developed a computer aided trainer called CAST (Computer-Assisted Surgical Trainer). Its architecture is presented in detail in previous work [2][3][4][5][6]. Its overall system block diagram is shown in Figure 1. The two main components are hardware and software "blocks" that cooperate to generate force feedback and provide visual guidance to the user. The hardware consists of two fixtures with surgical instruments, web camera and *exchangeable cassette*. Each fixture is symmetrically identical and equipped with sensors and motors for haptic guidance and instrument tip position tracking. The web camera imitates an endoscopic camera and the *exchangeable cassettes* provide realistic training task scenarios. We will describe the cassette concept in the next section.

SpringSim 2015, April 12 - 15, 2015, Alexandria, VA, USA
MSM 2015 - Modeling and Simulation in Medicine

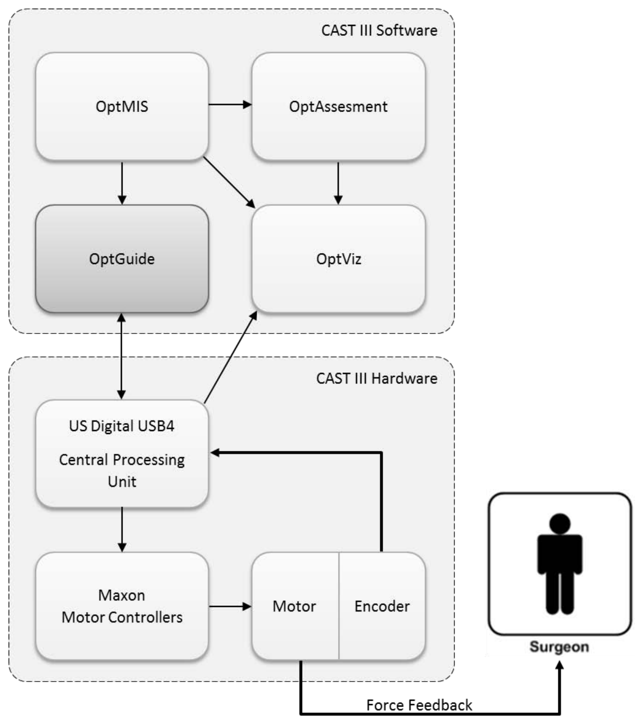

Figure 1. CAST Architecture

The software block consists of *optAssessment*, *optMIS*, *optVIZ*, and *optGuidance*. The *optAssessment* objectively quantifies competency in minimally invasive surgery (MIS) skills. In [2], we defined five relevant metrics for hand–eye coordination tasks which were validated by an experienced surgeon: time, movement economy ratio, movement direction profile, peak speed width and continuity of movement. The optimal motion planning module called *optMIS* [4][6] generates shortest, collision-free trajectories for laparoscopic instrument movements in the rigid block world used for hand-eye coordination tasks. The method consists of two sequential stages: 1) shortest path planning aimed at generating the path between start and target configurations of a laparoscopic instrument, and 2) time-optimal trajectory planning used to specify a time law on shortest paths in order to prevent collisions between instruments. Figure 2 represents a generated optimal path for a simple target contact task. The *optViz* [5] module is responsible for rendering the optimal instrument path over the camera image of the exercise space. "Live" data from the camera are merged with the virtual trajectory and instrument tip location. They are rendered on the display monitor as shown in Figure 3.

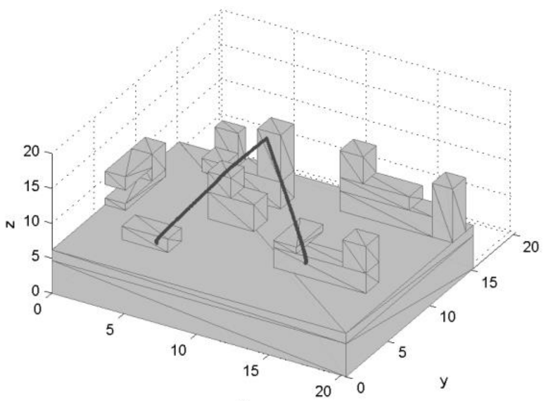

Figure 2. The optimal path generated by optMIS

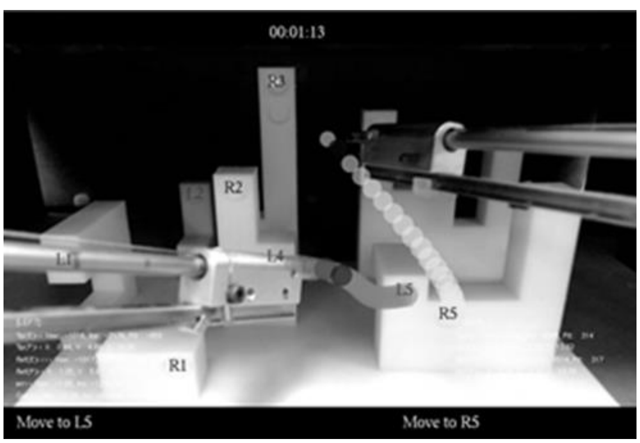

Figure 3. The screenshot of visual guidance system

The interaction between the software and the actual physical system is done through a guidance module (*optGuide*) and the US Digital USB4 central processing unit. We initially used a PD controller with simple reference generator to implement *optGuide*. To improve haptic guidance, we took into account friction, gravity, and other uncertainties and developed a prototype fuzzy logic controller (FLC) [5]. In this paper, we focus specifically on two new developments: 1) we propose an improved reference generator, and 2) we present work on designing an adaptive FLC.

EXCHANGEABLE TASK SCENARIO

A 3D printer produces 3D physical objects using a CAD model. One of benefits of 3D printing technology is rapid prototyping [7]. Also, the cost of printing devices and materials are getting cheaper. Due to these reasons, 3D printing is an emerging technology for medical applications such as surgical planning, implant and tissue designing, and medical education and training [8]. CAST uses 3D printing technology for generating training vignettes.

To become a laparoscopic surgeon, board certification requires completing the Fundamentals of Laparoscopic Surgery (FLS) program, offered through the Society of American Gastrointestinal and Endoscopic Surgeons, and the American College of Surgeons [9]. This program presents web-based content on non-procedure-specific laparoscopy basic skills, with multiple-choice tests and hands-on evaluations of skills, with proctors who are experienced surgeons. The hands-on exams of FLS require surgeons to evaluate performance on subjective criteria and time. The virtual reality simulators [10][11][12][13] are one class of training devices to prepare for hands-on exams. However, a 3D virtual reality environment on a 2D display monitor is insufficient for trainees to adapt well to the real operating theatre environment. Therefore, we propose a 3D printing-based test environment for quick, realistic and exchangeable scenario set-up.

The CAST system uses *exchangeable cassettes* and real surgical instruments to support a variety of practice scenarios. The cassettes support realistic test materials such as artificial tissue, 3D printed organ models, or simple practice setups to prepare for the FLS test. We have designed the foundation for various cassettes by considering workspaces of real surgical instruments. On top of this base, we attach realistic test materials. Figure 4 illustrates an example of a 3D printed heart model. We are developing more realistic practice procedures using this heart model. In this paper, we use a block world environment shown in Figure 3 to support the haptic guidance system.

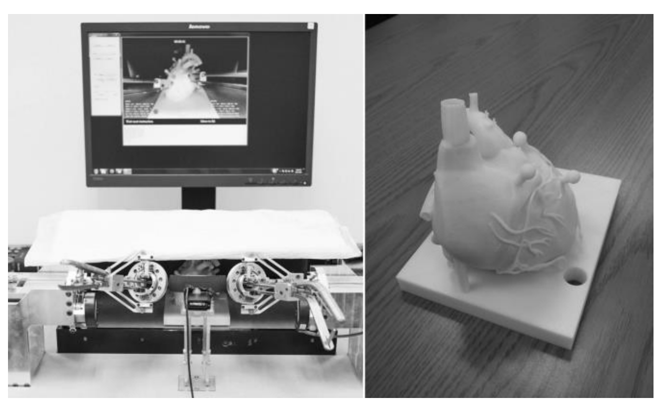

Figure 4. (a) Overall system, (b) 3D printed heart model

HAPTIC GUIDANCE

The *optGuide* module is designed as a skill transfer system for medical students to master laparoscopic surgical skills. The main role of *optGuide* is to provide haptic guidance to a novice trainee [5]. Whenever the instrument tip deviates from the optimal trajectory generated by *optMIS*, *optGuide* applies force to guide the user in the proper direction. It uses the optimal path and the actual tip position to update the reference position. The reference position is offered by the reference generator module. Also, the actual tip position is captured by the encoder unit. Both the reference position and the actual tip position are updated every sampling period ($\Delta T = 50ms$). We propose a curvature based reference generator and Fuzzy Logic Controller in this section.

Reference Generator

Haptic interface has been widely used for skill transfer system. Solis [14] proposed a reactive robot system to teach Japanese handwriting for a person who is unfamiliar with Japanese. Also, Teo [15] developed a robotic teacher to guide students how to write Chinese characters by using haptic and visual guidance system. Similarly, CAST has a haptic interface to transfer laparoscopic surgical skills and to prohibit wrong maneuvering. In this paper, we focus on developing haptic interface to support a simple target touch contact task related to hand-eye coordination skill. To provide reference input for haptic interface, the reference generator is designed.

We define a virtual tip that represents reference position (p_{ref}) and moves along the optimal path with speed ($|v|$). The virtual tip speed is determined by the speed profile or curvature of the optimal path generated by *optMIS*. For instance, the tip speed will increase based on the speed profile if the virtual tip moves along the straight path. However, if the virtual tip moves along a curvy path, the tip speed will decrease according to curvature --- very much like in driving a car. The speed profile is defined by acceleration and maximum speed. Also, the virtual tip speed becomes zero near a target, in order to take into account real surgery scenario. For example, a surgeon may perform a specific task such as cutting a tissue after moving a surgical instrument.

The reference generator requires the following information to update virtual tip position:

- Optimal path: a set of three-dimensional discrete points,
  $\mathbf{P}_{optimal} = \{p_1, p_2, \cdots, p_n\}$, where $p_i \in R^3, i = 1, \cdots, n$
- Virtual tip position ($p_{virtual}$): one of the elements in $P_{optimal}$, *i.e.*, $p_{virtual} \in \mathbf{P}_{optimal}$.
- Actual instrument tip position (p_{actual}) in three dimensional workspace.
- Braking distance (d_{brake})
- Distance error between virtual tip and actual tip, $d_{virtual}^{actual} = |p_{virtual} - p_{actual}|$
- Distance between virtual tip and target object, $d_{virtual}^{target} = |p_{virtual} - p_{target}|$

Algorithm 1 shows how to update virtual tip position. If $d_{virtual}^{actual}$ is greater than distance error threshold (δ_{error}), feedback force is applied to reduce distance error. If $d_{virtual}^{target}$ is less then d_{brake}, virtual tip speed is decreased based on the speed profile. Parameters, acceleration and maximum speed for speed profile are varied by the predefined skill level. Otherwise, curvature value (κ) is calculated to determine the virtual tip speed. For this calculation, we use a curvature approximation method presented in [16]. Once the tip speed is updated, the virtual tip position ($p_{virtual}$) is also updated by following steps: 1) Calculate direction vector ($\overrightarrow{dir}$), 2) Get estimated reference point ($\hat{p}_{virtual}$), 3) Find nearest point in $P_{optimal}$ from $\hat{p}_{virtual}$ because $\hat{p}_{virtual}$ may not be on the optimal path, 4) Update $p_{virtual}$ to apply guidance force.

Algorithm 1. Update virtual tip position

IF $d_{virtual}^{actual} > \delta_{error}$
  apply feedback force to reduce error
Else IF $d_{virtual}^{target} < d_{brake}$
  decrease speed based on speed profile
Else
  get neighbor points, p_{i-1}, p_{i+1}, where $p_i = p_{virtual}$
  calculate κ
  IF $\kappa < \kappa_{\min}$ **, then** use speed profile // straight path
  Else IF $\kappa > \kappa_{\max}$ **, then** $v = v_{\min}$
  Else
    $\overrightarrow{dir} = p_{i+1} - p_i / |p_{i+1} - p_i|, |v| = \frac{v_{\min} - v_{\max}}{\kappa_{\max} - \kappa_{\min}} \cdot (\kappa - \kappa_{\min}) + v_{\max}$
    $\hat{p}_{virtual} = p_{virtual} + \overrightarrow{dir} \cdot |v| \cdot \Delta T$
    find nearest point in $P_{optimal}$ from $\hat{p}_{virtual}$, and update $p_{virtual}$

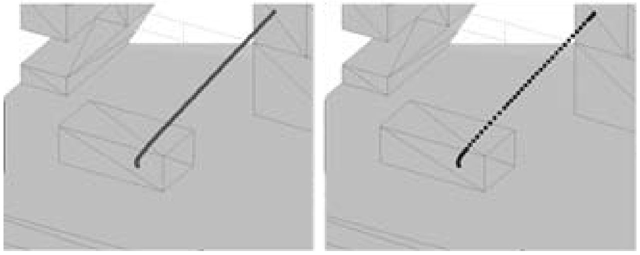

Figure 5. (a) optimal path from R1 to R2, (b) the trajectory of virtual tip

Consider the training scenario in which multiple targets must be touched using two instruments, as shown in Figure 3. There are four targets labeled by R1, R2, R3, and R5 for right instrument. Generated path from R1 to R2 is shown in Figure 5-(a). This path consists of 105 discrete 3D points. Figure 5-(b) illustrates the trajectory of the virtual tip. Due to the speed profile and curvature, the reference points reflect the shape of the path and the trajectory which consists of 56 points. Figure 6 shows the simulated result. We use $\kappa_{\max} = 1$, $\kappa_{\min} = 0$, $v_{\min} = 1$, and $v_{\max} = 4$ for simulation.

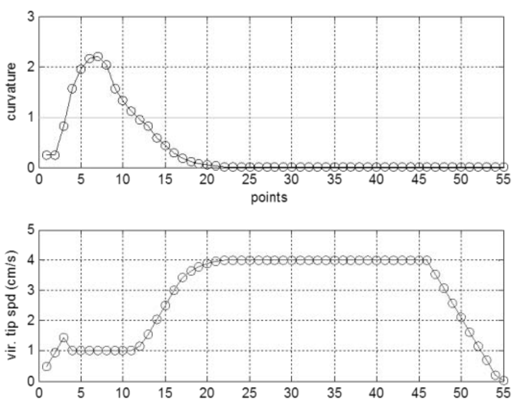

Figure 6. Curvature and virtual speed on the path from R1 to R2

Fuzzy Logic Controller

The Fuzzy Logic Controller (FLC) concept has been widely used for nonlinear systems. FLC can be easily implemented by human experts based on empirical knowledge without complex mathematical models [17]. CAST system is a nonlinear system and specifically involves a human in the control loop. This human factor is considered as a *disturbance* input so that it is unmeasurable. Therefore, it is appropriate to use the Adaptive Fuzzy Logic Controller for *optGuide*. Figure 7 illustrates the overall block diagram of *optGuide*. In [5], we proposed prototype FLC controller. Based on this prototype, we are developing an improved controller by adding more rules and applying adaptive laws.

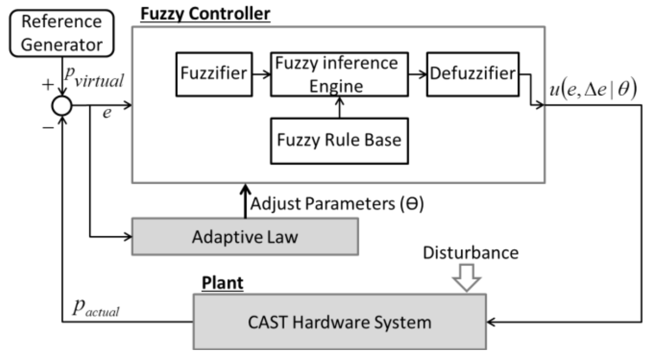

Figure 7. The block diagram of *optGuide*

The fuzzy logic system consists of four parts – *fuzzifier*, *fuzzy rule base*, *fuzzy inference engine* and *defuzzifier* [17]. The *fuzzifier* maps from real values to fuzzy sets. The *fuzzy rule base* consists of IF-THEN statements to describe the control algorithm derived by human experts based on empirical knowledge. The *fuzzy inference engine* combines IF-THEN rules to map from the input fuzzy sets to the output fuzzy sets. The *deffuzifier* produces a non-fuzzy control output value from the output fuzzy sets using a fuzzy inference engine. On top of this configuration, an adaptive law is added to adjust parameters that represent the fuzzy logic system. The expected performance provides proper force feedback by adjusting control parameters even though there is large disturbance caused by a human. The design process for *optGuide* is presented below.

The position error (e(k)) between the virtual and actual tip positions and derivative of the position error (Δe(k)) are used for antecedent variables. The set value for the motor controller ($\bar{y}^l(k)$) is assigned for a consequent variable. Five fuzzy sets, NB, N, ZO, P, PB, with Gaussian membership functions defined by mean and standard deviation are used for each antecedent variable (where NB, N, ZO, P, and PB represent negative big, negative, zero, positive, and positive big, respectively). There are 7 singleton membership functions, NB, N, NS, ZO, PS, P, PB, for consequent variable. (NS and PS represent negative small and positive small, respectively.) Figure 8 depicts these functions.

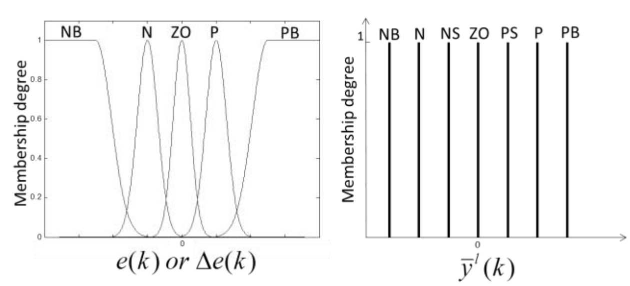

Figure 8. (a) Membership functions for e(k) or Δe(k), (b) singleton membership functions for $\bar{y}^l(k)$

The IF-THEN fuzzy rules are shown in the following form:

$$R^l : IF\ e(k)\ is\ F_1^l\ and\ \ \Delta e(k)\ is\ F_2^l,\ then\ \bar{y}^l(\mathrm{k})\ is\ G^l$$

where F_1^l, F_2^l, G^l are fuzzy sets of $e(k)$, $\Delta e(k)$, $\bar{y}^l(k)$, respectively. l represents the rule number. Since there are five fuzzy sets for each antecedent variable, the number of rules is $25 = 5^2$. The initial rule table is shown in Table 1. For example, if the position error is positive big (PB) and derivative of error is negative (N), then set value is positive small (PS). In this case, the actual tip moves toward the reference position because the error is decreasing even though the error value is still big. Thus, it is better to apply relatively small control input to reduce the position error.

$\Delta e(k)$ \ $e(k)$	NB	N	ZO	P	PB
NB	NB	NB	N	PS	PS
N	NB	N	NS	PS	PS
ZO	N	N	ZO	NS	P
P	NS	NS	PS	P	PB
PB	NS	NS	P	PB	PB

Table 1. Initial Fuzzy Rule table

The proposed FLC uses a singleton fuzzifier, product inference engine and the center average defuzzifier. Therefore, the actual control output ($u(k)$) is obtained from the following equation:

$$u(k) = \frac{\sum_{m=1}^{5}\sum_{n=1}^{5} \bar{y}^{(m,n)}(k) \cdot \mu_{e(k)}^{m} \cdot \mu_{\Delta e(k)}^{n}}{\sum_{m=1}^{5}\sum_{n=1}^{5} \mu_{e(k)}^{m} \cdot \mu_{\Delta e(k)}^{n}}$$

The consequent variable, $\bar{y}^l(k)$, is changed by the adaptive law. We are designing the adaptive law for direct adaptive fuzzy controller [18] to make the system robust under large disturbance.

Each fixture has three motors to control yaw, pitch, and insertion axis. Each axis has different physical characteristics. Therefore, we use three independent FLCs with different parameters to make the system stable and to consider the different characteristics of each axis. Also, the initial parameters are determined by experimental results. A CAST demonstration video is available on the web page: (http://mbdl.arizona.edu/projects/computer.assisted.minimally.invasive.surgery.html)

CONCLUSION

In this paper, we have presented a new fuzzy logic based controller design that enables force guidance in a laparoscopy training system. The key contribution is the ability to adapt haptic assistance to relatively unpredictable human operator behaviors. (Whereas more experienced trainees are relatively steady in how they operate the instruments, novices have a tendency to produce large scale disturbance inputs.)

We are developing more realistic training scenarios by using a 3D heart model and designing adaptive law for haptic guidance system. Also we are preparing a human subject study to evaluate the benefits of CAST system. Our preliminary simulation results are promising and indicate that the system will be relatively robust.

REFERENCES

1. J.W. Rozenblit, “Models and Techniques for Computer Aided Surgical Training,” in Lecture Notes in Computer Science, No. 6928, Part 2, pp. 233-241, Springer-Verlag, 2012.

2. Riojas, Mario, et al. "Knowledge elicitation for performance assessment in a computerized surgical training system." Applied Soft Computing 11.4 (2011): 3697-3708.

3. Feng, Chuan, et al. "Defining spatial regions in computer-assisted laparoscopic surgical training." Engineering of Computer Based Systems, 2009. ECBS 2009. 16th Annual IEEE International Conference and Workshop on the. IEEE, 2009: 176 - 183.

4. Napalkova, Liana, et al. "An Optimal Motion Planning Method for Computer-Assisted Surgical Training." Applied Soft Computing, vol 24, 2014: 889-899.

5. 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.

6. Rao, Aakarsh, et al. "Performance assessment and optimization of motion planning in a surgical trainer for potential space applications." Aerospace Conference, 2014 IEEE. IEEE, 2014: 1 - 12.

7. Berman, Barry. "3-D printing: The new industrial revolution." Business horizons 55, no. 2 (2012): 155-162.

8. Rengier, Fabian, et al. "3D printing based on imaging data: review of medical applications." International journal of computer assisted radiology and surgery 5, no. 4 (2010): 335-341.

9. Fundamentals of Laparoscopic Surgery, http://www.flsprogram.org

10. Dankelman, Jenny. "Surgical robots and other training tools in minimally invasive surgery." Systems, Man and Cybernetics, 2004 IEEE International Conference on. Vol. 3. IEEE, 2004: 2459 - 2464.

11. Simbionix Co. 2007. http://www.simbionix.com

12. Stylopoulos, N., et al. "Computer-enhanced laparoscopic training system (CELTS): bridging the gap." Surgical

Endoscopy and Other Interventional Techniques 18.5 (2004): 782-789.

13. Waxberg, S. L., et al. "Evaluation of physical versus virtual surgical training simulators." Proceedings of the Human Factors and Ergonomics Society Annual Meeting. Vol. 48. No. 15. SAGE Publications, 2004: 1675-1679.

14. Solis, Jorge, et al. "Reactive robot system using a haptic interface: an active interaction to transfer skills from the robot to unskilled persons." Advanced Robotics 21.3-4 (2007): 267-291.

15. Teo, Chee Leong, Etienne Burdet, and H. P. Lim. "A robotic teacher of Chinese handwriting." Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2002. HAPTICS 2002. Proceedings. 10th Symposium on. IEEE, 2002: 335 - 341.

16. Zhou, Shilin, Jianping Yin, Xiaolin Yang, and Junyun Wu. "Accurate Curvature Approximation of 3-Dimension Discrete Points." In Communication Systems and Information Technology, pp. 803-810. Springer Berlin Heidelberg, 2011.

17. Wang, Li-Xin. A course in fuzzy systems. Prentice-Hall press, USA, 1999.

18. Wang, Li-Xin. Adaptive fuzzy systems and control: design and stability analysis. Prentice-Hall, Inc., 1994.