

# Augmented Reality Visualization for Computer Assisted Surgical Training

Akash Shankaran and Jerzy W. Rozenblit

Department of Electrical and Computer Engineering,  
University of Arizona, Tucson, Arizona 85721, USA  
{akashs,jr}@email.arizona.edu

**Abstract.** *Laparoscopic surgical training systems limit the users' depth perception and hand-eye coordination due to the use of a 2D camera image. In this paper, we present an approach to enhance visualization using augmented reality which aids in hand-eye coordination and depth perception in the operating environment. An optimal path is calculated in a 3D space and projected onto a 2D camera image. The mapping from the 3D to 2D coordinate system is calculated using least squares camera calibration method. Instrument tip location in space is calculated using feedback from encoders attached to them. We combine the information about the instrument tip and optimal path to achieve robust visualization.*

**Keywords:** Laparoscopic surgical training, augmented reality, visualization, minimally invasive surgery

## 1 Introduction

Augmented Reality (AR) attempts to bridge the gap between reality and the virtual world. AR systems merge computer graphics and real imagery into a single, coherent perception of an enhanced world [9]. Visualization technologies range from real environment depictions, augmented reality, augmented virtuality, and complete virtual environment [7, 8].

Given recent advances in technology, such systems are being used more often to provide training. This information can be overlaid in the form of text, images, markers and videos, etc.; this helps to enhance user experience. There are many applications (apps) developed for smart phones which use this technology in areas such as ARBasketball [12], Layar mobile browser [13], and GE smart grid [20].

A promising and yet not fully tapped use of this technology is in the field of medical simulation, specifically in laparoscopic surgical training. The development of surgical training systems for minimal invasive surgery is geared towards medical students, residents, surgeons and medical personnel who need to perform this procedure efficiently [3]. The major challenges faced by a surgeon during laparoscopic surgery include hand eye coordination, depth perception, and limited haptic feedback [10]. Augmented reality has the potential to improve surgeon's

performance by reducing these aforementioned challenges, and thus reducing the amount of risk to patients.

Most of the present surgical trainers depend on the knowledge of an expert surgeon, who would analyze the trainee and provide feedback on his or her performance. Ideally, the surgeon would be standing behind the trainees while assessing a practice session. This has certain drawbacks; clearly, expert surgeons are not able to dedicate significant amount of time to such training sessions. Besides, data collection, statistical analysis, and outcome metrics can be efficiently generated using appropriate software and hardware tools.

Over the last several years, various surgical trainers that use virtual reality have been developed. To some extent, they eliminate the need for an expert surgeon's physical presence during training. Some of these trainers such as the LAP Mentor by Symbionix, EDGE by Simulab and CELTS are commercially available virtual reality (VR) systems with efficient feedback mechanisms [5, 17, 6]. Some others such as the laparoscopic training box provided by SAGES and Ethicon are quite simplistic with no objective feedback modules [18, 19]. There is a gap between the high end VR simulators and the naïve, simplistic training boxes. Little attempt has been made to bridge this gap and design systems that are relatively low cost, yet provide a suite of essential training features. In the work presented here, we propose an augmented reality approach which fits in the niche between the high end VR simulators and the traditional box trainers. This enables us to provide many cues which alleviate hand-eye coordination and depth perception problems during training exercises.

This paper is focused on developing the visualization system for the Computer Assisted Surgical Trainer [3, 4, 11] developed at the University of Arizona. We describe our approach using AR techniques. The visualization system is intended to provide robust visual guidance with enhanced depth perception in laparoscopy training.

## 2 System Overview

The computer assisted surgical trainer (CAST) in its current version III comprises five building blocks. A block diagram is shown in Figure 1. The OptMIS module uses Delaunay triangulation to generate an optimal path in space [4]. Opt-Assessment consists of an assessment algorithm based on fuzzy logic and uses parameters such as time, movement economy ratio, movement direction profile, peak speed width, and continuity of movement to assess trainees' performance [3]. OptGuide provides haptic guidance using a PD control algorithm by guiding the user along the optimal instrument approach path [11]. The hardware comprises of a pair of instrument fixtures, servo motors and controllers. The OptARViz module, which is the main focus of this paper, is responsible for visualization and AR aspects of the system. The following sections provide its detailed description.

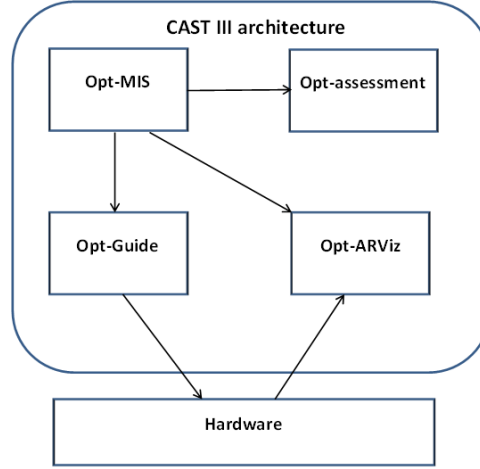


Fig. 1. System Block Diagram

## 2.1 Visualization System Overview

The main function of opt-ARViz is to visualize optimal instrument paths using data generated by optMIS [4]. We use real-time camera stream data and overlay it with the optimal path using computer graphics. The main *actors* involved in optARViz are the trainee, the instrument tip and the surgeon's assistant. The trainee moves the instrument in the block world (a training vignette), whereas the assistant may reposition the camera to provide a better field of view. The instrument is moved from target to target in the block world, along the optimal path.

OptARViz can be broken down into two major steps:

1. Camera Calibration
2. Visualizing and overlay of the Optimal Path.

We describe each steps in detail in Section 3 and Section 4.

## 3 Camera Model and Calibration

### 3.1 Camera Model

We use Logitech C920 HD Pro Webcam in CAST III. This camera comes with a full HD sensor with the resolution of 1920x1080 pixels and a Carl Zeiss autofocus lens. The camera can capture and compress videos to H.264, which reduces the video processing workload on the computer. It has a frame rate of 30fps, in the 1080p mode. It also provides an excellent image quality in low light settings and the function to enhance brightness using software.

### 3.2 Camera Calibration

Camera calibration is the process of finding the intrinsic and extrinsic parameters of a camera which enable us to obtain the camera matrix (1). The extrinsic parameters include the position and orientation of the camera in the  $X, Y$  and  $Z$  dimensions. The intrinsic parameters are more specific to the camera and include focal length, aspect ratio, principal point and the angle between axes of image plane [1]. The camera matrix is composed of the intrinsic and extrinsic parameters. The camera matrix denoted by  $\mathcal{M}$ , is the transform used to find where a point  $(x, y, z)$  in the real world will appear on the image. A general notation for the camera matrix in the form of its parameters is given in (1). Prior to the calibration process, we fix the camera at a known location and save the coordinate locations of the points of interest representing edges.

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \underbrace{\begin{pmatrix} \text{Transformation} \\ \text{representing} \\ \text{intrinsic parameters} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \text{Transformation} \\ \text{representing} \\ \text{extrinsic parameters} \end{pmatrix}}_{\text{Camera Matrix, } \mathcal{M}} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad (1)$$

We solve for the camera matrix using least squares. The first step in this transformation is to represent the points in the form of homogeneous coordinates [2]. According to the camera calibration method described in [1], we calculate the camera matrix, which is the transform between the real world and the image plane. To solve for the camera matrix, we solve the least squares solution of  $x$ , where

$$Ux = 0 \text{ subject to } |x| = 1. \quad (2)$$

where  $U$  has more rows than the variables we are solving for, which is the total number of intrinsic and extrinsic parameters. Hence, we minimize  $\|Ux\|$  subject to  $|x| = 1$ . The vector  $U$  is the set of points in the image corresponding to selected real world coordinates. The minimum is reached when  $x$  is set to the eigen vector corresponding to the minimum eigen value of  $U^T U$ . We are interested in finding the vector  $x$ , which is the camera matrix. The calibration process of finding  $x$  is analogous to a curve fitting problem, where we should not select too many points to overfit the curve or too few points to get a poor fitting curve. The CASTIII block world calibration was performed with 17 points which takes into account points at different depths in the real world. The camera matrix is a 3x4 matrix, which for our model was obtained as shown below:

$$\begin{bmatrix} -7.7658 \times 10^{-2} & 3.6395 \times 10^{-3} & -1.7204 \times 10^{-2} & 3.8046 \times 10^{-3} \\ -9.2778 \times 10^{-3} & 6.7931 \times 10^{-2} & -1.9927 \times 10^{-2} & -9.9426 \times 10^{-1} \\ -1.6618 \times 10^{-5} & 4.0317 \times 10^{-6} & -4.1977 \times 10^{-5} & -5.7953 \times 10^{-4} \end{bmatrix}$$

The squared error is  $9.7397 \times 10^{-6}$  which shows that the camera calibration process is very accurate in mapping the 3D points to the 2D points on the image.

Using the camera matrix calculated above, we are able to project the optimal path generated by optMIS. The projected path coordinates are calculated in homogeneous coordinates according to (3).

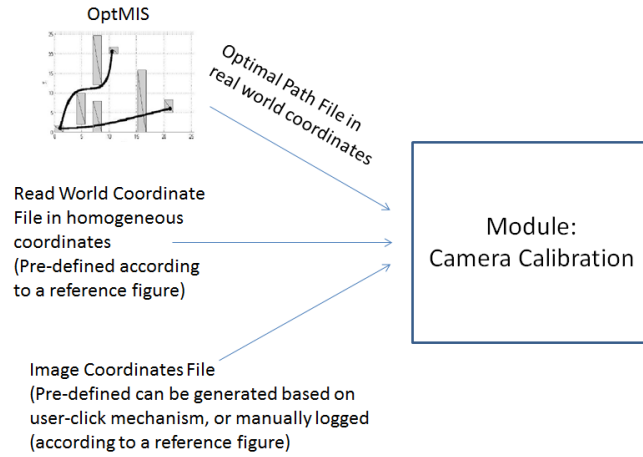
$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \mathcal{M} * \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}. \quad (3)$$

We can obtain the exact location of where each projected point in the homogeneous coordinate will appear on the image by dividing the  $u$  and  $v$  with their respective  $w$  component as shown in (4, 5). We project  $(U_i, V_i)$  on the visual display which represents the image location of the real world coordinate from the vector  $U$ .

$$U_i = u_i / w_i. \quad (4)$$

$$V_i = v_i / w_i. \quad (5)$$

Camera calibration is specific to the given training task. Hence, we have to pre-select certain points of interest in the real world as input along with their corresponding image location for our camera calibration module. We also provide the optimal path file with the real world coordinates. All the necessary inputs required to perform camera calibration procedure for CAST III are shown in Figure 2.



**Fig. 2.** Camera Calibration Overview in CAST

## 4 Visualization and Optimal Path Overlay

Two visualization feedback mechanisms have been incorporated in the AR overlay:

1. Objects further away from a camera in a scene appear relatively smaller in size than the objects closer to the camera.
2. Different colors are used to visualize the instrument tip when it is on the optimal path, a target point, or when it is anywhere else in the block world.

These visual cues are meant to assist the trainee in better hand eye coordination and depth perception.

Our visualization framework comprises a central renderer with additional supporting libraries. The central renderer is based on the Visualization Toolkit (VTK) library [15] which is a graphics visualization library written in C++. The camera input is captured using OpenCV [14] which is a supporting library. We convert the openCV image from the openCV Ipl format to the VTK image actor so that it is compatible with the VTK framework. Since a video stream is made up of a series of images, we use a timer callback method which captures frames every 50ms. The other set of inputs to the VTK visualization are the encoder values provided by CAST III USB4 encoders [16] and the optimal path file generated by camera calibration discussed in Section 3.2.

A screenshot of our test visualization with depth perception is shown in Figure 3. As most of the CAST III modules have been implemented in C++, we plan to port optARViz to C++ which will ease the system integration due to platform consistency. Hence, we are implementing a framework shown in Figure 4, which makes use of open source computer vision libraries (openCV)[14], visualization libraries (VTK)[15] and the USB4 encoder libraries.

## 5 Current and Future Research

Upon completion of the implementation in C++, the optARViz will be integrated with the optimal guidance control algorithm [11]. Integration with the control algorithm will involve obtaining encoder values that record the position of a surgical instrument in four degrees of freedom. Members of the research team are planning to conduct a small experiment to verify the usability of the system. This will involve a comparison in user performance between the augmented reality view with and without controls, compared to the traditional CAD model view. Based on the results, we will be in a position to justify if the trainee's depth perception and hand-eye coordination will have improved with the new visualization.

Further research direction will possibly include the use of a tracking algorithm, which will record augmented markers in the training space and dynamically update the optimal path visualization as the camera moves. Since this process is computationally very intense, perhaps an FPGA implementation could be used to provide faster processing. This approach would allow us to move the



Fig. 3. *OptARViz* screenshot from Matlab

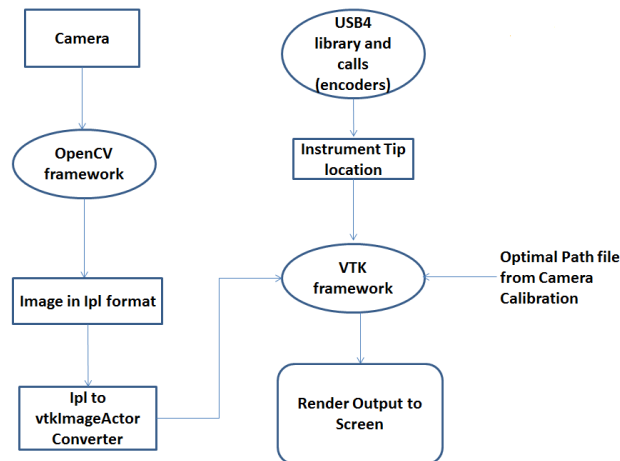


Fig. 4. *Framework design for optARViz*

camera freely in space and not restrict our system to the surgical training domain. There can be numerous uses of the Augmented reality technology in areas such flight simulators and entertainment technologies, car navigation, etc.

## References

1. Forsyth, D.A. and Ponce, J.,: Computer Vision: A Modern Approach, 2nd edition (2011)
2. Akenine-Möller, T., Haines, E., Hoffman, N.: Real-Time Rendering, 3rd Edition (2008)

3. Riojas, M., Feng, C., Hamilton, A.J., Rozenblit, J.W.: Knowledge elicitation for performance assessment in a computerized surgical training system. In: *Applied Soft Computing* 11(4), pp. 3697-3708 (2011)
4. Napalkova, L., Rozenblit, J.W., Hamilton, A., Hwang, G., Suantak, L.: An Optimal Motion Planning Method for Computer-Assisted Surgical Training. In: *Applied Soft Computing* (under review).
5. Martin W. von Websky, MD, Martina Vitz, MD, Dimitri A. Raptis, MD, R. Rosenthal, MD, P.A. Clavien, MD, Dieter Hahnloser, MD.: Basic Laparoscopic Training Using the Simbionix LAP Mentor: Setting the Standards in the Novice Group. In: *Journal of Surgical Education*, Volume 69, Issue 4, July–August 2012, Pages 459–467 (2012)
6. Stylopoulos, N., Cotin, S., Maithel, S. K., Ottensmeyer, M., Jackson, P. G., Bardsley, R. S., Dawson, S. L.: Computer-enhanced laparoscopic training system (CELTS): Bridging the gap. *Surgical Endoscopy*, 18(5), 782-9. doi:<http://dx.doi.org/10.1007/s00464-003-8932-0> (2004)
7. Hamza-Lup Felix G.: A Distributed Augmented Reality System for Medical Training and Simulation (2004)
8. Radermacher, K., Pichler, C.V., Fischer, S., Rau, G.: 3D-Visualisation in Surgery, Helmholtz-Institute Aachen (1998)
9. Fuchs, H., Livingston, M.A., Raskar, R., State, A., Crawford, J.R., Rademacher, P., Drake, S.H., Meyer, A.A: Augmented Reality visualization for laparoscopic surgery. In: *First International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 934–943. Springer-Verlag (1998)
10. Xin, H., Zelek, J.S., Carnahan, H.: Laparoscopic surgery, perceptual limitations and force: A review. In: *First Canadian Student Conference on Biomedical Computing*
11. Jan Nikodem, Hwang. G., Rozenblit, J.W., Napalkova, L., and Czapiewski, P.: Technical Real-ization of the Optimal Motion Planning Method for Minimally Invasive Surgery, In: *IEEE 19th International Conference and Workshops on Engineering of Computer Based Systems* (2012)
12. Augmented Reality Basketball App, <http://simiotica.com/app-1.html>
13. Layar Mobile Browser <http://www.layar.com>
14. Open Computer Vision Library (opencv), <http://opencv.willowgarage.com/wiki/>
15. Visualization Toolkit Library (VTK), <http://www.vtk.org>
16. USB4 Encoders, <http://www.usdigital.com/products/interfaces/pc/usb/usb4>
17. EDGE Virtual Reality Trainer, <http://www.simulab.com/product/surgery/laparoscopic/edge>
18. SAGES Training Box, <http://www.flsprogram.org/testing-information/trainer-box/>
19. Ethicon Training Box, <http://www.ethicon.com/>
20. GE Smart Grid, [http://ge.ecomagination.com/smartgrid/#/landing\\_page](http://ge.ecomagination.com/smartgrid/#/landing_page)