# 3D Position Tracking of Instruments in Laparoscopic Surgery Training

Shih-Fan Yang, Ming-Feng Shiu, Bo-Kai Shiu, Yuan-Hsiang Lin
National Taiwan University of Science and Technology
D10002104@mail.ntust.edu.tw

*Abstract — The VR laparoscope surgery training device is for junior hospital students to understand and train the laparoscope surgery. However, those developing training devices use special instruments for their training program, not real instruments. Therefore, this paper developed a surgery instruments' 3D location system for a VR laparoscope surgery training device while using real instruments. Moreover, this paper introduces a new method for hardware design in FPGA. This new method is using MATLAB's tools to develop the algorithm, generating the HDL and verifying the FPGA system. The system in this paper has less 1% error in the central operating area and could calculate two instrument positions every 200ms on the Altera's DE2-115 development platform. Thus, it could be employed in the VR training program.*

*Keywords — Laparoscope surgery, Laparoscope surgery training, VR Laparoscope surgery, FPGA, 3D locating, HDL coder, FPGA In the Loop, FIL, MATLAB, Simulink*

## I. INTRODUCTION

The minimally invasive surgery (MIS) is the popular issue in the surgery research, especially laparoscopic surgery. The laparoscopic surgery has lots of benefit for a patient such as reduced pain, smaller incisions and hemorrhaging, and the best advantage - shorter recovery time. However, the laparoscopic surgery is more difficult than traditional open surgery as surgeons cannot directly observe the surgery process directly with their eyes and can only observe the information though the TV monitor captured via an endoscope. Moreover, surgeons need to operate laparoscopic instruments, which are longer and harder for operating compared to traditional instruments. As a result, training a laparoscopic surgeon is very important and there is a need for a suitable training device, too.

In order to addressing the above approach, several researches developed different laparoscopic surgery simulation training systems to reduce the laparoscopic skill's learning time and curve [1]-[4]. The visual reality (VR) simulator is a developing area which provides visual laparoscopic surgery environment and could offer closer training models than traditional box trainer methods [5]-[8]. However, those researches used similar instruments in their training systems but not real instruments. Therefore, this paper introduces a 3D position location method for real instruments in the laparoscopic surgery training system.

This research uses the FPGA Cyclone IV DE2-115 platform which serves as the developing kernel for real-time calculation and cooperation with MathWorks' MATLAB and Simulink[9] for reducing the developing time. Simulink in this system captures two 2D images from the two USB cameras and sends two images to the DE2-115 for calculating the 3D positions of instruments, where

Simulink retrieves position data from the FPGA. After that, Simulink transmits the data to our visual reality laparoscopic surgery training system.

## II. BACKGROUND RESEARCH

### A. HDL coder & FPGA In the Loop

MATLAB is a widely known software algorithm development tool and can be used in many different areas. Especially, in 2012, the newest MATLAB released two new functions which are HDL coder [10] and FPGA In the Loop (FIL) [11]. HDL coder is a kind of code generator which can convert a Simulink design to portable, synthesizable VerilogHDL and be used in the FPGA design. FIL is a verification tool which can verify the implemented HDL code on FPGA boards. Figure 1 shows the loop of these two functions. The algorithm is designed in Simulink then converted to HDL code. In the end, this generated HDL code is verified in the FPGA in the loop and on hardware platforms, such as the DE2-115.

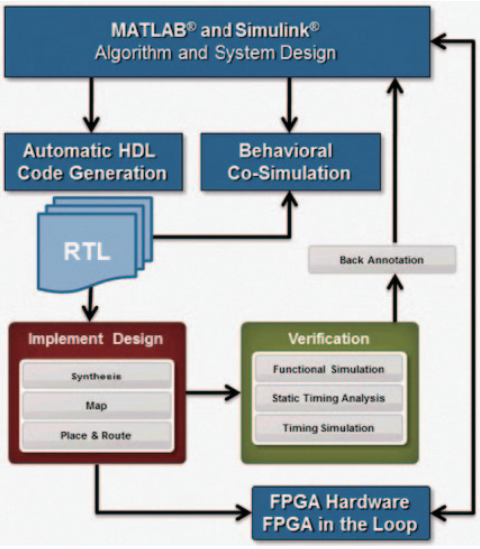


*Figure 1. The loop of the hardware algorithm design in the MATLAB [12]*

### B. 2D to 3D coordinate calculating

This research is based on using two cameras to find the location of instruments which is the technical of 2D to 3D converting. According to stereo imaging theory geometry [13], the 3D point's coordinate could be calculated from two 2D points by formula (1)-(3). Figure 2 displays the simplified stereo imaging system to indicate this principle. The two cameras are similar to human's eyes, having about 6cm distance. The object at (X,Y,Z) point will be captured by this two CCDs at (X1,Y1) and (X2,Y2). After getting position of the two points, the 3D point of the object can be derived.

$$x = b\frac{\frac{X_1 + X_2}{2}}{(X_1 - X_2)} \quad (1)$$

$$y = b\frac{(Y_1 + Y_2)\cdot\frac{f_x}{2}}{(X_1 - X_2)\cdot f_y} \quad (2)$$

$$z = b\frac{f_x}{(X_1 - X_2)} \quad (3)$$

Where the fx, fy are focal length and can be found by using MATLAB's Calibration Toolbox[14].

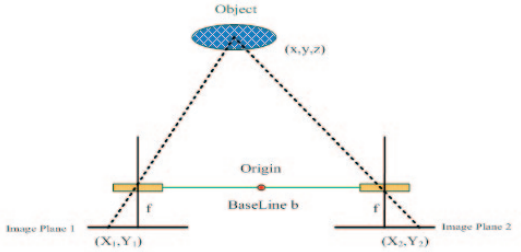


*Figure 2. A simplified stereo imaging system*

## III. SYSTEM ARCHITECTURE

The system architecture is shown in Figure 3. The instruments' photos are taken by the USB cameras and then collected in MATLAB. The MATLAB integrates the FIL interface to transmit data to the FPGA platform, DE2-115 for calculating. In the FPGA, there are image pre-processing parts and coordinate calculating

parts. The pre-processing part includes the color converter, Marker finder and morphology. The coordinate calculating part not only calculates the 2D and 3D coordinates of instruments but also calibrates the coordinates to improve accuracy.

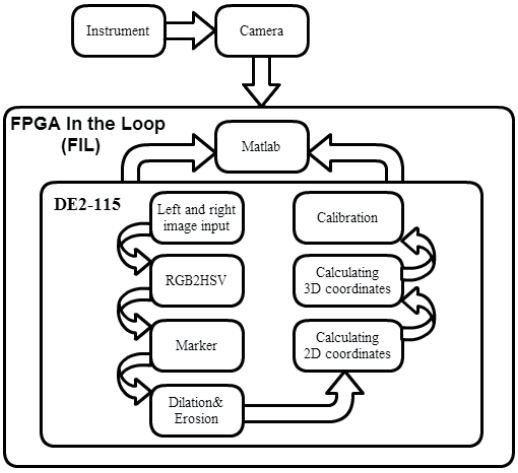


*Figure 3. The system architecture of this design*

### A. Pre-processing part

1) **Instruments & RGB2HSV:** In order to detect the instruments, the instruments have been labled a color sticker as shown in Figure 4. Therefore, the algorithm can identify instruments via different label colors such as red, yellow, green and blue. Moreover, HSV color model means the color is presented as hue, saturation and value, thus it is more suitable for color detection. However the source images are RGB color model therefore the RGB2HSV is used for converting the images from RGB color model to HSV color model.

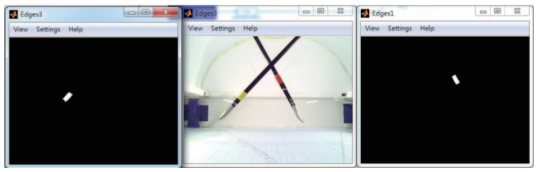

*Figure 4. Left: the yellow label's location; Middle: the instruments with label; Right: the red label's location*

2) **Marker finder:** After conversion to HSV, the red HSV(0,100,100), yellow HSV (60,100,100), green HSV(120,100,50) and blue HSV (240,100,100) are the basic color data and adopt the range of ±15 for the hue, the range of ±20 for the saturation and value to filtering the different colors in the image. Due to two source images and four colors label, this marker finder block will output eight Boolean data maps for next processing.

3) **Morphology:** The marked Boolean map image still has another problem that is noise.

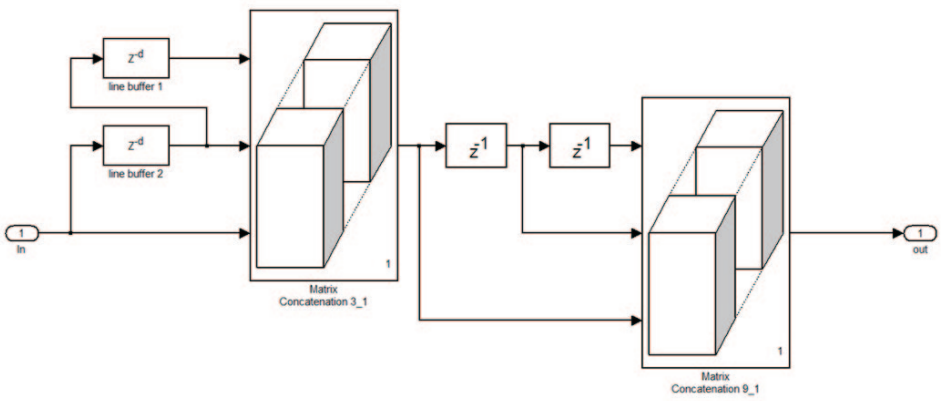


*Figure 5. The data restructured to the 3x3 matrix*

Although the photo is taken in a stable light source environment, it still can have glitches caused by the CCD sensor noise or other noise sources. Therefore, morphology can solve this problem. Dilation removes the unexpected points then the erosion linking and smoothing the labeled points for further stage.

The dilation and erosion in this design are a 3 by 3 mask but the data from previous stage is a pixel stream. Therefore, the data requires merging to 3x3 matrixes before the morphology. Figure 5 indicates how the pixel stream date be merged to matrix data type. First, the row direction data are delayed for combining to 3x1 arrays. Next, the 3x1 array data are delayed one array for merging 3x3 matrixes.

The Figure 6 demonstrates how the morphology block's function. The noise at upper right is been removed.

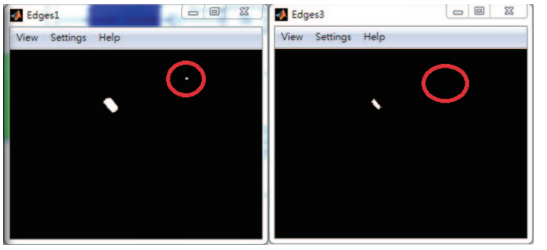


*Figure 6. The noise removing by the morphology block*

In additional, there are two instruments and four labels in one processing frame that means two 320*240 images and 3 bytes HSV data are compared with four colors mask and then removed noise at morphology block in one frame to find the label. In a software algorithm, the color conversion is pixel by pixel processing which means the two images take two conversion processing. Because there are four colors, the marker finding takes four times for four colors. After marker finding, there are eight Boolean maps needs to be process in the morphology stage. Therefore, these procedures spend much time for pure MATLAB calculation. However, because this design is implemented in a FPGA, the parallel processing could reduce the processing time. The two images are converted in the same time. Also, eight color filters and eight morphology blocks are for the eight Boolean maps' parallel processing. Therefore, the four labels' locating position are calculated out in the same and the processing speed is also increased.

## B. Coordinates calculating part

1) **2D coordinates Calculating:** After removing noise, the label's 2D coordinates could start to be located in the Boolean maps. Figure 7 shows how this system finds the label. The truth value is where the label at so this stage could search the truth value area and record the maximum X and Y point also the minimum X and Y point. Then, the central point's (X,Y) should be ((max(X)+min(X))/2, (max(Y)+min(Y))/2). This central point is defined as the label's 2D location in this project. Each label has two 2D locations, because each label has two images. Then, the label's 3D coordinate could be found from the two 2D coordinates.

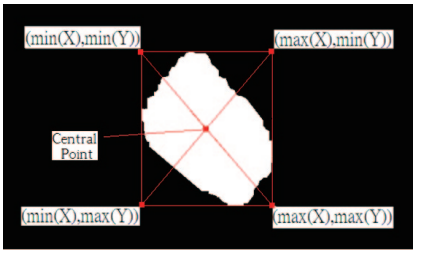


*Figure 7. How this system found the central point*

2) **3D coordinates Calculating:** This stage is for calculating the four label's 3D coordinate. Last stage has located the label's 2D central points in right and left images. After that, 2D coordinates are substituted in formula (1),(2) and (3) the label's (X,Y,Z) location is determined.

3) **Calibration**: Although the label's 3D coordinate have been observed, the value still

not correct. The curvature of camera's lens may cause this problem and require lens correction. Therefore, this stage is for calibrating the original coordinate and improves the accuracy.

## IV. RESULTS

The efficiency is improved by FPGA's cooperation. In the pure MATLAB environment, the throughput of this calculation is about 0.1 frames per second (FPS). On the other hand, the FIL environment raises the FPS to 5 frames. That is 50 times of software's speed.

Figure 8 shows the operating plane's accuracy map. The Blue points are the output of this design and the red points are the standard point. Most points are close to the standard points, however the points at up-right area have more error problem. The maximum error is about 3cm.

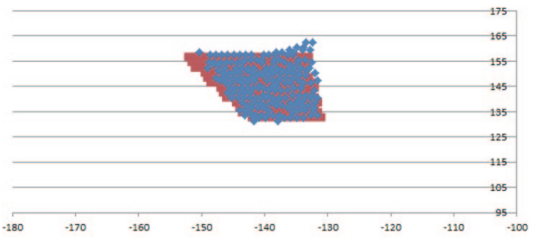


*Figure 8. The measured point (blue) VS standard point (red) in the operating area.(X,Y axia is the distance and the unit is CM )*

## DISCUSSION & CONCLUSION

The accuracy of this system is enough for usage in VR laparoscope training system. The center area has higher accuracy where the error is less than 1% and the surgery area is usually 10*10*10cm therefore the training program could operate at the center area void the error problem.

The 5 FPS calculation time seems to be not quick enough for high speed detection. We deduce the problem may be the communication between MATLAB and FPGA. This design has adopted the parallel blocks to increase the operation and the hardware should have the ability to process the loading. Therefore, the bottleneck probable is the transmitting interface of MATLAB.

This paper firstly designed the 3D coordinates calculating algorithm for locating surgery instruments in a laparoscope training system. Secondly, HDL coder is used to generate the Verilog HDL from the developed algorithm in Simulink. Next, utilization of the FIL is for verification the generated HDL coder and has provided this architecture could be employed in a laparoscope training system.

In the future, generated HDL code can be integrated into a system on programmable chip (SOPC) system for increasing the processing ability such as Altera's Qsys. If the FPGA could control the CCD sensor directly then the data could input to the processing hardware when the data just getting from sensor. Thus, the latency should be reduced and can move from MATLAB's FIL system to a standalone chip. In our estimations, the processing speed could rise to more than 30 FPS. Of course, designer can integrated this SOPC chip with any kind of computer to build a new laparoscope surgery training system.

## REFERENCES

[1]. R. Aggarwal, P. Crochet, A. Dias, A. Misra, P. Ziprin and A. Darzi, "Development of a virtual reality training curriculum for laparoscopic cholecystectomy," British Journal of Surgery, vol. 96, no. 9, pp. 1086-1093, Sep., 2009.

[2]. P. M. Pierorazio and m. E. Allaf, "Minimally invasive surgical training: challenges and solutions," Urologic Oncology, vol. 27, no. 2, pp. 208-213, Mar-Apr, 2009.

[3]. H. Egi et al., "Scientific assessment of endoscopic surgical skills," Minimally Invasive Therapy & Allied Technologies, vol. 19, no. 1, pp. 30-34, 2010.

[4]. J. Hwang et al., "A novel laparoscopic ventral herniorrhaphy training system," Surgical Laparoscopy Endoscopy & Percutaneous Techniques, vol. 20, no. 1, pp. e16-28, Feb., 2010.

[5]. C. R. et al., "Effect of virtual reality training on laparoscopic surgery: randomized control trial," BMJ, vol. 338, May, 2009.

[6]. P. Kanumuri et al., "Virtual reality and computer-enhanced training devices equally improve laparoscopic surgical skills in novices," JSLS, vol. 12, no. 3, pp. 219-226, Jul-Sep, 2008.

[7]. K. S. Gurusamy, R. Aggarwal, L. Palanivelu, B. R. Davidson, "Virtual reality training for surgical trainees in laparoscopic surgery," Cochrane Database Systemic Review, vol. 1, CD006575, Jan. 21, 2009.

[8]. E. M. McDougall EM et al., "Preliminary study of virtual reality and model simulation for learning laparoscopic suturing skills," The Journal of Urology, vol. 182, no. 3, pp. 1018-1025, Jul., 2009.

[9]. MathWorks, Simulink - Simulation and Model-Based Design [Online]. Available: http://www.mathworks.com/products/simulink/

[10]. MathWorks, VHDL code - HDL Coder - MATLAB & Simulink [Online]. Available: http://www.mathworks.com/products/hdl-coder/index.html

[11]. MathWorks, VHDL Test Bench - HDL Verifier - MATLAB & Simulink [Online]. Available: http://www.mathworks.com/products/hdl-verifier/index.html

[12]. MathWorks, Introduction to FPGA Design Using MATLAB and Simulink - MathWorks Webinar [Online]. Available: http://www.mathworks.com/company/events/webinars/wbnr60358.html?id=60358&p1=942727305&p2=942727310

[13]. A. D. Marshall, Vision Systems [Online]. Available: http://www.cs.cf.ac.uk/Dave/Vision_lecture/

[14]. J. Y. Bouguet (2010, July, 9), Camera Calibration Toolbox for Matlab [Online]. Available: http://www.vision.caltech.edu/bouguetj/calib_doc/