# ReachIN: A Modular Vision Based Interface for Teleoperation

*Release 1.00*

Kelleher Guerin[1], Balazs Vagvolgyi[1], Anton Deguet[1], Chi Chiung Grace Chen MD[2], David Yuh MD[2] and Rajesh Kumar[1]

August 14, 2010

[1]Johns Hopkins University, Baltimore, MD
[2]Department of Surgery, Johns Hopkins Hospital, Baltimore, MD

**Abstract**

Teleoperation is currently the dominant form of instrument control for surgical robotics. For example, the mechanical masters in the da Vinci surgical system provides a stable, precise interface for teleoperation, allowing execution of delicate tasks in complex surgical procedures. However, conventional teleoperation also adds the complexity of false feedback to the user from the interaction of the master manipulators with each other and other elements of the surgical console. Conventional teleoperation also has other inherent performance and accuracy limitations due to the mechanical devices integrated in such a system. As an alternative, we outline a hands-free system that could be improved without significant hardware redesign, and provide a sterile master workspace. Our *ReachIN* prototype is implemented by extending the Johns Hopkins University Surgical Assistant Workstation (SAW) framework to include additional vision interfaces and control methods. We present the design of this prototype and results from initial validation experiments.

## Contents

## 1 Introduction

Teleoperated systems such as the the da Vinci surgical system provide an operator with a stable, precise hands-on interface for teleoperated control, enabling the performance of delicate tasks in surgical procedures.[4] . But teleoperation also has several limitations. The complexity of the mechanical structures needed for precise and transparent manipulation also make the master manipulators difficult to sterilize. As the masters provide no tool-tissue interaction feedback due to the lack of instrument tip sensing, a vision based hands-free interface could provide an alternative with similar accuracy while reducing the complexity of operation, and with the inherent sterility of a zero contact interface. With appropriate vision and hand tracking methods, a surgeon could be garbed in normal sterile gloves without effecting the interfaces' performance.

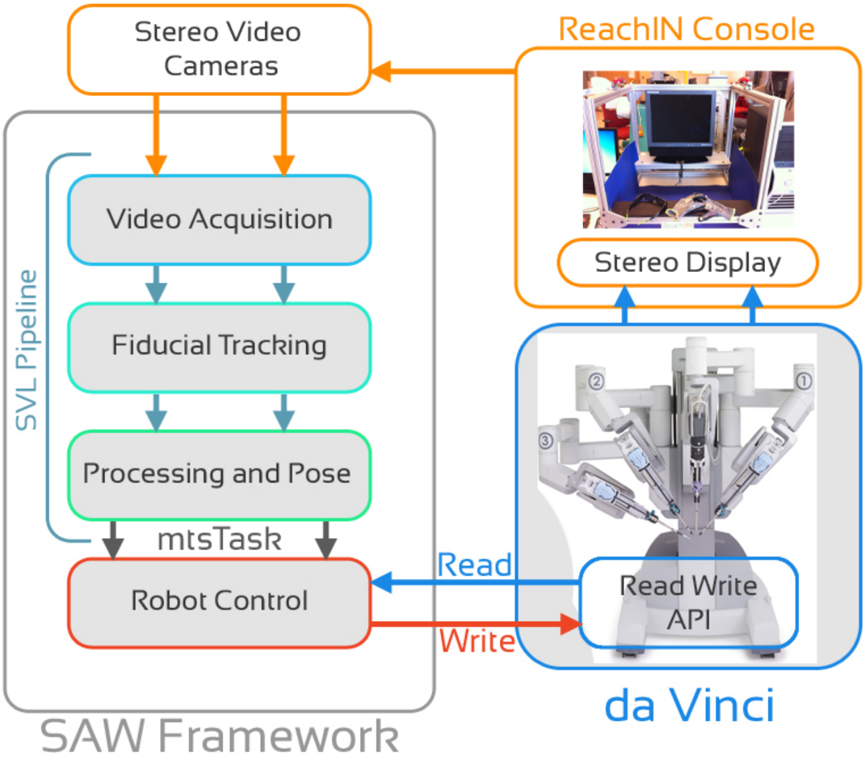

Figure 1: Overview of the SAW architecture for hands-free telemanipulation.

The Surgical Assistant Workstation (SAW)[7] framework provides a platform for rapid prototyping of such a vision based gestural interface. The SAW and and underlying cisst libraries [3] include integrated support for robotic devices, imaging sensors, a visualization pipelines[9], and networking support[6]. We extend the SAW architecture with additional vision based gestural input methods for teleoperation and telemanipulation.

# 2 Architecture

## 2.1 Hardware

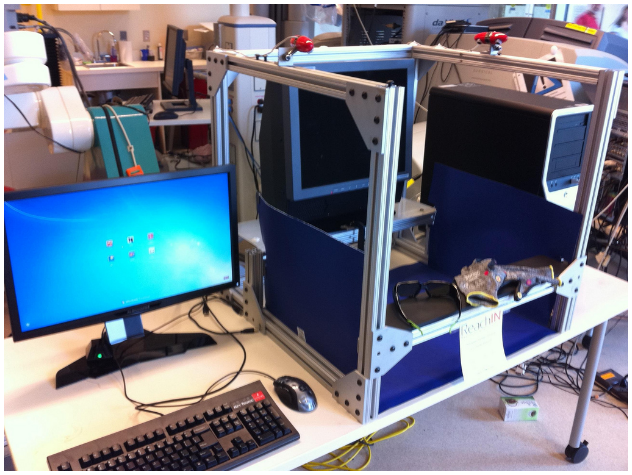

Figure 2: The prototype telemanipulation console

We use commercially available components in the prototype console. A rigid aluminum framework (figure 2) integrates a hand-rest similar to the conventional telemanipulation console and provides rigid hard points for the vision system. The physical configuration of the user's seated position, hand rest and usable hand volume are designed to match the user ergonomics of the da Vinci master console, to allow for direct comparison of the two methods of manipulation. The framework also defines the workspace for hand and gesture tracking. A Minoru USB stereo imager digitizes (640x480, RGB, stereo) each hand for further processing. A Dell Precision workstation (dual quad core 2.8 GHZ processors, 12GB RAM) allows for fast processing and analysis of the acquired data. The workstation display provides real-time task feedback and a command console for system mode changes, since the foot pedal control integrated in the master console have not been translated into hand gestures yet. A 3D display (1280x1024) is positioned at the user's eye level beyond the sensing volume. The monitor displays stereo endoscopic video digitized from the serial digital interface (SDI) using a separate workstation with dual Matrox Vaio digitizers that also supports robot control. The overall latency in the video chain is a comparable 2-3 frames per second. We will also shortly transition to a true HD (1920x1080) display to take full advantage of the HD video available from the surgical endoscope.

We use fiducial based hand tracking for system integration and testing. A user wears close fitting gloves with unique 0.5 inch planar disc fiducials allowing robust detection and tracking. The fiducials mark four key points, the tips of the thumb and forefinger, and the base of these fingers, providing 3 DOF tracking of hand position and a gripper position indicated by the fingers.

## 2.2 Software Architecture

The software architecture consists of three separate tasks (Figure 3) implemented using the *cisstMultiTask* framework[3] - a visualization task, a robot control task, and an image processing task.

The visualization task displays the stereo endoscopic da Vinci video on the 3D monitor using a Stereo Vision Library (SVL) pipeline [9]. The task consists of two threads, one for each channel of the stereo pair, and integrates filters for capture, conversion, and output of two video streams in a format compatible with inputs on the 3D display.

The robot control task reports the current instrument state and commands the instrument position. *cisstMultiTask* provides a framework for proxy based, thread safe data exchange between tasks. Each task is equipped with a *required interface* and *provided interface*, with communication occurring across the connection of a required interface to a provided one. In the case of the robot control task, data exchange with the robot takes place using the Internet Communication Engine (ICE) middle ware [6]. The robot control task,

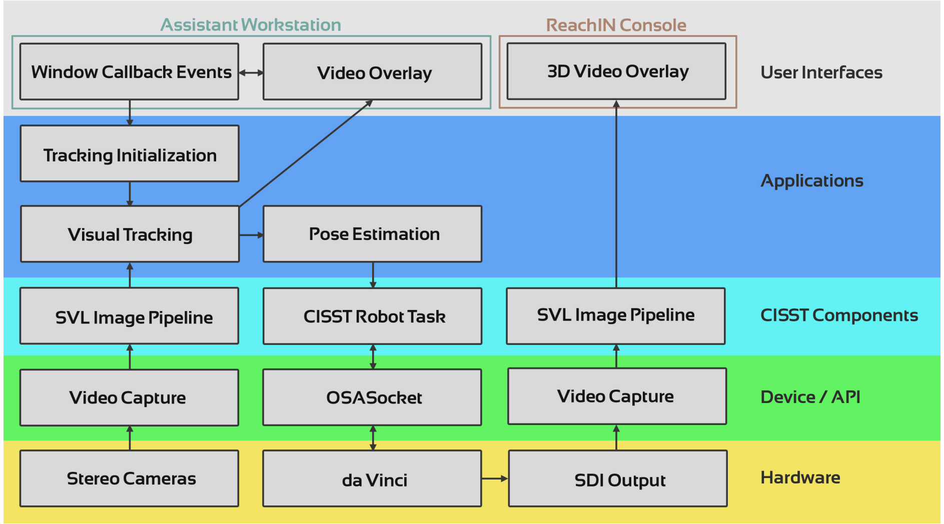

Figure 3: Software Overview

via *cisstMultiTask commands* and *required / provided interfaces*, receives robot pose and status information, and can send state commands (such as *clutch* or *following*) and Cartesian frame information to drive robot position and orientation.

The image processing task consists of four threads, two for each stereo camera, and integrates filters for acquiring video frames from the stereo cameras, detection and tracking of hand fiducials, and display of an image overlay of the fiducial position in video in the auxiliary user interface. The image processing task uses a pose estimation method for converting the fiducial locations into current hand pose and provides it to the robot control task. *cisstMultiTask interfaces* are again used to connect with image processing task with the robot control task.

**Calibration:** Camera calibration was accomplished using the *Matlab Camera Calibration Toolbox*[2] , with a reprojection error of less than .18 pixels for the hand pose cameras. Stereo calibration to estimate intrinsic and extrinsic parameters of the two stereo camera pairs were also performed with the Matlab Toolbox using a calibration checkerboard with 12.7mm squares printed at 300dpi. Each stereo camera reports the hand Cartesian pose in its own reference frame. To register the two hands, a calibration object (a box with fiducials) was imaged simultaneously with both imagers, and point correspondences for visible fiducials were manually computed. A rigid registration between the two cameras was computed (in the left hand Coordinate frame) was computed using a least-squares fit [1]. The relatively small RMS error (0.8 mm) is likely due to the manual selection of point correspondences.

**Image Processing:** Digitized images from each hand image are processed in the image processing task as follows:

1. A pair of stereo images is captured from each camera.
2. Each pair is converted into the YUV color space for robust tracking.
3. All hand fiducials are detected .
4. Fiducials positions are localized from the detected fiducials in each pair of images.
5. Fiducial positions are validated with a confidence factor.

6. Image positions are converted into 3D positions by back-projection.
7. A hand pose is computed using an articulated model consisting for the two fingers.
8. The image pair is displayed with an overlay of detected fiducials for development purposes on the auxiliary display.

SSD kernel based tracking [5] filters integrated in the SVL pipeline were used to track the fiducials. Initial kernels were acquired by manual selection on the auxiliary display. The image processing pipeline continues to run in the absence of initialized fiducials, with the tracking confidence is initialized to zero. Once the fiducials are initialized and are being tracked, full telemanipulation functionality of the pipeline is restored. Tracked fiducial positions are also passed to a video overlay which draws a circle around the tracked fiducial in the video output on the auxiliary display for for monitoring the tracker performance for development purposes.

**Pose Estimation:** Fiducial locations in the stereo images $(u,v)$ are converted into 3D world coordinates $(x,y,z)$ by back-projection. 3D fiducial coordinates are initially estimated in the individual stereo camera coordinate spaces, and transformed into the common reference frame using the registration obtained above. Given $F_{ib} = (x_{ib}, y_{ib}, z_{ib})$, $F_{tb} = (x_{tb}, y_{tb}, z_{tb})$, $F_{it} = (x_{it}, y_{it}, z_{it})$, and $F_{tt} = (x_{tt}, y_{tt}, z_{tt})$ as positions of the fiducials at the as the position of the two fiducials at the base of the index finger and thumb, and tip of the index finger and thumb respectively, a hand "grip" position (coupled to the instrument tips) can be calculated as

$$H_c = (F_{ib} + F_{tb})/2$$

The tip of the hand is calculated as

$$H_t = (F_{it} + F_{tt})/2$$

Though the orientation was not used here, the coordinate system for hand orientation $X_p, Y_p, Z_p$ was estimated as :

$$X_p = H_c + H_t$$
$$Z_p = \frac{H_c - F_{it}}{||H_c - F_{it}||} \times \frac{H_c - F_{tt}}{||H_c - F_{tt}||}$$
$$Y_p = X_p \times Z_p$$

The gripper angle is separately calculated by normalizing the distance between fingertip fiducials for each hand. A hand position and gripper angle is then provided to the robot control task using a *mtsTask required interface*. Let $F_x$ describe a position. The robot control task uses a conventional teleoperation control law:

$$F_{Instrument} - F_{InstrumentOffset} = scale * (F_{Hand} - F_{HandOffset})$$

A moving average of 5 frames is used to filter the estimated hand pose to reduce noise and jitter. In these experiments, the instrument orientation was not changed during hands-free control.

**Robot Control:** A SAW interface wrapping a custom proprietary interface allows position control of instrument slave on our research da Vinci S system to the commanded pose. This interface allows for commands to be sent to the robot and the current pose and state to be read at rates greater than 100Hz. The instruments are commanded in the endoscopic camera reference frame as usual, a known fixed transformation (rotation by 90 degrees about camera z axis) aligns the hand reference frame with the camera frame.

**User Interface:** An auxiliary display allows an assistant to perform mode changes in the hands-free system. ”Enable/Disable” commands activate and deactivate robot motion similar to the head-in/out sensing in the da Vinci console, preventing inadvertent motion. When instrument motion is disabled, a user may also freely reposition their hands, and when enabled, motion starts relative to the previous robot position. As another safeguard, the user is still required to pinch figures to start slave motion, much like the da Vinci console. In addition, the system also records the hand and instrument poses for post-processing and analysis.

## 3 Experiments

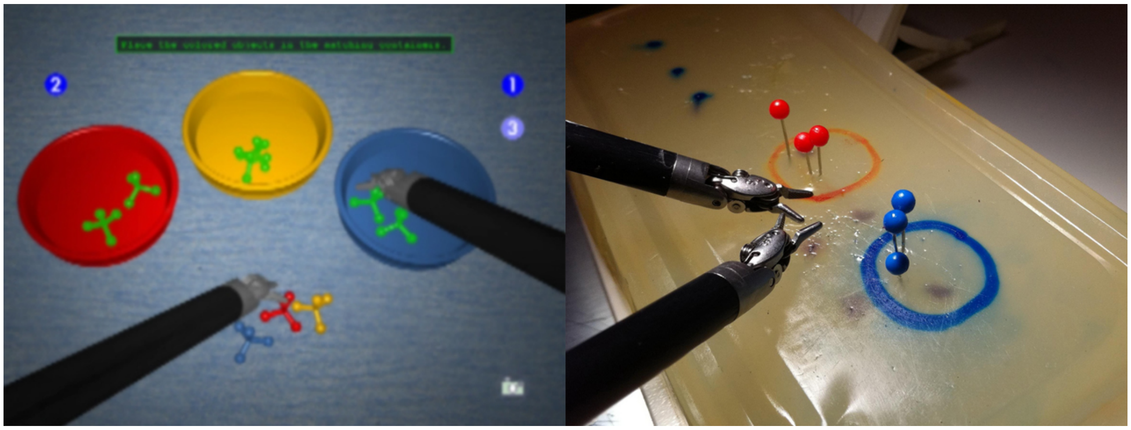

Figure 4: The pick and place Task: dV-Trainer simulator task (left), experimental setup for the ReachIn evaluation task upon completion.

We have performed initial manipulation experiments for system integration and to assess positioning accuracy, and to estimate parameter ranges for instrument manipulation. A precision pointing task was used to measure the accuracy and stability of the computed hand pose, and a pick and place task was used to evaluate usability.

An inclined planar grid of point fiducials (1.4 mm diameter) was mounted in the instrument workspace and the user was required to move the instrument tip to each point. Figure 5 shows the experimental setup. The three fiducials at the triangle vertices were used for computing positioning accuracy. The instrument tips were moved to these fiducials a total of 8 times and an average position error was computed. Table 1 shows the current positioning accuracy for each hand in the master workspace. With an appropriate scaling factor (3-5), the accuracy is comparable to current conventional manipulation.

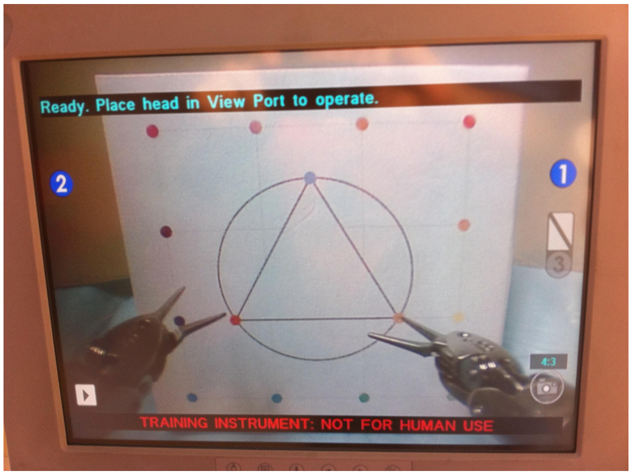
Ready. Place head in View Port to operate.
2
1
3
4:3
TRAINING INSTRUMENT: NOT FOR HUMAN USE

Figure 5: Pointing Task, showing the grid of 1.4 mm points.

A pick and place task required moving red and blue pins from a central location to two designated red or blue circular bins. This task is modeled after the jacks sorting task in the MIMIC Technologies dV-Trainer[8]. Figure 4 shows the finished pin positions after the ReachIN trial, and the MIMIC dV-Trainer task. The pin starting locations were chosen randomly to necessitate movement of the instruments across the center of the experimental workspace (blue and red pins were mixed together).

The task was performed using both the da Vinci master and the ReachIN console, and completion times were measured. With a robotic surgery novice user, it took 47.8 seconds to perform this task using the da Vinci console, and a very usable but slower 86.1 seconds using the hands free console.

| Accuracy (mm) | X | Y | Z |
|---|---|---|---|
| Right Hand | 1.97 | 1.98 | 3.82 |
| Left Hand | 3.19 | 1.59 | 2.18 |

Table 1: Positioning error (mm) in the pointing experiment.

## 4 Conclusion

We outline the development of a hands-free telemanipulation master prototype, and its integration into a telemanipulation prototype integrating a da Vinci surgical system using the JHU SAW framework. Initial results in limited system integration experiments show the promise of simpler telemanipulation using the developed prototype. Ongoing work includes using fiducial free hand tracking, multiple stereo cameras for each hand to obviate occlusion, and improved gesture interfaces, and detailed experimental validation.

## 5 Acknowledgments

This work was supported in part by a Phillips and Camille Bradford Fellowship , National Science Foundation grants #0941362, and #0931805, a collaborative agreement with Intuitive Surgical, Inc, and Johns Hopkins University internal funds. Special thanks to Nicolas Padoy at Johns Hopkins University for assistance with the da Vinci robot interface.

## References

[1] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Trans. Pattern Anal. Mach. Intell.*, 9(5):698–700, 1987. 2.2

[2] J. Y. Bouguet. Camera calibration toolbox for matlab, 2008. 2.2

[3] A. Deguet, R. Kumar, R. H. Taylor, and P. Kazanzides. The cisst libraries for computer assisted intervention systems. In *IJ - 2008 MICCAI Workshop - Systems and Architectures for Computer Assisted Interventions, Midas Journal, http://hdl.handle.net/10380/1465*, 2008. 1, 2.2

[4] G. Guthart and J. Salisbury. The Intuitive$^{TM}$ telesurgery system: Overview and application. *Proc. IEEE International Conference on Robots and Automation*, 1:618–621, 2000. 1

[5] G. D. Hager, M. Dewan, and C. V. Stewart. Multiple kernel tracking with ssd. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1:790–797, 2004. 2.2

[6] M. Y. Jung, R. Kumar, A. Deguet, R. Taylor, and P. Kazanzides. A surgical assistant workstation (saw) application for a teleoperated surgical robot system. In *IJ - 2009 MICCAI Workshop - Systems and Architectures for Computer Assisted Interventions, Midas Journal, http://hdl.handle.net/10380/3079*, 2009. 1, 2.2

[7] P. Kazanzides, S. P. DiMaio, A. Deguet, B. Vagvolgyi, M. Balicki, C. Schneider, R. Kumar, A. Jog, B. Itkowitz, C. Hasser, and R. Taylor. The surgical assistant workstation (saw) in minimally-invasive surgery and microsurgery. In *IJ - 2010 MICCAI Workshop - Systems and Architectures for Computer Assisted Interventions, Midas Journal, http://hdl.handle.net/1926/0*, 2009. 1

[8] E. McDougall. Validation of surgical simulators. *Journal of Endourology*, 21(3):244–247, 2007. 3

[9] B. Vagvolyi, S. DiMiao, A. Deguet, P. Kazanzides, R. Kumar, C. Hasser, and R. Taylor. The surgical assistant workstation. In *IJ - 2008 MICCAI Workshop - Systems and Architectures for Computer Assisted Interventions, Midas Journal, http://hdl.handle.net/10380/1466*, 2008. 1, 2.2