Towards webcam-based tracking for interventional navigation

Mark Asselin, Andras Lasso, Tamas Ungi, Gabor Fichtinger

Laboratory for Percutaneous Surgery, School of Computing, Queen's University, Kingston, Canada

ABSTRACT

PURPOSE: Optical tracking is a commonly used tool in computer assisted surgery and surgical training; however, many current generation commercially available tracking systems are prohibitively large and expensive for certain applications. We developed an open source optical tracking system using the Intel RealSense SR300 webcam with integrated depth sensor. In this paper, we assess the accuracy of this tracking system.

METHODS: The PLUS toolkit was extended to incorporate the ArUco marker detection and tracking library. The depth data obtained from the infrared sensor of the Intel RealSense SR300 was used to improve accuracy. We assessed the accuracy of the system by comparing this tracker to a high accuracy commercial optical tracker.

RESULTS: The ArUco based optical tracking algorithm had median errors of 20.0mm and 4.1 degrees in a 200x200x200mm tracking volume. Our algorithm processing the depth data had a positional error of 17.3mm, and an orientation error of 7.1 degrees in the same tracking volume. In the direction perpendicular to the sensor, the optical only tracking had positional errors between 11% and 15%, compared to errors in depth of 1% or less. In tracking one marker relative to another, a fused transform from optical and depth data produced the best result of 1.39% error.

CONCLUSION: The webcam based system does not yet have satisfactory accuracy for use in computer assisted surgery or surgical training.

Keywords: interventional navigation, tracking, webcam, open source

1. INTRODUCTION

There are many applications in computer assisted surgery where the tracking of a surgical tool is essential to guiding an intervention. Many methods have been devised to this effect, including those which depend on electromagnetic sensing (colloquially known as EM trackers) and those using RGB or infrared optical sensing methods (optical trackers). In typical use optical sensors have superior accuracy to their EM counterparts, but they require a clear line of sight between the tracked tool and the tracker itself. Most commercially available optical trackers rely on stereo infrared cameras to capture the pose of a tool in the scene. The current generation of commercially available optical tracking systems used in medical interventions are accurate, reliable, and designed to be integrated into the surgical workflow. The main drawback of such systems is their significant cost, which can place them out of reach of some health centers and keep such tools out of the hands of trainees. One other disadvantage of the commercial trackers is their bulky form factor which makes them inconvenient for use in many environments, such as in small operating rooms. The exquisite accuracy provided by these systems is essential for performing computer-guided interventions in operating rooms, but the prohibitive costs and inconvenient form factor limit widespread usage of this technology for a range of other applications, such as computer-assisted medical training.

The aforementioned drawbacks of current surgical navigation systems and the popularization of inexpensive RGB webcams have resulted in increased interest in the development of a pose tracking system which uses off the shelf components, costs far less, is easier to use and has a smaller footprint. Many libraries have been developed which have this goal. Most libraries rely on the commonplace RGB camera and make use of thresholding and pattern recognition to identify patterns of known geometry in the image. These pre-defined pattern geometries are usually called markers, and are used to track the tools to which they are attached. In many libraries these markers are square patterns of black and white bits, however some toolkits use circular or irregular geometries instead. Trackers making use of an RGB camera and these marker libraries often have good accuracy in the plane of the image, but struggle to determine the distance of objects. This is due to the inability of mono-optical cameras to directly measure depth – these systems rely on knowledge of camera intrinsics and marker geometry to compute the depth. Existing solutions which attempt to improve the tracking accuracy have leveraged emerging commercially available infrared depth sensors, such as the Microsoft Kinect (Microsoft Corporation, Redmond, WA, USA). Existing systems using the Kinect sensor can track needles [1] or to perform pose estimation of the human skeleton [2]. The original Intel RealSense SDK (Intel, Santa Clara, CA, USA) shipped with its own marker recognition and tracking toolkit [3]; however, in recent versions of the SDK this toolkit has been removed.

2. PURPOSE

In this paper, we develop a system which addresses the cost and form factor drawbacks of commercially available optical trackers and the depth reconstruction drawbacks of mono-optical solutions. Our optical tracking device uses a mono-optical tracking library to determine the marker pose components accurately computable from an RGB image. We augment this computation with the analysis of point-cloud data of the scene obtained by infrared depth sensor to generate the remaining components of the markers pose. Several inexpensive pieces of hardware have been developed by manufacturers with this ability to capture RGB mono-optical and depth point cloud images aligned in space and time. We selected the Intel RealSense SR300 (Intel, Santa Clara, CA, USA), a camera equipped with both a mono-optical RGB camera and infrared laser depth sensor. We use the Intel RealSense SDK's ability to output an infrared depth stream temporally and spatially aligned with its RGB video stream. The tracking system we built is designed to combine the strengths of the SR300's mono-optical and infrared depth tracking sensors to compute the pose of planar black and white markers in the field of view of the camera. We used the mono-optical sensor to perform tracking using the ArUco library, an OpenCV based toolkit for marker recognition and pose computation. This library, like all mono-optical solutions for pose tracking, struggles to reconstruct the depth of a marker in the scene. To improve upon this we leverage the infrared depth sensor of the SR300 to determine the component of the markers pose which is orthogonal to the plane of the camera sensor.

This solution has several advantages, including the ubiquity of laptop webcams, removing the barrier to entry of expensive and specialized tracking equipment that currently limits the accessibility of optical navigation. This technology increases the usability of interventional navigation outside of the OR environment, opening up the possibility of using optical tracking at the patient's bedside, or in clinics where the use of such tracking would previously have been infeasible.

3. METHODS

We began by extending the open source PLUS toolkit to include a device to support the acquisition of data from the Intel RealSense SR300's RGB optical and infrared depth sensors. We then utilized a marker tracking library to identify the markers in the scene and generate an initial 6DOF pose estimate of each marker visible to the camera. The system was then extended to be able to compute the pose of the marker using the infrared depth stream. We measured the position and orientation errors of the optical and infrared depth tracking algorithms against a high accuracy optical tracker, considering the centers and peripheries of the SR300's field of view. Next, we tested the accuracy of tracking one marker relative to another, a common task in surgical navigation when the patient's position is tracked by a reference marker fixed to the patient.

3.1 Camera Selection

Building on the work of House *et al.* [4], we selected the Intel RealSense SR300 Generation 2 camera (Intel, Santa Clara, CA, USA) as our camera system of choice. It has many desirable characteristics for motion tracking including; small form factor, low cost ($150US), and fixed focus. The main feature of the camera is that it is equipped with both an HD RGB optical sensor and an infrared scanning depth sensor. The optical sensor provides a color image of the scene in front of the camera, and the depth sensor provides a voxel aligned matrix representing the distances between the camera and the first solid object in each image voxel. Intel specifies that the SR300's depth sensor is optimized in the range of 20 to 150cm from the sensor [5], an ideal range for most computer-assisted interventions.

3.2 SR300 Camera Coordinate System

When working with pose tracking hardware, we must define the reference coordinate frame. While there are many plausible frames of reference, we facilitate the following discussion by defining the coordinate reference frame of the SR300 in an analog to the right-anterior-superior (RAS) system used in 3D Slicer. For an observer positioned at the camera looking outward into the camera frustum: the x axis points right, the z axis points upwards and the y axis points away from the camera orthogonally to the plane of the image sensor. These coordinates mimic the RAS coordinates that are intuitive to humans where the x axis maps to right, the y to anterior and z to superior. (Figure 1).

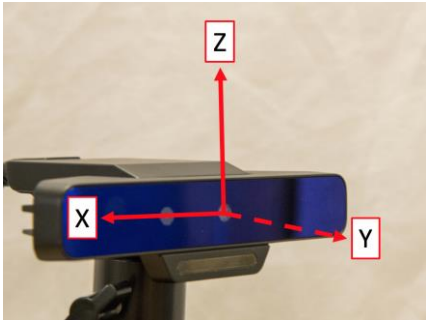

Fig 1. Intel SR300 showing the cameras local coordinate system.

3.3 Dual Modality Tracking Hardware

To measure the tracking errors of the RGB optical and depth tracking systems relative to the NDI Polaris optical tracker, we designed a rigid body which could be tracked by all three devices using SolidWorks 2015 SP05 (Dassault Systèmes, Vélizy, Île-de-France, France). The tool was 3D printed on a Dimension SST 1200es (Stratasys, Eden Prairie, MN, USA). We used the standard NDI mounting posts and reflective spheres intended for use with the Polaris optical tracker. The resulting tool has 4 reflective spheres organized in a geometry that can be tracked by the NDI Polaris, and a planar surface on which we stuck an ArUco marker printed on an adhesive label. The tool also has a pointed stylus which is used to perform pivot and spin calibrations to locate the tip of the tool with respect to the marker coordinate systems of the various trackers (Figure 2).

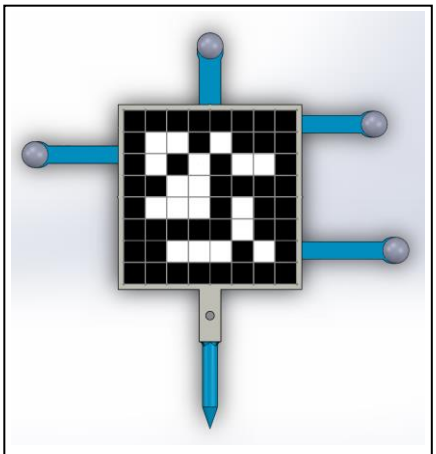

Fig 2. Triple modality stylus.

A challenge faced in setting up multi-camera optical tracking systems is locating cameras with respect to each other. One solution is to calibrate the cameras in their final positions, and then to take great care not to move any camera out of this position. A second solution is to mount both cameras on the same tripod, rigidly affixing them to one another. While this prevents tripod movement disturbing inter-camera calibration, we found this positioning caused reflections of the infrared strobes of the NDI Polaris Spectra optical tracker (NDI, Waterloo, ON, Canada) to interfere with the SR300. Ultimately, we resolved this problem by mounting a camera reference marker on top of the SR300 that could be tracked by the NDI Polaris (Figure 3). Though this reference, we could determine the position of the SR300 in the Polaris' field of view and could track the triple modality stylus using the NDI Polaris in addition to the optical and depth tracking algorithms based on the SR300 simultaneously.

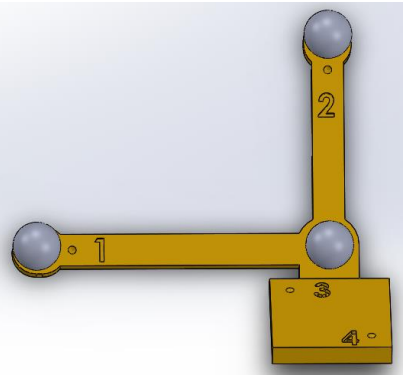

Fig 3. SR300 reference marker.

3.4 System Development

The software created for this experiment was built on top of the PLUS (Public Library for Ultrasound) toolkit created by Lasso *et al*. [6]. PLUS has an established codebase in medical tracking, providing support for many commercially available medical electromagnetic and optical tracking systems. This toolkit already included support for interfacing with the high accuracy NDI Polaris Spectra optical tracker which enabled us to acquire and transmit data from the tracker into the 3D Slicer application. The Polaris optical tracker is used as a ground truth to measure tracking error against. We then extended PLUS by adding a device to support the acquisition of images and depth data from the Intel SR300's optical sensor and infrared depth sensor.

To perform detection of fiducial markers using the SR300's optical stream, we utilized the open-source ArUco library [7] developed by Garrido-Jurado *et al.* ArUco uses its own planar marker set designed to maximize the inter-marker distance preventing confusion between markers of similar patterns (Figure 4). The ArUco marker library was selected over other similar libraries for its ability to perform error detection and correction. The ArUco library is able to detect the markers in the SR300's optical stream providing 6DOF tracking (position in x, y, z, and rotation about all three axes) as well as the location of the corners of the marker in the image.

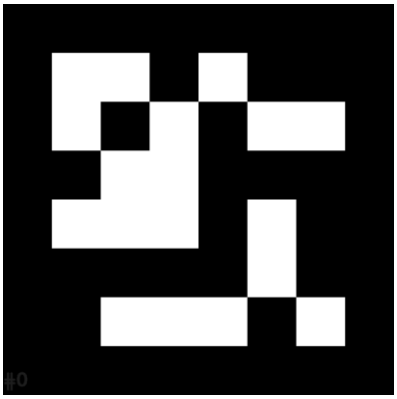
Fig 4. A typical ArUco library marker.

Early error evaluation indicated that the ArUco tracking system using only the RGB optical data stream suffered difficulty in computing the depth of markers in the scene. We suspected this inaccuracy was a result of trying to reconstruct the depth of the scene using only the pose of the marker in the optical video feed. This is a result of a mono-optical cameras lack of stereo vision – meaning it must rely on knowledge of camera intrinsics and marker geometry to compute the depth of markers in the scene.

Once the depth data frame has been received from the Intel RealSense, the first step in processing is to extract the plane of the ArUco marker from the point cloud data collected about the whole scene. The challenge with segmenting the ArUco marker plane is that it must be performed for every marker in the scene, and for every frame. Thus, we needed to be careful to select this data in an efficient manner. To achieve sufficient segmentation speeds, we generated the depth data using the RealSense SDK such that the point cloud was stored with voxels indexed based on the position the corresponding pixel in the optical image. This gives us constant time access to the depth voxel that corresponds to any pixel in the optical image, using only the (x, y) position of the pixel of interest in the optical image.

To extract the plane from the raw point cloud, we first compute the corners of the markers using the ArUco library. We then determine which of the three possible orientations the marker is in using the positions of the corners. (Figure 5). A marker is considered "Rotated" when the marker is rotated in the plane of the camera, and skew left or skew right when one side of the marker is farther away from the camera than the other. Once we have determined which orientation the marker is in, we compute the x positions of the left and right side of the image using lines drawn between adjacent corners. The left boundary is identified in Figure 5 in green, and the right boundary in red. To segment the marker plane, we simply iterate from the top of the marker to the bottom, selecting all points that lie in between the left and right boundaries. This operation is linear in complexity with respect to the area enclosed by the marker boundaries in voxels.

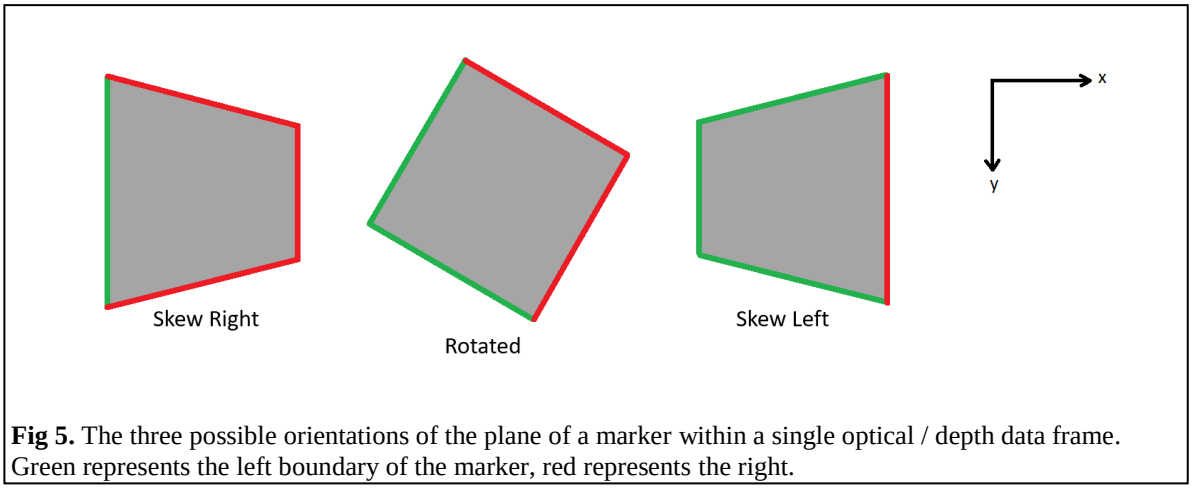

Fig 5. The three possible orientations of the plane of a marker within a single optical / depth data frame. Green represents the left boundary of the marker, red represents the right.

Using the segmented marker plane, a least squares plane fit is performed to identify the normal and center of mass of the plane points, which represents the surface of the marker in this depth frame. From this plane fit, and the x-axis provided by the ArUco 6DOF (degree of freedom) tracking, we compute the position and rotation of the depth plane in 5DOF (position in the y and z axes, and rotation about all three axes).

3.5 Experimental Setup

3.5.1 Measuring Overall Tracking Error

An important consideration when setting up an optical tracking system which makes use of multiple infrared trackers is the potential for inter-camera interference. If multiple trackers use similar infrared wavelengths, or are sensitive to the wavelengths emitted by the other trackers, then unexpected tracking abnormalities may result. To perform this experiment we needed to control these inter-camera artifacts.

After trying multiple different camera placements, we settled on the one pictured in Figure 6. This setup minimizes the interference between the SR300 and the Polaris by positioning the SR300's ArUco marker such that it won't reflect the Polaris' infrared light back into the SR300. Note the angle the SR300 is placed at relative to the NDI Polaris in this setup, this is to prevent the infrared light emitted by the Polaris from shining directly onto the SR300's infrared sensor.

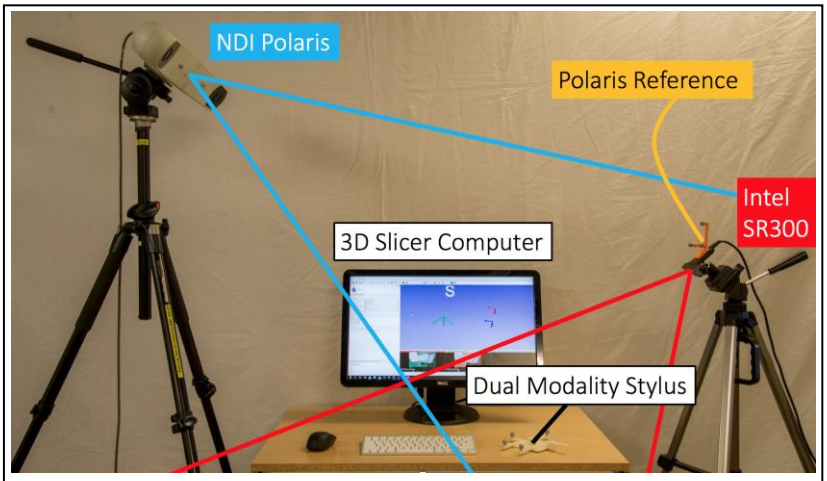

Fig 6. Experimental setup showing approximate boundaries of the Polaris and SR300's respective fields of view.

To measure the positional and orientation errors of tracking using the SR300's optical and infrared depth cameras, we used the TrackingErrorInspector extension to 3D Slicer created by Harish *et al*. [8]. First, we used its Dual Modality Calibration to align camera coordinate systems. Then, we examined the positional and orientation errors of the SR300's optical and depth tracking using the NDI Polaris as a ground truth. We examined two regions of interest, shown in Figure 7. The first was a 200x200x200mm cubic region in the center of the cameras field of view, simulating a reasonable working volume in a typical surgical procedure. The second region of interest was the periphery of the camera, which typically has the largest error. We wanted to measure the potential tracking error for other objects outside of the surgical field, such as the position of a surgeon's head or some another piece of equipment.

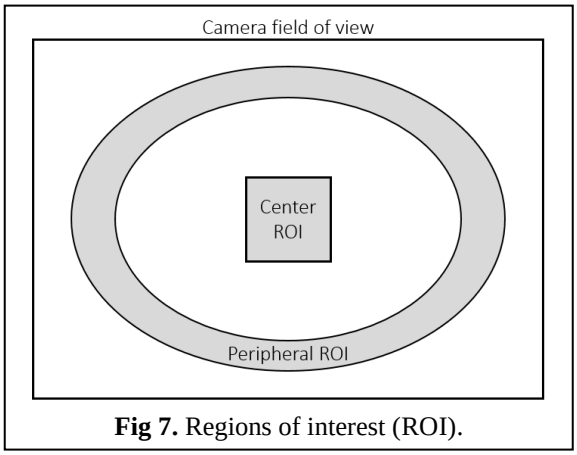

Fig 7. Regions of interest (ROI).

3.5.2 Measuring Absolute Error in Y-Direction

The setup detailed in section 3.5.1 has the disadvantage of aggregating all positional errors into a single metric. Visual observation of the errors and consideration of the difficulty of reconstructing the depth position from the corners of an ArUco marker led us to devise the following experiment to examine the absolute errors in the y axis. The Intel RealSense SR300 was placed on a gridded mat along with an ArUco marker separated by a known distance as shown in Figure 8. The marker was moved through a range of distances and the mean of 50 measurements of the position of the marker were taken using both the optical depth sensors at each distance. We compared the position returned from the optical and depth streams of the SR300 to ground truth distances calculated from the geometry of the mat.

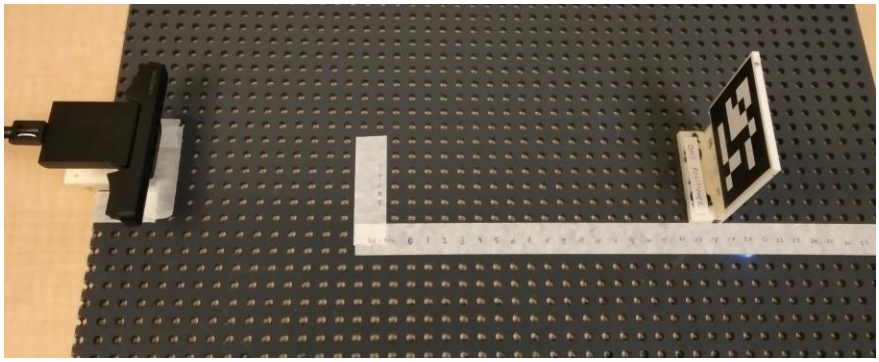

Fig 8. Measurement setup for absolute error in y direction.

3.5.3 Measuring Relative Tracking Error

In many but not all interventional navigation applications tools are tracked relative to a reference marker. In this configuration the inter-marker relative accuracy is of paramount importance to the absolute accuracy of the system. To specifically examine the relative error we fixed two markers a known distance apart (Figure 9) and then exposed the rigidly attached markers to the SR300 based tracking system. The system was configured to compute the distance between the two markers. We took 400 samples while moving the markers in 6 degrees of freedom in the field of view of the camera. We computed the percent error of the mean distance with respect to the known inter-marker distance as a measure of the inter-marker distance accuracy.

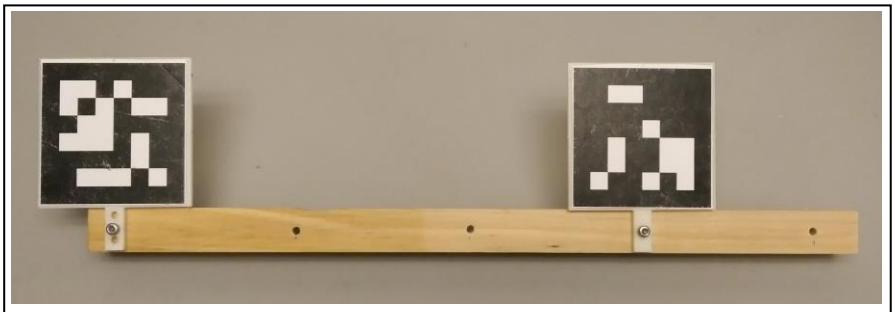

Fig 9. Measurement setup for assessing relative error.

4. RESULTS AND DISCUSSION

4.1 Overall Tracking Error

The overall position and rotation errors were measured using the experiment detailed in section 3.5.1. The positional error in the center ROI, shown in Table 1, ranged from 17.3mm for the depth sensor to 20.0mm for the optical sensor. The highest positional error was 36.1mm for the depth sensor in the peripheral ROI (Table 2). Conversely, the optical sensor had better orientation error performance in the center and periphery ROI versus the depth sensor, with errors of 4.1 and 7.1 degrees respectively. These error measurements group the errors from each degree of freedom into two aggregate error metrics, position and orientation. These metrics reliably assess overall error, but they do not provide the requisite insight to integrate the tracking streams optimally.

In this experiment specifically, we found that the SR300 was particularly sensitive to other infrared sources in its environment. If we oriented the SR300 such that its marker was allowed to face the Polaris, the Polaris' powerful infrared emitters would cause reflections off of the ArUco markers planar surface back to the SR300's infrared sensor. This caused significant artifacts to appear throughout the frame in the infrared depth feed provided by the SR300.

CENTER ROI	SR300-Optical	SR300-Depth
Median Position Error	20.0	17.3
Position Error IQR	[15.3 - 25.3]	[13.2 - 22.7]
Median Orientation Error	4.1	7.1
Orientation Error IQR	[3.4 - 5.7]	[5.3 - 8.8]

Table 1. Error measurements for the center region of interest. Position error is measured in mm, and orientation error in degrees.

PERIPHERAL ROI	SR300-Optical	SR300-Depth
Median Position Error	28.1	36.1
Position Error IQR	[24.4 - 31.6]	[23.7 - 65.0]
Median Orientation Error	6.4	13.2
Orientation Error IQR	[2.6 - 9.5]	[11.4 - 14.6]

Table 2. Error measurements for the peripheral region of interest. Position error is measured in mm, and orientation error in degrees.

4.2 Absolute Error in Y Direction

Using the experimental setup detailed in section 3.5.2 we measured the absolute tracking error of the optical and depth sensors in the y axis. As expected, the optical sensor was inaccurate in this direction due to the difficulty of reconstructing depth from the marker corners in a single mono-optical frame. However, as shown in Figure 10, the infrared depth sensor has excellent accuracy in this axis, with percent error of less than 1% for the depth sensor between 180 and 550mm from the tracker. The optical sensor in the same setup suffered errors from 11% to 15%.

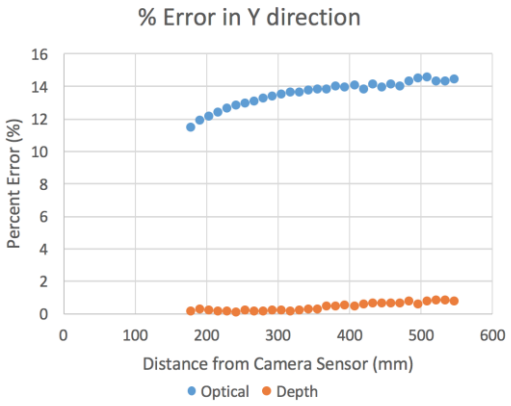

Fig 10. Percent error in the y axis.

These data suggested the use of a simplistic sensor fusion algorithm applied universally throughout the tracking volume, where the x and z position values are taken from the optical stream and the y value is taken from the depth stream. The goal of this component fusion is to correct for errors in the absolute position of the markers with respect to the camera.

4.3 Relative Tracking Error

We measured the accuracy of tracking a tool relative to a reference marker using the procedure described in section 3.5.3. The results of section 4.2 convinced us to implement the component fusion to improve the component of tracking perpendicular to the camera sensor. The mean inter-marker distance of both the optical only and component fusion tracking methods fell within 2% error of the actual measured inter-marker distance of 301.0mm (Table 3). The % error of the inter marker depth value was very high, however this was expected due to the difficulty of computing the x and z positions using this sensor. Further work fusing the optical and depth sensor readings must examine and address the consistency of the position results as well as provide adequate investigation into the properties of the rotations computed from each of these sensors.

Tracking modality	Mean Inter-marker distance	% error
SR300-Optical	306.5	1.82
SR300-Depth	394.0	30.8
SR300-Component Fusion	296.8	1.39

Table 3. Error in inter-marker distance of two relatively tracked markers.

5. CONCLUSION

The SR300 optical and depth tracking algorithms independently had positional and orientation errors which exceeded a reasonable minimum error requirement for use in interventional navigation.

ACKNOWLEDGEMENTS

This work was funded, in part, by NIH/NIBIB and NIH/NIGMS (via grant 1R01EB021396-01A1 - Slicer+PLUS: Point-of-Care Ultrasound) and by CANARIE's Research Software Program.

REFERENCES

[1] Wang, X. L., Stolka, P. J., Boctor, E., Hager, G. and Choti, M., "The Kinect as an interventional tracking system," Proc. SPIE 8316, 83160U (2012).

[2] Obdrzalek, S., Kurillo, G., Ofli, F. and Bajcsy, R., "Accuracy and robustness of Kinect pose estimation in the context of coaching of elderly population," Proc. IEEE Eng. Med Biol. Soc., 14633, 1188-1193 (2012).

[3] Intel, "The Metaio* Toolbox," Intel RealSense SDK 2016 R2 Documentation, https://software.intel.com/sites/landingpage/realsense/camera-sdk/v1.1/documentation/html/index.html?doc_ot_the_metaio_toolbox.html (January 17, 2018).

[4] House, R., Lasso, A., Harish, V. and Fichtinger, G., "Evaluation of the Intel RealSense SR300 camera for image-guided interventions and application in vertebral level localization," Proc. SPIE 10135, 101352Z (2017).

[5] Intel, “Intel RealSense Data Ranges,” Intel Developer Zone, 1 March 2016, https://software.intel.com/sites/landingpage/realsense/camera-sdk/v1.1/documentation/html/index.html?doc_advanced_module_range_specification.html (January 17, 2018).

[6] Lasso, A., Heffter, T., Rankin, A., Pinter, C., Ungi, T. and Fichtinger, G., “PLUS: open-source toolkit for ultrasound-guided intervention systems,” IEEE Transactions on Biomedical Engineering, 61(10), 2527-37 (2014).

[7] Garrido-Jurado, S., Munoz-Salinas, R., Madrid-Cuevas, F. J. and Marin-Jimenez, M. J., “Automatic generation and detection of highly reliable fiducial markers under occlusion.” *Pattern Recognition*, 47(6), 2280-2292 (2014).

[8] Harish, V., Baksh, A., Ungi, T., Lasso, A., Baum, Z., Gauvin, G., Engel, J., Rudan, J. and Fichtinger, G., “Measurement of electromagnetic tracking error in a navigated breast surgery setup,” Proc. SPIE 9786, 97862K (2016).