Johns Hopkins
Whiting School of Engineering

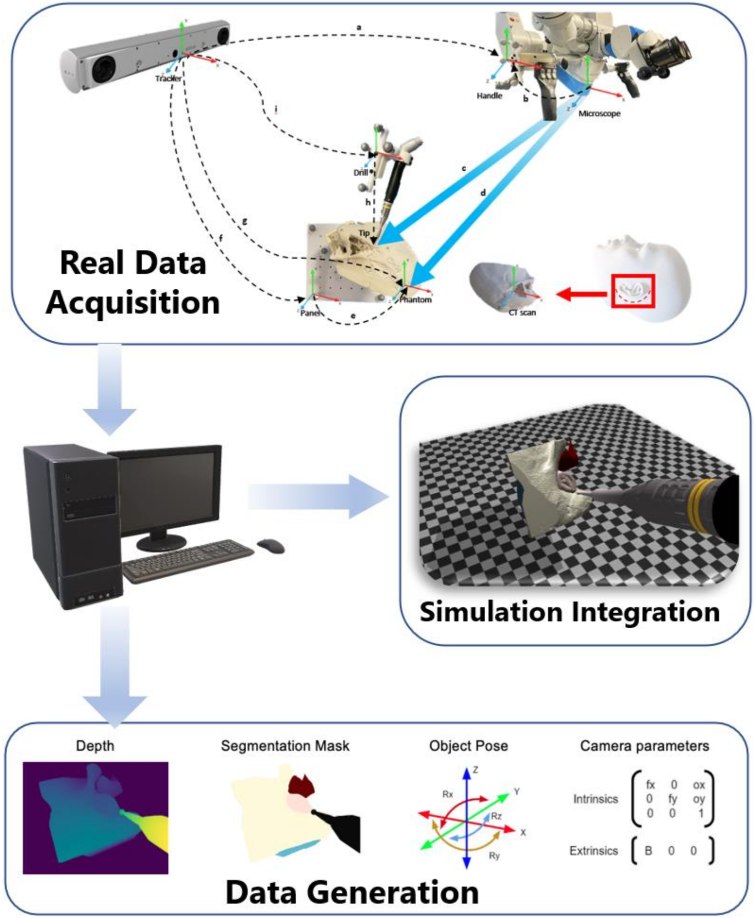
Background Presentation

VR Guided Surgery Registration Pipelines

Team members: Ruixing Liang, Hongchao Shu
Mentors: Max Zhaoshuo Li, Mathias Unberath, Russell Taylor

Goals Recap

- Maintain accurate patient-to-tool registration for downstream applications
  - Integration of Virtual Reality and Real-world data
  - To generate infinite and accurately-annotated images for Deep Neural Network training.



(Adapted from A. Munawar et al. ,2021)

Overview

Virtual Reality for Synergistic Surgical Training and Data Generation

Framework

Telecentric stereo micro-vision system: Calibration method and experiments

Calibration

Evaluation of Combined Time-Offset Estimation and Hand-Eye Calibration on Robotic Datasets

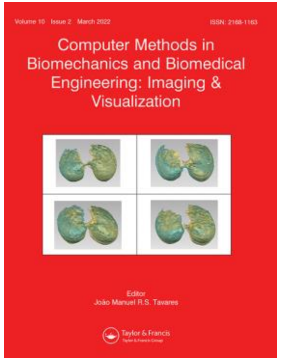
Hand Eye

References

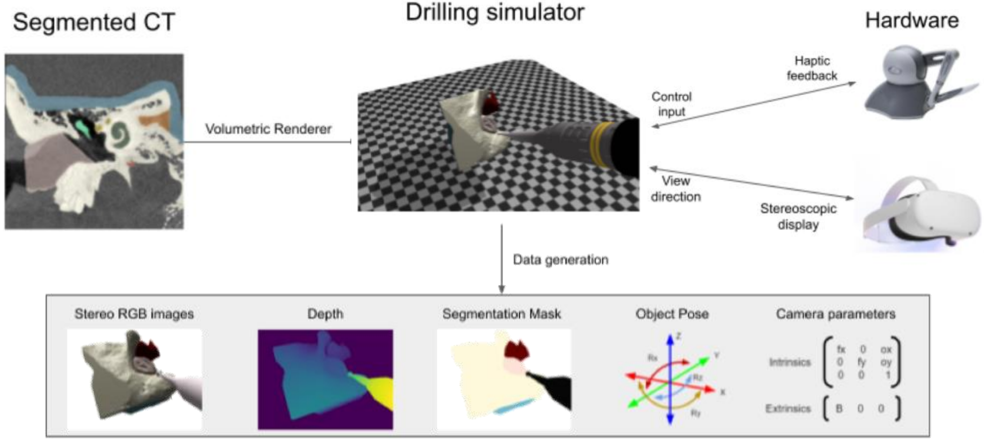
Reference

Paper 1: *Virtual Reality for Synergistic Surgical Training and Data Generation*

Adnan Munawar , Zhaoshuo Li , Punit Kunjam , Nimesh Nagururu , Andy S. Ding , Peter Kazanzides , Thomas Looi , Francis X. Creighton , Russell H. Taylora and Mathias Unberatha



- They presented developments of a cost-effective and synergistic framework, named Asynchronous Multibody Framework Plus (AMBF+), which generates data for downstream **algorithm** development simultaneously with users **practicing** their surgical skills.



1^{st} Technical Details

- ADF File (Blender Enabled)

Flexibility to create and refine objects

- CHAI3D Finger proxy collision algorithm to simulate haptic feedback
- Proposed Drill Shaft Collision Algorithm to

- Aside of great job for control and visualization

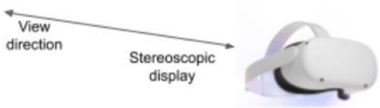
Table 1. Quantitative result (mean and standard deviation of L1 error) of ORB SLAM V3 applied on the synthetic stereo microscopic data generated by the drilling simulator.

	Translation Error (mm)	Rotation Error (deg)
Moving camera	40.97 ± 22.40	8.44 ± 3.07
Moving drill	$8.1E\text{-}1 \pm 9.1E\text{-}1$	$3.2E\text{-}3 \pm 3.6E\text{-}3$

Performance and Shading Textures

Algorithm 1 Depth Computation Shader

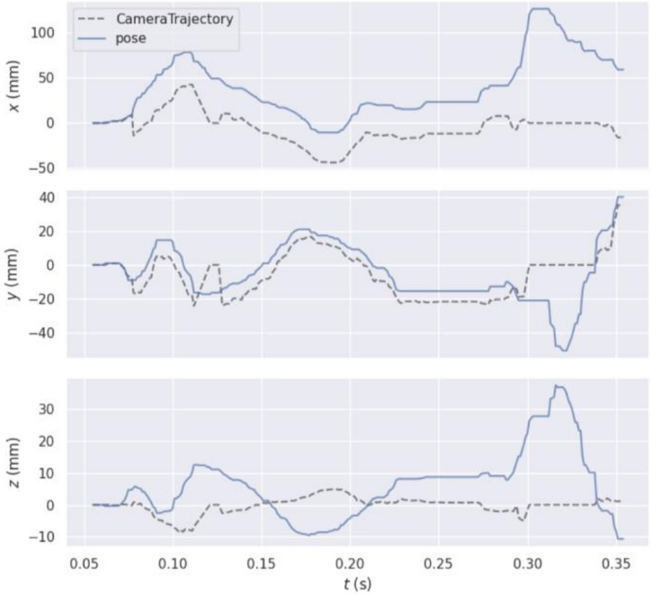
$F_x, F_y \leftarrow FragCoord.xy$ ▷ Input to the Shader
$B_0, B_1, B_2, B_3 \leftarrow Texture2D(F_x, F_y)$ ▷ Depth packed in 4 one-byte channels
$Z \leftarrow (B_3 \ll 24) \vee (B_2 \ll 16) \vee (B_1 \ll 8) \vee B_0$ ▷ Bit Shifting and Logical OR
$F_z \leftarrow Z/2^{24}$
$P_{norm} \leftarrow [F_x, F_y, F_z, 1.0]$
$P_{clip} \leftarrow (ProjectionMatrix)^{-1} P_{norm}$ ▷ Projection Matrix of Simulated Camera
$P_{cam} \leftarrow P_{clip}/P_{clip}.w$
$N_{xy} \leftarrow (P_{cam}.xy + \vec{MD}_{xy}/2.0)/\vec{MD}_{xy}$
$N_z \leftarrow (P_{cam}.z - n)/(f - n)$
$FragOutput \leftarrow [N_x, N_y, N_z, 1.0]$



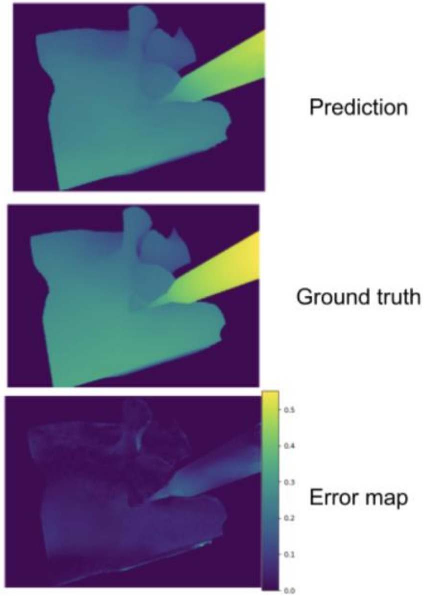
- OpenHMD Package to enable VR headset visualization and control of view

has been achieved to guarantee good simulation results on anatomy tracking and depth estimation

1st Technical Results



(a)



(b)

Table 1. Quantitative result (mean and standard deviation of L1 error) of ORB SLAM V3 applied on the synthetic stereo microscopic data generated by the drilling simulator.

	Translation Error (mm)	Rotation Error (deg)
Moving camera	40.97 ± 22.40	8.44 ± 3.07
Moving drill	8.1E-1 $\pm$ 9.1E-1	3.2E-3 $\pm$ 3.6E-3

1st Analysis

Pros Complete and Versatile

- Complete implementations with good documentations and guides to quick start.
- Open-Sourced code and library based, highly expandable.
- Simulation Environment alone has been tested solid in certain scenario and ready to use.

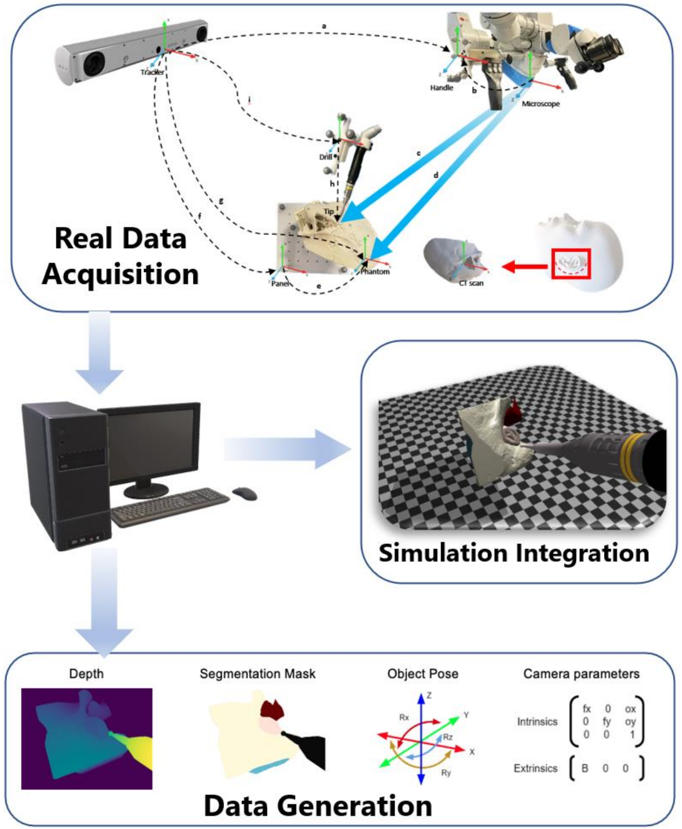
Cons Evaluation of Real

- Algorithm has not been tested in real environment to evaluate the significance of the simulation.
- Comparison of training results on surgeon performance has not been performed.
- Some of codes are hard-coded which requires minor fixes to adapt.
- Not clearly explained camera large error and failure in large motion.

1st Connection with us

- We are attempting to tackle some of these problems in this paper

1. By integrating Registered real world data stream into this framework or algorithm for evaluation.
2. By good calibration of microscope in real world to better analyze the possible causes for errors.



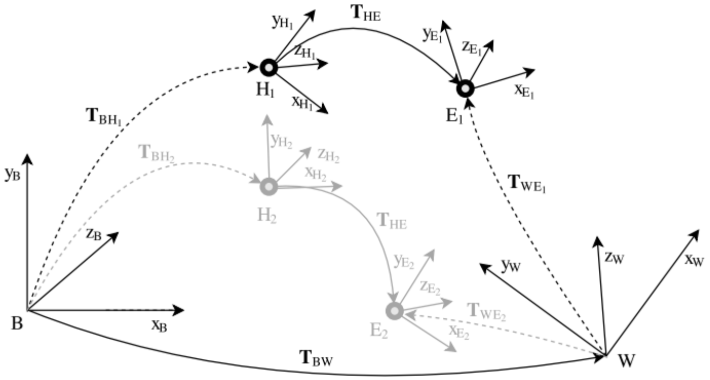
Paper 2: ***Evaluation of Combined Time-Offset Estimation and Hand-Eye Calibration on Robotic Datasets***

Fadri Furrer, Marius Fehr, Tonci Novkovic, Hannes Sommer, Igor Gilitschenski, and Roland Siegwart

- Optical Tracking
  - Tracker setup
  - Geometry define
  - Pivot calibration
  - Time alignment
  - Evaluate tracking accuracy
- Microscope Stereo Vision
  - Camera de-interlace
  - Stereo camera calibration
  - Stereo rectify
  - Microscope registration
  - Time alignment
  - Evaluate and refine calibration
- Hand-eye calibration
  - Setup AX=XB problem
  - Evaluate closed-loop error
- VR Environment
  - ADF File Generation using Blender
  - AMBF API Developme nt
  - Stereo Camera Setting
  - Synchroniz ation Test
  - Data Generation
- Sensitivity Analysis
  - Evaluate tracking accuracy
  - Evaluate Microscope calibration

2nd Related Work

- Solve rotation then translation
  - Use angle-axis representation
  - Use quaternions for rotations
  - …
- Solve rotation and translation simultaneously
  - Non-linear optimization method
  - SVD-based solution using dual-quaternion, based on screw-theory
  - SVD, using dual-quaternion, screw-theory, extended with RANSAC elimination of outliers



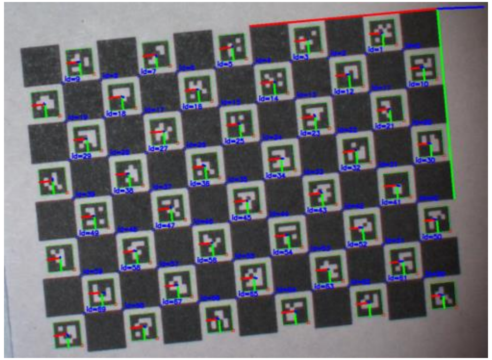
JOHNS HOPKINS WHITING SCHOOL of ENGINEERING

2nd Technical Details

- 1. Target extraction
  - Why:

    The poses of target are required to solve the AX=XB problem
  - Steps:
  - Detect corners on AprilTag
  - Find 3D coordinates of corners using RANSAC based PnP (Perspective-n-Point) method
  - Set a threshold λ_{th} to filter out frames with too many outliers
- This give us idea of using Charuco target to estimate poses from camera images.

AprilTag (umich.edu)

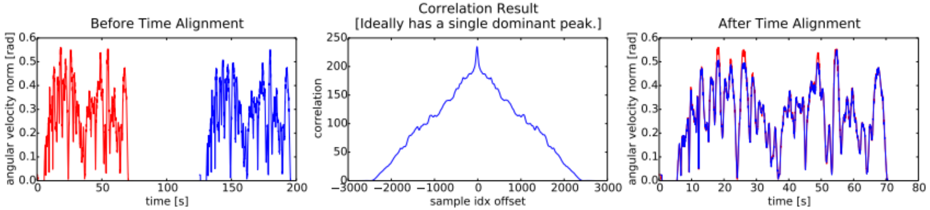


Corners detection on Charuco target

2nd Technical Details

- 2. Time alignment
  - Why:

    The sensors are not communicating, when calibrating a camera tracked by an external motion capture system
  - Steps:
  - Resample poses at lower frequency of two pose signals
  - Compute angular velocity norm
  - Time offset can be computed from the maximum value of the convoluted signal



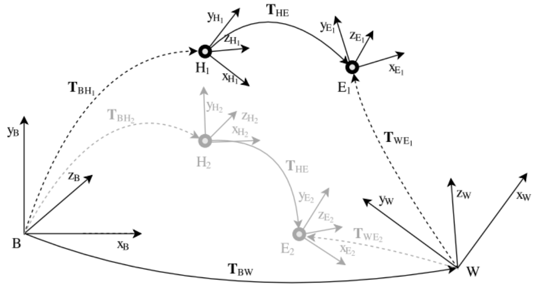
Time alignment result

2nd Technical Details

- 3. Hand-eye calibration
  - Steps:
  - Set up an AX=XB problem
  - Represent the rotation and translation with dual-quaternion
  - Filter out noise and outliers using screw-axis
    - Exclude pose pairs whose screw axis are almost parallel
  - RANSAC get initial hand-eye transformation
    - Randomly sample pose pairs, reject those whose scalar part of dual-quaternion are not equal
    - Iteratively choose the inliers agree with the hand-eye calibration
    - Repeat the hand-eye calibration on inliers
  - Refinement
    - Using the initial transformation above, perform a joint maximum likelihood optimization of calibration and trajectory

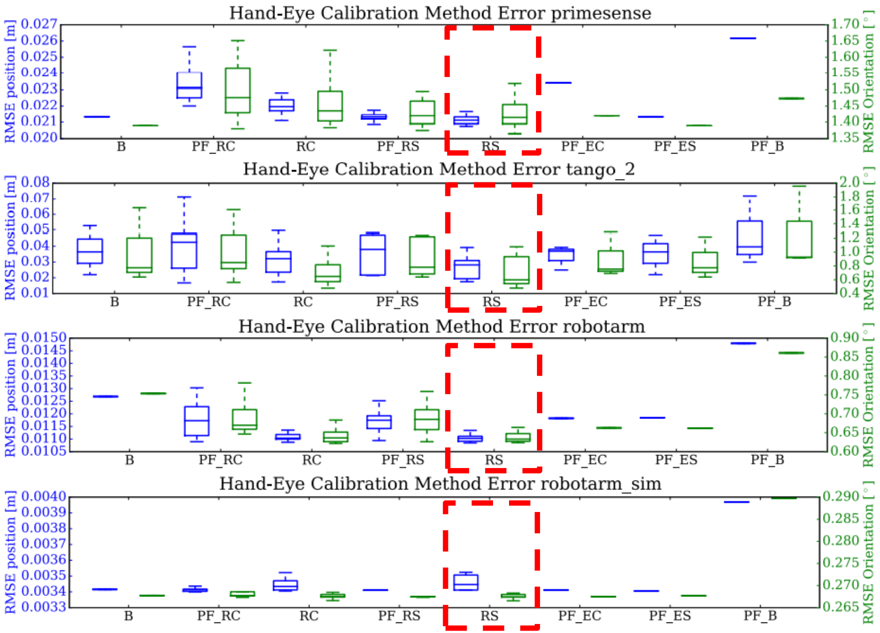
Algorithm 1: RANSAC based input pose pair selection for Eq. 1.

Data: A pair of vectors with time-aligned dual quaternions: $\mathbf{P}_{a,b} = [a, b]$
$a = [\check{\mathbf{q}}_{a,1} \cdots \check{\mathbf{q}}_{a,k}]^T$, $b = [\check{\mathbf{q}}_{b,1} \cdots \check{\mathbf{q}}_{b,k}]^T$, $RMSE_{best} = \infty$
Result: Static transform dual quaternion $\check{\mathbf{q}}_{a,b}$ and corresponding RMSE
Function HandEyeCalibrationRANSAC($\mathbf{P}_{a,b}$)
  $\mathbf{F}_{a,b} \leftarrow$ FilterPairs($\mathbf{P}_{a,b}$) // **PF**
  while *not reached probability of at least one inlier sample* **do**
    $\mathbf{S}_{a,b} \leftarrow$ SamplePairs($\mathbf{F}_{a,b}$)
    if *not* AllScalarPartsEqual($\mathbf{S}_{a,b}$) **then** **next** ;
    if *RC or EC* **then**
      $\check{\mathbf{q}}'_{a,b} \leftarrow$ ComputeHandEyeCalibration($\mathbf{S}_{a,b}$)
      $\mathbf{I}_{a,b} \leftarrow$ GetInliersBasedOnPoseError($\mathbf{F}_{a,b}$, $\check{\mathbf{q}}'_{a,b}$, $\lambda_{t,min}$, $\lambda_{r,min}$)
    else
      // **RS or ES**
      $\mathbf{I}_{a,b} \leftarrow$ GetInliersBasedOnScalarPartsEquality($\mathbf{S}_{a,b}$, $\mathbf{F}_{a,b}$)
    end
    if $|\mathbf{I}_{a,b}| <$ *required number of inliers* **then** **next** ;
    $\check{\mathbf{q}}'_{a,b} \leftarrow$ ComputeHandEyeCalibration($\mathbf{I}_{a,b}$)
    $(RMSE, \mathbf{I}_{a,b}) \leftarrow$ EvaluatePairs($\mathbf{P}_{a,b}$, $\check{\mathbf{q}}'_{a,b}$)
    if $RMSE < RMSE_{best}$ **then**
      $RMSE_{best} \leftarrow RMSE$
      $\check{\mathbf{q}}_{a,b} \leftarrow \check{\mathbf{q}}'_{a,b}$
    end
  end
  return ($RMSE_{best}$, $\check{\mathbf{q}}_{a,b}$)



2nd Results

- Evaluation of different filtering methods on the different datasets
- We choose the RS method which seems has the best performance overall.



2^{nd} Analysis

Pros

- A precise, applicable, and well-packaged hand-eye calibration solution
- Whole process optimization
  - Target extraction
  - Time alignment
  - Calibration algorithm
  - Refinement

Cons

- Hard to separate the four steps and use each one respectively
- Didn’t evaluate the generalization of the toolbox, performance could be limited with setup like ours (Very large focal length with very small calibration target and local motion poses)

References

- A. Munawar et al., "Virtual Reality for Synergistic Surgical Training and Data Generation," Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization, 1–9 (2021) [doi:10.1080/21681163.2021.1999331].
- F. Furrer et al., "Evaluation of Combined Time-Offset Estimation and Hand-Eye Calibration on Robotic Datasets," in Field and Service Robotics, M. Hutter and R. Siegwart, Eds., pp. 145–159, Springer International Publishing, Cham (2018)[doi:10.1007/978-3-319-67361-5_10].

JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING