TASK ORDER NUMBER: W81XWH-15-9-0001

MTEC RESEARCH PROJECT NUMBER: MTEC-20-05-IMPROVE-006

EGS NUMBER: MT20005.006

TITLE: Markerless Biomechanics to Investigate Performance

PRINCIPAL INVESTIGATOR: Dr. Daniel Nicolella

PERFORMING ORGANIZATION: Southwest Research Institute

CONTRACTING ORGANIZATION: Medical Technology Enterprise Consortium (MTEC)

REPORT DATE: 12/30/2022

TYPE OF REPORT: Final Report

PREPARED FOR: U.S. Army Medical Research and Development Command
Fort Detrick, Maryland 21702-5012

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE 12/30/2022	2. REPORT TYPE: Final Report	3. DATES COVERED 10/07/2020-03/31/2022

4. TITLE AND SUBTITLE

Markerless Biomechanics to Investigate Performance

5a. CONTRACT NUMBER W81XWH-15-9-0001

5b. GRANT NUMBER N/A

5c. PROGRAM ELEMENT NUMBER (Can be blank if don't

6. AUTHOR(S)

Travis Eliason

E-Mail:travis.eliason@swri.org

5d. PROJECT NUMBER MT20005.006

5e. TASK NUMBER W81XWH-15-9-0001

5f. WORK UNIT NUMBER N/A

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Southwest Research
Institute
6220 Culebra Rd
San Antonio, TX 78254

8. PERFORMING ORGANIZATION REPORT

[Enter RPA#]

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U.S. Army Medical Research and Development Command
Fort Detrick, Maryland 21702-5012

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S) N/A

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for Public Release, Distribution Unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Over the course of the program the base technology for an automated medical training feedback tool was created. This included a multi-stage hand tracking system which can track detailed subject hand kinematics from video without the need for any sensors or markers to be attached to the subject. A graphical user interface was created to incorporate this tracking pipeline along with the necessary tools for camera calibration as well as packaging for use by non-technical users. Several different approaches to automatically generate feedback were developed, and a test data set of medical students, residents, and attendings performing suturing was collected. Resulting algorithms were able to successfully identify suturing errors within the suturing data set automatically.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER *(include area code)*
Unclassified	Unclassified	Unclassified	Unclassified		

Standard Form 298 (Rev. 8-98)

MTEC Medical Technology Enterprise Consortium

TABLE OF CONTENTS

Final Technical Status Report for

Markerless Biomechanics to Investigate Performance Research Project No. 2018-652-003
EGS# MT20005.006
Reporting Period: 23 October, 2020 – 30 December, 2022

MTEC Research Project Awardee
Southwest Research Institute
Research Project Technical POC
Travis Eliason
6220 Culebra Rd
San Antonio, TX 78254
210-522-6926
travis.eliason@swri.org

Submitted: 30 December, 2022

MTEC
Medical Technology Enterprise Consortium

1. Project Status

a. Accomplishments

Tracking System Development

- Trained hand detector neural network to provide initial bounding box cropping of each hand
- Investigated several different underlying hand tracking neural networks to identify best performing combination
- Created a custom network based on High Resolution Networks (HRNet) which provided the best performance and flexibility for future improvements
- Sourced a variety of training data sets (ContactPose, CMU, HanCo) along with a variety of data augmentation techniques to provide sufficient data variety for network training
- Collected a validation dataset to quantify accuracy and reliability of hand tracking system
    - Synchronized video with Vicon ground truth measurements
    - 6 different camera configurations
- Processed collected data through subject specific biomechanical models
- Calculated median error and interclass correlation coefficient (ICC) for all trials and camera configurations
- Identified best performing camera configuration – High Front Sides
    - Median error < 1cm
    - ICC > 0.9
- Identified camera placement sensitivities in tracking system performance to help develop user guidance for best overall performance

Graphical User Interface (GUI)

- Created a graphical user interface to incorporate all developed algorithms and tools into a single application
- Built with flexible backend and submodule design to allow for easy feature additions in the future
- Loads multiple synchronized video files and allows simultaneous playback
- Intrinsic calibration of each camera incorporated using standard checkerboard target
- Extrinsic calibration calculated using object of known size in each camera view with keypoints selected by user and user defined measurements
- Incorporated hand tracking pipeline which includes hand detection, hand tracking, triangulation, and inverse kinematics
- Incorporated ability to filter both 2D and 3D joint locations
- Ability to overlay 2D tracking points and hand region of interests over original video
- 2D plotting of positional and kinematic results
- Initial implementation of 3D rendering engine

Feedback Algorithm Development

- Collected example data to use as testbed for development of feedback system
    - 7 Subjects
    - Motion requires grasping an instrument, along with spatial and temporal coordination of both hands

  - 5 expert takes per subject
  - Novice takes collected where subject intentionally performed motion incorrectly
- Developed initial temporal convolutional neural network to encode the motions and identify expert versus novice

Evaluating Expertness with Anomaly Detection

- Moved to an anomaly detection approach compared to previous classification network design
- Created a Long Short Term Memory (LSTM) network to predict future states from input kinematic data
  - Trained on expert data only
  - 260,000 individual 20-frame clips taken from 26 unique motions
- Reconstruction error metric allows for distinction between expert and novice on a clip by clip basis
  - Clear distinction of larger errors (sequence errors, speed differences, etc.)
  - Small differences difficult to detect (hand shakiness)

Multiple Instance Self Training (MIST)

- Developed a weakly supervised learning approach to more efficiently utilize datasets that do not contain clip level annotations
  - Automatically annotates individual clips to provide additional training data not available in previous anomaly detection approach
- Developed a temporal Convolutional Neural Network (CNN) autoencoder to extract features for the pseudo label generator
- Implemented a self-guided attention model
- Trained and evaluated 4 configurations of classification models
  - Kinematics versus 3D keypoint as input data
  - Pseudo labels generator versus self-guided attention network
- Generated receiver operating characteristics curves and associated area under the curve for all model configurations.

Suturing Data Collection

- Collected data on 13 subjects at UTHSCSA performing a suturing task
  - 5 cm simple running suture with instrument tie
  - Consistent DAISE training surrogate
  - Mix of medical students, surgical residents, and attendings
  - Collected video with 4 Sony RXO-II cameras
- Video Processing
  - All captured takes processed through hand tracking pipeline
    - Neural network to track hand key points
    - Inverse kinematics to fit biomechanics model
  - Inverse Kinematics -data was collated into expert and novice groups for training and evaluation of feedback algorithm

Suturing Anomaly Detection

- Data formatting
  - Normalize to starting position and orientation

- Normalize position and rotational data to consistent scale
- Convert to rotation matrices to eliminate Euler angle discontinuities

- Trained anomaly detection network on expert data to define exemplar model
  - Greater than 57,000 20-frame clips from 8 unique motions
- Evaluated performance of anomaly detection
  - Overall reconstruction error for withheld expert and novice takes
  - Individual takes to identify specific mistakes

b. Reportable Outcomes

No reportable outcomes for the current reporting period.

c. Progress Detail

Tracking System Development

In order to accomplish the overall project goal of developing a quantitative training feedback tool, an accurate, reliable, and robust hand tracking system was required. This system needs to be able to take calibrated and synchronized videos as input and track the kinematics of the subject's hands without the need for any external sensors or markers being placed on the subject. To accomplish this, a multi-step pipeline was developed which includes hand detection, hand tracking, triangulation, and inverse kinematics.

The first step of this process was to implement the hand detection stage which will draw a region of interest around each hand to crop the images prior to being based to the hand tracking network. Initially, an off the shelf network was trained to perform this task to see if we could leverage existing open source solutions to accelerate development progress. Through testing of the available options, it was identified that their performance was not sufficient to meet the accuracy and reliability goals of the project. They worked well in general but would have sporadic failures in certain background and hand orientations which was not acceptable for a general-purpose tool. To address this, we developed a custom network architecture and training protocol to build our own solution. This network was trained on a variety of publicly available datasets, along with one that specifically includes gloved hands which will ensure the hand detector will continue to work when latex gloves are worn. Additionally, training hyperparameters were optimized in an iterative approach to achieve the best possible performance.

With these custom improvements, performance of the hand detector was improved significantly compared to the off the shelf solutions to the point where it is now stable across a variety of environments and camera angles (Figure 1~~Figure 1~~). These stable regions of interest will provide a strong foundation for the overall hand tracking performance and give the hand tracking network the best chance to make accurate predictions.

Formatted: Font: (Default) +Headings (Calibri Light), 11 pt

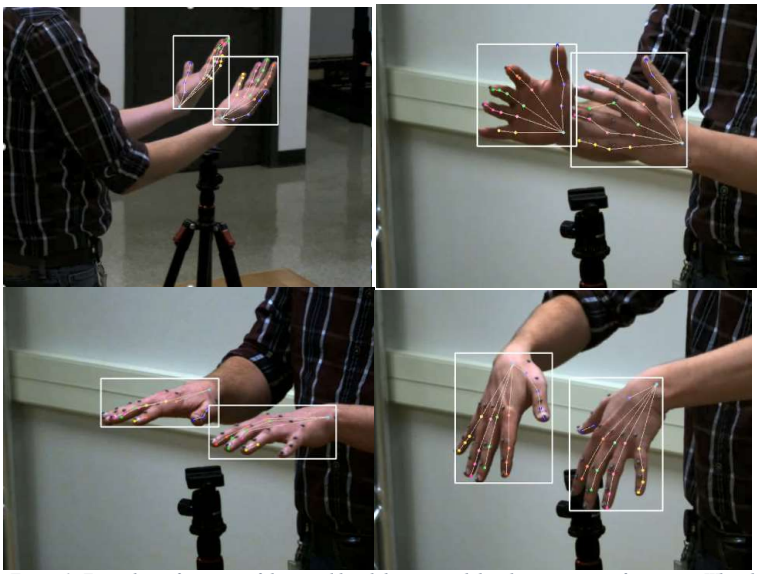

Figure 1: Example performance of the initial hand detector with hands in a variety of positions. White boxes drawn around each hand are the output of the hand detector that defines the regions of interest to be passed to the hand tracking network.

The next stage in the tracking pipeline is the hand tracking component which takes the cropped images from the hand detector and identifies 21 key locations on each hand. Similar to the hand detector we first evaluated several off the shelf hand tracking networks to identify potential open source solutions we might be able to leverage. The InterHand network had promise after initial testing showed good performance. However, further testing identified several limitations which eliminated its potential use. The most severe of these was the inconsistent performance of the network where it would perform reasonably well with some data and catastrophically fail with others. As more investigation was performed into that behavior it was identified that the network was highly sensitive to lighting, image resolution, and the relative scale of the hands. When these parameters matched well with the corresponding InterHand dataset it performed reasonably well, however when input video strayed the performance was hindered significantly (Figure 2~~Figure 2~~). This behavior is not acceptable for the medical training use case, as these variables will not be able to be controlled across the range of potential training environments it will be deployed in.

Formatted: Font: (Default) +Headings (Calibri Light), 11 pt

Formatted: Font: (Default) +Headings (Calibri Light), 11 pt

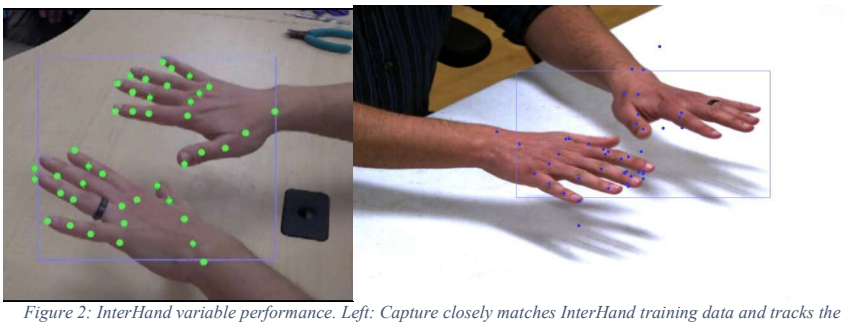

Figure 2: InterHand variable performance. Left: Capture closely matches InterHand training data and tracks the hand reasonably well. Right: With different camera and lighting performance degrades to an unusable state.

Given this information, two options were identified as potential paths forward. First, the InterHand network could be retrained with additional datasets containing more appearance variability to try and improve its overall robustness. This would require working around the training environment that the original authors had created and would need all additional training to be converted to their proprietary format. Additionally, as InterHand tracks both hands simultaneously training data would require both hands to be annotated, of which there is a limited amount compared to single hand training data sets.

Alternatively, a custom network could be designed and trained to track the hands, moving away from the InterHand architecture. Pursuing this path would give ultimate flexibility in design of the network and its integration within the overall software ecosystem developed for this project. The flexibility of creating a custom solution was ultimately chosen to get away from the limitations of InterHand, and to allow the use of a broader range of training data sets to create a robust solution that could work with a variety of cameras and environments. Rather than start from scratch, a previously developed network that was developed at SwRI for full-body pose estimation based on High Resolution Networks (HRNet) was used as a starting point. This network already has a robust ecosystem built around it for training and evaluating its performance which eliminated the necessity to generate these procedures from scratch. The first step to this adaptation was to modify the output layer of the network to switch from predicting the 47 full body keypoints to the 21 keypoints on a hand. Rather than try to track both hands simultaneously like InterHand this new network tracks each hand separately. This reduces the complexity of what the network needs to track, increases the number of datasets available for training, and gives greater flexibility when there may be an odd number of hands in the frame when performing collaborative tasks.

This network was then trained on a combination of publicly available datasets that include ContactPose, CMU, and HanCo which provide images of hands in a variety of poses, grasping objects, and at different scales. A subset of this data was held out as an evaluation data set so that a quantitative metric could be calculated to track performance. Current performance has achieved a percent correct keypoint, scaled by the index finger (PCKi) of 0.90 on the evaluation data set. This metric measures the percentage of predictions that match the ground truth annotations within a specified threshold where, 1 indicates perfect performance with all keypoints tracked correctly and 0 would indicate that the network did not get anything correct. A score of 0.90 indicates that overall the network is performing well, but the real test is how well the network performs in the wild data that does not match the datasets it was trained on. Qualitatively the current network is performing better than InterHand on unseen data (Figure 3~~Figure 3~~) and has shown to be more reliable across changing visual environments.

Formatted: Font: (Default) +Headings (Calibri Light), 11 pt

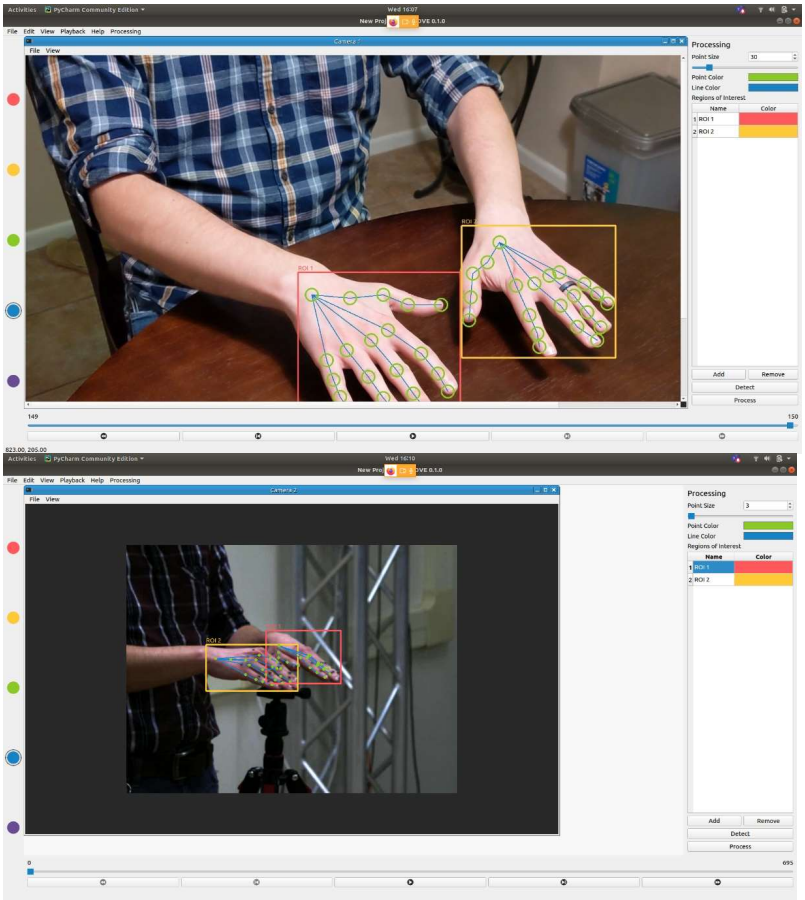

Figure 3: Example tracking performance on two sets of "in the wild" video data. Red and Yellow boxes are the tracked regions of interest for each hand. Green circles show the tracked keypoints on the hands with blue lines connecting corresponding segments.

To improve the overall robustness, even more background augmentation was implemented to extract more potential from the limited training data that was available. The HanCo dataset includes masks for the hands in each image which allows for the hand to be cut out of the image and then placed on a new background. By replacing the background, the same training image can be used multiple times within the training process, with a new background each time, to increase the generalizability of the network and make it more robust to background variety. A Poisson blending procedure, which allows the hand to be blended into the background (Figure 4) was used to ensure that the network did not learn to pick up artificial hard contrast lines that would be present with a simple cut and paste approach.

Figure 4: Example images where hands have been blended in with new background images. Rotation augmentation is also implemented where images are rotated prior to being input as training data such that the same hand pose can be used in multiple orientations to train the network.

While qualitatively the network was performing well, a quantitative evaluation was needed to understand the accuracy of the system as well as the variability of its performance across different camera configurations. This data is critical for instilling confidence in the measurements as well as providing best practice recommendation to ensure consistent performance across different locations and users.

To quantify the accuracy, data was collected on one subject sitting at a table performing three different motions (Flat Plane, 90 Degree Rotation, and Grasp). These motions were chosen to test various planes of hand motion as well as test occlusion that will occur when grasping an object (Error! Reference source not found.~~Figure 5~~). While the subject performed these motions, time-synchronized high-speed video data was captured from four video cameras, and from a marker-based motion capture system (Vicon) which was used to collect ground truth measurements. These motions were repeated with 6 different camera configurations (two positions x three heights) in order to investigate the relationship between camera setup (position and viewing angle) and hand tracking performance. Error! Reference source not found.~~Figure 6~~ shows the two camera positions (Corners, Front/Sides), each of which were tested at three heights: 1) table height (Low), 2) two feet above table height (Mid), and 3) four feet above table height (High). The global reference frame for the video cameras in each configuration were spatially calibrated to coincide with the Vicon global reference frame in order to facilitate comparing of the two systems.

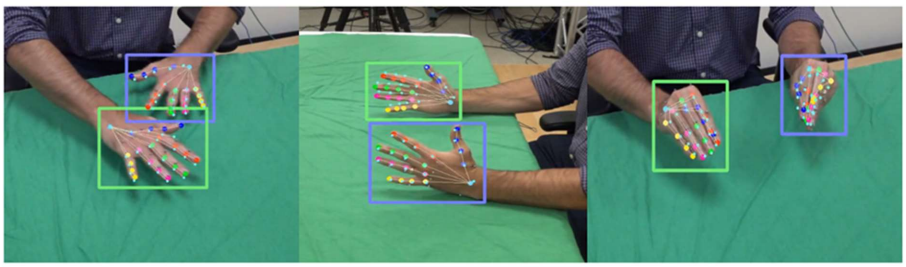

Figure 5: Visualization of the three movements performed by the subject. Left: Flat Plane, Middle: 90 Degree Rotation, Right: Grasp.

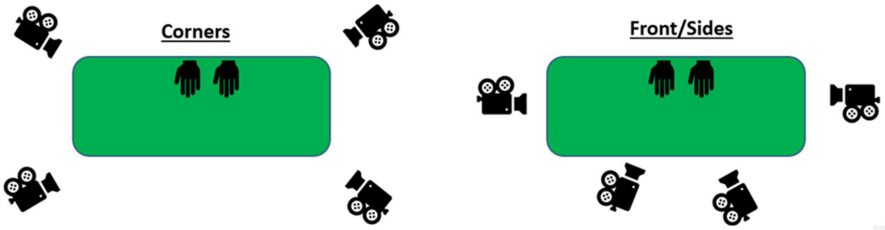

Figure 6: Visualization of the cameras positions relative to the subject.

The output from both the Vicon and hand tracking network are the 3D position time histories from 20 locations on each hand. These 3D locations from both systems were then used to drive a subject-specific inverse kinematics model (Error! Reference source not found.~~Figure 7~~). The locations of the joint centers from these models were extracted and used to calculate the median error and intraclass correlation coefficient (ICC) between the two systems which served as the quantitative metrics to evaluate overall tracking performance.

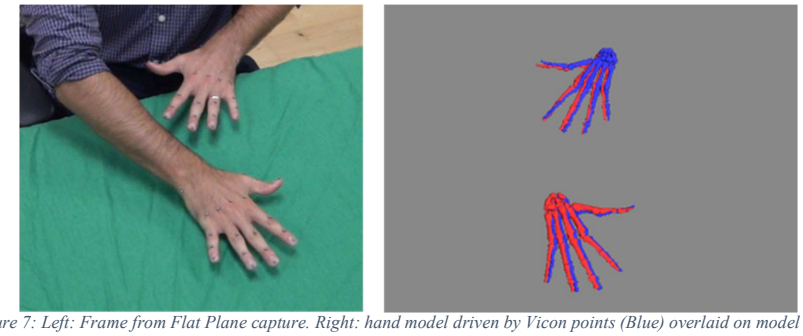

Figure 7: Left: Frame from Flat Plane capture. Right: hand model driven by Vicon points (Blue) overlaid on model driven by- the developed tracking system (Red).

Analysis of the six different camera configurations showed that three configurations (Mid Corners, Mid Front Sides, and High Front Sides) were consistently accurate across all motions exhibiting an average median error < 1 cm and an average ICC > 0.9 (Error! Reference source not found.~~Figure~~ 8). In general, Low camera configurations did not perform as well as Mid or High configurations. The Low camera configuration likely showed decreased performance for two reasons. First, these cameras had reduced depth information available to them because they were on the same plane as the hand. Secondly, Low cameras had more occlusions to overcome due to the viewing angle compared to the Mid or High configurations. Camera configurations in the front-sides positions tended to perform better than cameras in the corners position. However, the impact of the position was relatively minor compared to the differences between heights.

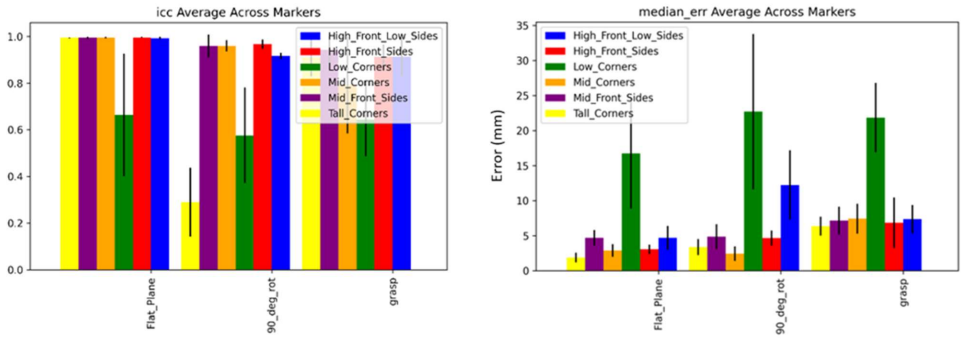

Figure 8: Average ICC and Median error across all 40 markers for the 6 camera configurations and 3 movements.

The most accurate configuration with respect to the Vicon ground truth was the High Front Sides configuration. Median error for each individual marker in this configuration is reported in Error! Reference source not found.~~Figure~~ 9 broken out by motion. In addition, examples of the X, Y, Z trajectories of the Vicon marker and SwRI predicted values are shown in Figure 10~~Figure 6~~. These markers were chosen to highlight the range of error that was observed. The r_thumb_tip exhibited the highest median error while the l_mcp4 had the lowest median error, both showing excellent agreement to the gold standard Vicon measurements.

Formatted: Font: (Default) +Headings (Calibri Light), 11 pt

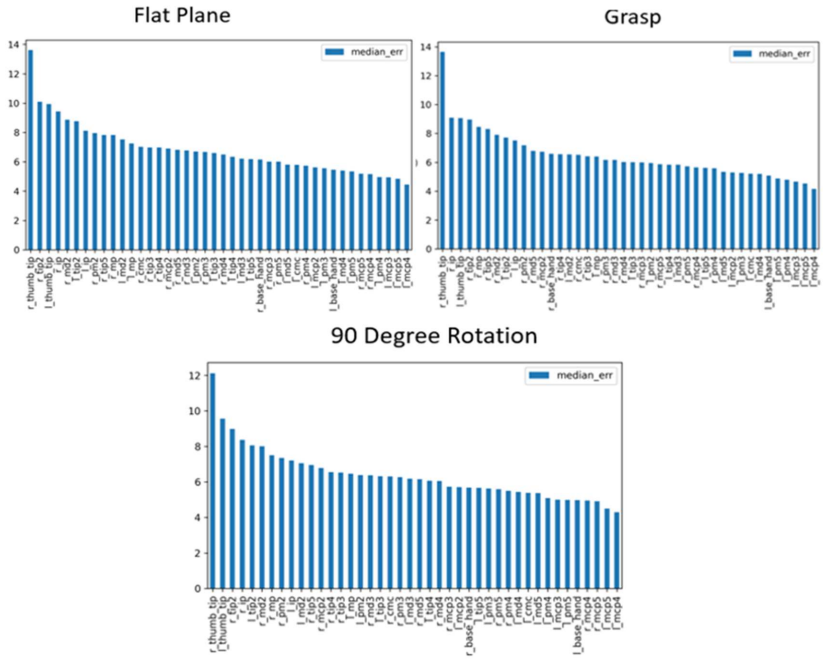

Figure 9: Median error for each marker for each motion in the High Front Sides configuration.

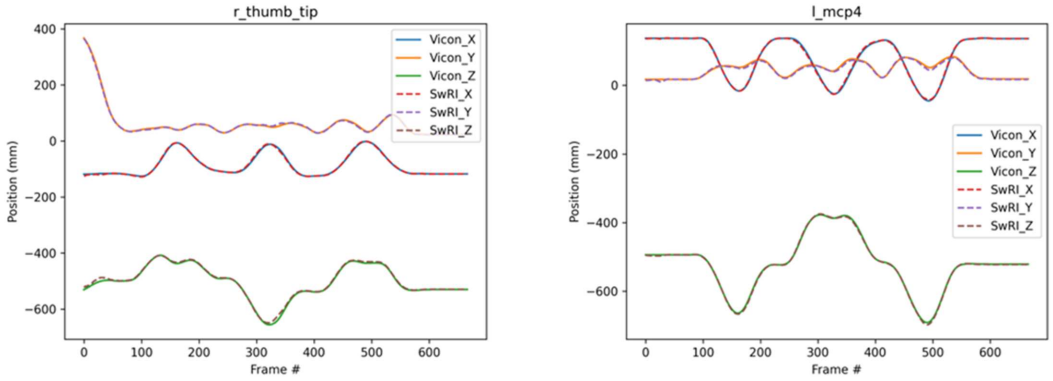

Figure 10: Vicon and SwRI trajectories for marker with highest median error (r_thumb_tip) and marker with lowest median error (l_mcp4).

With median errors of less than 1 cm and ICC greater than 0.9 the hand tracking system is now working at a sufficient level of accuracy to move forward with the feedback algorithm development. Knowledge gained from investigating the different camera configurations will be utilized when writing the user instructions for the delivered tool at the end of the program to ensure that users set up their cameras to achieve the best possible accuracy in their environment.

Graphical User Interface (GUI)

Development of the GUI was performed in parallel with the creation of the hand tracking system. This interface was created in order to package all of the various tools created to enable the hand tracking pipeline into a single tool that can be run by non-technical experts. Significant effort was put in at the beginning to design the software architecture to provide a robust back end while maintaining flexibility for future updates. This was done by setting up independent modes and submodules that work independently but share a backend to transfer data between them. This allows individual code bases from different tools that we have developed to be easily integrated without affecting already existing code. Similarly, it allows for easy integration of new features in the future as they can be written as new sub modules that can then be plugged into the existing GUI.

Along with integrating the hand tracking pipeline, the GUI provides additional functionality that is necessary for use as a standalone tool. The first of these is the ability to load in an arbitrary number of synchronized video files and control their play back simultaneously. This allows the user to view the raw video used as input to the tracking network and review the motion prior to processing. Each camera view is opened in a separate sub window (Figure 10) within the main viewing area and can be repositioned according to the user's needs. Several organizational presets (tiled, stacked, etc.) are available through keyboard shortcuts as a time saving convenience.

To use video as input to the hand tracking pipeline each camera needs to be calibrated for both its intrinsic and extrinsic parameters. The intrinsic calibration is performed first and requires the user to collect a trial in which they move a checkerboard of known size (Figure 10) around within the camera field of view. Using standard machine vision algorithms these checkerboard images are used to calculate the parameters of the cameras lens to correct for any distortion and calculate the focal length. In order to provide feedback to the user on the quality of the calibration, after the calculation is complete, the tracked points on the checkerboard are displayed overlaid on the original videos and

are color coded based on its quality. Green signifies “very good” (Figure 10), yellow means “passable”, and red if the calibration is very poor and needs to be repeated.

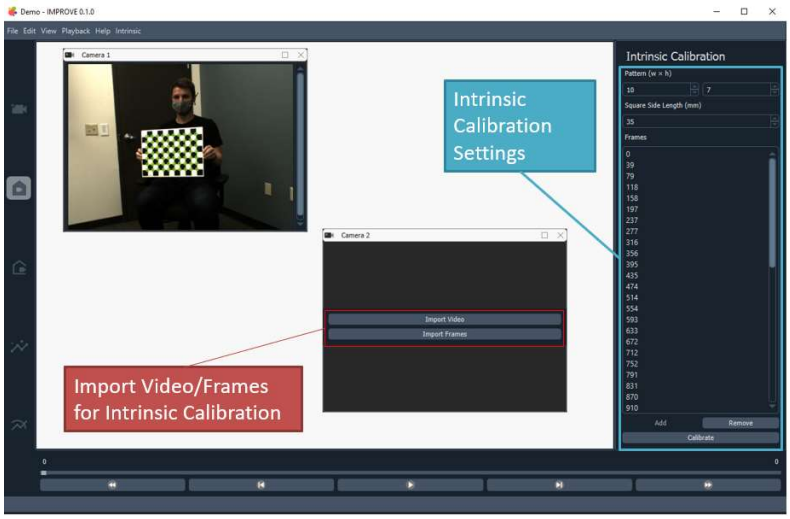

Figure 11: Multiple camera sub windows show for individual camera feeds. Intrinsic calibration with green dots displayed on top of the checkerboard indicating that a successful calibration was performed.

Following the intrinsic calibration, the user is guided to the extrinsic calibration mode within the GUI in order to spatially calibrate the cameras relative to each other as well as set the origin of the tracking coordinate system. In order to perform this calibration, the user collects video of a static object of known dimensions upon which there are distinct key locations that can be seen in each camera view. The user then inputs the dimensions of the object in the right-hand pane (Figure 11) which will be used in the extrinsic calculation. The key locations on the object are then manually identified in each camera view by clicking in the video window which sets their 2D locations for each camera. Each point is color coded to assist with visual verification that points are consistently placed in each camera view. Additional tools are available to change the size of the colored circles as well as zoom in to help with accurate placement. Once all points are accurately located, the 2D locations along with the input dimensions are used to back calculate the relative 3D locations of each camera relative to a consistent coordinate system. This information is used to enable accurate triangulation of the tracked hand keypoints which are needed to drive the biomechanical model. Once calculated, a coordinate system triad is displayed overlaid on each video as an additional visual verification that the calibration has been performed successfully.

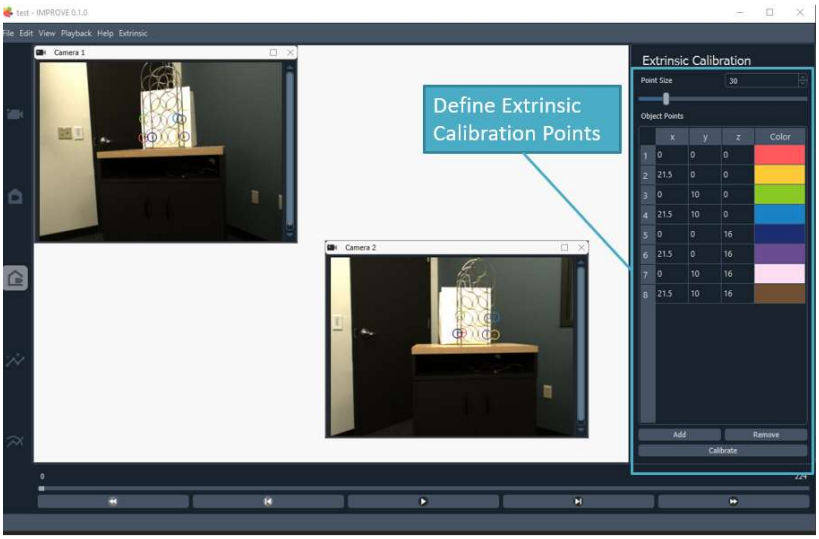

Figure 12: Extrinsic calibration with different colors used for each calibration point.

After calibration is complete the user is then directed to the processing mode which is used to process the input videos through the hand tracking pipeline. The first step is to run the hand detection network which identifies the regions of interest around each hand within the videos. Results are displayed as boxes overlaid over the original videos which are color coded (Figure 12). These colors allow for quick visual verification that the hand detection network is accurately identifying the right and left hand correctly in each camera view and is not flipping them. Following verification of the detection, the ROIs are processed through the tracking network to identify the 21 key locations on each hand. Similarly, the results of this network are overlaid on the original video (Figure 12) for visual assessment of its performance.

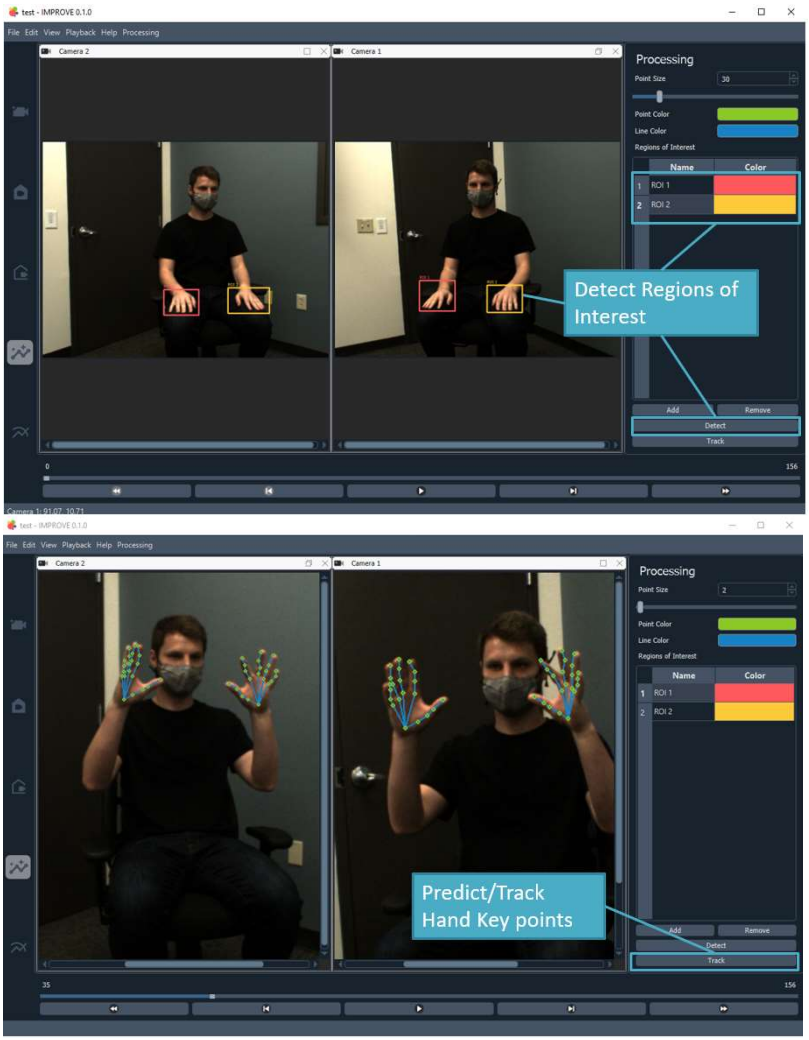

Figure 13: Top: Processing step showing defined regions of interest for each hand (red and yellow boxes). Bottom: Hand tracking results overlaid as green circles for each hand keypoint .

The results mode contains the remainder of the tools necessary to complete the full hand tracking pipeline following processing through the hand detection and hand tracking networks. This includes triangulation, which uses the camera calibration and 2D keypoint locations from each camera to calculate the 3D location of each hand keypoint. These 3D locations are then fed to the OpenSim API linked with the GUI which is used to scale a biomechanical model to the individual and then perform an inverse kinematics analysis. This analysis converts the raw 3D locations into the subject specific kinematic parameters that will be used as input to the feedback algorithms. Additionally, there are plotting capabilities to generate plots of any of the keypoint locations or any kinematic parameter from the OpenSIm model. These plots have a moving cursor that is synchronized with the video

playback to assist with plot interpretation (Figure 14~~Figure 14~~). Lastly, there are filters available that can be applied to either the 3D tracking information or the resulting inverse kinematics results.

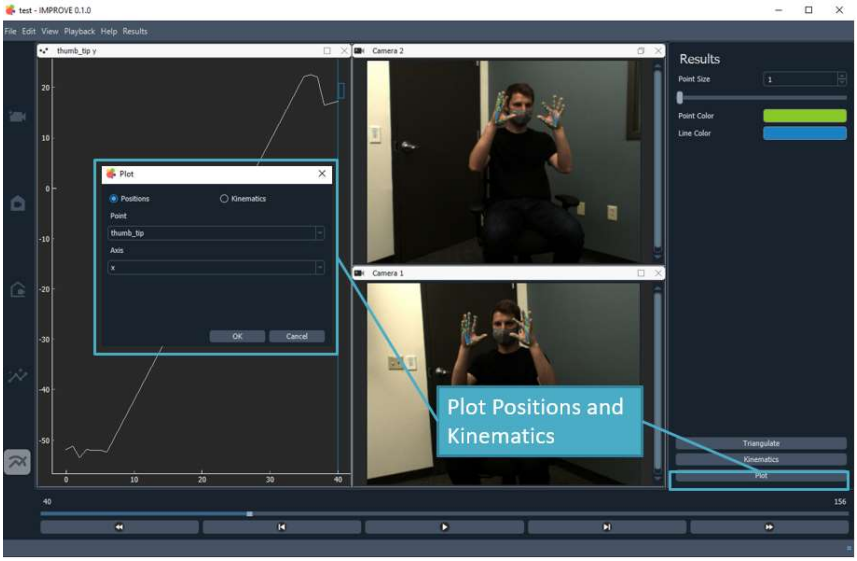

Figure 14: Results screen showing buttons for triangulation, inverse kinematics as well as plot generation. Plot window allows user to select the degrees of freedom they wish to plot, and multiple plots can be generated each in their own independent sub window.

Feedback Algorithm Development

With the tracking system reaching the desired level of accuracy, work began on developing the feedback algorithm that will be utilized to identify the differences between expert and novice practitioners. In order to develop this feedback system, a test data set that included motion data from both experts and novices was required. To collect this, the SwRI-developed hand tracking system was utilized to measure the movements from seven subjects as they performed an example motion. This motion included tracing a line on a piece of paper with a pair of tweezers, while simultaneously touching indicated points with their other hand. This motion was developed as a simple task that can be learned easily, while challenging the tracking and feedback system in similar ways that will be used on medical tasks. Using tweezers requires tracking of the hand while grasping and utilizing an instrument while the overall motion requires both spatial and temporal coordination between both hands (Figure 15~~Figure 15~~).

Formatted: Font: (Default) +Headings (Calibri Light), 11 pt

Formatted: Font: (Default) +Headings (Calibri Light), 11 pt

Formatted: Font: (Default) +Headings (Calibri Light), 11 pt, Not Italic

Figure 15~~15~~: Subject performing example task with inverse kinematics model from SwRI hand tracking system overlaid on video.

Each subject performed the motion correctly five times, which served as our expert samples to train the feedback system on what is correct. Additionally, each subject performed the motion incorrectly in specific ways that included both spatial errors (moving the tweezers too far, with shaky hand, and picking up tweezers between points), coordination errors (moving out of sequence, right/left hand out of order), temporal errors (moving too fast or too slow), and switching hands to perform the tracing with left instead of right hand. These motions were used as novice trials to test if the feedback system could identify differences between expert and novice and where the system may not be sensitive enough.

Building on previous work, SwRI's first approach centered on the application of a temporal convolutional neural network (Table 1~~Table 1~~) operating with the kinematics representations as inputs. The neural network design applies a series of convolution and pooling layers to distill each motion into a feature representation designed to distinguish between novice and expert motions. As each motion is similar to an image, in that it contains an overall structure (the bulk motion) and layers of texture (ranging from fine, repeated movements to general fluidity of motion), temporal CNNs are a logical choice. Additionally, our use of a fully-convolutional temporal neural network affords our solution some degree of invariance to when each motion's recording is started or stopped.

Formatted: Font: (Default) +Headings (Calibri Light), 11 pt

Table 1: Initial temporal-CNN design. The network processes a batch of size N, using the 52-dimensional kinematic parameters, sampled at 128 time steps, as the input. The entire motion is distilled into a two-dimensional representation which is trained to encode quality of task execution.

Network Operations	Input	Output
1x3 convolution, ReLU, Pool	N x 52 x 1 x 128	N x 64 x 1 x 64
1x3 convolution, ReLU, Pool	N x 64 x 1 x 64	N x 64 x 1 x 32
1x3 convolution, ReLU, Pool	N x 64 x 1 x 32	N x 64 x 1 x 16
1x3 convolution, ReLU	N x 64 x 1 x 16	N x 64 x 1 x 16
1x1 convolution, Global Pool	N x 64 x 1 x 16	N x 2 x 1 x 1

Though approachable as a two-task classification problem (expert class versus novice class), we apply a triplet-loss based optimization, with the goal of a future embedding tasked with distinguishing

multiple motions as well as ascertaining proficiency. To extend our dataset, we apply a series of data augmentations, including positional and temporal shifts. For validation, we withhold a subset (15 of 88 total samples) of collected motions. Using the 73 motions in the training set, we run a neural network training procedure that cycles through each sample, providing the sample, a sample from a matching class, and a sample from a non-matching class. Repeating this process for 200 iterations over the dataset and optimizing our weights using the Adam solver.

Preliminary results are shown visually in Figure 16~~Figure 16~~, which shows that the validation points are generally mapping to the expected cluster, though certain points in the validation set are ambiguous to the neural network. In this two-dimensional embedding, it is also clear that components 1 and 2 are not independent, though this is expected in the current 2-class problem. Inspection of the validation points that did not map as expected is helpful for further design. In particular, we notice that many of the vague points consist of motions with proper right-hand movement, with errors introduced intentionally only in the left hand. We also observe difficulty in ascertaining a novice class that involved picking up the instrument between points, moving in the correct horizontal plane pattern while moving vertically in arcs rather than in straight lines. While these results showed initial promise in differentiating between groups, greater specificity would be needed for the overall feedback system.

Formatted: Font: (Default) +Headings (Calibri Light), 11 pt

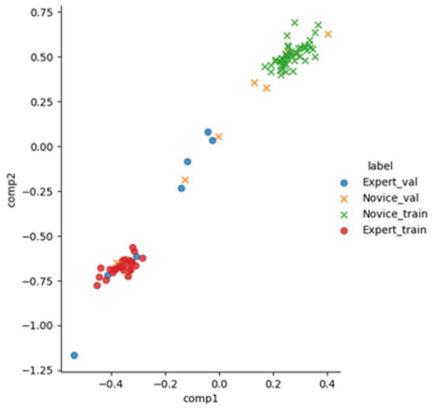

Figure 16: Initial results from the temporal CNN encoding of novice vs expert. Novice trials show as X's and expert shown as circles. As can be seen, both groups are generally clustering together allowing for the discernment of expert versus novice.

Evaluating Expertness with Anomaly Detection

In order to address the shortcomings of the first temporal CNN, a new approach, anomaly detection, was evaluated. Anomaly detection is a machine learning problem for examining data and detecting irregularities and outliers. Most often, this problem is approached with algorithms that are considered one-class classifiers. These classifiers are designed to learn on a single class of data: the nominal (anomaly-free) data. By mapping the distribution of nominal data, new samples can be evaluated for conformity based on a distance metric.

In deep learning, anomaly detection is generally conducted via a construct called an autoencoder. Autoencoders are neural networks that perform an encoding and decoding of incoming data samples, reducing dimensionality into a latent representation (the encoding) and reconstructing the input (via the decoder). At training time, the weights of the autoencoder are refined with the objective of

minimizing the reconstruction error (the difference between the input and output) on the nominal data. At inference time, the model's reconstruction error can be used as a metric for how anomalous a given sample appears. Samples which fit the domain on which the model has been trained will have lower errors, while more disparate samples will yield higher errors. In the evaluation of temporal sequences, autoencoding can also benefit from temporal shifts between the input and output, forcing an internal representation that must reconstruct current frames using information from the past (this amounts to asking the model to predict the future from the past series of observations). This paradigm forces the model to maintain a latent representation of the current state of a task and to predict the regular flow of data that should proceed from that point.

To create a method for examining performance and providing feedback, we apply the concept of anomaly detection to kinematic data derived from our markerless motion capture system. We model the evolution of the kinematic state in time using a LSTM recurrent neural network, predicting the future kinematic state from the encoded latent state. We train the model using only the expert, exemplar data, withholding some exemplar motions and all novice motions for testing.

The model is designed to work using the kinematic representations of motions (e.g., via a biomechanical model) that are extracted using our markerless motion capture system. The kinematic representation is highly advantageous compared to processing raw videos, as it completely removes appearance variations from the incoming data (e.g. background, camera pose, lighting, and hand appearance). Our model reduces the dimensionality of the kinematic representation via a fully connected neural network layer, and then applies this reduced representation to an LSTM module which evolves a hidden state. From the hidden state, the model decodes a prediction for the future movements of the hand. The model architecture is summarized in Table 2~~Table 2~~, where nT represents the length of the time series and nK represents the number of kinematic parameters (in the case of our model nK = 52):

Formatted: Font: (Default) +Headings (Calibri Light), 11 pt

Table 2: Anomaly Detection Network Structure

Operation	Output Size
Batch Normalization	1 x nK x nT
Fully-connected Layer	1 x 32 x nT
LSTM Module	1 x 32 x nT
Fully-connected Layer	1 x nK x nT

We trained the model using the majority of the expert samples, applying the Adam solver and iterating over 260,000 20-frame examples extracted from 26 unique motions.

Anomaly Detection Results

We evaluate our results by examining the distribution of reconstruction errors on withheld data (all novice motions and 7 withheld expert motions). These error distributions are shown in Figure 17~~Figure 17~~.

Formatted: Font: (Default) +Headings (Calibri Light), 11 pt

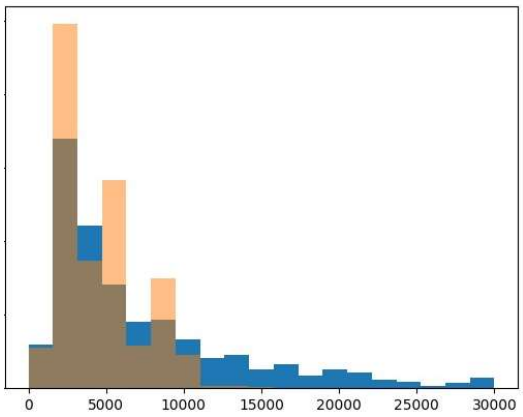

Figure 17: Distribution of reconstruction errors (sum squared errors over all kinematics) of withheld expert (tan) and withheld novice (blue) motions.

In general, it is apparent that the reconstruction errors of novice examples can be fairly low; however, the error distribution of the novice examples contains larger errors. It is the occurrence of these errors that suggests that specific times and actions of novices that deviate from expert behavior can be identified.

Looking at individual motions, our observations are that the method is generally effective, though some errors are more discernible than others. Motions that are performed with excessive speed or reversed hands have consistently high errors and are easily identified. Smaller deviations, such as the hand shaking but preserving a generally correct path, are subtler and sometimes indistinguishable from expert motions under this approach.

Overall, the method is promising. The deep-learning representation can accommodate complex distributions and thoroughly utilize available data without requiring extensive data curation or labeling, and without requiring novice data for training. The reconstruction error can also be decomposed per trial, in time, or within specific joint angles. The potential pitfall of the method is that the magnitude centric representation may not be able to quantify subtle differences, something that a supervised method (one requiring labeled examples of anomalies) could potentially accomplish given enough data. Another potential pitfall is that the exemplar data provided for training must necessarily encompass valid variations of motion in order to avoid penalizing valid alternatives.

Multiple Instance Self Training (MIST)

While the results from the anomaly detection approach are encouraging, the unsupervised framework limits the ability to identify subtle errors. Moving to a supervised approach could provide additional benefit, but obtaining the required clip-level annotations of errors would be time consuming and difficult to accomplish compared to the overall video-based labels. To address this issue, a new approach is needed that can use the high-level labels while retaining the ability to identify small errors within the data.

Leveraging a similar approach to Feng et al [1], we explored whether weakly-supervised learning techniques can allow us to extract as much value from our data as possible. It consists of two major stages. In the first stage, we train a model (G) to estimate clip-level labels using the available video-level annotations. Next, we train a subsequent model using these clip-level pseudo-labels. This network leverages a self-guided attention module that causes the network to direct its attention towards notable

features in the input data. Figure 18~~Figure 18~~ describes the method used to sample the data and train our pseudo-label generator. The algorithm is as follows. First, a pair of videos is selected with one containing a novice and one containing an expert. These videos are each uniformly sampled L times to form a bag with each sample containing T consecutive clips. Then, each clip is processed using a pre-trained feature extraction network that takes input kinematic data generated by the biomechanical model and extracts useful features. Once they have been processed, the pseudo-label generator network generates clip-level scores for each clip on a scale of 0 to 1 where 0 indicates the clip contained an expert movement. These scores are aggregated across the entire bag and a margin ranking loss is used to incentivize the model to generate different scores for expert and novice clips.

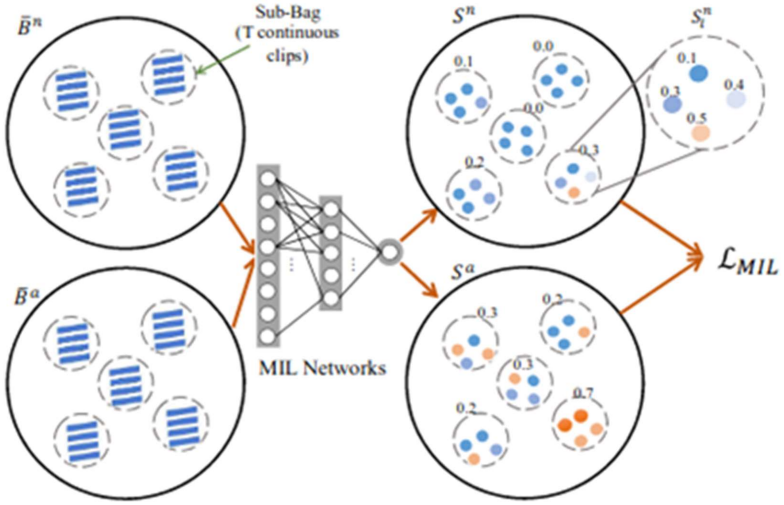

Figure 18: Pseudo-Label Generator Training Strategy

Once the pseudo-label generator is trained, it can be used to generate pseudo-labels that provide an expert versus novice score for each clip in a given video. These pseudo-labels may then be used to train any number of models rather than using the original video-level annotations.

As mentioned previously, this approach requires a pre-trained feature extractor that has been trained on a related task. This network generates useful features that the pseudo-label generator and self-guided attention network will use to determine whether a clip contains a novice or expert motion. Due to the specificity of this work, an off-the-shelf feature extractor was not available and we needed to construct our own. Leveraging fully-convolutional temporal neural networks, we created an auto-encoder that distilled a given clip into a set of features and tried to recreate the original clip from these features. This architecture allows the network to be trained using pre-existing motion data while still providing the feature extracting capabilities that we needed. Table 3~~Table 3~~ provides an overview of the temporal CNN autoencoder. This network was trained using the motion data captured previously to provide feature extraction for the rest of the training framework.

Formatted: Font: (Default) +Headings (Calibri Light), 11 pt

Formatted: Font: (Default) +Headings (Calibri Light), 11 pt

Table 3: Temporal Autoencoder Architecture

Temporal Autoencoder Architecture	
Encoder	1x3 convolution, ReLU, Pool
	1x3 convolution, ReLU, Pool
	1x1 convolution, Global Pool
Decoder	1x3 convolution, ReLU, Upsample
	1x3 convolution, ReLU, Upsample
	1x1 convolution

To quantify the performance of the MIST approach, and identify the ideal configuration, a series of 4 different classification models were trained. Performance of each model was evaluated by plotting their respective receiver operating characteristics (ROC) curves and calculating the associated area under the curves (AUC). Two different data types were used as input to the models to identify which may be more sensitive to identifying differences between novices and experts. Kinematic measures which describe the kinematic state of the biomechanical model were used as one data type, which was compared to using 3D keypoint positions. Both data types fully describe the captured hand motion, but encode that motion in different ways. Similarly, we investigated using the output of the pseudo-label generator directly input into the classification model compared to also using the trained self-guided attention network.

Results using the kinematic data as input showed results that were like the simpler autoencoding approach that was originally implemented (Figure 19~~Figure 19~~). AUC measures for both the pseudo-label generator and self-guided attention network were both 0.704 providing decent performance, but not to the level necessary to reliably differentiate novices and experts. Adding the self-guided network did not improve the performance over the pseudo-label generator, which indicates that there was a lack of data to fully take advantage of the weakly supervised approach. This could be caused by the relatively small dataset we currently have which makes extrapolation of these results difficult.

Formatted: Font: (Default) +Headings (Calibri Light), 11 pt

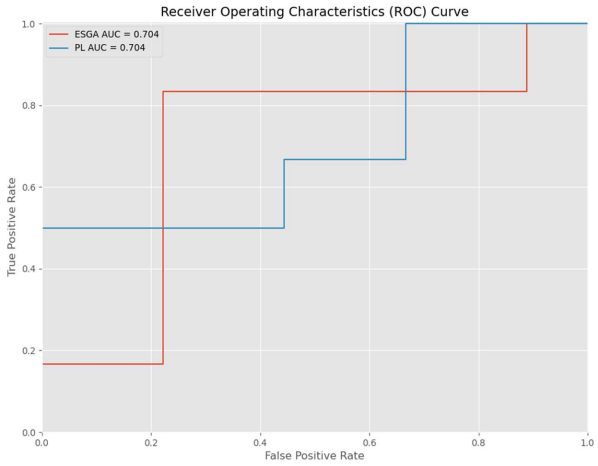

Figure 19: ROC curves for the self-guided attention network (red) and pseudo-label generator (blue) inputs using the biomechanical model kinematics data as input. AUC for both = 0.704

Compared to the kinematics input, using the 3D keypoint data showed significantly improved predictive performance for both model types (Figure 20~~Figure 20~~). AUC for the pseudo-label generator was 0.853 compared to 0.704 when using the kinematics data, an improvement of nearly 15%. Unlike previously, the addition of the self-guided attention network significantly increased the predictive performance over using just the pseudo-label generator, producing a perfect AUC of 1 on the validation data set. An AUC of 1 means that the model was able to perfectly classify all takes within the validation data set as expert or novice with no false positives or negatives. This is an incredible encouraging result, but no definitive conclusions can be made yet as there are still limitations with the test dataset. Once the performance can be replicated in a larger more representative dataset we can be confident that this approach will meet the overall project goals.

Formatted: Font: (Default) +Headings (Calibri Light), 11 pt

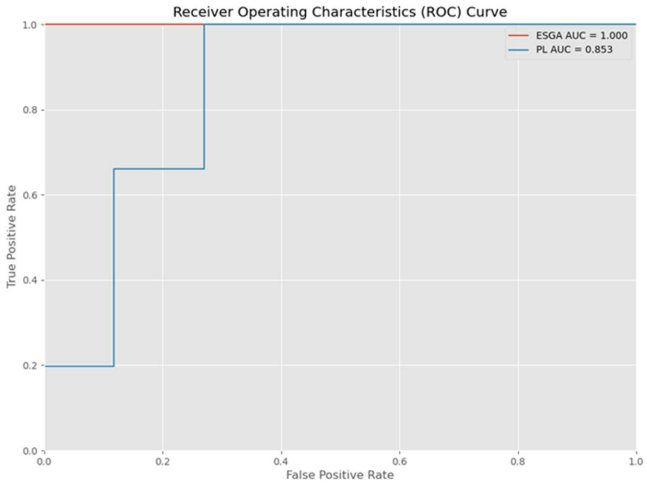

Figure 20: ROC curves for the self-guided attention network (red) and pseudo-label generator (blue) inputs using the 3D key points as input. AUC for the pseudo-label generator = 0.853 and 1.0 for the self-guided attention network.

[1] Feng, J.-C., Hong, F.-T., & Zheng, W.-S. (2021). MIST: Multiple Instance Self-Training Framework for Video Anomaly Detection. arXiv [cs.CV]. Opgehaal van http://arxiv.org/abs/2104.01633

Suturing Data Collection

On 22 August 2022, data collection was performed in collaboration with the surgical residency program at University of Texas Health Science Center San Antonio (UTHSCSA). A mixture of subjects including medical students, surgical residents, and attendings were recruited to perform a suturing task while being filmed. All subjects used a consistent DAISE training (Figure 21~~Figure 21~~) surrogate model to perform a 5 cm simple running suture with an instrument tie. Data was collected with four Sony RXO-II video cameras set up both in front and to the sides of the table. Figure 22~~Figure 22~~ shows still frames captured from one of the trials showing the four camera views collected. In total, 13 different subjects were recorded with 30 unique takes to provide both expert training data and novice training data for evaluation.

Formatted: Font: (Default) +Headings (Calibri Light), 11 pt

Formatted: Font: (Default) +Headings (Calibri Light), 11 pt

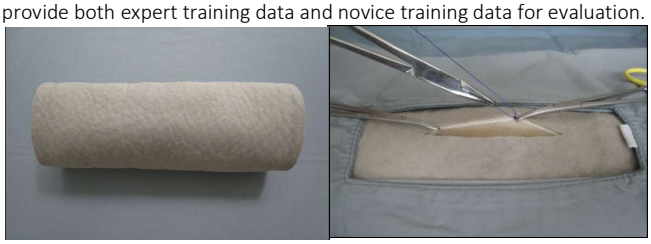

Figure 21: DASIE reusable surrogate model for suturing.

Figure 22: Camera angles used for the suturing data collection.

Following capture, all the video data was processed through the previously developed hand tracking system. This process included: calibrating the cameras, initial hand detection to crop video around each hand, 2D hand key point detections, triangulation, and an inverse kinematics fit of a biomechanical model. The result of this tracking pipeline is a biomechanical model with the associated kinematic representation of the captured motion (Figure 23~~Figure 23~~) which is used as the input for the feedback algorithm.

Formatted: Font: (Default) +Headings (Calibri Light), 11 pt

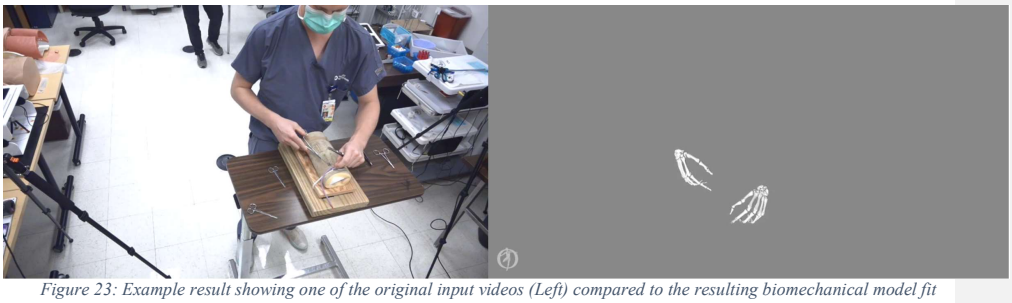

Figure 23: Example result showing one of the original input videos (Left) compared to the resulting biomechanical model fit of the same time point (Right).

Suturing Anomaly Detection

Following the capture of the suturing data the previously described anomaly detection approach was applied to the new data set. Because our analysis is based on movement in space, we utilize several important preprocessing steps for encoding the movements into a format that is ready for a machine learning analysis. First, we align all captures by applying a rotation about the vertical axis to ensure a consistent coordinate frame, ensuring that the positive x-axis is aligned with the subject's right, the y-axis is aligned with the direction the subject is facing, and the z-axis is facing up. In addition to this alignment of direction, we align the origins of different captures by taking the center location as the median location of the subject's right hand. To prevent discontinuities in the Euler angle representations, we encode all wrist rotations as rotation matrices. After these steps, we scale all joint angle kinematic parameters, dividing by 90 degrees. The result of the preprocessing is a 64-dimensional kinematic representation for

each time this is consistent in orientation, continuous in its representation of angles, and equalized across dimensions (the magnitude of each value being generally constrained to a maximum of 1). The entirety of a capture is then represented by an array of shape nT x 64, where nT is the number of frames. We trained the model using most of the expert samples, applying the Adam solver and iterating over greater than 57,000 20-frame examples extracted from 8 unique motions.

Results

We evaluate our results by examining the distribution of reconstruction errors on withheld data (all novice motions and 2 withheld expert motions). These error distributions are shown in Figure 24~~Figure 24~~.

Formatted: Font: (Default) +Headings (Calibri Light), 11 pt

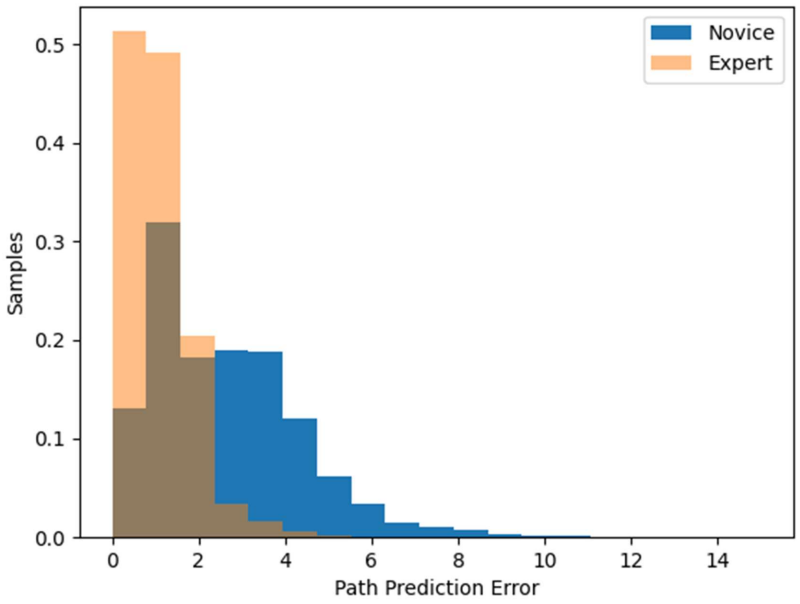

Figure 24: Normalize histograms of path prediction error for Novice and Expert motions. Against the withheld data, the model accurately predicted expert motions, while novice motions were much more likely to show higher path prediction errors.

In general, it is apparent that the reconstruction errors of novice examples can be fairly low; however, the error distribution of the novice examples contains larger errors. It is the occurrence of these errors that suggests that specific times and actions of novices that deviate from expert behavior can be identified (Figure 25~~Figure 25~~). Based on this histogram, when evaluating individual takes in order to identify errors a path prediction error of greater than 4 was used as a threshold. As a test, several takes which exhibited errors greater than 4 were qualitatively evaluated to see what errors the network was able to identify.

Formatted: Font: (Default) +Headings (Calibri Light), 11 pt

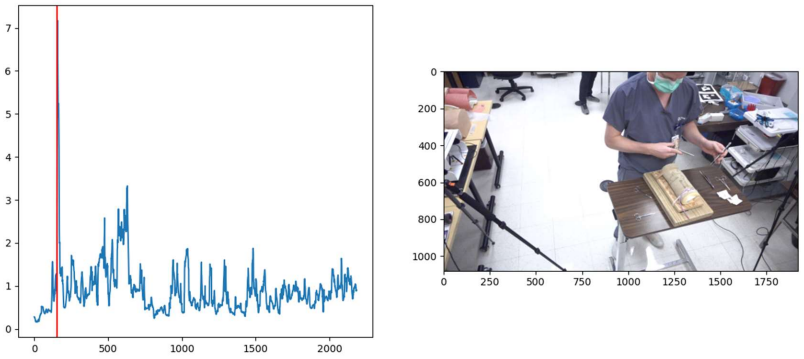

Figure 25: Example reconstruction error (Left) for one of the expert takes which contained a single path prediction error spike above 4.

Figure 26~~Figure 26~~ shows one such example where one of the left out expert takes was evaluated and a single spike above the error threshold was observed. When reviewing the video, the error that was being detected was not immediately obvious to a non-expert observer. After careful comparison to other expert takes, what had transpired was that instead of passing the needle to the tweezers after pushing through, the subject re-grasped with the needle driver. He then recognized his error and reset to tie the anchor for the first stitch and continued the rest of the task without issue.

Formatted: Font: (Default) +Headings (Calibri Light), 11 pt

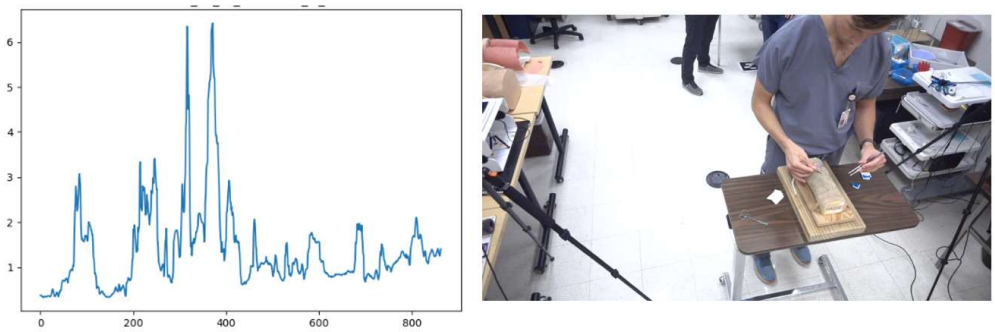

Figure 26: Example reconstruction error (Left) for one of the novice takes which contained errors above the threshold.

Similarly, Figure 26~~Figure 26~~ shows an example from one of the captured novice takes which also had errors above our threshold. The error for this example was more obvious upon review in which, while pushing the needle through, the needle driver slipped off the needle and then needed to be re-grasped. While these are only two examples from the many takes that were captured they highlight the ability of the system to identify fairly obvious errors (slipping off the needle) as well as more subtle ones (pulling through with wrong instrument). Not only can it identify errors in a single hand's motion but also in the coordination of both hands working together to accomplish the task showing the effectiveness of the underlying technology which can be used to create an automated training feedback tool.

Formatted: Font: (Default) +Headings (Calibri Light), 11 pt

2. Future Plans

Throughout the first phase of this program, we have developed a hand tracking pipeline which is able to accurately track the detailed kinematics of a subject's hands while performing a suturing task. This data is then input into an anomaly detection network to automatically identify time points within the motion in which the subject deviated from how an expert would perform the motion. This technology stack will serve as a strong foundation that can be built upon to expand this work to provide feedback for other tasks as well as other potential uses such as performance evaluations, and skills sustainment assessments.

We have proposed the following general tasks for a phase 2 of the program to continue improving the underlying technology as well as expand capabilities and make it easier to use by non-technical experts. The first of these tasks would be a series of improvements to the underlying tracking pipeline to ensure the most reliable data that will be input to the assessment algorithms. This would include building out a more robust training data set, optimizing the processing pipeline to increase speed, and expanding to include a temporal component rather than evaluated every frame independently. Additionally, we have proposed to expand beyond tracking just the hands so that more information about the motion can be captured and used as input. Upper body tracking (head, torso, arms) will be added such that more complete information about the subject's posture and approach can be directly measured rather than being inferred from the hand position. Similarly, instrument tracking will be included so that the instruments can also be tracked as they are often what is being used to directly interact with the patient. Combined with the improved hand tracking these improvements should provide a more robust tracking system that provides a comprehensive data set for the assessment algorithm.

Along with the tracking system performance improvements, additional tasks were proposed to improve the overall usability of the system for non-technical users. Using the current GUI that was developed in phase 1 as a base, camera control will be added such that data can be captured and processed all from a single program without the need to move data files around manually. Improvements will also be made to the data visualization interfaces to allow for more intuitive interpretation of the results and automated feedback generation. Lastly, the current feedback algorithm will be extended for use in performing skills sustainment assessments. This would include training the exemplar model on a single individual and then tracking them over time to see if their performance degrades over time. As exemplar models will need to be trained for every individual, this training process will be automated and integrated into the GUI. We will partner with UTHSCSA again to capture longitudinal data on a number of different medical skills to assess the performance of the system.

3. Financial Health

A no cost time extension was requested an approved during the program to ensure enough time was available to collect the suturing data set. Phase 2 tasks have been proposed along with cost estimates for additional funding to expand on accomplishments from phase 1.

4. Personnel Effort

Provide names of current staff along with their roles and percent effort of each on this project. Add additional rows if necessary to list the complete team. If there is more than one project on this award, breakdown according to each project (one table per project).

Personnel	Role	Percent Effort
Travis Eliason	Project Manager	50%
Omar Medjaouri	Neural Network Developer	50%
Koen Flores	GUI Developer	50%
Ty Templin	Biomechanical Model Developer	25%
Dan Nicolella	Primary Investigator	5%

5. Protocol and Activity Status

a. Human Use Regulatory Protocols

TOTAL PROTOCOLS: 1 human subject research protocol required for this program.

PROTOCOLS:

Protocol [HRPO Assigned Number]: E02028.1a
Title: Markerless Biomechanics to Investigate Performance
Target required for clinical significance:
Target approved for clinical significance:
Submitted to and Approved by:
Integreview – Approved
HRPO – Approved

STATUS: Recruited and collected data on 13 participants

b. Use of Human Cadavers for Research, Development, Test and Evaluation (RDT&E), Education or Training

No cadavers to be used during the course of the program.

c. Animal Use Regulatory Protocols

No animals to be used during the course of the program.

Annual Business Status Report for

Markerless Biomechanics to Investigate Performance Research Project No. 2018-652-003
EGS# MT20005.006
Reporting Period: 23 October, 2020 – 30 December, 2022

MTEC Research Project Awardee
Southwest Research Institute
Research Project Technical POC
Travis Eliason
6220 Culebra Rd
San Antonio, TX 78254
210-522-6926
travis.eliason@swri.org

Submitted: 30 December, 2022

MTEC
Medical Technology Enterprise Consortium

1. Current Staff

Personnel	*% of Effort on project*
Travis Eliason	50%
Omar Medjaouri	50%
Koen Flores	50%
Ty Templin	25%
Dan Nicolella	5%

2. Current Expenditures

A. Cost Reimbursable Contracts: Complete only if your contract is Cost Reimbursable or Cost Plus Fixed Fee.

Expenditures should be reflective of cost incurred to date, not exceeding awarded project ceiling. Expenditures should coincide with the latest invoice for the reporting period. For cost reimbursable contracts please use the table below.

Contract Expenditures	Current QTR Expenditures	*Cumulative To Date Expenditures*
Labor (Personnel and Fringe)	$1,132.83	$396,279.67
Supplies/Materials	$	$
Travel	$	$
Equipment	$0	$30,803.76
Subcontractors and Consultants	$	$
Other Direct Costs	$	$
Indirect Costs	$1,790.83	$647,872.57
Total	**$2,923.66**	**$1,074,956.00**

3. Status of Milestones– FILL OUT FOR ALL CONTRACT TYPES (all project milestones are to be included)

All project milestones from the Milestone Payment Schedule, in the project award, should be accounted for below.

Milestone	SOW Task	Significant Event/Accomplishments/Deliverable	Due Date	Proposed Program Funds	Percent Work Completed	Project Funds Expended
1		Project Start	9/30/2020	$0.00	100%	
2	1.2.2	HRPO IRB application submission	10/7/2020	$0.00	100%	
3		Quarterly Report 1 (Sept.)	10/25/2020	$0.00	100%	
4	1.1	Initial review current simulation-based training programs and evaluation systems	1/8/2021	$10,500.00	100%	$10,500
5		Quarterly Report 2 (Oct.-Dec.)	1/25/2021	$0.00	100%	
6	1.2.2	HRPO IRB Approval	4/8/2021	$0.00	100%	
7		Quarterly Report 3 (Jan. – March)	4/25/2021	$0.00	100%	
8	1.2.2	Exemplar Data Capture and Processing	4/25/2021	$91,000.00	100%	$91,000.00
9	1.2.1	Video-based 3D Markerless Kinematics Measurement System	6/25/2021	$193,000.00	100%	$193,000
10		Annual Report 1	7/25/2021	$0.00	100%	

11	1.2.3	Development of Training Feedback System	8/31/2021	$127,000.00	100%	$127,000.00
12	1.2.4	Development of GUI-based Training Evaluation and Feedback System	9/30/2021	$261,000.00	100%	$261,000.00
13		Quarterly Report 4 (July-Sept.)	10/25/2021	$0.00	100%	
14	1.3	Human Subject Testing #1	5/31/2022	$20,000.00	100%	$20,000.00
15		Quarterly Report 5 (Oct. – Dec.)	1/25/2022	$0.00	100%	
16	1.4	Generalized Tasks Demonstration	5/31/2022	$12,000.00	100%	$12,000.00
17	1.5	Coordinated Activity	7/29/2022	$180,000.00	100%	$180,000.00
18	1.6	Human Subject Testing #2	8/31/2022	$25,000.00	100%	$25,000.00
19	1.1	Comprehensive review current simulation-based training programs and evaluation systems	3/1/2022	$0.00	0%	
20		Final Report	12/30/2022	$155,456.00	100%	$155,456.00
21		Quarterly Report 6 (Jan-March)	4/25/2022	$0.00	100%	
22		Annual Report 2	7/25/2022	$0.00	100%	
23		Quarterly Report 7 (July-Sept)	10/25/2022	$0.00	100%	
		Total		$1,074,956.00		$1,074,956.00

4. Deviation from Project Plan

5. Switching from the original full body tracking to focusing on the hands presented additional challenges which took longer than originally planned. A no cost time extension was requested and approved to provide additional time to collect the suturing data set.