Hand Gesture Recognition

Samuel Oncken, Steven Claypool

EXECUTION AND VALIDATION PLAN

REVISION - 2
3 October 2022

Change Record

Rev.	Date	Originator	Approvals	Description
1	10/3/2022	Samuel Oncken Steven Claypool		Draft Release for FSR
2	11/25/2022	Samuel Oncken Steven Claypool		Validation Alterations for Final Presentation/Report

VALIDATION PLAN

Table 1: 403 Validation Plan

Test Name	Success Criteria	Methodology	Status	Responsible Engineer(s)
Benchmark Dataset Training	Gesture recognition neural network can run on our home computer and train using real dataset. Results quantified	Download the benchmark dataset and code for the CNN. Run the code and confirm similar accuracy to benchmark logs provided	TESTED – Pass	Steven Claypool
Virtual Dataset Training	Gesture recognition neural network can train using our built dataset and provide accuracy results	Take a final virtual dataset modeled after a real benchmark dataset and use it to train the same CNN as the benchmark. Obtain and compare recognition accuracy to real dataset training run.	TESTED - Pass	Steven Claypool
Gesture Recognition Accuracy	Accuracy is within 5% of benchmark accuracy using real dataset	Train gesture recognition neural network using real and synthetic sets and compare accuracy, testing virtual on virtual, real on real	TESTED - Pass	Steven Claypool
Synthetic Train on Real Test accuracy	Test real data with CNN trained on synthetic data and achieve accuracy similar or improved to benchmark train results	Train a CNN using different methods/compositions with synthetic and/or real data and test using real data	IN PROGRESS	Steven Claypool
Unity Hand Mapping	Real hand movement from LMC is mapped using Ultraleap prefab hand models in Unity	Set up Unity environment, install Ultraleap plug-ins, familiarize with Ultraleap scripts/prefabs, read Ultraleap Unity API documentation, and map hands.	TESTED - Pass	Samuel Oncken
Import a Rigged MakeHuman Model	A fully rigged MakeHuman model is imported into Unity with specific materials/textures extracted	Configure MakeHuman model with “Default” rig and desired appearance. Export model as .fbx file and import model into Unity. Make sure all bones are aligned properly and can be manipulated through space. Approve that model mesh (appearance) is as desired.	TESTED - Pass	Steven Claypool
Virtual Model Unity Hand Mapping	Map hand movement onto an imported rigged MakeHuman model with natural looking movements	Import a rigged MakeHuman model, access the wrist bone and add a hand binder component. Fix advanced issues such as bone orientation and mapping accuracy. Make sure user movement is accurate to movement of the model (without unnatural joint movements).	TESTED – Pass	Samuel Oncken
Mounting Stability	Head mounted LMC remains intact with mounting apparatus	Apply LMC to the mounting apparatus and plug the device into the computer. Rotate head in all directions (looking up, down, left,	TESTED - Pass	Samuel Oncken

	even if head motion is present	right) and shake head left to right. Make sure LMC is still firmly in place.		
Gesture Recording	From within the Unity environment, gesture animation clips can be produced and stored for application on future models	From within the scene used to validate the Unity hand mapping, download the Unity Recorder from the package manager in Unity. Choose output file name and destination, exporting as an animation clip, recorded from the lower arm bone of the virtual model	TESTED – Pass	Samuel Oncken
Apply Example Animation to Model	MakeHuman model is able to perform an imported full body gesture accurately.	Visit Mixamo.com, download a .fbx animation file from the choices listed. Apply the animation to the rigged human model using an Animator component to validate that the rigged structure is able to move as intended.	TESTED - Pass	Samuel Oncken
Apply Recorded Gesture Animation to Model	Freshly imported rigged MakeHuman model is able to perform the gesture animation as recorded in the Gesture Data Collection subsystem	After recording a gesture animation, apply it to a freshly imported MakeHuman model using the Animator component located in the model's lower arm bone	TESTED - Pass	Samuel Oncken
Create and Import Rigged MakeHuman Models to Unity	Minimum 30 MakeHuman models are generated uniquely and imported to Unity environment	Use MakeHuman “mass produce” function to generate unique character models, each fit with a “Default” rig. Produce around 20% of models with edge/corner case metrics (gloves, unnatural colors, etc.)	TESTED - Pass	Steven Claypool
Data Capture Output and File Type	Virtual camera outputs image data as .jpg file type	After images or videos are taken of a model performing a gesture, validate that the images are stored, organized, and are of the desired file type.	TESTED – Pass	Samuel Oncken
Final System Validation	With the press of a button, a large and diverse virtual training set is produced	Run system and validate in output files that each gesture has at least 500 images of gesture performance on differing human models from numerous camera angles	TESTED – Pass	Samuel Oncken

Execution Plan

Table 2: 403 Execution Plan Status Indicator Legend

Complete	
In progress	
Planned	
Behind Schedule	

Table 3: 403 Execution Plan

Task	Deadline	Responsibility	Status
Develop research outcomes/goals	9/7/2022	All	
Get familiar with LMC, Unity, and MakeHuman	9/14/22	All	
Write ConOps	9/15/22	All	
Determine optimal mounting position of LMC with testing of tracking accuracy	9/21/22	Samuel Oncken	
Generate 4 test MakeHuman models, instantiate them into Unity	9/21/22	Steven Claypool	
Map movement of hands onto Ultraleap hand prefabs in Unity	9/28/22	Samuel Oncken	
Add rig settings and animator component to humans in Unity, test online example animation on models	9/28/22	Samuel Oncken	
Set virtual cameras in Unity to face human model upon spawn	9/28/22	Steven Claypool	
Write FSR, ICD, Execution and Validation plans	10/3/2022	All	
Map real hand movements onto virtual human model and record a gesture animation	10/5/2022	Samuel Oncken	
Research Image Synthesis for Gesture Recording process	10/5/2022	Steven Claypool	
Find at least 2 real gesture datasets, tested against SOTA gesture recognition neural networks for us to replicate	10/5/2022	Steven Claypool	
Midterm Presentation	10/12/2022	All	
Record all gesture animations for the replicated virtual gesture datasets	10/12/2022	Samuel Oncken	
Build Unity environment with multiple virtual cameras angled at imported model hand	10/12/2022	Samuel Oncken	

Apply recorded animation to a virtual model	10/19/2022	Samuel Oncekn	
Generate human models using mass produce and export each as a .fbx file to import into Unity	10/19/2022	Steven Claypool	
Find gesture recognition neural network for each dataset and test on machine	10/26/2022	Steven Claypool	
Status Update Presentation	10/27/2022	All	
Complete Unity environment; spawn unique model, apply gesture, record image and store into dataset specific folders	11/4/2022	Samuel Oncken	
Set up Unity environment on WEB 156 lab computer – generate virtual datasets for sign language for numbers and alphabet	11/7/2022	All	
Preprocess training sets and begin training the gesture recognition neural networks using virtual datasets	11/11/2022	Steven Claypool	
Test various compositions of training set data. Combinations of real/synthetic data for NN training, test on more real data	11/18/2022	Steven Claypool	
Train using various compositions of real/synthetic data and test on pictures of our own hands	11/25/2022	Steven Claypool	
Explore future datasets for replication next semester (full body animations/full body images) ex: HaGRID	11/25/2022	All	
Final Demo/Presentation	12/1/2022	All	
Complete Final Paper	12/4/2022	All	

Note: The training/testing process using our virtual data has been much more time consuming and difficult than expected by our sponsor, hence the training being behind schedule. Steven plans to continue training and testing into winter break to explore the most optimal use of our virtual training sets when it comes to gesture recognition accuracy.

Performance on Execution Plan

At the beginning of this semester, we developed an ambitious execution plan with the help of our graduate student sponsor that moved at a much faster pace than we were able to follow. Thus, once again with the help of our sponsor, we had to make alterations to the plan throughout the course of this semester and include work into winter break. We were successful in completing the creation of a Unity environment capable of generating virtual hand gesture training sets for hand gesture recognition neural networks, though it did take longer than expected originally. As a result, training set testing was pushed back and is still in progress today, which is planned to be completed over winter break.

Performance on Validation Plan

Each subsystem within our Unity environment has been validated following our previously shown plan and has passed each test, proving that they are each functioning as we designed. The entire dataset generation system (with all subsystems merged) has also been validated and can replicate real benchmark datasets by capturing images of hand gestures being applied to randomly spawned, unique models under randomized lighting and background conditions and storing each into dataset specific folders sorted by gesture. Validation of the dataset's ability to properly train a hand gesture recognition neural network through testing with real data is still in progress, but we have been able to validate that the virtual data can be used in the training of hand gesture recognition neural networks and achieve >99% accuracy when testing with other virtual data.