Miniaturization and geometric optimization of SteamVR Active Optical Trackers

Sean Gow, Jared Macshane, Ali Ahmadinia, Kambiz Hamadani, Yuanyuan Jiang

***Abstract*—Active tracking enables higher precision in tracking the positions, orientations, and states of the virtualized objects. STEAMVR Lighthouse tracking base-stations can be used for tracking specific objects. However, current solutions are bulky and costly. The overall goal of this research work was to reduce the size and cost of active VR trackers to enable their attachment to ever smaller physical tools and objects to be tracked in the real world and displayed in a virtual reality environment.**

***Index terms*—** virtual reality, active optical tracking

I. Introduction

The current generation of SteamVR active trackers [1] are as large and sometimes larger than the objects to be tracked. This reduces the overall authenticity and immersion of the user experience in virtual and mixed reality environments. The size and bulk of previous generations of SteamVR hardware development kits (HDKs) has been one of the limiting factors in the further development and application of SteamVR tracking for diverse and cost-limited applications such as education. The recent release of a new SteamVR compatible HDK by Tundra-Labs [2] enables miniaturization and customization of the SteamVR electronic components making it possible to further reduce the size and cost of such active tracking components. Here we explored various SteamVR HDKs and created customized active trackers for hands-on virtual reality and mixed-reality science laboratory experiences. We present results that highlight best practices as well as some limitations of this technology for future applications.

II. Current Tracking Technologies and Limitations

There are several methods for positional and orientation tracking including acoustic, radio frequency, magnetic, optical, inertial sensor, and/or a combination of these approaches [3]. For a realistic immersion experience, accuracy and latency are significant factors. Optical and inertial methods and their combination have been the most widely used method in VR industry.

Optical tracking technologies are typically categorized based on the direction of tracking: inside-out and outside-in. When the tracking camera is on the Head-Mount Display, it is called inside-out tracking. When the HMD is tracked by cameras in an external environment, it is called outside-in tracking.

In the following, we discuss the tracking systems, based on the role markers: marker-based and marker-less tracking. For marker-based tracking, there are two types of visible markers: passive and active. Passive markers reflect infrared light (IR) towards the light source. In this case, the camera provides the IR signal that is reflected from the markers for detection. Active markers are IR lights that flash periodically and are detected by the cameras.

A. Passive Optical Tracking Systems

Passive optical methods track an object by placing stereoscopic imaging cameras in the periphery of a tracking volume where they detect light reflected off tracked objects in the center of the tracking volume. Such systems can be sub-classified into those which employ marker-based tracking methods and those which employ marker-less tracking methods.

- Marker based tracking systems such as Optitrack use infrared cameras to track reflective markers in predefined configurations and positions to calculate the position and orientation of tracked objects. Limitations include requiring unobstructed line of sight of tracking markers, and expensive cameras.
- Marker-Less tracking systems use depth sensing cameras and deep learning with inferential data sets of trained objects used to identify objects and movements within a scene. Limitations are line of sight, additional programming, and training of data sets for object identification, and computationally intensive processes that may increase latency and require more powerful and expensive processors.

B. Active Tracking Systems

Active tracking systems attempt to track an object's movement using battery-powered sensors that are mounted onto the tracked objects. Examples of such systems include SteamVR's Lighthouse tracking system, and other active tracking systems such as Antilatency's tracking system.

- The Lighthouse Tracking system is the technology behind SteamVR. Lighthouse is a laser-based inside-out positional tracking system developed by Valve for SteamVR and HTC Vive. This system uses base stations [4] which sweep the room with infrared light, the trackers have numerous IR sensors which are activated by the infrared light and are then used to calculate the tracker's position and orientation and merged with additional data from an IMU (Inertial Measurement Unit). Limitations include a minimum of 5 sensors and unobstructed line of sight to a single base station to initiate tracking, and an obstructed view of 4 sensors to continue tracking. Additional limitations on the maximum number of trackers are placed

on this technology by available 2.4ghz wireless spectrum and the RF environment.
- Other active tracking systems such as the Anti-Latency Tracking System use similar methods to track. The Anti-Latency System [5] uses an expandable grid of infrared transmitters at fixed distances apart and the trackers contain an infrared camera or receiver which then captures the positions and distances of the infrared transmitters, the trackers contain an IMU with 9 degrees of freedom and an FPGA (Field Programmable Gate Array) to process and merge the data. Limitations of the Antilatency system are line of sight and excessive costs.

III. Related Work

Maciejewski et al. [6], developed a Lighthouse compatible model with custom sensor placement on a 3D printed model measuring 40 mm x 140 mm x 97 mm with mass of 0.5 kg. This was mounted onto a rocket launcher for soldier training in a VR environment. They also tested the model against the SteamVR sensor simulation software and found the number of active sensors detected was predicted correctly. However, position and orientation measurements did not accurately cross over. Their explanation of the discrepancy is inaccurate sensor placement on the 3D printed model and sub ideal conditions in the testing scenario. Overall, their accuracy and precision were excellent in the testing scenario yielding mostly position precision within 1mm and orientation within 0.1 degree.

Ng et al. [7] presents an integrated circuit with sensors for a lighthouse-based system mounted on a pair of glasses. Their process involves iterative sensor placement and calibration. Unlike the previously mentioned work, testing against the simulation results in not performed.

IV. Tools Used

To design and test custom active trackers, several software and hardware tools are used in this project, listed below:
- Vive Tracking System, 2.0 Lighthouse Base Station
- Virtual Builds HDK
- Tundra-Labs HDK
- FDM 3D Printer
- SteamVR Virtual Reality Environment
- SteamVR Tracking HDK
- OpenSCAD
- Unity

The workflow and tools used are described in Fig. 1

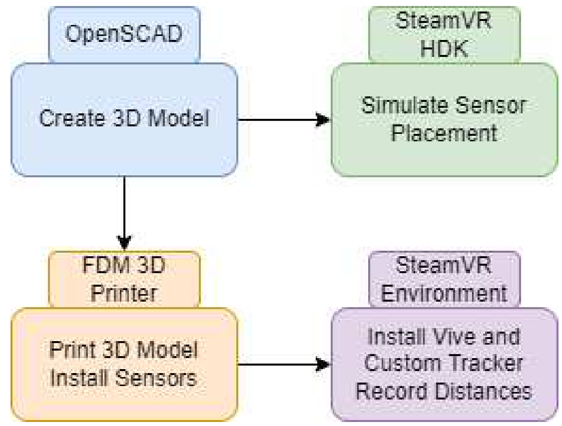

Fig. 1: Workflow and tools used

V. SteamVR Constraints and Considerations

A. Size

The current generation of trackers are large due to the size of the electronics, battery, and minimum sensor distance. In the sensor design and configuration there are 2 things to keep in mind when attempting to reduce the size of trackers, baseline, and jitter.

- Baseline is the measurable time or distance between sensors. Consider a tracker with two points(sensors) with a fixed distance apart, when viewed from the perspective of a fixed object(base station), as the tracker moves further away from the base station the two points appear closer together and eventually the 2 points are seen as a single point, this is where the baseline approaches zero.
- Jitter is an issue caused by the syncronization between the base station and the tracker. As the infrared laser sweeps the area and then hits a sensor, it can happen at any time during a clock cycle of the base station. However, in a basestation, similar to any processor-based system, all the operations are synchronized by the pulses of the clock. This means the sensor hits can only be processed as early as the next CPU cylce. The error introduced by jitter has the effect of reducing the available baseline. As the baseline is reduced the jitter error accounts for more of the measured time between sensors.

B. Technology Improvements

The Steam VR 1.0 uses 2 independent laser sweeps, each with their own baseline and jitter error in the X and Y axis, while the 2.0 system uses a single 45-degree laser sweep which eliminates the extra error introduced by using 2 laser sweeps. Unfortunately, the Steam VR 2.0 tracking algorithm has not been released. A more recent development was the release of the Tundra-Labs HDK which comes in a much smaller package than previous Steam VR HDK's allowing for smaller models to be attempted. Recently, both Tundra labs and HTC Vive have introduced miniaturized active SteamVR trackers that are smaller than previously available tracking units, however these devices are still not cost effective enough for educational VR laboratories.

C. SteamVR HDK Required

While the Steam VR Tracking HDK documentation has provided some guidance in creating customized trackers, it does not provide much insight into reducing the size of trackers. The documentation does not provide guidance for the minimum baseline needed to track an object, although they do provide a simulation tool which is used to simulate the tracking performance of an object or shape, but its results must be interpreted. The tool also provides a score from 1000 to 0,

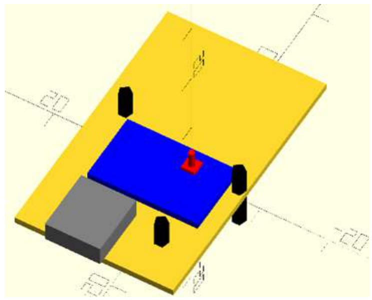

Fig. 2: Electronics board model centered about the IMU

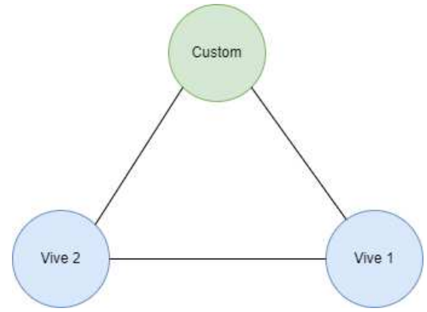

Fig. 3: Testing Mount Configuration

with lower scores being better, but if the tracker is not intended for a full 360-degree field of view the score may not be a good indicator for the intended field of view. Other tools provided and documented in the HDK are needed for calibration of the tracker. Other design constraints of the SteamVR system are a minimum of 5 sensors visible to initialize tracking and 4 visible sensors to continue tracking.[1]

D. Wireless Communication

The Steam VR tracking system and software was updated to support a maximum of up to 64 tracking devices, while the Unity Steam VR plugin was still limited to a maximum of 16. Although the actual maximum number of trackers will depend on the RF environment and interference from other 2.4ghz wireless devices. During usage, the wireless dongles had to be separated from each other or the trackers would experience a loss of tracking performance or strange behaviors. When using multiple base stations, the wireless trackers will also use more wireless bandwidth because of more sensors being activated simultaneously.

VI. Tracker Design and Testing Method

A. Tracker Design Method

We utilized the OpenSCAD modeling software incorporated into the Steam VR Tracking HDK software to generate our experimental models and run simulations. Running multiple simulations using the HMD Designer Software we were able to force sensor generation to the center of an 8mmx2mm cylinder.

Electronic components were modeled and centered around the IMU which was modeled and oriented according to its datasheet. This allowed for a flexible and iterative design process, where updates to the other components would be relative to the IMU, lessening the complexity of tracking various configurations.

Along with the simulated results, the output of the sensor configuration is generated and copied into the tracker configuration file. The sensors are placed according to the position mapped out by the simulation, and the simulation results are saved. The IMU position is then entered into the configuration file.

The tracker should then have its position fixed inside of a box and using the IMU calibration software from the SteamVR HDK, the IMU should be rotated on all 6 sides of the box and the results copied into the IMU section of the configuration file. The tracker is then used with the Lighthouse calibration software which can adjust sensor position and IMU position.

To calibrate, a tracker is moved manually through rotations and translations until the required number of sensors are seen and triggered and the resulting calibration file is saved to the tracker object

B. Testing Method

For each designed model we run the SteamVR HDK simulation software, which produces error charts when the tracker is in certain positions. In these figures we see 4 charts which provide tracking accuracy based on the trackers pitch and yaw. The first graph is the number of visible sensors at that pose. The bottom left graph with determine how likely the tracker will be able to resolve an initial pose and begin tracking. The two right graphs estimate the pose and translation errors based on pitch and yaw. Using this simulation tool, we can quickly iterate on various sensor configurations to find an optimal solution before proceeding to a real-world accuracy experiment.

Evaluating the performance of a tracker in isolation is a difficult task without expensive equipment used in [6]. To get a comparative result against a known accurate solution (Vive tracker), we compare the position of a custom tracker model against the two Vive trackers. The custom tracker and 2 Vive trackers were installed on a triangular mount. At each frame update, the distances between each tracker were recorded and analyzed.

VII. Design and Results

A. Dome model

Our first model, Dome model, targeted a small FoV and consisted of 12 sensors. The board and IMU were mounted to the bottom afterwards and the IMU's position measured from the center. We print a 3D model (fig. 5), place sensors in configuration (fig. 4) and run the simulation (fig. 6)

After running the simulation on the Dome model sensor configuration, we produce several informative graphs. The first figure (6), we see the number of visible sensors from the pitch and yaw angles of the model. A higher number of visible sensors will lead to better pose estimation. As expected from out limited FOV design, we see excellent coverage from atop on the sensor but little to none as we rotate it past the range of

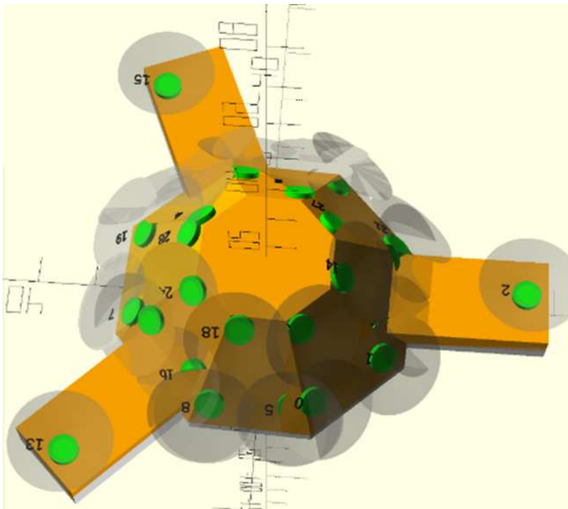

Fig. 4: The generated sensor map and placements for Dome model

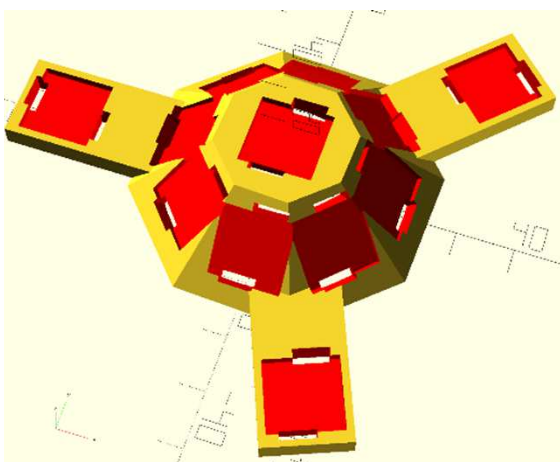

Fig. 5: 3D printable model

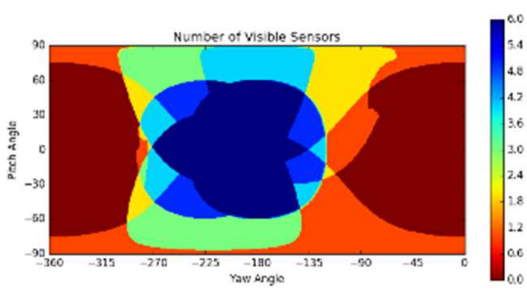

Fig. 6: Dome model Simulation

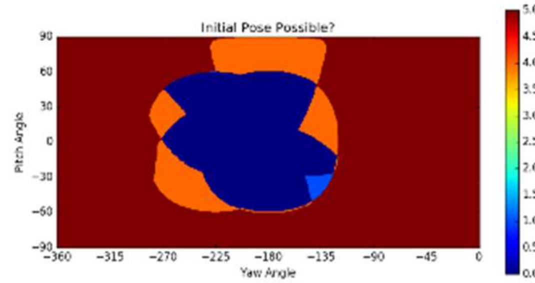

Fig. 7: Dome model Simulation

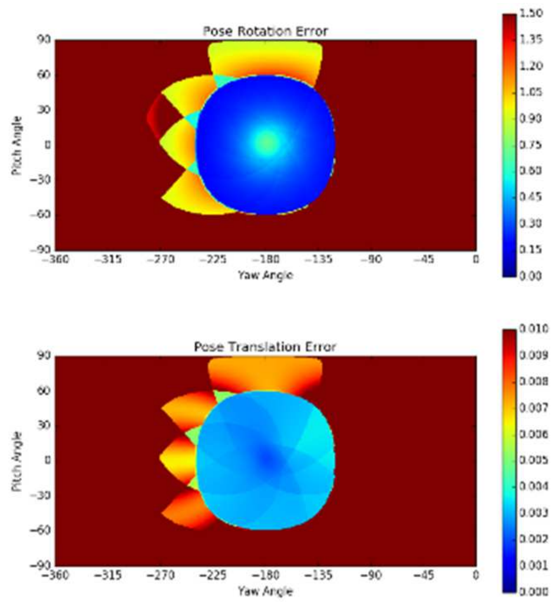

Fig. 8: Rotation and Translation Simulation Errors

-300 to -130 Yaw. This directly leads to the patterns of failure to find an initial pose (seen in Fig 7).

The other set of graphs we produce for each model from the simulation are the rotation and translation errors in the pose estimation, again as a function of pitch and yaw angles. We see in (Fig. 8), the number of visible sensors highly predicts the general shape of successful pose estimation. In the Dome model, we see a limited region of success only in a narrow region of pitch and yaw.

Dome Model Results: As each tracker is a vertex of an equilateral triangle, the distance of each side is constant. From Table 1, we see that the distance of the edge Dome-Vive1 is similar to the distance on edge Dome-Vive2 yet differs significantly from the distance of edge Vive1-Vive2. This suggests that while the Dome tracker may be consistent with itself, there is some error when comparing the methods.

TABLE I: Mean distance and standard deviation values for Dome model experiment

	Dome-Vive 1	Dome-Vive 2	Vive 1 - Vive 2
Mean(mm)	185.9151699	187.2932008	135.8645206
SD σ(mm)	15.23766448	13.42565155	2.600343282

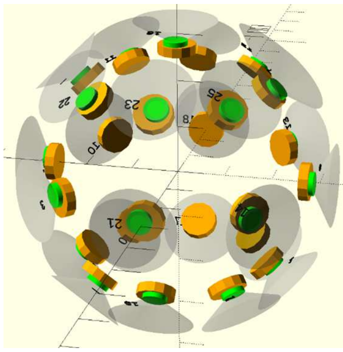

Fig. 9: Sphere model Sensors

Fig. 10: Sphere Model

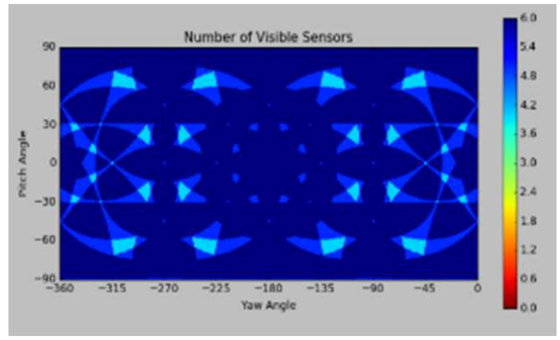

Fig. 11: Number of Visible Sensors Sphere model Simulation

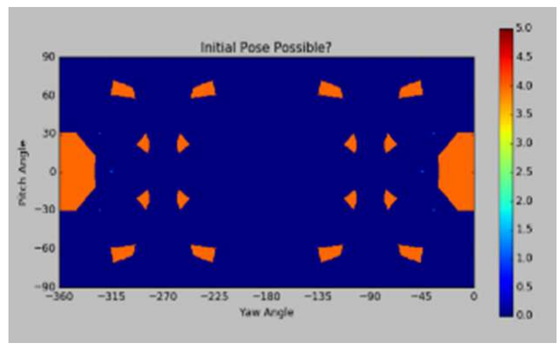

Fig. 12: Initial Pose Sphere model Simulation

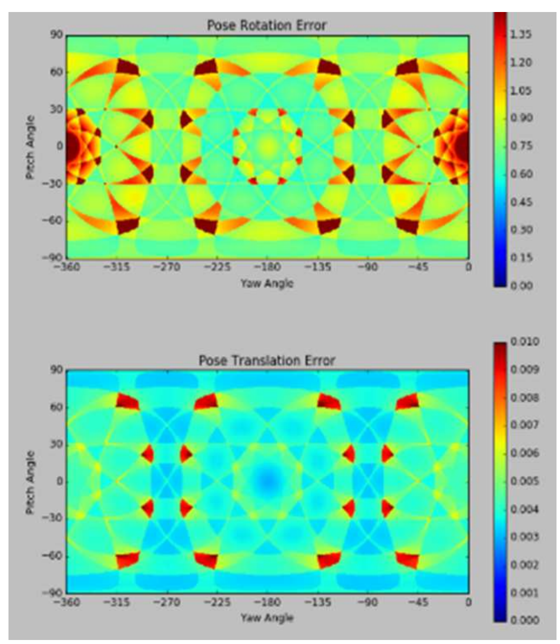

Fig. 13: Rotation and Translation Simulation Errors

Additionally, a higher standard deviation on the edges of the Dome tracker suggests less reliability frame to frame. (Table I)

B. Sphere Model

The Sphere model focuses on maximizing simultaneous visible sensors and a large FoV. Using the Tundra Labs HDK, we were able to put more sensors in a smaller region. This model is printed 55 mm radius sphere.

In Fig. 11, we see improved visibility of the sensors through a range of pitch and yaw angles. Interestingly, we see an unexpected pattern form in Fig. 12, where a circle region of failure to initialize a pose estimation forms between pitch 30 and -30 and yaw -45 and 45. This phenomena is not directly indicated by the simultaneous visible sensors.

From Fig. 13, pronounced rotational errors are found in the circular failure region as before but not translation errors.

Sphere Model Results: This tracker resulted in poor performance as predicted by the simulation(Fig. 11). As the tracker rotated, the position would become out of sync with the other trackers. The issue is believed to be an insufficient or lack of depth between sensors. Further exploration of the tracker was conducted with a single base station to identify the change

TABLE II: Mean distance and standard deviation values for Sphere model experiment at various distances from base station

	Sphere-Vive 1	Sphere-Vive 2	Vive 1 - Vive 2
1.34 from Base Station			
Mean(mm)	147.6109203	135.2449785	137.6254452
SD σ(mm)	11.15328334	11.0872362	5.690820653
2.8 from Base Station			
Mean(mm)	126.2272221	124.8273919	137.7797191
SD σ(mm)	26.811163	24.16739651	5.433525025
4.5 from Base Station			
Mean(mm)	145.7790724	119.8155212	134.3809292
SD σ(mm)	43.59053305	45.23590961	11.10614846
5.6 from Base Station			
Mean(mm)	90.54944422	172.4518982	145.8462658
SD σ(mm)	41.51769694	48.11635621	14.92893716

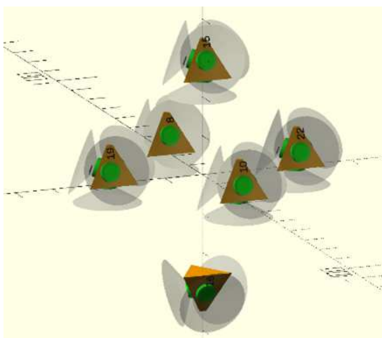

Fig. 14: Tower Model v1 Sensor configuration

in performance with distance and investigate how that might be impacting the tracker's behavior in a multi base station environment.

As the tracker was moved farther away from the base station it began to shift towards one Vive tracker or another instead of maintaining a relatively equal position. The standard deviation of the distances involving the Sphere Model tracker scaled worse than the Vive trackers. In the case of multiple base stations this can present a problem with unbalanced behavior as the tracker moves closer to one base station and away from others, resulting in different positions for the same object depending on which base stations are used. (Table II)

C. *Tower Model(s)*

Tundra labs suggest that an optimal sensor configuration is a grouping of 3 sensors onto a plane and an additional sensor not on the plane[2]. Using OpenSCAD, we developed variations of this idea, quickly simulating results on each sensor configuration. A 3D printed tracker is created and tested for the best performing simulation. These models are called Tower models as they all feature a central pillar with a sensor.

Tower Model v1: In Tower Model v1, we have pyramids, each face with a sensor. 4 of these pyramids are in a plane while 1 is above and one is below shown in Fig 14.

From the simulation on this iteration, we find problem regions consistent across both the number of visible sensors and initial pose estimation (Figs 15,16). Again these problem regions of pitch and yaw angles persist into both pose rotation and translation errors.

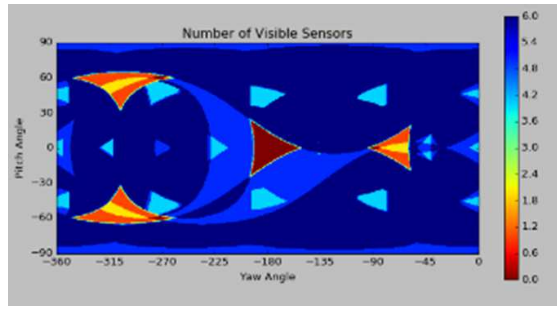

Fig. 15: Number of Visible Sensors Tower v1 model Simulation

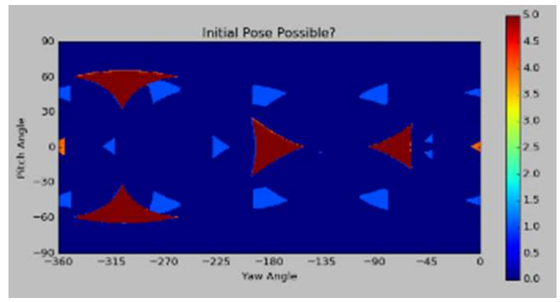

Fig. 16: Initial Pose Tower v1 model Simulation

Tower Model v2: In the second iteration of our tower model, (Fig 18), 3 cubes are placed on a rectangular plane where a fourth cube is placed atop a tower. Each cube has 1 sensor on each face. With this sensor configuration we see a loss of line of sight of a sensor when viewed from certain angles. As we need 5 trackers at a minimum for initial pose estimation this configuration leads to failure at significant portion of angles as we see in Fig. 20. The rotation and translation errors (Fig 21)

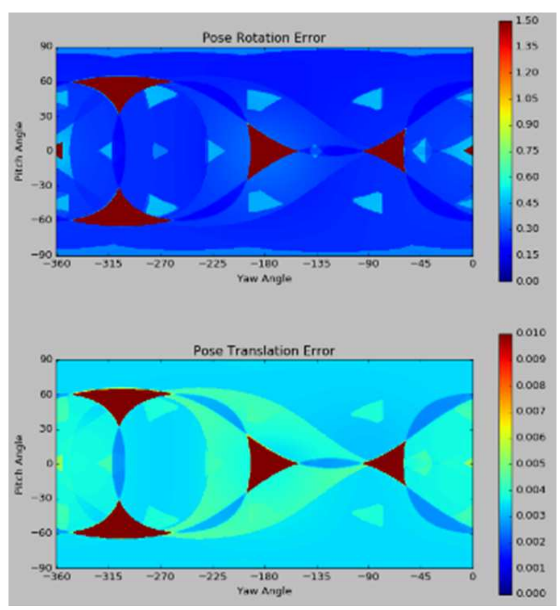

Fig. 17: Rotation and Translation Simulation Errors

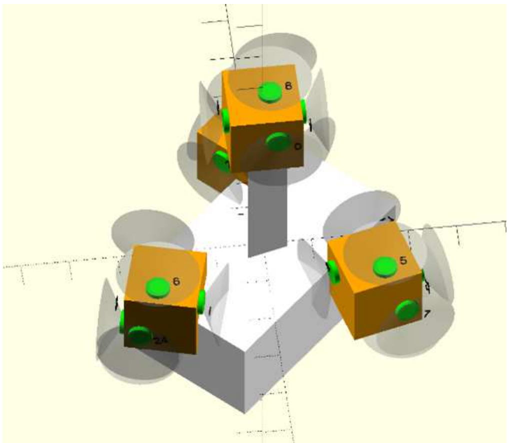

Fig. 18: Tower Model v2

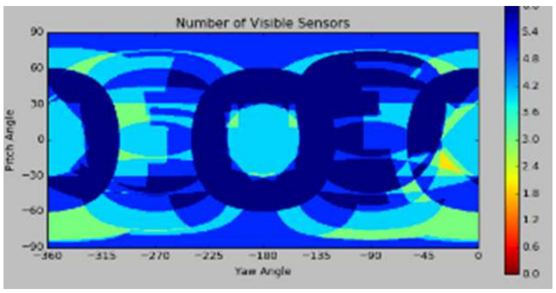

Fig. 19: Number of Visible Sensors Tower v2 model Simulation

for this sensor configuration are consistent with the number of visible sensors. However, across all angles the translation error appears higher.

Tower Model v3: An iteration on the previous model, Tower Model v3 places 4 cubes in the midpoint of each edge of the rectangular base with an additional cube atop a tower in the center. (Fig 22) We see similar issues with line of sight. From the simulation results in Fig. 24, we see initial pose estimation suffers from a variety of angles. However in Fig. 25, the rotation and translation errors experience much fewer dead regions due to 4 sensors being visible at almost all angles as seen in Fig. 23.

c In Tower Model v4, four rotated cubes are placed inside each corner of a rectangle with a singular cube sitting rotated atop a central pillar. (Fig. 29)

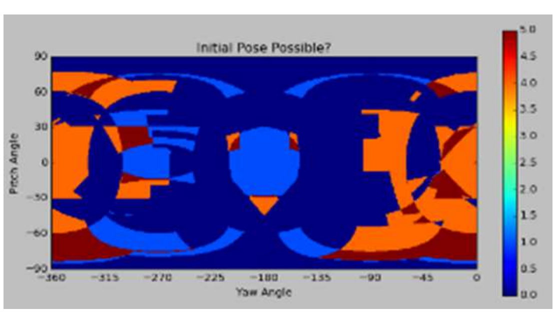

Fig. 20: Initial Pose Tower v2 model Simulation

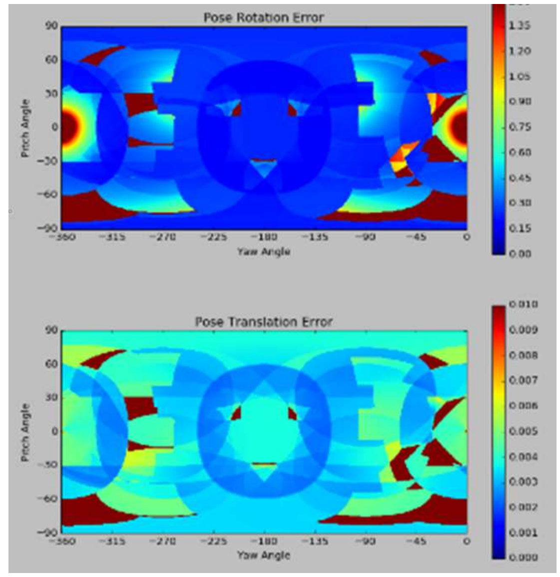

Fig. 21: Rotation and Translation Simulation Errors

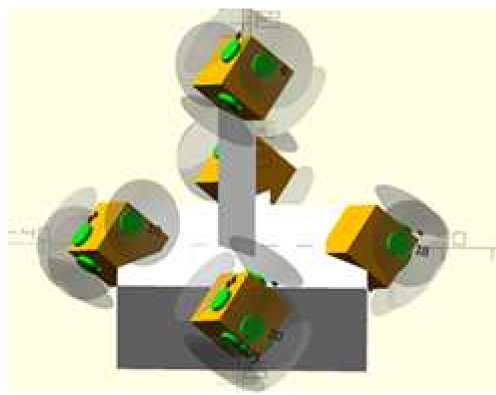

Fig. 22: Tower Model v3

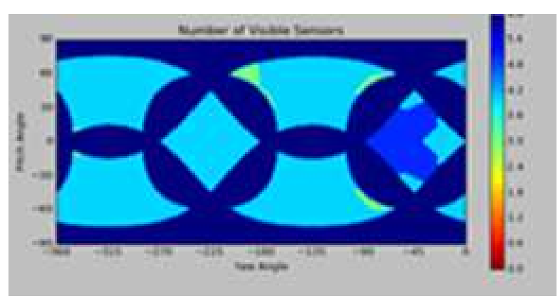

Fig. 23: Number of Visible Sensors Tower v3 model Simulation

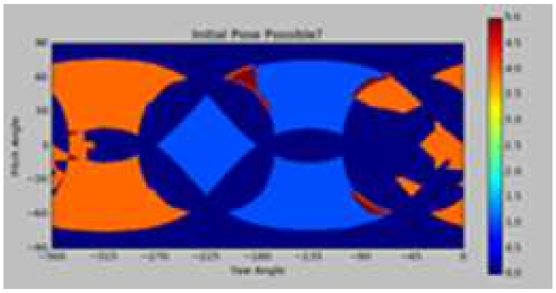

Fig. 24: Initial Pose Tower v3 model Simulation

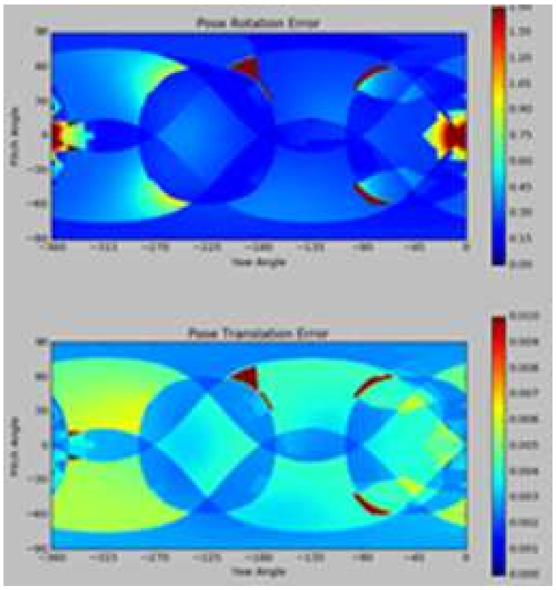

Fig. 25: Rotation and Translation Simulation Errors

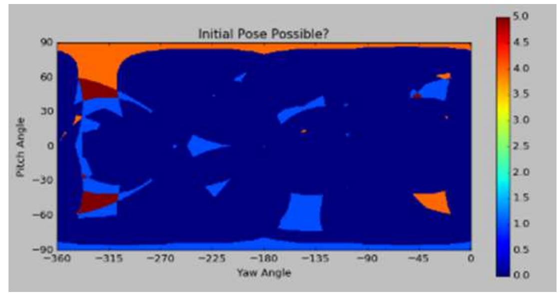

Fig. 27: Initial Pose Tower v4 model Simulation

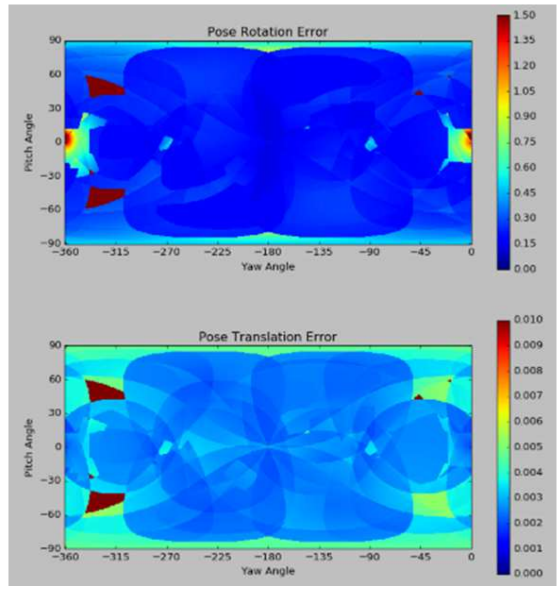

Fig. 28: Rotation and Translation Simulation Errors

Tower Model v4 provides a much improved initial pose estimation, with an error largely residing at pitch angles ¿ 85 (Fig. 20. For most angles, the number of visible sensors remains close to 5. (Fig. 19)

Sensor tracking is well spread and minimal dead regions as seen in Fig. 28. Excellent rotation error is seen for most angle ranges and translation error is consistent but on average worse.

Tower Model v5: Tower Model v5 features slightly different sensor placement on each face but the same cube positions as Tower Model v4. (Fig. 34)

Seen in simulation Fig. 31, this sensor configuration features even better visible sensor coverage, leading to the smallest dead regions from any model so far. (Fig. 32).

Tower Model v5 provides the minimal average error for pose rotation and translation estimation, while minimizing dead regions for initial pose estimation. This model is selected to for 3D printing and real-world evaluation.

Final Tower Model: We note that this final model configuration is within 4 mm average distance from the Vive-Vive distance and 3 mm standard deviation across all frames, a

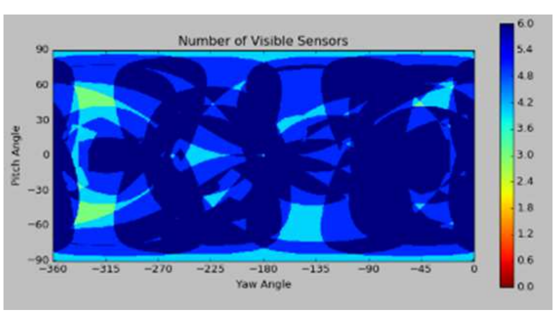

Fig. 26: Number of Visible Sensors Tower v4 model Simulation

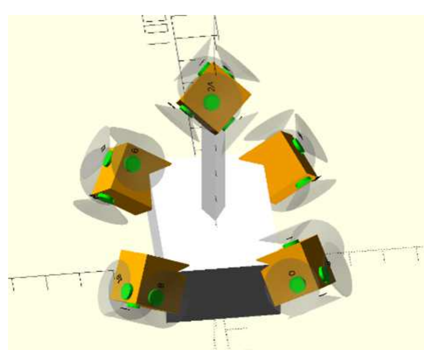

Fig. 29: Tower Model v4

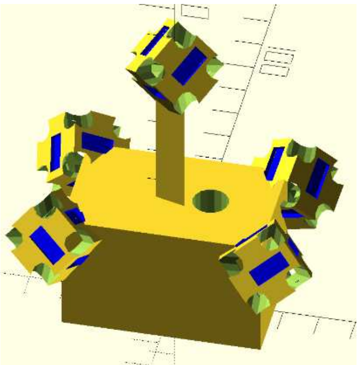

Fig. 30: Tower Model v5

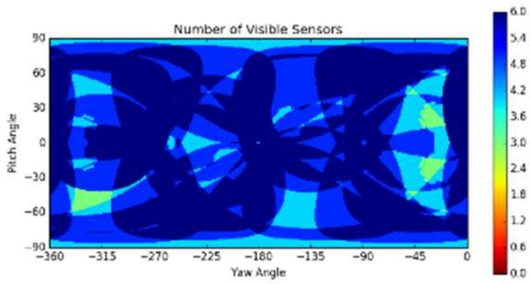

Fig. 31: Number of Visible Sensors Tower v4 model Simulation

marked improvement from previous attempts (Table III).

This model is a significant improvement from the Dome and Sphere models. From the Tower v5 model to each Vive tracker is within 4mm on average with a standard deviation of 7.5mm.

VIII. Conclusion

Our initial goal to reduce the size and cost of active trackers in the Lighthouse based tracking environment SteamVR while maintaining high accuracy was largely successful. Accuracy loss was limited to 4 mm on average for the best performing

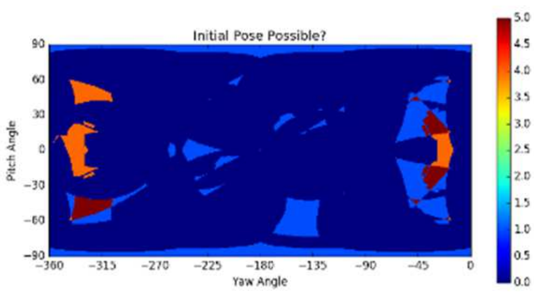

Fig. 32: Initial Pose Tower v4 model Simulation

TABLE III: Mean distance and standard deviation values for Model 3 experiment

	Tower v5-Vive 1	Tower v5-Vive 2	Vive 1 - Vive 2
Mean(mm)	131.2415568	137.0756916	133.5836188
SD σ(mm)	7.462882757	7.53684492	4.253859754

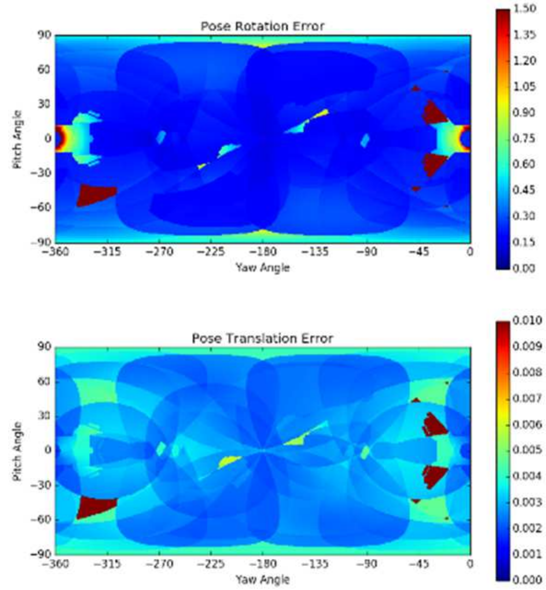

Fig. 33: Rotation and Translation Simulation Errors

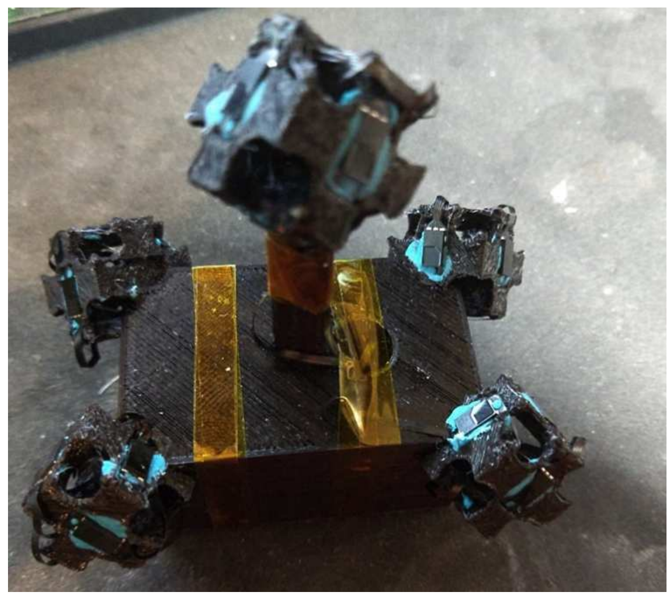

Fig. 34: Tower Model v5 3D Print

model. A weight savings from 75 g (Vive 3) to 65 g was achieved. Additionally, the cost of a Vive tracker is currently $125 while our tracker can be made for much cheaper at (??).

While improvements in the technology and reductions in the physical size of the electronics have made it possible to reduce the size of housing, there are still limitations of the tracking technology itself which make it difficult to further reduce the size of trackers without sacrificing accuracy. There may be more optimal sensor configurations that strike a better balance between accuracy and size and allow for better designs that integrate them into a smaller overall tracker.

References

[1] "SteamVR Tracking HDK (Hardware Development Kit)." https://partner.steamgames.com/vrlicensing (accessed Sep. 26, 2021).

[2] "Tundra Labs," Tundra Labs. https://tundra-labs.com/ (accessed Sep. 26, 2021).

[3] M. Mihelj, D. Novak, and S. Begus, Virtual Reality Technology and Applications. Springer Publishing Company, Incorporated, 2013.

[4] "Base Station — VIVE USA." https://www.vive.com/us/accessory/base-station/www.vive.com/us/accessory/base-station/ (accessed Sep. 26, 2021).

[5] "Antilatency tracking system for VR/AR, Virtual Production, Drones and Robotics." https://antilatency.com/ (accessed Sep. 26, 2021).

[6] Maciejewski, Marcin & Piszczek, Marek Pomianek, Mateusz & Pałka, Norbert. (2020). DESIGN AND EVALUATION OF A STEAMVR TRACKER FOR TRAINING APPLICATIONS – SIMULATIONS AND MEASUREMENTS. Metrology and Measurement Systems. 27. 601-614. 10.24425/mms.2020.134841

[7] A. K. T. Ng, L. K. Y. Chan and H. Y. K. Lau, "A low-cost lighthouse-based virtual reality head tracking system," 2017 International Conference on 3D Immersion (IC3D), 2017, pp. 1-5, doi: 10.1109/IC3D.2017.8251910.