frontiers | Frontiers in Virtual Reality

TYPE Original Research
PUBLISHED 19 July 2023
DOI 10.3389/frvir.2023.1169313

OPEN ACCESS

EDITED BY
Anıl Ufuk Batmaz,
Concordia University, Canada

REVIEWED BY
David Lindlbauer,
Carnegie Mellon University, United States
Louis Nisiotis,
University of Central Lancashire, Cyprus
Filip Škola,
CYENS Centre of Excellence, Cyprus

*CORRESPONDENCE
Dennis Reimer,
reimerde@rwu.de

RECEIVED 19 February 2023
ACCEPTED 10 July 2023
PUBLISHED 19 July 2023

CITATION
Reimer D, Podkosova I, Scherzer D and Kaufmann H (2023), Evaluation and improvement of HMD-based and RGB-based hand tracking solutions in VR. *Front. Virtual Real.* 4:1169313. doi: 10.3389/frvir.2023.1169313

Evaluation and improvement of HMD-based and RGB-based hand tracking solutions in VR

Dennis Reimer[1,2]*, Iana Podkosova[1], Daniel Scherzer[2] and Hannes Kaufmann[1]

[1]Faculty of Informatics, Research Unit Virtual and Augmented Reality, TU Wien, Vienna, Austria, [2]Faculty for Electrical Engineering and Computer Science, Ravensburg-Weingarten University, Weingarten, Germany

Hand tracking has become a state-of-the-art technology in the modern generation of consumer VR devices. However, off-the-shelf solutions do not support hand detection for more than two hands at the same time at distances beyond arm's length. The possibility to track multiple hands at larger distances would be beneficial for colocated multi-user VR scenarios, allowing user-worn devices to track the hands of other users and therefore reducing motion artifacts caused by hand tracking loss. With the global focus of enabling natural hand interactions in colocated multi-user VR, we propose an RGB image input-based hand tracking method, built upon the MediaPipe framework, that can track multiple hands at once at distances of up to 3 m. We compared our method's accuracy to that of Oculus Quest and Leap Motion, at different distances from the tracking device and in static and dynamic settings. The results of our evaluation show that our method provides only slightly less accurate results than Oculus Quest or Leap motion in the near range (with median errors below 1.75 cm at distances below 75 cm); at larger distances, its accuracy remains stable (with a median error of 4.7 cm at the distance of 2.75 m) while Leap Motion and Oculus Quest either loose tracking or produce very inaccurate results. Taking into account the broad choice of suitable hardware (any RGB camera) and the ease of setup, our method can be directly applied to colocated multi-user VR scenarios.

KEYWORDS

virtual reality, hand tracking, colocation, RGB tracking, mediapipe

1 Introduction

Optical markerless hand tracking enables Virtual Reality (VR) applications where users can interact intuitively using only their hands, without any additional input controllers or external tracking infrastructure. Numerous games[1] and applications[2] already make use of this technology.

As an example, (Khundam et al., 2021) argue that hand tracking will play an important role in medical applications since it is more natural for real-world situations and comparable in usability to conventional controllers. A user study by Voigt-Antons et al. (2020) also

1 https://arvore.io/project/the-line

2 https://www.vrdesktop.net/

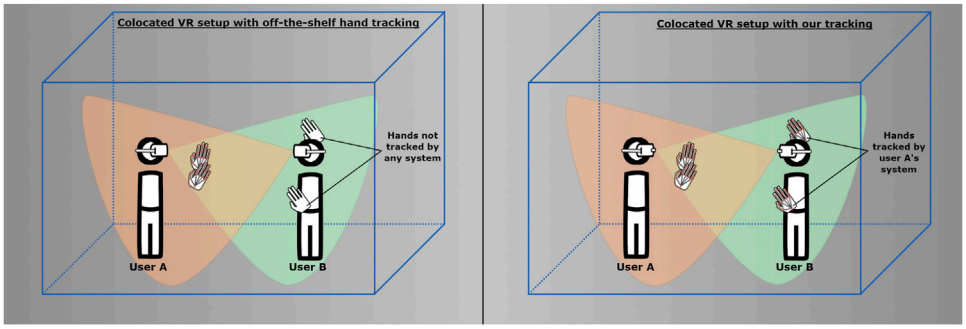


FIGURE 1
A colocated multiuser VR setup. User B's hands are outside the range of their own hand tracking, but visible to user A. With off-the-shelf solutions, only two hands can be detected at the same time within a short range (left). Our solution allows us to detect and position the hands of other users in 3D space. This way the hands of user B can still be detected (right).

revealed a higher presence for the user doing a grab interaction and better usability for typing interactions.

Several commercial state-of-the-art VR systems offer integrated hand tracking solutions: Oculus Quest (Han et al., 2020), HTC Vive Focus 3, and Leap Motion controller that can be easily attached in front of a head-mounted display. Although these systems are robust and easy to use, they primarily target single-user applications. Multi-user scenarios, especially in a colocated setup, are problematic for off-the-shelf hand tracking solutions. Usually, hand tracking algorithms expect only two hands within the tracking volume, those belonging to the user who is wearing the tracking device and is alone in the tracking space. When hands of colocated users enter the view frustum of the hand tracking device, they often get erroneously recognized in place of the correct pair of hands or disrupt an already established hand tracking process. A narrow field of view of tracking devices leads to a further problem: when the user's hands are not held in front of the device they are not tracked (this situation is illustrated in Figure 1 on the left). In single-user applications, the virtual hands are simply not rendered in such cases. For multi-user applications, however, loss of hand tracking is more problematic: if avatar hands stop being rendered, other users lose important information on the posture and activities of their collaborators; if the hands are rendered but not tracked the avatars of others take on unrealistic or uncanny postures. Such disruptions are especially noticeable in full-body avatars.

The global goal of our work is to enable natural and reliable hand interactions in colocated multi-user VR scenarios. In order to accomplish this, our aim is to enhance the 3D pose estimation of tracked hands within an existing hand tracking system, thereby enabling such interactions over a wide tracking range. Ensuring consistent tracking of all users' hands within the shared workspace is crucial for this endeavor. To achieve reliable hand tracking for all colocated users, we propose a method that takes advantage of the tracking system's capability to track more than two hands and operates within an extended range of distances, and furthermore accurately position the hands in a three-dimensional space, thereby facilitating colocated hand interactions; this way, each tracking device which is worn by a user can provide tracking input not only for this user's virtual hands but also for virtual hands of colocated others. This idea is presented in Figure 1 on the right: although the hands of user B are outside the field of view of their hand tracking camera, they are tracked by the camera of user A and can be rendered correctly. In this case, the camera of user A is tracking four hands at the same time. This is not possible with the integrated hand tracking of the Oculus Quest, where Han et al. (2020) present the recognition algorithm with the fact that they expect a maximum output of two hands, making the recognition of more than two hands impossible. The same applies to Leap Motion or integrated hand detection systems of other HMDs (like the HTC Vive Cosmos).

Since hand tracking methods integrated into off-the-shelf devices are closed systems, it is currently impossible to adjust them to closer align with the interaction requirements of colocated multi-user VR. For this reason, we turn to methods that use RGB input to detect the user's hands. RGB-based methods have certain advantages: they can work with any RGB source, not being bound to any specific hardware, and they work at larger distances, the limits of tracking being set only by the resolution of users' hands in the images. However, most RGB-based solutions offer the capability of detecting the hand pose in 2D image coordinates only, additional calculations being necessary to obtain the full 3D pose.

This paper presents a hand tracking method that is based on the MediaPipe framework, a cross-platform solution for object recognition (including hand recognition) in 2D images using machine learning. To calculate the full 3D hand pose based on finger joint coordinates provided by MediaPipe, we have developed an algorithm that uses an estimation of the user's hand size to obtain its distance from the tracking camera. We evaluate the performance of our method in comparison with hand tracking methods provided by Oculus Quest and Leap Motion, providing an accuracy assessment for each method in static and dynamic conditions in the range from 0.25 m to 3 m from the tracking camera. The results indicate comparable performance for all evaluated methods in the near range (at arm's length) distances; however, our RGB-based method provides better accuracy at mid-range and keeps working at larger distances. With good tracking accuracy at typical tracking area-scale distances and the ability to track more than two hands, our method presents a step towards enabling reliable natural hand interactions in colocated multi-user VR.

2 Related work

Research on hand recognition and its application in VR setups has gained significant attention. Precise and natural hand recognition is essential for enabling immersive interactions. Consequently, several studies have explored hand interaction in multiuser scenarios and demonstrated the positive impact of such interactions. For instance, Li et al. conducted user testing to assess the influence of interactions in colocated multiuser scenarios related to cultural heritage, and their findings indicated that social influence positively affects performance expectancy and effort expectancy (Li et al., 2018).

Streuber et al. conducted user tests in multiuser scenarios and discovered that the absence of haptic and tactile feedback can be compensated for if users are fully immersed in the virtual environment. Their study highlighted the significance of user immersion in mitigating the absence of physical feedback (Gong et al., 2020).

Furthermore, Gong et al. conducted a case study on interaction systems in multiuser VR settings, further contributing to the body of knowledge in this area (Gong et al., 2020).

In addition to these studies, the following sections will delve into research on RGB hand tracking and existing evaluations of hand tracking accuracy, which are crucial factors for effective hand interactions.

2.1 Hand tracking with RGB and RGB-D cameras

This section reports related work on hand detection and hand tracking based on single-image input from RGB and RGB-D cameras. In contrast to pure RGB, RGB-D cameras provide additional depth information on the scene in the image. However, RGB cameras are more commonly available, and therefore hand tracking methods based on RGB input only could apply to a wider range of hardware.

For our use case, we need a hand tracking algorithm, which is able to recognize the user's hand and finger joints in real-time. Various research was carried out in the area of hand detection (detect a user's hand in an image) and 3D position estimation (for finger joints) for RGB (Panteleris et al., 2017; Zhang et al., 2020; Sun et al., 2021) and RGB-D (Sharp et al., 2015; Malik et al., 2018; Huang et al., 2021) cameras.

Huang et al. (2021) conducted a survey and performance analysis of hand shape and pose estimation approaches using RGB and RGB-D cameras. They find that there exist several state-of-the-art methods which achieve a low estimation error (<10 mm) enabling usable interactable applications. In terms of effectiveness and efficiency, some challenges (such as occlusion or self-similarity) have to be tackled further. Malik et al. (2018) deduced 3D mesh representations of a hand from a single depth image. They used a synthetic dataset with accurate joint annotations (joint location error <15 mm), segmentation masks, and mesh files of depth maps for neural network recognition with a processing time from depth image to mesh of 3.7 ms. Zimmermann and Brox (2017) published a method of 3D hand pose estimation based on RGB input, tested on a synthetic dataset for neural network hand recognition. The performance of this method was comparable to existing depth approaches. Lin et al. (2020) used a neural network-based pipeline for hand tracking and created a new 3d dataset for training the algorithm. They did this for two simultaneously tracked hands. They report a mean End Point Error of 12.47 mm but only at an arm's length range and for a maximum of two simultaneously visible hands.

A method for transforming 2D finger joint poses into 3D points was developed by Panteleris et al. (2017). They used OpenPose for 2D hand recognition and non-linear least-squares minimization to fit a 3D model of the hand to the estimated 2D joint positions, recovering the 3D hand pose. Additional recognition methods were developed by Che and Qi (2021) and Sun et al. (2021) recognizing the hand region and deriving 2D and 2.5D points of the hand using networks. Wang et al. (2020) presented a method to track 3D real-time interactions with a monocular RGB camera.

MediaPipe framework (Lugaresi et al., 2019) offers perception pipelines to enable object detection among other things for RGB cameras with the help of machine learning. Zhang et al. (2020) showed a hand tracking implementation with MediaPipe offering 3D joint recognition for more than two simultaneously visible hands with good performance and precision. Due to its open accessibility and good performance (especially on mobile devices with 1.1 ms–7.5 ms on iPhone 11, depending on the used model (Zhang et al., 2020)), we chose the MediaPipe framework as the bases of our hand tracking method using RGB input. However, the method presented by Zhang et al. (2020), like many similar methods, provides 3D finger joint positions relative to the origin in the middle of the hand. The distance from the hand to the tracking camera remains unknown.

In this paper, we describe an extension of the method that provides the calculation of the hand position in the 3D coordinate frame of the tracking camera, resulting in the full world pose that is necessary for interactions in MR applications with objects positioned in the virtual environment.

2.2 Evaluating hand tracking accuracy

Previous studies have investigated various accuracies related to hand tracking systems. Schneider et al. (2021) report the accuracy of finger tracking for touch-based tasks such as pointing or drawing. They report lower spatial accuracy for HTC Vive hand tracking than for Oculus Quest and Leap Motion; users mostly preferred the Leap Motion sensor. Their previous results indicated better tracking accuracy for Leap Motion than for HTC Vive, with a Z-error of approximately 2.6 cm when interacting with horizontally aligned surfaces (and 1 cm for vertically aligned surfaces respectively) in walk-up-and-use scenarios Schneider et al. (2020).Another accuracy study for Leap Motion was performed by Vysocký et al. (2020), reporting an error that can be up to 1 cm for measurements made at a distance of 20 cm.

Mizera et al. (2020) compared the visual tracking accuracy of the Leap Motion sensor to the accuracy of two data gloves. They found that the Leap Motion was less precise when measuring finger bending but very precise in estimating fingertip positions. They describe the Leap Motion as the best device to do fine manipulation tasks with the thumb and an opposite finger.A framework to

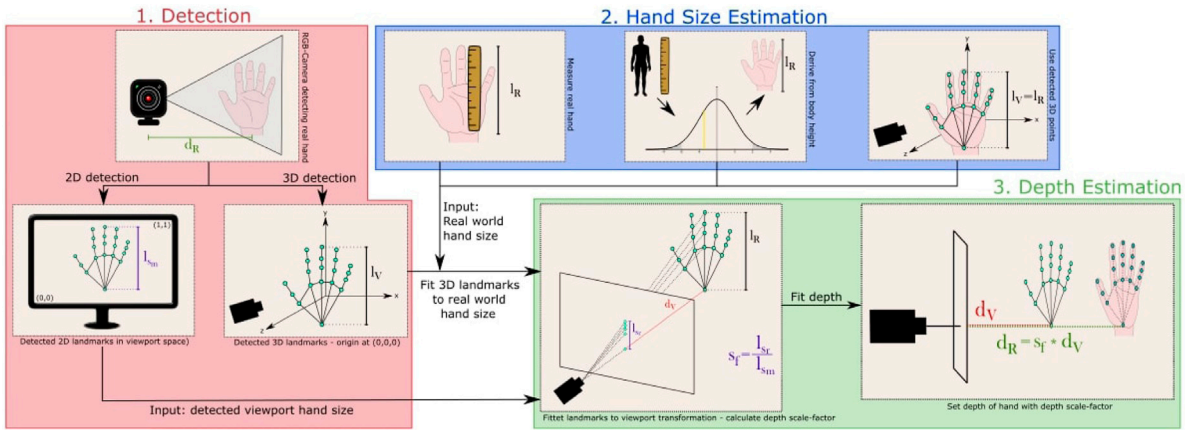


FIGURE 2
Step-by-step diagram for adjusting and positioning a detected hand from MediaPipe. After detection, the virtual hand is adjusted to the real-world hand size and then positioned with the help of the intercept theorem.

measure tracking accuracy for VR hand tracking systems was presented by Abdlkarim et al. (2022), demonstrating its use on Oculus Quest 2. They used a height-adjustable table which can be lowered to a height difference of 82 cm. The Oculus was attached on top so it could track the user's hand while executing the experiment. Ground truth data was collected from infrared markers that were put on the table and tracked by an optical camera system. Users were instructed to point at different markers during the experiment to collect data. The authors reported an average error of 1.1 cm in the position of fingertips for the Oculus Quest 2.Ferstl et al. (2021) introduced and evaluated strategies to mitigate the impact of hand tracking loss. Since they show that tracking loss can influence the experience of users, we designed our evaluation to include tracking loss distances for all assessed hand tracking systems.

Due to the lack of standardized evaluation techniques for dynamic hand movements [besides methods for machine tools and iGPS and laser tracker (Wang et al., 2011; Ding et al., 2020)], we present our own method, tailored to our experiment, to determine the tracking accuracy and error curve for dynamic hand movements.

3 Methodology

Our workflow of hand tracking based on RGB input consists of three stages:

1. **Hand Detection:** Hands are detected in the RGB image; 3D positions of finger joints are calculated relative to the center of each hand. This stage is performed by the hand tracking implementation of Zhang et al. (2020) in the MediaPipe framework.
2. **Hand Size Estimation:** Real-world hand size of the user is estimated according to one of the methods described in Section 3.2.
3. **Depth Estimation:** Distance of the hand to the tracking camera is calculated according to the method described in Section 3.3, using the estimation of the real hand size.

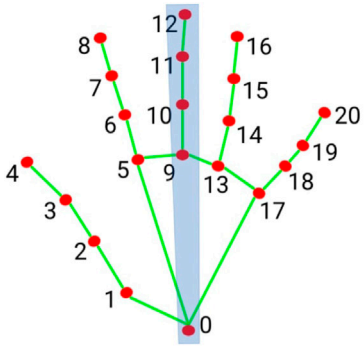
This workflow is presented in Figure 2, which includes the details of each stage described in the sections below.

Our objective is to develop a workflow for calculating the hand sizes of users and utilizing this information to accurately position the tracked virtual hand (using the MediaPipe framework) within a three-dimensional environment, thereby enabling natural hand interactions. To evaluate the effectiveness of our approach, we will compare it with existing off-the-shelf tracking solutions.By leveraging hand size estimation for positioning, we anticipate higher accuracy in hand tracking as the precision of hand size estimation improves. We also expect to achieve accurate hand size estimation by inferring the hand size from the user's body height.Overall, expect a system capable of facilitating 3D hand interactions with a much larger tracking range, surpassing the capabilities of existing off-the-shelf tracking solutions. This advancement will make our solution highly advantageous for use in colocated VR scenarios.

3.1 MediaPipe hand detection

We chose the MediaPipe framework (Lugaresi et al., 2019) as the hand detection step in our workflow (and the implemented hand and finger detection of Zhang et al. (2020)) due to its capacity to detect more than two hands at the same time. As they report an average precision between 86.22% and 95.7% for palm detection we can assume a similar detection accuracy in our experiment.

With the help of two TensorFlow machine learning models (palm detector and hand landmark model), MediaPipe tracks the finger joints of the hand with a high prediction quality. As a result of the recognition, we get the following information from the framework for each detected hand:



0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMB_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP
17. PINKY_MCP
18. PINKY_PIP
19. PINKY_DIP
20. PINKY_TIP

FIGURE 3
Landmark indices of the MediaPipe framework. Marked landmarks are used for hand length calculations.

- **Handedness:** A label ("left" or "right") and an estimation probability for this handedness.
- **World Landmarks:** 21 landmarks consisting of x, y, and z coordinates with the origin at the hand's approximate geometric center.
- **Normalized Landmarks:** 21 landmarks consisting of x, y, and z coordinates in the normalized viewport space of the camera.

The landmark definitions can be seen in Figure 3 (taken from the official MediaPipe hand tracking website[3]). The marked landmarks are later used for hand length calculation in the virtual space. Together with the remaining landmarks in the coordinate frame of the center of the hand and normalized landmarks, they are used in the calculation of the distance of the hand to the tracking camera. This detection step can be seen as the first step in Figure 2.

3.2 Hand size estimation

We use the real-world length of the user's hand to estimate the distance of the hand to the camera. To obtain the hand's size, three different methods are used, resulting in three variants of our hand tracking method that were evaluated. In Figure 2 these methods are visualized in the second step.

1. We use 3D hand landmarks with the origin in the center of the hand from MediaPipe to calculate the distance between the wrist position and the tip of the middle finger. This distance represents the length of the hand. We refer to this method of hand size calculation as *MediaPipeInternal* in the rest of the paper.
2. We measure the real hand length of the user (wrist to the tip of the middle finger) and use measurement as an input to our program (later referred to as *MediaPipeHand*).
3. The third method requires more calculations but could provide an easier setup experience for the user. Since most people do not know the length of their hand, we use the body height as an input parameter to infer the length of the hand (later referred to as *MediaPipeBody*).

Pheasant (2003) conducted an examination of different body part sizes and their frequency in the English population. Figure 4 is an excerpt from this book and shows different body part size estimations (in mm) with three percentiles (including the mean) and the standard deviation for a normal distribution. We use these values to create normal distributions and derive body part sizes from another reference body part size. Zafar et al. (2017) evaluated the body-hand relations by calculating the body size based on hand size with an accuracy of 2.9 cm. Since their method also requires the age of the user and we want to keep the input data set as small as possible, we calculate the hand size using the tables from Pheasant (2003).

This way, the actual body part does not have to be physically measured. In our case, we use the body size of the user to get its percentile in the normal distribution which is then used to recalculate the hand length size for this percentile. For this we use the following equations:

$$z = \frac{x_0 - \mu}{\sigma}$$
$$p = \frac{1}{2}\left[1 + erf\left(\frac{z}{\sqrt{2}}\right)\right] \quad (1)$$
$$x = \mu + (\sigma * z)$$

For our experiments, our user had a body size (height) of 1892 mm. For this size we look up $\mu = 1740$ mm (given at a percentile of 50%) and $\sigma = 70$ in Figure 4. With Eq. 1 we calculate a percentile of *0.985*.For the hand size we look up $\mu = 190$ mm and $\sigma = 10$. With the percentile of *0.985*, we can estimate a hand size of *211.71 mm* for the given body size. In comparison, we have determined a measured hand size of *213 mm* for the user.

We calculated the hand size based on the body height of all participants in our user test (see Section 4.2) and compared the resulting value to the measured hand size. The box-plot of the difference between the calculated and the measured hand length can be seen in Figure 5. With the mean difference of 0.0787 cm, it can be seen that the hand length can be calculated accurate from the body size, even though differences of up to 1 cm are possible depending on the user. The exact measurement results can be found in the Supplementary Material.

3 https://google.github.io/mediapipe/solutions/hands.html

Dimension	Men 5th %ile	Men 50th %ile	Men 95th %ile	Men SD	Women 5th %ile	Women 50th %ile	Women 95th %ile	Women SD
1. Stature	1625	1740	1855	70	1505	1610	1710	62
⋮								
28. Hand length	175	190	205	10	160	175	190	9
29. Hand breadth	80	85	95	5	70	75	85	4
⋮								

FIGURE 4
Body size estimations excerpt from Pheasant (2003).

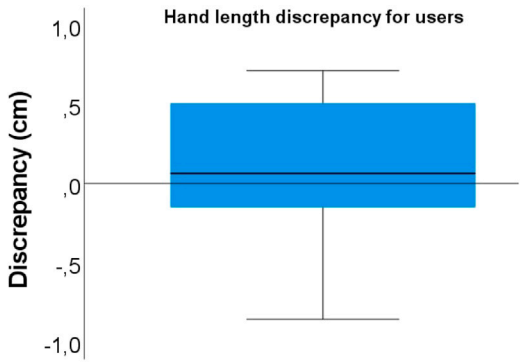


FIGURE 5
Discrepancy between measured hand length and calculated hand length of all participating users.

3.3 Hand depth estimation

The MediaPipe hand tracking solution detects 3D landmarks in the coordinate frame associated with the hand's approximate geometric center. Additionally, MediaPipe delivers 2D space coordinates of the detected landmarks in the camera image. To obtain the 3D position of the hand in the coordinate frame of the tracking camera, we use an expected real-world hand length of the user to adjust the 3D position of the landmarks to fit the size of the hand in the image and then set the depth of the hand along the z-coordinate of the camera frame.

To fit the landmarks to the expected size, we calculate a scaling factor s which we use to scale the landmark's 3D position along its connected finger part (e.g., see Figure 3—(12,11), (11,10), (10,9), (9,0)). This factor is calculated with:

$$s = \frac{l_R}{l_V} \tag{2}$$

where l_R is the expected hand length and l_V is the calculated virtual hand length. For the method where we use the given 3D landmarks of MediaPipe to calculate the expected hand length, the scaling factor will always be 1. In Figure 2 this can be seen in the third step.

3.3.1 Hand depth estimation with expected hand length

With the fitted 3D coordinates of the tracked landmarks we now have a correctly scaled hand with the expected hand length, which only has to be adjusted in the distance to the virtual camera. To achieve this we use the intercept theorem (Schupp, 1977), which describes rules about the ratio of parallel line segments which are intersected by a line. We calculate the hand length of the normalized 2D landmark positions in the camera's image space (l_{s_m}), which are obtained from MediaPipe. We also transform our fitted 3D coordinates to the viewport space and calculate the hand length of these transformed viewport points (l_{s_r}).

With the intercept theorem, we know the following about the ratios:

$$\frac{d_R}{d_V} = \frac{l_{s_r}}{l_{s_m}} \tag{3}$$

To get the final depth d_R to the camera we solve the equation to. $d_R = \frac{l_{s_r}}{l_{s_m}} * d_V = s_f * d_V$ where d_V in this step is the current distance of the virtual hand to the virtual camera. The step-by-step procedure from detecting the hand to virtual positioning can be seen in Figure 2.

4 Hand tracking evaluation

We conducted three experiments to evaluate the effectiveness of our hand tracking method. In **Experiment 1**, we conducted a comprehensive technical assessment to compare the accuracy of our method against two off-the-shelf solutions: Leap Motion and Oculus Quest. This evaluation focused on the hand data from a single user. **Experiment 2** aimed to validate the performance of our hand tracking method with hand images from multiple users. We analyzed three variants of our method using data input from nine users. In **Experiment 3**, we conducted a pilot test to assess the application of our hand tracking method in an actual colocated scenario, involving a pair of users.

The following sections will present each experiment sequentially, along with their corresponding results. This organization facilitates a clear grouping of each experiment with its respective results, enhancing readability and understanding of the outcomes.

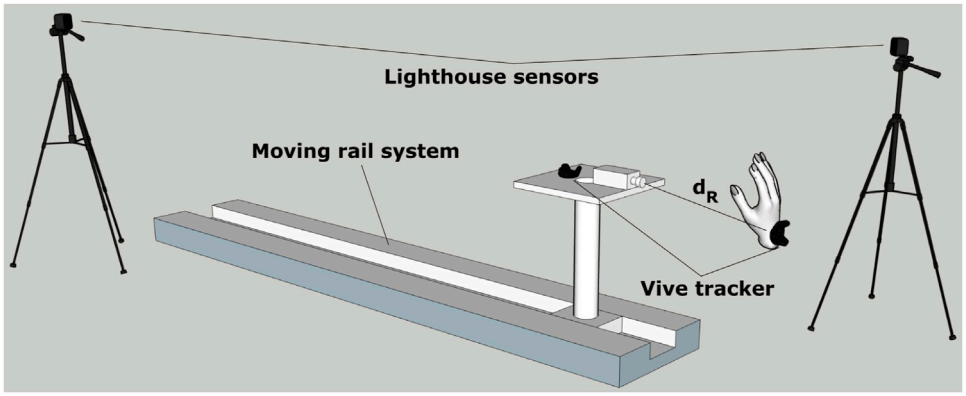


FIGURE 6
Sketch of the experimental setup.

4.1 Experimental setup

As hand tracking devices we used a Leap Motion sensor, an Oculus Quest 2 HMD with its integrated hand tracking, and a 1080p webcam with a 60° horizontal field-of-view and with MediaPipe as the tracking framework. Figure 6 shows a sketch of the experimental setup.

The tracking devices were mounted on a fixture that can move back and forth along a rail. The real-world position offset between the Vive tracker and the real-world position of the hand tracking device (as well as the offset between the Vive tracker and the real hand) was measured and taken into account in the virtual world calculations. The rail is 4 m long, which was sufficient for the maximum tracking distance of the evaluated methods. Vive Lighthouse base stations were positioned around the rail system. One Vive tracker was attached to a fixed position on the bracket and the other to the hand to be detected. Offsets to the tracking devices and the center of the hand were measured and added to the respective positions in the evaluation.

The VR application for collecting evaluation data was developed with Unity3D (v.2021.2.14). The rendering of tracked user hands was achieved with the help of an in-house developed framework that provides a universal layer for collecting and distributing hand tracking data obtained from any input source. For hand tracking the following versions of the hand tracking API were used: Oculus Integration v.38, Ultraleap Plugin v.5.4.0 and MediaPipeUnityPlugin[4] v.0.8.3 with MediaPipe backend v.0.8.9.

4.2 Experiment 1: hand tracking accuracy in comparison with integrated solutions

In this experiment, we access the performance of our hand tracking method based on RGB input by comparing it to the methods integrated into Oculus Quest and Leap Motion. All selected solutions can be combined with HMDs and are therefore in principle suitable for use in a colocated VR setup. We evaluate three variants of our RGB-based hand tracking method:

- **MediaPipeInternal:** Hand size is calculated based on 3D landmarks (in the coordinate frame of the hand center) detected by MediaPipe.
- **MediaPipeHand:** Hand size is given as input, following a physical measurement of the user's hand.
- **MediaPipeBody:** Hand size is approximated from the body height that is given as input.

Since our MediaPipe-based method is not fine-tuned for a specific interaction range, in contrast to Leap Motion and Oculus Quest, we do not expect it to be more accurate than those methods in the close range (within the arm's length). Nevertheless, we expect to be able to cover a larger tracking range with MediaPipe hand tracking and to achieve higher tracking accuracy at distances beyond arm's length. We expect our method to provide a usable tracking capability that works with a simple RGB camera, is simple to set up and has the ability to detect more than two hands at a time.

We analyze the following metrics:

- Static distance error: Error in the distance of the hand from the tracking device, compared to the ground-truth, while the hand is held still in one position. This metric is calculated at different distances from the tracking device.
- Dynamic distance error: Error in the distance of the hand from the tracking device, compared to the ground-truth, while the hand is moving away relative to the tracking device. We analyze the correlation between the dynamic error and the distance to the tracking device for all methods.
- Tracking lost and tracking acquired distances: Distance at which tracking is lost when the hand is moving away and the distance at which tracking is found when the hand is moving towards the tracking device.

The ground-truth distance d_r of the hand to the tracking device is measured with an externally mounted Lighthouse 2.0 tracking system,

4 https://github.com/homuler/MediaPipeUnityPlugin

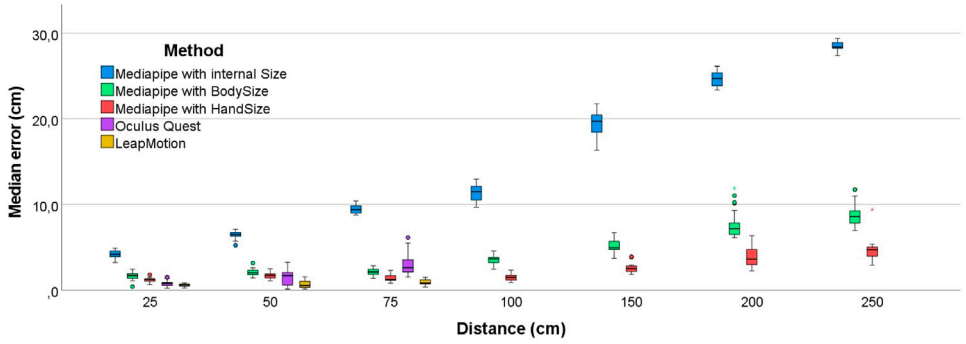


FIGURE 7
Box-plots for static data collection for each method and distance.

the tracking accuracy of which has been shown to be in the millimeter range with high replicability of position measurements (Borges et al., 2018; Bauer et al., 2021). HTC Vive trackers were attached to the wrist of the hand and the tracking device, allowing the ground-truth distance d_r in the frame t to be calculated as the distance between two trackers, adjusted by the offset between the center of the tracker and the center of the hand and between the center of the second tracker and the center of the tracking device.

The virtual distance d_v of the virtual hand to the virtual camera is calculated from the hand tracking data. The absolute distance error $\delta(t)$ for a frame t can thus be calculated with

$$\delta(t) = |d_v - d_r| \tag{4}$$

4.2.1 Measuring static tracking error

For the static error measurement, the tracking device and the tracked hand of a user were positioned at fixed distances from each other with $d \in 25, 50, 75, 100, 150, 200, 250$ cm. Smaller intervals of 25 cm were chosen in the range where Leap Motion and Oculus Quest could consistently track the hand. For $d >= 100$, the sampling interval was increased to 50 cm since only MediaPipe was able to consistently track the hand for these distances.

At every sampling position, we collected hand tracking data over $N = 400$ consecutive frames. This was done 25 times per position for each tracking method (Quest, Leap Motion, and three variants of the MediaPipe method). Each sample containing tracking data of 400 consecutive frames was collapsed to its median value, to account for possible small movements of the user's hand. This way, our resulting evaluation sample for each tracking method consists of 25 median static error values per one sampling position. The collections were only carried out if there was consistent tracking during the 400 frames. For Oculus Quest and Leap Motion this worked in the range [25 cm; 75 cm] and for all variations of the MediaPipe method in the range [25 cm; 250 cm].

4.2.2 Results: static tracking error

A few outliers were observed, which likely resulted from the temporary tracking loss of the deployed ground truth trackers in the environment. As these outliers were not attributed to the hand tracking itself, they were excluded from the analysis to prevent any false influence on the results. Following the removal of outliers, the Shapiro-Wilk normality test was performed on all median error samples. Because not all median error distributions were normal and because sampling position ranges were different for the evaluated methods, we compare the median static error of the evaluated methods separately for each sampling position, using the non-parametric Independent-Samples Median test. We also analyze the impact of the distance to the tracking device on the median static error for each method in the non-parametric Friedman's 2-way ANOVA test. The corresponding box-plots are presented in Figure 7.

As expected, the tracking error for close distances [25 cm; 50 cm] is the lowest for Oculus Quest and Leap Motion. Interestingly, MediaPipeHand delivered a lower error of 1.53 cm than Oculus Quest with a distance of 75 cm to the camera ($p < 0.001$), which is still within the user's arm's reach. Except for Leap Motion, the other tracking methods show an increasing median error for increasing distances to the camera. This is in line with our results from the dynamic evaluation.

For distances greater than 75 cm, only the values of the MediaPipe methods can be compared. However, a significant difference in mean error can be found between all three methods for all distances ($p < 0.001$). The error for MediapipeInternal is significantly larger at all further sampling positions than for MediaPipeBody and MediaPipeHand. For large distances, the error of MediaPipeInternal increases to values significantly above 10 cm, which can be too inaccurate for precise interactions. The results show that this error can be significantly reduced by using the real hand size ($p < 0.001$). With a maximum error of 4.47 cm at a distance of 250 cm from the camera, reasonably precise interactions in virtual space at greater distances are also possible with this method. The derivation of the hand size by the body size also shows a significantly lower error. As expected, however, the accuracy is not quite as good as when the actual hand size is given. With a maximum error of 8.57 cm at a distance of 250 cm from the camera, the error is significantly larger, but still improves the initial recognition of MediaPipe by more than three times (with an initial error of 28.48 cm). A summary of the resulting mean-, median- and p-values can be found in the Supplementary Material.

These results show that the accuracy of MediaPipe tracking in 3D space can be significantly improved by inputting the user's real hand size and allowing 3D interactions for large distances, which is not possible for the Leap Motion sensor or the Oculus Quest hand tracking.

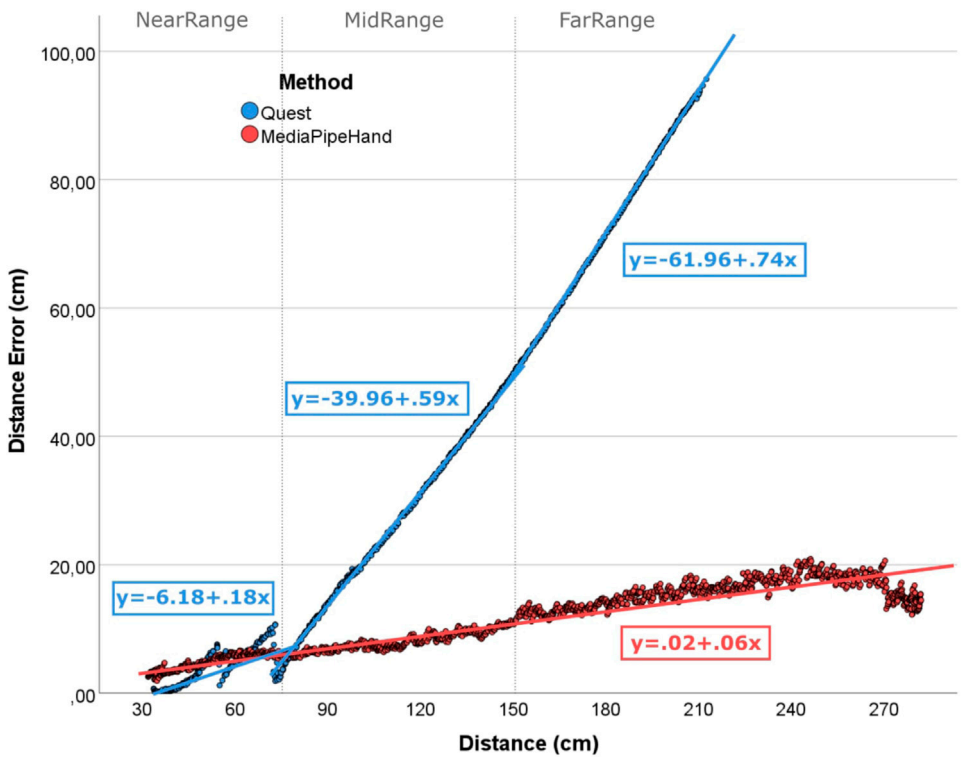


FIGURE 8
Scatter plots of two example dynamic error distributions for OculusQuest and MediaPipeHand. Regression lines for OculusQuest data are illustrated separately for **NearRange**, **MidRange** and **FarRange** due to the rising gradients in each range. The linear equations for the regression lines are in $\frac{cm}{cm}$. Quest median error shows a higher rising error for large distances.

4.2.3 Measuring dynamic tracking error

For the dynamic evaluation, the user held his hand at a fixed position while the tracking device moved away from it on a rail system (starting at the distance of 25 cm), at a constant speed up to a distance at the edge of the device tracking range. While a user walking away from the tracking device would have presented a more ecological hand tracking situation, the rail system was used in order to ensure repetitiveness and uniformity of data collection for this technical evaluation. The movement range was: for the Leap Motion sensor $r \rightarrow [0.25; 0.75]$, for Oculus Quest $r \rightarrow [0.25; 1.75]$ and for MediaPipe $r \rightarrow [0.25; 2.75]$. This procedure was repeated 10 times for each tracking method. Dynamic error data resulting from these recordings were averaged and analyzed according to a procedure described in detail in the following section.

4.2.4 Results: dynamic tracking error

For each frame, we get the real-world distance d_r and the virtual world distance d_v of the hand to the camera. Tracking error for was calculated with Eq. 4 and paired with the real-world distance d_r.Examples of dynamic error data samples for OculusQuest and MediaPipeHand prepared in this way are illustrated in the scatter plot in Figure 8, where the total range of the collection is shown.

The discretized dynamic error distributions are used to perform linear regression, with the gradient of the fitted regression line determining the rate of the error increase with distance from the tracking device. Since the tracking ranges of the evaluated tracking methods are different, we perform the linear regression separately in three distance ranges: **NearRange** [< 75 cm], **MidRange** [75 cm; 150 cm] and **FarRange** [>150 cm]. This procedure results in 10-entry distributions of regression coefficients for every tracking method, in distance ranges covered by the method. We can now compare all five tracking methods in **NearRange**, Oculus Quest and three MediaPipe variants in **MidRange**, and three MediaPipe methods in **FarRange**. We use one-way ANOVA to compare mean regression coefficient values between the methods within each distance range (data is normally distributed in the Shapiro-Wilk test).

The resulting ANOVA plots are shown in Figure 9. The scaling of the gradients is $\frac{cm}{cm}$, showing the increase of the tracking error in cm per each cm of distance from the tracking device.

In **NearRange**, Leap Motion shows the most consistent tracking with a mean gradient of 0.0007. This is a significantly smaller error increase rate than for the Oculus Quest and the MediaPipe methods. Oculus Quest has the steepest error increase rate with the distance with a mean gradient of 0.213 and shows significant differences to MediaPipeBody ($p = 0.006$) and MediaPipeHand ($p = 0.002$). The mean gradient of 0.098 for MediaPipeHand is the lowest for the three MediaPipe methods, with a mean gradient of 0.117 for MediaPipeBody and 0.162 for MediaPipeInternal. The statistical analysis, along with a substantial effect size of $\eta^2 = 0.827$ (as measured by eta-squared), reveals significant differences among all MediaPipe methods, except for MediaPipeHand and MediaPipeBody. Further information regarding means and pairwise comparisons can be found in the Supplementary Material.

MidRange shows significant differences ($p < 0.001$) and an effect size of $\eta^2 = 0.992$ for all pairwise comparisons of mean gradients,

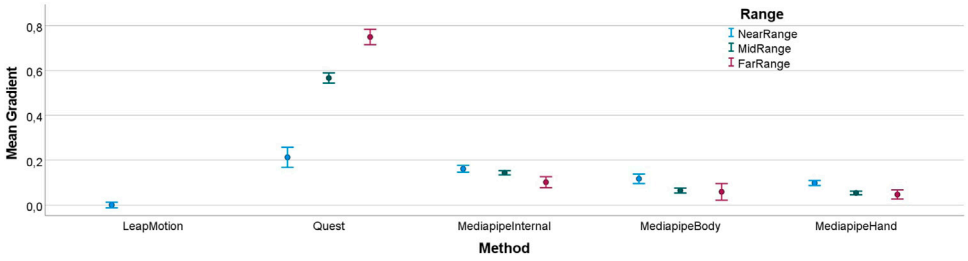


FIGURE 9
Mean gradient values for each method and distance range. Gradients are in $\frac{cm}{cm}$.

except (as in **NearRange**) between MediaPipeHand and MediaPipeBody (p = 0.333). With a gradient of 0.567, the Quest has the largest gradient here, which more than doubled compared to **NearRange**. Since the gradient of MediaPipeHand with 0.054 and MediaPipeBody with 0.064 has decreased somewhat compared to the **NearRange**, the tracking error in these areas does not increase as much. In comparison, the gradient of MediaPipeInternal with 0.144 is similarly high as in **NearRange**. This shows that the external input of hand size (whether measured or by body size) improves the tracking error. A significant difference between hand size by measurement and by body size cannot be found in **MidRange**.

In **FarRange**, the Oculus Quest again shows a higher gradient compared to the closer ranges (with a mean gradient of 0.749). This is again significantly higher than in the MediaPipe methods (p < 0.001). This shows a strongly increasing error for Oculus Quest and large distances and that the hand tracking of the Quest does not seem to be aligned for these distances. The MediaPipe methods again show similar mean gradients as in **MidRange**. With a mean gradient of 0.102, MediaPipeInternal shows the largest change in tracking error among the 3 methods, with a significant difference from MediaPipeHand (p = 0.006; a significant difference from MediaPipeBody could not be shown at p = 0.170). With a mean gradient of 0.059 for MediaPipeBody and 0.047 for MediaPipeHand, these two gradients are close to each other. As there is again no significant difference for **FarRange** between these two (p = 0.915), it can be generally stated that the increasing error can be significantly improved by entering the hand size compared to the internal hand size, but no significant difference could be determined between hand size calculation by measurement and by derivation by body size. The one-way ANOVA analysis yielded an effect size of η^2 = 0.98, as measured by eta-squared. This substantial effect size indicates a strong influence of the methods on the observed differences in the gradients. The findings highlight the significant impact that the choice of method has on the measured outcomes.

Figure 8 shows the distributed points with its linear regression lines where one can see, how fast the error rises for Oculus Quest in comparison to the MediaPipe method with inputting the measured real hand size.

4.2.5 Lost tracking distance

For each tracking method, we recorded the distance at which the tracking device loses the hand while being moved away. To do this, the user positioned his hand at a distance where it was reliably tracked. After the hand was tracked for at least 1 s, the device was moved away from the hand and the moment in which the device lost the hand was recorded. To avoid recordings for moments in which the tracking was only briefly disrupted, the hand had to be lost for at least one second. This procedure was repeated 25 times for each tracking method.

4.2.6 Results: lost tracking distance

A total of 25 data points were collected for each tracking method. The medians proved to be normally distributed in the Shapiro-Wilk test (Shapiro and Wilk, 1965) after three outliers (caused by interferences in the ground truth tracking) had been removed (p = 0.229 for the MediaPipe tracking, p = 0.557 for the Leap Motion, p = 0.293 for the Quest tracking). The resulting boxplots can be seen in Figure 10.

For the main analysis, Welch's test (Welch, 1947) was employed, yielding a significance level of p < 0.001 and an effect size of η^2 = 0.827. Subsequently, the post-hoc analysis was conducted using Games-Howell test. The findings from the post-hoc analysis demonstrated that the mean lost tracking distance for the MediaPipe tracking method (mean distance of 412.48 cm) was significantly greater than that of all other methods (p < 0.001). The mean lost tracking distance of the Oculus Quest was significantly larger than with Leap Motion (p < 0.001), but also significantly lower than with MediaPipe tracking (p < 0.001). The lowest mean distance where the tracking is lost is the lowest for Leap Motion with 95.6 cm. With 237.78 cm the Oculus Quest has quite a large tracking range, but also has severe tracking errors at longer distances, as can be seen in the previous sections. The detailed results can be found in the Supplementary Material.

4.2.7 Acquired tracking distance

The measurement of the distance at which the tracking device acquires the tracking of the hand follows a similar procedure. The device was set at a distance where the hand is not detected. After the program ensured that the hand was not tracked for at least 1 s, the device was moved toward the hand until the hand was tracked. Again, the hand has to be tracked for at least 1 s to record the distance. The procedure was repeated 25 times per method.

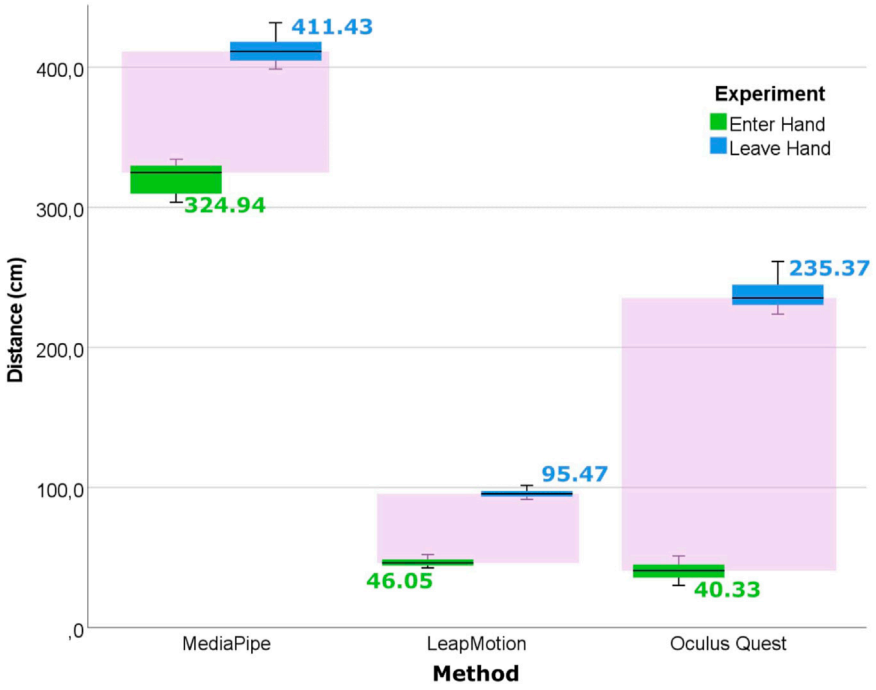


FIGURE 10
Box-plots medians for lost and acquired tracking distance for tracking methods. Labels are median values.

4.2.8 Results: acquired tracking distance

Distributions of recorded distances at which tracking was acquired deviated from normal (in the Shapiro-Wilk normality test) for two out of three tracking methods; therefore, median tracking acquired distance values were compared in the Independent-Samples Median test. The corresponding box-plots are illustrated in Figure 10.

The pairwise comparisons of the Independent-Samples Median test demonstrate significant differences between the methods, with a notable effect size (as measured by eta-squared) of $\eta^2 = 0.827$. These comparisons reveal that the median distance of 324.94 cm for MediaPipe is significantly greater than that of Leap Motion, which measures 46.05 cm ($p < 0.001$), as well as Oculus Quest, with a median distance of 40.33 cm ($p < 0.001$). The difference in acquired tracking distance between Oculus Quest and Leap Motion is also statistically significant ($p = 0.008$), although the corresponding median values are much closer (with a difference of 5.72 cm). The fact that MediaPipe consistently acquires hand tracking in a range of approximately 3 m shows that this method could be used for larger areas while Leap Motion and Oculus Quest are limited to near-range distances within arm's length. The full statistical results can be found in the Supplementary Material.

4.3 Experiment 2: accuracy MediaPipe-based hand tracking on data of multiple users

The primary objective of this second experiment was to validate the effectiveness of our MediaPipe-based method for various users and to assess the influence of hand size estimation on the accuracy of the 3D positioning error in greater depth. While the first experiment primarily focused on performance evaluation and method comparison, the second experiment aimed to explore the application of our method with multiple users. Although it did not involve an extensive user study with joint-error measurements, it served to confirm the usability of our method and highlight other impacts on positioning errors.

In this experiment, we repeated the procedure for static and dynamic tracking error measurement from **Experiment 1**, this time focusing only on MediaPipeInternal, MediaPipeBody, and MediaPipeHand methods. For the static error measurement, four hand tracking data recordings (400 frames each) per distance per user user were made at distances $d \in 25$, 50, 75, 100, 150, 200, 250 cm between the tracking device and the user's hand. One recording of dynamic tracking error data per user (for each method) was conducted. The calculations of the distance error in the static and dynamic conditioned were the same as in the previous experiment.

The hardware and software setup used for data recording was the same one as in **Experiment 1**. In total, hand tracking data was collected from 9 users, 3 female and 6 male, ranging in age from 20 to 56 years. Including measurements of body and hand size and introduction, the procedure took about 45 min per user.

4.3.1 Results: static tracking error

Figure 11 presents box plots illustrating the average static error of MediaPipeInternal, MediaPipeBody, and MediaPipeHand at different distances. To maintain a consistent analysis metric and account for the relatively low variance observed in the four median error values obtained for each user at each distance, we consolidated these errors by calculating their median value and employed this

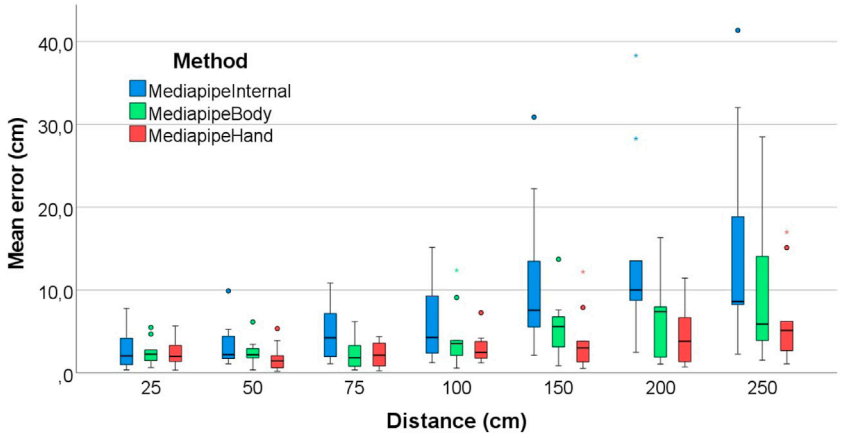


FIGURE 11
Box-plots of the mean error at every measured distance in the test with multiple users.

value for the analysis. As evident from the results, outliers are observed, which may have been caused by temporary tracking losses of the ground truth system. Despite their presence, these outliers were retained in the analysis because it could not be guaranteed that their removal would be advantageous for maintaining the integrity of the test, as previously mentioned.

With the data deviating from the normal distribution in many cases, we again used the Independent-Samples Median Test to find whether the error depends on the hand tracking method at each distance. The method was statistically significant for the error at the distance of 200 cm (p = 0.016; follow-up pairwise comparisons did not find any statistical significance), with the result at the distance of 250 cm being marginally below statistical significance. From the pattern seen in the box-plots with MediaPipeInternal producing visibly higher errors at larger distances, we believe that more detailed results could be achieved with a larger sample size of users. Nevertheless, the available data supports the results of the main analysis. The detailed results can be found in the Supplementary Material.

Analyzing the error dependence on the distance from the tracking camera for each method, we found that the error increases with the distance for MediaPipeInternal (p = 0.002 in Independent-Samples Median Test) and MediaPipeBody (p = 0.029). For MediaPipeHand, no statistically significant dependency of an error on the distance could be found.

4.3.2 Results: dynamic tracking error

We used the same approach for calculating the gradients of error change in the near, middle and far range on the evaluation data of multiple users. Figure 12 shows the gradients for each method in **NearRange**, **MidRange** and **FarRange**.

We used Mixed ANOVA with the range as a repeated-measures factor (with three levels) and the method as a between-subject factor (also with three levels). In this analysis, only the range was found to be statistically significant (F = 4.683, p = 0.014), with within-subject repeated contrasts showing the increase of gradient from **MidRange** (mean = 0.035) to **FarRange** (mean = 0.112), p = 0.002. Figure 12 shows mean gradients for each tested range and method.

4.4 Experiment 3: pilot demonstration in a colocated scenario

The goal of this last experiment was to test the usability of our hand tracking method in a real-world colocated VR scenario. To do this, we developed a simple VR application in which two users can see each other's avatars consisting of the virtual HMD and hands steered by head and hand tracking input. The aim was not to create a detailed qualitative user analysis, but rather a proof-of-concept demonstration.

In our setup, each user has an HTC Vive tracker attached to their hand, which provides position and rotation data of the real hand positions in the space of the Lighthouse 2.0 Tracking and serves as ground-truth. Both users have a VR headset on (we used Oculus Quest with its own hand tracking feature turned off). User 1 has an RGB camera attached to the HMD (in this scenario a ZED mini camera that only provided the RGB image of one lens). With this camera, all hands were detected and visualized in the shared virtual environment. Since only User 1 had a running hand detection system, it was ensured that only one system tracked all hands and placed them in the virtual space. The users stood facing each other at a distance of about 1.5 m and held their hands in the tracking area of the RGB camera. The camera simultaneously detected and tracked the hands of User 1 and User 2. In the virtual environment, the user's hands were positioned with the hand tracking input. Our proof-of-concept colocated VR scenario can be seen in the Supplementary Video S1 in the supplemental materials.

4.4.1 Results: experiment 3

Figure 13 shows a snapshot of the scenario experienced by users in the experiment. The RGB image of the tracking camera is overlayed with the virtual scene seen in VR to give a better illustration of hand tracking. The plane that can be seen in Figure 13 is the floor plane in the virtual environment. Since only User 1 had a hand tracking device attached to his HMD, the view of all virtual hands, his own and those of User 2, is enabled by his hand tracking camera. The same virtual hands were displayed to User 2 and animated by hand tracking data.

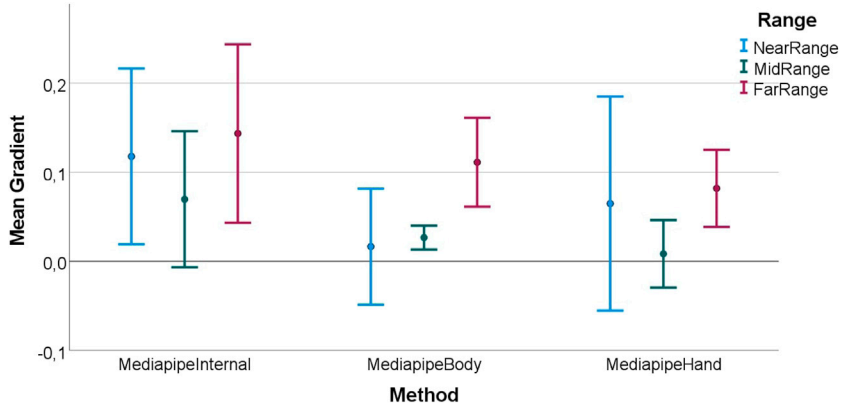


FIGURE 12
Mean gradient values for each method and distance range in **Experiment 2**. Gradients are in $\frac{cm}{cm}$.

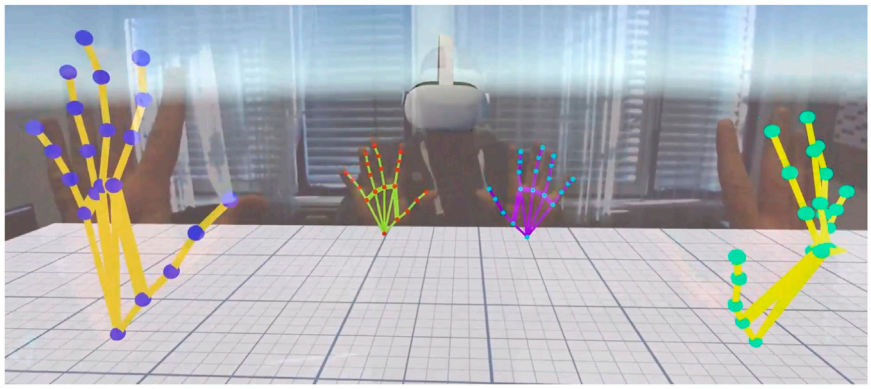


FIGURE 13
POV of the user with the tracking device attached tracking also the hands of another user. Both users are colocated, the real-world view of the camera can be seen slightly overlayed.

Hand tracking data was recorded over a period of about 30 s, resulting in snapshots of 1242 consecutive frames. Detected hands were assigned in pairs to the corresponding Vive trackers, and the deviation from the ground truth tracker position was calculated as tracking error. The position errors for each user are plotted in Figure 14. Peaks and missing values in the plot are moments where corresponding hands were not tracked for a moment. Since a smoothing filter is used when applying tracking input to virtual hands for smoother movements, inaccurate positioning can occur right before losing or after acquiring tracking.

The calculated mean error for the hands of User 1 was 5.1 cm with a standard error of 0.154. The 95% confidence interval provides a lower limit of 4.8 and an upper limit of 5.41 cm. For user 2, we obtain a mean of 8.21 cm with a standard error of 0.17, a lower limit of 7.88, and an upper limit of 8.55 cm for the 95% confidence interval.

The higher position error values for User 2 were expected since User 2 was further away from the camera. The values correspond to the expectations of a more realistic scenario based on the results of the dynamic and static tracking error evaluations.

Although we calculated the error of the tracked hands in this scenario, it is important to note that a single user pair is insufficient to provide a comprehensive user study for real-world colocated applications. Nevertheless, we aim to demonstrate that our method is not limited to controlled scenarios with fixed machinery but can also be applied in real colocated setups involving two users. To further evaluate the presented method, it would be valuable to conduct a qualitative user study that measures interaction precision, usability, and user experience. Such a study would provide an interesting avenue for future research and a more comprehensive evaluation of our approach.

5 Discussion

The results of **Experiment 1** show that Oculus Quest and Leap Motion are more accurate at arm's length range than RGB input-based tracking with MediaPipe, which was to be expected. The tracking errors for Oculus Quest and Leap Motion are in line with previous research results (Schneider et al., 2020; Abdlkarim et al.,

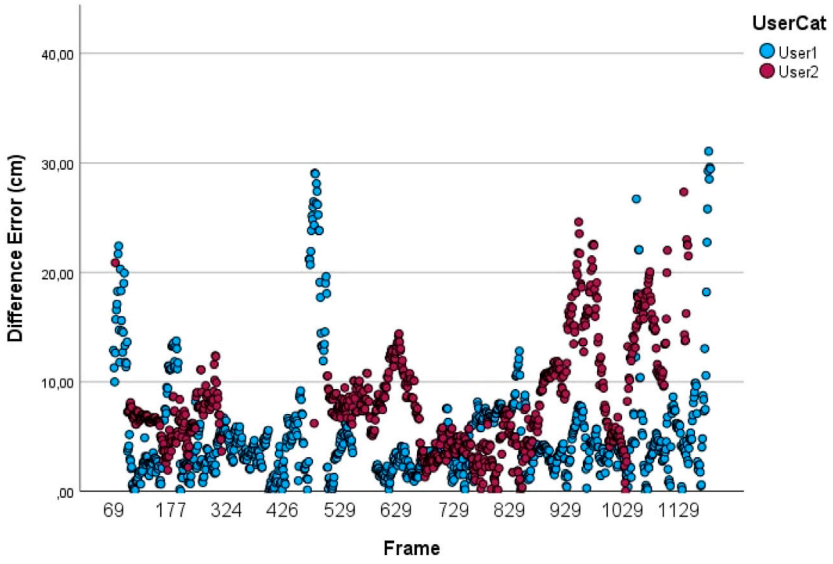


FIGURE 14
Error of hand pairs in centimeters during the preliminary user test. Spikes can occur due to tracking losses and wrongly positioning due to a smoothing filter.

2022). However, if the user's hand length is used to improve the MediaPipe-based method (either being estimated based on the height or entered from a real measurement), RGB tracking delivers comparable accuracy with a distance error in the range of 12.4 mm–21.3 mm (see the lowest and largest error in Figure 7 for MediaPipeHand and MediaPipeBody in distances between 25 cm and 75 cm). This shows that in this distance range the improved RGB hand recognition offers interactions with similar precision as the off-the-shelf solutions.

At the best value in the close range of 25 cm, the error of the RGB method with a real measured hand is 3.36 times lower than without external input. In the outer range at a distance of 250 cm even by a factor of 6. Due to the improvement through external hand size input, this even allows interactions in ranges of 2.5 m, which is not possible with Oculus Quest or Leap Motion. This also qualifies this method for use in colocated multi-user scenarios, although these would need to be investigated in more detail together.

The results of the acquired and lost tracking distance show that Oculus Quest and Leap Motion are in a similar range at which distance a hand is recognized for the first time. This makes sense in the sense that both systems were designed for interactions in the hand length range. In comparison, MediaPipe has an almost eight times higher distance at which the hand is recognized for the first time, which also allows for a significantly higher range and more possibilities, for example, to recognize the hands of other users that are further away from the own user.

After a hand has been detected once, all methods have a greater distance where the tracking of the hand is lost again. It is striking that Leap Motion only allows hand detection in the areas where tracking is guaranteed to be as accurate as possible, whereas the Oculus Quest goes beyond this range and continues tracking the detected hand up to a distance of 235 cm. These differences between acquired and lost tracking are shown by the purple boxes in Figure 10. This lack of limitation of the Oculus Quest for distances beyond arm's length also leads to very strongly increasing tracking errors in this case. This makes the Quest unusable due to high tracking errors in **Mid-** and **FarRange** detection, even though hand tracking would be possible for the system in this range.

This is also reflected in the results of dynamic tracking. Within the tracking range, Leap Motion has significantly fewer discrepancies in tracking error than, for example, the Oculus Quest. The latter shows significantly increasing tracking errors, especially in tracking outside the **NearRange**.

MediaPipe is much more consistent here, even though the tracking error still increases as the distance increases. However, this slope is nowhere near as steep as in Oculus Quest. Even between the different presented depth estimations for MediaPipe, the increasing error is significantly lower in the variants where the user's hand size was added externally. Thus, the error is less at larger distances. This is also reflected in the results of the static evaluation. This is a further indication that RGB methods (such as MediaPipe) can be improved and are thus suitable for enabling hand tracking and hand interactions even at greater distances.

During the real-world application, we observed that the computation time and occurrences of temporal tracking losses increased as the number of simultaneously detected hands increased. This behavior aligns with expectations for an image-tracking system designed for multi-object recognition. While the presence of four hands within the tracking frame did not cause significant issues, the limitations of the current state of the MediaPipe system became more apparent as the number of hands increased. However, it is important to note that these observations did not impact our calculations, and we anticipate that future versions of the framework will address these limitations and enhance stability. This information is mentioned here for the sake of completeness.

As has also been shown, the way the real hand size is determined has a significant impact on tracking accuracy. The more accurately

the hand size is determined, the smaller the error. Unfortunately, the 3D coordinates of the finger joints provided by MediaPipe did not accurately match the real hand size and the size had to be input externally to effectively improve depth calculation.

The results of **Experiment 2** largely reflect the pattern of the main analysis, although significance was not shown everywhere. The error of the 3D hand positioning increases with the distance to the tracking camera. However, the error can be reduced (especially for high distances) by inputting the users' hand size. We believe that a larger data set and extended user testing (which unfortunately was not possible at this time) would increase the significance and further increase the level of detail in the results.

Furthermore, it can be seen that when deriving from body size to hand size for 10 users (from **Experiment 1** and **Experiment 2**), the deviation of the calculated hand size to the actual hand length was less than 1 cm for all users. The algorithm can be used as an alternative when the user's body size is known rather than their hand length. For future applications, it would be interesting to find a method to perform this measurement more universally and without external input. One possibility would be a calibration step at the beginning of the application, where the hand is measured at a certain distance, or where the body size is determined based on the height of the VR headset and then the hand size is derived. For applications where full-body avatars are used, for example, these could also be used to improve the tracking error that still exists. If they are scaled to the user's size, their maximum arm length can be used as an improvement metric to position the hand more accurately to the avatar.

The mean hand position errors calculated in **Experiment 3** are in line with the results from the controlled static and dynamic tests. With continuously detected hands, consistent positioning can take place. In the test, we noticed that the quality of positioning is also dependent on the quality of the underlying tracking (MediaPipe in this case). A more consistent and better recognition in the future also improves the real error of our algorithm. Coupled with a high-resolution camera, consistent multi-user hand tracking in colocated rooms can be realized.Further user tests regarding user experience would be interesting in the future. Furthermore, this kind of multi-user hand tracking raises another problem. The recognized hands have to be reliably assigned to the virtual users. This would be a task for future work. Approaches to this already exist, such as by Tsutsui et al. (2020), but these are so far limited to a two-dimensional image and are not applied in three-dimensional space.

6 Conclusion

This study evaluated the tracking error and tracking range of three different hand tracking technologies, one of which works via RGB cameras and is not tied to a specific manufacturer's hardware. In addition, a method was developed that significantly improves the tracking result of RGB hand tracking.

The evaluation shows that although the hand tracking of Oculus Quest and Leap Motion is more accurate in the arm's length range, they are not designed for hand detection outside this range. Therefore, we achieve comparatively higher precision at larger distances with the RGB method. This can be further increased if additional information about the user's hand length is used in the calculation of the 3D positions. Direct measurement of the hand length is more precise, but deriving the hand length from the user's height also produces acceptable tracking errors. For more general use, the body height input is probably more intuitive, as many people know their body height rather than their hand length. It might even be possible to automatically determine the body height in a virtual reality application through an initialization phase. This would be a use case for future research.

It has also been shown that hand tracking using the (improved) RGB method has significantly higher tracking ranges (with usable distance well above arm's length range) as well as the ability to track more than just two hands at a time. This can be especially useful for hand recognition in larger tracking areas with multiple users. Thus, we have also shown that by inputting the user's real hand size, a tracking system based on a single image RGB camera can provide results that provide a tracking error that allows interactions in virtual space, as well as significantly expand the tracking range for hands. This method could be superior to the presented commercial systems such as Leap Motion or Oculus Quest for specific scenarios as colocated multi-user scenarios, where one also wants to track the hands of the other users.

Therefore, it would be interesting in future experiments to see the effects of such RGB tracking in multi-user VR scenarios and to find out how it can disrupt or improve hand tracking in such applications. Since our setup and use case focuses on a single camera for tracking, using a camera array to improve tracking of colocated users' hands would be an interesting extension for the future.

Data availability statement

The original contributions presented in the study are included in the article/Supplementary Material, further inquiries can be directed to the corresponding author.

Ethics statement

Ethical review and approval was not required for the study on human participants in accordance with the local legislation and institutional requirements. The patients/participants provided their written informed consent to participate in this study. Written informed consent was obtained from the individual(s) for the publication of any potentially identifiable images or data included in this article.

Author contributions

The authors in this research contributed to this article with the following: Conceptualization, DR; methodology, DR and IP; software and implementation, DR; validation, DR; formal analysis, DR and IP; investigation, DR; data curation, DR; writing—original draft preparation, DR; writing—review and editing, DR, IP, HK, and DS; visualization, DR; supervision, HK

and DS; project administration, DR. All authors contributed to the article and approved the submitted version.

Funding

The authors acknowledge TU Wien Bibliothek for financial support through its Open Access Funding Program.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/frvir.2023.1169313/full#supplementary-material

SUPPLEMENTARY TABLE S1
Statistical results for evaluating dynamic tracking error in the main analysis.

SUPPLEMENTARY TABLE S2
Statistical results for evaluating static tracking error in the main analysis.

SUPPLEMENTARY TABLE S3
Body Part sizes of users participating in the user evaluation.

SUPPLEMENTARY TABLE S4
Statistical results for evaluating lost and acquired tracking distances for the hand.

SUPPLEMENTARY VIDEO S1
Comparison of our multi-hand tracking solution with off-the-shelf tracking solutions.

SUPPLEMENTARY DATA S1
Statistical results for evaluating dynamic tracking error in the user test.

SUPPLEMENTARY DATA S2
Statistical results for evaluating static tracking error in the user test.

SUPPLEMENTARY DATA S3
Results for pilot demonstration in a colocated VR setup.

References

Abdlkarim, D., Di Luca, M., Aves, P., Yeo, S.-H., Miall, R. C., Holland, P., et al. (2022). A methodological framework to assess the accuracy of virtual reality hand-tracking systems: A case study with the oculus quest 2. *bioRxiv*. doi:10.1101/2022.02.18.481001

Bauer, P., Lienhart, W., and Jost, S. (2021). Accuracy investigation of the pose determination of a vr system. *Sensors* 21, 1622. doi:10.3390/s21051622

Borges, M., Symington, A., Coltin, B., Smith, T., and Ventura, R. (2018). "Htc vive: Analysis and accuracy improvement," in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2610–2615. doi:10.1109/IROS.2018.8593707

Che, Y., and Qi, Y. (2021). "Detection-guided 3d hand tracking for mobile ar applications," in 2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), 386–392. doi:10.1109/ISMAR52148.2021.00055

Ding, Q., Ding, J., Zhang, J., and Du, L. (2020). An attempt to relate dynamic tracking error to occurring situation based on additional rectilinear motion for five-axis machine tools. *Adv. Mech. Eng.* 12, 168781402096757. doi:10.1177/1687814020967573

Ferstl, Y., McDonnell, R., and Neff, M. (2021). "Evaluating study design and strategies for mitigating the impact of hand tracking loss," in ACM Symposium on Applied Perception 2021 (New York, NY, USA: Association for Computing Machinery). SAP '21. doi:10.1145/3474451.3476235

Gong, L., Söderlund, H., Bogojevic, L., Chen, X., Berce, A., Fast-Berglund, A., et al. (2020). Interaction design for multi-user virtual reality systems: An automotive case study. *Procedia CIRP* 93, 1259–1264. doi:10.1016/j.procir.2020.04.036

Han, S., Liu, B., Cabezas, R., Twigg, C. D., Zhang, P., Petkau, J., et al. (2020). Megatrack: Monochrome egocentric articulated hand-tracking for virtual reality. *ACM Trans. Graph.* 39. doi:10.1145/3386569.3392452

Huang, L., Zhang, B., Guo, Z., Xiao, Y., Cao, Z., and Yuan, J. (2021). Survey on depth and rgb image-based 3d hand shape and pose estimation. *Virtual Real. Intell. Hardw.* 3, 207–234. doi:10.1016/j.vrih.2021.05.002

Khundam, C., Vorachart, V., Preeyawongsakul, P., Hosap, W., and Noël, F. (2021). A comparative study of interaction time and usability of using controllers and hand tracking in virtual reality training. *Informatics* 8, 60. doi:10.3390/informatics8030060

Li, Y., Ch'ng, E., Cai, S., and See, S. (2018). "Multiuser interaction with hybrid vr and ar for cultural heritage objects," in 2018 3rd Digital Heritage International Congress (DigitalHERITAGE) held jointly with 2018 24th International Conference on Virtual Systems & Multimedia (VSMM 2018), 1–8. doi:10.1109/DigitalHeritage.2018.8810126

Lin, F., Wilhelm, C., and Martinez, T. R. (2020). Two-hand global 3d pose estimation using monocular RGB. CoRR abs/2006.01320.

Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., et al. (2019). Mediapipe: A framework for building perception pipelines. CoRR abs/1906.08172.

Malik, J., Elhayek, A., Nunnari, F., Varanasi, K., Tamaddon, K., Héloir, A., et al. (2018). Deephps: End-to-end estimation of 3d hand pose and shape by learning from synthetic depth. CoRR abs/1808.09208.

Mizera, C., Delrieu, T., Weistroffer, V., Andriot, C., Decatoire, A., and Gazeau, J.-P. (2020). Evaluation of hand-tracking systems in teleoperation and virtual dexterous manipulation. *IEEE Sensors J.* 20, 1642–1655. doi:10.1109/JSEN.2019.2947612

Panteleris, P., Oikonomidis, I., and Argyros, A. A. (2017). Using a single RGB frame for real time 3d hand pose estimation in the wild. CoRR abs/1712.03866.

Pheasant, S. (2003). *Bodyspace: Anthropometry, ergonomics and the design of work*. 2 edn. London: CRC Press.

Schneider, D., Otte, A., Kublin, A. S., Martschenko, A., Kristensson, P. O., Ofek, E., et al. (2020). "Accuracy of commodity finger tracking systems for virtual reality head-mounted displays," in 2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW), 804–805. doi:10.1109/VRW50115.2020.00253

Schneider, D., Biener, V., Otte, A., Gesslein, T., Gagel, P., Campos, C., et al. (2021). Accuracy evaluation of touch tasks in commodity virtual and augmented reality head-mounted displays. CoRR abs/2109.10607.

Schupp, H. (1977). *Elementargeometrie*. Schöningh. No. Bd. 1 in Grundkurs Mathematik.

Shapiro, S. S., and Wilk, M. B. (1965). An analysis of variance test for normality (complete samples). *Biometrika* 52, 591–611. doi:10.1093/biomet/52.3-4.591

Sharp, T., Keskin, C., Robertson, D., Taylor, J., Shotton, J., Kim, D., et al. (2015). "Accurate, robust, and flexible real-time hand tracking," in Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (New York, NY, USA: Association for Computing Machinery), 3633–3642. CHI '15. doi:10.1145/2702123.2702179

Sun, Z., Hu, Y., and Shen, X. (2021). "Two-hand pose estimation from the non-cropped rgb image with self-attention based network," in 2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR) (Los Alamitos, CA, USA: IEEE Computer Society), 248–255. doi:10.1109/ISMAR52148.2021.00040

Tsutsui, S., Fu, Y., and Crandall, D. (2020). Whose hand is this? Person identification from egocentric hand gestures. [Dataset]. doi:10.48550/ARXIV.2011.08900

Voigt-Antons, J.-N., Kojic, T., Ali, D., and Möller, S. (2020). "Influence of hand tracking as a way of interaction in virtual reality on user experience," in 2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX), 1–4. doi:10.1109/QoMEX48832.2020.9123085

Vysocký, A., Grushko, S., Oščádal, P., Kot, T., Babjak, J., Jánoš, R., et al. (2020). Analysis of precision and stability of hand tracking with leap motion sensor. *Sensors* 20, 4088. doi:10.3390/s20154088

Wang, Z., Mastrogiacomo, L., Franceschini, F., and Maropoulos, P. (2011). Experimental comparison of dynamic tracking performance of igps and laser tracker. *Int. J. Adv. Manuf. Technol.* 56, 205–213. doi:10.1007/s00170-011-3166-0

Wang, J., Mueller, F., Bernard, F., Sorli, S., Sotnychenko, O., Qian, N., et al. (2020). Rgb2hands: Real-time tracking of 3d hand interactions from monocular rgb video. *Arxiv* 39. doi:10.1145/3414685.3417852

Welch, B. L. (1947). The generalization of 'STUDENT'S' problem when several different population varlances are involved. *Biometrika* 34, 28–35. doi:10.1093/biomet/34.1-2.28

Zafar, U., Shafiq-Ur-RahmanHamid, N., Ahsan, J., and Zafar, N. (2017). Correlation between height and hand size, and predicting height on the basis of age, gender and hand size. *J. Med. Sci.* 25 (4), 425–428. Retrieved from https://jmedsci.com/index.php/Jmedsci/article/view/11.

Zhang, F., Bazarevsky, V., Vakunov, A., Tkachenka, A., Sung, G., Chang, C.-L., et al. (2020). Mediapipe hands: On-device real-time hand tracking. *Arxiv*. ArXiv abs/2006.10214. doi:10.48550/arXiv.2006.10214

Zimmermann, C., and Brox, T. (2017). "Learning to estimate 3d hand pose from single rgb images," in IEEE International Conference on Computer Vision (ICCV). Available at: Https://arxiv.org/abs/1705.01389.