sensors

MDPI

Article

Hand Tracking and Gesture Recognition Using Lensless Smart Sensors

Lizy Abraham *, Andrea Urru, Niccolò Normani, Mariusz P. Wilk, Michael Walsh and Brendan O'Flynn

Micro and Nano Systems Centre of Tyndall National Institute, University College Cork, Cork T12 R5CP, Ireland; andrea.urru@upf.edu (A.U.); niccolo.normani@gmail.com (N.N.); mariusz.wilk@tyndall.ie (M.P.W.); michael.walsh@tyndall.ie (M.W.); brendan.oflynn@tyndall.ie (B.O.)

* Correspondence: liszy.abraham@gmail.com; Tel.: +91-94-9512-3331

Received: 7 June 2018; Accepted: 18 August 2018; Published: 28 August 2018

Abstract: The Lensless Smart Sensor (LSS) developed by Rambus, Inc. is a low-power, low-cost visual sensing technology that captures information-rich optical data in a tiny form factor using a novel approach to optical sensing. The spiral gratings of LSS diffractive grating, coupled with sophisticated computational algorithms, allow point tracking down to millimeter-level accuracy. This work is focused on developing novel algorithms for the detection of multiple points and thereby enabling hand tracking and gesture recognition using the LSS. The algorithms are formulated based on geometrical and mathematical constraints around the placement of infrared light-emitting diodes (LEDs) on the hand. The developed techniques dynamically adapt the recognition and orientation of the hand and associated gestures. A detailed accuracy analysis for both hand tracking and gesture classification as a function of LED positions is conducted to validate the performance of the system. Our results indicate that the technology is a promising approach, as the current state-of-the-art focuses on human motion tracking that requires highly complex and expensive systems. A wearable, low-power, low-cost system could make a significant impact in this field, as it does not require complex hardware or additional sensors on the tracked segments.

Keywords: LSS; Infrared LEDs; Calibration; Tracking; gestures; RMSE; Repeatability; Temporal Noise; latency

1. Introduction

Three-dimensional (3D) ranging and hand tracking have a wide range of research applications in the field of computer vision, pattern recognition, and human–computer interaction. With the recent advancements of consumer-level virtual reality (VR) and augmented reality (AR) technologies, the ability to interact with the digital world using our hands gains significant importance. When compared to other body parts, tracking hand–finger movements is very challenging due to the fast motion, lack of visible textures, and severe self-occlusions of finger segments. The hand is one of the most complex and beautiful pieces of natural engineering in the human body. It gives us a powerful grip but also allows us to manipulate small objects with great precision. One reason for this versatility is its great flexibility, which allows it to move fast compared to other body parts such as legs or head, which is well explained in the work of Taylor et al. [1].

The evolution of hand tracking starts with conventional vision cameras. A detailed survey of various hand tracking methodologies and gesture recognition techniques using vision sensors are given in the works of Rautaray et al. [2]. The main drawback of vision cameras is their limitations in the two-dimensional (2D) domain, and in order to ensure consistent accuracy in tracking, it requires a focusing lens that makes the system bulky and expensive. Garg [3] used video cameras to recognize hand poses, ending in a highly complicated and inefficient process. Yang [4] suggested another method using vision sensors that also locate finger tips, but the method was not invariant to the orientation of the hand. Few other methods are dependent on specialized instruments and setup, such as the use of infrared cameras [5], and fixed background [6], which cannot be used in any environmental conditions. Stereoscopic camera systems make a good substitute for vision sensors, as they provide depth information about the scene, and thus the ability to reconstruct the hand in 3D in a more accurate fashion [7]. Yi-Ping et al. [8] suggested directionally variant templates to detect fingertips, which provide information about finger orientation. Some other works used the well-established Kalman filters together with depth data for more efficient tracking, leading to gesture recognition [9,10]. However, the uncontrolled environment, lighting conditions, skin color detection, rapid hand motion, and self-occlusions pose challenges to these algorithms when capturing and tracking the hand [11].

Over the past decades, a number of techniques based on glove technology have been explored to address the problem of hand–finger tracking. These can be categorized as data gloves, which use exclusive sensors [12], and marker-based gloves [13]. In some of the earlier works, the data gloves are constructed using accelerometers alone for the tracking [14,15], but in the recent works [16,17], data gloves are well equipped with inertial measurement units (IMUs) and magnetic field sensors that incorporate: an accelerometer, a gyroscope, and a magnetometer, whose combined outputs provide the exact orientation information. This allows for high dimensionality in terms of degrees of freedom, which indeed helps in tracking the movements of finger joints. Gloveone by Neuro Digital Technologies [18], VR Glove by Manus [19], and Noitom Hi5 VR Glove [20] are the predominant data gloves available in the market, which enable tracking all of the finger phalanges using a number of IMUs with low latency and often haptic feedback. Finger-joint tracking using hall-effect sensors [21] is another recently developed novel technology that provides higher joint resolutions than IMUs. However, the main drawback of these technologies is that they are not sufficient for providing exact location information, as the sensors are estimating only the orientation of the hands and fingers. Data gloves require complex algorithms to extract useful information from IMUs, and are not sufficient for providing exact location information. Instead of using sensors, gloves fitted with reflective markers or color patches are also developed, which provide a less complex solution than standard data capture gloves, but the device itself might inhibit free hand movements, and as such cannot be tracked precisely [22,23].

Recently, the various depth sensors in the market have become popular in the computer vision community exclusively for hand tracking and gesture recognition research activities. Leap Motion [24], Time-of-Flight (ToF) sensors [25], and Kinect depth sensors [26] are some of the commercially available depth sensors. The associated Software Development Kit (SDK) provided along with the sensors estimates the depth information [27–29] irrespective of the color values, which provides convenient information for gesture recognition [30].

The marker-less tracking of hands and fingers is another promising enabler for human–computer interaction. However, tracking inaccuracies, an incomplete coverage of motions, a low frame rate, complex camera setups, and high computational requirements are making it more inconvenient. Wang et al. [31] and Sridhar et al. [32] present remarkable works in this field, but with the expense of dense camera installations and high setup costs, which limits the deployment. Multi-camera imaging systems are also investigated that can estimate hand pose with high accuracy [33]. However, the calibration of multi-camera setups make these methods difficult to adopt for practical applications, as the natural hand movement involves fast motions with rapid changes and self-occlusions.

Most recently, new methods in the field of hand tracking and gesture recognition involve the utilization of a single depth camera as well as a fusion of inertial and depth sensors. Hand motion and position tracking are made possible by continuously processing 2D color and 3D depth images with very low latency. The approach uses strong priors and fails with complex self-occlusions, which often make it unsuitable for a real-time hand movement tracking, although it is widely accepted for tracking robotic trajectories [34]. The most recent and more predominant works in this field are by Sridhar et al. [35] and Tkach et al. [36], using either Intel Real Sense or Microsoft Kinect as a Red-Green-Blue-Depth (RGBD) camera. Another recent approach is the fusion of inertial and depth sensors [37,38], which is a practically feasible solution if very high accuracy in estimating 3D locations together with orientations are required.

In general, motion analysis using inertial measurement units (IMUs) requires complex hardware, as well as embedded algorithms. Conventional stereoscopic cameras or optical depth sensors used for high precision ranging are not suitable for a wearable system due to their high cost and impractical technical requirements. The other technologies that have been described require highly complex and expensive infrastructure. The applications of hand-tracking demand tracking across many camera-to-scene configurations, including desktop and wearable settings. As seen by the recent development of VR systems, such as the Oculus rift developed by Facebook, Google Glass by Google, or Microsoft HoloLens, the key requirement is the integration of mobility, low-power consumption, small form factor, and highly intelligent and adaptable sensor systems. Taking into account all of these constraints, this work is focused on the use of a novel Lensless Smart Sensor (LSS) developed by Rambus, Inc. [39]. The novel algorithms developed for this sensing technology allow the basic tracking of a number of infrared (IR) light-emitting diodes (LEDs) placed on key points of the hand, and thereby achieve the detection of both the position and orientation of the hand in real-time to millimeter-level accuracy with high precision at low cost. The tracking system requires reduced complexity hardware on the hand segments. As opposed to requiring significant infrastructural investment associated with the imaging cameras as the traditional approach to solving this problem, the hand-tracking methodology of this work uses light sensors only.

The paper is organized as follows: A brief description of Rambus's LSS technology that is given in Sections 2 and 3 describes the developed hand tracking and gesture recognition methodology in a systematic way. Both hand tracking and gesture recognition methods are validated in Section 4 using various experiments conducted with precise measurements. Finally, the work is concluded in Section 5.

2. Rambus Lensless Smart Sensor

Lensless Smart Sensors (LSS) stemmed from a fundamental question of what could be the smallest optical element that can be used to sense. The traditional optical lens is usually the limiting factor, in terms of defining how small the imaging device can be. Rambus pioneered a near field diffraction grating (patents issued and pending) that creates predictable light patterns on the sensor. Thus, the lens used in a conventional vision sensor is replaced by a spiral-shaped diffraction grating. The 1-mm thick LSS ensures a very small form factor for a miniaturized, wearable tracking system. The unique diffraction pattern associated with the LSS enables more precise position tracking than possible with a lens by capturing more information about the scene. However, instead of getting the image as such, the sensors will only provide unclear blob-like structures [40] that are rich in information. A proper reconstruction algorithm is developed using the blob data and the knowledge of grating to reconstruct the image without losing accuracy, which is well explained in previous works [41,42]. The sensing technology, as well as the optical, mathematical, and computational foundations of the LSS, are discussed in detail in the works of Stroke [43] and Gill [44].

3. Methodology

In order to make the system usable in a wide range of environmental conditions, infrared LEDs have been used in the tracking system, as they are impervious to white light sources in standard daylight conditions, when using the LSS in combination with an infrared filter. The method to range and track a single light source is well described in previous work [41,42]. The work described in this paper is focused on developing a method for tracking multiple light points simultaneously, and thereby estimating hand location and orientation; thus, enabling smart tracking of a hand.

3.1. Physical Setup

The stereoscopic field of vision (FoV) is defined on the basis of the following parameters (Figure 1):

(i) the longitudinal distance along the Z-axis between the LED and the sensor plane goes from 40 cm to 100 cm;

(ii) the distance between the right and left sensors (*Sen_R* and *Sen_L* respectively, as shown in Figure 1) measured on the central points of the sensors, called baseline (b) is 30 cm;

(iii) the combined FoV of 80°.

The work envelope is defined in this way since the typical range of motion for hand tracking and gesture recognition applications is confined more or less within these constraints [2].

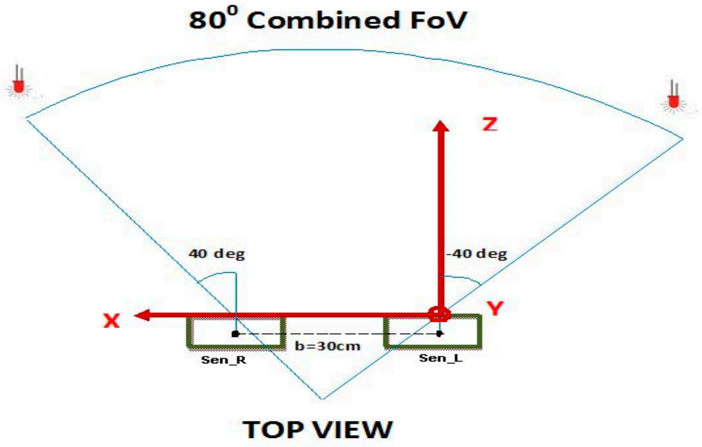


Figure 1. System overview.

3.2. Constraints for Multiple Points Tracking

The first stage of the experimental setup was an initial validation phase; a feasibility study was carried out by tracking two points simultaneously to understand the possible issues that can occur when dealing with tracking multiple points of light. Based on this initial analysis, two constraints are observed:

(i) Discrimination—When the distance between two light sources along the X- (or Y-) axis is less than 2 cm, the two sources will merge to a single point in the image frame, which makes them undistinguishable.

In the graph below (Figure 2), the two LEDs are progressively moved closer to each other, from 2 cm to 1 cm along the X-axis, at a distance of Z = 40 cm and Y = 0 cm. Both LEDs are in a central position between the two LSSs, and it is seen that the two light sources merge into a single point that can no longer be distinguished.

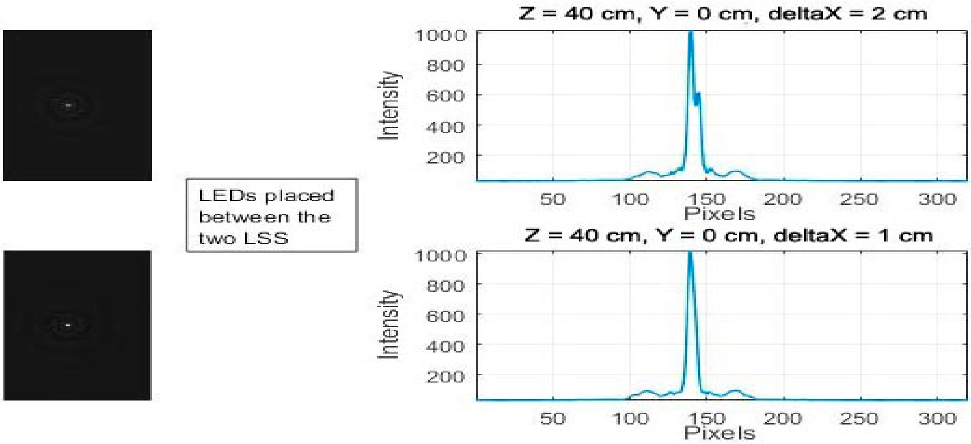


Figure 2. Light-emitting diodes (LEDs) placed at Z = 40 cm moved closer from 2 cm and 1 cm.

(ii) Occlusion—Occlusions occur when the light source is moved away from the sensor focal point along the *X*-axis. In Figure 3, at extreme lateral positions when the FoV ≥ 40°, even when the LEDs are 3 cm away, one LED is occluded by the other.

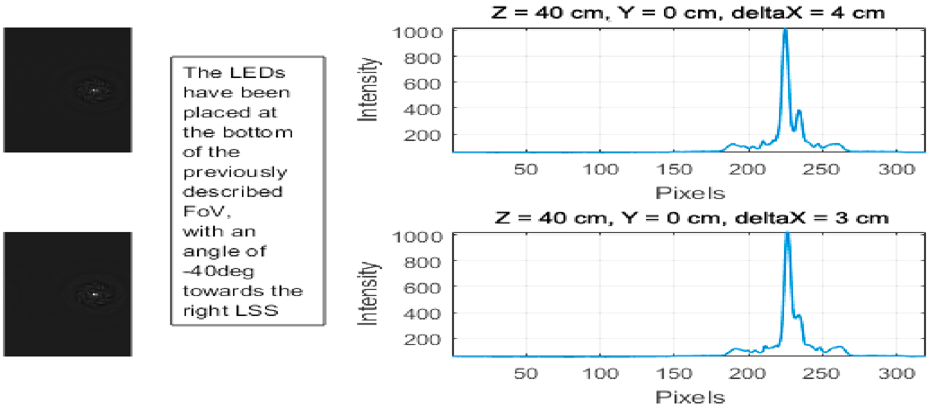
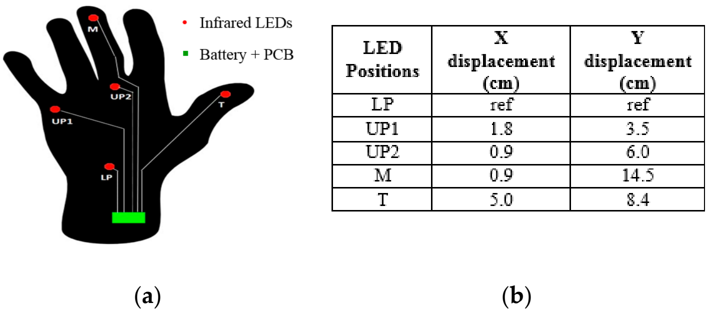


Figure 3. LEDs placed at FoV = 40° moved closer from 4 cm to 3 cm.

3.3. Placement of Light Points

The LEDs are placed on the hand in such a way as to minimize both the discrimination/occlusions problems, and the number of LEDs adopted, yet still allowing the recognition of the hand. Thus, three LEDs are used on the palm with the maximal possible distance between them (LP—Lower Palm, UP1—Upper Palm 1, UP2—Upper Palm 2). In this scenario, the palm is considered a flat surface to make the model simple for the first basic version of tracking using the LSS. The main objective was to make a workable system using light sensors alone, taking account of more constraints in the next versions by combing the light sensors with other possible sensors. It was important, because at least three points are required to estimate the orientation of a plane, which is how the hand orientation is determined. In addition, one LED is used in the middle finger (M) just above the upper palm LED to track the movement that is assigned to all of the other fingers, except for the thumb, to prove the concept of finger/multiple segment tracking. An additional LED is placed on the thumb (T) to track its movement. Figure 4a shows the positional overview of LEDs on the hand. Taking *LP* LED as a reference, the *X* and *Y* displacements of all of the other LEDs are calculated (Figure 4b).



LED Positions	X displacement (cm)	Y displacement (cm)
LP	ref	ref
UP1	1.8	3.5
UP2	0.9	6.0
M	0.9	14.5
T	5.0	8.4

(**a**) (**b**)

Figure 4. (**a**) Placement of LEDs in hand; (**b**) *X* and *Y* displacement of LEDs.

3.4. Hardware Setup

The experimental setup fixture consists of a 3D printed holder that is designed to hold five LEDs in the prescribed locations on the hand. This is then secured to the user's hand through simple straps, as shown in Figure 5a. The LSSs are placed in 3D-printed casings with the infrared filters [42] attached, as seen in the final setup shown in Figure 5b. It allows for an embedded platform that can be easily mounted on top of any laptop display. Since the sensing is based on diffraction optics, it is necessary to ensure that the LSSs are both carefully aligned on a common plane with no possible undesired rotations or translations while fixing them inside the enclosures.

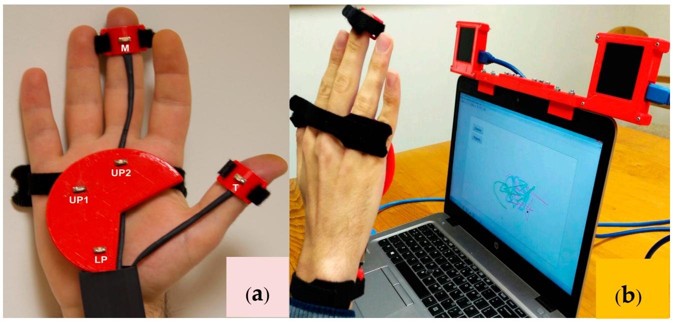


Figure 5. (**a**) Setup for hand; (**b**) final setup for hand tracking.

3.5. Multiple Points Tracking

The single point-tracking algorithm can be extended to a multiple points tracking scenario. As the sensors are not providing the hand image as such, and no IMUs are fitted on the hand, only the 3D LED locations are obtained. Therefore, the multiple points tracking methodology is formulated in a different way, utilizing the basic ideas of Extended Kalman filter tracking, and Random Sample Consensus (RANSAC) algorithms from the target-tracking works of [45–47]. The algorithm can be structured into two main phases, namely: calibration and tracking.

3.5.1. Calibration Phase

In this phase, the hand must be kept stationary in front of the sensors, with the palm and fingers open, and the middle finger pointing upwards. The calibration is performed only once to save the relative distances of LEDs, and use them as a reference in the next phase of tracking. The raw images are captured and reconstructed according to the method explained in [41]. In short, the spiral-shaped diffractive grating of the LSS creates equally spaced dual spirals for a single point source of the light, as shown in Figure 6a. This is taken as the reference point-spread function (PSF) for the LSS. The regularized inverse of the Fourier transformation of the PSF, which is also called the kernel, is then calculated. The inverse Fourier transform of the product between the kernel and the raw image obtained from the LSS, which is the dual spiral image itself for a point source of light, is the final reconstructed image. It is seen from Figure 6b that the point source of light is reconstructed with low noise from the dual spiral pattern provided by the LSS. The two spirals on the dual gratings are not identical to each other; indeed, one is rotated 22° with respect to the other so that one spiral could cover the potential deficiencies present in the other sensor, and vice versa. Therefore, the reconstruction result for each spiral is used separately and then added together. This also allows effectively doubling the amount of light captured, thus reducing noise effects.

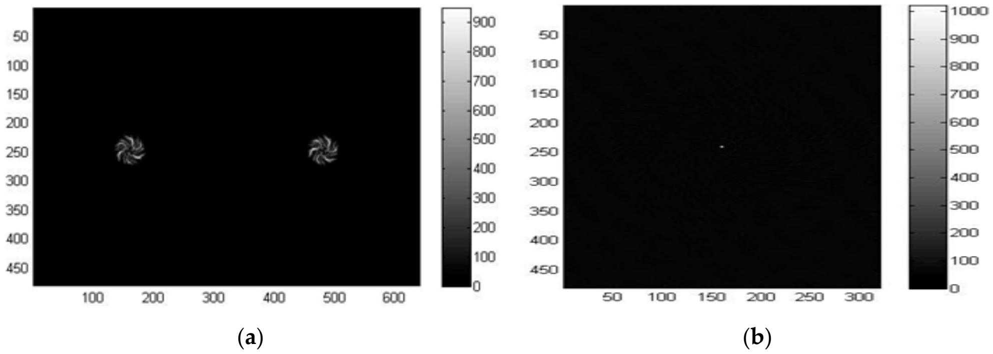


Figure 6. (**a**) Reference point-spread function (PSF); (**b**) reconstructed point.

The algorithm then looks for as many points as specified as the inputs (three for the palm, two for fingers) by searching the local maxima. The maxima detection algorithm starts with the highest threshold (*th* = 1) and iteratively decreases the threshold for each image frame until all five LEDs are detected correctly by both sensors. The lowest threshold is saved as the reference (th_{ref}) for the tracking phase. When the five detected points are correctly identified during the process, every point is labeled in order to assign each to the right part of the hand. The reconstructed frames have their origin in the top-left corner, with a resolution of 320 × 480, which is half the size along the *X*-direction and the full size along the *Y*-direction compared to the original image frames, as shown in Figure 6b.

Consequently, the following a priori information is considered for labeling based on the positioning of LEDs in the hand (Figure 5a):

a. The middle finger point (*M*) has the lowest row coordinate in both images.
b. The lower palm point (*LP*) has the highest row coordinate in both images.
c. The first upper palm point (*UP*1) has the lowest column coordinate in both images.
d. The second upper palm point (*UP*2) has the second column coordinate in both images.
e. The thumb point (*T*) has the highest column coordinate in both images.

The 2D coordinates in the image domain, relative to the two sensors together with the assigned labels, are processed according to the point tracking algorithm [41,42], using the reference that the PSF created. Thus, a matrix of relative distances of the palm coordinates is computed, as given in Table 1 using Equation (1). This unique measure together with the reference threshold will be used in the following phase to understand the configuration of the hand.

$$d_{p_i,p_j} = \sqrt{(x_{p_i} - x_{p_j})^2 + (y_{p_i} - y_{p_j})^2 + \left(z_{p_i} - z_{p_j}\right)^2}, \quad p_i, p_j = LP,\ UP1,\ UP2 \tag{1}$$

Table 1. Palm relative distances.

	LP	UP1	UP2
LP	$d_{LP,\,LP}$	$d_{LP,\,UP1}$	$d_{LP,\,UP2}$
UP1	$d_{UP1,\,LP}$	$d_{UP1,\,UP1}$	$d_{UP1,\,UP2}$
UP2	$d_{UP2,\,LP}$	$d_{UP2,\,UP1}$	$d_{UP2,\,UP2}$

3.5.2. Tracking Phase

During the tracking phase also, the raw images from the left and right sensors are captured and reconstructed according to the procedure explained in [39], which is briefly described in the calibration phase. The algorithm then looks for local maxima that lie above the reference threshold th_{ref}, which was saved during the calibration phase. The tracking phase is developed in such a way that it is impossible to start tracking if at least three LEDs are not visible for the first 10 frames, and the number of points in the left and right frames is not equal, in order to always maintain a consistent accuracy. As such, if mL and mR represent the detected maxima for *sen_L* and *sen_R* respectively, the following different decisions are taken according to the explored image:

a. $|mL| \neq |mR|$ and $|mL|, |mR| > 3$: a matrix of zeros is saved and a *flag* = 0 is set encoding "Unsuccessful Detection".
b. $|mL|, |mR| < 3$: a matrix of zeros is saved and a *flag* = 0 is set encoding "Unsuccessful Detection".
c. $|mL| = |mR|$ and $|mL|, |mR| > 3$: the points are correctly identified, *flag* = 1 is set encoding "Successful Detection", and the 2D coordinates of LED positions are saved.

(a) Unsuccessful Detection (*Flag* = 0)

In this case, the useful information to provide an estimate of the current coordinates is determined using a polynomial fit function based on the 10 previous positions of each LED. The current coordinates to be estimated are defined as in Equation (2):

$$[x_{0,i}, y_{0,i}, z_{0,i}]\ i = M, LP, T, UP1, UP2 \tag{2}$$

At each iteration, the progressive time stamp of the 10 previous iterations (3) and tracked 3D coordinates of each axis relative to the 10 iterations (4) are saved.

$$[t_{-10} \ldots t_{-1}] \tag{3}$$

$$\{[x_{-10,i}, \ldots, x_{-1,i}], [y_{-10,i}, \ldots, y_{-1,i}], [z_{-10,i}, \ldots, z_{-1,i}]\} \tag{4}$$

The saved quantities are passed to a polynomial function, as shown in Equation (5):

$$f_{poly_{p,j}} = a_{p,i}\, t^2 + b_{p,i}\, t + c_{p,i},\ p = [x, y, z] \tag{5}$$

The polynomial function performs the fitting using the least squares method, as defined in Equation (6):

$$\left[\hat{a}_{p,i}, \hat{b}_{p,i}, \hat{c}_{p,i}\right] = \mathrm{argmin}_{a_{p,i}, b_{p,i}, c_{p,i}} \sum_{j=-10}^{-1} \left(p_{j,i} - a_{p,i}{t_j}^2 + b_{p,i}t_j + c_{p,i}\right)^2 \tag{6}$$

Thus, the function f_{poly} estimates the coordinates of all five LED positions related to the current iteration, based on the last 10 iterations, as shown in Equation (7):

$$p_{0,i} = \hat{a}_{p,i}{t_0}^2 + \hat{b}_{p,i}t_0 + \hat{c}_{p,i} \tag{7}$$

(b) Successful Detection (*Flag* = 1)

In this case, the coordinates are preprocessed by a sub-pixel level point detection algorithm [42,48], which enhances the accuracy of peak detection by overcoming the limitation of the pixel resolution of the imaging sensor. The method will avoid many worst case scenarios, such as the striking of the LEDs on the sensors being too close to one another and the column coordinates of different points perhaps sharing the same value, ending in a wrong 3D ranging. The matrix of coordinates can then be used to perform the labeling according to the proposed method, as described below:

(i) Labeling Palm Coordinates

Given m_i, where i = L, R, as the number of maxima detected by both sensors, the combinations without repetitions of three previously ranged points are computed. Indeed, if five points are detected, there will be 10 combinations: four combinations for four detected points, and one combination for three detected points. Thus, triple combinations of C points are generated. The matrix of relative distances for each candidate combination of points, as $k \in [1, \ldots, |C|]$, is calculated using Equation (1). The sum of squares of relative distances for each matrix is then determined by Equations (8) and (9) for both the calibration and tracking Phases, as $k \in [1, \ldots, |C|]$:

$$[S_{d_{p_i,p_j}}(k)] = \sum\nolimits_{p_i,p_j \in ck} d^2_{p_i,p_j} \tag{8}$$

$$[Sref] = \sum\nolimits_{p_i,p_j \in ref} d^2_{p_i,p_j} \tag{9}$$

The results are summed again to obtain a scalar, which is the measure of the distance between the points in the tracking phase and the calibration phase, as shown in Equations (10) and (11), respectively:

$$Sum_k = \sum [S_{dp_i,p_j}] \tag{10}$$

$$Sum_{ref} = \sum [S_{ref}] \tag{11}$$

where both $[S_{d_{p_i,p_j}}] \in \mathbb{R}^3$ and $[S_{ref}] \in \mathbb{R}^3$ are vectors.

The minimum difference between the quantities Sum_k and Sum_{ref} can be associated with the palm coordinates. To avoid inaccurate results caused by environmental noise and possible failures in the local maxima detection, this difference is compared to a threshold to make sure that the estimate is sufficiently accurate. Several trials and experiments were carried out to find a suitable threshold to the value $th_{distance}$ = 30 cm. The closest candidate $\hat{k}$ was then found using Equation (12):

$$\hat{k} = argmin_{k \in c_k} \left| Sum_k - Sum_{ref} \right| < th_{distance} \tag{12}$$

The final step was to compute all of the permutations of the matrix of relative distances $\left[S_{d_{p_i,p_j}}(\hat{k})\right]_{perm}$, and the corresponding residuals with respect to the reference matrix $\left[S_{ref}\right]$ (13):

$$Res_{perm} = \left[S_{d_{p_i,p_j}}(\hat{k})_{perm} - \left[S_{ref}\right]\right] \forall permutations \tag{13}$$

The permutation, which shows the minimum residual, corresponds to the labeled triple of the 3D palm coordinates. The appropriate labels are assigned to each of the points, as follows: X_{LP1}, X_{UP1}, X_{UP2}.

(ii) Labeling Finger Coordinates

After identifying the palm coordinates, the fingers that are not occluded are identified using their positions with respect to the palm plane by knowing their length and the position of the fingertips. The plane identified by the detected palm coordinates X_{plane} is the normal vector to the palm plane. It is calculated using the normalized cross-product of the 3D palm coordinates of the points using the right-hand rule, as in Equation (14):

$$X_{plane} = \frac{(X_{UP1} - X_{LP1}) \times (X_{UP2} - X_{LP1})}{|(X_{UP1} - X_{LP1}) \times (X_{UP2} - X_{LP1})|} \tag{14}$$

The projections of the unlabeled LEDs onto the palm plane (if there are any) are computed based on the two possible cases:

Case 1—*Two Unlabeled LEDs*:

The projected vectors for unlabeled LEDs UL_1 and UL_2 onto the palm plane are calculated by finding the inner product of X_{plane} and $(X_{UL_i} - X_{UP2})$. It is then multiplied by X_{plane} and subtracted from the unlabeled LED positions, as shown in Equation (15):

$$Proj_{UL_i} = X_{UL_i} - \left[X_{plane}\left(X_{UL_i} - X_{UP2}\right)\right] X_{plane},\ i = 1, 2 \tag{15}$$

Two candidate segments X_{UL_1} and X_{UL_2} are then calculated as shown in Equations (16) and (17) and compared to the reference palm segment defined by X_{ref} in Equation (18):

$$X_{UL_1} = Proj_{UL_1} - X_{UL_2} \tag{16}$$

$$X_{UL_2} = Proj_{UL_2} - X_{UP2} \tag{17}$$

$$X_{ref} = X_{UP2} - X_{LP} \tag{18}$$

The results are used for the calculation of angles between them, as shown in Equations (19) and (20):

$$\theta_{UL_1} = arccos\left(\frac{X_{UL_1} \cdot X_{ref}}{\left|X_{UL_1}\right|\left|X_{ref}\right|}\right) \tag{19}$$

$$\theta_{UL_2} = arccos\left(\frac{X_{UL_2} \cdot X_{ref}}{\left|X_{UL_2}\right|\left|X_{ref}\right|}\right) \tag{20}$$

The middle finger is selected as the unlabeled LED UL_i, which exposes the minimum angle, while the thumb is selected as the maximum one, as shown in Equations (21) and (22):

$$X_T = argmax_{UL_1, UL_2}\left[\theta_{UL_1}, \theta_{UL_2}\right] \tag{21}$$

$$X_M = argmin_{UL_1, UL_2}\left[\theta_{UL_1}, \theta_{UL_2}\right] \tag{22}$$

Case 2—*One Unlabeled LED*:

If only one of them is visible, the single projected vector is used to compute the angle between the segment X_{UL} and the segment X_{ref} in the same way, as shown in Equation (19). The decision is made according to an empirically designed threshold th_{angle} = 30°, as shown in Equation (23):

$$UL = \begin{cases} X_M,\ if\ \theta_{UL} \leq 30^\circ \\ X_T,\ if\ \theta_{UL} > 30^\circ \end{cases} \tag{23}$$

(c) Occlusion Analysis

When the flag is set to 1, but one of the LEDs is occluded, its position is predicted using the method given in Section 3.5.2(a). The possibility of two different cases are as follows:

Case 1—*Occluded Finger*:

In this case, if there is no occlusion in the previous frame, the procedure is same as that of Section 3.5.2(ii). If a previous frame has occlusion, the coordinate is chosen according to the coordinate X_{UP2}.

Case 2—Occluded Palm:

If none of the candidate combinations satisfy the $th_{distance}$ constraint given in Equation (12), proceed in the same way as that of Section 3.5.2(a). Thus, the proposed novel multiple points live tracking algorithm labels and tracks all of the LEDs placed on the hand.

3.5.3. Orientation Estimation

Although the proposed tracking technique has no other sensors fitted on the hand to provide the angle information, and the tracking based on light sensors is able to provide only the absolute positions, the hand and finger orientations can be estimated. The orientation of the hand is computed by finding the plane identified by the palm X_{plane} in Equation (14) and the direction given by the middle finger. If the palm orientation, the distance between initial and final segments (S_0 and S_3) of middle finger (d), and the segment lengths (S_{il}, i = 0,1,2,3) are assigned as known variables, the orientation of all of the segments can be estimated using a pentagon approximation model, as shown in Figure 7.

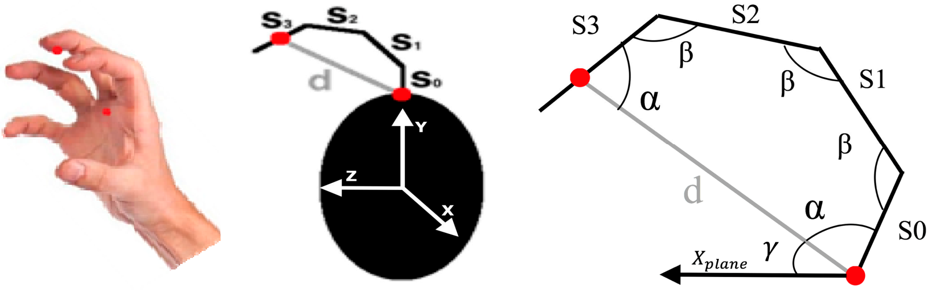


Figure 7. Modeling of palm to middle finger orientation.

For this purpose, the angle γ between the normal vector exiting the palm plane X_{plane}, as previously calculated in Equation (14), and the middle finger segment ($X_{UP2} - X_M$), is calculated as shown in Equation (24). The angle between S_0 and d (=α) is its complementary angle is shown in Equation (25).

$$\gamma = arcos\left(\frac{(X_M - X_{UP2}\ )\cdot X_{plane}}{|X_M - X_{UP2}|\left|X_{plane}\right|}\right) \tag{24}$$

$$\alpha = 90^\circ - \gamma \tag{25}$$

For the approximation model, it is assumed that the angle between S_0 and d is equal to the angle between S_3 and d. Thus, knowing that the sum of the internal angles of a pentagon is 540°, the value of the other angles (=β) is also computed assuming that they are equal to each other, as shown in Equation (26).

$$\beta = (540 - 2 \times \alpha)/3 \tag{26}$$

The vector perpendicular to the middle finger segment and X_{plane} is then computed as shown in Equation (27), and the rotation matrix R is built using angle α around the derived vector according to Euler's rotation theorem [49]. This rotation matrix is used to calculate the orientation of segment S_0 in Equation (28) and its corresponding 2D location in Equation (29).

$$X_{\perp(plane \leftrightarrow M)} = \frac{X_{plane} \times (X_M - X_{UP2})}{\left|X_{plane} \times (X_M - X_{UP2})\right|} \tag{27}$$

$$S_{0\nearrow} = R \times \left(\frac{(X_M - X_{UP2})}{|X_M - X_{UP2}|}\right) \tag{28}$$

$$X_{S_0} = X_{UP2} + S_{0\nearrow} \times S_{0l} \tag{29}$$

Thus, the orientation of all of the phalanges (S_1, S_2, and S_3), as well as their 2D locations, is calculated. The same method is applied for finding the orientation of all of the thumb segments using the segment lengths and 2D location of lower palm LED (X_{LP}), but with a trapezoid approximation for estimating the angles.

3.6. 3D Rendering

A simple cylindrical model is used to render the hand in 3D, with the knowledge of the positional information of five LEDs and the orientation of the palm plane and fingers (Figure 8). It is evident that at least three LEDs are required to reconstruct a plane; using the developed multiple points tracking algorithm, as seen from the figures, we are able to reconstruct the hand with the minimum number of LEDs. Figure 8a depicts the calculated 3D LED locations. From *LP*, *UP*1, and *UP*2 LED locations, the palm plane is reconstructed (Figure 8b). The middle finger location is used to reconstruct the middle finger and index, the ring finger, and the little fingers as well (Figure 8c,d). The thumb is reconstructed with the location information of the thumb LED (Figure 8e).

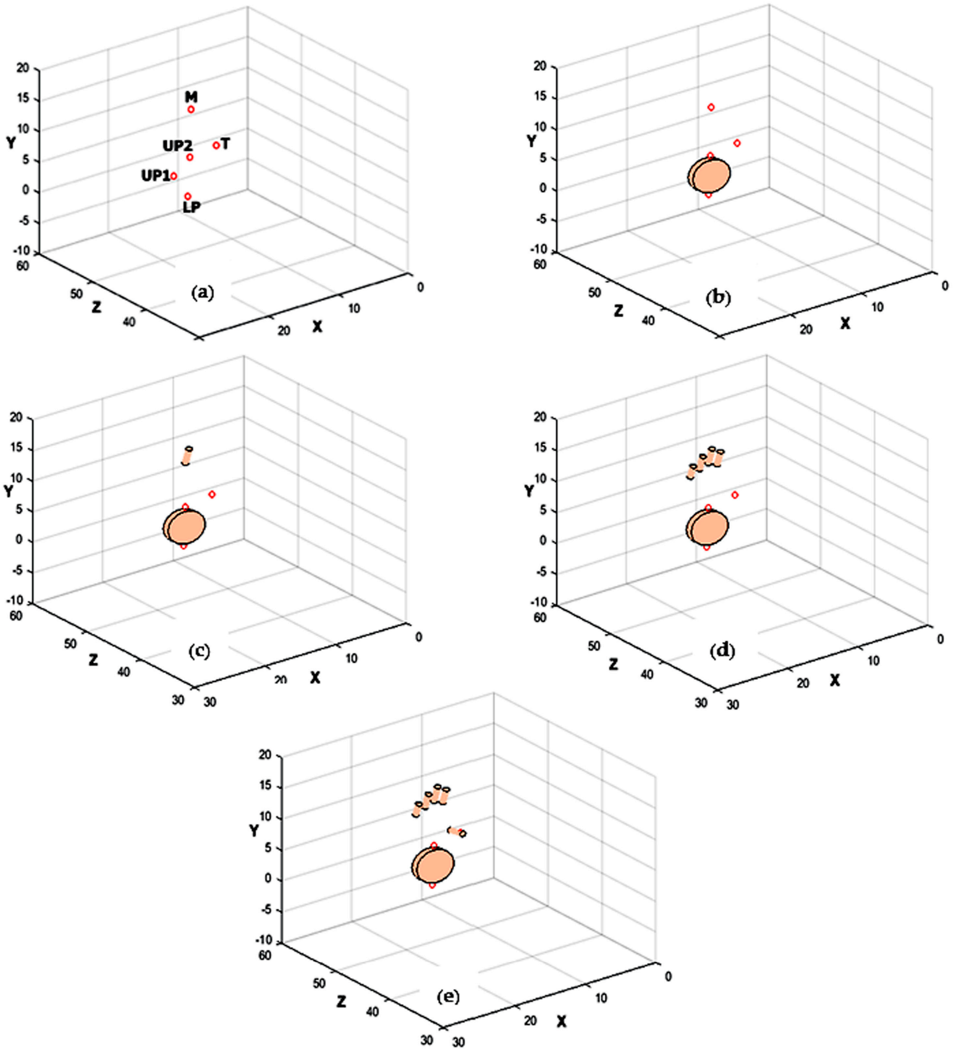


Figure 8. 3D Rendering of hand (**a**); calculated LED positions (**b**); palm plane reconstructed using Three LEDs on Palm (*LP*, *UP*1, and *UP*2) (**c**); middle finger reconstructed with *M* LED (**d**); index, ring, and little fingers reconstructed using the properties of *M* LED (**e**); thumb reconstructed with *T* LED.

3.7. Gesture Recognition

As an application of the novel multiple points tracking algorithm with merely five LEDs placed on the key points of the hand, a machine learning-based gesture recognition method is formulated. In principle, a simple gesture performed by the hand in front of the LSS sensors is expressed as a vector containing the sequence of LED positions indexed by time. Due to the new approach provided by the LSS and the novelty of the capturing itself, a simple dataset is collected to assess the performance in view of future analysis and improvements with more complex gestures. At present, the number of gestures–classes is fixed as $k = 6$. The vocabulary of gestures is given in Table 2.

Table 2. Vocabulary of gestures.

Gesture Label	Gesture Description
0 – Forward	Forward Movement along Z axis
1—Backward	Backward Movement along Z axis
2—Triangle	Triangle performed on X-Y plane: Basis parallel to the X axis
3—Circle	Circle performed on X-Y plane
4—Line Up → Down	Line Up/ Down direction on X-Y plane
5—Blank	None of the previous gestures

The gestures are performed at different distances from the sensors using the setup given in Figure 5a. The user is told to perform each of the combinations of gestures at least once during the data collection. The training dataset is then structured involving 10 people, with a total of 600 gestures taken. The feature extraction consists of a procedure to gather the principal characteristics of a gesture to represent it as a unique vector that is able to describe it. A frame-based descriptor approach is used for the same [50,51]. As illustrated in [50], the frame-based descriptor is well suited for extracting features from inertial sensors. However, in contrast to the descriptor created using inertial sensors, in this approach, it is applied to a dataset of different nature that consists of trajectories of the absolute positions indexed by time provided by the developed multiple points tracking algorithm.

Based on an experimental analysis, the number of captures to describe a gesture is set as 50 captures at 20 fps, where each capture refers to the 3D positions of five LEDs. This corresponds to a fixed length window of 2.5 s for each gesture. Even if the system is capable of a frame rate equal to 40 fps, it is sufficient to represent a gesture by down sampling to 20 fps. The mathematical modeling of a gesture descriptor is well described in the latest work of authors [51]. The dataset is then properly formulated and applied in the training and testing phases of a Random Forest (RF) model [52] to validate the performance of gesture recognition.

4. Results and Discussion

The effectiveness of the hand tracking algorithm is validated by following experiments: (1) Validation of Rambus LSS, (2) Validation of Multiple Points Tracking, (3) Validation of Latency, and (4) Validation of Gesture Recognition. The algorithms are tested using a computer with IntelCore i5 3.5 GHz CPU, 16 GB RAM.

4.1. Validation of Rambus LSS

The LSSs are fixed on a non-reflective black surface with a spacing of 30 cm (baseline b = 30 cm), and the light source is fixed at 50 cm away from the sensor plane (Z = 50 cm). The accuracy and precision measurements are done by fixing the light point at five different positions along the experimental plane: along the center of the baseline, on the external sides with a FoV equal to ±40°, and with a FoV equal to ±60°. The root mean square error (RMSE) is used as a measure of accuracy. For evaluating the precision of the sensors, Repeatability (30) and Temporal Noise (31) are used as the standard measures. Then, 1000 frames (N = 1000) are taken for each position for the estimation of RMSE and Repeatability.

The Temporal Noise is calculated by placing the LED at the same position for 10 min (600 s). In both cases, the Z readings (y_i) are plotted. The corresponding plots are shown in Figure 9a,b, respectively. The results are provided in Table 3.

$$Repeatability = \sqrt{\frac{\sum_{i=1}^{N}\left(\hat{y}_i - \sum_{i=1}^{N}\frac{\hat{y}_i}{N}\right)^2}{N}} \tag{30}$$

$$Temporal\ Noise = \sqrt{\frac{\sum_{i=1}^{N}[\hat{y}_i - (\hat{y}_i - 1)]^2}{N}} \tag{31}$$

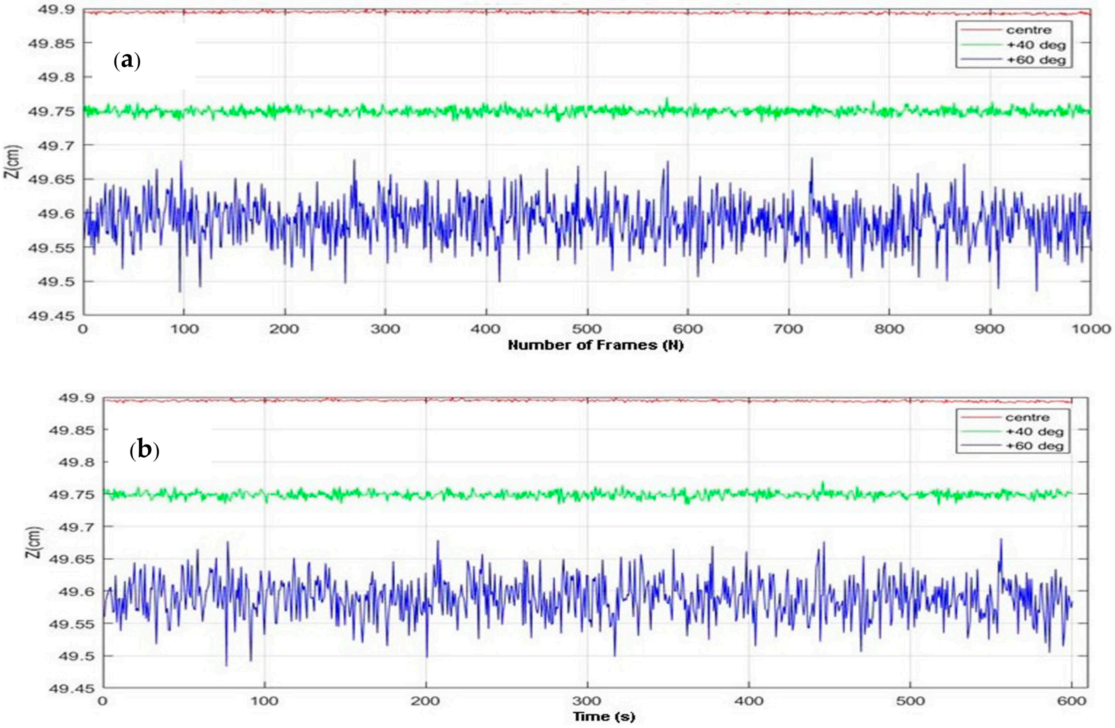


Figure 9. Measured Z at different fields of vision (FoVs) (**a**) for 1000 frames (**b**) for 10 min.

It is inferred from all of the plots that there is no significant variation in either accuracy or precision; the error is low (≈0). Precision in terms of Repeatability and Temporal Noise is high for the sensors, which implies consistency in the performance of the LSS technology. The results show that the LSS sensors are able to track the points down to millimeter-level accuracy, which underlines the quality of human hand tracking using these sensors.

Table 3. Accuracy and precision of the Rambus Lensless Smart Sensor (LSS).

Precision	Centre	+40 deg	−40 deg	+60 deg	−60 deg
RMSE (cm)	0.2059	0.2511	0.2740	0.3572	0.4128
Repeatability (cm)	0.0016	0.0058	0.0054	0.0210	0.0313
Temporal Noise (cm)	0.0027	0.0082	0.0078	0.0435	0.0108

4.2. Validation of Multiple Points Tracking

A wooden hand model with a length of 25 cm, measured from wrist to the tip of the middle finger and a width of 8.3 cm, is used to validate the multiple points tracking accuracy. Using a real hand for these measurements was impractical, because humans cannot maintain their hands in constant positions for the extended periods of time that was required in these experiments. The light points were placed in the exact same positions as described in Section 3.3 for the hand. The wooden hand was then moved onto a rectangular plane in front of the LSSs in the same way as that of single point tracking. The 3D positions of: *LP*, *UP*1, *UP*2, *M*, and *T* referred in Figure 4a were tracked using the developed multiple points tracking algorithm. The extremities of the plane were fixed at −25.2 cm to 25.2 cm laterally along the *X*-axis (which is located at the edges of the 40-deg FoV) with respect to the center axis, and 40 cm to 80 cm longitudinally along the *Z*-axis from the sensor plane. The plane is fixed in such a way as most of the hand movements are in this vicinity. On the horizontal sides, data points are collected for a total of 51 positions, and for the vertical sides, data are collected for a total of 41 positions spaced at 1 cm in order to cover the entire plane. For each point, five raw image frames were acquired by both sensors, and averaged to reduce shot noise. Figure 10 shows the *X*, *Y*, and *Z* measurements of the tracked plane with respect to the reference rectangular plane for all of the five LEDs; namely the *LP*, *UP*1, *UP*2, *M*, and *T* (placed at different locations on the hand). It is remarkable that, for all five positions, the tracked plane is nearly aligning with the reference plane. The corresponding RMSE along each axis and the three-dimensional error using Equation (32) are given in Table 4.

$$Total_{RMSE} = \sqrt{(X_{RMSE})^2 + (Y_{RMSE})^2 + (Z_{RMSE})^2} \tag{32}$$

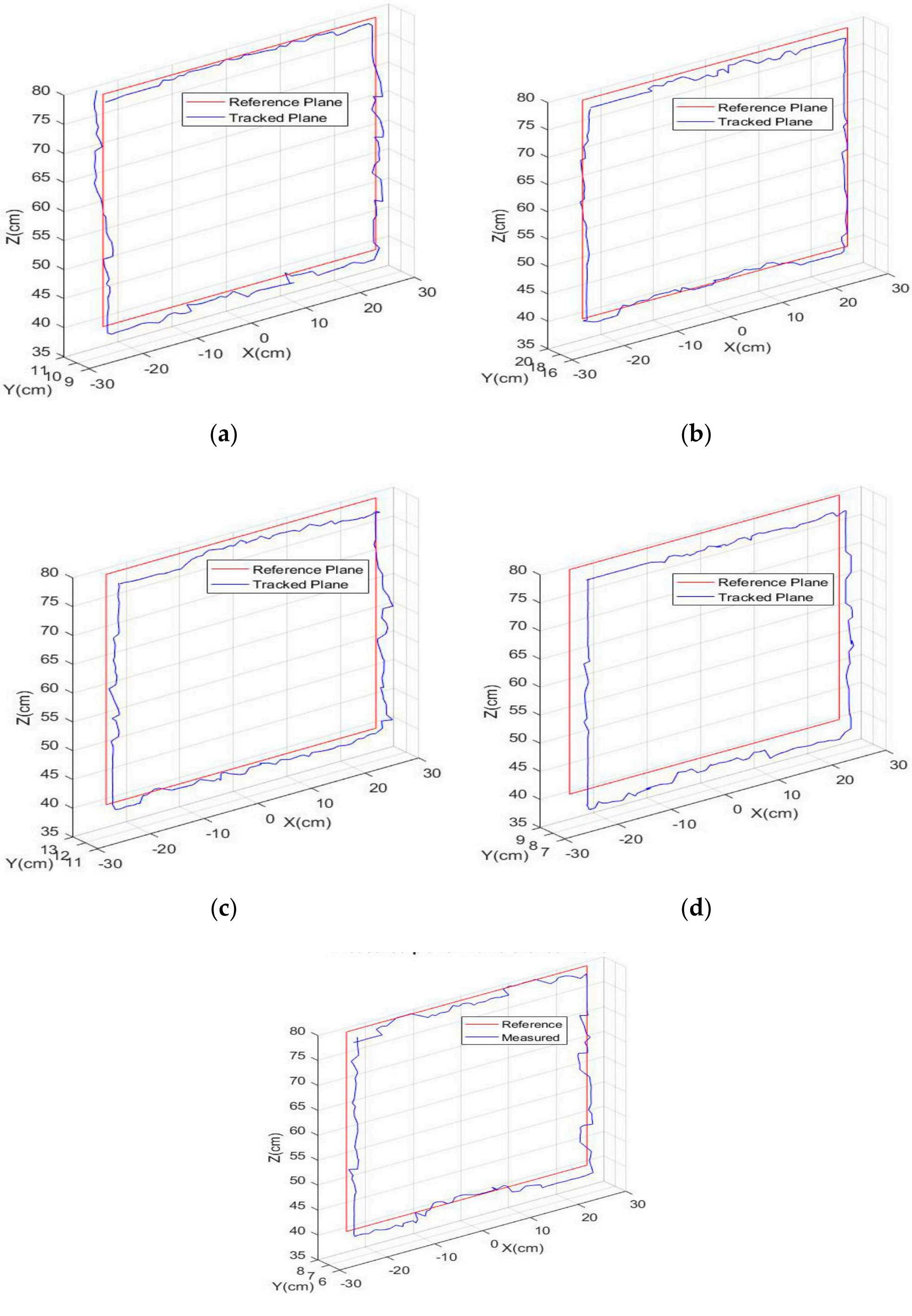
Table 4. Comparison of accuracies for each LED position.

RMSE (cm)	LP	UP1	UP2	M	T
X	0.5054	0.6344	0.5325	0.7556	0.7450
Y	0.3622	0.3467	0.5541	0.9934	0.5222
Z	0.8510	1.0789	0.9498	1.2081	0.7903
Total	1.0540	1.2987	1.1457	1.7370	1.2051

For the second experiment, the hand model is moved along the same rectangular plane, using the *LP* LED as the reference, and aligning it with each position. The position of all of the other LEDs was tracked at each point. In each case, the *X* and *Y* displacements of the LEDs with respect to *LP* is found, as shown in Figure 11. It is evident from the plots that the displacement of all of the LED positions along the entire reference plane, along both *X* and *Y* axes, are nearly constant. The deviation for the five positions, with respect to the mean, is computed to validate the results, as shown in Table 5. It can be noted that the mean deviation error is negligible. Using the actual *X* and *Y* displacements from Figure 4b, the displacement error, with respect to *LP* LED positions, is also calculated. The maximum error was recorded for the middle finger LED (*M*), which is maximally displaced with respect to *LP*. The lowest error was recorded for *UP*1, which was the least displaced from *LP*. *UP*2 had a comparable error to that of *UP*1, which is located close to the *UP*1. In all of the cases, the displacement error was less than 1 cm. From the inferences, it can be validated that the relative distance between the LED positions is constant with good accuracy, which implies the quality of the developed hand tracking methodology using the Lensless Smart Sensors.

Table 5. Displacement error for each LED position with respect to LP LED and mean.

RMSE (cm)	*X*-Axis		*Y*-Axis	
	With Respect to LP	With Respect to Mean	With Respect to LP	With Respect to Mean
UP1	0.5054	0.6344	0.5325	0.7556
UP2	0.3622	0.3467	0.5541	0.9934
M	0.8510	1.0789	0.9498	1.2081
T	1.0540	1.2987	1.1457	1.7370



(**a**)

(**b**)

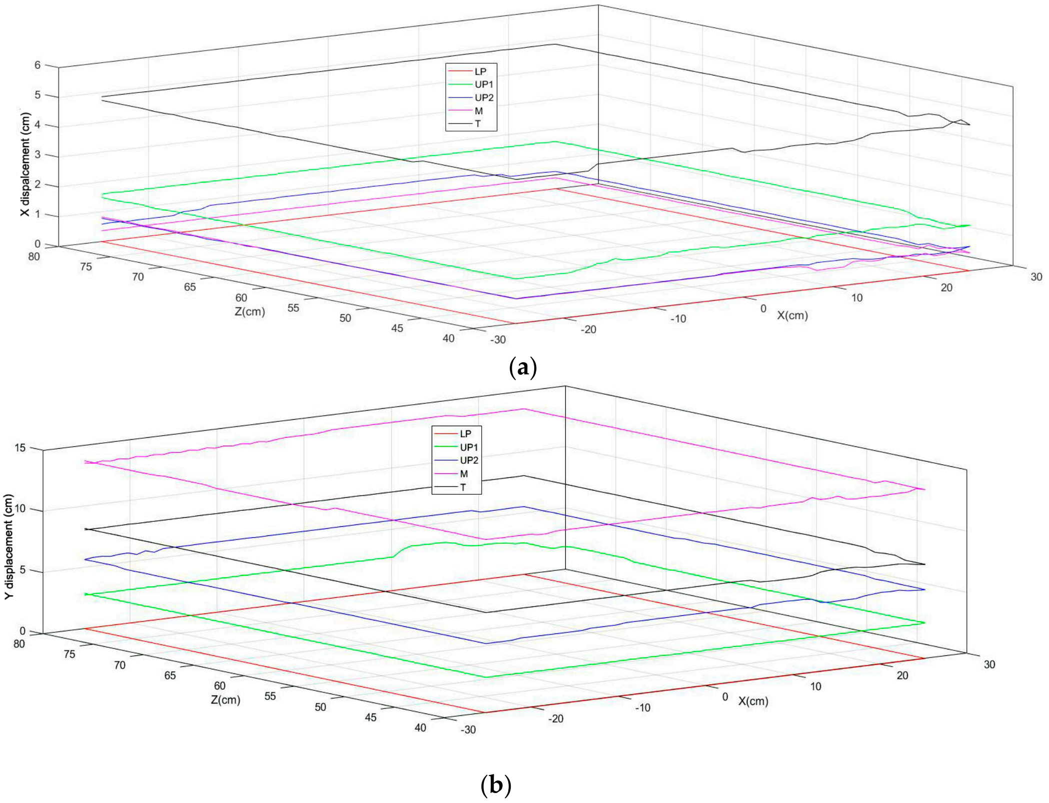
(**c**)

(**d**)

(**e**)

Figure 10. (**a**) *LP*; (**b**) *UP*1; (**c**) *UP*2; (**d**) *M*; (**e**) *T*. Tracked plane with respect to the reference plane for all LED positions.

The approach developed is robust and able to overcome the difficulty in detecting and distinguishing between the LEDs, i.e., the hand segments, using the image features. On the other hand, the method strongly relies on the accurate and precise ranging of LEDs achieved by the novel sensors. Even if the dynamics of the hand is higher than what the system can cope with—that is, the movement of the fingers is highly non-linear, and LED positions occasionally become close to each other—by modeling in this way, we are able to identify different configurations of the hand.



(**a**)

(**b**)

Figure 11. LED positions with respect to LP LED (**a**) *X*-Displacement (**b**) *Y*-Displacement.

The orientation estimation is formulated to compute the position and orientation of the middle finger. It is used to approximate the positions and orientations of the remaining fingers, as the simplified model at present has no LEDs fitted on these fingers. The properties obtained for the middle finger are assigned for the index, ring, and little fingers. Some of the other example hand orientations are also shown in Figure 12, which proves that the algorithm is able to recognize different configurations of the hand. The calculated hand orientation angles are shown as a function of the actual hand angles in Figure 13a-c. The axes of rotation are as given in Figure 7. Along the *Z*-axis, hand orientations are computed from −50° to +50°, whereas along the *X* and *Y* axes, they are calculated only from −30° to +30°, as after this point, no more LEDs are visible in either direction. The finger joint angle between S_3 and d (=α) is estimated and plotted with respect to the actual angles in Figure 13d. Here also, the possible three readings are plotted, based on the visibility of the LED fitted on the middle finger. The other joint angles are derived from this angle itself, as explained in Section 3.5.3.

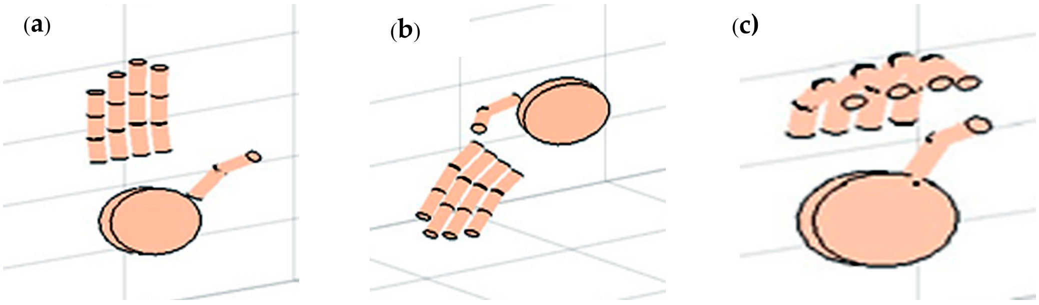


Figure 12. Different orientations of hand in front of LSSs: (**a**) hand held straight; (**b**) hand held upside down at an inclination; (**c**) fingers bend.

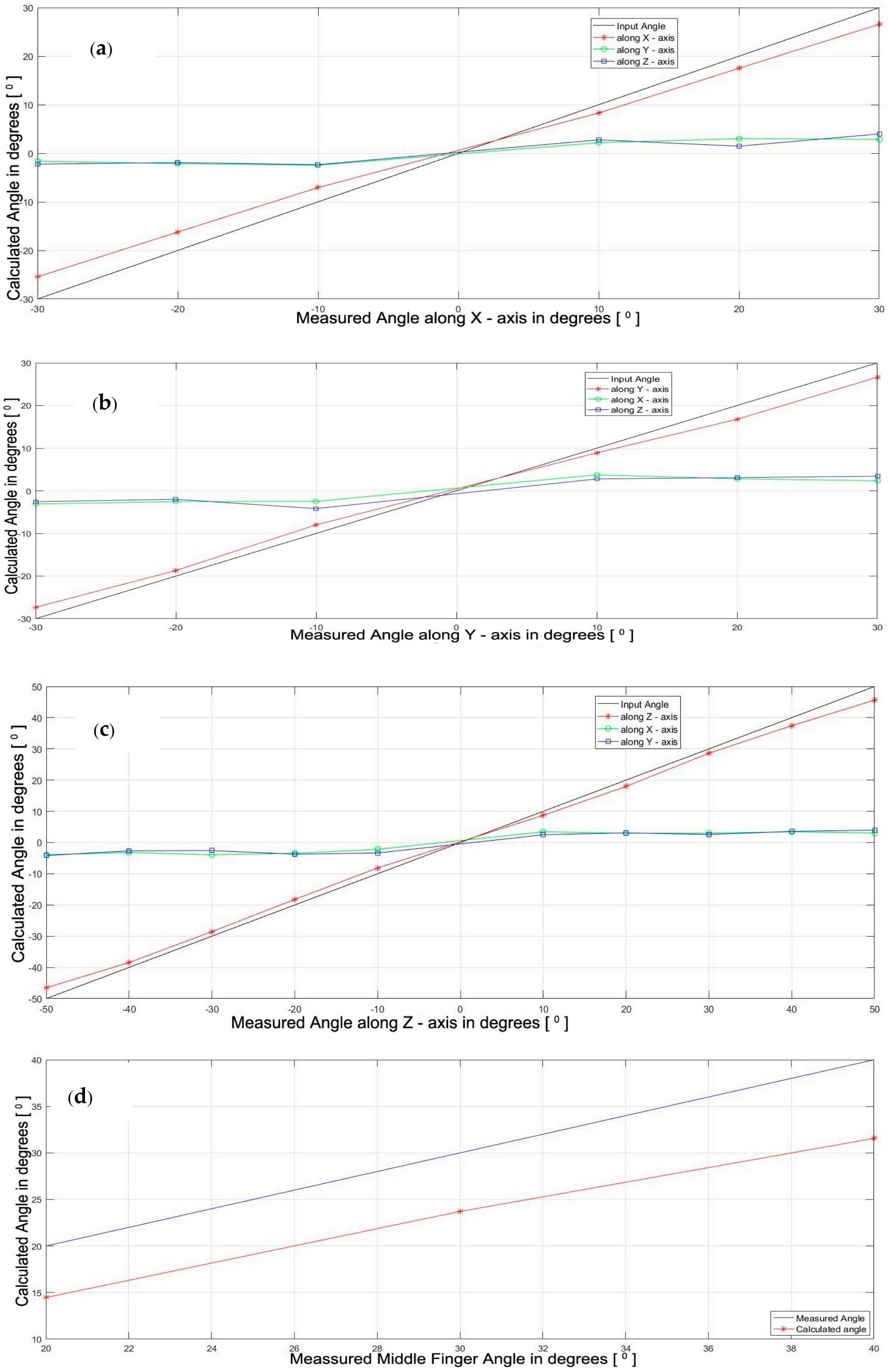


Figure 13. Calculated vs. Actual Orientation: (**a**) Along *X*-Axis; (**b**) Along *Y*-Axis; (**c**) Along *Z*-Axis; (**d**) Along Middle Finger between *S*3 and *d* (α).

Since the light sensors provide only the location information, and there are no other sensors fitted on the hand to measure the exact orientation, it is not expected to be highly accurate, as inferred from the graphs. Since it is difficult to place LEDs on all of the fingers and on all of the phalanges of the fingers because of the occlusions and discriminations investigated in Section 3.2, the algorithm is based on an approximation model, which has its own drawbacks. However, it should be noted that the main objective of this work is to develop a workable system giving emphasis to tracking accuracy rather than orientation. The focus on orientation estimation is to make it simple, yet workable. At this stage, it is impractical to make it much more accurate without using additional sensors that are able to provide angular information. A working demonstrator system can be seen in [53]. In future work, a minimum number of inertial sensors can be considered for integration with the LEDs to estimate all of the orientations of all of the fingers and the related phalanges. Further investigation should be carried out to find out an optimal balance between the number of LEDs and the inertial sensors to achieve accurate tracking, in terms of both the position and orientation, while still considering cost, power, and system complexity.

4.3. Validation of Latency Improvements

In order to reduce latency and facilitate fast hand movement tracking, we have been looking to maximize the frame rate of the data capture, whilst at the same time maintaining a good accuracy and precision in the measurements using the LSS. As discussed in the previous section, frames were averaged during the acquisition stage itself, and consecutively, results averaging two and one frames are compared in terms of the computational time and precision of measurements. The same computer with MATLAB was used for the tests. As a second step, instead of processing the entire image frame, a region of interest (ROI) of reduced size was chosen, and the light points were reconstructed with a PSF of the same size. Specifically, 140 black pixels from the bottom and the top of the image frames, along the *Y*-direction, were removed. It did not negatively affect the diffraction pattern in the acquired frames and PSF, but it did have an impact on the reconstructed frames (Figure 14).

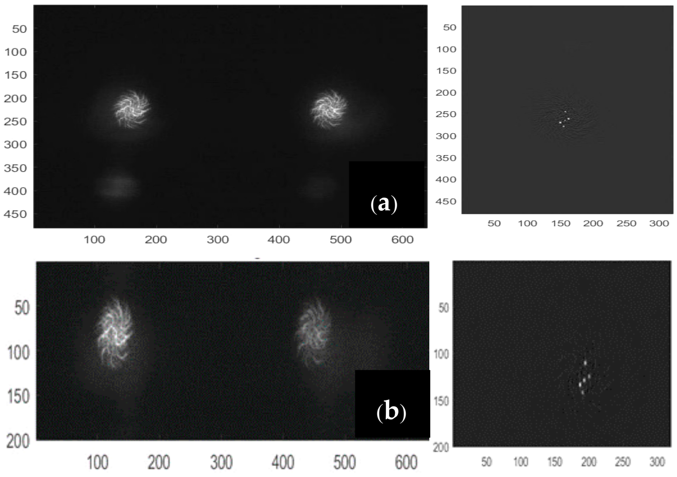


Figure 14. Image Frames and Reconstructed Frames (**a**) before Reducing the Region of Interest (ROI); (**b**) after Reducing the ROI.

In particular, the time difference between consecutive frames (dt) and estimated frames per second (EFps) are computed for multiple points tracking, as shown in Table 6. The accuracy and precision for the *LP* LED positioned at the center axis of the plane was calculated for the frames of both the full-size and the reduced in size ROI, in the same way as in Section 4.1, by reducing the number of frames to be averaged from five to one using the wooden hand. The results are given in Table 7. As the shot noise is higher, when the number of frames to be averaged is decreased, the accuracy and precision are negatively affected to an extent, but the latency is improved significantly. Although the image quality is degraded, the accuracy is not effected to the same extent because of the implementation of sub-pixel level point detection, which the results confirm. As seen from Table 7, the error keeps on increasing when the FoV is increased, and is varied more at extreme lateral zones. Since the hand tracking and gesture recognition is mostly confined within the region between the two sensors of a baseline b = 30 cm, the reduction does not have a significant impact on the application. Therefore, for the final setup, the image frames of the reduced ROI are used without averaging, i.e., one frame/acquisition. This solution led to the achievement of a frame rate of 40 *fps*, which is remarkable about the sensors. It also proves the effectiveness of the proposed tracking algorithm, which does not rely on a complex sensor configuration attached to the hand. The frame rate can be increased further by using high-speed embedded algorithms.

Table 6. Execution time for full and reduced ROI.

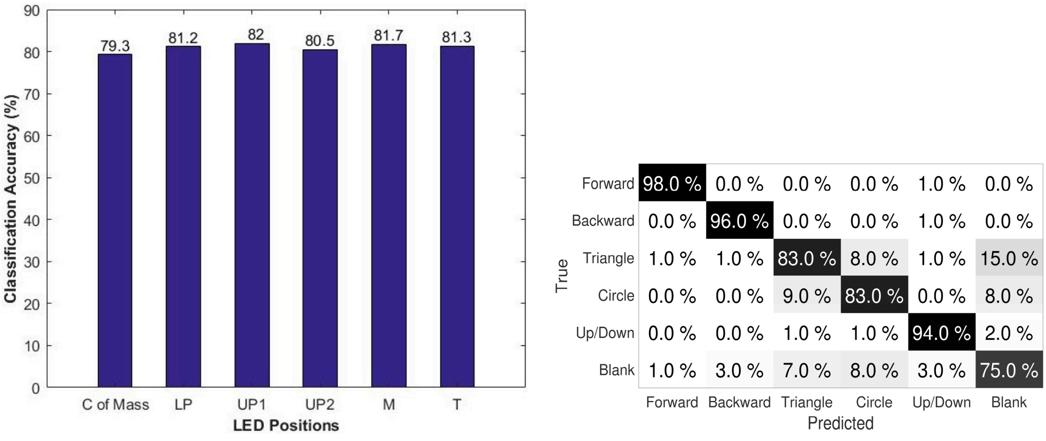
No. of Frames Averaged	Full Image Frames (480 × 320)		Image Frames with Reduced ROI (200 × 320)	
	Dt (s)	EFPs	Dt (s)	EFPs
5	0.0553 ± 0.0086	≈ 18	0.0482 ± 0.0134	≈ 21
2	0.0439 ± 0.0182	≈ 23	0.0296 ± 0.0085	≈ 34
1	0.0368 ± 0.0036	≈ 27	0.0248 ± 0.0047	≈ 40

Table 7. Accuracy and precision for LP LED at the center.

No. of Frames Averaged	Full Image Frames (480 × 320)			Image Frames with Reduced ROI (200 × 320)		
	RMSE (cm)	Repeatability (cm)	Temporal Noise (cm)	RMSE (cm)	Repeatability (cm)	Temporal Noise (cm)
5	0.4814	0.0025	0.0031	0.5071	0.0018	0.0036
2	0.5333	0.0047	0.0052	0.5524	0.0041	0.0049
1	0.5861	0.0086	0.0074	0.6143	0.0088	0.0091

4.4. Validation of Gesture Recognition

The gesture recognition methodology is employed in a simple painting application where each gesture is associated with a specific task, which can be viewed in [54]. It validated the performance of the Rambus LSS system in gesture recognition applications. A more detailed description of different machine learning approaches and an analysis of the results are provided in [51]. In this work, the classification accuracy as a function of LED position using Random Forest training [52] is evaluated to validate the performance. The graph in Figure 15a shows the classification accuracies of the LEDs at each position on the hand, along with the center of mass. It is noticeable that in all of the cases, the recognition accuracy is greater than 80%, which supports the results. The confusion matrix for the given dataset for the center of mass, considering the five LEDs, for the six gestures given in Table 2 is shown in Figure 15b. It is inferred that the matching accuracy for the predicted gestures with respect to true ones are more than 75% in all of the cases with the limited position and orientation information. This can be further improved by obtaining the information of all of the finger segments, and is able to add more gestures to the dataset.



	Forward	Backward	Triangle	Circle	Up/Down	Blank
Forward	98.0 %	0.0 %	0.0 %	0.0 %	1.0 %	0.0 %
Backward	0.0 %	96.0 %	0.0 %	0.0 %	1.0 %	0.0 %
Triangle	1.0 %	1.0 %	83.0 %	8.0 %	1.0 %	15.0 %
Circle	0.0 %	0.0 %	9.0 %	83.0 %	0.0 %	8.0 %
Up/Down	0.0 %	0.0 %	1.0 %	1.0 %	94.0 %	2.0 %
Blank	1.0 %	3.0 %	7.0 %	8.0 %	3.0 %	75.0 %

(**a**) (**b**)

Figure 15. (**a**) Classification accuracy as a function of LED positions; (**b**) confusion matrix for the dataset.

5. Conclusions

In this work, a novel method for positioning multiple 3D points, using the novel Lensless Smart Sensor (LSS), is proposed. It leverages the spatial resolution capabilities of the LSS, and allows for hand movement tracking utilizing vision sensors only. The research activities were focused on hand tracking. It was also extended to a wearable gesture recognition system with high classification accuracy. The location accuracy provided by the proposed hand-tracking algorithm is remarkable, given the type and specifications of the LSS sensors. In spite of having no lens to provide high quality images, it turned out to have advantages for wearable applications when compared to more conventional stereoscopic cameras or the depth sensors. Other techniques using IMUs tend to suffer from drifts, and therefore cannot be used to reliably determine the absolute locations in the world coordinate system. The LSS technology proves to be a promising alternative that is capable of recognizing rotations and translations around both palm and finger coordinates to an extent through the proposed approximation model and tracking algorithms. At present, the palm is considered as a flat plane, but in the future, the model can be calibrated by developing an error function, taking the irregularities of the palm surface into account. In future works, these imaging sensors can be integrated with the inertial sensors in a multimodal configuration, so that shortfalls of individual sensors can be complemented. A constrained biomechanical model of the hand is also considered as an extension of the work. At present, a workable system is developed that is considerably fast at tracking, but in the future, high-speed embedded algorithms will replace the current implementation, and further decrease the latency. Thus, an optimal solution can be achieved that can be suitable for a wearable platform in terms of speed, accuracy, size, and power consumption. This innovative technology can be integrated into a wearable full hand–finger tracking head-mounted system for nearly infrastructure-less virtual reality applications.

Author Contributions: L.A. contributed to the design and conception of study, data analysis, and interpretation; A.U. and N.N. performed the experiments and analyzed the data; M.P.W., M.W. and B.O. contributed to the draft of the manuscript; L.A. wrote the paper.

Funding: This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) and is co-funded under the European Regional Development Fund under Grant Number 13/RC/2077.

Acknowledgments: The authors thank Evan Erickson, Patrick Gill and Mark Kellam from Rambus, Inc., USA for the considerable technical assistance.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Taylor, C.L.; Schwarz, R.J. The Anatomy and Mechanics of the Human Hand. *Artif. Limbs* **1995**, *2*, 22–35.
2. Rautaray, S.S.; Agrawal, A. Vision Based Hand Gesture Recognition for Human Computer Interaction: A Survey. *Springer Trans. Artif. Intell. Rev.* **2012**, *43*, 1–54. [CrossRef]
3. Garg, P.; Aggarwal, N.; Sofat, S. Vision Based Hand Gesture Recognition. *World Acad. Sci. Eng. Technol.* **2009**, *3*, 972–977.
4. Yang, D.D.; Jin, L.W.; Yin, J.X. An effective robust fingertip detection method for finger writing character recognition system. In Proceedings of the 4th International Conference on Machine Learning and Cybernetics, Guangzhou, China, 18–21 August 2005; pp. 4191–4196.
5. Oka, K.; Sato, Y.; Koike, H. Real time Tracking of Multiple Fingertips and Gesture Recognition for Augmented Desk Interface Systems. In Proceedings of the 5th IEEE International Conference on Automatic Face and Gesture Recognition (FGR.02), Washington, DC, USA, 21 May 2002; pp. 411–416.
6. Quek, F.K.H. Finger Mouse: A Free hand Pointing Computer Interface. In Proceedings of the International Workshop on Automatic Face and Gesture Recognition, Zurich, Switzerland, 26–28 June 1995; pp. 372–377.
7. Crowley, J.; Berard, F.; Coutaz, J. Finger Tacking as an Input Device for Augmented Reality. In Proceedings of the International Workshop on Automatic Face and Gesture Recognition, Zurich, Switzerland, 26–28 June 1995; pp. 195–200.
8. Hung, Y.P.; Yang, Y.S.; Chen, Y.S.; Hsieh, B.; Fuh, C.S. Free-Hand Pointer by Use of an Active Stereo Vision System. In Proceedings of the 14th International Conference on Pattern Recognition (ICPR), Brisbane, Australia, 20 August 1998; pp. 1244–1246.
9. Stenger, B.; Thayananthan, A.; Torr, P.H.; Cipolla, R. Model-based hand tracking using a hierarchical bayesian filter. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 1372–1384. [CrossRef] [PubMed]
10. Elmezain, M.; Al-Hamadi, A.; Niese, R.; Michaelis, B. A robust method for hand tracking using mean-shift algorithm and kalman filter in stereo color image sequences. *Int. J. Inf. Technol.* **2010**, *6*, 24–28.
11. Erol, A.; Bebis, G.; Nicolescu, M.; Boyle, R.D.; Twombly, X. Vision based hand pose estimation: A review. *Comput. Vis. Image Underst.* **2007**, *108*, 52–73. [CrossRef]
12. Dipietro, L.; Sabatini, A.M.; Dario, P. A Survey of Glove-Based Systems and their Applications. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2008**, *38*, 461–482. [CrossRef]
13. Dorfmüller-Ulhaas, K.; Schmalstieg, D. *Finger Tracking for Interaction in Augmented Environments*; Technical Report TR186-2-01-03; Institute of Computer Graphics and Algorithms, Vienna University of Technology: Vienna, Austria, 2001.
14. Kim, J.H.; Thang, N.D.; Kim, T.S. 3-D hand Motion Tracking and Gesture Recognition Using a Data Glove. In Proceedings of the 2009 IEEE International Symposium on Industrial Electronics, Seoul, Korea, 5–8 July 2009; pp. 1013–1018.
15. Chen, Y.P.; Yang, J.Y.; Liou, S.N.; Lee, G.Y.; Wang, J.S. Online Classifier Construction Algorithm for Human Activity Detection using a Triaxial Accelerometer. *Appl. Math. Comput.* **2008**, *205*, 849–860.
16. O'Flynn, B.; Sachez, J.; Tedesco, S.; Downes, B.; Connolly, J.; Condell, J.; Curran, K. Novel Smart Glove Technology as a Biomechanical Monitoring Tool. *Sens. Transducers J.* **2015**, *193*, 23–32.
17. Hsiao, P.C.; Yang, S.Y.; Lin, B.S.; Lee, I.J.; Chou, W. Data glove embedded with 9-axis IMU and force sensing sensors for evaluation of hand function. In Proceedings of the 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Milan, Italy, 25–29 August 2015; pp. 4631–4634.
18. Gloveone-Neurodigital. Available online: https://www.neurodigital.es/gloveone/ (accessed on 6 November 2017).
19. Manus, V.R. The Pinnacle of Virtual Reality Controllers. Available online: https://manus-vr.com/ (accessed on 6 November 2017).
20. Hi5 VR Glove. Available online: https://hi5vrglove.com/ (accessed on 6 November 2017).
21. Tzemanaki, A.; Burton, T.M.; Gillatt, D.; Melhuish, C.; Persad, R.; Pipe, A.G.; Dogramadzi, S. μAngelo: A Novel Minimally Invasive Surgical System based on an Anthropomorphic Design. In Proceedings of the 5th IEEE RAS EMBS International Conference on Biomedical Robotics and Biomechatronics, Sao Paulo, Brazil, 12–15 August 2014; pp. 369–374.
22. Sturman, D.; Zeltzer, D. A survey of glove-based input. *IEEE Comput. Graph. Appl.* **1994**, *14*, 30–39. [CrossRef]

23. Wang, R.Y.; Popović, J. Real-time hand-tracking with a color glove. *ACM Trans. Graph.* **2009**, *28*, 63. [CrossRef]
24. Weichert, F.; Bachmann, D.; Rudak, B.; Fisseler, D. Analysis of the accuracy and robustness of the leap motion controller. *Sensors* **2013**, *13*, 6380–6393. [CrossRef] [PubMed]
25. Nair, R.; Ruhl, K.; Lenzen, F.; Meister, S.; Schäfer, H.; Garbe, C.S.; Eisemann, M.; Magnor, M.; Kondermann, D. A survey on time-of-flight stereo fusion. In *Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications, LNCS*; Springer: Berlin/Heidelberg, Germany, 2013; Volume 8200, pp. 105–127.
26. Zhang, Z. Microsoft Kinect sensor and its effect. *IEEE Multimed.* **2012**, *19*, 4–10. [CrossRef]
27. Suarez, J.; Murphy, R.R. Hand Gesture Recognition with Depth Images: A Review. In Proceedings of the IEEE RO-MAN, Paris, France, 9–13 September 2012; pp. 411–417.
28. Cheng, H.; Yang, L.; Liu, Z. Survey on 3D hand gesture recognition. *IEEE Trans. Circuits Syst. Video Technol.* **2016**, *26*, 1659–1673. [CrossRef]
29. Tzionas, D.; Srikantha, A.; Aponte, P.; Gall, J. Capturing hand motion with an RGB-D sensor, fusing a generative model with salient points. In *GCPR 2014 36th German Conference on Pattern Recognition*; Springer: Cham, Switzerland, 2014; pp. 1–13.
30. Li, Y. Hand gesture recognition using Kinect. In Proceedings of the 2012 IEEE International Conference on Computer Science and Automation Engineering, Beijing, China, 22–24 June 2012; pp. 196–199.
31. Wang, R.; Paris, S.; Popović, J. 6D hands: Markerless hand-tracking for computer aided design. In *24th Annual ACM Symposium on User Interface Software and Technology*; ACM: New York, NY, USA, 2011; pp. 549–558.
32. Sridhar, S.; Bailly, G.; Heydrich, E.; Oulasvirta, A.; Theobalt, C. Full Hand: Markerless Skeleton-based Tracking for Free-Hand Interaction. In *Full Hand: Markerless Skeleton-based Tracking for Free-Hand Interaction*; MPI-I-2016-4-002; Max Planck Institute for Informatics:: Saarbrücken, Germany, 2016; pp. 1–11.
33. Ballan, L.; Taneja, A.; Gall, J.; Van Gool, L.; Pollefeys, M. Motion capture of hands in action using discriminative salient points. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7577, pp. 640–653.
34. Du, G.; Zhang, P.; Mai, J.; Li, Z. Markerless Kinect-based hand tracking for robot teleoperation. *Int. J. Adv. Robot. Syst.* **2012**, *9*, 36. [CrossRef]
35. Sridhar, S.; Mueller, F.; Oulasvirta, A.; Theobalt, C. Fast and robust hand tracking using detection-guided optimization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3213–3221.
36. Tkach, A.; Pauly, M.; Tagliasacchi, A. Sphere-Meshes for Real-Time Hand Modeling and Tracking. *ACM Trans. Graph.* **2016**, *35*, 222. [CrossRef]
37. Chen, C.; Jafari, R.; Kehtarnavaz, N. A survey of depth and inertial sensor fusion for human action recognition. *Multimed. Tools Appl.* **2015**, *76*, 4405–4425. [CrossRef]
38. Liu, K.; Chen, C.; Jafari, R.; Kehtarnavaz, N. Fusion of inertial and depth sensor data for robust hand gesture recognition. *IEEE Sens. J.* **2014**, *14*, 1898–1903.
39. Lensless Smart Sensors-Rambus. Available online: https://www.rambus.com/emerging-solutions/lensless-smart-sensors/ (accessed on 6 November 2017).
40. Stork, D.; Gill, P. Lensless Ultra-Miniature CMOS Computational Imagers and Sensors. In Proceedings of the International Conference Sensor Technology, Wellington, New Zealand, 3–5 December 2013; pp. 186–190.
41. Abraham, L.; Urru, A.; Wilk, M.P.; Tedesco, S.; O'Flynn, B. 3D Ranging and Tracking Using Lensless Smart Sensors. Available online: https://www.researchgate.net/publication/321027909_Target_Tracking?showFulltext=1&linkId=5a097fa00f7e9b68229d000e (accessed on 13 November 2017).
42. Abraham, L.; Urru, A.; Wilk, M.P.; Tedesco, S.; Walsh, M.; O'Flynn, B. Point Tracking with lensless smart sensors. In Proceedings of the IEEE Sensors, Glasgow, UK, 29 October–1 November 2017; pp. 1–3.
43. Stork, D.G.; Gill, P.R. Optical, mathematical, and computational foundations of lensless ultra-miniature diffractive imagers and sensors. *Int. J. Adv. Syst. Meas.* **2014**, *7*, 201–208.
44. Gill, P.; Vogelsang, T. Lensless smart sensors: Optical and thermal sensing for the Internet of Things. In Proceedings of the 2016 IEEE Symposium on VLSI Circuits (VLSI-Circuits), Honolulu, HI, USA, 15–17 June 2016.
45. Zhao, Y.; Liebgott, H.; Cachard, C. Tracking micro tool in a dynamic 3D ultrasound situation using kalman filter and ransac algorithm. In Proceedings of the 9th IEEE International Symposium on Biomedical Imaging (ISBI), Barcelona, Spain, 2–5 May 2012; pp. 1076–1079.

46. Hoehmann, L.; Kummert, A. Laser range finder based obstacle tracking by means of a two-dimensional Kalman filter. In Proceedings of the 2007 International Workshop on Multidimensional (nD) Systems, Aveiro, Portugal, 27–29 June 2007; pp. 9–14.
47. Domuta, I.; Palade, T.P. Adaptive Kalman Filter for Target Tracking in the UWB Networks. In Proceedings of the 13th Workshop on Positioning, Navigation and Communications, Bremen, Germany, 19–20 October 2016; pp. 1–6.
48. Wilk, M.P.; Urru, A.; Tedesco, S.; O'Flynn, B. Sub-pixel point detection algorithm for point tracking with low-power wearable camera systems. In Proceedings of the Irish Signals and Systems Conference, ISSC, Killarney, Ireland, 20–21 June 2017; pp. 1–6.
49. Palais, B.; Palais, R. Euler's fixed point theorem: The axis of a rotation. *J. Fixed Point Theory Appl.* **2007**, *2*, 215–220. [CrossRef]
50. Belgioioso, G.; Cenedese, A.; Cirillo, G.I.; Fraccaroli, F.; Susto, G.A. A machine learning based approach for gesture recognition from inertial measurements. In Proceedings of the IEEE Conference on Decision Control, Los Angeles, CA, USA, 15–17 December 2014; pp. 4899–4904.
51. Normani, N.; Urru, A.; Abraham, L.; Walsh, M.; Tedesco, S.; Cenedese, A.; Susto, G.A.; O'Flynn, B. A Machine Learning Approach for Gesture Recognition with a Lensless Smart Sensor System. In Proceedings of the IEEE 15th International Conference on Wearable and Implantable Body Sensor Networks (BSN), Las Vegas, NV, USA, 4–7 March 2018; pp. 1–4.
52. Tian, S.; Zhang, X.; Tian, J.; Sun, Q. Random Forest Classification of Wetland Landcovers from Multi-Sensor Data in the Arid Region of Xinjiang, China. *Remote Sens.* **2016**, *8*, 954. [CrossRef]
53. Hand Tracking Using Rambus LSS. Available online: https://vimeo.com/240882492/ (accessed on 6 November 2017).
54. Gesture Recognition Using Rambus LSS. Available online: https://vimeo.com/240882649/ (accessed on 2 November 2017).