Detecting 3D Hand Pointing Direction from RGB-D Data in Wide-Ranging HRI Scenarios

Mirhan Ürkmez & H. Işıl Bozma
Intelligent Systems Laboratory, Electrical & Electronics Engineering
Boğaziçi University, Istanbul, Turkey
mirhan.urkmez@boun.edu.tr

2022 17th ACM/IEEE International Conference on Human-Robot Interaction (HRI) | 978-1-6654-0731-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/HRI53351.2022.9889385

***Abstract*—This paper addresses the detection of 3D hand pointing direction from RGB-D data by a mobile robot. Considering ubiquitous forms of pointing gestures, the 3D pointing direction is assumed to be inferable from hand data only. First, a novel sequential network-based learning model is developed for the simultaneous detection of hands and humans in RGB data. Differing from previous work, its performance is shown to be both accurate and fast. Following, a new geometric method for estimating the 3D pointing direction from depth data of the detected hand is presented along with a mathematical analysis of sensor noise sensitivity. Two new data sets for pointing gesture classification and continuous 3D pointing direction with varying proximity, arm pose and background are presented. As there are no such data sets to the best of our knowledge, both will be publicly available. Differing from previous work, the robot is able to estimate the 3D hand direction both accurately and fast regardless of hand proximity, background variability or the detectability of specific human parts - as demonstrated by end-to-end experimental results.**

***Index Terms*—Continuous pointing direction, deictic pointing, human robot communication, pointing data sets**

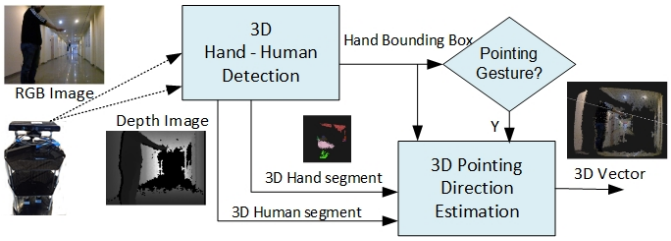
I. Introduction

The understanding of pointing is important for fostering communicative strategies between humans and robots and to increase their social acceptance. This is because humans use deictic pointing commonly among themselves for communication – as they complement or even substitute verbal descriptions [25]. A pointing gesture typically specifies a direction from a person - indicating a location, a thing or another person [41]. As such, it can be used to direct the robot's attention to a place or an object of interest in human robot interaction (HRI). As such, the ability to estimate pointing direction flexibly is critical for referential communication [40]. For example, it could be used in an interactive object modeling and labeling system for service robots [23] or robot's learning enhanced with social cues [39].

Interestingly, there is no common consensus in regards to how the pointing direction is estimated. It is known that people point as to align their eyes (gazes in fact), their index finger and the target from an egocentric perspective. However, the detection of such alignments may not be feasible in many HRI scenarios due to: i) Close-range interactions in which the robot sees only the body, but not the eyes; ii) Far-range interactions in which eye positions cannot be accurately detected; iii) Interactions in which the eyes are not either visible (i.e. the person is wearing glasses). Furthermore, the person performing the pointing gesture may not always look to the pointing target. Hence, in most cases, the typical recourse is to determine the 3D pointing direction from the hand alone [49].

While pointing gestures have protean nature, the most ubiquitous forms of pointing gestures consist of either i) extending the index finger with the remaining fingers flexed into the palm (one finger form) or ii) the thumb flexed into the palm with the remaining fingers extended (flat hand form) as shown in Fig. 1a [30], [9]. Even so, the reliable estimation of the 3D pointing direction has proven to be a complex task for a mobile robot due to two primary reasons: First, the robot views the hand from a single perspective. Second, HRI scenarios can vary with respect to setting conditions such as hand proximity, background variability and the visibility of certain entities such as human body parts or objects.

(a) Prototypical views of ubiquitous pointing gestures: One finger forms and flat hand forms.



(b) The robot first detects the 3D hand and human and then estimates the 3D pointing direction from the respective geometry.

Fig. 1: Pointing gestures and processing pipeline of 3D pointing direction estimation.

There has been extensive work within both robotics and machine vision communities. Most work explore only close-range interactions in which the hand is the dominant object in the scene [10]. In many HRI applications such as service robotics, this assumption does not hold and the pointing gesture can be distal. As this complicates the problem, the proposed methods consider assumptions such as having a fixed background [16], [36] or using background information [10]. Interestingly, neural networks have not been effective for this

problem due to the absence of a large 3D pointing direction data set. Recent works have addressed these challenges, however the proposed approaches have limitations as they find two-dimensional (2D) pointing directions [33] or output quantized 3D directions [5] or require specific body parts such as faces or elbows to to be detected [4], [13]. Thus, they are either not applicable in tasks in which the nature of human-robot interaction requires a fine-grained 3D direction estimate or in which the estimation must be possible without requiring the visibility of detectability of specific body parts.

This paper presents a novel approach for detecting the 3D pointing direction from a single perspective. This approach is applicable by any mobile robot that is endowed with a RGB-D camera. It is assumed that the pointing gestures are of one-finger or flat hand forms with pointing directed towards frontal hemisphere. The processing pipeline consists of two stages as shown in Fig. 1b. First, the robot detects the hand and human from the RGB data and if it finds the hand to be in a pointing gesture, it then obtains the 3D hand segment. Following, it infers the 3D pointing direction - considering the associated covariance matrix and the relative positioning of hand with respect to human. The contributions of the paper are as follows:

- A novel sequential learning model is developed for the simultaneous detection of hand and human in RGB data. Differeing from previous work, its performance is shown to be both accurate and fast.
- A new geometric method for estimating pointing direction from the hand depth data.
- An analysis of the sensitivity of the estimated direction to sensor noise is presented.
- Two new benchmark data sets for pointing gesture classification and continuous 3D pointing direction with varied hand proximity, arm pose and background. To the best of our knowledge there is no existing benchmark for the 3D pointing directions targeted in this work, so both will also be publicly available.

Differing from previous work, the robot can estimate the 3D hand direction both accurately and fast regardless of hand proximity, background variability or the detectability of specific human parts – as demonstrated experimentally. The code of our approach is available at: `github.com/islboun/3D_Pointing_Estimation`.

The outline of the paper is as follows: Related literature is reviewed in Section II. Section III. Experimental results on comparative hand detection performance as well as end-to-end 3D pointing direction detection are discussed in Section V. The paper concludes with a brief summary and future directions.

II. Related Literature

Most of the proposed approaches for detecting the 3D pointing direction take advantage of the extensive work done on hand detection. First, we present a brief review of this work. Next, we review work on the detection of 3D hand pointing direction. Finally, we briefly review the available data sets.

TABLE I: Related work: - indicates not required or available.

Method	Input	Proximity constraints?	Background variability?	Specific other body parts reqd?	Accuracy-Error
[35]	2 RGB	✓	✓	Head	25°
[18]	2 RGB	✗	✗	-	91%
[47]	RGB-D	✓	✓	Legs & face	-
[13]	ToF	✓	✓	Head & torso	2.79°
[1]	RGB-D	✓	✓	Skelatal	-
[16]	1-2 RGB	✓	✗	-	72.8%
[15]	RGB-D	✓	✓	Skelatal	76-100%
[43]	RGB-D	✗	✗	-	93.45% (10°)
[21]	RGB	✓	✗	-	-
[10]	D	✓	✗	-	76.8-92%
[37]	RGB-D	✗	✗	Skelatal	97-99.6%
[4]	RGB-D	✓	✓	Face	16.1-48.4 cm
Proposed	RGB-D	✓	✓	-	93.2% (10.69°)

A. Hand Detection

Most works on hand detection are based on RGB data. Early works have mostly used skin color [48], [11], [35], shape features via boosted classifiers [14], [26], context information from human body parts [22], [27], [8] or their various combinations [32]. However, their reliability tends to deteriorate if the robot is to operate in a wide-range of scenarios where hand appearances vary greatly with respect to color, shape and size. Recently, CNN-based detectors have been addressing these issues [38], [29], [12], [28] - as large data sets such as [31] have become publicly available. Mask-RCNN is used to reliably predict a bounding box segmentation of hands in cluttered environments [33], however it is slow for real-time applications. A modified MobileNet with SSD has been developed as a real-time hand detector [54]. A detector based on the faster R-CNN architectures detector has been presented in [52], however the required computational resources are very high for real-time applicability. There are also approaches that consider other modalities such as depth obtained from infrared sensing [42], [46]. However, these either assume the segmented body to be available or hand to be the closest object - as detection performance tends to decrease distance due to increased sensing noise. A combination of modalities such as RGB-D have also been used [53]; however the available training data sets are quite small for deep network based learning and hence most currently used hand detection methods continue to be RGB-data based.

B. 3D Hand Pointing Direction Detection

The estimation of 3D pointing direction follows hand detection. The applicability of these methods vary depending on the sensing, hand range, background variability, whether they require the detection of specific body parts and whether a continuous or quantized 3D direction estimate is output. A list of the proposed approaches along with how they fare with

respect to these issues is presented in Table I. Of course, a pointing gesture provides only a coarse spatial information of the targeted object. Nevertheless, it has been shown that potential pointing targets can be determined with an average angular error of less than 20° at a distance of about 2.5 meters [45].

The sensor type is important because the type of data (RGB image, depth, RGB-D) determines what can be computed from the data. For example, typically 2D fingertip positions can be found with a RGB camera [6], [19], [51], [20]. While some approaches attempt to infer 3D pointing direction from the 2D position data [21], their accuracy is limited as there is no distance information in 2-D data. One remedy is to use multiple RGB cameras with different locations [18]. However, the applicability of these methods tends to be limited in mobile robotics applications. Thus, the reliable estimation tends to be problematic with solely RGB data.

Hence, depth data as obtained from stereo cameras [35], depth cameras [10], time-of-flight cameras [13], or RGB-D sensors such as Kinect [47] have become integral for this task. There are also works that assume the availability of Kinect skeletal data [1], [15]; however these approaches are only applicable if there is a movement of the pointing hand. From the robotics perspective, it is important to detect hands in unconstrained conditions [50].

The remaining issues pertain to the applicability of the method in different HRI interaction scenarios. In some work, the hand is assumed to occupy a large part of the robot's sensory viewing volume and hence hand detection is rather easy [18], [51], [20]. These approaches are not applicable in many HRI scenarios since they require the hand to be the closest object to camera. Another issue is regarding the background variability. In some work, a fixed background is assumed [18], [43]. Another issue is that some approaches use the relative geometry between the hand and another body part to determine the 3D pointing direction. For example, the estimation is computed from the line from face to hand or elbow to hand [13], wrist to hand[47] or head to hand [4]. The former is motivated by the findings that suggest people are point as to align their eyes (gazes in fact), their index finger and the target from an egocentric perspective. In addition, some approaches require specific hand topologies - such as the back of the hand facing upwards [16] or person-specific fist models [10]. In all of these methods, the detection of these body parts needs to be both possible and reliable. As discussed previously, this is not always possible due to the nature of the interactions with respect to human's proximity and visibility. Furthermore, even if body parts are found, hand pointing direction may not be aligned with the line from elbow to hand or face to hand. In such cases, the only recourse is to determine the 3D pointing direction from the hand alone. Finally, in some methods, the 3D pointing direction is estimated in a quantized manner - such as the 26 direction quanta [5]. As such, the resulting estimates turn out be very coarse due to the quantization.

C. Hand Pointing Direction Data Sets

Most of the data sets on human pointing direction are based on close range and/or egocentric RGB images of the hand [19], [51]. Hence, the data set is not suitable for testing performance with varying hand distances. A RGB data set with 2D pointing directions is given in [5], however, it is not possible to compute estimation error as ground truth pointing directions are quantized into 8 levels. They also provide RGB-D frames of video recordings for quantized 3D direction classes. However, ground truth is not readily available. A data set of 180 pointing gestures with ground truth is given in [43]; however, the scenario is close-range and is limited to a fixed table setup.

III. Proposed Approach

The proposed approach consists of first i) determining the 3D hand segment followed by ii) the inference of the 3D pointing direction considering the associated covariance matrix and the relative positioning of human with respect to the hand.

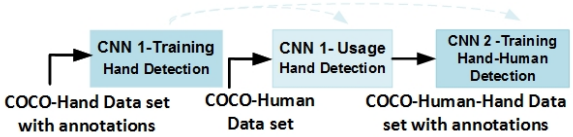


Fig. 2: Sequential learning model for hand and human detection. The first CNN is first trained to detect hands only and then used to detect hands on the COCO human data set. Finally, a second CNN is trained to detect both hands and humans.

A. 3D Hand and Human Detection

3D Hand detection is done in three stages: First, the robot detects the hand in the incoming RGB data. Simultaneously, it also detects the human. The latter is done since the human constitutes a reference for the positive pointing direction as we will explain in Section III-B. However, no specific human part is required to be seen by the RGB-D sensor - so this does not impose any restrictions on our target scenarios. Next, it checks if the hand is in a pointing gesture or not. Finally, if the hand is in a pointing gesture, the 3D hand segment is estimated from the depth data associated with the hand.

For hand and human detection, only RGB data is used - due to the availability of large publicly available RGB training data sets. Unfortunately there is no available network that can simultaneously detect hands and humans. Thus, a novel sequential learning model as shown in Fig. 2 has been developed for this purpose. This is because the training cannot be done simultaneously since there is no available data set with both hand and human annotations. Rather, the available data sets such as the COCO-hand data set [33] and COCO data set [31] provide each annotation separately. Both networks are selected to be convolutional neural networks (CNNs) - due to the reliability of CNN learning methods running on RGB images. The first network is trained to detect hands only using

the COCO-Hand data set. The resulting network is then used to detect hands on the COCO human data set in order to create an annotated data set with hand and human labels. Of course, there are bound to be false positives as we are using hand detector to create labels. However, as the performance of the hand detector is quite reliable, we expect a slight decrease in performance. Following, a second network is trained to detect both hands and humans using the annotated data from the second stage. Hence, the network learns to detect both hands and humans and outputs two corresponding bounding boxes with respective labels. An example of the resulting hand and human detection is given in Figure 5a.

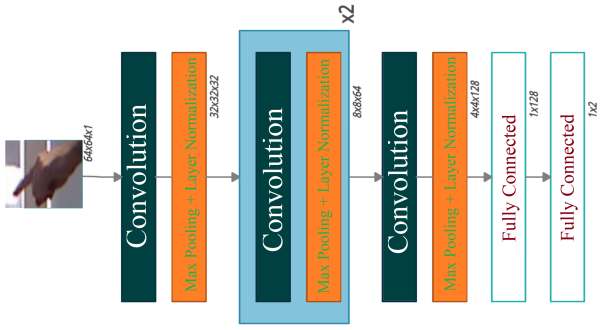


Fig. 3: Architecture of pointing gesture classifier

Once the robot detects a hand, it checks whether it is in a pointing gesture or not. For this, a two class CNN as shown in Fig. 3 is trained. This architecture is similar to that of [17] that is known to have reliable detection performance [2]. Rectified linear activation function (ReLU) is applied to the outputs of each convolution layer. Max pooling layers are used to reduce the computational complexity.

Finally, if the hand is in a pointing gesture, the 3D hand segment and human segment are found through lifting the corresponding 2D bounding boxes to 3D depth data and then applying a connected component labeling based 3D segmentation algorithm within the resulting volume. This requires the RGB data and depth data to be calibrated. The calibration is achieved using a method as prescribed in [3]. A sample case is a shown in Fig 5b. Both segments are found based on the observation that each occupies the biggest area and is also the closest object within this volume. Thus, each segment can be associated with a score that measures how large it is and how close it is. The segment with the highest score is assumed to be the hand or human segment respectively. Typically, we expect this assumption to hold. For example, the reliability of 3D hand segments are supported by high accuracy performance as given in Table IIIa.

B. Pointing Direction Estimation

In the second stage, the geometry of the 3D hand segment is exploited to estimate 3D pointing direction. In particular, in most common forms, a big part of the hand is physically gathered around a direction. This implies that the pointing direction corresponds to the direction with the largest extent of point cloud data. Note that this holds even when non-index finger such as the thumb is visible - unless that particular finger is longer than the index finger. This observation yields a simple mathematical method that can be easily applied once the 3D hand segment is found.

(a) The set $\mathcal{H}_2$ is derived from the plane $\perp$ to v_2' with maximum neighboring point cloud points.

(b) The estimate is based on the set $\mathcal{H}_2 \cap \mathcal{H}_3$

Fig. 4: The vector v_1' is an initial estimation of the 3D pointing direction. The actual direction is derived through finding the pointing part of the hand.

The method starts by forming an initial coarse estimate of the pointing direction from the covariance matrix Σ' of the 3D hand segment. This is a positive semi-definite matrix with eigenvalues $\lambda_1' \geq \lambda_2' \geq \lambda_3' \geq 0$ and the corresponding eigenvectors v_1', v_2', v_3'. Similar to principal component analysis, the method uses these vectors in order to determine the 3D pointing direction. However, differing from it, there is no projection of data onto the principal directions. Rather, they are used to determine the affine vector corresponding to the pointing direction. In particular, it is observed that while the 3D pointing direction will be close to v_1', it is actually defined by the pointing part of the hand. Hence, the direction is determined by refining v_1' based on the pointing part of the hand.

This is achieved by considering the affine space defined by v_1' and the set of two orthogonal planes containing v_1' with maximal neighborhood cardinality of hand points. First, we consider the planes with normal v_2' as shown in Fig. 4a. A set of N_2 equidistant parallel planes with separation $\delta_{v_2} > 0$ between the two outermost hand points along v_2' is checked for maximal neighborhood cardinality. A point $p \in \mathcal{H}$ is considered to be in the neighborhood of a plane if its distance to the plane is less than $\epsilon_2 \delta_{v_2}$. Finally, the plane with the largest cardinality is determined. Let the respective neighboring points be denoted by $\mathcal{H}_2 \subset \mathcal{H}$. Following, the process is repeated for the v_3 direction. In this case, N_3 planes are placed at equidistant locations with respect to the vector v_3'. For each plane, the neighboring points are determined by comparing their distance to $\epsilon_3 \delta_{v_3}$. Again, the plane with the largest point cloud cardinality is selected. The set $\mathcal{H}_2 \cap \mathcal{H}_3$ consists of 3D points corresponding to the pointing part of the hand. The translation vector o is selected randomly from the set $\mathcal{H}_2 \cap \mathcal{H}_3$

and the actual 3D pointing direction estimation v_{pd} is found by considering the principal eigenvector of the covariance matrix of the set $\mathcal{H}_2 \cap \mathcal{H}_3$ going through o. For example, in the classical pointing gesture which consists of a fist and a pointing index finger, the set $\mathcal{H}_2 \cap \mathcal{H}_3$ will consist of points corresponding to the index finger as shown in Fig. 4b.

After finding the 3D direction vector, it remains to determine which end of the line is the pointing part. The human or human body part is used to resolve this. If the vector from the hand center to one end of the line makes a positive angle with the line from human center to hand center, then the corresponding end of the line is taken as the pointing side. If not, the other end is selected.

C. Noise Sensitivity

The covariance matrix Σ' associated with the 3D hand segment encodes noisy data. The relation between the noisy Σ' and true covariance matrix Σ can be approximated as:

$$\Sigma' = \Sigma + N_s \tag{1}$$

where $N_s = \sigma_a^*$ is the covariance of the noise. Typically, an empirically derived sensor noise model is formulated by Gaussian noise $\mathcal{N}(0, \sigma_a^2)$. Suppose $\sigma_a < \sigma_a^*$ is an upper bound of noise as derived from the sensor's noise model. Then $N_s = \sigma_a^* I_3$ (I_3 is a 3-dimensional identity matrix) is an worst-case estimate N_s.

Now let $\lambda_1 \geq \lambda_2 \geq \lambda_3$ be the eigenvalues of Σ. The coarse estimate of the actual hand will in fact be defined by λ_1 and the corresponding principal eigenvector v_1. First, we consider whether λ_1' of Σ' is a good approximation of λ_1 associated with Σ. This is addressed by Weyl's perturbation theorem that provides the following deterministic bound [44]:

Theorem 1: (Weyl's Inequality)

$$\max_{1 \leq i \leq 3} |\lambda_i - \lambda_i'| \leq \|N\| = {\sigma_a^*}^2 \tag{2}$$

Thus, the ordered eigenvalues of Σ are fairly stable with respect to small sensing perturbations.

Next, we consider the similarity of principal eigenvector v_1' of Σ' with v_1 of Σ. A canonical measure of similarity between two unit vectors is to evaluate their $\angle(v_1, v_1')$ taken to be in $\left[0, \frac{\pi}{2}\right]$. Define eigengap $\delta_i = \lambda_i - \lambda_{i+1}$ with $\delta = \delta_0$ The eigengap δ between the first and second eigenvalues of Σ determines the upper bound on this distance measure:

Theorem 2: (Davis-Kahan sine theorem for the first eigenvector)

$$\sin \angle(v_1, v_1') \leq \frac{\|N\|}{\delta} = \frac{{\sigma_a^*}^2}{\delta} \tag{3}$$

Note that since

$$\lambda_1' = max_{\|u\|=1} u^T (\Sigma + N) u \leq \lambda_1 + {\sigma^*}^2 \tag{4}$$

Thus, δ can be approximated by $\delta \cong \delta'$ where $\delta' = \lambda_1' - \lambda_2'$. Thus, Eq. 5 gives an upper bound on this error.

$$\sin \angle(v_1, v_1') \leq \frac{\|N\|}{\delta'} = \frac{{\sigma_a^*}^2}{\delta'} \tag{5}$$

Eq. 5 implies the following: i) Recalling that $\sigma_a \leq \sigma_a^*$, as the proximity of human-robot interaction decreases, the error of the estimated 3D pointing direction will also decrease accordingly; ii) Furthermore, as the distance between the first two eigenvalues of Σ' gets larger, v_1' becomes a better approximation of v_1. This will occur again as the human gets closer to the robot so that the variation along the pointing direction is much larger the remaining eigendirections.

IV. Pointing Gesture & 3D Pointing Direction Data Sets

Two separate benchmark data sets have been built - as there are no such publicly available data sets with distance and background variability. The data sets are obtained by a mobile robot viewing a human in HRI scenarios and acquiring RGB-D data using its Kinect sensor (with max range of 4.5 meters) located at a height of 74cm on the robot. Both are available at: `github.com/islboun/3D_Pointing_Estimation`.

The first is a 2-class data set containing RGB images of pointing and non-pointing gestures. The images are acquired by a mobile robot viewing humans performing pointing and non-pointing gestures at distances varying from 0.5m to 5 m in in various settings such as simpler (corridor) or cluttered (such as lab) environments. Following, the hand detector as developed in Section III-A is applied to these images in order to extract the bounding boxes corresponding to the hands. The bounding boxes are then used to crop the original images as to obtain the hand images. Finally, all the hand images are manually labeled as pointing or non-pointing. In total, there are 2557 pointing and 3156 non-pointing gestures. The data set is divided into training, validation and test data sets.

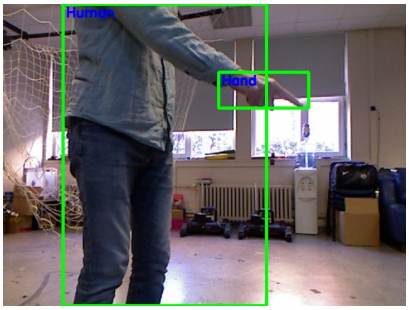
The second data set is the 3D pointing direction data set. It consists of 184 RGB-D images of hand pointing gestures with varied distance, pose and background. The rotation and translation parameters between RGB and depth sensors and the camera parameters that map 2D RGB-D data to 3D point cloud data along are provided. The viewing distances range from close to distant as seen in Table II. We have tried to include various pointing gestures that may be encountered in real-life applications. For example, the arm of the pointing hand may be extended. As such, hand locations vary from being above the robot's horizon to being below it. Another variation with the one-finger form is that the pointing finger may not be parallel to the arm. Backgrounds also vary from being less to more cluttered. Image labels contain the 3D pointing directions. The pointing hand locations in the RGB images are also provided for studies that focus purely on the performance of the direction estimation. Hand location and 3D pointing directions are manually labeled using a 3D point cloud viewer software that has been developed within our lab. Since this is a manual process, ground truth directions for close-range interactions have been easier to determine.

V. Experiments

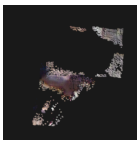
In this section, we present our experimental results. In the sequential learning model developed for hand and human

TABLE II: 3D Pointing direction data set.

Range	Hand distance range (m)	# Images
Close	0.5 - 1.5	42
Far	1.5 - 2.5	116
Distant	2.5 - 4.5	26



(a) Hand and human detection



(b) 3D hand segment.

Fig. 5: After detecting the hand and the human from RGB data, RGB-D data of the hand bounding box is used to derive the 3D hand segment.

TABLE III: Comparative performance - Hand detection on Oxford hand data set.

(a) Average precision (AP)

Method	AP
Faster R-CNN + skin[38]	49.51%
RPN [12]	57.7%
RPN + Rotation Estimator [12]	58.1%
MS-RFCN[28]	75.1%
Hand-CNN[33]	78.8%
MobileNet-SSD[54]	83.2%
Faster R-CNN + GAN[52]	87.6%
Proposed YOLOv4-based model	**84.78%**

(b) Computation time.

Method	Time (sec.)	Hardware
RPN [12]	0.1	Titan X
RPN + Rotation Estimator [12]	1	Titan X
MobileNet-SSD[54]	0.0072	Titan X
Faster R-CNN + GAN[52]	0.1121	GTX1080Ti
Proposed	**0.032**	GTX 970

detection, CNNs based on the YOLOv4 networks [7] are trained. The input RGB images are of 416×416 dimension. A larger image size may possibly lead to better accuracy, but the network would slow down. Altogether 73672 frames are used in learning. For pointing gesture detection, 4113 images are used for learning and training is done for 20 epochs.

TABLE IV: Accuracy ratings for pointing gesture classification.

Data	# Pointing Images	Accuracy (%)	# Non-Pointing Images	Accuracy (%)	# Total	Accuracy (%)
Training	1860	99.9	2253	98.8	4113	99.3
Validation	441	95.0	587	92.5	1028	93.5
Test	256	94.5	316	93.7	572	94.0

TABLE V: Comparative angular error ($\bar{e}_a$) and accuracy on IPO data set

	Proposed Method			[43]		
τ_p (°)	10°	15°	20°	10°	15°	20°
$\bar{e}_a$	6.43°	9.05°	10.69°	6°	9°	10°
Accuracy	40.4%	73.0%	93.2%	62%	83%	93.4%

A. Hand & Pointing Gesture Detection

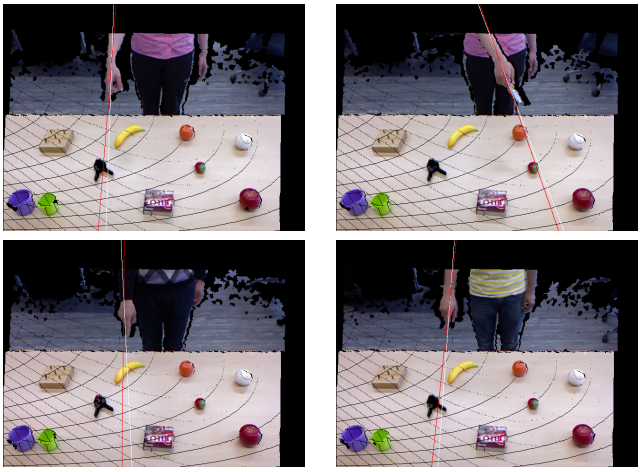
First, the performance of hand detection from RGB data is compared with the state-of-art hand detection methods. Hand detection experiments are conducted using the benchmark Oxford hand data set. Average precision and speed of hand detection are compared with the previous approaches in Table III. While the approach of [52] is the only one that has better average precision, its run-time is significantly higher than our detector. On the other hand, the fastest model [54] has significantly lower precision. Thus, with the proposed learning model, there is a good balance between average precision and run-time.

Next, the accuracy of the pointing gesture classifier is investigated. The data set that is build is divided into learning, validation and test sets.The results are given in Table IV. The accuracy of detection of positive samples is 94.5% while that negative samples is 94%. These are found to be sufficient - considering that the pointing gestures have been made at a wide range of distances to the camera and accuracy is higher than that of hand detection.

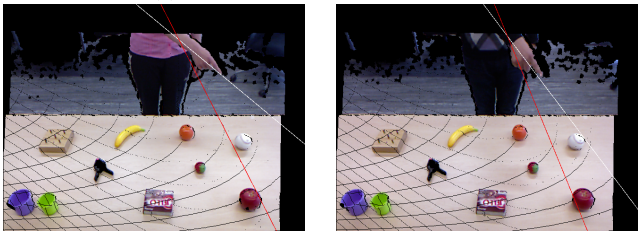
B. 3D Pointing Direction Estimation

Following, the end-to-end system is tested using a publicly available data set and our data set - including a study of performance sensitivity to the parameter values. The parameter settings are experimentally determined as $N_2 = 4$, $\epsilon_2 = 0.33$, $N_3 = 3$ and $\epsilon_3 = 0.33$ and are used consistently in all the experiments. The average execution time of the end-to-end method is 0.19 sec on a computer with 2.59 GHz CPU and GTX 1650. This time is observed to change between 0.1 sec and 0.5 sec depending on the proximity of the hand and thus the size of the hand bounding box. During testing, a sample is assumed to be correct if the angular error $e_a = \arccos(\frac{v^T \hat{v}}{(|v||\hat{v}|)}$ - namely the angle between the estimated direction ($\hat{v} \in R^3$) and ground truth ($v \in R^3$) is less than a given threshold τ_p. Average angular error $\bar{e}_a$ is computed over all samples in the test data set.

The first tests are done using the publicly available Innsbruck Pointing at Objects (IPO) data set [43]. The IPO data set contains 180 close-range RGB-D images with manually labeled ground truth. The data is acquired by a camera looking down on a scene at 0.9m to 1.5 m distance in which a person points to the different objects on a table setup. Hence, the hand is always close to the human body. It is observed that with some samples, ground truth may be slightly different it is computed considering the pose of the hand with respect to the object's centroid while in actuality, the pointing is slightly to the right or left of the centroid. In most of the frames, the top



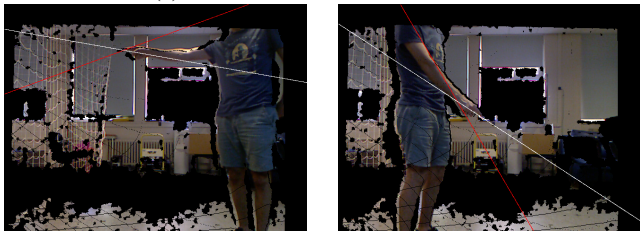
(a) Correct 3D direction estimates.



(b) Erroneous 3D direction estimates.

Fig. 6: Sample erroneous estimations in IPO data set. Estimated 3D direction (white) and ground truth direction (red) are displayed on RGB data.

(a) Correct 3D direction estimates.



(b) Erroneous 3D direction estimates.

Fig. 7: Sample results from the new data set. Estimated 3D direction (white) and ground truth direction (red) are displayed on RGB-D data.

of the pointing hand is facing upwards. In such cases, the main challenge is to account for hand points that are almost equally distributed along all directions in this case. Sample results with no angular error are shown in Fig. 6a. The estimations are observed to be correct if the pointing finger is fully extended. Errors can possibly occur if the pointing gesture has a non-prototypical view. For example, with the one-finger form, this happens when the finger's extension is not full, but rather is bent in a curvy manner. Some such cases are shown in Fig. 6b. Interestingly, the ground truth pointing direction is extremely difficult to discern in these cases. Performance statistics are given in Table V along with a comparative study[1] - considering three different precision levels as defined by threshold levels 20°, 15° and 10°. It is observed that for an average angular error $\bar{e}_a \cong 10°$ (corresponding to threshold of 20°) accuracy is around 93% for both of the approaches. In the remaining two cases, while the performance of [43] is slightly better, it should be noted that the method is only applicable for close-range applications. In this range, pointing directions with angular variations less that 10° are both hard to discern and are likely to point to the same entity anyways.

Next, the performance with the newly introduced 3D pointing direction data set is studied. Sample scenarios with increasing hand distance and varying arm pose in which the direction is estimated correctly are shown in Fig. 7a. Note that even if the hand's orientation (ie palm or top showing) changes or arm pose changes, the robot is able to estimate the pointing direction reliably. It is observed that with one-finger pointing gestures in which either finger shape or direction are not in its prototypical forms, the estimate may be erroneous. Sample cases are shown in Fig. 7b. The first case is due to be the finger being slightly bent. The second case is also similar, but it is much more subtle and it is hard even for the external viewer to detect the ground truth direction. Again, three angular error thresholds are considered - 20°, 15° and 10°. In the first case, the average estimation error is 10.57° with an accuracy of 83.7% accuracy. As expected, this is lower than that of the close-range only case. This is attributed to the

[1]Note that results of [43] are not exact since they are taken from a plot.

TABLE VI: New data set results

(a) Average angular error ($\bar{e}_a$) and accuracy for varying thresholds

τ_p	10°	15°	20°
$\bar{e}_a$	6.73°	8.82°	**10.57°**
Accuracy	41.3%	66.3%	**83.7%**

(b) Variation vs. hand proximity.

Distance	$\bar{e}_a$ (°)	Accuracy (%)
Close	10.97	77.3
Far	10.67	84.2
Distant	9.62	92.3

TABLE VII: Accuracy rates and average angular error ($\bar{e}_a$) for different parameter selections.

(N_2, ϵ_2) (N_3, ϵ_3)	$\bar{e}_a$ (°)	Accuracy (%)
(3,0.5) (2,0.5)	11.24°	83.1
(4,0.33) (2,0.5)	10.94°	81
(4,0.33) (3,0.33)	10.57°	83.7

fact that sensing noise increases with distance as discussed. The results are in agreement with the previous finding that 'it is possible to disambiguate possible pointing targets with an average error of less than 20° at a distance of about 2.5 meters' [45], [43]. The variations of average angle error and accuracy with respect to the hand distance are as shown in Table VIb. Interestingly, while average angular error tends to be around 10° in all three cases, accuracy tends to increase with distance. This is somewhat surprising considering that the data noise increases with distance. With close analysis, this is attributed to the fact that ground truth 3D directions are computed more accurately in close-range interactions as compared to far range as it is easier to disambiguate the pointing targets in the point cloud data.

Recall that the data is obtained with Kinect sensor. As Kinect depth data is known to be noisy, various noise models that are either theoretically or empirically derived have been presented [34], [24]. In all, the spread of axial noise distributions (obtained in meters) increases quadratically with distance. In [24], the empirically derived model is formulated by Gaussian noise $\mathcal{N}(0, \sigma_a^2)$ with the standard deviation defined as:

$$\sigma_a(\rho,\theta) = \begin{cases} 0.0012 + 0.0019(\rho - 0.4)^2 \text{ if } |\theta| \leq 60^\circ \\ 0.0012 + 0.0019(\rho - 0.4)^2 \\ \quad + \frac{0.0001}{\sqrt{\rho}} \frac{\theta^2}{(\frac{pi}{2} - \theta)^2} \quad \text{otherwise} \end{cases} \tag{6}$$

In this model, θ refers to the angle between sensor normal and hand normal. It is noted that as $\theta \rightarrow 90^\circ$, the measurement becomes unstable. If $\theta \leq 80^\circ$ and $0.5 \leq \rho \leq 4.5$, then $0.0012 \leq \sigma_a \leq 0.0335$ m. Then, the upper bound on σ_a is defined by $\sigma_a^* = 0.0335$m. It corresponds to the maximum of standard deviation of depth measurements as defined in Section III-B.

Finally, the effect of parameter values used in the geometric reasoning on are also investigated. Recall that there are 4 parameters: (N_2, ϵ_2) values for planes orthogonal to v_2 and (N_3, ϵ_3) values for planes orthogonal to v_3. The results are given in Table VII. It is observed that the performance under different set of parameter values are close to each other. Hence, the estimated directions can be estimated reliably even with small parameter variations.

In summary, most related works have assumptions regarding either hand proximity, background variability or the detectability of specific human parts. The only work with 3D pointing direction labeled data set contains scenarios only from close range - namely 0.9-1.5m [43]. As seen in Table V, the results are similar. However, their approach is only applicable in close-range. In contrast, the proposed approach can also be used reliably in mid and far range as shown in Table VI.

VI. Conclusion

This work proposes a novel approach that enables a mobile to estimate the 3D hand pointing direction based on RGB-D data from a single perspective. The proposed approach is motivated by the need for natural human-robot interaction. The focus is on ubiquitous pointing gestures such as one-finger or flat hand forms with pointing directed towards frontal hemisphere. Differing from previous approaches, the robot is able to estimate the 3D pointing direction with high accuracy rates regardless of hand proximity, background variability or specific human parts being visible. We also present a noise sensitivity analysis that can be used to find an upper bound on the estimation error. As part of this work, we release two new data sets for pointing gestures and 3D pointing directions with varied hand distance, arm pose and background. To the best of the authors' knowledge, there are no such data sets available, and hence we believe both are also valuable contributions to the community. Our end-to-end experimental results demonstrate the robot can estimate the 3D hand direction both accurately and fast regardless of hand proximity, arm pose, background variability or the detectability of specific human parts. To the best of our knowledge, this is the first framework than can accurately predict the 3D pointing direction from a single RGB-D image with the least amount of assumptions and is real-time applicable on mobile robots. For future work, we plan to consider whether the temporal nature of the incoming RGB-D data can be used to improve the accuracy. We also plan to extend the approach to the inference of 3D pointing direction associated with pointing gestures involving multiple body parts such as face-to-hand and wrist-to-hand.

Acknowledgments

This work has been supported in part by TUBITAK EEEAG-118E857. We thank Özgür Erkent for his comments on an earlier version of the paper. We also thank Mehmet Yasin Özkan for his contributions during the collection of the 3D Pointing Direction data set.

References

[1] S. Abidi, M. Williams, and B. Johnston. Human pointing as a robot directive. In *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 67–68, 2013. 2, 3

[2] A. A. Alani, G. Cosma, Aboozar Taherkhani, and T. McGinnity. Hand gesture recognition using an adapted convolutional neural network with data augmentation. *2018 4th International Conference on Information Management (ICIM)*, pages 5–12, 2018. 4

[3] K. S. Arun, T. Huang, and S. D. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9:698–700, 1987. 4

[4] B. Azari, A. Lim, and R. Vaughan. Commodifying pointing in hri: Simple and fast pointing gesture detection from rgb-d images. In *16th Conference on Computer and Robot Vision (CRV)*, pages 174–180, 2019. 2, 3

[5] O. L. Barbed, P. Azagra, L. Teixeira, M. Chli, J. Civera, and A. C. Murillo. Fine grained pointing recognition for natural drone guidance. *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 4480–4488, 2020. 2, 3

[6] M. K. Bhuyan, D. R. Neog, and Mithun Kumar Kar. Fingertip detection for hand pose recognition. *Int'l J. of Advanced Trends in Computer Science and Engineering*, 4(3):501–511, 2012. 3

[7] Alexey Bochkovskiy, Chien-Yao Wang, and H. Liao. Yolov4: Optimal speed and accuracy of object detection. *ArXiv*, abs/2004.10934, 2020. 6

[8] P. Buehler, M. Everingham, D. Huttenlocher, and A. Zisserman. Long term arm and hand tracking for continuous sign language tv broadcasts. In *Proceedings of the British Machine Vision Conference*, pages 110.1–110.10. BMVA Press, 2008. doi:10.5244/C.22.110. 2

[9] Kensy Cooperrider. Fifteen ways of looking at a pointing gesture. *PsyArXiv*, doi:10.31234/osf.io/2vxft, 2020. 1

[10] Shome S. Das. Precise pointing direction estimation using depth data. *2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 202–207, 2018. 1, 2, 3

[11] A. Y. Dawod, J. Abdullah, and M. J. Alam. Adaptive skin color model for hand segmentation. In *2010 International Conference on Computer Applications and Industrial Electronics*, pages 486–489, 2010. 2

[12] X. Deng, Y. Zhang, S. Yang, Ping Tan, Liang Chang, Ye Yuan, and H. Wang. Joint hand detection and rotation estimation using cnn. *IEEE Transactions on Image Processing*, 27:1888–1900, 2018. 2, 6

[13] D. Droeschel, J. Stückler, and Sven Behnke. Learning to interpret pointing gestures with a time-of-flight camera. *ACM/IEEE Int'l Conf. on Human-Robot Interaction*, pages 481–488, 2011. 2, 3

[14] Eng-Jon Ong and R. Bowden. A boosted classifier tree for hand shape detection. In *Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004. Proceedings.*, pages 889–894, 2004. 2

[15] A. Fernandez, L. Bergesio, A. Bernardos, J. Besada, and J. Casar. A kinect-based system to enable interaction by pointing in smart spaces. *2015 IEEE Sensors Applications Symposium (SAS)*, pages 1–6, 2015. 2, 3

[16] Dai Fujita and T. Komuro. Three-dimensional hand pointing recognition using two cameras by interpolation and integration of classification scores. In *ECCV Workshops*, 2014. 1, 2, 3

[17] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 4

[18] Kaoning Hu, Shaun J. Canavan, and L. Yin. Hand pointing estimation for human computer interaction based on two orthogonal-views. *Int'l Conf. on Pattern Recognition*, pages 3760–3763, 2010. 2, 3

[19] Yichao Huang, X. Liu, X. Zhang, and Lianwen Jin. A pointing gesture based egocentric interaction system: Dataset, approach and application. *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 370–377, 2016. 3

[20] V. Jain, G. Garg, R. Perla, and R. Hebbalaguppe. Gestarlite: An on-device pointing finger based gestural interface for smartphones and video see-through head-mounts. *ArXiv*, abs/1904.09843, 2019. 3

[21] Shruti Jaiswal, P. Mishra, and G. C. Nandi. Deep learning based command pointing direction estimation using a single rgb camera. *2018 5th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)*, pages 1–6, 2018. 2, 3

[22] L. Karlinsky, M. Dinerstein, D. Harari, and S. Ullman. The chains model for detecting parts by their context. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 25–32, 2010. 2

[23] Charles C. Kemp, Cressel D. Anderson, Hai Nguyen, Alexander J. Trevor, and Zhe Xu. A point-and-click interface for the real world: Laser designation of objects for mobile manipulation. In *2008 3rd ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 241–248, 2008. 1

[24] Kourosh Khoshelham and Sander Oude Elberink. Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors*, 12(2):1437–1454, 2012. 8

[25] S. Kita. *Pointing: Where language, culture and cognition meet*. Psychology Press, 2003. 1

[26] M. Kolsch and M. Turk. Robust hand detection. In *Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004. Proceedings.*, pages 614–619, 2004. 2

[27] M. P. Kumar, A. Zisserman, and P. H. S. Torr. Efficient discriminative learning of parts-based models. In *2009 IEEE 12th International Conference on Computer Vision*, pages 552–559, 2009. 2

[28] T. Le, Kha Gia Quach, Chenchen Zhu, C. N. Duong, K. Luu, and M. Savvides. Robust hand detection and classification in vehicles and in the wild. *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1203–1210, 2017. 2, 6

[29] T. H. N. Le, Y. Zheng, C. Zhu, K. Luu, and M. Savvides. Multiple scale faster-rcnn approach to driver's cell-phone usage and hands on steering wheel detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 46–53, 2016. 2

[30] D. A. Leavens and W.D. Hopkins. The whole-hand point: the structure and function of pointing from a comparative perspective. *J Comp Psychol.*, 113(4):417–425, 1999. 1

[31] Tsung-Yi Lin, M. Maire, Serge J. Belongie, James Hays, P. Perona, D. Ramanan, Piotr Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 2, 3

[32] A. Mittal, Andrew Zisserman, and P. Torr. Hand detection using multiple proposals. In *BMVC*, 2011. 2

[33] Supreeth Narasimhaswamy, Zhengwei Wei, Yang Wang, Justin Zhang, and Minh Hoai. Contextual attention for hand detection in the wild. In *International Conference on Computer Vision (ICCV)*, 2019. 2, 3, 6

[34] Chuong V. Nguyen, Shahram Izadi, and David Lovell. Modeling kinect sensor noise for improved 3d reconstruction and tracking. In *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization Transmission*, pages 524–530, 2012. 8

[35] K. Nickel and R. Stiefelhagen. Visual recognition of pointing gestures for human-robot interaction. *Image Vis. Comput.*, 25:1875–1884, 2007. 2, 3

[36] M. Pateraki, H. Baltzakis, and P. Trahanias. Visual estimation of pointed targets for robot guidance via fusion of face pose and hand orientation. In *ICCV Workshops*, 2011. 1

[37] A. Rahman, J. A. Mahmud, and M. Hasanuzzaman. Pointing and commanding gesture recognition in 3d for human-robot interaction. In *Int'l Conf. on Innovation in Engineering and Technology*, pages 1–10, 2018. 2

[38] Kankana Roy, Aparna Mohanty, and R. R. Sahay. Deep learning based hand detection in cluttered environment using skin segmentation. *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 640–649, 2017. 2, 6

[39] Akanksha Saran, Elaine Schaertl Short, Andrea Thomaz, and Scott Niekum. Enhancing robot learning with human social cues. In *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 745–747, 2019. 1

[40] Allison Sauppé and Bilge Mutlu. Robot deictics: How gesture and context shape referential communication. In *2014 9th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 342–349, 2014. 1

[41] Chris L. Schmidt. Adult understanding of spontaneous attention-directing events: What does gesture contribute? *Ecological Psychology*, 11(2):139–174, 1999. 1

[42] Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. Real-time human pose recognition in parts from single depth images. In *CVPR 2011*, pages 1297–1304, 2011. 2

[43] D. Shukla, Özgür Erkent, and J. Piater. Probabilistic detection of pointing directions for human-robot interaction. *2015 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–8, 2015. 2, 3, 6, 7, 8

[44] G. W. Stewart and Ji guang Sun. *Matrix Perturbation Theory*. Academic Press, 1990. 5

[45] R. Stiefelhagen, C. Fugen, R. Gieselmann, H. Holzapfel, K. Nickel, and A. Waibel. Natural human-robot interaction using speech, head pose and gestures. In *IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, volume 3, pages 2422–2427 vol.3, 2004. 3, 8

[46] Jonathan Tompson, Murphy Stein, Yann Lecun, and Ken Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics*, 33(5), 2014. 2

[47] M. Van den Bergh, D. Carton, R. De Nijs, N. Mitsou, C. Landsiedel, K. Kuehnlenz, D. Wollherr, L. Van Gool, and M. Buss. Real-time 3d hand gesture interaction with a robot for understanding directions from humans. In *2011 RO-MAN*, pages 357–362, 2011. 2, 3

[48] Wei Wang and Jing Pan. Hand segmentation using skin color and background information. In *2012 International Conference on Machine Learning and Cybernetics*, volume 4, pages 1487–1492, 2012. 2

[49] M. Wnuczko and J. M. Kennedy. Pivots for pointing: Visually monitored pointing has higher arm elevations than pointing blindfolded. *Journal of Experimental Psychology: Human Perception and Performance*, 37:1485–1491, 2011. 1

[50] Nelson Wong and Carl Gutwin. Where are you pointing?: the accuracy of deictic pointing in cves. In *SIGCHI Conf. on Human Factors in Computing Systems*, page 1029–1038, 2010. 3

[51] Wenbin Wu, C. Li, Zhuo Cheng, X. Zhang, and Lianwen Jin. Yolse: Egocentric fingertip detection from single rgb images. *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*,

pages 623–630, 2017. 3

[52] Chi Xu, Wendi Cai, Yongbo Li, J. Zhou, and Longsheng Wei. Accurate hand detection from single-color images by reconstructing hand appearances. *Sensors (Basel, Switzerland)*, 20, 2020. 2, 6

[53] Chi Xu, J. Zhou, Wendi Cai, Yunkai Jiang, Yongbo Li, and Y. Liu. Robust 3d hand detection from a single rgb-d image in unconstrained environments. *Sensors (Basel, Switzerland)*, 20, 2020. 2

[54] L. Yang, Zhi Qi, Zeheng Liu, H. Liu, Ming Ling, L. Shi, and Xinning Liu. An embedded implementation of cnn-based hand detection and orientation estimation algorithm. *Machine Vision and Applications*, pages 1–12, 2019. 2, 6