Gaze Direction Estimation under Varying Head Positions using a Pepper Robot's In-Built Camera

Marinos Savva
Vrije Universiteit Amsterdam
Department of Computer Science

July 21, 2021

Abstract

Eyes are often described as the mirror to the soul. Eye gaze direction can provide considerable insight about ones underlying psychological state as well as describe intent in a social interaction. This research paper describes an approach for combining eye and head directional data in estimating gaze direction without the use of specialized equipment and with a low fidelity, built-in Pepper robot camera. Detecting human gaze direction in a human-robot interaction could provide a foundation for systems in experimental psychology and social artificial intelligence gauging behavioural features such as attention or task engagement. Head and eye directions are estimated and combined through appearance and mathematical methods based on a facial landmark prediction model, utilizing statistical filtering methods such as moving average and Kalman for noise reduction. The system is tested on a group of human participants to evaluate performance. Results show that algorithm is adequate at determining gaze direction of participants at a distance of 80cm, with an average accuracy of 67% along the x-axis decreasing to 51% at 120cm. Results on accuracy along the y-axis where inconclusive due to experimental limitations.

1 Introduction

In a natural social interaction a plethora of social cues are exchanged, from facial expressions to speech and gestures, all serving the aim of communicating information to our conversational partner. As such, gazing behaviour does not only serve to attend to such information but it can also function as a transmitter of information as well [1]. In human-to-human interaction, extensive research has been made in an effort to understand and interpret the psychological underlying of gazing behaviour [2–5].

Measuring gazing behaviour in a natural conversation, could open up further research opportunities in the the fields of experimental psychology and artificial intelligence. When it comes to studying the psychology of gazing behaviour, research does not always translate from an experimental setting to the real world as it fails to capture the nuances of a natural social context [6]. Similarly, social cognition is intrinsically different when measured from within a social interaction rather than from an observer's perspective. Hence, methods that allow the real-time study of social encounters in an interactive setting are especially valuable [7]. By implementing gaze tracking in experimental robots we can analyze behaviours from within such social interactions from the perspective of a participant. As for the field of artificial intelligence, gaze detection can be utilized as a measure of conversational attention and interaction engagement [8–10]. It has also been shown that gaze following and joint attention increase engagement as well as likeability in a human-robot interaction [11].

Work so far has focused on tracking gaze direction from information extracted exclusively from the eyes. Gaze tracking techniques have been developed that utilize unique features of the eye such as corneal reflections and glints produced by infared lighting or other visible light sources [12, 13]. These feature-based techniques provide exceptionally accurate results but tend to suffer outside a controlled lab environment and are especially sensitive to environmental lighting. Research has also been done with shape-based techniques which work by using elliptical [14–16], or more complex, geometric eye models [17, 18]. This category of techniques is especially effective and robust in varying scales and head poses. Others used appearance-based techniques basing their calculation on measures such as pixel intensities and color distribution constructed through regression to localize the eye [19, 20]. Although specializing in both indoors and outdoors scenarios with varying illumination, these techniques require large amounts of training data. In processing, implementations have also seen the use of neural networks [21, 22]. Other researchers took a more mathematically grounded approach to gaze tracking operations by analyzing the cumulative distribution function of pixels in regions of interest to localize the eye [23, 24]. Research has also focused in combining both head and eye data to calculate gaze direction in the wild. Valenti et al.proposed a hybrid scheme combining the two measures where the eye location is utilized to correct the pose estimation procedure [25], while Lu et al. proposed a combination of learning-based and geometric-based methods for combining head and eye data[26]. Gaze tracking has also been implemented in robots such in the case of Saran et al., utilizing deep neural networks to generate a heat map of gaze target probability [27].

In this project we will be addressing the following questions; how can we estimate human eye gaze direction from a low fidelity camera input? Due to be dealing with noisy sensor data and uncontrolled lighting variability, statistical models, namely moving average and Kalman filters, will need to be investigated. More importantly, how can we implement such an algorithm using a mobile Pepper robot so that estimate gaze direction is accurate in varying head positions? Since eye gaze direction is relative to head pose, a system will need to be developed that combines the two into a single gaze direction. The system will then be tested in an experimental settings with human participants.

2 Methods

2.1 Frame Rate, Resolution and Computational Time

Performing image and gaze tracking using a common everyday camera presents a balancing challenge between camera frame rate, image spatial resolution and computational time. This is especially challenging when using an embedded camera with frame rate and resolution being quite restricted.

With the human eye being able to reach extraordinary speeds, frame rate is vital for accurately tracking its movements. A saccade, the fastest type of eye movement, can have a duration ranging anywhere from 10ms to 100ms [28] with typical values lying in the 10ms to 40ms range, meaning that in order for a camera to register the entirety of the eye movement spectrum it requires a frame rate of at least 10 to 20 frames per second up to a maximum of 200 frames per second. Thus, a higher frame per second boundary correlates with a higher likelihood of detecting eye movement with fewer events going unnoticed. However, an everyday commodity camera is highly unlikely to be able to reach any range higher than 60 frames per second. Similarly in the case of Pepper's built-in camera, there are only two frame rate settings for the camera; one with a limit of 15 frames per second and one with 30, which are dependent on image resolution. When bench marked, the realistic frame rates of the two settings are 12 and 25 frames per second respectively.

Image spatial resolution is also essential to the current project. A higher resolution provides more accurate and reliable detection of spatial information [29] and especially when detecting facial landmarks. For each frame rate setting mentioned previously, Pepper's camera can capture images at resolutions of 640x480 and 320x240 respectively.

However, in the case of a real-time application a higher volume of data is more often than not accompanied by higher processing times which could have throttling effects to the speed and overall robustness of our system. Considering our in-built camera is capable of capturing live video at a maximum of 30 frames per second, each iteration of image processing cannot exceed 0.03 seconds as that would result in a bottleneck, backing up our frames and in effect cause a delay in detection. Considering the projects goal of a real-time system for social interactions, speed will be one of our primary concerns. Reliability and robustness are of equal importance as our system will need to perform under varying circumstances.

2.2 Gaze Tracking

Attempting to detect eye movement from appearance based factors without the use of specialized eye tracking equipment presents a plethora of challenges when it comes to reliability. By using a combination of eye gaze direction and head pose estimates we can create an increasingly more reliable system than one using just a single appearance based measure [30]. This has previously been replicated, where a combined adaptive estimate of the head pose combined with the eye gaze direction resulted in an increase in accuracy from using merely the eyes [31].

2.2.1 Face Detection and Landmark Prediction

Human facial data, specifically facial landmarks are informative in operations requiring head orientation or region of interest extraction. As such, the head pose and eye gaze direction estimations will be utilizing the human facial landmarks. By applying face detection to a given video frame first, the area of interest can be reduced significantly to a bounding box of the face henceforth greatly improving the speed of the landmark predictor. For the face detection the dlib library's frontal face detector was chosen as it outperforms other OpenCV algorithms tested, namely Haarcascades and DNN [32] and faster when compared in practise. To further increase the speed of the face detector, the captured frame is converted into gray scale, effectively reducing the amount of data needed to be processed by the face detector by x3, as only one color channel is compared for gray-scale values instead of three for RGB. 68 facial Landmarks (see Figure 1) are detected through a function provided by the dlib library based on the work of Kazemi & Sullivan [33] and uses the bounding box generated by the face detector as input in order to detect facial features using a pre-trained shape predictor model.

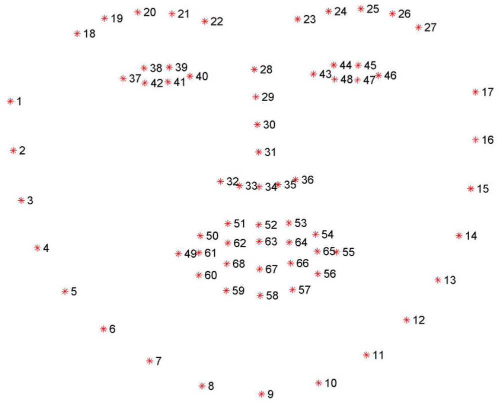

Figure 1: A visual representation of the facial landmarks predicted by the dlib shape predictor

2.2.2 Head Pose Estimation

Estimating the head pose is achieved by matching the facial landmarks of the two-dimensional frame to key features in a generic three-dimensional ellipsoid-face model [34, 35].

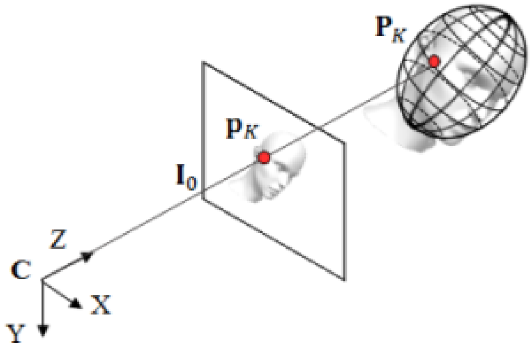

Figure 2: Computation of a single 3D key-point (P_k) using a 2D facial landmark (p_k given in the visual frame (I_0 (Source: [35])

Using OpenCV's perspective-n-point solver function, the pose of the calibrated camera is calculated given the three-dimensional world model key-points and their corresponding two-dimensional facial landmarks. By inverting the camera pose, a rotational vector for the head in respect to the camera position is created and converted into a rotation matrix using Rodrigues' formula. In turn, by using the formulas for Euler angles on the resulting matrix, the head's yaw and pitch are calculated.

2.2.3 Eye Gaze Angle Estimation

To estimate the eye gaze angle, a combination of an appearance based approach to localize the pupil utilizing the variation in pixel intensities and a model based approach for pupil contouring was adopted. Due to the multitude of constraints of eye detection this was the most feasible approach as no specialized equipment would be available for a feature based approach and data collection for a neural network based method would be hard. However, it is noteworthy to mention that appearance based methods suffer greatly in low lighting environments comparatively to others and as such we need to pay close attention to the image contrast [30].

The eye's region of interest is located using the landmarks acquired previously. Creating a mask layer, we extract it from the rest of the frame and perform a series of image processing techniques to localize the pupil. To do this, the image is initially blurred with a Gaussian blur to remove noise before applying contrast stretching through histogram equalization, in effect increasing the contrast between the white of the sclera and the dark of the pupil [22].

Through an inverted binary threshold a black and white image is created where pixel colour values exceeding a pre-set threshold are reassigned to the colour white and the rest of the lighter areas to the colour black. To further improve the thresholded image a morphological closing operation is performed, also known as a dilation followed by erosion, filling up any possible white spots in the pupil caused by surface light reflections. A morphological opening operation, also known as an erosion followed by dilation, removes any noise created during the thresholding process. Both morphological operations use a symmetrical kernel maintaining the cornea's circularity and for smaller images a smaller kernel is used avoiding data loss.

Using the the post-processed eye image, white space contours are detected and then sorted by area, with the largest contour, which should represent the highest concentration of darker pixels and by proxy the cornea, selected. The selected contour's convexity is increased making up for any surface loss created during processing. To remove any false positives from being detected the contour is tested for circularity by calculating the isoperimetric quotient given by the ratio of the curve area to the area of the circle with the same perimeter. Lastly, by detecting the concentration of pixel intensity or image moments the central coordinates of the cornea are calculated and the pupil localized. A visualisation of each of the image processing method and other operation steps used can be found in Figure 9 in the Appendix.

To mathematically estimate the gaze angle a simplified model of the eyeball is used, considering the distinct nature of the human eye. Firstly, we generalize the shape of the eyeball to that of a sphere with constant radius. Secondly, we assume that the radius of the eyeball can be approximated to the average human eyeball radius from the centre of rotation which is 10.94mm [36]. Thirdly, since the visual axis would be extremely hard to calculate in such a system, we assume no angle kappa (see Figure 3 for explanation); assuming no difference between the pupillary and the visual axes [37].

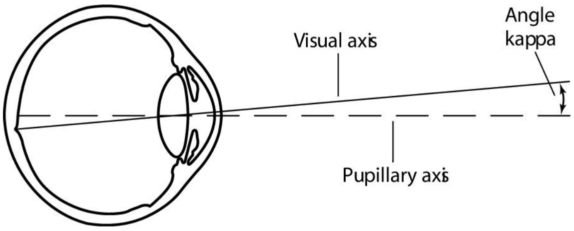

Figure 3: The two main axes of the human eye are the visual axis and the pupillary axis. The visual axis is the line formed by the eye's fixation point and the center of the fovea, indicating the eye's line of view. The pupillary axis is the line perpendicular to the cornea passing through the center of the eyeball. Angle kappa is the angle formed between these two axes.

Since the corneal displacement forms a perpendicular line with the one formed by the center of rotation and the eyeball center, we can use right triangle trigonometric operations to calculate the eye angle. The same logic can also be used for the y angle of the eye gaze angle by replacing the horizontal corneal displacement (dx) with the vertical one (dy).

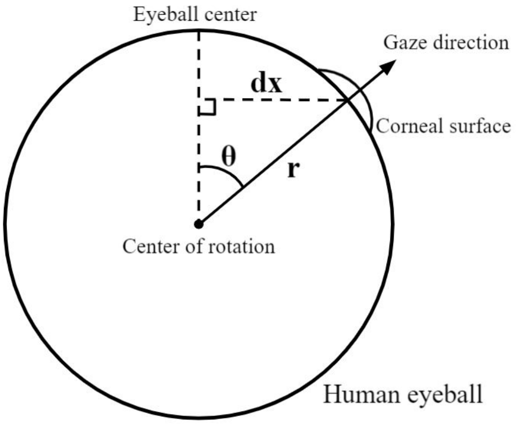

Figure 4: Relationship between corneal displacement (dx), eye gaze angle (θ) and eyeball radius (r)

Knowing that in a right triangle the sine of an angle is given by the opposite side of the triangle over the hypotenuse we can extract the following equation, where dx represents the corneal displacement, r the radius of the eyeball and θ the eye angle:

$$\theta = \arcsin(\frac{d\text{x}}{r}) \tag{1}$$

2.2.4 Combined gaze target estimation

When combining head pose and eye gaze direction, the former will be used as the base of the calculation while the latter will improve the result. The reason for this is because of the head pose angle being detected more reliably with less noise and since in extreme angles the pupils can be obstructed from the camera view. We first calculate the distance of the projection from the origin of the two angles onto the visible frame (the entire detected image).

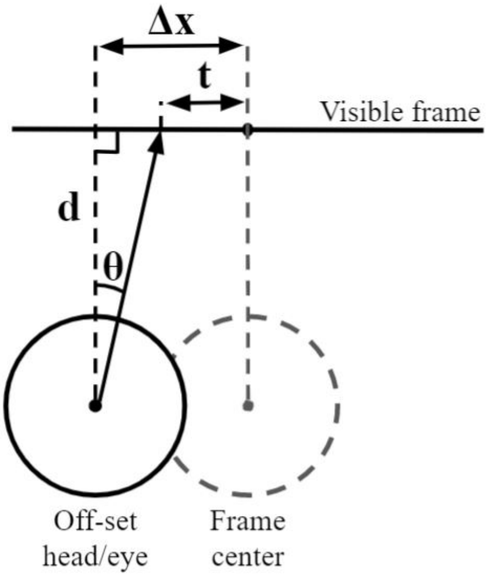

Figure 5: Projection of eye or head angle onto frame in off-centered base position

The relationship between an angle θ, its projection onto the frame and the perpendicular distance of d can be described by $\tan(\theta)$ being equal to its opposite over its adjacent side. The same equation can be used for both the head and eye calculations, as the only thing different would be the angle used. Solving for distance of the projection from the origin (t), henceforth referred to as target coordinates:

$$\mathrm{t} = \Delta\mathrm{x} - \mathrm{d} \times \tan(\theta) \tag{2}$$

Computing the combined target coordinates, results of the head and eye target coordinates are summed giving a larger magnitude to the eyes, since although dependent on the head pose provide a better estimator of exact gaze location. However, due to the nature of the calculations being based on angular values, the projection results get exponentially larger as the base angles increase. To counteract that interaction two coefficients ($\mathrm{k}_1, \mathrm{k}_2$) are provided increasing the effects of the eye gaze target coordinates (t_{eyes}) when the head pose target coordinates (t_{head}) becomes exceedingly large while also reducing the effects of the head pose itself. In order to determine whether a target coordinate is large it is compared with the frames width referred to as w in the equations. Putting everything together yields the final combined gaze target coordinates (t_{comb}).

$$\mathrm{k}_1 = \frac{|\mathrm{t}_{head} - \Delta\mathrm{x}| - \frac{2}{\mathrm{w}} \times \mathrm{t}_{head} \times \Delta\mathrm{x}}{\mathrm{w}} \tag{3}$$

$$\mathrm{k}_2 = \frac{|\mathrm{t}_{eyes} - \mathrm{t}_{head}| - \frac{2}{\mathrm{w}} \times \mathrm{t}_{eyes} \times \mathrm{t}_{head}}{\mathrm{w}} \tag{4}$$

$$\mathrm{t}_{comb} = (1 + \mathrm{k}_1 + \mathrm{k}_2) \times \mathrm{t}_{eyes} + (\mathrm{k}_1 - \mathrm{k}_2) \times \mathrm{t}_{head} \tag{5}$$

The same equation is used in the vertical axis by replacing the head displacement from the visual frame center along the x-axis (Δx) with the displacement along the y-axis (Δy) and by replacing the frame width (w) with the frame height (h).

2.3 Real world and digital unit operations

For the aforementioned calculations to be valid it is important that the distance of the participant from the camera is represented in digital pixel units. Thus, to ensure the calculations are correct we need to digitize all real-world measures, converting them to pixel based units. For this operation a chessboard pattern will be used as the OpenCV library has a built-in function to detect chessboard corners in an image. By placing the pattern at a selected distance, the measured chessboard digital width can be divided by its real-world equivalent, resulting in a ratio for the specific distance. This ratio can be multiplied to any real-world measure to convert it to a digital pixel based one given its corresponding distance from the camera.

2.4 Noise Reduction

The lack of specialized measuring equipment, the low camera resolution as well as the variation in environment lighting are some of the few but most notable causes of noise when trying to calculate the gaze direction. In order to prevent noise from majorly affecting our calculations two main statistical noise reduction methods are used.

2.4.1 Moving Average Filter

A moving average filter performs well when removing extreme outliers from the operations. The implemented method works in two ways. If a value provided approximates the mean of previously observed values then the mean is used to more accurately predict its true value without noise. To avoid overfitting the data, only the latest 10 observations are used to calculate the mean. Any more observations would result in the observation mean being impervious to the effect of future values. If an angle value provided is significantly different from the mean of previously observed values then that value is assumed an extreme outlier and hence ignored. The error margin differs for each estimate, ranging from 0.4 radians to 0.6 based on the range of available angle observations for the eye and head respectively. The filter is used in both calculations of the target coordinates of the head and eyes.

2.4.2 Kalman Filter

In most operations the moving average filter does not suffice in completely reducing noise thus another statistical method, the Kalman filter, is applied. The Kalman filter performs exceptionally well with uncertainty, constant streams of measures provided over time and being extremely lightweight makes it ideal for real-time applications [15]. Using a probabilistic approach the filter can narrow down on the true value after just a few inputs. It works by determining confidence in measurement accuracy of a given value t based on the previous error in estimation (E_{est}) and constant error in measurement (E_{mea}). This confidence is given as a probability, the Kalman gain (KG), where a value of 1 signifies a high probability of an accurate reading and a value of 0 indicates a high probability of an erroneous reading. The Kalman gain is used to then calculate an estimate (EST_t) for the given value utilizing the last known estimate (EST_{t-1}), before recalculating the new error in estimation to be used in future iterations.

$$\mathrm{KG} = \frac{\mathrm{E}_{est}}{\mathrm{E}_{est} + \mathrm{E}_{mea}} \tag{6}$$

$$\mathrm{EST}_t = \mathrm{EST}_{t-1} + \mathrm{KG} \times [\mathrm{t} - \mathrm{EST}_{t-1}] \tag{7}$$

$$\mathrm{E}_{est_t} = [1 - \mathrm{KG}] \times (\mathrm{E}_{est_{t-1}}) \tag{8}$$

The Kalman filter is applied to observed values after the moving average filter. The two statistical methods work well in conjunction, with the moving average filter correcting average values and eliminating extreme ones, and the Kalman filter improving overall accuracy of neighbouring results. When tested in an informal experiment with a single round of the five points as mentioned in 3.1.4 the combination of the methods provided a significant decrease in noise (see Figure 10 for comparison of the two filters separately).

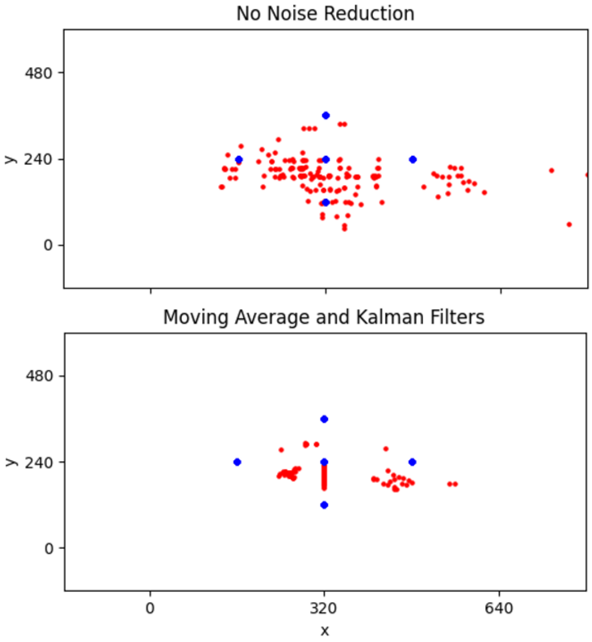

Figure 6: Comparing the effect of no filters for noise reduction with the combination of the moving average and Kalman filters. The blue dots represent the expected coordinates and the red dots the gaze direction estimates.

3 Results

3.1 Experimental Setup

3.1.1 Participants

Twelve participants were recruited randomly from an opportunity sample to participate in testing the effectiveness of the algorithm. The amount of participants would be sufficient for establishing an indication of performance and useful for observing variability between individual participants. The participants were either lab personnel or acquaintances that were available and willing to participate.

3.1.2 Design

All participants were used in all testing phases of the experiment. To test the accuracy of the gaze tracking algorithm the error in Euclidean distance would need to be calculated. The distance between each of 5 the projected points (illustrated by the blue points in Figure 8 and as seen on screen during the experimentation in Figure 7) and the corresponding estimates was used to evaluate the algorithm. Other variables were also measured to provide context in case of result deviation such as the eye and head angles measured in each position (see 3.1.4 for positions). Additionally, the height of the participant was measured as it could hypothetically have an effect on head pose estimation while the eye color was measured to explain possible variations in eye angle likely caused by image processing.

3.1.3 Materials

For the experiment a laptop, a large display of a minimum width of 80cm, a version 1.8a Pepper robot, measuring tape and tape where needed. The Pepper robot was turned on and placed in the front-middle of the large display, facing away and perpendicular to the display, with its head tilted at 0.4 radians vertically and 0 radians horizontally. 80cm away from Pepper and in-line with both Pepper and the display center, tape was placed to mark the middle-most 80cm point. 40cm to the right and 40cm to the left of the middle-most point two more points were marked with tape, forming a line parallel to the display. The same parallel line, including 3 additional points was marked at 120cm away from Pepper. The laptop was connected to the display to project the points on.

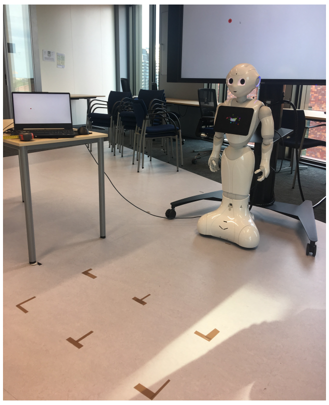

Figure 7: The experimental setup used to test the gaze tracking algorithm

3.1.4 Procedure

Each participant was requested to stand in the central 80cm mark, with their feet touching, standing in an up-right position, minimizing their movement as much as possible, with their head orientated vertically towards Pepper's head and horizontally towards the middle of the screen. Each participant was then instructed to look at the screen above Pepper's head and to focus their gaze on dots that would appear on screen. Each dot was on screen for 5 seconds before a new random dot, from the ones not shown yet, replaced the one on screen. After all 5 dots were projected, the test paused and the participant was instructed to move to the left-most (facing the screen) 80cm mark. The instructions given were the same with the exception of the horizontal head orientation that had to remain facing straight ahead and not to the middle of the screen. After 5 more dots were projected, the test was paused and the participant moved to the right-most 80cm mark. The same participant was then requested to repeat the same 3 horizontal positions with the same instructions but on the 120cm marks. Before the participant departed they were asked for their height and eye color.

3.2 Experimental Results

A summary of the mean euclidean distance error can be found for each participant as well as the accuracy comparing the algorithm estimates with the real expected values in Table 1. The mean errors range from 122.41px up to 170.58px and respectively the accuracy goes from 61% down to 49% for the 80cm points. A further decrease in accuracy can be observed for the 120cm points. In total, the average accuracy of the algorithm when detecting gaze target location was 52% with an error of 158.49px, which equates to an error of approximately 16.7 degrees in visual angle, for the 80cm points. The average accuracy for the 120cm points was 32% with an error of 257.65px, corresponding to an error of about 17.5 degrees in visual angle.

Observing the results in Table 2 in the Appendix we observe a difference in accuracy for the two different axes. While the accuracy for the x-coordinate averages 67% the accuracy for the y-coordinate averages 38%.

To better understand and observe the data, the estimation results were plotted against the expected coordinates (see Figure 8 for an example).

Participant	80cm		120cm	
	Error(px)	Accuracy	Error(px)	Accuracy
1	122.41	61%	159.95	55%
2	151.41	54%	225.54	43%
3	159.29	53%	266.36	33%
4	167.66	50%	298.63	23%
5	165.25	50%	268.66	30%
6	170.58	49%	295.40	20%
7	164.10	51%	288.00	23%
8	162.35	51%	270.35	28%
9	162.59	51%	258.83	30%
10	159.3	52%	247.76	33%
Mean	158.49	52%	257.65	32%

Table 1: The mean euclidean distance error from the expected coordinates and the corresponding accuracy for each participant in each of the two set distances

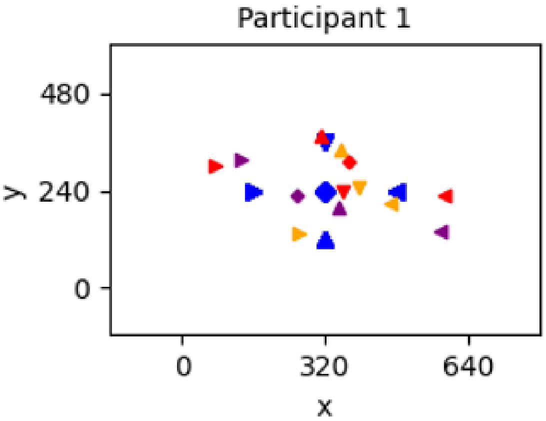

Figure 8: Example estimation results for participant 1 at 80cm in all three standing positions for all five projected points as described in 3.1.4. Blue markers are the expected coordinates while red, purple and yellow markers are the means of estimates in the first, second and third position respectively. Each marker style signifies each of the five expected positions.

Although the points seem to fall close to the expected coordinates, their seem to be variations between each of these plots with no particular pattern being detected for all participants. Graphs for all the participants can be found in the Appendix at Figure 11 and Figure 12.

In regards to personal information collected from each participant such as eye color and height their appears to be no correlation between any of the data and the estimated coordinates. It is important to note however that some of the experiments were under atypical conditions which can explain some effects seen in Figures 11-12. Specifically, participant 1 was tested at a different time of day than the rest, participant 4 did not strictly follow the experiment instructions and participant 6 was wearing both glasses and a face mask during the experiment. The effects for the estimates regarding participants 4 and 6 can be seen in their respective graphs. In the graph for participant 6 we observe an increased spread of estimates while in the graph of participant 4 we can see an abnormal spread pattern.

4 Discussion

4.1 Interpretation of Results

The gaze tracking algorithm using Peppers camera achieved an average accuracy of 52% corresponding to 158.49px of error with a variance of 5% between each participant, with an exception for participant 1 with an accuracy of 61%. As the distance from Pepper increased to 120cm the accuracy of detection decreased by 10-20% for each participant. Characteristics such as eye color and height did not affect the end result.

The algorithm's performance along the x and y axis showed significant differences in accuracy (see Table 2). The average x coordinate estimate accuracy equated to 67% at 80cm and 51% at 120cm, performing adequately at tracking the horizontal component of gaze direction, even reaching an accuracy of 78% with participant 1 at 80cm. However, the average y coordinate estimate accuracy equated to 38% at 80cm and 12% at 120cm. Performance along the vertical axis was suboptimal even resulting to 0% estimate accuracy with participant 4 at 120cm. These results could be explained by the nature of the experimental setup, where participants were asked to look at points on a display over Peppers head. Since the algorithm estimates coordinates of gaze direction relative to Pepper's visual frame, the resulting estimates where offset on the vertical axis. Although a large boundary in testing, this was the most viable option out of our available experimental setups. A digital screen could project precise points but with Pepper placed in front of the screen these points would have been obstructed from the participants view. In future experiments a micro-camera using the same settings as Pepper's camera is recommended instead, allowing unobstructed view while providing reliable estimates for both measures.

In summary, we cannot conclusively assess the performance of the system when tracking the vertical component of gaze direction. As a result, the overall performance as shown in Table 1 might be inaccurate in providing an indicator of performance. The estimates made on the horizontal axis may serve as better indicators of the algorithm's overall performance since the horizontal component can be measured reliably given the experimental setup. From that we can deduce that the algorithm can adequately estimate general gaze direction but is insufficient in high precision estimations.

4.2 Limitations

Perhaps the largest limitation to the project was the low camera resolution. Facial landmark detection as well as reliable image processing rely heavily on an adequate camera resolution. Using a better camera will improve coordinate estimates.

Although cannot be proven through data collected, lighting conditions could hypothetically play a vital role in accurate estimation. The testing for participant 1 occurred at a time where environmental luminescence was minimal and as observed in Tables 1-2 estimates were significantly more accurate.

The fixed distance required for the participant to interact with Pepper presented a significant obstacle in flexibility. As mentioned in 2.3 calibration is vital for valid calculations using real-world measures and the distance needs to be known before execution of the gaze estimator. However, it is unnatural that a participant will stay at a fixed distance during a natural interaction.

Lastly, the landmark predictor model used in 2.2.1 has only been trained on frontal faces and as such can reliably detect facial landmarks on heads at a maximum yaw of 0.70 and a maximum tilt of 0.50.

4.3 Future Improvements

The system developed consists of a multitude of components and as such improvements can be made in various departments. Starting from the facial detection, while Dlib provided an accurate, fast and easy to implement method, it struggled with smaller faces as well as faces that were not frontal. While not a limitation in the current project, an improved facial detection method could be developed that not only tackles these issues for use in a larger scope but that is also even faster and more accurate.

However, their would be no reason for improvement in facial detection before refining the landmark prediction algorithm. The Dlib shape predictor method is widely used for facial landmark creation in research, but can also still be improved on. The pre-trained model used was designed to predict 68 facial landmarks in a frontal face (see Figure 1), but an improved version can be trained. Within the duration of the project we trained our own 13 facial landmark shape predictor, which although faster in performance was not as accurate given the relatively minor amount of training data provided. Additionally, training time given our current computational capabilities would amount to days, with optimization taking more than a week at a time. If improvement is to be made in the landmark prediction method, a large pool of data for training will need to be provided which also include side-profiles for a larger degree of prediction and computational power allocated for training.

Perhaps the first area to be addressed for improvements in the future is the eye gaze direction estimation. As mentioned in 4.2 we suspect that environmental lighting may affect the accuracy of estimation. If that is the case, a model based approach, similar to the one used in 2.2.2, might prove to be increasingly optimal for determining eye gaze direction. The current implemented approach could also be improved. Opting for an adaptive thresholding algorithm for image processing (step in 2.2.3) instead of a fixed threshold should improve robustness in different lighting conditions.

Finally, since the project focused on detecting gaze for one person at a time, it is not optimized for groups of people. As a result, time complexity increases linearly for every individual in the visual frame. If this system is used in real-world situations in the future then either the facial detection system will have to be limited, detecting a single face or parallelism implemented, maintaining a steady frame per second cap.

4.4 Future Research

The possibilities of an eye tracking system in a mobile system and more specifically in a robot, present a plethora of research opportunities. Robots are useful tools in field of experimental psychology and cognitive science already [38] and this research could open doors to experimental research connecting gazing behaviour in natural interactions to human behaviour. Social cognition is intrinsically different when measured from within a social context as a participant rather than an observer [7].

Similarly, a gaze tracker can be used in real-time by the Pepper robot helping it to better understand human gaze cues and adapting to the level of interaction. Gaze duration and direction has been shown to signal desire, or indifference, for communication in humans [39] and detecting this could signal Pepper to take a different approach in real-time.

5 Conclusion

In this paper we have developed a gaze tracking system for gaze detection using the Pepper robot's built in camera without the need for specialized equipment. We have shown how head and eye angles can be detected using landmark prediction and combined to form a prediction of gaze direction. Using a moving average filter in conjunction with a Kalman filter, we were able to significantly reduce sensor noise creating a system capable of detecting general gaze direction in any commodity camera.

By implementing improvements in the future as mentioned in 4.3 we believe this system could become accurate and reliable enough to have a practical use in the field of human-robot interaction and experimental psychology.

6 Acknowledgements

I would like to sincerely thank Dr. Koen Hindriks for giving me the opportunity to work on a project that was not only challenging but also connected my passion for Artificial Intelligence to my Computer Science degree. I also would like to thank Dr. Artem Belopolsky for his invaluable insight and expertise provided throughout the duration of this projected. Lastly, I would like to thank the team at the Social AI lab for providing me with technical assistance and support when I needed it most.

References

[1] R. Cañigueral and A. F. d. C. Hamilton, "The role of eye gaze during natural social interactions in typical and autistic people," *Frontiers in Psychology*, vol. 10, p. 560, 2019.

[2] A. Georgescu, B. Kuzmanovic, L. Schilbach, R. Tepest, R. Kulbida, G. Bente, and K. Vogeley, "Neural correlates of "social gaze" processing in high-functioning autism under systematic variation of gaze duration," *NeuroImage: Clinical*, vol. 3, pp. 340–351, 2013.

[3] N. Emery, "The eyes have it: The neuroethology, function and evolution of social gaze," *Neuroscience Biobehavioral Reviews*, vol. 24, no. 6, pp. 581–604, 2000.

[4] M. Argyle, R. Ingham, F. Alkeman, and M. McCallin, "The different functions of gaze," vol. 7, pp. 19–32, 1973.

[5] M. Argyle, M. Cook, and D. Cramer, "Gaze and mutual gaze," *British Journal of Psychiatry*, vol. 165, no. 6, pp. 848–850, 1994.

[6] E. F. Risko, D. C. Richardson, and A. Kingstone, "Breaking the fourth wall of cognitive science: Real-world social attention and the dual function of gaze," *Current Directions in Psychological Science*, vol. 25, pp. 70–74, 2016.

[7] L. Schilbach, B. Timmermans, V. Reddy, A. Costall, G. Bente, T. Schlicht, and K. Vogeley, "Toward a second-person neuroscience," *Behavioral and Brain Sciences*, vol. 36, no. 4, pp. 393–414, Aug. 2013.

[8] I. Ryo, S. Yuta, I. N. Yukiko, and T. Nishida, "Combining multiple types of eye-gaze information to predict user's conversational engagement," *2011 International Conference on Intelligent User Interfaces (IUI2011), Workshop on Eye Gaze in Intelligent Human Machine Interaction*, 2011.

[9] A. Kendon, "Some functions of gaze-direction in social interaction," *Acta Psychologica*, vol. 26, pp. 22–63, 1967.

[10] R. Vertegaal, R. Slagter, G. van der Veer, and A. Nijholt, "Eye gaze patterns in conversations: There is more to conversational agents than meets the eyes," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Seattle, Washington, USA: Association for Computing Machinery, 2001, pp. 301–308.

[11] C. Willemse, S. Marchesi, and A. Wykowska, "Robot faces that follow gaze facilitate attentional engagement and increase their likeability," *Frontiers in Psychology*, vol. 9, p. 70, 2018.

[12] K. Koshikawa, M. Sasaki, T. Utsu, and K. Takemura, "Polarized near-infrared light emission for eye gaze estimation," New York, NY, USA: Association for Computing Machinery, 2020.

[13] E. Wood and A. Bulling, "Eyetab: Model-based gaze estimation on unmodified tablet computers," in *Proceedings of the Symposium on Eye Tracking Research and Applications*, New York, NY, USA: Association for Computing Machinery, 2014, pp. 207–210.

[14] A. Pérez, M. L. Córdoba, A. G. Dopico, R. Méndez, M. L. Muñoz, J. L. Pedraza, and F. M. Sánchez, "A precise eye-gaze detection and tracking system," in *WSCG*, 2003.

[15] Q. Li, R. Li, K. Ji, and W. Dai, "Kalman filter and its application," in *2015 8th International Conference on Intelligent Networks and Intelligent Systems (ICINIS)*, 2015, pp. 74–77.

[16] C. Morimoto, D. Koons, A. Amir, and M. Flickner, "Pupil detection and tracking using multiple light sources," *Image and Vision Computing*, vol. 18, no. 4, pp. 331–335, 2000, ISSN: 0262-8856. DOI: `https://doi.org/10.1016/S0262-8856(99)00053-0`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0262885699000530`.

[17] A. Yuille, D. Cohen, and P. Hallinan, "Feature extraction from faces using deformable templates," in *Proceedings CVPR '89: IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1989, pp. 104–109. DOI: `10.1109/CVPR.1989.37836`.

[18] L. Zhang, "Estimation of eye and mouth corner point positions in a knowledge-based coding system," in *Other Conferences*, 1996.

[19] I. Fasel, B. Fortenberry, and J. Movellan, "A generative framework for real time object detection and classification," *Computer Vision and Image Understanding*, vol. 98, no. 1, pp. 182–210, 2005.

[20] D. Cristinacce and T. Cootes, “Feature detection and tracking with constrained local models,” in *BMVC*, 2006.

[21] M. Reinders, R. Koch, and J. Gerbrands, “Locating facial features in image sequences using neural networks,” in *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, 1996, pp. 230–235. DOI: `10.1109/AFGR.1996.557269`.

[22] W. Sewell and O. V. Komogortsev, “Real-time eye gaze tracking with an unmodified commodity webcam employing a neural network,” *CHI ’10 Extended Abstracts on Human Factors in Computing Systems*, 2010.

[23] M. Asadifard and J. Shanbehzadeh, “Automatic adaptive center of pupil detection using face detection and cdf analysis,” 2010.

[24] M. Ciesla and P. Koziol, “Eye pupil location using webcam,” 2012.

[25] R. Valenti, N. Sebe, and T. Gevers, “Combining head pose and eye location information for gaze estimation,” *IEEE Transactions on Image Processing*, vol. 21, no. 2, pp. 802–815, 2012.

[26] F. Lu, T. Okabe, Y. Sugano, and Y. Sato, “A head pose-free approach for appearance-based gaze estimation,” in *BMVC*, 2011.

[27] A. Saran, S. Majumdar, E. S. Short, A. Thomaz, and S. Niekum, “Human gaze following for human-robot interaction,” *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8615–8621, 2018.

[28] A. T. Duchowski, *Eye tracking methodology: Theory and practice.* 2007, vol. 373.

[29] A. George, “Image based eye gaze tracking and its applications,” *CoRR*, vol. abs/1907.04325, 2019.

[30] M. Q. Khan and S. Lee, “Gaze and eye tracking: Techniques and applications in adas,” *Sensors (Basel, Switzerland)*, vol. 19, 2019.

[31] R. Valenti, N. Sebe, and T. Gevers, “Combining head pose and eye location information for gaze estimation,” *IEEE Transactions on Image Processing*, vol. 21, no. 2, pp. 802–815, 2012.

[32] B. Johnston and P. de Chazal, “A review of image-based automatic facial landmark identification techniques,” *EURASIP Journal on Image and Video Processing*, vol. 2018, 2018.

[33] V. Kazemi and J. Sullivan, “One millisecond face alignment with an ensemble of regression trees,” in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1867–1874.

[34] L. Liu, Z. Ke, J. Huo, and J. Chen, “Head pose estimation through keypoints matching between reconstructed 3d face model and 2d image,” *Sensors*, vol. 21, no. 5, 2021.

[35] J. M. D. Barros, B. Mirbach, F. Garcia, K. Varanasi, and D. Stricker, “Real-time head pose estimation by tracking and detection of keypoints and facial landmarks,” in *VISIGRAPP*, 2018.

[36] F. C. Donders and D. Doijer, “The location of the pivot point of the eye. part ii.,” *Strabismus*, vol. 24, no. 4, pp. 184–188, 2016.

[37] N. M. Scoville, R. Y. Lu, and H. Jung, “Optical axes and angle kappa,” [Online]. Available: `https://eyewiki.aao.org/Optical_Axes_and_Angle_Kappa`.

[38] A. Wykowska, “Robots as mirrors of the human mind,” *Current Directions in Psychological Science*, vol. 30, no. 1, pp. 34–40, 2021.

[39] F. Ho S and K. A. T, “Speaking and listening with the eyes: Gaze signaling during dyadic interactions,” *PLoS One*,

7 Appendix

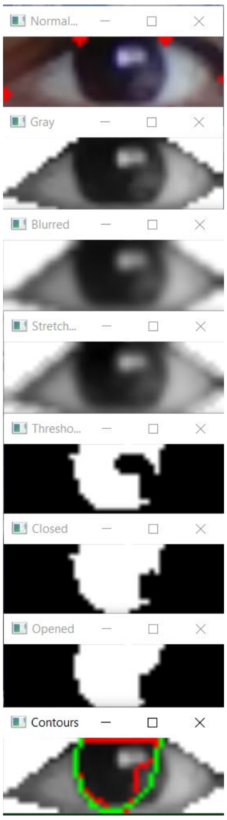

Figure 9: Example of each image processing step and operation performed on the eye region, namely from top to bottom, normal region of interest, gray-scaled eye region with background removed, Gaussian blur, Histogram equalization, Threshold operation, Morphological close, Morphological open, Contouring (Red line indicates initial contour and green the convex hull)

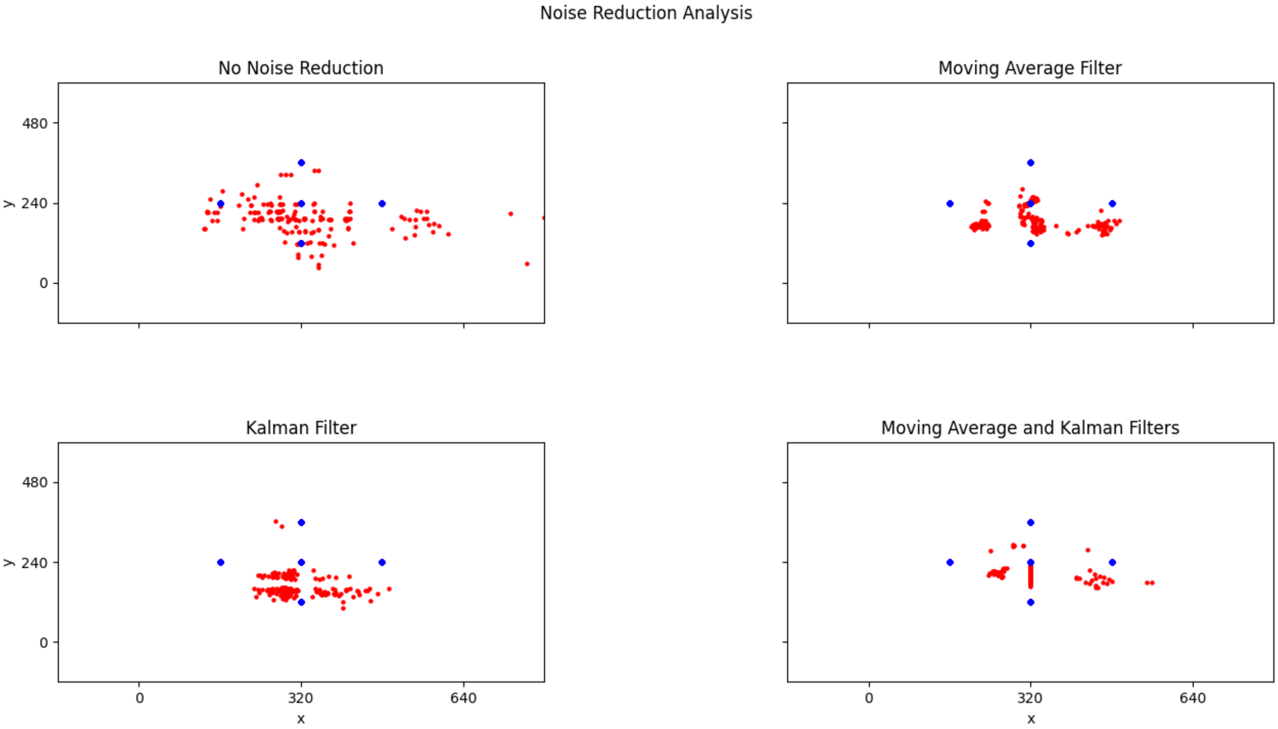

Figure 10: Comparing the effect of no filters for noise reduction, only moving average filter, only Kalman filter and the combination of moving average and Kalman. The blue dots represent the expected coordinates and the red dots all the estimations

Participant	80cm		120cm	
	Accuracy X-Coordinate	Accuracy Y-Coordinate	Accuracy X-Coordinate	Accuracy Y-Coordinate
1	78%	44%	74%	37%
2	69%	40%	63%	24%
3	66%	41%	65%	1%
4	65%	34%	47%	0%
5	65%	36%	51%	9%
6	62%	38%	38%	3%
7	65%	39%	38%	8%
8	67%	36%	42%	14%
9	68%	35%	45%	16%
10	69%	36%	48%	19%
Mean	67%	38%	51%	12%

Table 2: The accuracy of estimation on each axis for each participant in the two distances

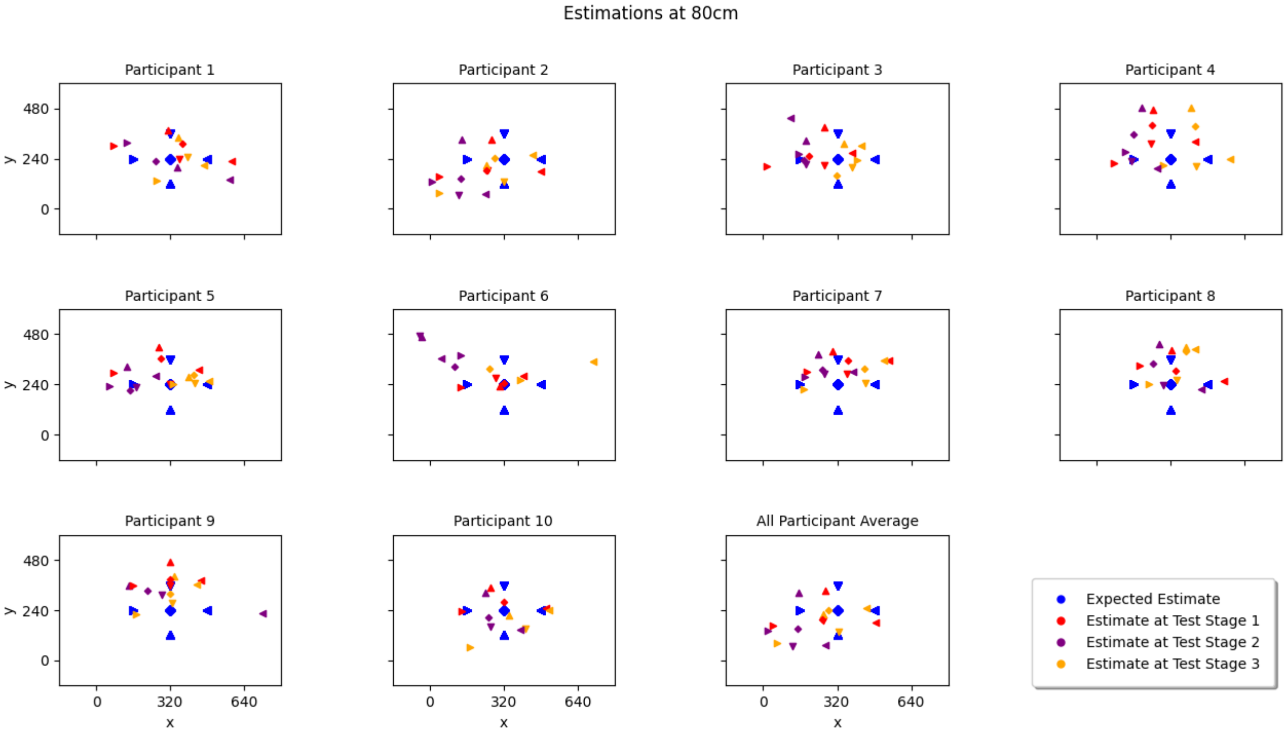

Figure 11: Estimation results for each participant at 80cm in all three standing positions for all five projected points as described in 3.1.4. Each marker style signifies each of the five expected positions.

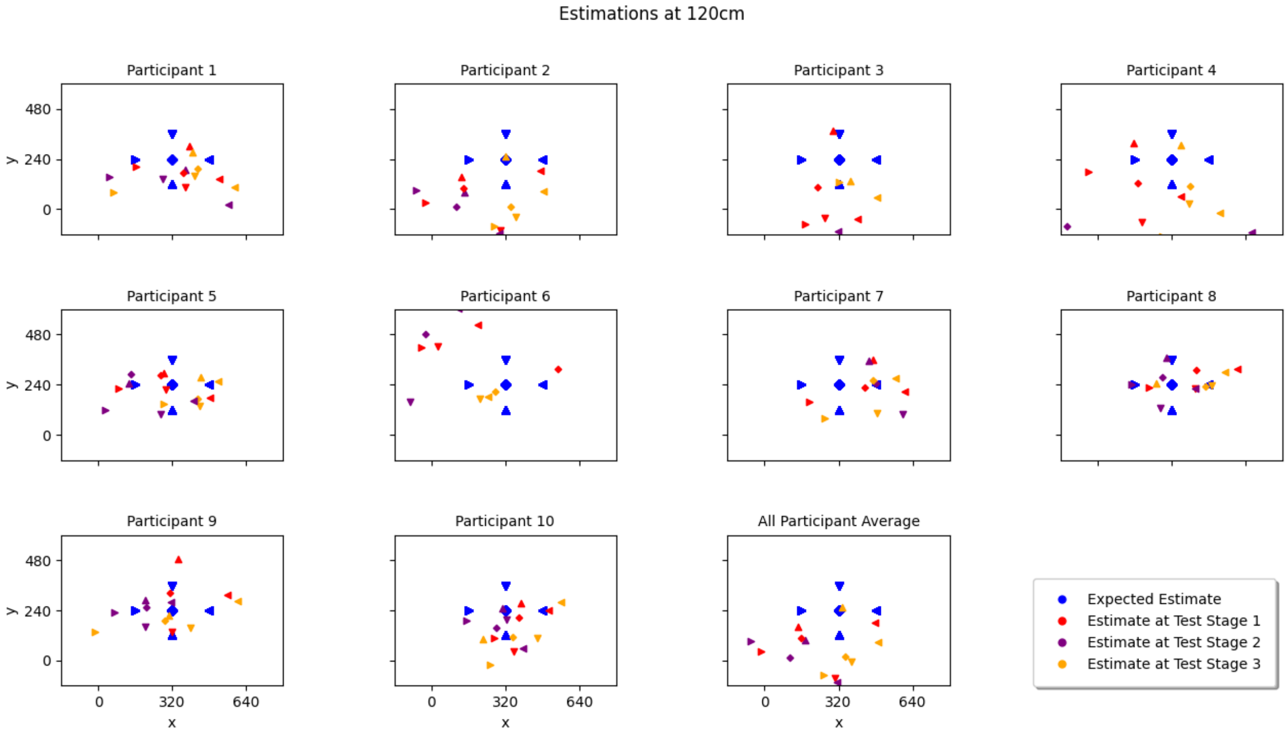

Figure 12: Estimation results for each participant at 120cm in all three standing positions for all five projected points as described in 3.1.4. Each marker style signifies each of the five expected positions.