2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)
November 3 - 5, 2015, Seoul, Korea

Yet Another Gaze Detector: An Embodied Calibration Free System for the iCub Robot

Lars Schillingmann[1] and Yukie Nagai[1]

Abstract— **The recognition of gaze as for example mutual gaze plays an important role in social interaction. Previous research shows that already infants are capable of detecting mutual gaze. Such abilities are relevant for robots to learn from interaction, for example detecting when the robot is being addressed. Although various gaze tracking methods have been proposed, few seem to be openly available for robotic platforms such as iCub. In this paper we will describe a gaze tracking system for humanoid robots that is completely based on freely available libraries and data sets. Our system is able to estimate horizontal and vertical gaze directions using low resolution VGA images from robot embodied vision at 30 frames per second. For this purpose we developed a pupil detection algorithm combining existing approaches to increase noise robustness. Our method combines positions of the face and eye features as well as context features such as eyelid correlates and thus does not rely on fixed head orientations. An evaluation on the iCub robot shows that our method is able to estimate mutual gaze with 96% accuracy at 8° tolerance and one meter distance to the robot. The results further support that mutual gaze detection yields higher accuracy in an embodied setup compared to other configurations.**

I. Introduction

Current humanoid robots are already capable of exhibiting a wide range of social signals. Interacting with humans requires them also to interpret social cues from speech, facial expressions, gesture, and gaze. The latter is frequently proposed as fundamental to natural human communication and learning. From birth, infants are sensitive to eye-contact [1], [2]. In western cultures mutual gaze plays an important role in children's social-cognitive development, though not exclusively [3]. Since humans are able to interpret gaze cues so early in life, it must be possible to extract at least basic information without using special sensors. Indeed, researchers have proposed various methods that are able to interpret gaze cues [4].

In contrast to the importance of gaze outlined for human communication embodied gaze detection has lower priority for robotic scenarios. When experiments require gaze detection, specialized hardware can be used. In other cases it is argued that gaze correlates with head pose and it is thus sufficient to approximate gaze from head pose estimations. However, [5] showed that head pose estimation alone is not reliable enough by comparing head pose estimated using the Microsoft Kinect 3D sensor with manually coded gaze. Methods such as eye-trackers require the user to wear special hardware and to undergo a calibration procedure which results in a less natural interaction scenario. Thus, eye-trackers are not generally used in human-robot interaction scenarios. An embodied approach to gaze detection has the advantage that no hardware is interfering with the interaction. Furthermore, the relation of the sensor to the robot does not have to be calibrated because making eye contact with the robot corresponds to a gaze direction of zero degree from the robot's eyes viewpoint. Surprisingly, only few calibration free gaze detection systems that only rely on camera images are openly available. Interpreting eye-gaze reliably using standard robotic vision systems seems difficult. One reason might be that even recent robotic systems still have cameras with relatively low VGA sensor resolution. An openly available calibration free gaze detector could lower the effort of integrating real gaze estimation into human robot interaction scenarios.

In this paper we will describe our calibration free gaze detector that has been developed to detect gaze relative to the observing camera. Our aim was to develop an open system. Thus, we took care that the individual components used for the development are open source and freely available. Additionally, we also used openly available training data to train our models. Neither running nor training our system requires commercial software.

II. Related Work

The body of literature on human gaze estimation techniques is large. Our main interest is a system that can be used with robot embodied vision. It should be possible to estimate gaze without prior calibration. These requirements do not allow for using head mounted eye trackers or other specialized sensors. Therefore, we focus on reviewing vision based gaze estimation systems. Since humanoid robots frequently use wide angle cameras, we cannot assume a high resolution image focusing on only one face. In [4] a review of eye and gaze detection is provided. The authors include eye detection algorithms in their analysis, because it is typically required to locate the eye region before further processing steps. Shape based models are commonly used due to the eyes distinctive features. Another approach is to determine the eye region based on non-rigid model which is aligned to facial features [6]. For our use case including the facial context is sufficient, because we do not assume close-up vision of only the eye region. Furthermore, an open source implementation is available [7].

A. Gaze Features

Estimating human gaze-direction typically requires two components: The head-orientation and the eye-gaze direction.

[1] Both authors are with the Graduate School of Engineering, Osaka University, 2-1 Yamadaoka, Suita, Osaka, 565-0871 Japan {lars, yukie}@ams.eng.osaka-u.ac.jp

978-1-4799-6885-5/15/$31.00 ©2015 IEEE

The head orientation is also encoded in the appearance of the eye region. Appearance based methods exploit this circumstance. In [8] for example mutual gaze against non-mutual gaze is classified based on eye region patches. Handling lower resolutions might require more contextual information. Different methods of combining head pose and gaze features are summarized in [4]. On the one hand, eye gaze and head pose can be estimated completely separately and combined model-based. On the other hand, pure feature based integration initially combines gaze and head related features before making any attempt to estimate the gaze-direction. In this case the head pose does not need to be estimated explicitly. Hybrid methods perform the combination on an intermediate level. For example, [9] combine head pose and eye gaze by adapting the eye-region related features based on head pose tracking results. However, the head tracker's yaw and pitch parameters are initialized assuming a frontal face view while the roll can be estimated based on eye locations. Since the primary use of our method is mutual gaze detection for human robot interaction, an initialization phase would make the method less flexible and less suitable for natural interaction scenarios. Model-based approaches might require additional scenario dependent parameters, which could provide higher accuracy but might restrict possible scenarios. Additionally, noisy estimations from both head pose and eye-gaze estimation might result in a higher joint error rate. To provide noise robust estimations, our method combines features before estimating the gaze-direction. Thus, we aim to not explicitly estimate head pose, unlike the majority of other systems [4].

B. Head Pose Features

Although, explicit head pose estimation is not the aim of the present work, it is still relevant to include features that correlate to head pose. These features provide contextual information to more estimate gaze direction from different viewing angles more accurately. The review of [10] on head pose estimation methods shows that one common approach is to utilize parameters of a non-rigid face model as features. Subsequently, the parameters are mapped to head pose angles for example by a regression model. Face alignment features inherently allow for localizing the eye region. Thus, head pose features can be extracted with a very low additional computational cost with this method.

C. Pupil Features

Determining the position of the pupil in human eyes is one of the main goals of eye tracking systems. Various methods typically operate under the assumption that a high resolution recording of the eye region is available. However, considering the problem of eye localization more generic methods can be found. Typically, the specific shape of the eye region and the pupil is exploited. The standard evaluation criterion for eye localization rewards accurate localization of pupil centers. Thus, successful methods can be also applied to localize pupils if the eye region is already known. A subset of methods uses geometric properties of the pupil such as circularity or rotation invariance to estimate the exact pupil center. [11] showed in an evaluation of eye localization systems that methods from this subset achieve nearly the same accuracy as more complex methods incorporating machine learning. In the evaluation, the method by [12], which uses no machine learning, is ranked 4th. The method by [13] is ranked 5th in this comparison. It is used in the system by [9], described in Section II-A.

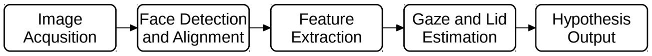


Fig. 1: Pipeline architecture of the gaze detection system

Both methods try to estimate the center of circular structures such as the pupil. A method presented in [13] exploits that circular structures typically consist of points having similar luminance at similar radii around a center. Timm and Barth describe a method [12] that follows a similar idea but it directly considers gradients pointing to a common center. Both methods generate a topographic map. The location of the maximum value in the map is the estimated center location of the circular structure.

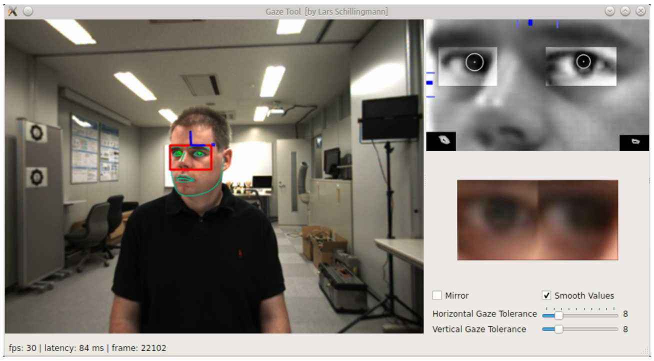
III. System Overview

Our system is based on horizontal and vertical gaze regression models. For mutual gaze detection, this seems to be more than necessary. However, this approach has several advantages. While a binary mutual gaze classifier such as proposed in [8] would only provide the mutual gaze state, the continuous output of a regression model allows for more flexible usage scenarios. Tolerance parameters can be adjusting without retraining the regression models. Gaze outside the mutual gaze area can be estimated which helps to realize gaze following scenarios. The gaze estimation system consists of several modules which are coupled based on a producer-consumer communication pattern (see Fig. 1). Each module consist of several worker threads. This ensures that on recent machines (2-4 cores) 30fps performance is achieved by processing several frames in parallel. A slight latency —typically around 80ms— is induced by this approach.

a) Image Acquisition: The image acquisition module is able to receive images from different sources. The YARP middleware is supported to integrate with the iCub robot [14]. Annotated data such as image lists can be processed to train the system. In addition, video and webcam input is supported for standalone operation if YARP is not required.

b) Face Detection and Alignment: The face detection module uses the dlib library's [7] face detection and face alignment algorithms to find faces in each frame provided by the image acquisition module. The face alignment module aligns a non-rigid facial feature point model to each detected face [6].

c) Feature Extraction: The feature extraction module calculates features for each detected face. The features include feature points provided by the previous module. Based on this initial feature set this module estimates pupil positions and extracts histogram of oriented gradients (HOG) descriptors (dlib's HOG implementation [7]) on the eye



□ Mirror ☑ Smooth Values

Horizontal Gaze Tolerance 8

Vertical Gaze Tolerance 8

fps: 30 | latency: 84 ms | frame: 22102

Fig. 2: Gaze tool displaying detected eye regions (top right), pupil positions, gazing angles (blue) and mutual gaze detection (red).

regions. Section IV describes the feature extraction methods in detail.

d) Gaze and Lid Training and Estimation: This module estimates horizontal gaze, vertical gaze, and lid closeness with epsilon insensitive support vector regression models. The models were trained on different feature subsets. Dlib's implementation of the insensitive support vector regression method is used [7].

e) Visualization and Hypothesis Output: Our system is able to display detected features and the resulting horizontal and vertical gaze estimations as well the eye closeness (see Fig. 2). Gaze and lid closeness estimations can be communicated via YARP or logged to a file. Furthermore, a vertical and horizontal tolerance setting is used to threshold gaze values to a binary mutual gaze status. Optional smoothing can be enabled that filters the output using a recursive least square filter.

IV. Feature Extraction

As reviewed in the related work section, the system requires features correlating with the head pose and the location of the pupil within the visible part of the sclera. To estimate the lid closeness, corresponding features from the eye region are required. The methods used in our system are described in the following.

A. Pupil Localization

Our pupil localization approach is based on [12]. The main idea of the method is that gradient vectors $\boldsymbol{g}_i$ in the eye region point to a common center, the pupil. An optimization was described by [15] who also provides a reference implementation of [12]: Only the gradients that point into darker areas such as the pupil (e.g. only positive gradients, depending on gradient implementation) are taken into account. Thus, the optimal center $\boldsymbol{c}^*$ given pixel positions $\boldsymbol{x}_i$, $i \in \{1, \ldots, N\}$ is according to [12]:

$$\boldsymbol{c}^* = \underset{\boldsymbol{c}}{\operatorname{argmax}} \left\{ \sum_{i=1}^{N} w_c \left(\boldsymbol{d}_i^T \boldsymbol{g_i} \; [\boldsymbol{g}_i > 0] \right)^2 \right\} \tag{1}$$

$$\boldsymbol{d}_i = \frac{\boldsymbol{x}_i - \boldsymbol{c}}{\|\boldsymbol{x}_i - \boldsymbol{c}\|_2}, \quad \forall i : \|\boldsymbol{g}_i\|_2 = 1 \tag{2}$$

In [12] it is further proposed to include prior knowledge by giving higher weight to dark pixels because they likely belong to the pupil center. This is weight is represented by w_c.

Finding $\boldsymbol{c}^*$ can be achieved by evaluating the sum term in equation 1 for all pixel positions. The position of resulting map's maximum is most likely the center of a circular structure such as the pupil. The idea of our implementation is to include more previous knowledge in the algorithm by restricting it to the exact eye regions only. This allows for making better use of the contrast between the pupil and sclera. Including our modification pupil positions are detected by performing the following steps:

1) The face alignment module provides a 6-point polygon of each eye region. A rectangular region of interest (ROI) based on the bounding box of each eye polygon is defined on a gray-scale copy of the input image.
2) Histogram equalization is performed for each eye ROI.
3) The ROI is scaled to a fixed width of 48 pixels.
4) For each eye ROI masks are generated based on the eye polygons.
5) A blurred copy of each eye region is used to provide the weights w_c.
6) It is assumed that each pupil lies within the masked ROI. Equation 1 is sampled to a map for all non-masked pixels. Subsequently, the map is normalized. The maximum of the map corresponds to $\boldsymbol{c}^*$.
7) At this point no structural information about bright and dark regions in the eye are considered. For example few dark pixels in a mainly bright region could be noise. Furthermore, a large dark area might indicate the presence of the pupil. The center of the pupil would be approximately at the center of that region. To provide additional weights that depend on the structure of dark and bright regions, a binarized copy of the eye ROI is created. The binarization threshold is determined adaptively by the OTSU method [16]. Subsequently, a distance transformation is applied to the binary image to create an additional weight map with high values assigned to the center of large dark regions such as the pupil. The resulting map is normalized and added to the map of the previous step.
8) Finally, the maximum of the combined maps is determined. To further reduce noise a truncated mean is calculated on values close to the maximum.

The pupil detection method is a heuristic. The evaluation will address the question in how far this method performs better than training the regression models directly on HOG features.

B. Positional Face Features

The face alignment module provides 68 feature points for each detected face [7], [17]. Including two detected pupil positions 70 feature points are available resulting in a 140 dimensional feature vector. Since we observed that the inner eye corners remained least affected by noise or alignment errors, the feature points are transformed into a relative coordinate system defined by the axis between the eye corners and the root of the nose. Furthermore, the distance between

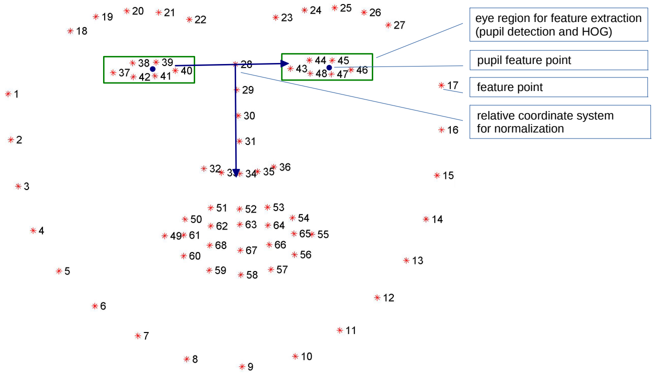

eye region for feature extraction (pupil detection and HOG)
pupil feature point
feature point
relative coordinate system for normalization

Fig. 3: 68 point feature points markup [17] with annotated features and regions used by our system.

the inner eye corners is used to normalize the feature points with regard to the size of each face (see Fig. 3).

C. Eye Region HOG Features

Detecting pupil positions might not be accurate enough to estimated vertical gaze. Due to the coupling between the upper eyelid and the vertical pupil position, lid features improve vertical gaze estimation. Furthermore, the pupil might not be visible at all if eyes are closed, so accurate gaze estimation is not possible. Lid closeness can be used to identify such cases. In our system this contextual information is represented using HOG features because they perform well in classifying eye closeness. In [18] several methods for eye closeness detection were evaluated. High classification accuracy was found for HOG and modified HOG features. Thus, our system calculates a size and orientation normalized feature vector for eye closeness based on the points provided by the face alignment module in the following way:

1) Based on the eye polygon provided by face alignment module an affine transformation is defined which projects the eye region on a 32x32 eye patch. The transformation is defined such that the angle between the left and the right corner is parallel to the x-axis of the patches coordinate system. Furthermore, the transformation scales the distance between the left and the right corner to 32 pixels.
2) The patches are horizontally concatenated.
3) HOG features are extracted with dlib's [7] implementation. A cell size of 8 pixels resulting in a 372 dimensional feature vector is used.

D. Combined HOG and Positional Features

Combining eye region HOG features and positional features creates a feature vector representing both the facial context and the eye region context. We assume this combined feature set will exhibit a better vertical gaze estimation performance. The combined feature vector can also be beneficial for eye closeness estimation, because it is able to disambiguate variations in camera perspective that affect the eye appearance. In this case pupil positions are excluded.

V. Model Estimation

A. Datasets

1) Columbia Gaze Data Set: Training a gaze regression model requires an annotated corpus. Our system is trained on the Columbia Gaze Data Set, which is openly available for non-commercial usage [8]. The database contains facial images with annotated head pose and gaze direction relative to the camera position. One feature of the corpus is that the same gaze directions ($\{\pm 15, \pm 10, \pm 5, 0\}$ degree horizontal and $\{\pm 10, 0\}$ degree vertically) are annotated for different head poses $\{\pm 30, \pm 15, 0\}$. This allows for training models that generalize over different head poses. The images were downscaled to 480x320 pixels to reflect the typically smaller sizes of faces in 640x480 robot camera images.

2) Closed Eyes in the Wild: The eye closeness models were trained on the Closed Eyes in the Wild (CEW) database [19]. The database contains images of faces with open and closed eyes. It is also openly available for non-commercial research purposes.

B. Training Support Vector Regression Models

The system uses dlib's epsilon insensitive support vector regression implementation [7]. Since the corpora only provide certain discrete angles, a linear kernel was chosen to avoid unpredictable deviations from the annotated values when the model has to interpolate and extrapolate gaze angles. The Columbia gaze corpus was used to train regression models for horizontal and vertical gaze. The Closed Eyes in Wild Dataset was used to train a regression model for opened and closed eyes. SVR parameters (c, epsilon, epsilon-insensitivity) were optimized using grid search. An optional PCA to reduce the dimensionality was included in the parameter set. For each model parametrization a 3-fold cross evaluation was carried out to identify the model with the lowest error. An overview of the optimized models is shown in Table I.

TABLE I: Model Optimization Results

Model	Feature Set	pca-ϵ	c	ϵ	ϵ-ins.	MSE	R^2
horiz. gaze	positional	-	1	0.1	2	33.25	0.68
horiz. gaze	HOG	-	0.1	0.5	2	36.14	0.66
horiz. gaze	HOG+pos.	-	1	0.1	1	23.40	0.78
vert. gaze	positional	-	0.1	0.5	2	41.69	0.43
vert. gaze	HOG	-	0.1	0.05	1	30.18	0.59
vert. gaze	HOG+pos.	0.8	0.1	0.5	2	27.24	0.61
lid	HOG	0.85	5	0.025	0.1	0.05	0.80
lid	HOG+pos.	0.9	5	0.05	0.1	0.048	0.80

VI. Evaluation

A. iCub Gaze Data Set

To evaluate our system we recorded a gaze data set with participants facing the iCub robot. The scenario is depicted in Fig. 4. The robot and 8 additional gaze targets were presented to 13 participants of 5 different nationalities and genders (unbalanced). Participants were seated at 90-110cm distance to the iCub's eyes. The targets were distributed on a wall with 50cm distance behind the robots eyes. Each target's gazing angles are plotted as vertical lines in Fig. 8 and Fig. 9. Participants were asked to look at each target

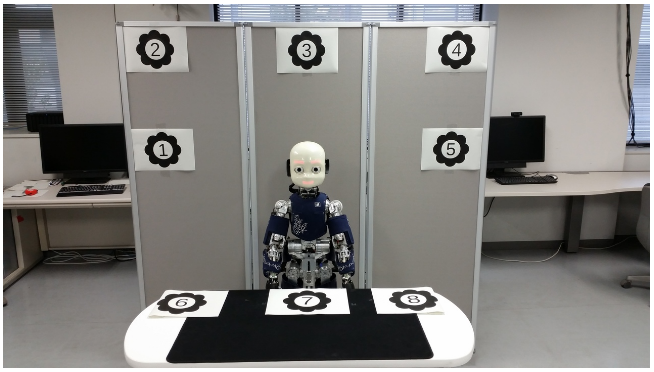


Fig. 4: Evaluation setting: numbers indicate gaze targets

constantly until they are instructed to look at the next target. From the participants 9 adjusted their head orientation during gaze shifts, while 4 decided not to do so. In a second trial, participants were instructed to look at the target while keeping the head orientation fixed. The interval participants were looking at each target was logged during data recording.

B. Results

Best performing models were evaluated on the iCub gaze dataset, which was not used during training or model optimization. The focus of our evaluation is the accuracy in detecting mutual gaze towards the robot. First, we evaluated the accuracy in detecting mutual gaze when participants were looking at the robot. From the iCub Gaze Dataset 30 frames corresponding to each target were selected. The frames were taken around the center of each interval of target directed gazing. Fig. 5 shows both the percentage of true positives and false positives over the mutual gaze detection tolerance which is shown on the horizontal axis. One can observe that the positional feature set performs generally better for horizontal gaze detection despite the lower error that was measured during model optimization. This indicates that the positional feature set leads to a better generalizing model. For vertical gaze the contextual information within the eye such as eyelids plays an important role. Thus, the combined positional and HOG feature set yields the best result. Combining the best models 96.6% of gaze directed at the robots face within a tolerance window of ±8 degree can be correctly identified while 4.5% are incorrectly classified (see Fig. 7).

To confirm that this result is not strongly depending on whether participants move their head when shifting gaze, we repeated the evaluation with data where participants' head position was not changing. The model performance in Fig. 6 clearly shows that the HOG and Positional feature set exhibits better results than positional features for detecting vertical gaze. Again, given a mutual gaze detection tolerance of ±8 degree, 92% of the gaze directed at the robots face can be correctly identified while 6.1% are incorrectly classified as mutual gaze (see Fig. 7). At this distance of the participants to the iCub robot, their corner to corner eye size is at about 16 pixels due to the robots relatively wide field of view (87.5°). Given the low resolution and that not all gaze angles included the evaluation were also present in the training data shows that our system generalizes well and yields accurate mutual gaze detection results.

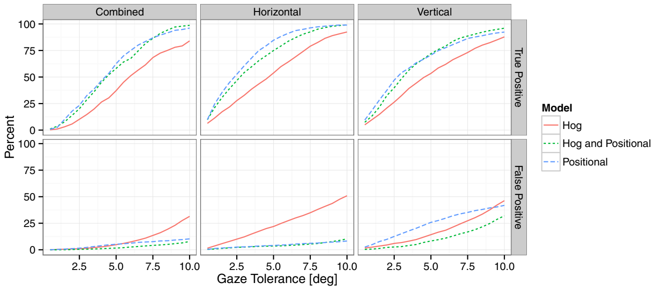


Fig. 5: Modelperformance for uninstructed participants with mixed gaze and head motion adjustments.

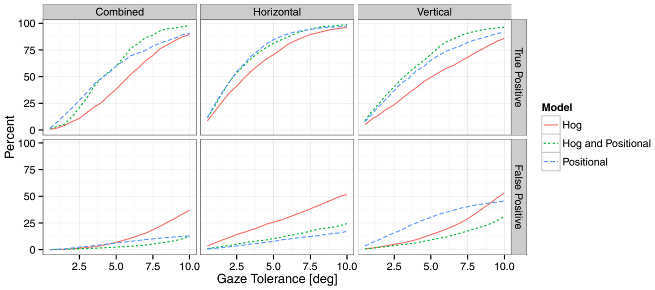


Fig. 6: Modelperformance for participants with fixed head pose.

Further analysis of the distribution of detected gaze directions for each gaze target reveals that at larger angles the distribution is biased indicating that the error increases with the gazing angle (see Fig. 8 and 9). The generalization capabilities of the models are limited but sufficient to extrapolate gaze directions relative to the camera position. Furthermore, the continuous output range supports defining custom gaze detection tolerances which would be more difficult using a binary classifier. Comparing the average standard deviation of all gaze targets (horizontal=5.48, vertical=4.18) with the standard deviation of gaze detections to the robot's face (horizontal=3.43, vertical=3.72) shows that detecting mutual gaze using an embodied setup yields more accurate results. Using an external camera with the same specifications would lead to a wider distribution, because users would not look into the camera that has to detect mutual gaze.

Eye closeness detection has not been evaluated on the iCub gaze corpus. However, preliminary tests have shown its capability to detect eye closeness reliably such that it also can be used for other tasks such as for blink detection.

VII. Conclusion

We have presented a gaze detection system that is able to estimate human eye gaze even if the face is not frontal to the camera. The system's feature sets includes head pose features, eye region HOG features and noise robust pupil detection. Feature sets are mapped to horizontal and vertical gaze angles by epsilon insensitive support vector regression models. Additionally, an eye closeness detector allows for detecting eye blink and rejecting closed eyes. The

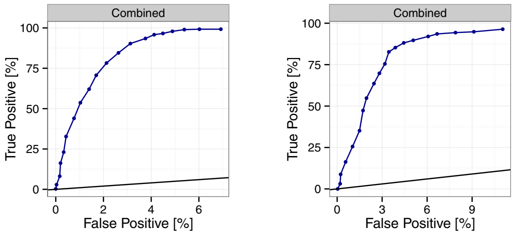


Fig. 7: ROC curve of the combined best performing models for participants with mixed gaze and head motion adjustments (left) and participants with fixed head pose (right).

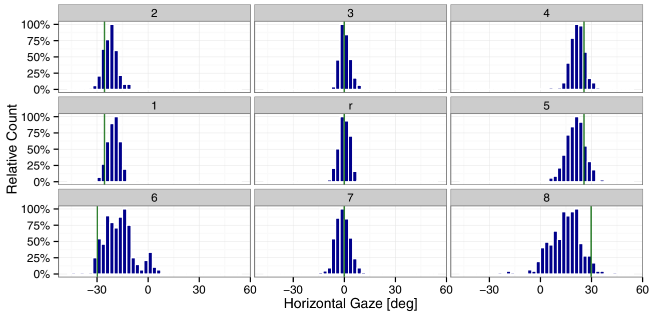


Fig. 8: Per gaze target histograms of detected gaze angles for the best horizontal gaze model using the positional feature set. Headers correspond to the gaze targets in Fig. 4.

evaluation on the humanoid robot iCub showed that our system is able to estimate gaze and detect mutual gaze even in low resolution images of 640x480 pixels. Furthermore, the system generalizes well and outputs sensible results for gazing angles outside the training data. We could show that using iCub's embodied cameras results in a lower variance of robot directed gaze detections compared to other angles, indicating the advantage of the embodied approach. The system supports multiple users interacting with the iCub at a typical interaction distance of 1m. A frame rate of 30 fps with a latency of about 80ms makes the system suitable for real time processing. First gaze estimates are instantly available without prior calibration or assumptions about initial head pose or gaze direction. Our system consistently uses openly available libraries and data sets. Consequently, it is available from `github.com/lschilli/gazetool` to simplify developing human robot interaction scenarios.

Acknowledgments

This work has been supported in part by MEXT KAKENHI "Constructive Developmental Science" (24119003).

References

[1] T. Farroni, G. Csibra, F. Simion, and M. H. Johnson, "Eye contact detection in humans from birth." *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, no. 14, pp. 9602–5, July 2002.

[2] E. Parise and G. Csibra, "Neural responses to multimodal ostensive signals in 5-month-old infants." *PloS one*, vol. 8, no. 8, p. e72360, Jan. 2013.

[3] N. Akhtar and M. A. Gernsbacher, "On Privileging the Role of Gaze in Infant Social Cognition." *Child development perspectives*, vol. 2, no. 2, pp. 59–65, Aug. 2008.

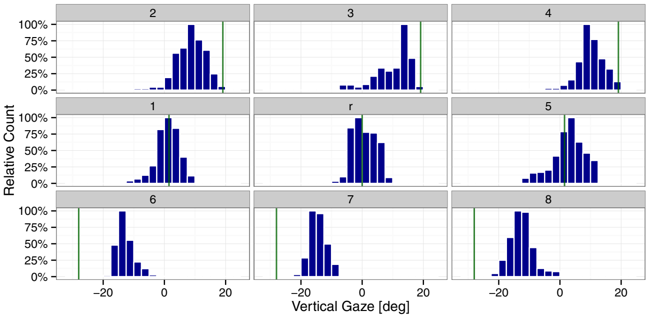


Fig. 9: Per gaze target histograms of detected gaze angles for the best vertical gaze model using the HOG and positional feature set. Headers correspond to the gaze targets in Fig. 4.

[4] D. W. Hansen and Q. Ji, "In the eye of the beholder: a survey of models for eyes and gaze." *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 3, pp. 478–500, Mar. 2010.

[5] J. Kennedy, P. Baxter, and T. Belpaeme, "Head Pose Estimation is an Inadequate Replacement for Eye Gaze in Child-Robot Interaction," in *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction Extended Abstracts - HRI'15 Extended Abstracts*. New York, New York, USA: ACM Press, Mar. 2015, pp. 35–36.

[6] V. Kazemi and J. Sullivan, "One Millisecond Face Alignment with an Ensemble of Regression Trees," in *In Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2014)*, 2014.

[7] D. E. King, "Dlib-ml: A Machine Learning Toolkit," *The Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, Dec. 2009.

[8] B. A. Smith, Q. Yin, S. K. Feiner, and S. K. Nayar, "Gaze locking: passive eye contact detection for human-object interaction," in *Proceedings of the 26th annual ACM symposium on User interface software and technology - UIST '13*. New York, New York, USA: ACM Press, Oct. 2013, pp. 271–280.

[9] 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.

[10] E. Murphy-Chutorian and M. M. Trivedi, "Head pose estimation in computer vision: a survey." *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 4, pp. 607–26, Apr. 2009.

[11] K. Jariwala and U. Dalal, "Performance Analysis of Eye localization Methods for Real Time Vision Interface using Low Grade Video Camera," *International Journal of Computer Applications*, vol. 114, no. 2, pp. 33–40, 2015.

[12] F. Timm and E. Barth, "Accurate eye centre localisation by means of gradients," in *Proceedings of the International Conference on Computer Vision Theory and Applications*, vol. 1. Algarve, Portugal: SciTePress - Science and and Technology Publications, 2011, pp. 125–130.

[13] R. Valenti and T. Gevers, "Accurate eye center location and tracking using isophote curvature," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, June 2008, pp. 1–8.

[14] P. Fitzpatrick, G. Metta, and L. Natale, "Towards long-lived robot genes," *Robotics and Autonomous Systems*, vol. 56, no. 1, pp. 29–45, Jan. 2008.

[15] T. Hume, "eyeLike," 2012. [Online]. Available: http://thume.ca/projects/2012/11/04/simple-accurate-eye-center-tracking-in-opencv/

[16] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[17] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "300 Faces in-the-Wild Challenge: The First Facial Landmark Localization Challenge," in *2013 IEEE International Conference on Computer Vision Workshops*. IEEE, Dec. 2013, pp. 397–403.

[18] F. Song, X. Tan, X. Liu, and S. Chen, "Eyes closeness detection from still images with multi-scale histograms of principal oriented gradients," *Pattern Recognition*, vol. 47, no. 9, pp. 2825–2838, Sept. 2014.

[19] Z. Zhu, Q. Ji, and K. P. Bennett, "Nonlinear eye gaze mapping function estimation via support vector regression," in *Proceedings - International Conference on Pattern Recognition*, vol. 1, no. c, 2006, pp. 1132–1135.