Characterizing Joint Attention Behavior during Real World Interactions using Automated Object and Gaze Detection

Pranav Venuprasad
University of California San Diego
La Jolla, California
pvenupra@ucsd.edu

Tushal Dobhal
University of California San Diego
La Jolla, California
tdobhal@ucsd.edu

Anurag Paul
University of California San Diego
La Jolla, California
a8paul@ucsd.edu

Tu N.M. Nguyen
Minerva Schools at the Keck
Graduate Institute
tu@minerva.kgi.edu

Andrew Gilman
Massey University
Auckland, NZ
A.Gilman@massey.ac.nz

Pamela Cosman
University of California San Diego
La Jolla, California
pcosman@ucsd.edu

Leanne Chukoskie
University of California San Diego
La Jolla, California
lchukoskie@ucsd.edu

ABSTRACT

Joint attention is an essential part of the development process of children, and impairments in joint attention are considered as one of the first symptoms of autism. In this paper, we develop a novel technique to characterize joint attention in real time, by studying the interaction of two human subjects with each other and with multiple objects present in the room. This is done by capturing the subjects' gaze through eye-tracking glasses and detecting their looks on predefined indicator objects. A deep learning network is trained and deployed to detect the objects in the field of vision of the subject by processing the video feed of the world view camera mounted on the eye-tracking glasses. The looking patterns of the subjects are determined and a real-time audio response is provided when a joint attention is detected, i.e., when their looks coincide. Our findings suggest a trade-off between the accuracy measure (Look Positive Predictive Value) and the latency of joint look detection for various system parameters. For more accurate joint look detection, the system has higher latency, and for faster detection, the detection accuracy goes down.

CCS CONCEPTS

• **Human-centered computing → Collaborative interaction**; *Laboratory experiments*; Sound-based input / output; Auditory feedback; • **Applied computing** → Consumer health.

KEYWORDS

Joint Attention, Object Detection, Deep Learning, Computer Vision, Autism, Eye-Tracking, Gaze Behavior

ACM Reference Format:
Pranav Venuprasad, Tushal Dobhal, Anurag Paul, Tu N.M. Nguyen, Andrew Gilman, Pamela Cosman, and Leanne Chukoskie. 2019. Characterizing Joint Attention Behavior during Real World Interactions using Automated Object and Gaze Detection. In *2019 Symposium on Eye Tracking Research and Applications (ETRA '19), June 25–28, 2019, Denver, CO, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3314111.3319843

1 INTRODUCTION

Joint attention is a crucial skill that most children learn in their first year of life. By sharing gaze on an object with a caregiver, and looking between the object and the caregiver, children learn names of items, rules about object use, and motor behaviors, all of which are important for scaffolding later developmental skills. Unfortunately, joint attention skill does not come easily to some children, including children with autism spectrum disorders (ASD) [Kasari et al. 1990] who also show deficits in orienting attention more generally [Miller et al. 2014; Townsend et al. 1996; Zwaigenbaum et al. 2005]. Clinical psychologists have created intensive behavioral interventions for improving joint attention, involving many hours per week of therapy from a professional with training in this specific behavioral intervention (see for example the JASPER [Goods et al. 2013] intervention from Kasari and colleagues, and a review of a range of joint attention interventions from [Paparella and Freeman 2015]). These individuals are rare and their limited time also limits the availability of these important interventions. To continue the valuable intervention in the absence of professionals, parents must be independently trained to deliver the therapy which can be difficult especially in children who frequently show a greater interest in orienting to objects than to people.

Nature provides a wide array of cues and experiences that are enough to help most children develop along a relatively predictable, albeit complex path. For children with neurological differences,

ETRA '19, June 25–28, 2019, Denver, CO, USA

ACM ISBN 978-1-4503-6709-7/19/06…$15.00
https://doi.org/10.1145/3314111.3319843

however, these cues may not be adequate and their experiences may be atypical enough that the child does not experience the learning benefits enjoyed by their typical peers. By harnessing the power of technology, we can provide a different kind of feedback for a child to learn where and when to look. Here, we seek to deliver a tool that is relatively low cost and can be used by parents and caregivers in their homes as often as is convenient. This approach removes the barriers of time and place that are often created by the standard nature of clinical care. Moreover, a child is more likely to engage in an intervention that is fun, rewarding and that creates a sense of mastery. Incorporating these game-based elements into the intervention tool will help to deliver a therapeutic device that young children will actually want to use.

The recent development of affordable mobile eye trackers has facilitated the examination of real-world gaze behavior involving dynamic social interactions. In this paper, we use these glasses to develop an easy-to-use system that can detect joint attention between two people in real-time.

2 RELATED WORK

Several others have developed methods that have expanded our understanding of dynamic coordinated gaze interaction of young children and caregivers [Franchak et al. 2014; Slone et al. 2018; Yu and Smith 2016]. These systems and methods developed for research with young children are providing new insights into the real-world interactions of children and their caregivers. In most of these methods, the looking patterns of the subjects are obtained offline by manually analyzing their gaze stream with the help of a software, which makes the process tedious and inefficient. These methods of analysis can benefit from automating the system, as we explain in this paper. Systems and methods for near real-time detection of a range of joint attention behaviors have been advanced by researchers seeking ways to mimic human-like social gaze behavior in robotic agents [Rabbitt et al. 2015]. Kajapoulos *et al.* in [2015] reported using a social robot to train children to make an orienting head movement in response to the robot's head movement that initiates gaze toward an object. However, this training effort did not involve the eye component of gaze behavior.

3 METHOD

We have developed a real-time joint attention detection system that efficiently coordinates data from two eye-tracking glasses and provides a feedback when a joint look is detected between the wearers. Two subjects (potentially the child and the therapist) wearing eye-tracking glasses interact with each other and with multiple predefined indicator objects present in the room. The data from the world cameras on the glasses is fed to our system that has been trained to detect the objects present in the wearers' field of vision. The looking pattern for each subject across time is obtained using the detection bounding boxes around the objects and the gaze information. An audio response is provided in real-time when the looks of both subjects fall on the same object object simultaneously.

We first provide an overview of our implementation and then in later sections describe the individual processes in detail. The necessary code for implementing the system can be found at *https://github.com/tushardobhal/pupil*.

3.1 System Overview

3.1.1 Hardware. We gave higher priority to mobility and portability than sheer computing power while selecting our hardware and therefore used MSI GT62VR-7RE notebook (Intel Core i7-7700k, 16GB RAM and 8GB Nvidia GTX 1070 graphics card) for running our custom software, while Pupil Labs [Kassner et al. 2014] glasses were used for eye tracking. The glasses consist of a world-view RGB camera capable of producing 720 x 1280 frames at 30 fps located above the eyebrows on the glass frame, and two infrared cameras capable of 120 fps pupil tracking positioned below each eye on the glasses. This binocular eye setup outputs the gaze location data and a confidence score which is a measure of the accuracy of the gaze location in the world-view frame.

3.1.2 Software. Figure 1 details the individual look detection process for each subject. The data produced by the glasses are consumed by the three asynchronous message listeners running on our system. These consumers receive the gaze positions and confidence scores for each eye from the eye cameras and the world-view frames from the world-view camera. The gaze position data from each eye is smoothed using a Kalman filter, and their confidence scores are used to obtain a single rectified gaze stream. An object detector, as defined in Section 3.2, inputs the world-view frames and outputs the bounding boxes describing the location of the objects present in the frames. These bounding boxes are then adjusted using the obtained gaze confidence score to account for inaccuracies in the position of the gaze dot. Because the frame rate of the eye cameras is four times that of the world-view camera, the data is processed at this point only when a new world-view frame is received. This allowed us to reduce the number of computations for real-time usage without impacting the overall detection accuracy. For each bounding box obtained, we check if the gaze dot is inside, to obtain a Boolean decision. The series of decision outputs is then filtered using a Runlength algorithm which defines and detects "looks" on each object as discussed in Section 3.4. This "look" data from both the glasses is then passed to the Joint-Attention detector which provides an audio feedback when a "joint look" is successfully detected.

Occasionally, the gaze location data stream was observed to provide measurements with less than acceptable confidence scores, and also, at times to contain jitters either due to eye movement or sensor errors. As gaze tracking forms an integral part of our system, we formulated techniques discussed in Section 3.3 to overcome the above issues.

3.2 Object Detection

To detect objects in the world-view of the subjects, we use a compact version of the YOLOv3 network [Redmon and Farhadi 2018] named YOLOv3-tiny. While there are other Convolutional Neural Network-based approaches for object detection which give better detection accuracy such as Faster R-CNN [Ren et al. 2015] and R-FCNNs [Dai et al. 2016], YOLOv3-tiny performs detection in real-time with reasonably good accuracy. The algorithm uses Tiny Darknet, a small

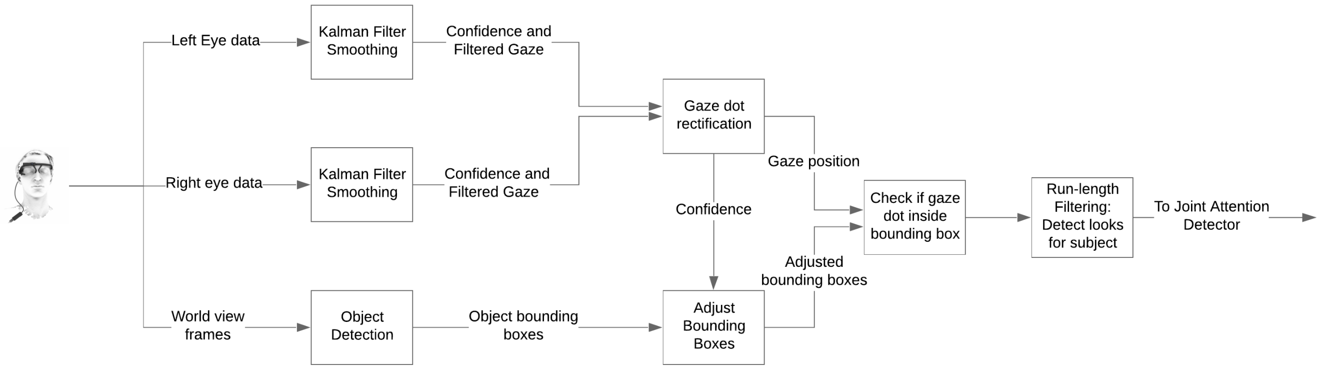

Figure 1: Block diagram of look detection for each subject

Darknet model [Redmon 2016] which has 23 layers as opposed to other conventional networks which have more than 100 layers.

Our object detector was trained on a custom dataset which included five objects chosen for the experiment: dice, key, taboo card, green spiky ball and map. Of these, the dice, key and map were placed on the wall approximately eight feet away from the subjects, while the ball and card were kept on the table in front. Training images were collected from the world-view camera with eight different subjects wearing the glasses. In total, there were approximately 100,000 training images. Table 1 shows the split of instances of the objects in the training images. A minimum enclosing rectangle was manually placed around each object, as shown in Figure 2, to serve as the ground truth (GT) during training with the help of Computer Vision Annotation Tool (CVAT) [Manovich et al. 2019]. We drew a bounding box around an object if and only if at least 50% of the object is present in the world frame according to visual inspection. The CVAT tool has an interpolation feature, which it uses to generate bounding boxes across adjacent frames, that significantly reduced the time required to ground truth the videos.

For training our object detector, we used pre-trained weights from YOLOv3-tiny model trained on the COCO dataset [Lin et al. 2014]. The COCO dataset is a large-scale object detection, segmentation, and captioning dataset with 80 classes corresponding to various common objects found in daily life. We trained our model for 150,000 iterations with a learning rate of 0.001 and momentum of 0.9. We evaluated the performance by computing the following metrics:

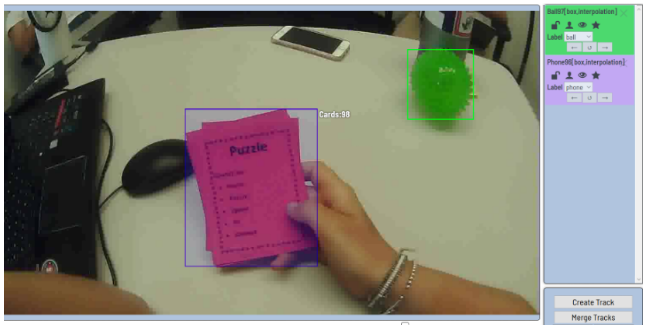

Figure 2: Minimum enclosing rectangles for annotation

(1) Intersection over Union (IoU): The IoU for an object in a frame is defined as the area of intersection divided by the area of union of the bounding boxes obtained from the network and manually for the object in the frame. For each object, we compute the mean IoU of every detected occurrence of the object in the test frames. The definition of IoU is illustrated in Figure 3.

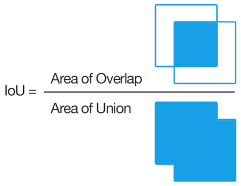

Figure 3: Illustration of IoU

(2) True Positive Rate (TPR) : TPR is defined as the fraction of the objects which were manually annotated in the test dataset which are detected by the network with a non-zero overlap between the detected and the ground truth bounding boxes.

(3) Precision : The precision for an object is defined as the percentage of correct detections of the object with respect to the total number of detections of the object in the test dataset.

$$Precision = \frac{No.\ of\ Correct\ detections\ of\ the\ object}{Total\ number\ of\ detections\ of\ the\ object}$$

3.3 Smoothing and Corrections

3.3.1 Kalman Filtering and Gaze Rectification. The Kalman Filter [Thrun et al. 2005] is an efficient recursive algorithm for estimating the state of a system. Here, we use Kalman filtering to correct for jitter in the gaze stream. The Kalman filter in our implementation is initialized with a 8-dimensional state vector with a two dimensional

Table 1: Number of instances of each object in the training set

	Ball	Dice	Key	Map	Cards
Number of Frames	51798	39835	42102	30195	30248
Approx. Time (min)	14	11	11	8	8

measurement update. Gaze dot coordinates from the Pupil Labs' glasses received at a rate of 120 fps serve as the measurement update. We use a variable velocity and acceleration model for our Kalman filter. In order to select the initial noise and delta time estimates, we conducted multiple experiments with a range of values for noise and delta time and selected the estimates which provided the best results empirically. In particular, we found that using noise estimate, η = 0.2 and delta time, dt=2.0 worked best under our experimental settings.

Along with Kalman Filtering, we also implemented an algorithm for removing the noise introduced due to blinking, as the glasses report a rapid change in the gaze location during blinking. With a proper threshold, noisy measurements due to blinking can be identified and removed from the data stream before being sent to the Kalman filter. For these noisy measurements, an estimated position of the gaze location predicted by the Kalman filter for that frame was used. Figures 4 and 5 show the effect of Kalman filtering and Blink removal in smoothing the initial gaze data stream.

We further utilized the gaze confidence score to rectify the gaze stream and ensure better tracking results. If both the pupil-tracking cameras detected the person's gaze with a confidence greater than a threshold of 0.8, their mean was computed, otherwise, the data of the one with a higher gaze confidence was used.

3.3.2 Bounding Box Correction. When the subject looks at the designated object, sometimes their gaze dot is just outside the bounding box for that object, possibly due to poor calibration, poor sensor data or gaze drift. To correct for this, bounding boxes around the various objects used in the experiment were expanded based on the depth of the object in the world view and based inversely on the confidence of the gaze dot location. Let w, h respectively be the width and height of the bounding box as provided by the object detection system, c ($c \in [0, 1]$) be the confidence provided by the system regarding gaze dot position, and $\alpha(x)$ be the correction based on depth x of the object, then the corrected width and height of the object i are given as:

$$\mathbf{w} = w + \{(1 - c) * \alpha(x_i)\} \tag{1}$$

$$\mathbf{h} = h + \{(1 - c) * \alpha(x_i)\} \tag{2}$$

For the objects in this paper, we consider only two depths: Approximately 8 feet for the objects which are on the wall (key, dice, map) and 1.5 feet for the objects which are on the table (ball, card). The corresponding values of $\alpha(x)$ are 30 pixels for x=8ft and 15 pixels for x=1.5ft. The intent of this adjustment is to expand the bounding box when the glasses provide low confidence on the gaze location, and the expansion is larger for objects that are farther away.

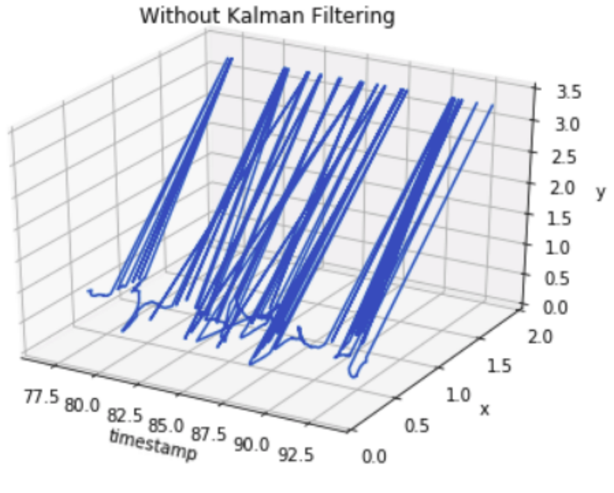

Figure 4: Initial unfiltered raw gaze data stream

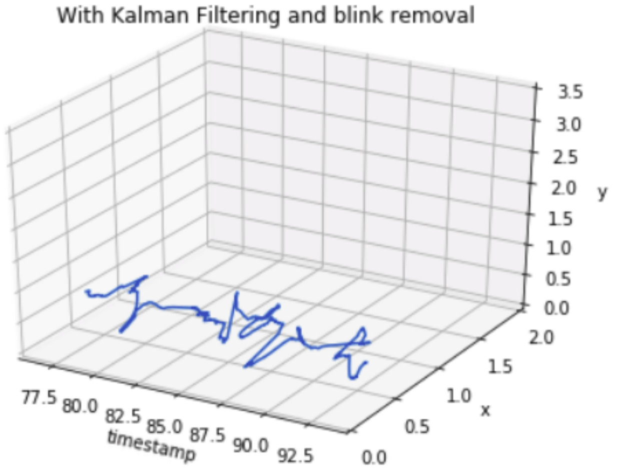

Figure 5: Impact of Kalman Filtering and Blink Removal on the initial gaze data stream

3.4 Determination of a "look"

Chukoskie et al. 2018 provide the definition of a look as an inspection typically involving an aggregated series of fixations and small reorienting saccades or microsaccades. A saccade is a reorienting movement which is interposed in steady intervals of fixations which characterize human gaze behavior. A typical look examining an object can last from 100-200 ms to a number of seconds. For joint attention, we are interested in the onset of a look and the earliest we can declare with reasonable certainty that both subjects are looking at the same object.

3.4.1 Runlength Filtering. The object detection system provides bounding boxes for the objects detected in a particular frame for the world view of both the subjects. For each subject, if the gaze dot is within the bounding box of an object, we call it a 'hit' for the object, else we call it a 'miss'. Runlength filtering with parameters **W** and **O** is applied to this data stream of hits and misses to determine a subject's look on a particular object. **W** specifies the window size or the number of frames that should be considered together and **O** specifies the minimum number of hits in that window to declare it to be a look. For instance, a runlength filter RL(4, 3) with parameters **W** = 4 and **O** = 3 would require at least 3 hits in any 4 consecutive frames for it to be declared as the beginning of look. At any time instant, if the look for both subjects fall on the same object, then a joint look is detected and a beep sound is provided to the experimenter as a feedback for detecting joint attention.

3.4.2 Latency. Going by the above definition of runlength filtering, we can see that a filter with RL(1,1) provides an earlier positive indication of a joint look than would RL(4,3). We define our first metric *'Latency'* to be the time it takes from the instruction to look at a particular object to the detection of a joint look at that object. Hence, we can say that RL(1,1) will have a lower latency than RL(4,3). Since the orienting behavior of the two subjects can be somewhat variable, we estimate that the *'Latency'* metric accounts for approximately 500ms of reaction time for the two subjects. Considering that saccadic reaction time of healthy young adults in a tightly controlled experimental setting with a visually-cued sudden-onset target is approximately 250ms (estimate from [Kenward et al. 2017]), 500ms is a conservative estimate for an orienting movement that may include a small head movement and/or a corrective saccade.

3.4.3 Estimated Detection Time. We define the metric *'Estimated Detection Time'*, as the time elapsed between the estimated onset of a joint look and the instant when the system detects the joint look. As per the discussion above, Estimated Detection time can be expressed as Latency minus Estimated joint reaction time, which is 500ms. For future developments in this system, we will manually examine the gaze trace to estimate saccade reaction time for individual subjects. This will be important as we seek to trade off the speed of algorithmic detection of a joint look with the cost of the computer needed to run the system.

3.4.4 Look Positive Predictive Value (LPPV). Another point to observe is that that RL(1,1) would be particularly vulnerable to false positive detection of a joint look as it could declare a single frame glance passing across an object to be a look. False positives are penalized by the metric *'Look Positive Predictive Value (LPPV)'* which is defined as follows:

$$LPPV = \frac{Number\ of\ True\ Joint\ Looks\ Detected}{Total\ Number\ of\ Joint\ Looks\ Detected}$$

To determine this metric, we need to be able to define the ground truth (GT) on looks and joint looks. That could be done with a time-consuming manual process. However, in our case, since the algorithms under study are attempting to respond (with a feedback signal) in real-time to the onset of a joint look, the GT can be defined using a non-real-time process with a longer time window than is used by the algorithms under study. Here, the GT on looks is defined first and the GT on a joint look is obtained when the GT of looks for both subjects gives a hit. In this approach, the GT algorithm for looks begins by finding a series of 4 consecutive hit frames, which are considered to be the core part of a look. The data stream is then examined both forwards and backwards in time from that core, to bridge small gaps (up to an including gaps of length 3), to find the total extent of the look. For example, if the sequence of hits (`X`) and misses (`0`) is as given in the first row below:

```
0X000XXXX00XX000000XX00
0XXXXXXXXXXXX000000XX00
```

then the GT look is determined to have a total extent of 12 frames with an onset in position 2 as shown in the second row above. Here, the onset of the look would be correctly found by an RL(1,1) algorithm. The runlength algorithms RL(2,2) up through RL(4,4) will detect the look in the case above, but will have some latency for the detected onset. On the other hand, a sequence like:

```
0X0000XXX000000XX000000
```

does not have a true look at all, and any of the RL(**W**,**O**) algorithms with **W**<4 and **O**≤**W** would detect a false positive. Hence, we expect to have a clear trade-off between latency and LPPV.

It is apparent from the above definition of true look that the RL(4,4) will never have a false positive. All the other algorithms, such as RL(1,1), RL(2,2), RL(3,2), RL(4,3), RL(3,3) will have lower latency than RL(4,4) but will have some false look detections. Also, note that for LPPV calculation, we first obtain the LPPV for each pair of subjects and then calculate the mean and standard deviation of those values to obtain the final LPPV measures.

In general, a metric of positive predictive value (PPV) which penalizes false positives is used in conjunction with true positive rate (TPR) which penalizes false negatives. In our case, for all the runlength algorithms under study, TPR for joint looks is 1 because all true looks are eventually detected by all algorithms. (Note that TPR at the frame level for object detection is not generally equal to unity, but TPR at the level of joint-look detection is.) So the relevant quantities are LPPV and latency, where the latency metric penalizes the delay in declaring a look.

4 EXPERIMENTS

4.1 Calibration Routine

Calibration of the glasses is the first step in our experiment and is used to map the pupil gaze location captured by the binocular eye-facing infrared cameras to the real-world frames captured by the world-view camera. We select the manual calibration mode in the Pupil Labs software and use a manual marker with a bullseye pattern. The participants were asked to keep their head still and only move their eyes to look around. This method of calibration, where the wearer keeps the head still during calibration, was observed to give better accuracy of the gaze position than when the head is moving around.

During calibration, we first adjust the eye cameras of the subjects to ensure that the pupil is being detected and tracked. Pupil Capture software gives the flexibility to adjust the intensity, and minimum and maximum pupil size parameters. Next, they are instructed to look at the center of the marker as it is moved across the wall from one extreme location to another. The bullseye pattern is detected by the software in the world-view camera, which is then correlated with the gaze position captured at that time instant, assuming that the subject was looking at the center of the pattern. A total of at least nine points are collected and a non-linear transformation is used by the software to map the gaze positions to the incoming video frames. We selected a confidence threshold of 0.5; the system will reject points which it deems to have lower accuracy than this threshold. This routine is done separately for each of the glasses as the extrema of the field of view depend greatly on the position and orientation of the volunteer. After the calibration is completed, both volunteers are allowed to move their heads normally and are asked to look at the individual objects to perform a quick verbal check of the calibration routine.

4.2 Experiment Design

A total of eight volunteers, all current graduate students, were used to gather data for our experiments. The average age of the volunteers was 23, and half of them wore glasses. For each sitting, we randomly selected two students for the joint attention experiment. The volunteers were asked to switch places during the experiment. As explained in Section 3.2, we have a total of five objects, with three placed on the wall and two on the table in front of them. Our experiments involve one person who conducts the experiment and two participants who wear the glasses for the joint attention detection. We ask the volunteers to look at the individual objects and we record the time taken by the pair to jointly fixate their gaze at that object. To accurately capture the time when the instruction is given to look at a particular object, we developed a simple Android app which the experimenter uses to signal the participants. It consists of buttons of individual objects, which when pressed send a message to the receiver running on the server. A "cancel" button is provided to discard the current reading in case of accidental clicks. The server records the timestamp when the message is received and produces a beep, which acts as a cue for the volunteers to look at the specified object. Our system provides an audio feedback in real-time when the joint look on the specified object is detected. To minimize any phase difference between the recorded timestamps of the instruction to look at an object and the actual looks of the two subjects, the same hardware is used for running our message receiver and our custom software. A total of five rounds are conducted, with each round constituting all the five objects. Then participants are asked to switch places for another five rounds, after which the procedure is repeated with a new pair of volunteers. A total of 500 joint looks were obtained using different combinations of the volunteers. A sample world-view frame as recorded by the glasses with the gaze dot is shown in Figure 6.

5 RESULTS AND DISCUSSIONS

5.1 Object Detection

We ran the object detector on 80,000 annotated test images. As mentioned in Section 3.2, the performance of the detector was evaluated by computing the IoU, TPR and Precision. The average results for each object are summarized in Table 2.

The object detector gives an average True Positive Rate of 95.9%. The missed cases (False negatives) are observed to be mostly the edge cases where only a part of the object is visible in the world-view frame and it has been annotated in the GT image. This will cause an error in joint attention detection only if: (1) the object is visible partially, and (2) the wearer looks at the object out of the corner of his eyes. Since the wearer moves his head around normally, this edge case is anticipated to occur rarely and hence it is not expected to affect the overall performance of the system. The network gives a relatively lower detection rate for the cards since the portion of the cards captured by the world-view camera was heavily dependent on its orientation with respect to the subject.

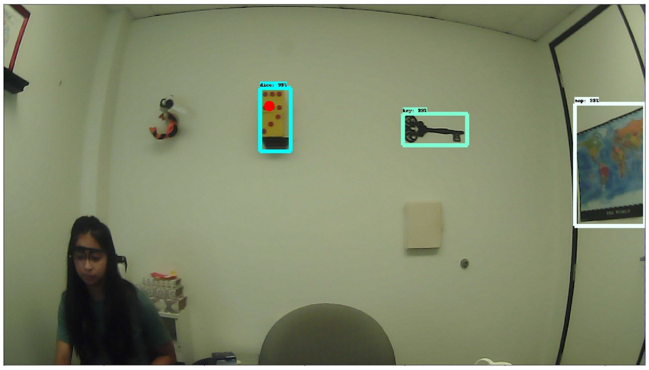

Figure 6: Sample world-view frame recorded by the glasses with the gaze dot and detection bounding boxes

Table 2: Performance of the object detector

	Ball	Dice	Key	Map	Cards
Mean IoU	92.3%	92.9%	82.1%	86.2%	87.5%
TPR	98.9%	99.1%	96.8%	94.5%	90.3%
Precision	97.0%	97.7%	98.2%	96.3%	99.3%

5.2 Experiment

On running our software with both the glasses, the system gives a frame rate of 16 fps. The lower frame rate is mainly due to the two object detection networks running in parallel on the GPU. There were a few readings that were discarded either because the experimenter pressed the button twice or the participants missed the audio cue provided. Apart from these scenarios, no other procedure was used for pre-processing of data points.

5.3 Latency and Estimated Detection Time

The results for the Latency of the system with various runlength parameters for different objects are illustrated in Table 3. Estimated Detection Time can be obtained from latency by deducting the average reaction time of 500 ms from the latency measurement. We observe in general that the latency for each runlength parameter follows the order Map < Key < Dice < Ball < Cards. The high latency for Cards can be explained by the relatively low detection rate for cards. For the other objects, the latency for a given runlength filter generally follows the order of decreasing object sizes. This is because our system is designed to detect the start of a look as soon as the gaze dot falls within the object bounding boxes, which is easier if the box size is larger.

For the same object across different runlength parameters, the latency increases as expected from RL(1,1) to RL(4,4). This is apparent from our definition of the filter in Section 3.4.1.

Table 3: Performance of the Joint look detector : Latency (ms): Mean (Std. Dev.)

	RL(1,1)	RL(2,2)	RL(3,2)	RL(3,3)	RL(4,3)	RL(4,4)
Ball	873.0 (169.6)	924.5 (181.8)	965.1 (187.9)	978.6 (190.2)	1017.6 (193.3)	1025.6 (197.5)
Cards	1008.6 (171.5)	1071.2 (184.5)	1111.0 (184.5)	1159.9 (253.2)	1196.8 (256.1)	1214.4 (268.2)
Dice	833.9 (233.2)	911.6 (263.5)	946.7 (264.4)	974.0 (290.8)	1007.8 (290.2)	1011.1 (290.3)
Key	826.3 (226.8)	887.0 (242.8)	926.2 (243.6)	965.5 (301.0)	1005.0 (302.0)	1008.4 (300.7)
Map	795.1 (137.8)	849.2 (155.1)	889.0 (164.0)	903.7 (167.7)	945.5 (168.5)	951.1 (168.9)

Table 4: Performance of the Joint look detector : LPPV%: Mean (Std. Dev.)

	RL(1,1)	RL(2,2)	RL(3,3)	RL(3,3)	RL(4,3)	RL(4,4)
Ball	88.2 (8.55)	90.4 (5.03)	90.4 (5.03)	97.8 (2.67)	97.8 (2.67)	100.0 (0.0)
Cards	82.9 (10.30)	85.1 (6.97)	85.1 (6.97)	95.7 (10.67)	95.7 (10.67)	100.0 (0.0)
Dice	85.1 (7.20)	93.6 (10.80)	93.6 (10.80)	100.0 (0.0)	98.9 (1.54)	100.0 (0.0)
Key	85.1 (11.64)	90.4 (8.96)	90.4 (8.96)	98.9 (1.33)	98.9 (1.33)	100.0 (0.0)
Map	91.4 (18.79)	93.6 (16.25)	93.6 (16.25)	97.8 (5.96)	97.8 (5.96)	100.0 (0.0)

5.4 Look Positive Predictive Value (LPPV)

The results for the LPPV of the system with various runlength parameters for different objects are illustrated in Table 4. The definition of a true joint look involves having a core of 4 consecutive hit frames and hence RL(4,4) has LPPV=100%. In general, as the window size parameter **W** increases, the LPPV also increases. This is expected as increasing window size makes the runlength filter reach closer to the size of the core-look part of the GT look and hence increases the possibility of detecting a true joint look.

Across objects, we observe that LPPV generally increases with increasing bounding-box size for smaller runlength parameters. This is due to the inherent jitter in the gaze position calculation caused by the pupil detection algorithm. As we increase our runlength filter parameters, our system becomes more robust and hence we observe that the change in LPPV across objects is almost negligible.

As expected, we observe a trade-off between Latency and LPPV measurements across runlength parameters. For lower values of **W** and **O**, we obtain a faster detection but with a lower accuracy measure. Hence for latency sensitive applications such as interactive AR and VR gaming, smaller runlength parameters are suitable, whereas for use in therapeutic applications, which require higher detection accuracy, larger parameters might be preferred.

6 LIMITATIONS AND FUTURE WORK

We have designed an inexpensive system running on a standard gaming laptop to detect joint looks to particular objects in real time. To our knowledge, this is the first such demonstration. Going forward, we would like to augment this system with other abilities. For example, we would like to detect what is called a gaze triad, that is, the look to a face before and after a look to an object. We also hope to detect shared eye gaze. Currently, we obtain the ground truths for joint look detection using an algorithm. To make our latency and LPPV calculations more accurate, we would like to compare the system detections to ground truths of joint looks labeled manually by an expert. Also, we would like to track the movement of gaze dot relative to the object bounding boxes across frames to speed up the look detection process.

As discussed in Section 1, children typically learn joint attention skills beginning in the first year of life through sharing gaze with a caregiver. However, when they do not develop this skill typically, as often happens for children on the autism spectrum, interventions to correct this important skill for early learning are costly and time consuming. Our goal is to develop a low-cost and easy to use system for rewarding joint attention behaviors in young children who need this support. The first step in implementing gaze triad detection would be to add a face detection network to our system. As this is a highly GPU-intensive task, it will require a more powerful GPU. We would also like to extend our system to include more objects from the real world. In the next step, we want to analyze the joint attention behavior of children as compared to adults by having child subjects for the experiment. Finally, as we develop these capabilities, we would like to detect and analyze gaze patterns of children with ASD. We will need to include adaptive rewards to broaden what constitutes a close-enough look spatially and temporally early in the training process. This will help scaffold the child's behavior and allow the system to narrow the reward specificity with training.

7 CONCLUSION

This project brings together different techniques to build a system that characterizes joint attention. This can help us broaden our understanding of human gaze behavior in real-world situations. The system described here allows us to detect joint looks in real time, and can be used as a tool in social therapies to analyze the response of a subject to a joint attention cue. Given the wide prevalence of ASD and the role of joint attention as a relevant social skill associated with it, our tool can be helpful in conducting social therapies and analyzing their progress over time.

ACKNOWLEDGMENTS

We would like to thank Mr. Zhishen Ding, Ms. Sarah Hacker, Ms. Chelsea Barrows, Mr. Harshul Gupta, Mr. Aman Raj and Ms. Janel

Moira Cuevas for helping us with the experiments and data collection. We are grateful to the UCSD Chancellor's Research Excellence Scholarships for partly funding the project and also would like to acknowledge support from the National Science Foundation under Grants SBE-0542013 and IIS-1522125. We also thank Mr. Daniel Truong and Mr. Yiming Zhang for helping with the project in its initial stages.

REFERENCES

Leanne Chukoskie, Shengyao Guo, Eric Ho, Yalun Zheng, Qiming Chen, Vivian Meng, John Cao, Nikhita Devgan, Si Wu, and Pamela C. Cosman. 2018. Quantifying Gaze Behavior During Real-World Interactions Using Automated Object, Face, and Fixation Detection. *IEEE Transactions on Cognitive and Developmental Systems* 10, 4 (2018), 1143–1152. https://doi.org/10.1109/TCDS.2018.2821566

Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. 2016. R-FCN: Object Detection via Region-based Fully Convolutional Networks. *CoRR* abs/1605.06409 (2016). arXiv:1605.06409 http://arxiv.org/abs/1605.06409

John M. Franchak, Kari S. Kretch, and Karen E. Adolph. 2014. See and be seen: Infant-caregiver social looking during locomotor free play. *Developmental Science* 21, 4 (2014), e12626. https://doi.org/10.1111/desc.12626 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/desc.12626

Kelly Stickles Goods, Eric Ishijima, Ya-Chih Chang, and Connie Kasari. 2013. Preschool Based JASPER Intervention in Minimally Verbal Children with Autism: Pilot RCT. *Journal of Autism and Developmental Disorders* 43, 5 (01 May 2013), 1050–1056. https://doi.org/10.1007/s10803-012-1644-3

Jasmin Kajopoulos, Alvin Hong Yee Wong, Anthony Wong Chen Yuen, Tran Anh Dung, Tan Yeow Kee, and Agnieszka Wykowska. 2015. Robot-Assisted Training of Joint Attention Skills in Children Diagnosed with Autism. In *Social Robotics*, Adriana Tapus, Elisabeth André, Jean-Claude Martin, François Ferland, and Mehdi Ammi (Eds.). Springer International Publishing, Cham, 296–305. https://doi.org/10.1007/978-3-319-25554-5_30

Connie Kasari, Marian Sigman, Peter Mundy, and Nurit Yirmiya. 1990. Affective sharing in the context of joint attention interactions of normal, autistic, and mentally retarded children. *Journal of Autism and Developmental Disorders* 20, 1 (01 Mar 1990), 87–100. https://doi.org/10.1007/BF02206859

Moritz Kassner, William Patera, and Andreas Bulling. 2014. Pupil: An Open Source Platform for Pervasive Eye Tracking and Mobile Gaze-based Interaction. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication (UbiComp '14 Adjunct)*. ACM, New York, NY, USA, 1151–1160. https://doi.org/10.1145/2638728.2641695

Ben Kenward, Felix Koch, Linda Forssman, Julia Brehm, Ida Tidemann, Anett Sundqvist, Carin Marciszko, Tone Hermansen, Mikael Heimann, and Gustaf GredebÃďck. 2017. Saccadic Reaction Times in Infants and Adults: Spatiotemporal Factors, Gender, and Interlaboratory Variation. *Developmental Psychology* 53 (12 2017). https://doi.org/10.1037/dev0000338

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. *CoRR* abs/1405.0312 (2014). arXiv:1405.0312 http://arxiv.org/abs/1405.0312

Nikita Manovich, Boris Sekachev, Andrey Zhavoronkov, Victor Salimonov, Dmitry Sidnev, and Sebastian Yonekura. 2018–2019. Computer Vision Annotation Tool (CVAT). https://github.com/opencv/cvat.git.

Michael Miller, Leanne Chukoskie, Marla Zinni, Jeanne Townsend, and Doris Trauner. 2014. Dyspraxia, motor function and visual-motor integration in autism. *Behavioural Brain Research* 269 (2014), 95 – 102. https://doi.org/10.1016/j.bbr.2014.04.011

Tanya Paparella and Stephanny F N Freeman. 2015. Methods to improve joint attention in young children with autism: a review. *Pediatric Health, Medicine and Therapeutics* 2015 (05 2015). https://doi.org/10.2147/PHMT.S41921

Sarah M. Rabbitt, Alan E. Kazdin, and Brian Scassellati. 2015. Integrating socially assistive robotics into mental healthcare interventions: Applications and recommendations for expanded use. *Clinical Psychology Review* 35 (2015), 35 – 46. https://doi.org/10.1016/j.cpr.2014.07.001

Joseph Redmon. 2013–2016. Darknet: Open Source Neural Networks in C. http://pjreddie.com/darknet/.

Joseph Redmon and Ali Farhadi. 2018. YOLOv3: An Incremental Improvement. *CoRR* abs/1804.02767 (2018). arXiv:1804.02767 http://arxiv.org/abs/1804.02767

Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *CoRR* abs/1506.01497 (2015). arXiv:1506.01497 http://arxiv.org/abs/1506.01497

Lauren K. Slone, Drew H. Abney, Jeremy I. Borjon, Chi-hsin Chen, John M. Franchak, Daniel Pearcy, Catalina Suarez-Rivera, Tian Linger Xu, Yayun Zhang, Linda B. Smith, and Chen Yu. 2018. Gaze in Action: Head-mounted Eye Tracking of Children's Dynamic Visual Attention During Naturalistic Behavior. *J. Vis. Exp.* 141 (2018), e58496. https://doi.org/10.3791/58496

Sebastian Thrun, Wolfram Burgard, and Dieter Fox. 2005. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press.

Jeanne Townsend, Eric Courchesne, and Brian Egaas. 1996. Slowed orienting of covert visual-spatial attention in autism: Specific deficits associated with cerebellar and parietal abnormality. *Development and Psychopathology* 8, 3 (1996), 563âĂŞ584. https://doi.org/10.1017/S0954579400007276

Chen Yu and Linda B. Smith. 2016. The Social Origins of Sustained Attention in One-Year-Old Human Infants. *Current Biology* 26, 9 (2016), 1235 – 1240. https://doi.org/10.1016/j.cub.2016.03.026

Lonnie Zwaigenbaum, Susan Bryson, Tracey Rogers, Wendy Roberts, Jessica Brian, and Peter Szatmari. 2005. Behavioral manifestations of autism in the first year of life. *International Journal of Developmental Neuroscience* 23, 2 (2005), 143 – 152. https://doi.org/10.1016/j.ijdevneu.2004.05.001 Autism: Modeling Human Brain Abnormalities in Developing Animal Systems.