VADS: Visual Attention Detection with a Smartphone

Zhiping Jiang*, Jinsong Han*, Chen Qian†, Wei Xi*, Kun Zhao*, Han Ding*,
Shaojie Tang‡, Jizhong Zhao*, Panlong Yang§
* Xi'an Jiaotong University, China, † University of Kentucky, U.S,
‡ University of Texas at Dallas, U.S, § PLAUST, China
{*jiangzp.cs, dinghanxjtu, panlongyang*}@gmail.com, *qian*@cs.uky.edu,
shaojie.tang@utdallas.edu, {*hanjinsong, xiwei, zhaokun2012, zjz*}@mail.xjtu.edu.cn

***Abstract*—Identifying the object that attracts human visual attention is an essential function for automatic services in smart environments. However, existing solutions can compute the gaze direction without providing the distance to the target. In addition, most of them rely on special devices or infrastructure support. This paper explores the possibility of using a smartphone to detect the visual attention of a user. By applying the proposed VADS system, acquiring the location of the intended object only requires one simple action: gazing at the intended object and holding up the smartphone so that the object as well as user's face can be simultaneously captured by the front and rear cameras. We extend the current advances of computer vision to develop efficient algorithms to obtain the distance between the camera and user, the user's gaze direction, and the object's direction from camera. The object's location can then be computed by solving a trigonometric problem. VADS has been prototyped on commercial off-the-shelf (COTS) devices. Extensive evaluation results show that VADS achieves low error (about 1.5° in angle and 0.15m in distance for objects within 12m) as well as short latency. We believe that VADS enables a large variety of applications in smart environments.**

I. Introduction

Visual attention potentially represents human mental activities such as planning or purpose [1]. Human visual attention detection determines the relative location of the object that a person is looking at. It provides tremendous benefits for intelligent services in smart environments. We highlight several typical scenarios where visual attention detection plays an important role.

- **Smart control system.** People may perform automatic control of factory machinery and home appliances without physical contact. Suppose a teacher walks into a classroom and wants to turn on a light in the back of the room. She can simply watches the light. The system will identify the light and then automatically turn it on.
- **Smart labelling.** People may easily label any item in the smart environment with written information. In a furniture store, a customer may look at a mattress and write her review such as "This mattress is the firmest in the store". This customer-generated tag will be sent to and stored in the smart environment system.
- **Smart information retrieval.** People may request related information of an object by simply looking at it. In a shopping mall, a customer may look at a restaurant and let a mobile device understand her visual attention. Then she can read all related information of the restaurant such as business hours, menu, and customer reviews.

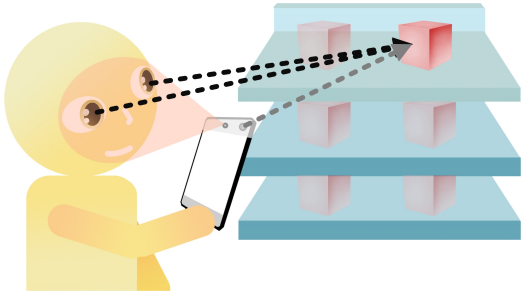


Fig. 1: Illustration for VADS system. User can instantly access the information of the intended object via only a gaze.

Obviously, conventional human-computer interface, such as keyboard, voice, touchscreen, or wireless smart sensing based techniques [2]–[5] *etc.* can hardly provide such convenience.

Recent studies have been investigating how to recognize human visual information using wearable devices such as Google Glass. For example, iGaze [6] and iShadow [7] are two recent systems that track human eye gaze. These approaches have two main limitations. First, glass-like wearable devices are not ubiquitous and it is cost-inefficient to wear special devices just for interacting with a smart environment. In fact, methods that can be implemented on smartphones are much more desired. Second, these approaches can only detect gaze directions and do not provide the *distance* of the intended object. They could be error-prone when multiple objects sit on a same gaze direction.

In this work we design and implement a smartphone-based visual attention detection system, called VADS. The user operation of VADS is very simple. As shown in Fig. 1, a user stares at the intended object directly, while holding up her smartphone as to take a photo for the object. VADS can simultaneously capture the object as well as the user's face from the rear and front cameras respectively, and then compute the relative location, which includes both the gaze direction and distance to the object.

978-1-4673-9953-1/16/$31.00 ©2016 IEEE

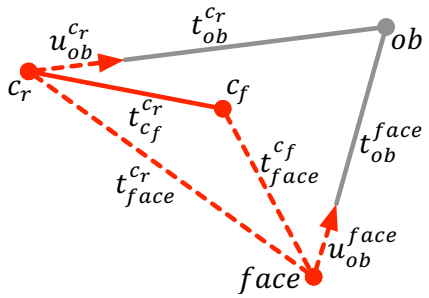


Fig. 2: Geometry expression for VADS. The notations are described in Table. I.

TABLE I: Symbols used in this paper.

Symbol	Description
$navi$	The local-navigational frames
c_r, c_f	The rear and front camera-centered coordinate system (frame)
$face$	The face coordinate system (frame)
ob	The intended object
t_p^{fx}	The position (vector) to p in fx-frame
u_p^{fx}	The unitary directional vector towards p in the fx-frame
R_{fo}^{fn}	The rotation matrix which transform the coordinate system from the fo-frame to fn-frame

In this system, we build an accurate facial model in advance as the reference for calculating the distance and angle from the user to the front camera. Then the smartphone tracks the angle from the object to the rear camera. In addition VADS also computes the gaze direction from the user to the object. With these parameters, VADS forms an *virtual binocular vision* system. The distance from the user to the object can be thereby calculated by trigonometric computation, and hence the object is localized and further identified. The information about the intended object can be ultimately retrieved from a location based service (LBS) of the smart environment.

We summarize the major contributions of this work as follows.

• We design a visual attention detection system and implement it on Commodity Off-The-Shelf (COTS) smartphones. Compared to previous approaches, VADS extends the spatial resolution of visual attention detection from one-dimension (angle-only) to 3D (angle-distance).

• VADS is based on existing smartphone systems and does not rely on any extra infrastructure. It is easy to operate and significantly reduces the infrastructure and equipment cost.

• We extend the current advances of computer vision and design algorithms to estimate the gaze direction of a human face in a picture. Innovations of VADS's gaze direction estimation include unsupervised face modeling, a novel iris center localization method, and an accurate yet efficient linear gaze model.

• We prototype VADS on the iOS platform. Extensive experimental results show that VADS achieves high accuracy and low latency.

II. Related Work

Capturing human's visual attention is a very challenging task. In the literature, biological signals based brain imaging are extensive studied for interpreting the intention. For example, fMRI based approaches [8] can localize the brain activation area and then correlate to certain stimulus. However, the mapping mechanism between human intention and activation remains unclear [9]. Besides directly sensing the brain signal, researchers find that the human's gaze is quite indicative for identifying human attention [10], and can be correlated to certain brain activities [11]. To realize the visual attention detection, It is necessary to achieve accurate gaze estimation. Prior works for gaze estimation can be categorized into two groups: model-based approaches and appearance-based approaches. The model-based approaches [6], [12] usually adopt a 2D or 3D eyeball model. By localizing the iris center, the poses of the eyeballs are estimated, and eventually the gaze direction can be computed. Since the human cornea has strong reflection in visible light spectrum, some approaches use infrared cameras to capture the eyes image [13]. Appearance-based approaches [7], [14], [15] avoid the complex modeling for eyeballs. They treat the complete eye image as a description vector. Feng *et al.* [14] estimate the gaze angle by comparing the captured eye image to a large set of synthesized ones. Yusuke *et al.* [15] use a Gaussian process regression to establish the mapping between gaze point and image saliency vector. However, most of previous approaches operate in controlled scenarios. Recently, mobile devices are leveraged for gaze estimation. iGaze [6] and iShadow [7] are designed and implemented based on the glass-style hardware. Specifically, iGaze [6] uses a fine-grained 3D eyeball model based algorithm, while iShadow [7] utilizes a feed-forward neural network based scheme.

Two most similar work to VADS are OPS [16] and CamLoc [17]. Both of them can localize an remote object or building. However, each of them has certain rigid requirements that constrain their usage. OPS requires the user to take multiple photos from different positions. The building's position is then estimated via multilateration. Compared to VADS, OPS also needs a large angle span, which is inconvenient for instant use. CamLoc estimates the distance by comparing the object's appearance size in two photos. Thus, it requires the user to take two pictures with special arm gestures. However, it suffers from low accuracy when the building is poorly segmented from the pictures.

III. System Overview

The idea of VADS is based on a simple 3D geometric problem. From measurement and computation, we may obtain four elements using the smartphone shown as the red elements in Fig. 2. They are the directional vectors from the rear camera and user's face to the intended object, denoted as $u_{ob}^{c_r}$ and u_{ob}^{face} respectively, the face position in the front camera view

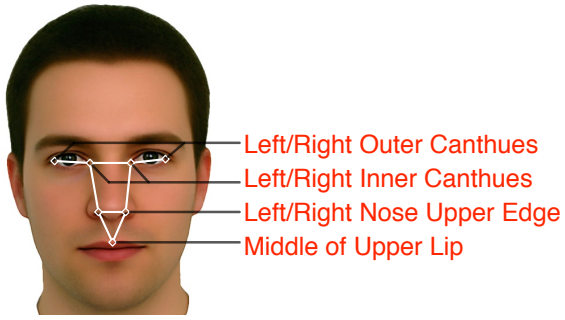
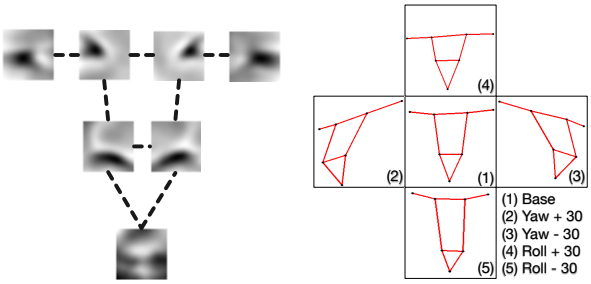


Fig. 3: The selected facial anchor nodes. The white lines show their shape constraints.



(a) Trained 2D Patches (b) Trained Shape Model

Fig. 4: (a) shows the trained patch model for a volunteer. (b) shows the trained shape model, which has two intrinsic parameters, the yaw and roll angle.

t^{cf}_{face}, and the the position of the front camera *w.r.t.* the rear camera t^{cr}_{cf}. Given this information, the goal of VADS is to determine the relative location of the intended object, denoted as t^{face}_{ob}. The visual attention can then be easily computed on the smartphone.

VADS works in three phases, face tracking & pose estimation, gaze direction estimation, and visual attention localization. In the first phase, VADS tracks the user's face from her/his smartphone's front camera. The face position t^{cf}_{face} and the pose R^{cf}_{face} are determined. In the second phase, with the face direction obtained in the first phase, VADS localizes the iris center and estimates the user's gaze direction u^{face}_{ob}. In the last phase, VADS calculates its directional vector u^{cr}_{ob}. Since t^{cr}_{cf} is a pre-known element, VADS can calculate t^{cr}_{ob} and t^{face}_{ob}.

There are 3 main challenges in the VADS design. We will address them in the following sections.

- Prior face tracking methods usually demand high-performance computing capability and cannot be directly used on the computational resource-limited mobile platform. We have to pursue an effective face tracking solution that satisfies the following requirements: high accuracy, high robustness, and low latency.
- It is not easy to acquire an accurate 3D model for facial features using existing solutions. Prior work usually requires high quality images, which is difficult to be obtained by mobile smartphones. Meanwhile, they do not have a general parameter tuning mechanism and reducing the tuning overhead is also non-trivial.
- In the phase of gaze estimation, a big challenge is to accurately localize the iris center. Prior approaches usually work in an ideal condition, where most iris area should be observed by the camera. In the VADS operating scenario, large part of iris area, however, may be occluded by eyelids. Furthermore, the front camera of user's smartphone is often in low-resolution. In addition, the cornea's reflection poses non-negligible impact to the localization. Thus we cannot directly utilize existing solutions.

IV. Face Tracking and Pose Computation

Two factors are indispensable to determine the position and pose of user's face (t^{cf}_{face}, and R^{cf}_{face} respectively). They are the accurate tracking for some anchor nodes on user's face and the accurate 3D coordinates for these nodes. In our system, seven highly distinguishable facial features are chosen as the facial anchor nodes, as shown in Fig. 3.

To track these anchor nodes, One major challenge is that prior approaches are inaccurate or inefficient on the mobile platform. there are very few options that can track these anchor nodes accurately and efficiently on the mobile platform. For example, conventional object tracker [18] is easy to drift away. Active Appearance Model (AAM) [19] or its variants [20] suffer from a low tracking accuracy. The feature-alignment based approach, such as ERT [21], is accurate and robust. However current smartphones cannot afford its computational overhead.

We choose Active Shape Model (ASM) [22]–[24], a simple yet efficient approach, as our tracking framework. Traditional ASM is not robust for wide dynamic range of observation angle. We tailor it for the use in mobile phones and optimize it in terms of tracking accuracy, robustness, and efficiency.

A. ASM-based Facial Anchor Nodes Tracking

ASM is a global optimization-based approach. Instead of localizing each feature point individually, ASM operates in another way. It treats the features together as an intact object and then globally optimize the feature localization according their spatial constraints. To realize this idea, ASM relies on two models, namely the *patch* model and *shape* model. The *patch* is a detector trained for a specific feature, while the *shape* model encapsulates the geometric constraints. ASM is computational efficient and suitable for mobile platform. However, a drawback of ASM is its unreliability for human face tracking applications [23], [24]. The 1-D gradient vector based patch model is unstable for any small pose variation. To deal with this problem, we enhance the ASM with correlation-based 2D patch model. while enables accurate and robust tracking for facial anchor nodes on the mobile platform.

Correlation-based 2D Patch Model: For a given facial feature f_k, we consider the optimal correlation-based patch

$\hat{\mathbf{P}}_{f_k}$ as the solution to the following optimization problem:

$$\begin{aligned} \hat{\mathbf{P}}_{f_k} &= \arg\min_{\mathbf{P}} \mathcal{F}(\mathbf{P}) \\ \mathcal{F}(\mathbf{P}) &= \sum_{i=1}^{N} \sum_{x,y} \|\mathbf{R} - \mathbf{P} \cdot \mathbf{T}^i_{(x,y)}\|_F^2 \\ s.t. \quad & \arg\max_{\mathbf{T}^i_{(x,y)}} \sum_{i=1}^{N} \sum_{x,y} \mathbf{P} \cdot \mathbf{T}^i_{(x,y)} = \mathbf{T}^i_{f_k} \end{aligned} \quad (1)$$

where $\mathbf{R}$ is an ideal response map that has a centered 2D-Gaussian distribution with very small σ, $\mathbf{T}^i_{(x,y)}$ is the small image tile located at (x, y) of the i-th training image, and $\mathbf{T}^i_{f_k}$ is the small tile right-centered at feature f_k in the i-th image. The idea behind Eq. 1 is intuitive: *the optimal patch* $\hat{\mathbf{P}}_{f_k}$ *should yield the highest response* ***iff*** *the test image tile* $\mathbf{T}^i_{(x,y)}$ *contains the feature* f_k.

Under this 2D patch model, the training process for $\hat{\mathbf{P}}_{f_k}$ can be very efficient. Actually, Eq. 1 is in the standard form of linear least square (LLS). The optimal solution can be approximated using stochastic gradient descent approach. The gradient of $\mathcal{F}(\mathbf{P})$ in the i-th step is:

$$\nabla\mathcal{F}(\mathbf{P}) = -2\sum_{x,y}(\mathbf{R} - \mathbf{P}\mathbf{T}^i_{(x,y)})\mathbf{T}^i_{(x,y)} \quad (2)$$

and $\hat{\mathbf{P}}_{f_k}$ can be obtained iteratively as

$$\hat{\mathbf{P}}^i_{f_k} = \hat{\mathbf{P}}^{i-1}_{f_k} - \alpha\nabla\mathcal{F}(\mathbf{P}) \quad (3)$$

Before the first use of VADS, a user records a short training video, and it is used to train his/her correlation-based 2D patch model. In this training video, the user's head keeps still and the phone moves around the face to cover wide viewing angles. The per-frame ground-truth of facial anchor nodes is obtained via ERT approach [21]. In this way, we train specific patch model and shape model for the user. Fig. 4 (a) shows an example of the trained 2D patches for the selected facial anchor nodes. The shape model is also extracted from the training images, following the standard ASM approach. It uses the Principle Component Analysis (PCA) to capture the shape variations *w.r.t.* the average shape. In our system, two most dominating parameters (yaw and roll angle) are captured. Fig. 4 (b) shows the extracted shape and its variants in both yaw and roll direction.

B. Facial 3D Model Extraction & Face Pose Computation

Obtaining the accurate 3D coordinates of the facial anchor points is the other indispensable task. Structure from Model (SfM) based solutions [17], [25] are widely adopted for similar problems. However, when we extracted a 3D surface from images using SfM, it suffers from unstable performance, high computation overhead, and low accuracy. In addition, its parameter tuning requires professional knowledge and is not easy to implement.

In our system, we decompose the 3D model extraction into two much easier tasks: obtaining the facial anchor nodes' 2D position model, and compute the relative heights for these nodes.

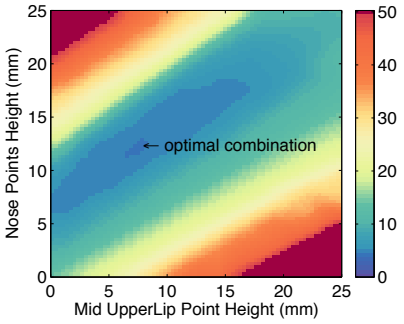


Fig. 5: The re-projection error after Gaussian filtering. The darkest point corresponds to the optimal combination of heights.

1) Obtain The Optimal 2D Model for Facial Anchor Nodes: The user is required to selected the most "right-up-front" face from the training video. The facial anchor nodes on this face are used as the anchor nodes. To prevent the error introduced by human visual imperfection, user is required to select multiple "right-up-front" faces, and the final coordinate for an anchor node is then the median position of these candidate faces.

2) Compute Relative Heights of Facial Anchor Nodes: With the constraint of the facial 2D model, a by-product of face pose estimation process can help VADS determine the relative heights. Pose estimation is an Perspective-*n*-Points (P*n*P) problem. Given the 3D coordinates of the anchor points on the object, and their corresponding 2D pixel coordinates on the image, a P*n*P solver [26] can estimate the object's 3D position and pose by solving an optimization problem, of which the optimization objective is the *re-projection error*, e^{rpj}. Apparently, if we feed a more accurate heights combination into the PnP solver, e^{rpj} is lower.

We use e^{rpj} to help us identify the optimal heights combination. To minimize the search space, we first assume the *left-outer*, *left-inner*, *right-inner*, and *right-outer* canthus points are coplanar at base height, *i.e.*, their heights are all 0. We then assume the heights of two nose edge points are identical. We further assume the height ranges of nose-edge points and mid-lip point are the same, $0mm < h_{nose} < 25mm$ and $0mm < h_{midlip} < 25mm$. With these three assumptions, we now only need to determine the optimal heights of nose-edge h_{nose} and mid-upper lid h_{midlip}.

We use the EPNP [26] algorithm as the P*n*P solver. We record the error for each heights combination. A Gaussian kernel filter is further applied to smooth the error surface. The point with minimal error is the chosen as the optimal heights combination. Fig. 5 shows the filtered error surface and the optimal height combination for h_{nose} and h_{midlip}. At last, VADS obtains the accurate 3D coordinates of facial anchor nodes.

3) Face Pose & Position Estimation: With the accurate 3D coordinates and 2D pixel coordinates of facial anchor nodes,

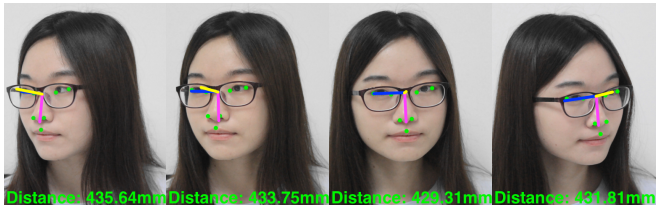


Fig. 6: The screenshot of face pose estimation module. The green dots denote the facial anchor nodes, and the colored lines denote the axes of $face$-frame. The distances from $face$-frame to front camera $t^{c_f}_{face}$, are shown at the bottom.

we use EPNP to compute the face pose and position *w.r.t.* front camera, denoted as $R^{c_f}_{face}$ and $t^{c_f}_{face}$ respectively.

To obtain the position *w.r.t.* rear camera, denoted as $t^{c_r}_{face}$, we need to change the coordinate system from c_f to c_r, which is a *rigid transformation*, as shown below.

$$t^{c_r}_{face} = \left[\begin{array}{c|c} R^{c_r}_{c_f} & t^{c_r}_{c_f} \\ \hline \mathbf{0} & 1 \end{array}\right] \times t^{c_f}_{face} \tag{4}$$

$R^{c_r}_{c_f}$ and $t^{c_r}_{c_f}$ together describe the coordinate system transformation from c_f to c_r. Fig. 6 shows a screenshot of face pose estimation module on mobile platform.

V. Computing Gaze Direction

In this section, we describe the method to compute user's gaze direction. We divide this task into two stages: iris center localization and gaze direction computation. The difficulties mainly exist in the accurate localization for iris center.

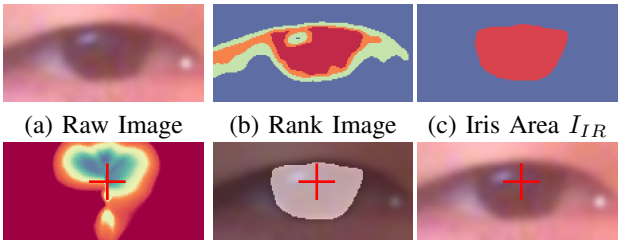
A. Iris Center Localization

Three major difficulties bring great challenge to the iris center localization. The first is the occlusion. The eyelids and eyelash often occlude a large portion of the iris area. The shadow casted by eyelids further blurs the boundary between the eyelid and iris. The second problem is the cornea's strong reflection. It often results in a bright spot on the iris boundary. The last one is the low quality imaging. The smartphone's front camera is much more inferior than the rear camera.

Various of existing solutions are tried to localize the iris center. However, none of them achieve high accuracy and low latency simultaneously. Circle detection based approach [27] cannot work because the iris boundary is ambiguous and incomplete. Gradient or isophote based approaches [28], [29] are not accurate due to strong noise around the iris boundary.

Facing these difficulties, we propose our two-step approach. We first perform an accurate iris area segmentation using an adaptive pixel ranking algorithm, and then we identify the iris center via a customized convexity metric.

1) Iris Area Segmentation: Given an eye image, it is intuitive to segment the iris area using color-based thresholding. However, the non-uniform luminance distribution and glow spot brought by cornea's reflection make the thresholding not workable. We propose an adaptive pixel ranking technique to handle this problem. The pixels' spatial relevance is first taken into consideration to preserve the structure information. It then uses an iterative pixel clustering process to overcome the non-uniform luminance distribution.



(a) Raw Image (b) Rank Image (c) Iris Area I_{IR}

(d) SpinS Response (e) Overlay View (f) Final Result

Fig. 7: An example of iris center localization. (a) is the raw eye image. (b) is the ranked-graph of the eye. (c) shows the best rank which closely covers the iris area. (d) shows the SpinS response and the minimum point. (e) and (f) shows the overlay view and the final result of iris center localization.

Given a color image, we denote its pixel set as P. For each pixel $p \in P$, we enlarge its data dimension such that $p = (r, g, b, \alpha x, \alpha y)$ where r, g, b is the RGB color values, (x, y) are the pixel's coordinate, and α, usually within $[0.1, 0.4]$, is the ratio used to combine the color and coordinate. By involving the pixel' coordinate, the impacts from both the color and position domains are taken into the consideration.

We then perform a k-round clustering process on P. In the i-th ($1 \leq i \leq k$) round, a 2-means clustering is performed. The input pixel set of i-th round, denoted by P_i ($P_1 = P$), is split into two clusters. One is lighter-colored and the other is the darker-colored. The darker-colored pixel set then becomes the input set for the next round, and their $rank$ increase 1. After k rounds clustering, each pixel is associated with a rank value. We denote this rank image as I_{rank}.

Given I_{rank}, the goal of iris area segmentation becomes to determine in which rank and above, the rank image is most probably the iris area. The solution is inspired by a finding observed in the preliminary evaluations: the iris area usually has *the most convex* shape, *i.e.*, the rank image with the highest convexity. The convexity of a shape S, denoted by $Cvxt(S)$, is defined as

$$Cvxt(S) = Ar(S)/Ar(CvxHl(S)) \tag{5}$$

where $Ar(S)$ returns S's area. and $CvxHl(S)$ returns S's *convex hull*. *Convex hull* is the smallest convex shape which contains the original one. We select the rank with the highest $CvxHl$, and use its *convex hull* as the iris area, denoted as I_{IR}. Fig. 7 (b) to (c) show the ranked image generation and the best rank extraction for the example eye image in Fig. 7 (a).

2) Iris Center Localization: The next step is to localize the iris center. We should note that the true iris center is not the shape center of I_{IR} but the center of the smallest circumscribe circle of I_{IR}. The challenge here is how to describe this iris

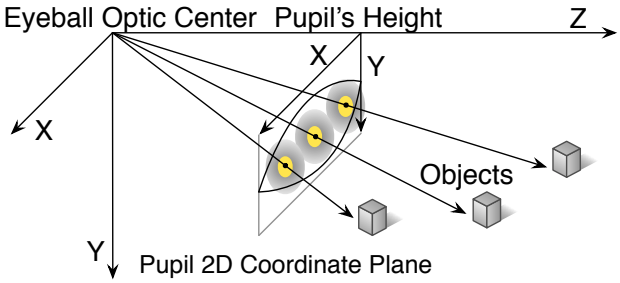


Fig. 8: Planar Gaze Projection Model.

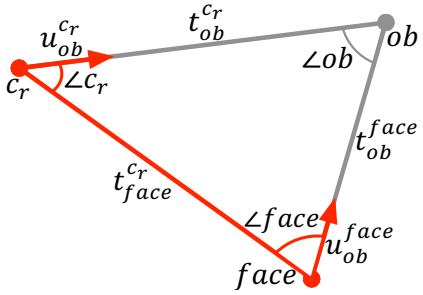


Fig. 9: A simplified model for intended object localization.

center. However, since I_{IR} is an irregular shape, there can hardly be a geometrical definition.

We propose an Rotation & Superimposition based iris center localization algorithm. The core idea is based on the following observation. Imagine that we randomly rotate a 2D shape S around a fix point p to yield a new 2D shape. Repeating n times, all shapes generated by the rotation are then superimposed into one shape called S_{si}. As the value of n gets larger, S_{si} will eventually become a disk. However, the *speed of becoming a disk* varies with different p. We find the speed reaches its maximum when p is at the shape center of S.

Based on this observation, we develop a customized metric, called **SpinS** (Spin and Superimposition). Given a rotation center p within I. SpinS first rotates the image multiple times around p and then superimposes the rotated images into one. It then evaluates the convexity of the image via Eq. 5. We denote this series of operations as $SpinS(I, p)$. The iris center localization can then be formulated as the following optimization problem

$$p_{is}(x, y) = \underset{p}{\operatorname{argmin}}\, SpinS(I_{IR}, p), p \in I_{IR} \tag{6}$$

The Stochastic Gradient Descent algorithm is adopted as the optimizer to minimize the computational cost.

Obtaining the 2D pixel coordinate of iris center $p_{is}(x, y)$, the 3D coordinate of iris center *w.r.t.* front camera, denoted as $t_{is}^{c_f}$, is simply the intersection point between the directional vector towards the iris center $u_{is}^{c_f}$ and the face plane P_{face}. We solve it using analytic geometry. Transforming $t_{is}^{c_f}$ from the c_f-frame to the $face$-frame, we obtain the coordinates of the iris center in the $face$-frame, t_{is}^{face}.

B. Gaze Direction Estimation

The task here is to establish the mapping between the iris center's coordinate t_{is}^{face} and the gaze direction in $face$-frame, denoted as u_{gaze}^{face}. Traditional approaches [30] use the eyeballs' 3D spherical model to establish the mapping. However, it requires accurate 3D parameters of eyeballs, which is usually not available for common mobile devices.

In the typical application scenario of VADS, the gaze angle are mostly within $\pm 30°$. In such a small angle span, a planar iris-gaze model is more suitable [30], as illustrated in Fig. 8. It is quite similar to the pinhole camera model [31]. The task now turns to compute the "intrinsic parameters" of the gaze pinhole model, denoted as A_{gaze}. It correlates the front camera's position $t_{c_f}^{face}$ with the iris center's coordinate t_{is}^{face} via a linear mapping. The training video is used again to help calculate A_{gaze}, and it is estimated via a simple linear regression. When a user stares at the intended object, $u_{ob}^{face} = u_{gaze}^{face}$.

VI. Computing Visual Attention

In this section, we compute the distance from the object to the user's eyes. With the user's gaze direction u_{ob}^{face}, the location of intended object is computed. Ultimately, the information of the object can be retrieved via location-based services (LBS).

A. Tracking Intended Object from the Rear Camera

VADS is designed to track the intended object in real-time and calculate its directional vector $u_{ob}^{c_r}$. The challenge here is the drift caused by the lack of training data. Without sufficient training images of the intended object, traditional tracking algorithms [18] may lost the target if it has significant movement. We use a long-term object tracking algorithm, namely TLD [32], to tackle this problem. It simultaneously learns the appearance model of the object and corrects the tracker.

Suppose TLD has successfully captured the object ob. Let (u, v) denote the ob's pixel coordinate in the image. The direction vector to ob in the c_r-frame, *i.e.* $u_{ob}^{c_r}$, is then obtained by

$$u_{ob}^{c_r} = A_{c_r}^{-1} \times [u, v, 1]^T \tag{7}$$

where A_{c_r} is the c_r's camera intrinsic matrix, which is available for each smartphone.

B. Visual Attention Detection

We have collected three key elements: $u_{ob}^{c_r}$, u_{ob}^{face}, $t_{c_r}^{face}$. Now we need to estimate the distance and position of the intended object t_{ob}^{face}.

To simplify the computation model, we assume $u_{ob}^{c_r}$ and u_{ob}^{face} are pointing to the same points on ob. In another word, $u_{ob}^{c_r}$, u_{ob}^{face}, and $t_{face}^{c_r}$ are coplanar. Therefore, the distance computation is simplified to an planar trigonometry problem,

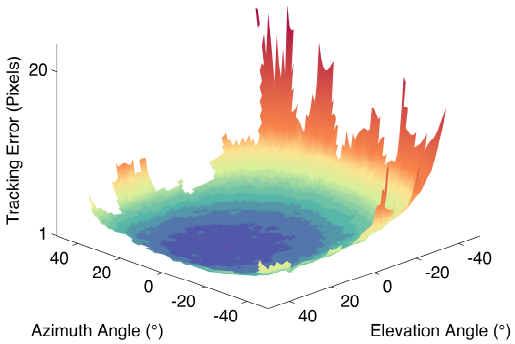


Fig. 10: ASM Face Tracking Accuracy *w.r.t.* heading Angle.

as shown in Fig. 9. Let $\angle c_r$, $\angle face$, $\angle ob$ be the three interior angles. According to the Law of sines, we have

$$\left\|t_{ob}^{face}\right\| = \sin c_r \times \frac{\left\|t_{face}^{c_r}\right\|}{\sin ob} \tag{8}$$

Therefore, the 3D coordinates of the object is

$$t_{ob}^{face} = \left\|t_{ob}^{face}\right\| \times u_{ob}^{face} \tag{9}$$

We assume an accurate localization system and corresponding LBS has been deployed in the environment. Let t_{face}^{navi} be the user's position. Therefore, the object's location $t_{ob}^{navi} = t_{face}^{navi} + t_{face \to ob}^{navi}$. The information of the object is finally retrieved by querying t_{ob}^{navi} in the LBS system.

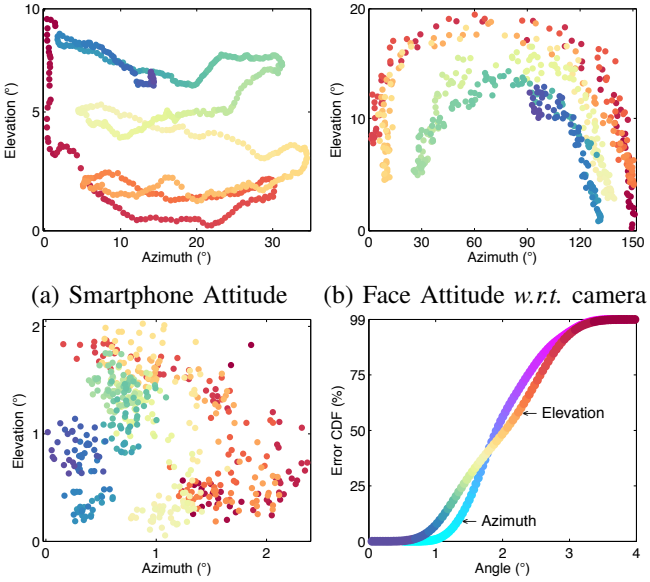
VII. Evaluation

VADS is built on OpenCV Library [33], and currently prototyped on iOS platform. Ten volunteers are invited to participate in the experiments and evaluate the performance of VADS.

There is one thing to note about the rear camera. The development of mobile devices offers great facilitates to the implementation of VADS With the rapid growth of computational performance, many smartphones have come up with their proprietary "Dual Shot" mode, which can liveview and capture the images from both the front and rear cameras simultaneously. We believe the standard camera API stack will support this new feature in the near future. However, public APIs for Dual Shot is still unavailable currently. To overcome this problem, we use a Wi-Fi connected camera, Sony Qx10 [34], to substitute for the built-in rear camera in our prototype. With public API, the 640x480 liveview image stream can be fetched at 20fps. Note that liveview image stream of built-in rear camera has a much higher performance than the Qx10 camera.

A. Evaluation of Face Tracking and Pose Estimation

In this set of experiments, a test video is captured for each volunteer to evaluate the accuracy of the extracted 2D/3D facial model and the face pose estimation.



(a) Smartphone Attitude (b) Face Attitude *w.r.t.* camera

(c) Face Pose in $navi$-frame (d) Face Pose Error CDF

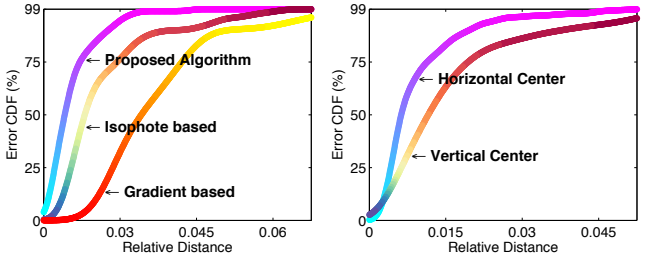
Fig. 11: (a), (b) and (c) show the angle span of u_{body}^{navi}, u_{body}^{face}, and u_{face}^{navi} in an single test. The points are colored in sampling order. (d) shows the total error CDF cumulated from all tests and all participants.

1) Accuracy of ASM-based Face Tracking: To evaluate the accuracy of tracking facial features, a volunteer rotates his/her head arbitrarily. A smartphone placed in front continuously tracks his/her facial landmarks. The per-frame ground truth is obtained via the ERT approach [21].

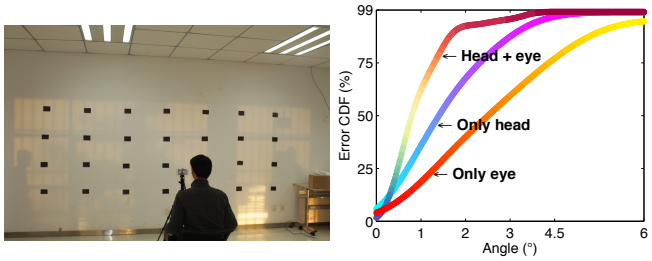
We first evaluate the accuracy of ASM-based f acial anchor nodes tracking. Fig. 10 shows the error surface after smoothing. Horizontally, the error distribution is symmetric. The error is small when the azimuth angle is within $\pm 20°$, and it grows quickly when beyond $\pm 35°$. However, the error distribution is not symmetric vertically. The error grows much faster in negative elevation angle, as the user lowers his/her head. It is because the nose edge feature points may lose tracking when the user lowers his/her head. In real applications, the phone is usually held in a lower position, *i.e.* in a positive elevation angle, and the horizontal viewing angle is usually within $\pm 20°$. Obviously, the ASM-based tracking can provide highly accurate tracking on the facial features.

2) Accuracy of Face Pose Estimation: Next we evaluate the accuracy of face pose estimation. However, a challenge arises: there is no ground-truth data of the face pose.

To solve this problem, we propose an indirect evaluation method. One volunteer, denoted as PA, keeps his head still, while another volunteer randomly moves a smartphone in front of PA to capture his/her face. Since PA keeps the head still, the estimated facing direction R_{face}^{navi} should be stable too. We assume the error in R_{body}^{navi} is much smaller than $R_{face}^{c_f}$. Therefore, the error in u_{face}^{navi} reflects the error in the face



(a) CDF of Iris Center Loc. (b) Anisotropy in localization



(c) Gaze estimation setup (d) CDF of gaze estimation

Fig. 12: (a) the CDF comparison of iris center localization algorithms. (b) the slight error distribution difference in different direction. (c) gaze estimation setup. (d) the CDF of gaze estimation error.

pose computation. Fig. 11 shows the results of an example. In particular, Fig. 11 (a) shows the smartphone's facing direction u_{body}^{navi} during the random movement. Fig. 11 (b) shows the corresponding face pose *w.r.t.* front camera u_{face}^{cf}. Fig. 11 (c) shows the slight error in u_{face}^{navi}.

Each volunteer conducts the same test. Figure. 11 (d) shows the error CDF in estimating their face poses. We split the errors into the azimuth and elevation direction. In more than 80% experiments, the error is within $3°$ for both azimuth and elevation. We observe that there is slight deviation in these two directions. This is mainly caused by the anisotropic error distribution when tracking the facial features.

B. Accuracy of iris Center Localization

To evaluate iris center localization, 3 short videos are recorded for every volunteer. In each video, 50 near sclera images are randomly selected. The ground truth of iris center is manually annotated. Two state-of-the-art iris center localization algorithms, namely the gradient-based [28] and isophote-based [29] algorithms, are also implemented as comparison.

Fig. 12 (a) shows the overall error CDF of the three approaches. It is clear that VADS outperforms other two approaches in terms of accuracy and robustness. Fig. 12 (b) presents the error distribution in horizontal and vertical directions. From the figure, we find that the error in the vertical direction is much larger than that in the horizontal direction. This is because the eyelid occlusion in vertical direction poses a significant impact on the accuracy of iris center localization.

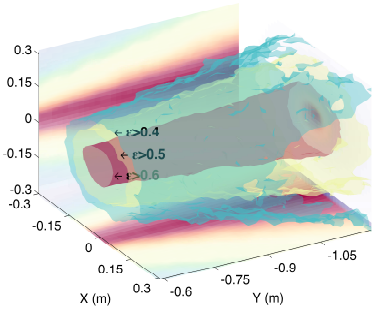


Fig. 13: VADS Ranging error by moving the face in a 3D space.

C. Accuracy of Gaze Direction Estimation

To check the accuracy of gaze direction computation, we setup an experiment testbed as shown in Fig. 12 (c). The smartphone is placed in front of the volunteer. Each volunteer stares at the landmarks on the wall in sequence, 3 seconds for each. The experiment repeats in three groups. In the first group, the volunteer only moves his/her eyes for staring while his head keeps still. In the second group, the gaze action is mainly via the head rotation. Note that the eyes keep unmoved in this round. In the last group, volunteers are encouraged to move their heads and eyes together.

Fig. 12 (d) shows the error CDF. The error of all three groups is small. The eye-only moving group has the largest error, around $3°$ in average. VADS achieves the minimum error, $1.3°$ in average, when moving both the head and eyes for gazing at some object. The error is higher in the case in the head-only or eyes-only groups. The reason is that if both of the head and eyes rotate, they will be complementary to each other to achieve a higher accuracy with a relatively small rotation, compared to the head-only or eyes-only moving.

D. Accuracy of Object Localization

In this subsection, we evaluate the distance computation under variant face positions and directions. In the experiments, a intended object ob is placed $5m$ behind the smartphone. Volunteers move and rotate their face sufficiently and freely in front of the smartphone, while keeping gazing at the object. The moving range is within the following 3D space: $[-0.3\text{ m}, 0.3\text{ m}]$ in the x-axis, $[-0.6\text{ m}, -1.2\text{ m}]$ in the y-axis, and $[-0.3\text{ m}, 0.3\text{ m}]$ in the z-axis.

Fig. 13 visualizes the distance error by varying the face position in the 3D space, where three levels of ranging error, denoted as ϵ, are shown, *i.e.* $\epsilon > 0.4$ m, $\epsilon > 0.5$ m, and $\epsilon > 0.6$ m. The figure shows that the error increases rapidly when the head is close to the optic axis. In this case, $\angle ob$ is too small to have stable measurements. We also observe that the distance between the user and smartphone in the y-axis, *i.e.* $|t_{face}^{cf}|$, has a small influence on the distance accuracy. This is because the ranging error mainly comes from gaze direction estimation, which is not significantly affected by the distance.

We conduct another set of experiments to evaluate the ranging accuracy of distance computation. In these experiments,

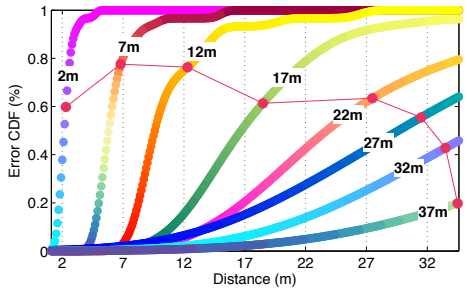


Fig. 14: The red dots are final estimated value. The label on each curve is the groundtruth distance.

each time the intended object is moved 0.5 meter further away from a volunteer, and the volunteer stands at the same position to measure the distance.

We show the CDF of measured distances in Fig. 14, The red dots are the average values that are used as the reported results of VADS. We see that, although the distribution covers a wide range, the reported results have low errors, especially when the distance is less than 20m. For example, the error is about 0.15m for the 12m test, and less than 2m for the 17m test.

VIII. Conclusion

In this paper, we present VADS, a smartphone-based visual attention detection system. It enables computation of both the gaze direction and distance towards the intended object. A series of computer vision techniques are proposed to achieve this goal on smartphone platforms. Extensive evaluation results demonstrate the high accuracy of VADS.

Acknowledgment

This work is sponsored by the National Natural Science Foundation of China (NSFC) under Grant No. 61325013, 61572396, 61402359, 61190112, and 61473109. Chen Qian is sponsored by National Science Foundation grant CNS-1464335 and University of Kentucky College of Engineering Startup Grant.

References

[1] J.-D. Haynes and G. Rees, "Decoding Mental States from Brain Activity in Humans," *Nature Reviews Neuroscience*, vol. 7, no. 7, pp. 523–534, 2006.

[2] G. Wang, Y. Zou, Z. Zhou, K. Wu, and L. M. Ni, "We Can Hear You with Wi-Fi!" in *Proc. ACM MobiCom*, 2014, pp. 593–604.

[3] K. Ali, A. X. Liu, W. Wang, and M. Shahzad, "Keystroke Recognition using Wi-Fi Signals," in *Proc. ACM MobiCom*, 2015, pp. 90–102.

[4] Z. Li, Y. Xie, M. Li, and K. Jamieson, "Recitation: Rehearsing Wireless Packet Reception in Software," in *Proc. ACM MobiCom*, 2015, pp. 291–303.

[5] Y. Xie, Z. Li, and M. Li, "Precise Power Delay Profiling with Commodity Wi-Fi," in *Proc. ACM MobiCom*, 2015, pp. 53–64.

[6] L. Zhang, X.-Y. Li, W. Huang, K. Liu, S. Zong, X. Jian, P. Feng, T. Jung, and Y. Liu, "It Starts with iGaze: Visual Attention Driven Networking with Smart Glasses," in *Proc. ACM MobiCom*, 2014, pp. 91–102.

[7] A. Mayberry, P. Hu, B. Marlin, C. Salthouse, and D. Ganesan, "iShadow: Design of a Wearable, Real-Time Mobile Gaze Tracker," in *Proc. ACM MobiSys*, 2014, pp. 82–94.

[8] W. Zwaag, A. Schäfer, J. P. Marques, R. Turner, and R. Trampel, "Recent Applications of UHF-MRI in The Study of Human Brain Function and Structure: a Review," *NMR in Biomedicine*, 2015.

[9] S. Uithol, D. C. Burnston, and P. Haselager, "Why We May not Find Intentions in the Brain," *Neuropsychologia*, vol. 56, pp. 129–139, 2014.

[10] Y. Jang, R. Mallipeddi, and M. Lee, "Identification of Human Implicit Visual Search Intention based on Eye Movement and Pupillary Analysis," *User Modeling and User-Adapted Interaction*, 2014.

[11] Y.-C. Chen and S.-L. Yeh, "Look into My Eyes and I Will See You: Unconscious Processing of Human Gaze," *Consciousness and cognition*, vol. 21, no. 4, pp. 1703–1710, 2012.

[12] R. Valenti, N. Sebe, and T. Gevers, "Combining Head Pose and Eye Location Information for Gaze Estimation," *Image Processing, IEEE Transactions on*, vol. 21, no. 2, pp. 802–815, 2012.

[13] A. Uhl and P. Wild, "Multi-Stage Visible Wavelength and Near Infrared Iris Segmentation Framework," *Image Analysis and Recognition*, pp. 1–10, 2012.

[14] F. Lu, Y. Sugano, T. Okabe, and Y. Sato, "Head Pose-Free Appearance-Based Gaze Sensing via Eye Image Synthesis," in *Proc. IEEE ICPR*, 2012, pp. 1008–1011.

[15] Y. Sugano, Y. Matsushita, and Y. Sato, "Appearance-based Gaze Estimation using Visual Saliency," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 2, pp. 329–341, 2013.

[16] J. G. Manweiler, P. Jain, and R. Roy Choudhury, "Satellites in Our Pockets: an Object Positioning System using Smartphones," in *Proc. ACM MobiSys*, 2012, pp. 211–224.

[17] L. Shangguan, Z. Zhou, Z. Yang, K. Liu, Z. Li, X. Zhao, and Y. Liu, "Towards Accurate Object Localization with Smartphones," *IEEE Transactions on Parallel and Distributed System*, vol. 25, no. 10, pp. 2731–2742, 2014.

[18] Z. Kalal, K. Mikolajczyk, and J. Matas, "Forward-Backward Error: Automatic Detection of Tracking Failures," in *Proc. IEEE ICPR*, 2010, pp. 2756–2759.

[19] I. Matthews and S. Baker, "Active Appearance Models Revisited," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 135–164, 2004.

[20] P. Sauer, T. F. Cootes, and C. J. Taylor, "Accurate Regression Procedures for Active Appearance Models," in *Proc. BMVC*, 2011, pp. 1–11.

[21] V. Kazemi and J. Sullivan, "One Millisecond Face Alignment with an Ensemble of Regression Trees," in *Proc. IEEE CVPR*, 2014, pp. 1867–1874.

[22] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active Shape Models-Their Training and Application," *Elsevier CVIU*, vol. 61, no. 1, pp. 38–59, 1995.

[23] S. Milborrow and F. Nicolls, "Locating Facial Features with an Extended Active Shape Model," in *Proc. ECCV*, 2008, pp. 504–513.

[24] K. Seshadri and M. Savvides, "Robust Modified Active Shape Model for Automatic Facial Landmark Annotation of Frontal Faces," in *Proc. IEEE BTAS*, 2009, pp. 1–8.

[25] C. Wu, "Towards Linear-Time Incremental Structure from Motion," in *Proc. IEEE 3DV*, 2013, pp. 127–134.

[26] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: An Accurate O(n) Solution to the PnP Problem," *International Journal of Computer Vision*, vol. 81, no. 2, pp. 155–166, 2009.

[27] Y. Ito, W. Ohyama, T. Wakabayashi, and F. Kimura, "Detection of Eyes by Circular Hough Transform and Histogram of Gradient," in *Proc. IEEE ICPR*, 2012, pp. 1795–1798.

[28] F. Timm and E. Barth, "Accurate Eye Centre Localisation by Means of Gradients," in *Proc. VISAPP*, 2011, pp. 125–130.

[29] V. Roberto and G. Theo, "Accurate Eye Center Location Through Invariant Isocentric Patterns," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 9, pp. 1785–1798, 2012.

[30] D. W. Hansen and Q. Ji, "In the Eye of the Beholder: A Survey of Models for Eyes and Gaze," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 3, pp. 478–500, 2010.

[31] O. Faugeras, *Three-Dimensional Computer Vision: a Geometric Viewpoint*. MIT press, 1993.

[32] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-Learning-Detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 7, pp. 1409–1422, 2012.

[33] "Opencv Library," http://www.opencv.org.

[34] "Sony Lens-Style Camera," http://www.sony.co.uk/electronics/attachable-lens/t/lens-style-cameras.