What are you looking at? Automatic estimation and inference of gaze

Valenti, R.

Publication date
2011

5

Improving Visual Gaze Estimation by Saliency

5.1 Introduction

Visual gaze estimation is the process which determines the 3D line of sight of a person in order to analyze the location of interest. The estimation of the direction or the location of interest of a user is key for many applications, spanning from gaze based HCI, advertisement [96], human cognitive state analysis, attentive interfaces (*e.g.* gaze controlled mouse) to human behavior analysis.

Gaze direction can also provide high-level semantic cues such as who is speaking to whom, information on non verbal communications (*e.g.* interest, pointing with the head/with the eyes) and the mental state/attention of a user (*e.g.* a driver). Overall, visual gaze estimation is important to understand someone's attention, motivation and intentions [44].

Typically, the pipeline of estimating visual gaze mainly consists of two steps (see Figure 5.1): (1) analyze and transform pixel based image features obtained by sensory information (devices) to a higher level representation (*e.g.* the position of the head or the location of the eyes) and (2) map these features to estimate the visual gaze vector (line of sight), hence finding the area of interest in the scene.

There is an abundance of research in the literature concerning the first component of the pipeline, which principally covers methods to estimate the head

[0]R. Valenti, N.Sebe, and T. Gevers, "What are you looking at? Improving Visual Gaze Estimation by Saliency", Pending revision in International Journal on Computer Vision, 2011.

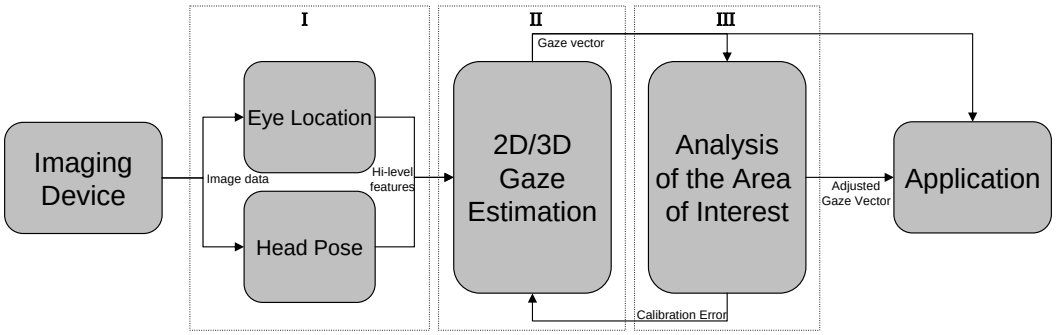


Figure 5.1: The visual gaze estimation pipeline, extended as proposed in this chapter.

position and the eye location, as they are both contributing factors to the final estimation of the visual gaze [66]. Nowadays, commercial eye gaze trackers are one of the most successful visual gaze devices. However, to achieve good detection accuracy, they have the drawback of using intrusive or expensive sensors (pointed infrared cameras) which cannot be used in daylight and often limit the possible movement of the head, or require the user to wear the device [9]. Therefore, recently, eye center locators based solely on appearance are proposed [25, 61, 109] which are reaching reasonable accuracy in order to roughly estimate the area of attention on a screen in the second step of the pipeline. A recent survey [44] discusses the different methodologies to obtain the eye location information through video-based devices. Some of the methods can be also used to estimate the face location and the head pose in geometric head pose estimation methods. Other methods in this category track the appearance between video frames, or treat the problem as an image classification one, often interpolating the results between known poses. The survey collected by [82] gives a good overview of appearance based head pose estimation methods.

Once the correct features are determined using one of the methods and devices discussed above, the second step in gaze estimation (see Figure 5.1) is to map the obtained information to the 3D scene in front of the user. In eye gaze trackers, this is often achieved by direct mapping of the eye center position to the screen location. This requires the system to be calibrated and often limits the possible position of the user (*e.g.* using chinrests). In case of 3D visual gaze estimation, this often requires the intrinsic camera parameters to be known. Failure to correctly calibrate or comply to the restrictions of the gaze estimation device may result in wrong estimations of the gaze.

In this chapter, we propose to add a third component in the visual gaze estimation pipeline, which has not been addressed in the literature before: the analysis of the area of interest. When answering the question "what am I looking

at?", the visual gaze vector can be resolved from a combination of body/head pose and eyes location. As this is a rough estimation, the obtained gaze line is then followed until an uncertain location in the gazed area. In our proposed framework, the gaze vector will be steered to the most probable (salient) object which is close to the previously estimated point of interest. In the literature, it is argued that that salient objects might attract eye fixations [97, 32], and this property is extensively used in the literature to create saliency maps (probability maps which represent the likelihood of receiving an eye fixation) to automate the generation of fixation maps [55, 86]. In fact, it is argued that predicts where interesting parts of the scene are, therefore is trying to predict where a person would look. However, now that accurate saliency algorithms are available [110, 51, 72, 68], we want to investigate whether saliency could be used to adjust uncertain fixations. Therefore, we propose that gaze estimation devices and algorithms should take the gazed scene into account to refine the gaze estimate, in a way which resembles the way humans resolve the same uncertainty.

In our system, the gaze vector obtained by an existing visual gaze estimation system is used to estimate the foveated area on the scene. The size of this foveated area will depend on the device errors and on the scenario (as will be explained in Section 5.2). This area is evaluated for salient regions using the method described in Section 5.3, and filtered so that salient regions which are far away from the center of the fovea will be less relevant for the final estimation. The obtained probability landscape is then explored to find the best candidate for the location of the adjusted fixation. This process is repeated for every estimated fixation in the image. After all the fixations and respective adjustments are obtained, the least-square error between them is minimized in order to find the best transformation from the estimated sets of fixations to the adjusted ones. This transformation is then applied to the original fixations and future ones, in order to compensate for the found device error.

The novelty in this chapter is the proposed third component of the visual gaze estimation pipeline, which uses information about the scene to correct the estimated gaze vector. Therefore, the contributions are the following:

- We propose a method to improve visual gaze estimation systems.
- When a sequence of estimations is available, the obtained improvement is used to correct the previously erroneous estimates. In this way, the proposed method allows to re-calibrate the tracking device if the error is constant.
- We propose to use the found error to adjust and recalibrate the gaze estimation devices at runtime, in order to improve future estimations.

- The method is used to fix the shortcoming of low quality monocular head and eye trackers improving their overall accuracy.

The rest of the chapter is structured as follows. In the next section, we describe the errors affecting visual gaze estimation. In Sections 5.3 and 5.4, the methodology used to extract the salient regions and to correct the fixation points is discussed.

In Section 5.5, the procedure and the scenarios used for the experiments are described. Section 5.6 discusses the obtained results. After some additional discussion on the findings is Section 5.7, the conclusions are given in Section 5.8.

5.2 Device Errors, Calibration Errors, Foveating Errors

Visual gaze estimators have inherent errors which may occur in each of the components of the visual gaze pipeline. In this section, we describe these errors, to derive the size of the area where we should look for interesting locations. To this end, we identify three errors which should be taken into account when estimating visual gaze (one for each of the components of the pipeline): the device error, the calibration error and the foveating error. Depending on the scenario, the actual size of the area of interest will be computed by cumulating these three errors (ϵ_{total}) and mapping them to the distance of the gazed scene.

5.2.1 The device error ϵ_d

This error is attributed to the first component of the visual gaze estimation pipeline. As imaging devices are limited in resolution, there are a discrete number of states in which image features can be detected and recognized. The variables defining this error are often the maximum level of details which the device can achieve while interpreting pixels as the location of the eye or the position of the head. Therefore, this error mainly depends on the scenario (*e.g.* the distance of the subject from the imaging device, more on this on Section 5.5) and on the device that is being used.

5.2.2 The calibration error ϵ_c

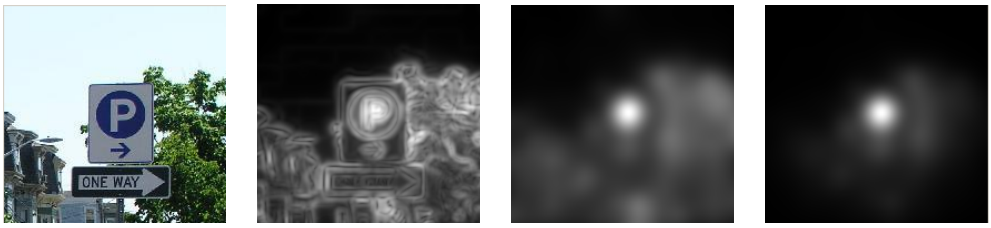
This error is attributed to the resolution of the visual gaze starting from the features extracted in the first component. Eye gaze trackers often use a mapping between the position of the eye and the corresponding locations on the screen. Therefore, the tracking system needs to be calibrated. In case the subject moves from his original location, this mapping will be inconsistent and the system may erroneously estimate the visual gaze. Chinrests are often required in these situations to limit the movements of the users to a minimum. Muscular distress, the length of the session, the tiredness of the subject, all may influence the calibration error. As the calibration error cannot be known a priori, it cannot be modeled. Therefore, the aim is to isolate it from the other errors so that it can be estimated and compensated (Section 5.4).

5.2.3 The foveating error ϵ_f

As this error is associated with the new component proposed in the pipeline, it is required to analyze the properties of the fovea to define it. The fovea is the part of the retina responsible for accurate central vision in the direction in which it is pointed. It is necessary to perform any activities which require a high level of visual details. The human fovea has a diameter of about $1.0mm$ with a high concentration of cone photoreceptors which account for the high visual acuity capability. Through saccades (more than 10,000 per hour according to [37]), the fovea is moved to the regions of interest, generating eye fixations. In fact, if the gazed object is large, the eyes constantly shift their gaze to subsequently bring images into the fovea. For this reason, fixations obtained by analyzing the location of the center of the cornea are widely used in the literature as an indication of the gaze and interest of the user.

However, it is generally assumed that the fixation obtained by analyzing the center of the cornea corresponds to the exact location of interest. While this is a valid assumption in most scenarios, the size of the fovea actually permits to see the central two degrees of the visual field. For instance, when reading a text, humans do not fixate on each of the letters, but one fixation permits to read and see the multiple words at once.

Another important aspect to be taken into account is the decrease in visual resolution as we move away from the center of the fovea. The fovea is surrounded by the parafovea belt which extends up to $1.25mm$ away from the center, followed by the perifovea ($2.75mm$ away), which in turn is surrounded by a larger area that delivers low resolution information. Starting at the outskirts of the



(a) (b) (c) (d)

Figure 5.2: (a) An example image; (b) the saliency map of the image obtained as in [110]; (c) The saliency map used in the proposed method. The latter displays less local maxima and retains more energy towards the center of image structures, therefore is fit for our purposes. (d) is the saliency map filtered by the Gaussian kernel modeling the fovea decrease in resolution.

fovea, the density of receptors progressively decreases, hence the visual resolution decreases rapidly as it goes far away from the foveal center [91]. We model this by using a Gaussian kernel centered on the foveated area, with standard deviation as a quarter of the estimated foveated area. In this way, areas which are close to the border of the foveated area are of lesser importance. In our model, we consider this region as the possible location for the interest point. As we are going to increase the foveated area by the projection of ϵ_{total}, the tail of the Gaussian of the foveated area will aid to balance the importance of a fixation point against the distance from the original fixation point (Figure 5.2(d)). As the point of interest could be anywhere in this limited area, the next step it to use saliency to extract potential fixation candidates.

5.3 Determination of salient objects in the foveated area

The saliency is evaluated on the interest area by using a customized version of the saliency framework proposed by [110]. The framework uses isophote curvature to extract the displacement vectors, which indicate the center of the osculating circle at each point of the image. In Cartesian coordinates, the isophote curvature κ is defined as:

$$\kappa = -\frac{L_y^2 L_{xx} - 2L_x L_{xy} L_y + L_x^2 L_{yy}}{(L_x^2 + L_y^2)^{3/2}}.$$

Where L_x represent the first order derivative of the luminance function in the x direction, L_{xx} the second order derivative on the x direction, and so on. The

isophote curvature is used to estimate points which are closer to the center of the structure it belongs to, therefore the isophote curvature is inverted and multiplied by the gradient. The displacement coordinates $D(x, y)$ to the estimated centers are then obtained by:

$$D(x,y) = -\frac{\{L_x, L_y\}(L_x^2 + L_y^2)}{L_y^2 L_{xx} - 2L_x L_{xy} L_y + L_x^2 L_{yy}}.$$

In this way every pixel in the image gives an estimate of the potential structure it belongs to. To collect and reinforce this information and to deduce the location of the objects, $D(x, y)$'s are mapped into an accumulator, weighted according to their local importance defined as the amount of image curvature and color edges. The accumulator is then convolved with a Gaussian kernel so that each cluster of votes will form a single estimate. This clustering of votes in the accumulator gives an indication of where the centers of interesting or structured objects are in the image.

The method discussed in [110] uses multiple scales. Here, since the scale is directly related to the size of the foveated area, the optimal scale can be determined once and then linked to the foveated area itself. Furthermore, in [110], the color and curvature information to the final saliency map is added, while here this information is discarded. The reasoning behind this choice is that this information is mainly useful to enhance objects on their edges, while the isocentric saliency is fit to locate the adjusted fixations closer to the center of the objects, rather than on their edges. Figure 5.2 shows the difference between the saliency map obtained by the framework proposed in [110] and the single isocentric-only saliency map used here. While removing this information from the saliency map might reduce the overall response of salient objects in the scene, it brings the ability to use the saliency maps as smooth probability density functions.

5.4 Adjustment of the Fixation Points and Resolution of the Calibration Error

Once the saliency of the foveated region is obtained, it is masked by the foveated area model as defined in Section 5.2. Hence, the Gaussian kernel in the middle of the foveated area will aid in suppressing saliency peaks in its outskirts. However, there may still be uncertainties about multiple optimal fixation candidates.

Algorithm 3 Pseudo-code of the proposed system

Initialize scenario parameters
- Assume $\epsilon_c = 0$
- Calculate the $\epsilon_{total} = \epsilon_f + \epsilon_d + \epsilon_c$
- Calculate the size of the foveated area by projecting ϵ_{total} at distance d as $\tan \epsilon_{total} * d$

for each new fixation point p **do**
- Retrieve the estimated gaze point by the device
- Extract the foveated area around each the fixation p
- Inspect the foveated area for salient objects.
- Filter the result by the Gaussian kernel
- Initialize a meanshift window on the center of the foveated area

while maximum iterations not reached or $\Delta p <$ threshold **do**
climb the distribution to the point of maximum energy
end while

- Select the saliency peak closest to the center of the converged meanshift window as being the correct adjusted fixation.
- Store the original fixation and the adjusted fixation, with weight w found on the same location on the saliency map
- Calculate the weighted least-squares solution between all the stored points to derive the transformation matrix T
- Transform all original fixations with the obtained transformation matrix
- Use the transformation T to compensate the calibration error in the device

end for

Therefore, a meanshift window with a size corresponding to the standard deviation of the Gaussian kernel is initialized on the location of the estimated fixation point (corresponding to the center of the foveated region). The meanshift algorithm will then iterate from that point towards the point of highest energy. After convergence, the closest saliency peak on the foveated image is selected as the new (adjusted) fixation point. This process is repeated for all fixation points on an image, obtaining a set of corrections. We suggest that an analysis of a number of these corrections holds information about the overall calibration error. This allows for estimation of the current calibration error of the gaze estimation system which thereafter can be used to compensate it. The highest peaks in the saliency maps are used to align fixation points with the salient points discovered in the foveated areas.

A weighted least-squares error minimization between the estimated gaze locations and the corrected ones is performed. In this way, the affine transformation

matrix T is derived. The weight is retrieved as the confidence of the adjustment, which considers both the distance from the original fixation and the saliency value sampled on the same location. The obtained transformation matrix T is thereafter applied to the original fixations to obtain the final fixation estimates. We suggest that these new fixations should have minimized the calibration error ϵ_c. Note that here we assume that the non linearity of the eye anatomy and the difference between the visual axis and the optical axis are already modeled and compensated on the second step of the gaze estimation pipeline. In fact, we argue that the adjustments of the gaze estimates should be affine, as the calibration error mainly shifts or scales the gazed locations on the gazed plane.

The pseudo code of the proposed system is given in Algorithm 3.

5.5 Evaluation

To test our claims, we tested the approach on three different visual gaze estimation scenarios: (1) using data from a commercial eye gaze tracker, (2) using a webcam based eye gaze tracker and (3) using a webcam based head pose estimator. The used measure, the dataset descriptions, the experimental settings and the size of the foveated areas for each of the scenarios are discussed in this section.

5.5.1 Measure and Procedure

The most common evaluation method for gaze estimation algorithms consists in asking the subjects to look at known locations on a screen, indicated by markers. Unfortunately, this evaluation cannot be performed on the proposed method: as the markers are salient by definition, this evaluation method will not yield reliable results. This is because the fixations falling close to the markers would automatically be adjusted to their center, suggesting a gaze estimation accuracy close or equal to 100%. Since this traditional experiment would over-estimate the validity of the approach, it is necessary to use a different kind of experimental setup, which makes use of real images. The problem, in this case, is the acquisition of the ground truth.

When building fixation maps from human fixations, it is commonly assumed that by collecting the fixation from all users into an accumulator and by convolving it with a Gaussian kernel has the effect of averaging out outliers, yielding high values to interesting (*e.g.* salient) locations. By choosing a Gaussian

kernel with the same size as the computed foveated area, we suggest that this process should average out the calibration errors of each user. More specifically, one subject might have a systematic calibration error to the right, another one to the left, another one to the top etc. We argue that by averaging all the fixations together it is possible to create a calibration error free saliency/fixation map.

Under this assumption, it is possible to evaluate our approach in a rather simple manner. If, after the proposed gaze correction, the fixation points of a subject are closer to the peaks of the calibration free fixation map, then the method improved the fixation correlation between the current subject and all the others. Hence, the proposed method helped in reducing the calibration error for the given subject.

Therefore, in our experimentation, all the fixations (except the one for the subject that is being evaluated) are cumulated into a single fixation map. The fixation map is then convolved with a Gaussian kernel with the same standard deviation as used in the foveated area, merging fixations which are close to each other. This maps contains

The fixation map F is then sampled at the location of the i^{th} fixation f_i of the excluded subject. To obtain values which are comparable, the value of each sampled fixation is divided by the maximum value in the fixation map ($\max(F)$). The final value of the measure is the average of the sampled value at each fixation point:

$$C_s = \frac{1}{n} \sum_{i=0}^{n} \frac{F(f_i)}{\max(F)}$$

The returned value indicates a correlation between the subject's fixations and all the others (*e.g.* how many other subject had a fixation around the subject's fixations), it can be evaluated locally for each fixation, and it provides values which are comparable even when only one fixation is available. Note that proposed experimentation procedure considers the size of the foveated area, is independent of the number of available fixations and measures the agreement with the fixations of all other subjects. Hence, we believe that the described experimentation procedure is a sound validation for the proposed method.

To better understand the rationale behind the proposed evaluation procedure, let us use a comparison with a real world example. We compare the noisy gaze estimates to inaccurate GPS information. In modern navigation systems, the noisy GPS information (in our case the raw gaze estimates) is commonly adjusted to fit known street information (*i.e.* the ground truth). If we do not have the street information (*i.e.* the real gazed locations), we argue that it is possible

reconstruct it by collecting raw GPS information of cars which are freely roaming the streets (*i.e.* the fixations of all the subjects). Averaging this information will give a good indication of the street locations (*i.e.* by averaging the raw fixations in the fixation map, we obtain the ground truth of the important objects in the scene). In our case we will evaluate whether the adjustment proposed by our system will bring the raw information closer to the ground truth obtained by averaging raw information.

5.5.2 Commercial Eye Gaze Tracker

For this experiment, the eye gaze tracking dataset by [55] is used. The dataset consists of fixations obtained from 15 subjects on 1003 images, using a commercial eye tracker. As indicated in [55] the fixations in this dataset are biased towards the center of an image. This is often the case as typically the image is shot by a person so that the subject of interest is in the middle of it. Therefore, we want to verify if the used measure increase if, instead of looking at the center of the image, we use the fixation points of a subject versus the fixation point of all other subjects. The parameters for this experiment are the following. As the subjects are sitting at a distance of $750mm$, the projection of $\epsilon_f = 2.0°$ corresponds to $26.2mm$. ϵ_d is usually claimed to be $0.5°$. While this is a nominal error, this corresponds to only $6.5mm$ on the screen, which is highly unrealistic. In screen resolution, the projection of $\epsilon_{total} = 2.5°$ is $32.7mm$, which approximately corresponds to 115 pixels.

5.5.3 Webcam Based Eye Gaze Tracker

For this experiment, the eye locator proposed by [109] is used, which makes use of standard webcam (without IR) to estimate the location of both eye centers. Starting from the position of the eyes, a 2D mapping is constructed as suggested by [123], which sacrifices some accuracy to assume a linear mapping between the position of the eyes and the gazed location on the screen. The user needs to perform a calibration procedure by looking at several known points on the screen. A 2D linear mapping is then constructed from the vector between the eye corners and the iris center and recorded at the known position on the screen. This vector is then used to interpolate between the known screen locations. For example, if we have two calibration points P_1 and P_2 with screen coordinates α and β, and eye-center vector (with the center of the images as the anchor point) x and y, we can interpolate a new reading of the eye-center vector to obtain the

screen coordinates by using the following linear interpolant:

$$\alpha = \alpha_1 + \frac{x - x_1}{x_2 - x_1}(\alpha_2 - \alpha_1),$$

$$\beta = \beta_1 + \frac{y - y_1}{y_2 - y_1}(\beta_2 - \beta_1).$$

For the experiment, we asked 15 subjects to look at the first 50 images (in alphabetical order) of the dataset used in the previous experiment. Between each image, the subject is required to look at a dot in the center of the screen. As no chin rest was used during the experiment, this dot is used to calculate an average displacement to the center of the image, which is then used in the next image.

While the projection of ϵ_f is the same as in the previous experiment, the device error ϵ_d is very high, as there are two aspects of the device error that should be taken into consideration:

- The resolution of the device: In our experiments, the calibration shows that the eye shifts of a maximum of 10 pixels horizontally and 8 pixels vertically while looking at the extremes of the screen. Therefore, when looking at a point on the screen with a size of 1280x1024 pixels, there will be an uncertainty window of 128 pixels.
- The detection error: to the previously computed estimate, we should add the possibility of the eye locator to commit a mistake on the eye center location. The system proposed by [109] claims an accuracy close to 100% for the eye center being located within 5% of the interocular distance. With a device resolution of 640x480 pixels and a user distance of $750mm$, the interocular distance measures 85 pixels. Therefore, 5% of the interocular distance of 85 pixels corresponds to 4 pixels, hence to an error of 64 pixels in each direction on the screen. However, since the tracker does not constantly make mistakes, we halved the latter figure, obtaining a foveated region of 160 pixels.

5.5.4 Head Pose Tracker

For this experiment we used a cylindrical 3D head pose tracker algorithm based on Lukas-Kanade optical flow method [119]. The depth of the head, which describes the distance of the head from the screen, is assumed to start from $750mm$ from the camera center. The method assumes a stationary calibrated

camera. The gazed scene is recorded by another camera (also with a resolution of 640x480 pixels) in order to be able to evaluate the saliency of the area. The subjects are required to look at a calibration point in the center of the scene before starting the experiment.

The head pose experiment consists of gazing at different objects in the scene. To keep the affine assumption for the gaze adjustment, the objects were placed in the same plane. The subjects participating in the experiments were requested to gaze at the center of the objects in a fixed sequence, so that the expected ground truth for the gaze location is known. The subjects were instructed to "point with the head", stopping at the center of the called objects. This generates what we call "head fixations", which we evaluate in the same way as we did in the previous experiments. As the ground truth of the head fixations is available, we are also able to estimate the head pose error and check if this can be improved using the found calibration error.

The device error of the used head tracker is $5.26°$ for the vertical direction, and $6.10°$ for the horizontal direction. For simplicity, we fix the device error as the average of the two errors, therefore $\epsilon_d = 5.8°$. Since the objects are placed at distance $d = 2000mm$, this error gives an uncertainty of the estimation of approximately $203.1mm$. The contribution of ϵ_f increases to $69.8mm$. Therefore, the final size of the foveated region will be $272.9mm$. In the scene camera resolution, an object measuring $273mm$ at $2000mm$ distance, appears approximately 80 pixels wide.

5.6 Results

5.6.1 Eye Gaze Tracker

To better understand the improvement obtained by the proposed method over the original fixations, it is necessary to analyze it in the foveated area context. Therefore, we determine the maximum improvement obtainable (upperbound) by selecting the location within the foveated region which yields the maximum value with respect to the fixations of all users. This is computed by looking for the highest value in the fixation map within foveated area, and it indicates which point in the foveated area should be selected by the gaze adjustment method to withhold the maximum possible improvement on the overall correlation. Once this limit is determined, the percentage of improvement can be obtained as the increase towards that limit. Table 5.1 lists the result for each of the subject in the dataset, averaged over all images. Note that the average

Table 5.1: Correlation results for the eye gaze tracker experiment

Subject #	Fixations	Adjusted Fixations	Upperbound	Improvement	# Images Improved
1	33.09	34.49	42.53	14.86%	674/1003
2	28.53	30.33	38.49	18.07%	718/1003
3	34.56	35.82	44.22	13.03%	650/1003
4	32.04	32.95	39.69	11.92%	671/1003
5	32.26	33.94	41.73	17.75%	680/1003
6	37.8	38.9	47.49	11.41%	656/1003
7	32.88	34.24	42.82	13.72%	662/1003
8	25.26	26.9	35.24	16.46%	702/1003
9	29.1	29.77	37.28	8.24%	630/1003
10	38.38	39.65	48.42	12.61%	638/1003
11	32.68	34.24	42.42	16.07%	700/1003
12	35.22	36.91	45.87	15.88%	682/1003
13	38.56	39.4	47.04	9.87%	621/1003
14	36.22	37.28	44.99	12.03%	648/1003
15	31.6	33.4	42.32	16.77%	691/1003
Mean	33.21	34.54	42.70	13.91%	668/1003

correlation of every subject increased by an average of 13.91%, with a minimum improvement of 8.24% and a maximum of 18.07%. This figure is reflected in the amount of images in which the overall correlation improved. In fact, using the proposed method, an average of 668 (out of 1003) images were improved. In comparison, using a random point in the foveated area as the adjusted fixation, only 147 images were improved. An additional test is performed regarding the discussed center bias of human fixations in the dataset. Therefore, we also compare the accuracy obtained by selecting the center of the image as sole fixation. In this case, only 319 images were improved. Therefore, in this scenario, our method outperforms the bias to the center.

5.6.2 Webcam Based Eye Gaze Tracker

The results for this second scenario are listed in Table 5.2. When comparing the original fixations correlation obtained by this system to the one in the previous experiment, it is possible to notice that it is larger. The reason behind this lies in the size of the foveated area which is larger in this experiment than in the previous one. As a consequence, the blurring kernel on the fixation map is larger. Therefore, given the smoothness of the fixation map, less gaps exists between the fixations. Hence, when evaluating a fixation, it is more likely that will hit a tail of a Gaussian of a close fixation. Furthermore, as the eye locator commits mistakes while estimating the center of the eyes, some of the fixations are erroneously recorded, increasing the overall value on uninteresting locations. This effect can be seen in Figure 5.3, which compares the fixation map obtained

(a) (b) (c)

Figure 5.3: (a) The fixation map obtained by the eye gaze tracker; (b) the one obtained by the webcam based tracker; (c) the fixation map obtained by the adjusted webcam based tracker.

Table 5.2: Correlation results for the webcam based eye gaze tracker experiment

Subject #	Fixations	Adjusted Fixations	Upperbound	Improvement	# Images Improved
1	40.22	44.51	49.15	48.04%	41/50
2	41.71	44.44	50.84	29.9%	34/50
3	28.04	35.52	36.71	86.27%	46/50
4	44.81	47.51	53.71	30.34%	34/50
5	47.96	50.48	56.05	31.15%	34/50
6	35.28	40.79	44.41	60.35%	41/50
7	30.98	37.15	39.92	69.02%	43/50
8	41.29	45.94	50.59	50.00%	38/50
9	34.81	38.23	43.26	40.47%	39/50
10	36.28	41.76	45.57	58.99%	37/50
11	32.81	37.28	40.97	54.78%	41/50
12	45.3	47.23	53.53	23.45%	31/50
13	29.51	36.45	38.7	75.52%	41/50
14	36.65	42.02	45.14	63.25 %	43/50
15	32.68	37.1	40.55	56.16%	43/50
Mean	37.22	41.76	45.94	51.85%	39.07/50

by the foveated area of the previous experiment (a) and the one used in this experiment (b) on the same image.

5.6.3 Head Pose Tracker

In this scenario, only one image is available for each subject, that is, the image taken by the scene camera. Note that all objects were placed on the same plane so that the adjustment obtained by the proposed method can still be linear. Table 5.3 shows the mean results between all subjects. Although all the subjects were asked to gaze at the same objects and the subject correlation is expected to be high, the small size of the foveated area gives the fixation map a very small space for improvement. However, the head fixations still improved the subject

(a) (b) (c)

Figure 5.4: (a) The system setup consisting of a "subject camera" (white) and a "scene camera" (black); (b) The displacements (red) between the original location of the "head fixations" (black) and the adjusted fixations (white); (c) The correction of wrongly estimations of the head pose tracker.

Table 5.3: Correlation results for the head pose tracker experiment

Subject #	Fixations	Adjusted Fixations	Upperbound	Improvement	# Subjects Improved
Mean	17.50	18.87	27.27	10.23%	11/15

correlation on 11 subjects out of 15, with an average improvement of 10.23% towards the upperbound. Additionally to the correlation test, in this scenario we analyzed the possibility of adjusting the calibration error of the device. The transformation matrix obtained by our system fed back to the head pose estimator and it is used to adjust the estimated horizontal and vertical angles of the head pose. In our experimentation, using the object location as a ground truth, the tracking accuracy improved by an average of $0.5°$ on the vertical axis and $0.6°$ on the horizontal one. Analyzing the results, we found that while gazing at a certain location, the system would always converge to the closest salient region. This behavior can be seen in Figure 5.4(b), where the clouds of the original fixations (black) are always adjusted (red) to the closest salient object (white). The results of this experiment hint that it is possible to create self-calibrating system which uses known salient locations on the scene to find the right parameters in case the initialization was erroneous. Figure 5.4(c) shows the difference between the pose estimated by the incorrectly initialized head pose tracker (arrow to the right) and the suggested correction (arrow in the center).

5.7 Discussion

The fact that the correlation is improved by 51.85% indicates that it is possible to achieve almost the same accuracy of an (uncorrected) commercial eye tracker. Figure 5.3(c) is an example of this effect. The corrected correlation between 15

subjects is in fact very similar to the one obtained by the eye gaze tracker. Since the system uses saliency, it is important to mention the system could fail when used on subjects which does not have "normal" vision. In fact, if a color-blind person is faced with a color blind test, he might not be able to successfully read the colored text at the center of the image. However, if the subject fixates to the center of the image, the system will probably think that he is looking at the text, and will suggest an erroneous correction. Nonetheless, if other fixations are available, the system might find that the best fit is obtained by not correcting that specific fixation, and might still be able to find the calibration error and improve the overall correlation.

By analyzing the obtained results, we realize where the system breaks down. For instance, when analyzing the fixations on a face, the average fixation (mouth, nose, eye) would have the center of the face as the maximum value for correlation between the subjects. However, if a fixation occur at the center of a face, the most salient regions around it (*e.g.* the eyes, the mouth) will attract the fixation, dropping the correlation. Also, if the foveated region is too big, the fixation will always be attracted by the most salient object in the scene. This might either result in a very good improvement or in a decrease in correlation, as the saliency algorithm might be wrong. Figure 5.5 shows some examples of success and failure of the proposed method. The blue line shows the location of the fixations obtained by the eye gaze tracker, the white line is the suggested adjustment and the black is the final adjustment by the derived transformation matrix. In Figure 5.5 (top-left) it is clear that the subject fixated the sea lion on the right, although the fixation is found in the water. The white line shows the fixations adjusted by the proposed method. The transformation matrix obtained by this adjustment is then used on the original fixation point, obtaining the black line, which now spans between both sea lions. The same argument holds for the face image, where the real fixations were clearly targeted the eyes instead of two undefined points between the eyes and the eyebrows, while the corrected fixations cover both eyes and the mouth. In the images containing text this behavior is more evident, since it is clear that the real fixations were targeted at the text, but the ones recorded by the eye tracker have a clear constant offset, which is fixed by the proposed method. Although the method is shown to bring improvement to 668 pictures in the dataset, there are still 335 cases in which the method fails. This is the case of the bottom-right image in Figure 5.5: while the original fixation ends in an irrelevant location in the sky and the adjusted points span both structures, the transformation matrix obtained by the least-squares minimization is not sufficient to stretch both original fixations to that extent, hence dropping the subject correlation. However, note that this does not happen very often, as the proposed system is still capable of improving the

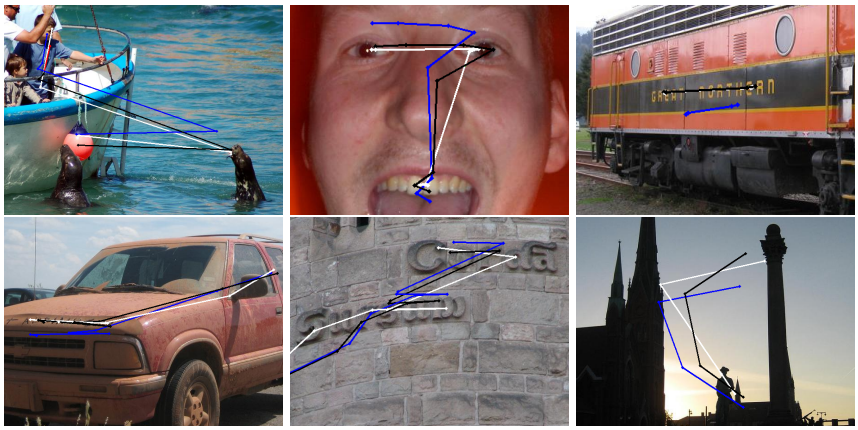


Figure 5.5: Example of success and failure while adjusting the fixations on the eye gaze tracking dataset. The blue line indicates the original fixations, the white line are the fixations corrected by the proposed method, while the black line represent the location of the original fixations transformed by the found calibration error.

correlation with the other subjects in two thirds of the full dataset.

We foresee this method to be used for automatic adjustment of the calibration, and in situations in which the accuracy of the visual gaze estimation device is not enough to clearly distinguish between objects. Furthermore, we foresee the proposed method to pave the way to self-calibrating systems and to contribute in loosening the strict constraints of current visual gaze estimation methods.

5.8 Conclusions

In this chapter, we proposed to add a third step in the visual gaze estimation pipeline, which considers salient parts of the gazed scene in order to compensate for the errors which occurred in the previous steps of the pipeline. The saliency framework is used as a probability density function, so that it can be climbed using the meanshift algorithm. We tested the proposed approach on three different visual gaze estimation scenarios, where we successfully improved the gaze correlation between the subjects. We believe that the proposed method can be used in any existing and future gaze estimation devices to lessen the movement constraints on the users and to compensate for errors coming from an erroneous calibration.