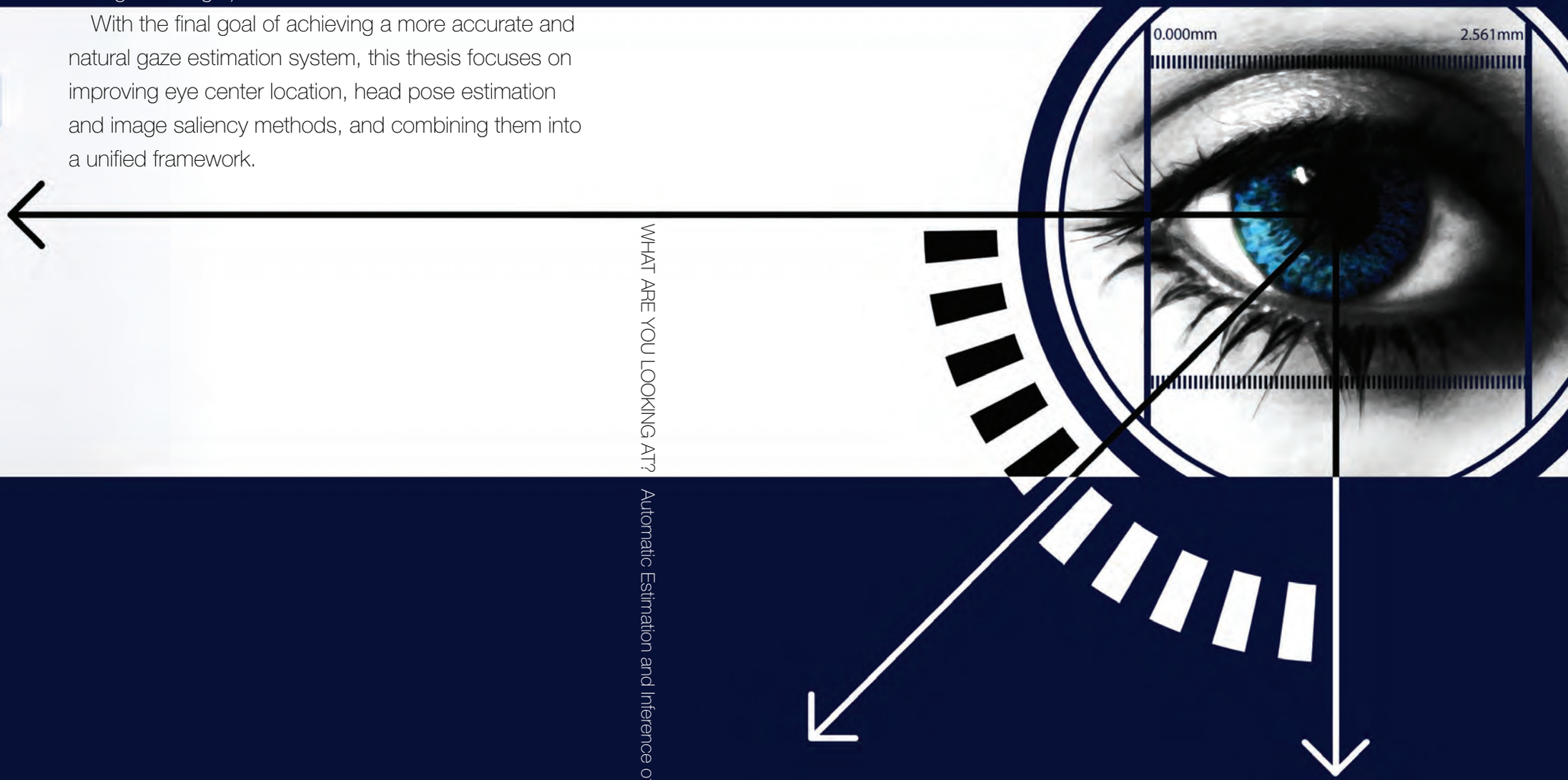
WHAT ARE YOU LOOKING AT?
Automatic Estimation and Inference of Gaze

Roberto Valenti

This thesis is focused on gaze estimation, that is, providing an automated answer to the question “What are you looking at?”. To this end, the problem of gaze estimation is divided into three sub problems, namely detection (i.e. extracting the relevant visual cues from the faces), estimation (i.e. combining them to compute a rough line of interest), and inference (i.e. using information about the gazed scene to infer the most probable gazed target).

With the final goal of achieving a more accurate and natural gaze estimation system, this thesis focuses on improving eye center location, head pose estimation and image saliency methods, and combining them into a unified framework.



WHAT ARE YOU LOOKING AT? Automatic Estimation and Inference of Gaze

ISBN 978-94-6182-046-4

Roberto Valenti

WHAT ARE YOU LOOKING AT?

Automatic Estimation and Inference of Gaze

Roberto Valenti

This book was typeset by the author using LaTeX 2_ε.

Printing: Off Page, Amsterdam

ISBN 978-94-6182-046-4

WHAT ARE YOU LOOKING AT?

Automatic Estimation and Inference of Gaze

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof. dr D. C. van den Boom
ten overstaan van een door het college voor promoties
ingestelde commissie,
in het openbaar te verdedigen in de Agnietenkapel
op woensdag 7 december 2011, te 12:00 uur

door

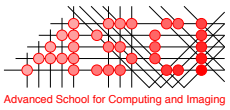
Roberto Valenti

geboren te San Benedetto del Tronto, Italië

Promotiecommissie:

Promotoren:	Prof. dr ir A. W. M. Smeulders Prof. dr T. Gevers
Overige leden:	Prof. dr ir F. C. A. Groen Prof. dr ir B. J. A. Kröse Dr N. Sebe Dr A. A. Salah

Faculteit der Natuurwetenschappen, Wiskunde en Informatica



The work described in this thesis has been carried out within the graduate school ASCI, at the Intelligent Systems Lab Amsterdam of the University of Amsterdam.
ASCI dissertation series number 244.

UNIVERSITEIT VAN AMSTERDAM

"Any intelligent fool can make things bigger,
more complex, and more violent.
It takes a touch of genius – and a lot of courage –
to move in the opposite direction."

Albert Einstein

Contents

Publications

This work is composed by the following papers:

- **Chapter 2**:
  - R. Valenti and T. Gevers, "Accurate Eye Center Location through Invariant Isocentric Patterns", Pending minor revision in IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011.

  Ideas previously appeared in:

  - R. Valenti and T. Gevers, "Accurate Eye Center Location and Tracking using Isophote Curvature", In IEEE Conference on Computer Vision and Pattern Recognition, 2008.
- **Chapter 3**:
  - R. Valenti, N.Sebe, and T. Gevers, "Combining Head Pose and Eye Location Information for Gaze Estimation", IEEE Transactions on Image Processing, 2011.

  Ideas previously appeared in:

  - R. Valenti, A. Lablack, N.Sebe, C. Djeraba, and T. Gevers, "Visual Gaze Estimation by Joint Head and Eye Information". International Conference on Pattern Recognition, 2010.
  - R. Valenti, Z. Yucel and T. Gevers, "Robustifying Eye Center Localization by Head Pose Cues", IEEE Conference on Computer Vision and Pattern Recognition, 2009.
- **Chapter 4**:
  - R. Valenti, N.Sebe, and T. Gevers, "Image Saliency by Isocentric Curvedness and Color", IEEE International Conference on Computer Vision, 2009.

  Ideas previously appeared in:

  - R. Valenti, N.Sebe, and T. Gevers, "Isocentric Color Saliency in Images", IEEE International Conference on Image Processing, 2009.

- **Chapter 5**:
  - 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.

Other papers published during the course of the Ph.D but not included in this work:

- J. Machajdik, J. Stöttinger, E. Danelova, M. Pongratz, L. Kavicky, R. Valenti, A. Hanbury, "Providing Feedback on Emotional Experiences and Decision Making", In IEEE AFRICON, 2011.
- R. Valenti, A. Jaimes, N. Sebe, "Sonify Your Face: Facial Expressions for Sound Generation", In ACM International Conference on Multimedia, 2010.
- H. Dibeklioglu, R. Valenti, A. A. Salah, T. Gevers, "Eyes Do Not Lie: Spontaneous Versus Posed Smiles", In ACM International Conference on Multimedia, 2010.
- H. Joho, I. Arapakis, J. Jose, R. Valenti and N. Sebe, "Exploiting Facial Expressions for Affective Video Summarization", In ACM International Conference on Image and Video Retrieval, 2009.
- R. Valenti, N.Sebe, and T. Gevers, "Simple and Efficient Visual Gaze Estimation", In International Workshop on Multimodal Interactions Analysis of Users in a Controlled Environment, 2008.
- R. Valenti, N.Sebe, and T. Gevers, "Facial Features Matching using a Virtual Structuring Element", In IS&T/SPIE 20th annual Symposium on Electronic Imaging, 2008.
- R. Valenti, N.Sebe, and T. Gevers, "Facial Expression Recognition as a Creative Interface", In International Conference on Intelligent User Interfaces, 2008.
- R. Valenti, N. Sebe, T. Gevers, and I. Cohen, "Machine Learning Techniques for Multimedia", Chapter 7, pages 159-188. Springer, 2008.
- R. Valenti, N.Sebe, and T. Gevers, "Facial Expression Recognition: a Fully Integrated Approach", In International Workshop on Visual and Multimedia Digital Libraries, 2007.
- R. Valenti, N.Sebe, and T. Gevers, "Facial Feature Detection using a Virtual Structuring Element". In Annual Conference of the Advanced School for Computing and Imaging, 2007.

1

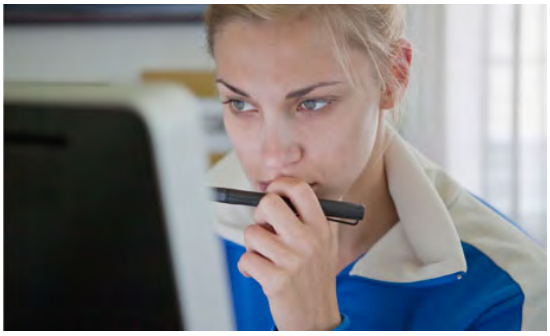
Introduction

As suggested by the title, this body of work is targeted on giving an automatic answer to a seemingly simple question: "What are you looking at?"

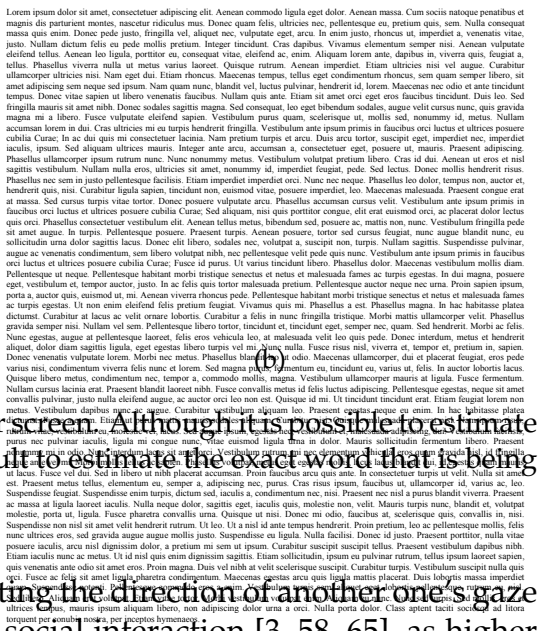
Humans often believe that they are incredibly good at answering this question. To convince the reader that this is a wrong belief, let us consider a simple example scenario in which we want to estimate the gaze point of a person which is sitting in front of a computer screen (Figure 1.1(a)). We are standing in a position in which the face of the person is clearly visible and, on our desk, we have an exact copy of the page that is displayed on the gazed screen (Figure 1.1(b)). When asked "what is this person looking at?", we will already struggle to roughly guess which side of the page is being gazed, while it will be basically impossible for us to name the exact gazed word, let alone the single character contained within it.

Although our gaze estimation capabilities are not as accurate as we would expect, we are still very confident about them. This is because we use them everyday, starting from the earliest cognitive developments when we were toddler.

Developmental psychology, in fact, argues that understanding the gaze of a person is fundamental in infants [14, 93, 15]: By learning how to make sense of the visual channel, infants start with recognizing the face of their caregiver, then learn to follow head movements, and finally learn to follow gaze directions. Later on, recognizing gaze enables an infant to engage in joint visual attention (*i.e.* sharing interest by looking at the same object), which in turn aids the infant in learning social, cognitive and communicative abilities. An example of this is given in [8, 11, 102], which argue that infants are facilitated to learn which utterance correlates to which object by understanding the gaze of the adult. Apart



(a)



(b)

Figure 1.1: (a)A subject looking at a computer screen. Although it is possible to estimate the general gazed area, it is difficult to estimate the exact word that is being gazed (b).

from being useful in early learning, detecting the direction of another one's gaze quickly becomes a crucial component of social interaction [3, 58, 65], as higher level cues about human behavior and intention (such as attention, eye contact, who is speaking to whom, cognitive state and non-verbal communication) can be extrapolated from it. For instance, gaze is a necessary ingredient for understanding the context and the full meaning of a displayed emotion [81, 89]. Examples occur when someone displays a scared face while gazing at a dangerous situation, or displays a happy face while gazing at his beloved one.

Since understanding gaze is fundamental for our development, especially for our human-human interaction, we seek to answer the question "what are you looking at?" in an automatic manner, so that it could be used to achieve more natural human-computer interaction.

1.1 Gaze Estimation and Inference

Two main cues are clearly involved in the estimation of the human's gaze: the position of the head and the location of the eyes relative to it [66]. A number of studies have investigated which of the two cues is more important for gaze estimation. They found that the position of the head is often sufficient to determine the overall direction of attention [23, 34, 77], while the location of the eyes is used to fine tune attention and can hint to an additional layer of infor-

mation regarding thoughts, intentions, desires, and emotions. This is why, in many cultures, eyes are believed to be the "mirror of the soul".

Contrary to these findings, most automatic gaze estimation techniques in computer vision rely either on information about head pose [82] or eye location [44] in order to reduce the complexity of the problem. Furthermore, commercial eye-gaze tracking systems employ a calibration procedure and additional constraints (like restricting head movement or using head mounted cameras) to increase the accuracy and stability of the system. This often involves creating a direct mapping between the location of the eyes and a known position on the screen.

In the example scenario in Figure 1.1, using this technique would imply asking the subject to look at the corners of the text and to record the relative eye locations. Then use these locations as a reference for newly gazed locations. Furthermore, in order to avoid recalibrating at each head location, the subject would not be allowed to move from the calibrated position. Although the accuracy of our estimation would significantly improve with respect to the original rough gaze estimate, we can conclude that this procedure is very different than the way humans achieve the same task. But what is an alternative to this method?

According to [80] an ideal gaze estimator should:

- be accurate, i.e., precise to minutes of arc;
- be reliable, i.e., has constant, repetitive behavior;
- be robust, i.e., should work under different conditions, such as indoors and outdoors, for people with glasses and contact lenses;
- be non-intrusive, i.e., cause no harm or discomfort;
- allow for free head motion;
- not require calibration, i.e., instant set-up;
- have real-time response.

Existing gaze estimation systems forfeit some of these requirements for the sake of accuracy. On the other hand, as shown in the example scenario in Figure 1.1, the human gaze estimation system fulfills all of the requirements, except for high accuracy. Hence, here we want to investigate whether, by weakening the accuracy requirements (and therefore putting more focus on the usability requirements), it could be possible to develop a truly non-intrusive, more accessible and user-friendly gaze estimation system which is similar to the humans'.

Figure 1.2: The famous scene in the movie "Taxi Driver" where the question "Are you talking to me?" is raised.

To this end, in this work we will focus on appearance based methods only. Although they are considered as the less intrusive methods available, appearance based methods tend to be inaccurate. This is mainly due to the reduction of the amount of resolution available to capture both the head and the eye location information in a single frame.

Therefore, we argue that the gaze estimates obtained by the head pose and eye location should only be considered as a rough indication of the direction of interest, and that additional information about the gazed scene needs to be considered to improve accuracy. For instance, in the famous scene from the movie "Taxi Driver", the main character asks himself: "Are you talking to me?" (Figure 1.2), while definitely looking at himself in the mirror. He then turns around to check the rest of the scene. Although it is an acted scene, this behavior clearly hints that taking the context of the gazed scene into consideration is important to completely understand an uncertain gaze estimate. But how can the information about the gaze scene help in gaze estimation? Recalling the scenario in Figure 1.1, if the plain text in Figure 1.1(b) would be replaced by a small red ball on the right side of the image, it would be natural to assume that the subject is gazing at it, and the assumption would probably be correct. Therefore, we argue that the gazed scene needs to be inspected to *infer* the most likely gazed object in the scene to adjust uncertain gaze locations, obtaining an improvement in the accuracy of the gaze estimation system.

1.2 Objectives and Approach

To automatically answer the question "What are you looking at?", in this work we arrive at the following three main research objectives:

- **Detection:**
  In order to fulfill the requirements for an ideal gaze estimator, we need to restrict it to only use appearance information, as it is the sole non-intrusive system which can be used in different conditions. The difficulty here is to find a way to extract, starting from low resolution images of faces, information that is accurate enough to be used for gaze estimation. Furthermore, we need to investigate whether the inaccurate estimations of the head pose and eye location could be used together to reinforce each other in order to yield better overall results.

  The first question we need to answer is how to perform accurate eye center location (*i.e.* within the area of the pupil) on low resolution images (*i.e.* captured by a simple webcam). Accurate eye center location can already be achieved using commercial eye-gaze trackers, but additional constraints (*e.g.* head mounted devices or chin rests) and expensive hardware make these solutions unattractive and impossible to be used on standard (*i.e.* visible wavelength), low-resolution images of eyes. Systems based solely on appearance (*i.e.* not involving active infrared illumination) are present in the literature, but their accuracy does not allow locating and being able to distinguish eye centers movements in these low-resolution settings. The question is discussed in Chapter 2.

  A second question relates to head pose estimation. Head pose and eye location for gaze estimation have been studied separately in numerous works [82, 44]. They show that satisfactory accuracy in head pose and eye location estimation can be achieved in constrained settings. However, due to distorted eye patterns, appearance based eye locators fail to accurately locate the center of the eyes on extreme head poses. In contrast, head pose estimation techniques are able to deal with these extreme conditions, so they may be suited to enhance the accuracy of eye localization. Therefore, in Chapter 3 we consider the question whether a hybrid scheme to combine head pose and eye location information could achieve enhanced gaze estimation.

- **Estimation:**
  Once relevant cues are extracted, they need to be combined in a sensible way to generate a single estimate. However, the low resolution of the

images limits us in the construction of a geometrically accurate eye and head model. Therefore, the aim is to find a way to interpret and combine the obtained information:Instead of trying to geometrically combine the gaze vectors suggested independently by the head pose and by the eye location, in Chapter 3 we want to investigate whether these gaze vector could be combined in a cascade in order to reduce the problem of 3D gaze estimation into subsets of 2D problems.

- **Inference:**
  Finally, our last objective is to investigate an algorithm that, without using prior knowledge about the scene, will efficiently extract likely gaze candidates in the scene. As common salient features as edges, contrast and color only defines local saliency, in Chapter 4 we study whether it is possible to estimate global salient features starting from the local ones, and whether these features help into estimating interesting objects in the scene. To this end, in Chapter 4 we investigate a computational method to infer visual saliency in images.

  In Chapter 5 we will subsequently investigate whether it is possible to adjust the fixations which were roughly estimated by the gaze estimation device. We will test whether the approach can be efficiently applied to different scenarios: using eye tracking data, enhancing a low accuracy webcam based eye tracker, and using a head pose tracker.

Therefore, starting from finding the location of the eyes in a face and ending on finding the most probable object of interest in the gazed scene, the overall focus of this work is to investigate a unified way to improve each single step of the gaze estimation pipeline, in order to achieve a more accurate and natural gaze estimation system.

2

Eye Center Location[1]

2.1 Introduction

As shown by increasing interest on the subject [19, 40, 53], eye center location is an important component in many computer vision applications and research. In fact, the information about the location of the eye center is commonly used in applications as face alignment, face recognition, human-computer interaction, control devices for disabled people, user attention and gaze estimation (*e.g.* driving and marketing) [40, 12]. Eye center location techniques can be divided into three distinct categories which employ different modalities [29]: (1) Electro oculography, which records the electric potential differences of the skin surrounding the ocular cavity; (2) scleral contact lens/search coil, which makes use of a mechanical reference mounted on a contact lens, and (3) photo/video oculography, which uses image processing techniques to locate the center of the eye. The highly accurate eye center information obtained through the mentioned modalities is often used in eye-gaze trackers to map the current position of the eyes to a known plane (*i.e.* a computer screen) as a user's visual gaze estimate. Unfortunately, the common problem of the above techniques is the requirement of intrusive and expensive sensors [9]. In fact, while photo/video oculography is considered the least invasive of the described modalities, commercially available eye-gaze trackers still require the user to be either equipped

[1] R. Valenti and T. Gevers, "Accurate Eye Center Location through Invariant Isocentric Patterns". Pending minor revision in IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011. Ideas previously appeared in: R. Valenti and T. Gevers, "Accurate Eye Center Location and Tracking using Isophote Curvature". In IEEE Conference on Computer Vision and Pattern Recognition, 2008.

Method	Pre-Requirements	Approach	Uses Learning	Used Feature	Used Model/Learning Scheme
Asteriadis [4]	Detected face	Feature Based	-	Edges	Eye model for init + edge crossing count
Jesorsky [52]	Converged face model	Model Based	X	Edges	Hausdorff distance on eye model
Cristinacce [25]	Detected face	Model Based	X	Pixels	PRFR + AAM
Türkan [106]	Detected face	Hybrid	X	Edges	SVM
Bai [7]	Detected face	Feature Based	-	Gradient	-
Wang [117, 118]	Detected face	Model Based	X	RNDA	Boosted classifiers cascade
Campadelli [16]	Detected face	Hybrid	X	Haar Wavelets	SVM
Hamouz [43]	Correct constellation	Model Based	X	Gabor filters	Constellation of face features + GMM
Kim [57]	Normalized face images	Model Based	X	Gabor jets	Eye model bunch
Niu [84]	Detected face	Model Based	X	Haar Wavelets	Boosted classifiers cascade
Wang [116]	Both eyes visible	Hybrid	X	Topographic labels	SVM
Huang [49]	Detected face	Hybrid	X	Mean, std, entropy	Genetic Algorithms + Decision trees
Kroon [62]	Detected face	Model Based	X	Pixels	Elastic bunch graph + LDA
Our Method	Detected face	Feature Based	-	Isophotes	-
Our Method	Detected face	Hybrid	X	SIFT on Isophotes	kNN

Table 2.1: Differences between the methods discussed in this chapter.

with a head mounted device, or to acquire high resolution eye images through zoomed cameras [20] combined with a chinrest to limit the allowed head movement. Furthermore, daylight applications are precluded due to the common use of active infrared (IR) illumination to obtain accurate eye location through corneal reflection [80]. Approaches that fuse IR and appearance based modalities are also proposed in literature [124], but dedicated hardware is still required.

In situations in which a closed up/infrared image of the eye is not available, the low resolution information about the location of the center of the eye is still very useful for a large number of applications (*e.g.* detecting gaze aversion, estimating the area of interest, automatic red eye reduction, landmarking, face alignment, gaming, and HCI). Therefore, in this chapter we want to focus on appearance based eye locators which can operate when infrared corneal reflections or high resolution eye images are not available. Many appearance based methods for eye center locators in low resolution settings are already proposed in literature, which can be roughly divided in three methodologies: (1) Model based methods, (2) Feature based methods and (3) Hybrid methods.

The model based methods make use of the holistic appearance of the eye (or even of the face). These approaches often use classification of a set of features or the fitting of a learned model to estimate the location of the eyes (possibly in combination with other facial features). By using the global appearance, model based methods have the advantage of being very robust and accurate in detecting the overall eye locations. However, as the success of these methods depends on the correct location of many features or the convergence of a full model, the importance of eye center location is often neglected due to its variability and learned as being in the middle of the eye model or of the two eye corner features. Therefore, in these cases, since the rest of the model is still correct and the minimization function satisfied, these methods are usually not very accurate

when they are faced with subtle eye center movements.

On the contrary, features based methods use well known eye properties (*i.e.* symmetry) to detect candidate eye centers from simple local image features (*e.g.* corners, edges, gradients), without requiring any learning or model fitting. Therefore, when the feature based methods are not confused by noise or surrounding features, the resulting eye location can be very accurate. However, as the detected features might often be wrong, the feature based methods are less stable then the model based ones.

Finally, in the hybrid methods, the multiple candidate eye locations obtained by a feature based method are discriminated by a classification framework, therefore using a previously learned eye model to achieve better accuracy.

Within the state of the art methods in each of the described methodologies, we studied the following subset: The method proposed by Asteriadis *et al*. [4] assigns a vector to every pixel in the edge map of the eye area, which points to the closest edge pixel. The length and the slope information of these vectors is consequently used to detect and localize the eyes by matching them with a training set. Jesorsky *et al*. [52] proposed a face matching method based on the Hausdorff distance followed by a Multi-Layer Perceptron (MLP) eye finder. Cristinacce *et al*. [25, 24] utilize a multistage approach to detect facial features (among them the eye centers) using a face detector, Pairwise Reinforcement of Feature Responses (PRFR), and a final refinement by using Active Appearance Model (AAM) [22]. Türkan *et al*. [106] apply edge projection (GPF) [122] and support vector machines (SVM) to classify estimates of eye centers. Bai *et al*. [7] exploit an enhanced version of Reisfeld's generalized symmetry transform [88] for the task of eye location. Wang *et al*. [117, 118] use statistically learned non-parametric discriminant features combined into weak classifiers, using the AdaBoost algorithm. Hamouz *et al*. [43] search for ten features using Gabor filters, use features triplets to generate face hypothesis, register them for affine transformations, and finally verify the remaining configurations using two SVM classifiers. Campadelli *et al*. [16] employ an eye detector to validate the presence of a face and to initialize an eye locator, which in turn refines the position of the eye using SVM on optimally selected Haar wavelet coefficients. Duffner [30] makes use of convolutional neural networks. The method by Niu *et al*. [84] uses a iteratively bootstrapped boosted cascade of classifiers. Kim *et al*. [57] discuss a multi-scale approach to localize eyes based on Gabor vectors. Wang *et al*. [116] treat faces as a 3D landscape, and they use the geometric properties of this terrain to extract potential eye regions. These candidates are then paired and classified using a Bhattacharyya kernel based SVM. Huang and Wechsler [49] also treat the face image as a landscape, where final state automata are genetically

evolved to walk the landscape and derive a saliency map for the best plausible location of the eyes. These salient regions are then classified as eyes by using genetically evolved decision trees. Finally, Kroon *et al*. [62] apply a Fisher Linear Discriminant to filter the face image and select the highest responses as eye center.

A summary of the characteristics of the discussed literature is presented in Table 2.1.

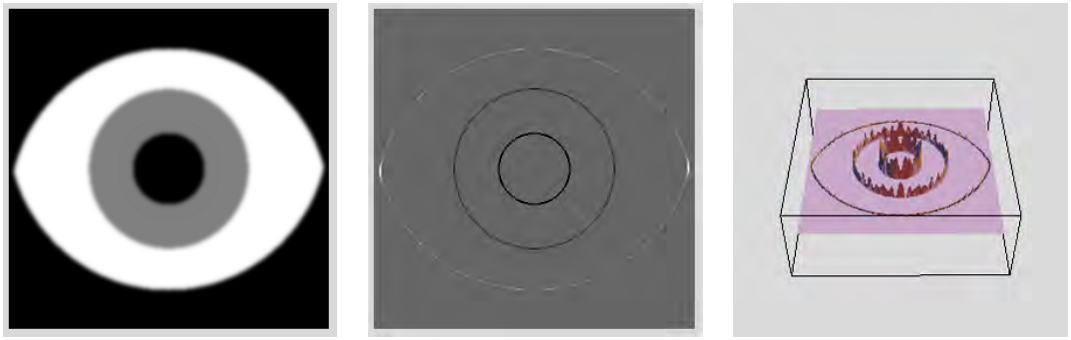
As indicated by the last lines of Table 2.1, this chapter will describe a feature based eye center locator which can quickly, accurately, and robustly locate eye centers in low resolution images and videos (*i.e*. coming from a simple web cam). Further, this chapter will also show how the method is easily extended into a hybrid approach. Hence, we made the following contributions:

- A novel eye location approach is proposed, which is based on the observation that eyes are characterized by radially symmetric brightness patterns. Contrary to other approaches using symmetry to accomplish the same task [7], our method makes use of isophotes (Section 2.2) to infer the center of (semi)circular patterns and gain invariance to in-plane rotation and linear lighting changes.
- A novel center voting mechanism (Section 2.3) based on gradient slope is introduced in the isophote framework to increase and weight important votes to reinforce the center estimates.
- The integration of our method in a scale space framework to find the most stable results.

In this chapter we study the accuracy and the robustness of the proposed approach to lighting, occlusion, pose and scale changes, and compare the obtained results with the state of the art systems for eye location in standard (*i.e*. visible wavelength), low resolution imagery (Section 2.4).

2.2 Isophotes Curvature Estimation

The iris and pupil are very prominent circular features which are characterized by an approximately constant intensity along the limbus (the junction between the sclera and the iris), and the iris and the pupil. We can therefore represent these features using isophotes, which are curves connecting points of equal intensity (one could think of isophotes as contour lines obtained by slicing the intensity landscape with horizontal planes). Since isophotes do not intersect each



(a) (b) (c)

Figure 2.1: The original image (a), its isophote curvature at the edges (b), and the 3D plot of the latter (c).

other, an image can be fully described by its isophotes. Furthermore, the shape of the isophotes is independent to rotation and linear lighting changes [67]. Due to these properties, isophotes have been successfully used as features in object detection and image segmentation [36, 56, 67].

To better illustrate the isophote framework, the notion of intrinsic geometry is introduced, *i.e.* geometry with a locally defined coordinate system. In every point of the image, a local coordinate frame is fixed in such a way that it points in the direction of the maximal change of the intensity, which corresponds to the direction of the gradient. This reference frame $\{v, w\}$ is also referred to as the *gauge coordinates*. Its frame vectors $\hat{w}$ and $\hat{v}$ are defined as:

$$\hat{w} = \frac{\{L_x, L_y\}}{\sqrt{L_x^2 + L_y^2}}; \hat{v} = \perp\hat{w}; \tag{2.1}$$

where L_x and L_y are the first-order derivatives[2] of the luminance function $L(x, y)$ in the x and y dimension, respectively. In this setting, a derivative in the w direction is the gradient itself, and the derivative in the v direction (perpendicular to the gradient) is 0 (no intensity change along the isophote).

In this coordinate system, an isophote is defined as $L(v, w(v)) =$ constant and its curvature is defined as the change w'' of the tangent vector w'. By implicit differentiation with respect to v of the isophote definition, we obtain:

$$L_v + L_w w' = 0; \qquad w' = -\frac{L_v}{L_w}. \tag{2.2}$$

[2]In our implementation, we use the fast anisotropic Gauss filtering method proposed in [39] to compute image derivatives. The used sigma parameter is equal in both direction (isotropic), with a rotation angle of $0°$

Since $L_v = 0$ from the gauge condition, then $w' = 0$. Differentiating again with respect to v, yields

$$L_{vv} + 2L_{vw}w' + L_{ww}w'^2 + L_w w'' = 0. \tag{2.3}$$

Solving for $\kappa = w''$ (the isophote curvature) and recalling that $w' = 0$, the isophote curvature is obtained as

$$\kappa = -\frac{L_{vv}}{L_w}. \tag{2.4}$$

In Cartesian coordinates, this becomes [26, 113, 47]

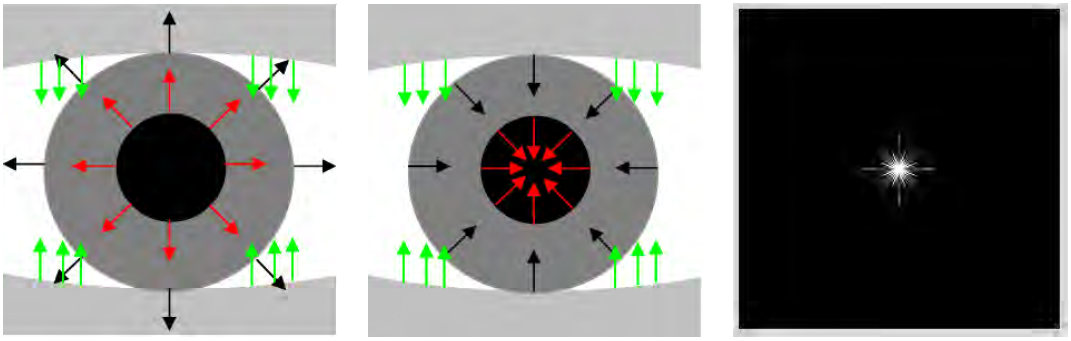
$$\kappa = -\frac{L_{vv}}{L_w} = -\frac{L_y^2 L_{xx} - 2L_x L_{xy} L_y + L_x^2 L_{yy}}{(L_x^2 + L_y^2)^{3/2}}. \tag{2.5}$$

To better illustrate the effect of the theory on an image, a simplistic eye model is used, shown in Figure 2.1(a). The isophote curvature of the eye model is shown in Figure 2.1(b). For presentation purposes, the shown curvature belongs to the isophote under the edges found in the image using a Canny operator. The crown-like shape of the values in the 3D representation (Figure 2.1(c)) is generated by the aliasing effects due to image discretization. By scaling[3] the original image this effect is reduced, but at higher scales the isophotes curvature might degenerate with the inherent effect of losing important structures in the image.

2.3 Isophote Centers

For every pixel, we are interested in retrieving the center of the circle which fits the local curvature of the isophote. Since the curvature is the reciprocal of the radius, Eq. (2.5) is reversed to obtain the radius of this circle. The obtained radius magnitude is meaningless if it is not combined with orientation and direction. The orientation can be estimated from the gradient, but its direction will always point towards the highest change in the luminance (Figure 2.2(a)). However, the sign of the isophote curvature depends on the intensity of the outer side of the curve (for a brighter outer side the sign is positive). Thus, by multiplying the gradient with the inverse of the isophote curvature, the sign of

[3]Scale in this context represents the standard deviation of the Gaussian kernel or its derivatives with which the image is convolved. See [26, 59] for more details.



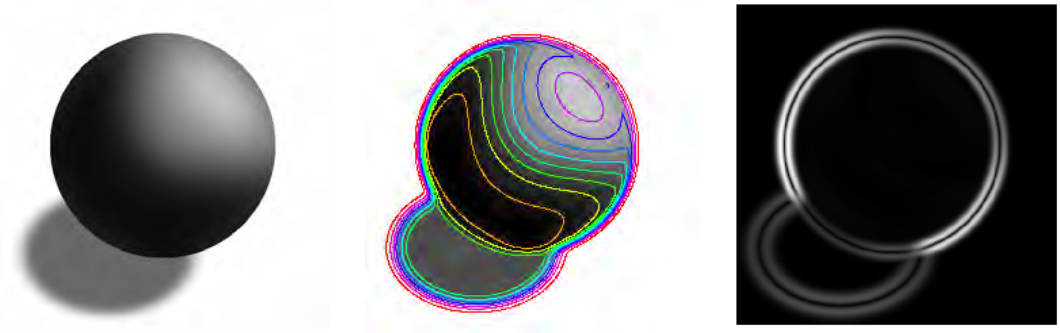
(a) (b) (c)

Figure 2.2: A detail showing the direction of the gradient under the image's edges (a), the displacement vectors pointing to the isophote centers (b), and the centermap (c).

the isophote curvature helps in disambiguating the direction to the center. Since the unit gradient can be written as $\frac{\{L_x, L_y\}}{L_w}$, we have

$$\begin{aligned} \{D_x, D_y\} &= \frac{\{L_x, L_y\}}{L_w}\left(-\frac{L_w}{L_{vv}}\right) = -\frac{\{L_x, L_y\}}{L_{vv}} \\ &= -\frac{\{L_x, L_y\}(L_x^2 + L_y^2)}{L_y^2 L_{xx} - 2L_x L_{xy} L_y + L_x^2 L_{yy}}. \end{aligned} \tag{2.6}$$

where $\{D_x, D_y\}$ are the displacement vectors to the estimated position of the centers, which can be mapped into an accumulator, hereinafter *"centermap"*. Note that sometimes the isophote curvature could assume extremely small or big values. This indicates that we are dealing with a "straight line" or a "single dot" isophote. Since the estimated radius to the isophote center would be too high to fall into the centermap or too little to move away from the originating pixel, the calculation of the displacement vectors in these extreme cases can simply be avoided. The set of vectors pointing to the estimated centers are shown in Figure 2.2(b). When compared to Figure 2.2(a), it is possible to note that the vectors are now all correctly directed towards the center of the circular structures. Figure 2.2(c) represents the cumulative vote of the vectors for their center estimate (*i.e.* the accumulator). Since every vector gives a rough estimate of the center, the accumulator is convolved with a Gaussian kernel so that each cluster of votes will form a single center estimate. The contribution of each vector is weighted according to a relevance mechanism, discussed in the following section.



(a) (b) (c)

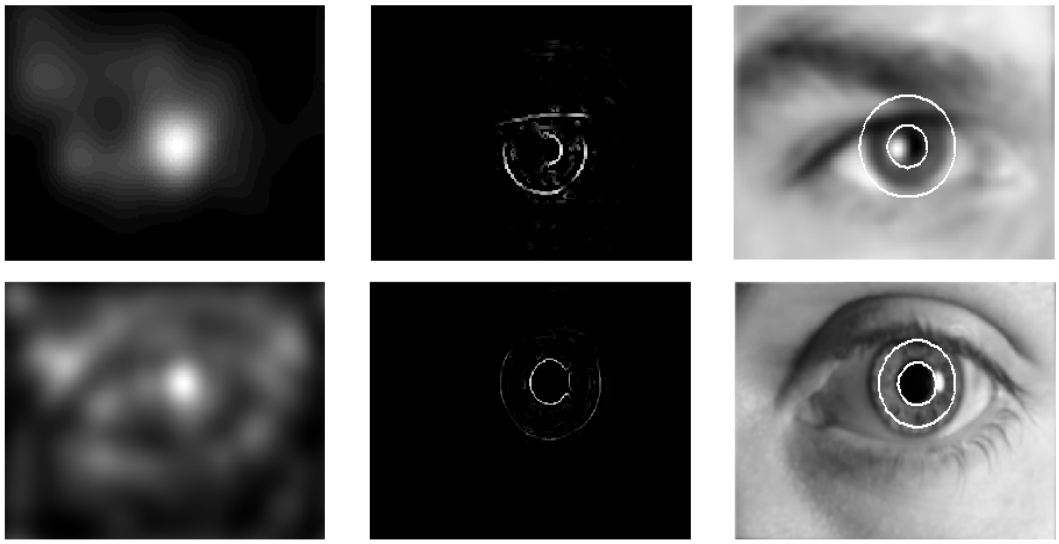
Figure 2.3: A sphere illuminated from above and casting a shadow (a), a sample of the isophotes superimposed to the image (b), the curvedness value of the same image (c).

2.3.1 Center Voting

So far an edge-based approach and a simplistic eye model were used to ease the explanations. Instead of using the peaks of the gradient landscape (*i.e.* edges), we propose to use the slope information around them, as they contain much more information.

In the simplistic eye model in Figure 2.1(a) there are only three isophotes: one describing the pupil, one describing the iris and one describing the boundary of the sclera. By convolving the eye model with a Gaussian kernel, it can be observed that the number of isophotes increases around the edges as the steepness of the edge decreases, and that each of these new isophotes is similar to the original isophotes (besides some creations and annihilations), so they can be used to generate additional evidence to vote for a correct center. The main idea is that by collecting and averaging local evidence of curvature, the discretization problems in a digital image could be lessened and an invariant and accurate eye center estimation could be achieved.

Contrary to the shown example, in real world environments there are no guarantees that the boundaries of an object are of the same intensity, *i.e.* that there is a sole isophote under the object's edges. In this case, allowing every single isophote to vote for a center will produce meaningless results since, due to highlights and shadows, the shape of the isophotes significantly differs from the shape of the object (Figure 2.3(a)(b)). In order to cope with this drawback, only the parts of the isophotes which are meaningful for our purposes should be considered.



(a) (b) (c)

Figure 2.4: The obtained centermap (a), the edges that contributed to the vote of the MIC (b), the average of the two biggest clusters of radiuses which voted for the found MIC (c).

To this end, an image operator that indicates how much a region deviates from flatness is needed. This operator is the curvedness [59], defined as

$$\text{curvedness} = \sqrt{L_{xx}^2 + 2L_{xy}^2 + L_{yy}^2}. \tag{2.7}$$

The curvedness can be considered as a rotational invariant gradient operator, which measures the degree of steepness of the gradient. Therefore, it yields low response on flat surfaces and edges, whereas it yields high response around the edges (Figure 2.3(c)). Since isophotes are slices of the intensity landscape, there is a direct relation between the value of the curvedness and the density of isophotes. Therefore, denser isophotes are likely to belong to the same feature (*i.e.* edge) and thus locally agree on the same center. A comparison between Figures 2.3(b) and 2.3(c) shows this relation between the curvedness and the image isophotes. It is clear that the curvedness is higher where the isophotes are denser. Therefore, by only considering the isophotes where the curvedness is maximal, they will likely follow an object boundary. The advantage of the proposed approach over a pure edge based method is that, by using the curvedness value as the weighting scheme for the importance of the vote, every pixel in the image may be used to contribute to a final decision. By summing the votes, we obtain high responses around the center of isocentric isophotes patterns. We

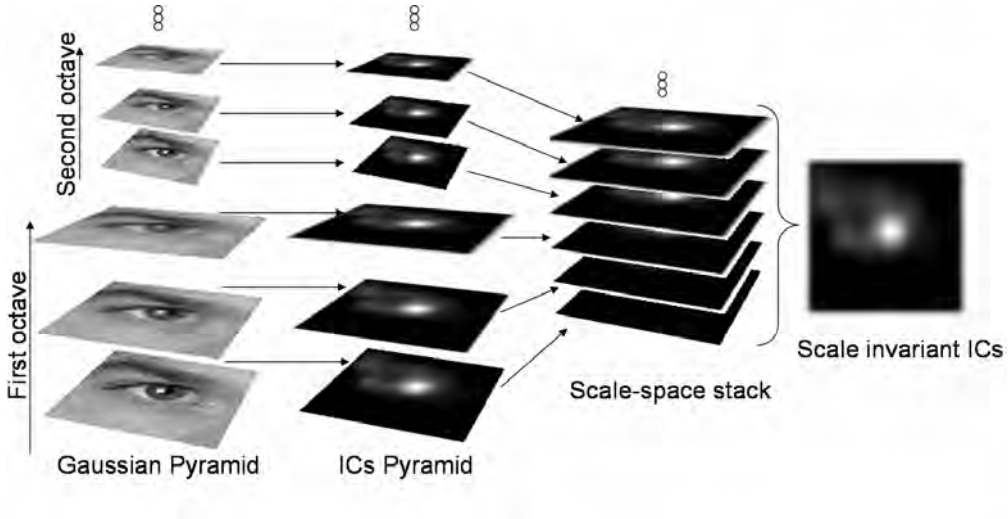


Figure 2.5: The scale space framework applied to eye location: the grayscale image is downscaled to different octaves, each octave is divided into intervals. For each interval, the centermap is computed and upscaled to a reference size to obtain a scale space stack. The combination of the obtained results gives the scale invariant isocenters.

call these high responses *"isocenters"*, or ICs. The maximum isocenter (MIC) in the centermap will be used as the most probable estimate for the soughtafter location of the center of the eye.

2.3.2 Eye Center Location

Recalling that the sign of the isophote curvature depends on the intensity of the outer side of the curve, it can be observed that a negative sign indicates a change in the direction of the gradient (*i.e.* from brighter to darker areas). Therefore, it is possible to discriminate between dark and bright centers by analyzing the sign of the curvature. Regarding the specific task of pupil and iris location, it can be assumed that the sclera is brighter than the iris and the pupil, therefore the votes which move from darker to brighter areas (*i.e.* in which the curvature agrees with the direction of the gradient), can be simply ignored in the computation of the isocenters. This allows the method to cope with situations in which strong highlights are present (*e.g.* when using an infrared illuminator or in the eye images in Figure 2.4), as long as the circular pattern of the eye is not heavily disrupted by them. Once the MIC is found, it is possible to retrieve a distribution of the most relevant radii (*i.e.* the pupil and the iris) by clustering together the distance to the pixels which voted for it. Figure 2.4 shows the results of

Algorithm 1 Pseudo-code for estimating isocenters.

- Compute first order, second order and mixed image derivatives $L_x, L_{xx}, L_{xy}, L_y, L_{yy}$
- Compute curvedness $= \sqrt{L_{xx}^2 + 2L_{xy}^2 + L_{yy}^2}$
- Compute $D = -\frac{L_x^2+L_y^2}{L_y^2 L_{xx} - 2L_x L_{xy} L_y + L_x^2 L_{yy}}$
$Dx = L_x * D$
$Dy = L_y * D$
Initialize an empty centermap
for $i = 1$ to image width **do**
  for $j = 1$ to image height **do**
    $x = i + round(Dx(i, j))$
    $y = j + round(Dy(i, j))$
    if $D > 0$ (for dark centers only) **then**
      centermap$(x, y) +=$ curvedness(i, j)
    end if
  end for
end for
- Convolve centermap with a Gaussian kernel to cluster votes

the procedure applied on two high resolution images of eyes. Note from Figure 2.4(b) that the vote contribution coming from highlights are not considered in the computation of the MIC.

The proposed method for estimating isocenters can be summarized by the pseudo-code in Algorithm 1.

2.3.3 Eye Center Location: Scale and Scale Space

Although the proposed approach is invariant to rotation and linear illumination changes, it still suffers from changes in scale. While in the previous work [109] the scale problem was solved by exhaustively searching for the scale value that obtained the best overall results, here we want to gain scale independence in order to avoid adjustments to the parameters for different situations. Firstly, since the sampled eye region depends on the scale of the detected face and on the camera resolution, to improve scale independency each eye region is scaled to a reference window. While this technique is expected to slightly decrease the accuracy with respect to the previously proposed approach (due to interpolation artifacts), once the correct scale values are found for the chosen reference window, the algorithm can be applied at different scales without requiring an

exhaustive parameter search.

Furthermore, to increase robustness and accuracy, a scale space framework is used to select the isocenters that are stable across multiple scales. The algorithm is applied to an input image at different scales and the outcome is analyzed for stable results. To this end, a Gaussian pyramid is constructed from the original grayscale image. The image is convolved with different Gaussians so that they are separated by a constant factor in scale space. In order to save computation, the image is downsampled into octaves. In each octave the isocenters are calculated at different intervals: for each of the image in the pyramid, the proposed method is applied by using the appropriate σ as a parameter for image derivatives. In our experiments (Section 2.4), we used three octaves and three intervals for each octave (as in [70]). This procedure results in a isocenters pyramid (Figure 2.5). The responses in each octave are combined linearly, then scaled to the original reference size to obtain a scale space stack. Every element of the scale space stack is considered equally important therefore they are linearly summed into a single centermap. The highest peaks in the resulting centermap will represent the most scale invariant isocenters.

2.3.4 Eye Center Location: Mean Shift and Machine Learning

Although the MIC should represent the most probable location for the eye center, certain lighting conditions and occlusions from the eyelids are expected to result in a wrong MIC. In order to avoid obtaining other isocenters as eye center estimates, two additional enhancements to the basic approach presented in the previous section are proposed, the first using mean shift for density estimation and the second using machine learning for classification.

Mean shift (MS) usually operates on back-projected images in which probabilities are assigned to pixels based on the color probability distribution of a target, weighted by a spatial kernel over pixel locations. It then finds the local maximum of this distribution by gradient ascent [21]. Here, the mean shift procedure is directly applied to the centermap resulting from our method, under the assumption that the most relevant isocenter should have higher density of votes, and that wrong MICs are not so distant from the correct one (*e.g.* on an eye corner). A mean shift search window is initialized on the centermap, centered on the found MIC, with dimensions equal to half the detected eye region's height and width. The algorithm then iterates to converge to a region with maximal distribution of center votes. The isocenter closest to the center of the converged search window is selected as the new eye center estimate.

Method	Pixels	Sift
Fisher Discriminant	14.05%	10.82%
Nearest Mean	30.75%	14.02%
Scaled nMean	30.38%	13.54%
Parzen	7.36%	6.92%
Neural Network	25.00%	29.38%
kNN	7.78%	**6.10%**

Table 2.2: Mean errors obtained by some of the tested classification methods on the raw pixels and the SIFT-like descriptor.

Machine Learning: instead of considering the strongest isocenter as eye center estimate, the aim is to consider the n most relevant ones and to discriminate between them using any classification framework. In this way, the task of the classifier is simplified as it only has to deal with a two class problem (eye center or not) and to discriminate between a couple of features (centered on the n most relevant isocenters). Note that the final performance of the system will always be bounded by the quality of the candidate locations (more on this in Section 2.4.3). For our experimentation, two different input features are used, centered on the candidate isocenters: 1) the pixel intensity sampled from a fixed window (dimensions depending on the detected face boundary) scaled to a 256 dimensional feature vector and 2) a SIFT [70] based descriptor, which differs from the SIFT as it does not search for scale invariant features, since the location of the feature is already known. Removing invariances from SIFT in an application-specific way has been shown to improve accuracy in [98].

The reasoning behind the choice of these two specific features is that 1) intensity is a rich source of information, and should naturally be included as baseline and 2) SIFT features have been shown to yield good generalization due to the reduced feature space and robustness. Both descriptors are computed on the original image, centered on the location of each of the candidate isocenters. Afterwards, the obtained descriptors are scaled to a reference size.

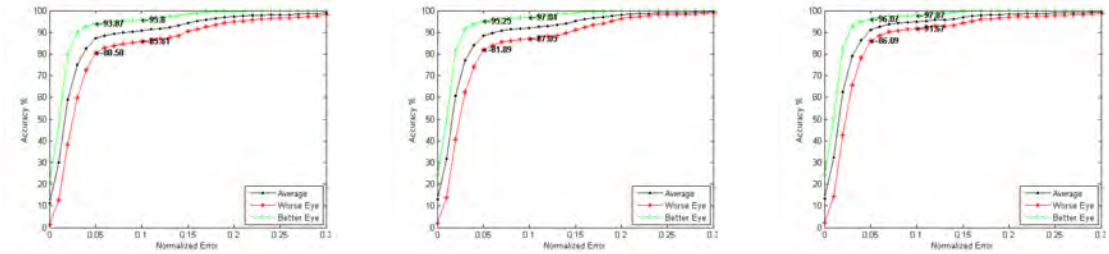
The following classification frameworks were selected to be representative of different classification approaches which are sensible to the selected features and method [31]: A one-against-all linear Fisher discriminant; A nearest mean and a scaled nearest mean classifier (in which the features are scaled to fit a normal distribution); A Parzen density estimator with feature normalization for each class based on variance; An automatically trained feed-forward neural network classifier with a single hidden layer; A kNN classifier, where k is optimized with respect to the leave-one-out error obtained from the database.

Figure 2.6: Sample of success (first row) and failures (second row) on the BioID face database; a white dot represents the estimated center.

For the sake of completeness, the obtained classification results are shown in Table 2.2. We have used 10-fold cross-validation in each experiment, where both training and validation folds are actually selected from the original training set. The test set, on which we report our overall results, is not seen during cross-validation. Given the simplicity of the problem, it is not surprising that the kNN classifier with the more robust SIFT-based descriptor is able to achieve the best results (Table 2.2). This is because the features are extracted around a point suggested by our method, hence it is quite likely that the training and testing feature vectors will not be exactly aligned (*e.g.* it not always centered on the eye center or the eye corner). Hence, the robustness of the feature descriptor to minor perturbations from the target location plays an important role, and SIFT provides for this by allowing overlaps in shifted vectors to result in meaningful similarity scores. In view of the high accuracies, computational cost is also a major guiding factor, hence the combination of the SIFT based feature and the kNN classification framework is used in the evaluation as an example of a hybrid variant of our method.

2.4 Evaluation

So far, high resolution images of eyes have been used as examples. In this section, the proposed method is tested on low resolutions eye images, *e.g.* coming from face images captured by a web cam. Additionally, the method is tested for robustness in changes in pose, illumination, scale and occlusion.



(a) MIC (b) MIC+MS (c) MICs+ SIFT kNN

Figure 2.7: Accuracy vs. minimum (better eye) and maximum (worse eye) normalized error obtained by the proposed methods on the BioID database.

2.4.1 Procedure and Measures

In order to obtain low resolution eye images from face images in the used test sets, the face position of each subject is estimated by using the boosted cascade face detector proposed by Viola and Jones [114][4]. The rough positions of the left and right eye regions are then estimated using anthropometric relations[5]. The proposed procedure is then applied to the cropped eye regions in order to accurately locate the center of the eye.

The *normalized error*, indicating the error obtained by the worse eye estimation, is adopted as the accuracy measure for the found eye locations. This measure was proposed by Jesorsky *et al*. [52] and is defined as:

$$e = \frac{\max(d_{\text{left}}, d_{\text{right}})}{w}, \tag{2.8}$$

where d_{left} and d_{right} is the Euclidean distance between the found left and right eye centers and the ones in the ground truth, and w is the Euclidean distance between the eyes in the ground truth. In this measure, $e \leq 0.25$ (a quarter of the interocular distance) roughly corresponds to the distance between the eye center and the eye corners, $e \leq 0.1$ corresponds to the range of the iris, and $e \leq 0.05$ corresponds the range of the pupil. To give upper and lower bounds to the accuracy, in our graphs (Figures 2.7 and 2.9) the *minimum normalized error* (obtained by considering the better eye estimation only) and an average between the better and worse estimation are also shown. These values are also needed

[4]The OpenCV implementation is used in our experiments

[5]We empirically found that, in the used datasets, eye centers are always contained within two regions starting from 20%x30% (left eye) and 60%x30% (right eye) of the detected face region, with dimensions of 25%x20% of the latter.

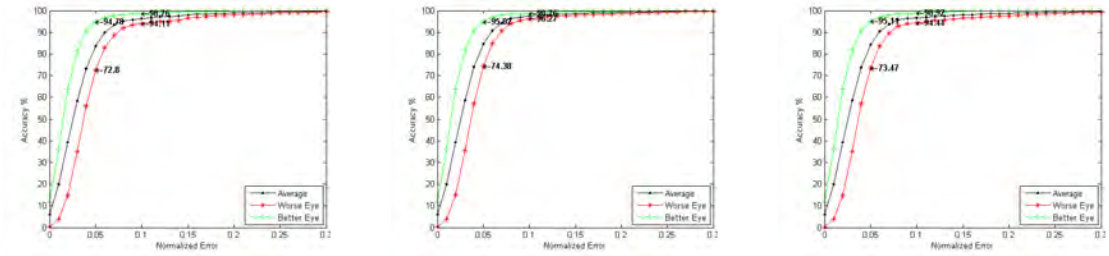
Figure 2.8: Sample of success (first row) and failures (second row) on the color FERET face database; a white dot represents the estimated center, while a red dot represents the human annotation.

in order to compare our results with other published works which make use of the normalized error measure in a non standard way.

2.4.2 Results

The BioID [10] and the color FERET [87] databases are used for testin. The BioID database consists of 1521 grayscale images of 23 different subjects and has been taken in different locations and at different times of the day (*i.e.* uncontrolled illumination). Besides changes in illumination, the positions of the subjects change both in scale and pose. Furthermore, in several samples of the database the subjects are wearing glasses. In some instances the eyes are closed, turned away from the camera, or completely hidden by strong highlights on the glasses. Due to these conditions, the BioID database is considered a difficult and realistic database. The size of each image is 384x288 pixels. A ground truth of the left and right eye centers is provided with the database.

The color FERET database contains a total of 11338 facial images collected by photographing 994 subjects at various angles, over the course of 15 sessions between 1993 and 1996. The images in the color FERET Database are 512 by 768 pixels. In our case we are only interested in the accuracy of the eye location in frontal images, therefore only the frontal face (fa) and alternate frontal face (fb) partitions of the database are considered. Figure 2.6 and Figure 2.8 show the qualitative results obtained on different subjects of the BioID and the color FERET databases, respectively. We observe that the method successfully deals with slight changes in pose, scale, and presence of glasses. By analyzing the fail-



(a) MIC (b) MIC+MS (c) MICs+ SIFT kNN

Figure 2.9: Accuracy vs. minimum (better eye) and maximum (worse eye) normalized error obtained by the proposed methods on the color FERET database.

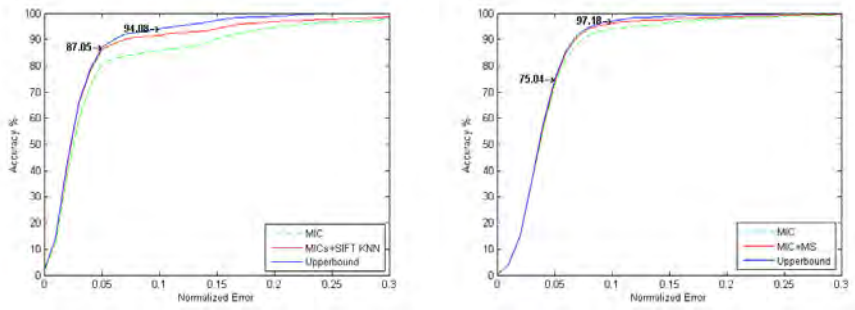


Figure 2.10: A summary of the better and worse results obtained on the BioID and on the color FERET databases in comparison with the respective upper bound curves.

ures (second rows) it can be observed that the system is prone to errors when the circular eye pattern is altered by the presence of closed eyelids or strong highlights on the glasses. When these cases occur, the iris and pupil do not contribute enough to the center voting, so the eyebrows or the eye corners assume a position of maximum relevance.

The graphs in Figure 2.7(a) and Figure 2.9(a) quantitatively show the accuracy of our method for different e. While it is clear that most of the results are nearly optimal, there is a saddle on the normalized error around the value of 0.15. This clustering of errors proves that few errors occur between the real eye centers and the eye corners/eyebrows. The improvement obtained by using the mean shift procedure for maximum density can be seen by comparing the graphs in Figures 2.7(a) and (b). Without any additional constraint, the results improved with $\approx 2.5\%$ over the basic approach. The graphs in Figure 2.7 (c) and 2.9 (c) show the accuracy obtained by using the kNN classifier to discriminate between the top MICs, which in case of the BioID database achieved better results than

Method	Accuracy ($e \leq 0.05$)	Accuracy ($e \leq 0.10$)	Accuracy ($e \leq 0.25$)
MIC [109]	77.15%	82.11%	96.35%
MIC+MS [109]	79.56%	85.27%	97.45%
MICs+SIFT [109]	**84.10%**	**90.85%**	**98.49%**
MIC	80.58%	85.81%	96.56%
MIC+MS	81.89%	87.05%	**98.00%**
MICs+SIFT	**86.09%**	**91.67%**	97.87%
Asteriadis [4]	74.00%*	81.70%	97.40%
Jesorsky [52]	40.00%	79.00%	91.80%
Cristinacce [25]	56.00%*	96.00%	98.00%
Türkan [106]	19.00%*	73.68%	**99.46%**
Bai [7]	37.00%*	64.00%	96.00%
Campadelli [16]	62.00%	85.20%	96.10%
Hamouz [43]	59.00%	77.00%	93.00%
Kim [57]	n/a	**96.40%**	98.80%
Niu [84]	**75.00%***	93.00%	98.00%*
Kroon [62]	65.00%*	87.00%	98.80%*

Table 2.3: Comparison of accuracy vs. normalized error in the BioID database. *= the value estimated from author's graphs.

both the basic and the mean shift approaches, while the results on the color FERET database show a slight drop in accuracy, which becomes comparable to the basic approach. This can be explained by the fact that by using classification the successful outcome of the system will inherently depend on the conditions it was trained, together with the fact that the annotation in the color FERET database is sometimes unreliable. In fact, it can be seen from Figure 2.8 that the human annotation (indicated by a red dot) is sometimes less accurate than the estimated eye center (indicated by a white dot). This negatively affects the accuracy for accurate eye center location and its effect can be seen by comparing the graphs in Figure 2.9 to the ones in Figure 2.7: the differences between the results at $e \leq 0.05$ and the ones at $e \leq 0.1$ are significantly higher than the ones found on the BioID database.

2.4.3 Comparison with the State of the Art

Our results are compared with state of the art methods in the literature which use the same databases and the same accuracy measure. While many recent results are available on the BioID database, results on the color FERET database

Method	Accuracy ($e \leq 0.05$)	Accuracy ($e \leq 0.10$)	Accuracy ($e \leq 0.25$)
MIC	72.80%	94.11%	98.21%
MIC+MS	**74.38%**	**96.27%**	**99.17%**
MICs+SIFT	73.47%	94.44%	98.34%
Campadelli [16]	67.70%	89.50%	96.40%
Duffner [30]	79.00%*	97.00%*	99.00%*
Kim [57]	91.80% ($e \leq 0.07$)		

Table 2.4: Comparison of accuracy vs. normalized error in the color FERET database.*= uses average normalized error.

are often evaluated on custom subsets and with different measures, therefore not directly comparable. This is the case of Kim *et al.* [57] which only use 488 images of the "fa" subset (frontal face, neutral expression) and of Duffner [30] which, instead of using the maximum error measure as in this work, evaluates the normalized error on both eyes instead of the worse one only. This is equivalent to the "Average" curves in Figure 2.9 where the best variant (MIC+MS) obtains an accuracy of 85.10% for $e \leq 0.05$ versus Duffner's 79.00%. Tables 2.3 and 2.4 show the comparison between our methods and the state of the art methods mentioned in Section 2.1 for several allowed normalized errors. Where inexplicitly reported by the authors, the results are estimated from their normalized error graphs, safely rounded up to the next unit. In case of the comparison on the BioID database the table also reports the accuracy reported in our previous work [109] in order to have a direct comparison with the obtained results, where it is clear that the use of the scale space further improved every result by about 2%. It can be seen that, for an allowed normalized error smaller than 0.25, we achieved accuracy comparable to the best methods. For iris location ($e \leq 0.1$), our method shows less accuracy with respect to the some of the other methods. This can be justified by the fact that the other methods exploit other facial features to estimate and adjust the position of the eyes (*i.e.* the eye center is in between the eye corners) which works extremely well to find a point in the middle of two eye corners, but often does not have enough information to locate the exact position eye center in between them. However, our approach excels for accurate eye center location ($e \leq 0.05$), even by using the basic approach.

To measure the maximum accuracy achievable by our method, we computed the normalized error obtained by selecting the isocenter closest to the ground truth. The graphs in Figure 2.10 show the comparison between the better and worse performing variants of the proposed method and an additional curve which represents the found upper bound on the BioID and color FERET databases.

Figure 2.11: Effect of changes in illumination and pose (last row) on a subject of the Yale Face Database B.

It is possible to see that the proposed extensions helped in increasing the bending point of the curve, while the rest of the curve is similar in all the cases. This means that the extensions reduced the number of times an eye corner or an eyebrow is detected as the MIC, moving the results closer to the upper bound. Note that the SIFT extension almost follows the upper bound for $e \leq 0.05$.

2.4.4 Robustness to Illumination and Pose Changes

To systematically evaluate the robustness of the proposed eye locator to lighting and pose changes, two subsets of the Yale Face Database B [38] are used. The full database contains 5760 grayscale images of 10 subjects each seen under 576 viewing conditions (9 poses x 64 illuminations). The size of each image is 640x480 pixels. To independently evaluate the robustness to illumination and pose, the system is tested on frontal faces under changing illumination (10 subjects x 64 illuminations) and on changing pose under ambient illumination (10 subjects x 9 poses).

The first two rows of Figure 2.11 show a qualitative sample of the results obtained for a subject in the illumination subset. By analyzing the results, we note that the system is able to deal with light source directions varying from $\pm 35°$ azimuth and from $\pm 40°$ elevation with respect to the camera axis. The results obtained under these conditions are shown in Table 2.5. When compared to the previously published results in [109], the improvement in accuracy obtained by the scale space framework is about 2%, especially for the MS extension. For

Figure 2.12: Effect of changes in illumination (horizontally) and pose (vertically) on a subject of the Multi-PIE database.

Method	Accuracy ($e \leq 0.05$)	Accuracy ($e \leq 0.10$)	Accuracy ($e \leq 0.25$)
MIC	77.68%	85.32%	95.72%
MIC+MS	79.82%	88.07%	96.64%
MICs+SIFT	80.12%	86.85%	96.73%

Table 2.5: Accuracy vs. normalized error for illumination changes on the Yale Face Database B.

higher angles, the method is often successful for the less illuminated eye and sporadically for the most illuminated one: if the eye is uniformly illuminated, its center is correctly located, even for low intensity images; if, on the other hand, the illumination influences only parts of the eye, the shape of the isophotes is influenced by shadows, resulting in an unreliable MIC.

The last row in Figure 2.11 shows the results of the eye locator applied to a subject the pose subset of the Yale Face Database B. The quantitative evaluation on this dataset shows the robustness of the proposed approach to pose changes: due to the higher resolution and the absence of occlusions and glasses, all the variants achieved an accuracy of 100.00% for $e \leq 0.05$. The first errors are actually found by considering $e \leq 0.04$ for the basic method (MIC), where the system achieves an accuracy of 95.45%.

To systematically evaluate the combined effect of lighting and pose changes, the CMU Muliti-PIE database [42] is used. The database contains images of 337 subjects, captured under 15 view points and 19 illumination conditions in four

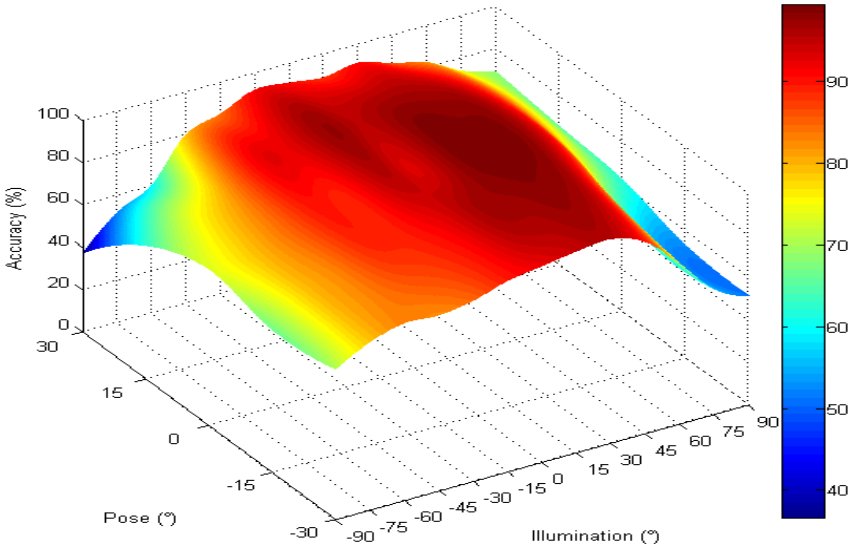


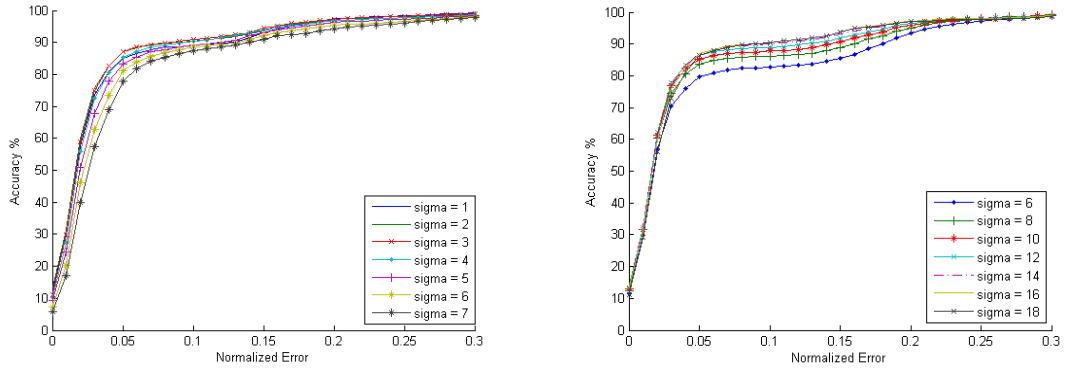
Figure 2.13: A interpolated 3D representation of the data in Table 2.6.

	Illumination												
Pose	$-90°$	$-75°$	$-60°$	$-45°$	$-30°$	$-15°$	$0°$	$+15°$	$+30°$	$+45°$	$+60°$	$+75°$	$+90°$
$-30°$	70.28%	78.71%	83.13%	82.33%	84.74%	89.56%	91.97%	94.38%	95.58%	89.56%	74.70%	58.23%	51.00%
$-15°$	66.67%	78.31%	82.73%	87.95%	88.76%	91.57%	96.39%	97.19%	97.99%	94.78%	81.12%	57.43%	52.61%
$0°$	73.09%	78.31%	83.94%	89.96%	89.16%	95.58%	93.17%	98.80%	98.80%	97.59%	89.56%	71.89%	61.45%
$+15°$	62.25%	71.89%	78.71%	91.16%	92.37%	97.19%	95.18%	96.79%	96.79%	95.18%	88.76%	77.51%	64.66%
$+30°$	36.55%	51.41%	59.84%	79.12%	84.34%	91.16%	89.96%	87.95%	90.36%	85.54%	81.53%	73.09%	68.27%

Table 2.6: Combined effect of changes in head pose and illumination in the Multi-PIE database for $e \leq 0.05$, using MIC+MS.

recording sessions for a total of more than 750,000 images. The database shows very challenging conditions for the proposed method, as many subjects have closed eyes due to the natural reaction to flashes, or the irises are occluded due to very strong highlights on the glasses, generated by the flashes as well.

As no eye center annotation is provided with the database, we manually annotated the eye centers and the face position of all the subjects in the first session (249), in 5 different poses (the ones in which both eyes are visible), under all the different illumination conditions present in the database. This annotation is made publicly available on the author's website. Figure 2.12 shows a qualitative sample of the database, together with the annotation and obtained result. Table 2.6 and the interpolated 3D plot in Figure 2.13 quantitatively show the result of this experiment for $e \leq 0.05$, using the MIC+MS variant. As with the YALE Face Database B, this variant obtained better results with respect to



(a) (b)

Figure 2.14: The effect of changing the parameters on the BioID database. Changing the size of (a) Gaussian kernel for image derivatives (b) Gaussian kernel for the centermap.

the MICs+SIFT variant due to the variance present in the training data, which makes it difficult for the classifier to find a clear decision boundary to discriminate eye centers from the rest of the features.

By analyzing the results, it is possible to derive insights about the accuracy, the success and failures of the proposed method: Although the frontal face with frontal illumination is expected to achieve the best accuracy, the fact that the flash directly reflects on subjects wearing glasses contributes to a drop in accuracy in that specific setting. However, if the illumination is shifted by just $15°$, the system is able to achieve an accuracy of 98.80%, which is the best result obtained in this experiment. Furthermore, it is possible to note that the accuracy is higher when the face is turned towards the light. This is because the shape of the irises in these situations will not be affected by shadows. This behavior is very clear from the 3D plot in Figure 2.13.

2.4.5 Robustness to Scale Changes

The system uses only two parameters: the "scale" of the kernel (σ_{total}) with which the image derivatives are computed and the "scale" of the Gaussian kernel with which the centermap is convolved (*i.e.* how much near votes affect each other). Figure 2.14(a) shows the changes in accuracy for different values of σ_{total}. It can be seen that, by changing this parameter, the curves shift vertically, therefore the value that results in the highest curve should be selected as the best σ_{total} (in this case, 3). This is not the case with the graph in Figure 2.14(b) which shows the

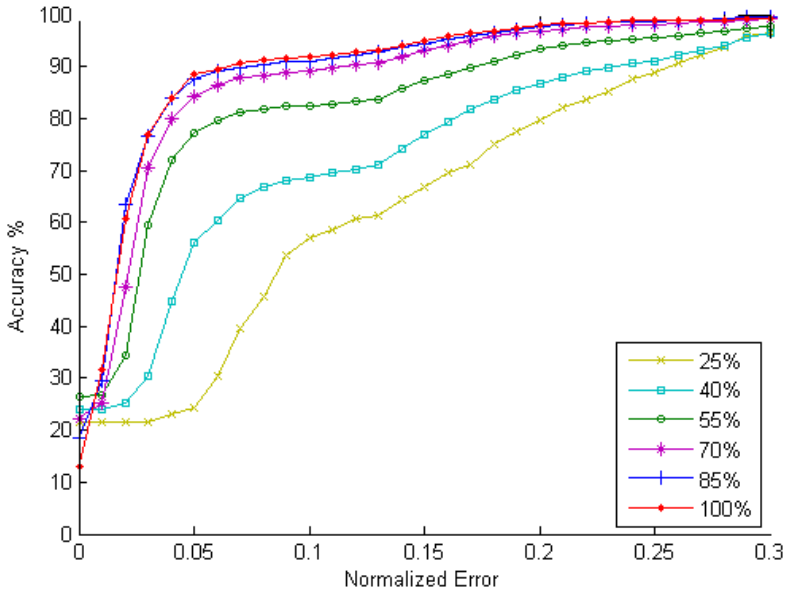


Figure 2.15: The effect of scaling down of the images on the BioID database, at different percentages of the original size.

effect of changing the blurring parameter of the centermap (*i.e.* how near votes affect each other). In this case, the accuracy remains basically unchanged for accurate results ($e \leq 0.4$), while selecting a proper kernel size (*e.g.* 16) improves the bending point of the curve (*i.e.* the errors between the eye centers and eye corners). In order to study the effect of changing the scale now that the best parameters are known, the test images are downscaled to fixed ratio values: 25%, 40%, 55%, 70%, 85% and 100% of the original image size. The eyes are then cropped and upscaled to a reference window size (*e.g.* 60x50 pixels) where the best value of the size of the Gaussian kernel for the image derivatives is experimentally known. The scale space isocenters pyramid is then computed with a value of σ^2 at interval i calculated by

$$\sigma^2_{\text{total}} = \sigma^2_i + \sigma^2_{i-1}, \tag{2.9}$$

therefore

$$\sigma_i = \sqrt{\sigma^2_{\text{total}} - \sigma^2_{i-1}}. \tag{2.10}$$

The result of this experiment is shown in Figure 2.15. Note that downscaling from 100% to 85% and to 70% does not significantly affect the results, while the rest of the results are still acceptable considering the downsampling artifacts

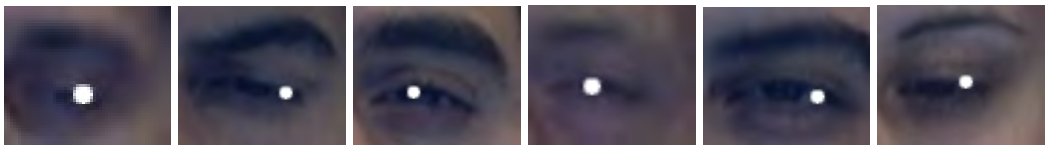


Figure 2.16: First frames in which the eye center estimations are off by more than 5% of the interocular distance.

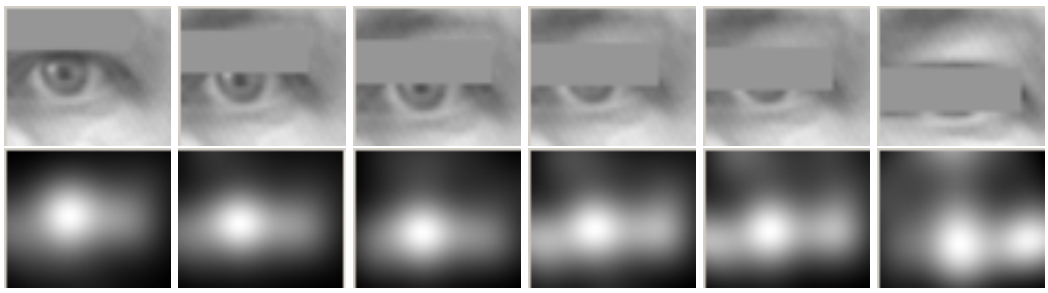


Figure 2.17: The effect of eye occlusion on the centermap. Other dark features gain more relevance as the eye's circular pattern gets occluded.

and the size of the images.

2.4.6 Robustness to Occlusions

Since the proposed method is based on the assumption that the eye pattern is circular and that is visible, it is important to evaluate the robustness of the approach to partial occlusion which might result from eye blinking, facial expressions and extreme eye positions. Since many subjects in the BioID database display closed or semi-closed eyes, the obtained overall accuracy can already give an indication that the proposed approach is able to handle eye occlusion. To validate the robustness to occlusion of the proposed method, a simple experiment was performed on 10 subjects. The subjects were requested to gaze at the camera and slowly close their eyes. The system recorded the first image in which the eye center estimation would move by more than 5% of the interocular distance from their initial position. A sample of the results is shown in Figure 2.16, where it is clear that the system is able to handle situations in which the iris is almost completely occluded.

To give a better overview of the behavior of our method to progressive occlusions, we designed a procedural experiment that simulates eye occlusions by a shifting rectangle. The color of the rectangle was sampled from the average color of eyelids in the database. Note that, since the rectangle's edges are

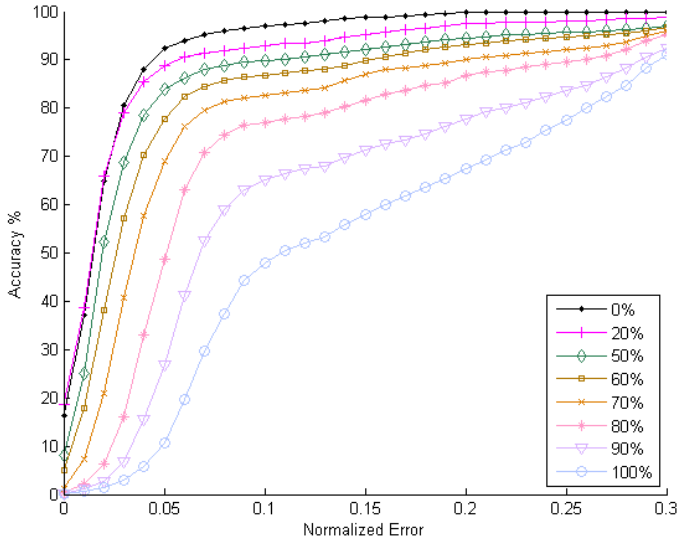


Figure 2.18: The effect of occluding eyes at different percentages on the BioID database.

straight (null curvature), the votes generated by the rectangle are automatically discarded by the method and will not affect the center detection. In order to analyze every percentage of occlusion, a subset of the subjects displaying completely open eyes where selected from the BioID database. In our experiment, we define a 0% eye occlusion when the lower side of the occluding rectangle touches the uppermost point the iris, a 50% occlusion when it passes through the middle of the pupil, and a 100% occlusion when is tangent to the lowest point in of the iris. The graph in Figure 2.18 shows that the proposed method can successfully detect eye centers even if they are occluded by more than 60%. In fact, up to 50% occlusion, the method degrades in accuracy only by less than 10% for accurate eye center location ($e \leq 0.05$). An insight that arises from this experiment is that at 100% occlusion the system is sometimes able to locate the eye center. This is because the closed eye region is generally darker than the features around the eye, and therefore it can still generate votes which fall into the pupil area. An example of the occlusion procedure in this experiment is shown in Figure 2.17. Note that, since the centermap is always normalized, it does not seem to change significantly. However, it is possible to see that the found MIC moves down and that the right eye corner gains more votes as the circular iris pattern disappears.

Figure 2.19: Some of the subjects displaying extreme eye rotations at the bottom-leftmost key location in the collected dataset.

2.4.7 Robustness to Eye Rotations

To systematically evaluate the robustness of the proposed approach to eye rotation, an additional experiment in which 21 subjects followed a moving dot on a computer screen was performed. In the experiment, the dot crosses key locations, in which the frames are saved and manually annotated for the eye location. The key locations are defined by the pixel value in which the dot is displayed on the screen in 6x4 key locations, starting at 50x50 pixels and ending at 1200x740 pixels, in increments of 230 pixels on the horizontal and vertical direction, respectively. Given the size of the screen (40 inches) and the distance of the subjects (750mm), this value indicates a horizontal and an approximate vertical span of 46° and 24°, respectively. The subjects were requested to keep the head as static as possible while following the dot. However, we noted that every subject performed some slight head movements to be able to comfortably gaze at the dot moving at peripheral locations of the screen. This indicates that the subjects were not comfortable to reach the peripheral key location without moving their head. Therefore, we can argue that these peripheral locations reached the limit of 'natural' eye rotation. Since the built dataset was free of occlusions (besides the occlusion caused by the eyelids when the eye is significantly rotated), the achieved accuracy for $e \leq 0.05$ was 100% in all key locations.

This result proves that the proposed method is not significantly affected by natural eye rotations, and therefore it is not affected by the natural occlusions from

	Horizontal (pixels)					
Vertical (pixels)	50	280	510	740	970	1200
50	76.19%	80.95%	100%	71.43%	85.71%	80.95%
280	90.48%	85.71%	95.24%	100%	85.71%	100%
510	90.48%	85.71%	80.95%	100%	76.19%	71.43%
740	95.24%	71.43%	76.19%	71.43%	76.19%	66.67%

Table 2.7: Effect of changes in eye rotation for $e \leq 0.02$, using MIC+MS.

the eyelids in extreme locations and by the change in shape of the iris due to the rotation. Table 2.7, shows the average accuracy at the selected key locations for $e \leq 0.02$. At this extremely small range, errors start to be significant when moving away from the central area. Note that in some peripheral areas the accuracy is still 100%. We believe that this is due to the head movements required to gaze at the moving dot comfortably.

2.4.8 Discussion

As stated in the introduction, the accuracy of the proposed system should not be compared to commercial eye-gaze trackers. The approach discussed here is targeted to niche applications where eye location information is useful but constrained on low resolution imagery, for applications in which close up view or corneal reflection is unavailable (*e.g.* facebook images) and where the use of an eye-gaze tracker would be prohibitively expensive or impractical (*e.g.* automatic red eye reduction on a picture camera).

One of the advantages of the proposed approach that should be discussed is its low computational complexity, since the basic system (without scale space and classification) only requires the computation of image derivatives which is linear in the size of the image and the scale ($O(\sigma N)$). This allows for a real-time implementation while keeping a competitive accuracy with respect to the state of the art. On a 2.4GHz Intel Core 2 Duo, using a single core implementation, the system was able to process ≈ 2500 eye regions per second on a 320x240 image. Including the face detector and the mean shift procedure, the algorithm takes 11ms per frame, which roughly corresponds to 90 frames per second. Therefore, the final frame rate is only limited by the web cam's frame rate. By using the scale space approach, the accuracy improved by about 2% and the system benefits from improved independence to scale conditions. In this way, the method can be applied to different situation without needing an ad-hoc param-

eter search. By using the scale space pyramid, the computational complexity increases linearly with the number of intervals that are analyzed. A tradeoff between the discussed increase in accuracy and the computational complexity must be chosen according to the desired target application.

For instance, even if the best results are obtained by the MICs+SIFT method, applying it to video frames thirty times per second will necessarily result in unstable estimates. However, the MIC+MS method scales perfectly to use temporal information: the converged position of the MS window can be kept as initialization for the next frame, and the eye locator can be used to reinitialize the tracking procedure when it is found to be invalid (*i.e.* when the current MIC falls outside the mean shift window). This synergy between the two components allows the tracking system to be fast, fully autonomous and user independent, which is preferable to the less stable, data dependent but more accurate MICs+SIFT variant.

Given the high accuracy and low computational requirements, we foresee the proposed method to be successfully adopted as a preprocessing step to other systems. In particular, systems using classifiers (*e.g.* [16, 52, 106]) should benefit from the reduction in the search and learning phases and can focus on how to discriminate between few candidates. Furthermore, note that our system does not involve any heuristics or prior knowledge to discriminate between candidates. We therefore suggest that it is possible to achieve superior accuracy by integrating the discussed method into systems using contextual information (*e.g.* [25, 43]).

2.5 Conclusions

In this chapter, a new method to infer eye center location using circular symmetry based on isophote properties is proposed. For every pixel, the center of the osculating circle of the isophote is computed from smoothed derivatives of the image brightness, so that each pixel can provide a vote for its own center. The use of isophotes yields low computational cost (which allows for real-time processing) and robustness to rotation and linear illumination changes. A scale space framework is used to improve the accuracy of the proposed method and to gain robustness to scale changes.

An extensive evaluation of the proposed approach was performed, testing it for accurate eye location in standard low resolution images and for robustness to illumination, pose, occlusion, eye rotation, resolution, and scale changes. The

comparison with the state of the art suggested that our method is able to achieve highest accuracy and can be successfully applied do very low resolution image of eyes, but this is somewhat bounded by the presence of at least 40% of the circular eye pattern in the image. Given the reported accuracy of the system, we believe that the proposed method provides enabling technology to niche applications in which a good estimation of the eye center location at low resolutions is fundamental.

3

Synergetic Eye Center Location and Head Pose Estimation[1]

3.1 Motivation and Related Work

Image based gaze estimation is important in many applications, spanning from human computer interaction (HCI) to human behavior analysis. In applications where human activity is under observation from a static camera, the estimation of the visual gaze provides important information about the interest of the subject, which is commonly used as control devices for disabled people [1], to analyze the user attention while driving [28], and other applications. It is known that gaze is a product of two contributing factors [66]: the head pose and the eye locations. The estimation of these two factors is often achieved using expensive, bulky or limiting hardware [19]. Therefore, the problem is often simplified by either considering the head pose or the eye center locations as the only feature to understand the interest of a subject [69, 90].

There is an abundance of literature concerning these two topics separately: recent surveys on eye center location and head pose estimation can be found in

[1] R. Valenti, N.Sebe, and T. Gevers, "Combining Head Pose and Eye Location Information for Gaze Estimation", IEEE Transactions on Image Processing, 2011. Ideas previously appeared in:
R. Valenti, A. Lablack, N.Sebe, C. Djeraba, and T. Gevers, "Visual Gaze Estimation by Joint Head and Eye Information", International Conference on Pattern Recognition, 2010, and
R. Valenti, Z. Yucel and T. Gevers, "Robustifying Eye Center Localization by Head Pose Cues". IEEE Conference on Computer Vision and Pattern Recognition, 2009.

[44] and [82]. The *eye location* algorithms found in commercially available eye trackers share the problem of sensitivity to head pose variations, require the user to be either equipped with a head mounted device, or to use a high resolution camera combined with a chinrest to limit the allowed head movement. Furthermore, daylight applications are precluded due to the use of active infrared (IR) illumination to obtain accurate eye location through corneal reflection. The appearance based methods which make use of standard low resolution cameras are considered to be less invasive and so more desirable in a large range of applications. Within the appearance-based methods for eye location proposed in literature, [84, 57, 62, 109] reported results support the evidence that accurate appearance based eye center localization is becoming feasible and that it could be used as an enabling technology for a various set of applications.

Head pose estimation often requires multiple cameras, or complex face models which requires accurate initialization. Ba *et al*. [5] improve the accuracy of pose estimates and of the head tracking by considering these as two coupled problems in a probabilistic setting within a mixed state particle filter framework. They refine this method by fusion of four camera views in [6]. Huang *et al*. propose to integrate a skin-tone edge-based detector into a Kalman filter based robust head tracker and hidden Markov model based pose estimator in [50]. Hu *et al*. describe a coarse to fine pose estimation method by combining facial appearance asymmetry and 3D head model [48]. A generic 3D face model and an ellipsoidal head model are utilized in [105] and [2], respectively. In [79] an online tracking algorithm employing adaptive view based appearance models is proposed. The method provides drift-free tracking by maintaining a dynamic set of keyframes with views of the head under various poses and registering the current frame to the previous frames and keyframes.

Although several head pose or eye location methods have shown success in gaze estimation, the underlying assumption of being able to estimate gaze starting from eye location or head pose only is valid in a limited number of scenarios [96, 115]. For instance, if we consider an environment composed of a target scene (a specific scene under analysis, such as a computer monitor, an advertising poster, a shelf, etc.) and a monitored area (the place from which the user looks at the target scene), an eye gaze tracker alone would fail when trying to understand which product on the shelf is being observed, while an head pose gaze estimator alone would fail in finely control the cursor on a computer screen.

Hence, a number of studies focused on *combining head and eye information* for gaze estimation are available in literature: Newman and Matsumoto [83, 75] consider a tracking scenario equipped with stereo cameras and employ 2D fea-

ture tracking and 3D model fitting. The work proposed by Ji *et al*. [54] describe a real-time eye, gaze and head pose tracker for monitoring driver vigilance. The authors use IR illumination to detect the pupils and derive the head pose by building a feature space from them. Although their compound tracking property promote them against separate methods, the practical limitations and the need for improved accuracy make them less attractive in comparison to monocular low resolution implementations.

However, no study is performed on the feasibility of an accurate appearance-only gaze estimator which considers both the head pose and the eye location factors. Therefore, our goal is to develop a system capable of analyzing the visual gaze of a person starting from monocular video images. This allows to study the movement of the user's head and eyes in a more natural manner than traditional methods, as there are no additional requirements needed to use the system.

To this end, we propose a unified framework for head pose and eye location estimation for visual gaze estimation. The head tracker is initialized using the location and orientation of the eyes while the latter are obtained by pose-normalized eye patches obtained from the head tracker. A feedback mechanism is employed in the evaluation of the tracking quality. When the two modules do not yield concurring results, both are adjusted to get in line with each other, aiming to improve the accuracy of both tracking schemes. The improved head posed estimation is then used to define the field of view, while displacement vectors between the pose-normalized eye locations and their resting positions are used to adjust the gaze estimation obtained by the head pose only. In this way, a novel, multimodal visual gaze estimator is obtained.

The contributions are the following:

- Rather than just a sequential combination, we propose a unified framework which provides a deep integration of the used head pose tracker and the eye location estimation methods.
- The normal working range of the used eye locator ($\sim 30°$) is extended. The shortcomings of the reported eye locators due to extreme head poses are compensated using the feedback from the head tracker.
- Steered by the obtained eye location, the head tracker provides better pose accuracy and can better recover the correct pose when the head tracker is lost.
- The eye location and head pose information are used together in a multimodal visual gaze estimation system, which uses the eyes to adjust the

gaze location determined by the head pose.

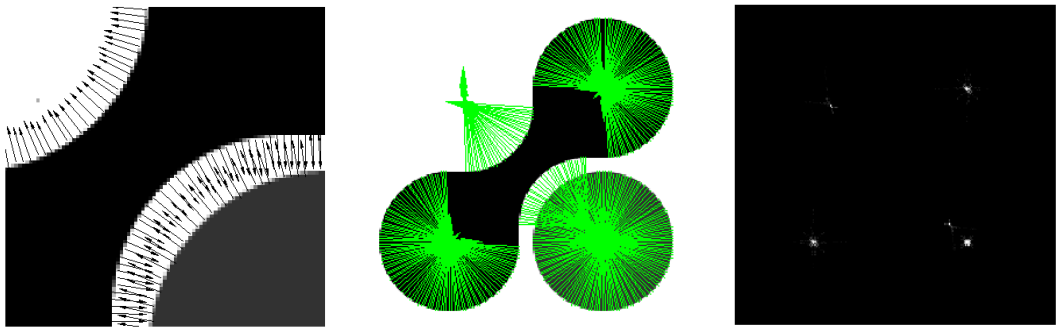
The chapter is structured as follows: the reason behind the choice and the theory of the used eye locator and head pose estimator will be discussed in Section 3.2. In Section 3.3, the discussed components will be combined in a synergetic way, so that the eye locator will be aided by the head pose, and the head pose estimator will be aided by the obtained eye locations. Section 3.4 will describe how the improved estimations could be used to create a combined gaze estimation system. In Section 3.5, three independent experiments will analyze the improvements obtained on the head pose, on the eye location and on the combined gaze estimation. Finally, the discussions and conclusions will be given in Section 3.6.

3.2 Eye Location and Head Pose Estimation

To describe how the used eye locator and head pose estimator are combined in Section 3.3, in this section the used eye locator and the head pose estimator are discussed.

3.2.1 Eye Center Localization

As we are discussing appearance based methods, an overview of the state of the art on the subject is given. The method used by Asteriadis *et al.* [4] assigns a vector to every pixel in the edge map of the eye area, which points to the closest edge pixel. The length and the slope information of these vectors is consequently used to detect and localize the eyes by matching them with a training set. Cristinacce *et al.* [25] use a multistage approach to detect facial features (among them the eye centers) using a face detector, Pairwise Reinforcement of Feature Responses (PRFR), and a final refinement by using Active Appearance Model (AAM) [22]. Türkan *et al.* [106] use edge projection (GPF) [122] and support vector machines (SVM) to classify estimates of eye centers. Bai *et al.* [7] use an enhanced version of Reisfeld's *generalized symmetry transform* [88]) for the task of eye location. Hamouz *et al.* [43] search for ten features using Gabor filters, use features triplets to generate face hypothesis, register them for affine transformations and verify the remaining configurations using two SVM classifiers. Finally, Campadelli *et al.* [16] use an eye detector to validate the presence of a face and to initialize an eye locator, which in turn refines the position of the eye using SVM on optimally selected Haar wavelet coefficients. With re-



(a) (b) (c)

Figure 3.1: The direction of the gradient under the image's edges (a), the displacement vectors pointing to the isophote centers (b), and the centermap (c)

spect to the aforementioned methods, the method proposed in [109] achieves the best results for accurate eye center localization, without heavy constraints on illumination, rotation, and robust to slight pose changes, and will therefore be used.

The method uses isophotes (i.e., curves connecting points of equal intensity) properties to obtain the center of (semi)circular patterns. This idea is based on the observation that the eyes are characterized by radially symmetric brightness patterns, hence it looks for the center of the curved isophotes in the image. In Cartesian coordinates, the isophote curvature k is expressed as:

$$\kappa = -\frac{\frac{\delta I}{\delta y}^2 \frac{\delta^2 I}{\delta x^2} - 2\frac{\delta I}{\delta x}\frac{\delta^2 I}{\delta x \delta y}\frac{\delta I}{\delta y} + \frac{\delta I}{\delta x}^2 \frac{\delta^2 I}{\delta y^2}}{(\frac{\delta I}{\delta x}^2 + \frac{\delta I}{\delta y}^2)^{\frac{3}{2}}}.$$

where, for example, $\frac{\delta I}{\delta x}$ is the first order derivative of the intensity function I on the x dimension. The distance to the center of the iris is found as the reciprocal of the above term. The orientation is calculated using the gradient but its direction indicates always the highest change in luminance (Figure 3.1(a)). The gradient is then multiplied by inverse of the isophote curvature to disambiguate the direction of the center. Hence, the displacement vectors from every pixel to the estimated position of the centers, $D(x, y)$ are found to be

$$D(x,y) = -\frac{\{\frac{\delta I}{\delta x}, \frac{\delta I}{\delta y}\}(\frac{\delta I}{\delta x}^2 + \frac{\delta I}{\delta y}^2)}{\frac{\delta I}{\delta y}^2 \frac{\delta^2 I}{\delta x^2} - 2\frac{\delta I}{\delta x}\frac{\delta^2 I}{\delta x \delta y}\frac{\delta I}{\delta y} + \frac{\delta I}{\delta x}^2 \frac{\delta^2 I}{\delta y^2}}.$$

In this way, every pixel in the image gives a rough estimate of its own center the center as shown in Figure 3.1(b). Since the sign of the isophote curvature

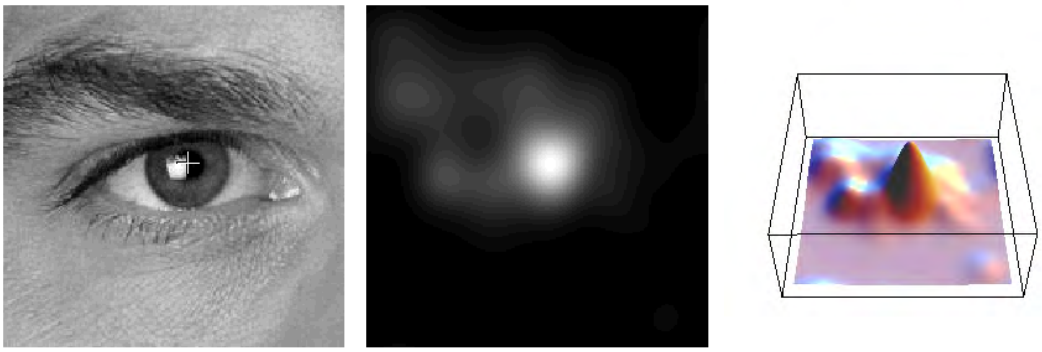


Figure 3.2: The source image, the obtained centermap and the 3D representation of the latter

depends on the intensity of the outer side of the curve, bright and dark centers can be discriminated by the sign of the curvature. Since sclera is assumed to be brighter than the cornea and the iris, votes with a positive isophote curvature are ignored as they are likely to come from non-eye regions or highlights. In order to collect this information and deduce the location of a global eye center, $D(x, y)$'s are mapped into an accumulator (Figure 3.1(c)).

Instead of attributing the same importance to every center estimate, a relevance mechanism is used to yield more accurate center estimation, in which only the parts of the isophote which follow the edges of the object are considered. This weighting is performed by using curvedness [59]:

$$curvedness = \sqrt{\frac{\delta^2 I}{\delta x^2}^2 + 2\frac{\delta^2 I}{\delta x \delta y}^2 + \frac{\delta^2 I}{\delta y^2}^2}.$$

The accumulator is then convolved with a Gaussian kernel so that each cluster of votes will form a single estimate. The maximum peak found in the accumulator is assumed to represent the location of the estimated eye center. An example is illustrated in Figure 3.2. For this case, the eye center estimate can clearly be seen on the 3D plot.

In [109], it is shown that the described method yields low computational cost allowing real-time processing. Further, due to the use of isophotes, the method is shown to be robust against linear illumination changes and to moderate changes in head pose. However, the accuracy of the eye center location drops significantly in the presence of head poses which are far from frontal. This is due to the fact that, in these cases, the analyzed eye structure is not symmetric anymore and thus the algorithm delivers increasingly poor performance with respect to the distance from the frontal pose. This observation shows that it is desirable

to be able to correct the distortion given by the pose so that the eye structure under analysis keeps the symmetry properties. To obtain the normalized image patches invariant to changes in head pose, a head pose estimation algorithm will be employed.

3.2.2 Head Pose Estimation

Throughout the years, different methods for head pose estimation have been developed. The 3D model based approaches achieve robust performance and can deal with large rotations. However, most of the method work reasonably in restricted domains only, *e.g.* some systems only work when there is stereo-data available [78, 92], when there is no (self-) occlusion, or when the head is rotating not more than a certain degree [18]. Systems that solve most of these problems, usually do not work in real-time due to the complex face models they use [120], or require accurate initialization. However, if the face model complexity is reduced to simpler ellipsoidal or cylindrical shape, this creates a prospect for a real-time system, and can be simply initialized starting from eye locations. The cylindrical head model (CHM) approach has been used by a number of authors [13, 18, 119]. Among them, the implementation of Xiao *et al.* [119] works remarkably well. This cylindrical approach is still capable of tracking the head also in situations where the head turns more than 30° from the frontal position and will therefore be used in this work, and will be outlined as follows.

To achieve good tracking accuracy, a number of assumptions are considered for the simplification of the problem. First of all, camera calibration is assumed to be provided beforehand and a single stationary camera configuration is considered. For perspective projection a pin hole camera model is studied.

The initial parameters of the cylindrical head model and its initial transformation matrix are computed as follows: Assuming that the face of the subject is visible and frontal, its size is used to initialize the cylinder parameters and the pose $\mathbf{p} = [\omega_x, \omega_y, \omega_z, t_x, t_y, t_z]$ according to anthropometric values [41, 27], where ω_x, ω_y, and ω_z are the rotation parameters and t_x, t_y, t_z are the translation parameters. The eye locations are detected in the face region and are used to give a better estimate of the t_x and t_y. The depth, t_z, is adjusted by using the distance between the detected eyes, d. Finally, since the detected face is assumed to be frontal, the initial pitch (ω_x) and yaw (ω_y) angles are assumed to be null, while the roll angle (ω_z) is initialized by the relative position of the eyes.

To analyze the effect of the motion of the cylindrical head model on the image

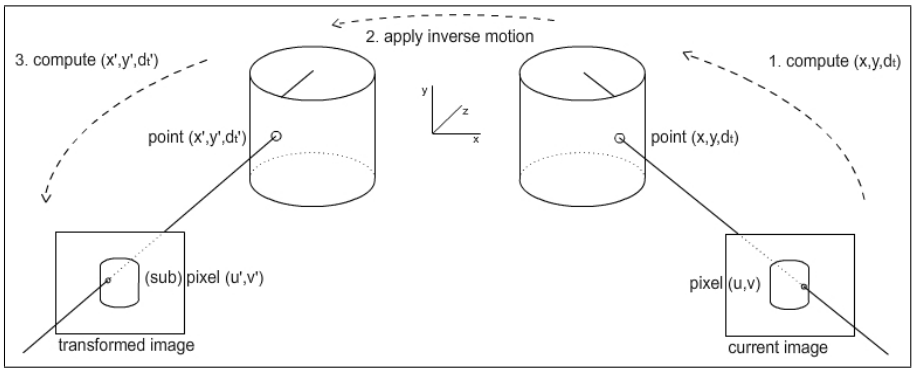


Figure 3.3: Orientation of the cylinder and its visualization on image plane

frame, the relation between the 3D locations of the points on the cylinder and their corresponding projections on the 2D image plane need to be established. Therefore the 3D locations of the points with respect to the reference frame need to be determined first. This is obtained by sampling points on the cylinder. After obtaining the coordinates of these points on the 3D elliptic cylindrical model, perspective projection is applied to get the corresponding coordinates on the 2D image plane.

Since the cylindrical head model is assumed to be aligned along $\mathbf{y}$-axis of the reference frame and to be positioned such that the center coincides with the origin (as shown in Figure 3.3), any point $p = (p_x, p_y, p_z)^T$ on the cylinder satisfies the following explicit equation:

$$\left(\frac{p_x}{\mathbf{r_x}}\right)^2 + \left(\frac{p_z}{\mathbf{r_z}}\right)^2 = 1, \tag{3.1}$$

where $\mathbf{r_x}$ and $\mathbf{r_z}$ stand for the radii of the ellipse along $\mathbf{x}$- and $\mathbf{z}$-axes respectively. To calculate the coordinates of the points on the visible part of the cylinder, the front region is sampled in an $N_s \times N_s$ grid-like structure on $x - y$ plane and corresponding depth values are obtained by using Equation 3.1. These sampled points are considered to summarize the motion of the cylinder and they are employed in Lukas Kanade optical flow algorithm. The perspective projection of the 3D points on the elliptic cylindrical face model gives the 2D pixel coordinates in the image plane. Let point $p = (p_x, p_y, p_z)^T$ in Figure 3.3 be a point sampled on the cylinder and point $u = (u_x, u_y)^T$ be its projection on the image plane. Figure 3.4 illustrates the side view of this setting by making a pin hole camera assumption for the sake simplification. Using similarity of triangles in Figure 3.4, the following equations apply for the relation between p and u

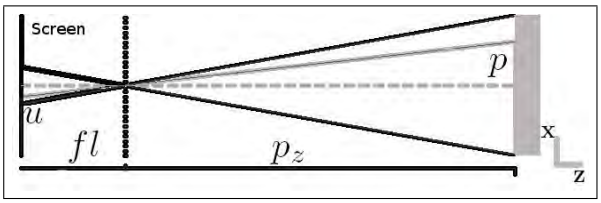


Figure 3.4: Perspective projection of point p onto image plane by a pin hole camera assumption

$$\begin{aligned} p_x &= \tfrac{p_z u_x}{fl}, \\ p_y &= \tfrac{p_z u_y}{fl}, \end{aligned} \tag{3.2}$$

where fl stands for the focal length of the camera. This relation is summarized by a perspective projection function $\mathbf{P}$, which maps the 3D points onto the 2D image plane employing the above given identities,

$$\mathbf{P}(p) = u.$$

As shown in Figure 3.3, the cylinder is observed at different locations and with different orientations at two consecutive frames F_i and F_{i+1}. This is expressed as an update in pose vector $\mathbf{p_i}$ by the rigid motion vector $\Delta\mu_i = [\omega_x^i, \omega_y^i, \omega_z^i, \tau_x^i, \tau_y^i, \tau_z^i]$. To compute this motion vector, it is required to establish the relation between p_i and u_i of F_i and their corresponding locations on F_{i+1}. In formulation of this relation, three transformation functions are employed as illustrated in Figure 3.3. The 3D transformation function $\mathbf{M}$ maps p_i to p_{i+1}, whereas the 2D transformation function $\mathbf{F}$ maps u_i to u_{i+1} and the perspective projection function $\mathbf{P}$ maps p_i to u_i.

It can be derived that the explicit representation of the perspective projection function in terms of the rigid motion vector parameters and the previous coordinates of the point is [119]:

$$\begin{aligned} \mathbf{P}(\mathbf{M}(p_i, \Delta\mu)) = &\begin{bmatrix} p_i^x - p_i^y\omega_z + p_i^z\omega_y + \tau_x \\ p_i^x\omega_z + p_i^y - p_i^z\omega_x + \tau_y \end{bmatrix} \\ &\times \tfrac{fl}{-p_i^x\omega_y + p_i^y\omega_x + p_i^z + \tau_z}. \end{aligned}$$

In the next section, the estimated head pose will be used to obtain the pose normalized eye patches.

Figure 3.5: Examples of eye regions sampled by pose

3.3 Synergetic Eye Location and CHM Tracking

As mentioned in the previous section, the CHM pose tracker and the isophote based eye location estimation methods have advantages over other reported methods. However, taken separately, they cannot work adequately under certain circumstances. In [109], the eye region is assumed to be frontal so that the eye locator can use curved isophotes to detect circular patterns. However, since the method is robust to slight changes in head pose, the system can still be applied with head poses up to $> 30^{\circ}$ at the cost of accuracy. On the other hand, the CHM pose tracker may erroneously converge to local minima and, after that, may not able to recover the correct track. By integrating the eye locator with the cylindrical head model, we aim to obviate these drawbacks.

Instead of a sequential integration of the two systems, an early integration is proposed. Relevant to our work is the approach proposed in [100]. The authors combine a cylindrical head model with an Active Appearance Model (AAM) approach to overcome the sensitivity to large pose variations, initial pose parameters, and problems of re-initialization. In the same way, we make use of the competent attributes of the cylindrical head model together with the eye locator proposed in [109] to broaden the capabilities of both systems and to improve the accuracy of each individual component. By comparing the transformation matrices suggested independently by both systems, in our method the eye locations will be detected given the head pose, and the head pose will be adjusted given the eye locations.

To this end, after the cylinder is initialized in 3D space, the 2D eye locations

detected in the first frame are used as reference points (*e.g.* the "+" markers in Figure 3.7). These reference points are projected onto the cylindrical head model, so that the depth values of the eye locations are known. The reference eye points are then used to estimate the successive eye locations and are in turn updated by using the average of the found eye locations.

3.3.1 Eye Location by Pose Cues

Figure 3.6: Examples of extreme head poses and the respective pose-normalized eye locations. The results of the eye locator in the pose normalized eye region is represented by a white dot.

Around each reference point projected onto the 3D model, an area is sampled and transformed by using the transformation matrix obtained by the head pose tracker (Figure 3.5). The pixels under these sampled points are then remapped into a normalized canonical view (Figure 3.6). Note that extreme head poses are also successfully corrected, although some perspective projection errors are retained. The eye locator described in Section 3.2.1 is then applied to these pose normalized eye regions. The highest peak in the obtained accumulator which is closer to the center of the sampled region (therefore closer to the reference eye location obtained by pose cues), is selected as estimated eye center (the white dots in Figure 3.6 and the "x" markers in Figure 3.7) . In this way, as long as the CHM tracker is correctly estimating the head pose, the localized eyes can

Figure 3.7: A mistake of the standard eye locator (.), corrected by the pose cues (x) according to the reference points (+)

be considered to be optimal. Figure 3.7 shows two examples in which the default eye locator would fail ("." marker) but the pose normalized eye estimation would be correct ("x" marker).

3.3.2 Pose Estimation by Eye Location Cues

Since there is uncertainty about the quality of the pose obtained by the head tracker, the found pose-normalized eye location can be used as a cue for quality control. Given that the 3D position of the eyes is known, it is possible to calculate its pose vector and compare it with the one obtained by the head tracker. When the distance between the two pose vectors is larger than a certain threshold, the vectors are averaged and the transformation matrix of the tracker is recomputed. In this way, the head model is adjusted to a location that should ease the correct convergence and therefore recover the correct track. As an additional quality control, the standard eye locator is constantly used to verify that the eye location found by pose cues is consistent with the one obtained without pose cues. Therefore, as in [79], when reliable evidence (*e.g.* the eye location in a frontal face) is collected and found to be in contrast with the tracking procedure, the latter is adjusted to reflect this.

In this manner, the eye locations are used to both initialize the cylinder pose and update it in case it becomes unstable, while the pose normalized eye locations are used to constantly validate the tracking process. Therefore, the CHM tracker and the eye locator interact and adjust their own estimations by using each other's information. This synergy between the two systems allows for an initialization-free and self-adjusting system. A schematic overview of the full system is shown in Figure 3.8, while its pseudo code is presented in Algorithm 2.

Algorithm 2 Pseudo-code of estimating eye locations by head pose

Initialize parameters
- Detect face and initialize cylinder parameters
- Get reference eye regions, R_r and R_l.
- Use distance between the eyes to get the depth element, t_z.
- Initialize pose $\mathbf{p}$ using eye locations

Iterate through all the frames
for $t = 0$ to last frame number **do**
- Assume intensity is constant between consecutive frames, $I_{t+1} = I_t$.
- Compute the gradient $\bigtriangledown I_{t+1}$ and the corresponding Gaussian pyramid for the current frame
- Initialize pose to the previous pose $p_{t+1} = p_t$

For all levels of Gaussian Pyramid
for $l = 0$ to 2 **do**
- Calculate motion between two frames $\mathbf{m} = \mathbf{p_{t+1}} * \mathbf{p_t}^{-1}$
- Load Gaussian pyramid image $I(l)$
- Initialize $\Delta\vec{p} = [0, 0, 0, 0, 0, 0]$

while maximum iterations not reached or $\Delta\vec{p} <$ threshold **do**
- Transform pixels p of $I(l)$ to p' with transformation matrix $\mathbf{M}$ and parameters $\mathbf{p}$ to compute $I_t(\mathbf{p})$
- Update and scale face region boundaries $(\vec{u}, \vec{v})$
- Do ray tracing to calculate t_z for each $p \in (\vec{u}, \vec{v})$
- Apply perspective projection, $p_x = \vec{u}_n * t_z$, $p_y = \vec{v}_n * t_z$
- Use inverse motion $\mathbf{m}'$ to get from p to p'
- With back-projection calculate pixels (u', v')
- Compute I_t with $I_{t+1}(\mathbf{m}) - I_t(\mathbf{m}')$.
- Compute $\bigtriangledown I_{t+1}(\mathbf{m}) \frac{\partial \mathbf{T}}{\partial \mathbf{p}}$ where T summarizes the projection model.
- Compute Hessian matrix in
- Compute $\sum w \left[\begin{array}{c} \bigtriangledown I_{t+1} \frac{\partial T}{\partial p} \end{array} \right]^T \sum \left[\begin{array}{c} I_t - I_{t+1} \end{array} \right]$
- Compute $\Delta\vec{p}$ using
- Update the pose and motion:
- $\mathbf{p_{t+1}} = \Delta\vec{p} \circ \mathbf{p_{t+1}}$
- $\mathbf{m} = \Delta\vec{p} \circ \mathbf{m}$

end while
- Update transformation matrix $\mathbf{M} = \Delta\vec{p} \circ \mathbf{M}$
- Transform reference eye regions R_r and R_l using $\mathbf{M}$
- Remap eye regions to pose normalized view
- Compute displacements vectors D on pose normalized eye regions accordingly to [109], using,

$$D(x,y) = -\frac{\{\frac{\delta I}{\delta x}, \frac{\delta I}{\delta y}\}(\frac{\delta I}{\delta x}^2 + \frac{\delta I}{\delta y}^2)}{\frac{\delta I}{\delta y}^2 \frac{\delta^2 I}{\delta x^2} - 2\frac{\delta I}{\delta x}\frac{\delta^2 I}{\delta x \delta y}\frac{\delta I}{\delta y} + \frac{\delta I}{\delta x}^2 \frac{\delta^2 I}{\delta y^2}}.$$

- Vote for centers weighted by $\sqrt{\frac{\delta^2 I}{\delta x^2}^2 + 2\frac{\delta^2 I}{\delta x \delta y}^2 + \frac{\delta^2 I}{\delta y^2}^2}$.
- Select isocenter closer to the center of eye region as eye estimate
- Remap eye estimate to cylinder coordinates
- Create pose vector from eye location and compare it to head tracker's

if distance between pose vector $>$ threshold **then**
average pose vectors and create the new $\mathbf{M}$

end if

end for

end for

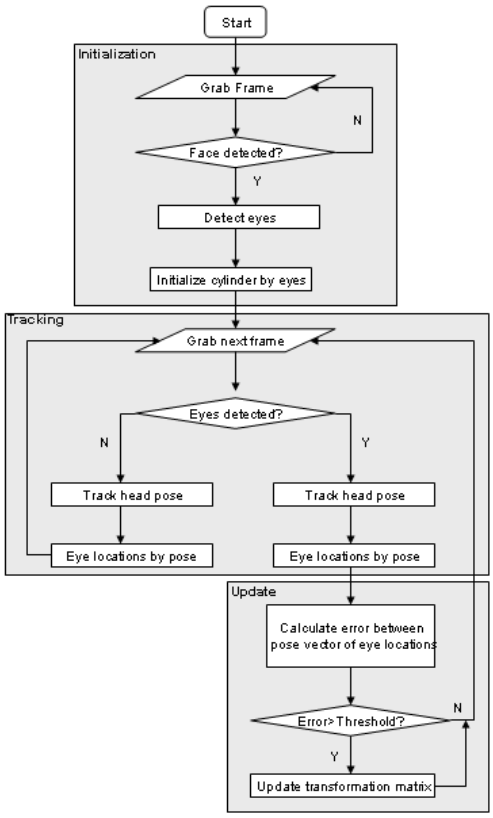


Figure 3.8: A schematic diagram of the components of the system

3.4 Visual Gaze Estimation

In the previous section, we described how the 2D eye center locations detected in the first frame are used as reference points (Figure 3.12). These reference points are projected onto the cylindrical head model and are then used to estimate the successive, pose normalized eye center locations. In this section, the displacement vectors between the resting position of the eyes (reference points) and the estimated eye location will be used to obtain joint visual gaze estimation, constrained within the visual field of view defined by the head pose.

3.4.1 The Human Visual Field of View

Studies on the human visual field of view [85] show that, while looking straight ahead, it has a vertical span of 130° (60° above and 70° below) and approxi-

mately $90°$ on each side, which corresponds to a photographic objective angle of $180°$ (Figure 3.9). The common field of view of the two eyes is called binocular field of view and spans $120°$. It is surrounded by two monocular fields of view of approximately $30°$.

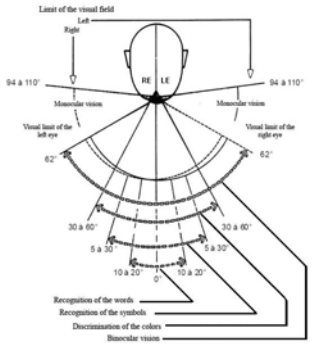


Figure 3.9: The binocular field of view in humans

The field of view can be approximated by a pyramid $OABCD$ where O represents the point between the two eyes and rectangle $ABCD$ represents the visual field of view at distance d. Further, the angles α and β denote the horizontal and vertical angles of the visual field of view in binocular human vision, respectively [63]. Since the pyramid is an approximation of the field of view, we are able to center it on the gaze point P so that it is in the middle of the field of view. In this case, the vector OP denotes the visual gaze vector (Figure 3.10).

The width (W) and height (H) of the visual field at distance d are computed by:

$$W = 2PF = 2d \tan \frac{\alpha}{2} \quad , \quad H = 2PG = 2d \tan \frac{\beta}{2}.$$

The projection of the visual field of view on the gazed scene in front of a user is a quadrilateral $A'B'C'D'$ with central gaze point P', and it is calculated by the intersection between the plane of the target scene P: $ax + by + cz + d = 0$ and lines (OA), (OB), (OC), (OD), and (OP). The head pose parameters computed by the method described in Section 3.2.2 are used to define the projection of the region of interest in the target scene.

3.4.2 Pose-Retargeted Gaze Estimation

So far, we considered the visual field of view defined by the head pose only, modeled so that the visual gaze of a person (the vector defining the point of interest) corresponds to the middle of the visual field of view. However, it is clear

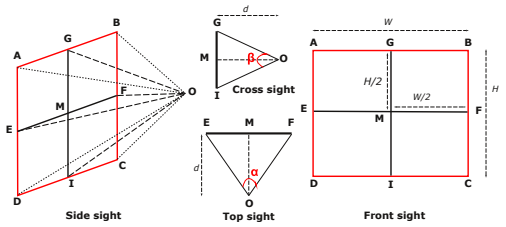
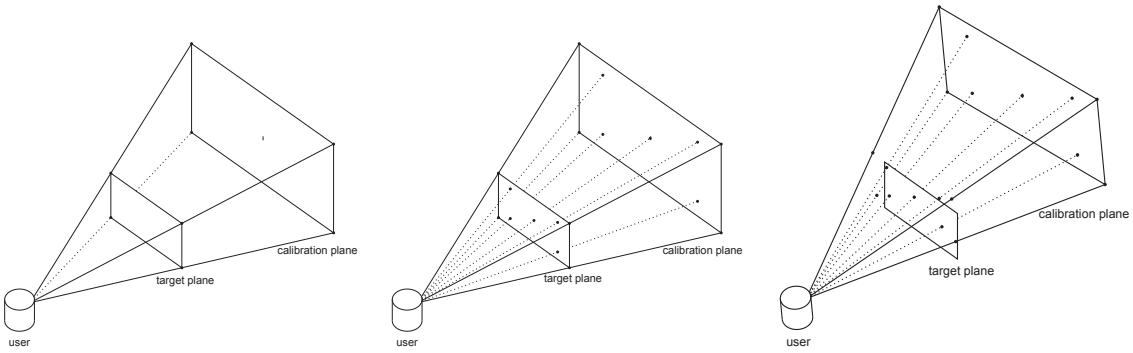


Figure 3.10: Representation of the visual field of view at distance d

that the displacements of the eyes from their resting positions will influence the estimation of the visual field of view. In general, most methods avoid this problem by assuming that the head does not move at all and assume that the eyes do not rotate in the ocular cavities but just shift on the horizontal and vertical plane [111]. In this way, the problem of eye displacement is simply solved by a 2D mapping of the location of the pupil (with respect to an arbitrary anchor point) and known locations on the screen. The mapping is then used to interpolate between the known target locations in order to estimate the point of interest in the gazed scene. This approach is often used in commercial eye trackers, using high resolution images of the eyes and infrared anchor points. However, this approach forces the user to use a chin rest to avoid head movements which will results in wrong mappings.



(a) (b) (c)

Figure 3.11: (a) The construction of the calibration plane, (b) the intersection of the calibration points on known target plane points, (c) the effect on the known points on the target plane while moving the head in the 3D space.

In this work, instead of focusing on modeling the shape of the eyes or the mapping between their displacement vectors, we make the assumption that the visual field of view is only defined by the head pose and that the point of interest (defined by the eyes) does not fall outside the head-pose-defined field of view. This assumption corresponds to the study of [99], where it is shown that head

pose contributes to about 70% of the visual gaze. Here, we make the observation that the calibration step is not directly affected by the head position. For example, when the calibration is performed while the head is slightly rotated and/or translated in space, the mapping is still able to compute the gazed location by interpolating between known locations (as long as the head position does not vary). In this way, the problem of 3D gaze estimation is reduced to the subproblem of estimating it in 2D (*e.g.* using eyes only), removing the constraints on head movements.

Instead of learning all possible calibrations in 3D space, we propose to automatically retarget a set of known points on a target plane (*e.g.* a computer screen) in order to simulate a re-calibration each time the user moves his/her head. In fact, if the known points are translated accordingly to the parameters obtained from the head pose, it is possible to use the previously obtained displacement vectors and re-calibrate using the new known points on the target plane. To this end, a *calibration plane* is constructed, which is attached to the front of the head as in Figure 3.11 (a), so that it can be moved using the same transformation matrix obtained from the head pose estimator (to ensure that it moves accordingly). The calibration plane is then populated during the calibration step, where the user is requested to look at a known set of points on the target plane. The ray between the center of the head and the known point on the target plane is then traced until the calibration plane is intersected. In this way, the relation between the calibration plane and the target plane (*e.g.* a computer screen) is also computed.

Since the calibration points are linked to the head-pose-constructed visual field of view, their locations will change when the head moves in the 3D space in front of the target plane. Hence, every time that the head moves, the intersection points between the ray going from the anchor point to the calibration point are computed in order to construct the new set of known points on the target plane. Using this new set of known points and the known pose-normalized displacement vectors as collected during the calibration phase, it is possible to automatically recalibrate and learn a new mapping. Figure 3.11(b) shows how the calibration points are projected on the calibration plane and Figure 3.11 (c) illustrates how these points change during head movements, obtaining new intersections on the target plane (the *Pose-Retargeted* known points).

3.5 Experiments

Here, three components need an independent evaluation: (1) the accuracy provided by the eye center location given the head pose, (2) the accuracy obtained by the head pose estimation given the eye center location and (3) the accuracy of the combined final visual gaze estimation. In the following sections, the datasets, error measures, and the result for each of three components are described and discussed.

3.5.1 Eye Location Estimation

The performance obtained by using head pose cues in eye location are evaluated using the Boston University head pose database [17]. The database consists of 45 video sequences, where 5 subjects were asked to perform 9 different head motions under uniform illumination in a standard office setting. The head is always visible and there is no occlusion except for some minor self-occlusions. Note that the videos are in low resolution (320×240 pixels), hence the iris diameter roughly corresponds to 4 pixels.

A Flock of Birds tracker records the pose information coming from the magnetic sensor on the person's head. This system claims a nominal accuracy of 1.8 mm in translation and 0.5 degrees in rotation. However, Cascia *et al*. [17] have experienced a lower accuracy due to the interfering electromagnetic noise in the operating environment. Nonetheless, the stored measurements are still reliable enough to be used as ground truth. As no annotation of the eye location on this dataset is available, we manually annotated the eyes of the subjects on 9000 frames. These annotations are publicly available at [108].

In quantifying the error, we used the 2D *normalized error*. This measure was introduced by Jesorsky *et al*. [52] and is widely used in eye location literature [7, 16, 43, 106, 122]. The normalized error represents the error obtained by the worse eye estimation and is defined as:

$$e = \frac{\max(d_{left}, d_{right})}{d}, \tag{3.3}$$

where d_{left} and d_{right} are the Euclidean distance between the located eyes and the ones in the ground truth, and d is the Euclidean distance between the eyes in the ground truth. For this measure, $e \leq 0.25$ (a quarter of the interocular distance) corresponds roughly to the distance between the eye center and the

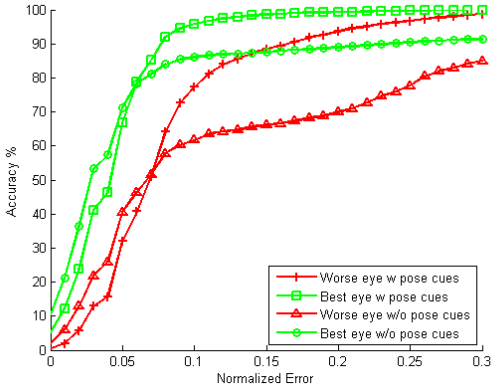


Figure 3.12: A comparison between the eye detection results with and without pose cues

Table 3.1: Effect of pose cues in eye localization

	Worse eye		Best eye	
	Without pose	With pose	Without pose	With pose
$e \leq 0.05$	40.6	31.93	66.78	71.27
$e \leq 0.1$	61.73	77.27	86.03	95.81
$e \leq 0.15$	66.14	88.46	87.87	98.6
$e \leq 0.2$	70	93.67	93.67	99.29
$e \leq 0.25$	77.72	96.74	96.74	99.73

eye corners, $e \leq 0.1$ corresponds to the range of the iris, and $e \leq 0.05$ corresponds the range of the cornea. In order to give upper and lower bounds to the accuracy, in Figure 3.12 we also show the *minimum normalized error*, obtained by considering the best eye estimation only.

The accuracy achieved by the proposed unified approach is presented in Figure 3.12 together with the baseline accuracy obtained by the standard eye locator [109]. In the latter, the approximate face position is estimated using the boosted cascade face detector proposed by Viola and Jones [114], where the rough positions of the left and right eye regions are estimated by anthropometric relations [41]. For the cases in which the face cannot be detected, the maximum possible localization error is assigned considering the limits of the detected face and anthropometric measures as follows. The maximum achievable error is assumed to be half of the interocular distance, which corresponds to 0.5. Therefore, a default error value of 0.5 is assigned to both eyes for the frames in which a face is not detected. In our experiments, the faces of the subjects were not detected in 641 frames, which corresponds to 7.12% of the full

dataset. The working range of the face detector is around 30° around each axis, while certain head poses in the dataset are larger than 45°. The accuracy is represented in percentages for a normalized error of range $[0, 0.3]$. A performance comparison is provided for the best and worse eye location estimations, where certain precise values are also given in Table 3.1 for several normalized error values.

From Figure 3.12, it is shown that the pose cues improve the overall accuracy of the eye detector. In fact, for an allowed error larger than 0.1, the unified scheme provides an improvement in accuracy from 16% to 23%. For smaller error values, the system performs slightly worse than the standard eye locator. The eye detection results obtained by using pose cues depict a significant overall improvement over the baseline results. However, we note a small drop in accuracy for precise eye location ($e \leq 0.05$). This is due to interpolation errors occurring while sampling and remapping the image pixels to pose-normalized eye regions. In fact, as shown in Figure 3.6, in specific extreme head poses, the sampled eye may not appear as completely circular shapes due to perspective projections. Therefore, the detection is shifted by one or two pixels. Given the low resolution of the videos, this shift can easily bring the detection accuracy beyond the $e \leq 0.05$ range. However, given the low resolution, this error is barely noticeable.

3.5.2 Head Pose Estimation

Table 3.2: Comparison of RMSE and STD

	Fixed template				Updated template				Sung	An
	With eye cues		Without eye cues		With eye cues		Without eye cues		*et al.* [100]	*et al.* [2]
	RMSE	STD	RMSE	STD	RMSE	STD	RMSE	STD	RMSE	RMSE
Pitch (ω_x)	5.26	4.67	6.00	5.21	5.57	4.56	5.97	4.87	5.6	7.22
Yaw (ω_y)	6.10	5.79	8.07	7.37	6.45	5.72	6.40	5.49	5.4	5.33
Roll (ω_z)	3.00	2.82	3.85	3.43	3.93	3.57	4.15	3.72	3.1	3.22

Since the ground truth is provided by the Boston University head pose database [17], it is also used to evaluate the effect of using eye location cues in head pose estimation. To measure the pose estimation error, the root mean square error (RMSE) and standard deviation (STD) values are used for the three planar rotations: ω_x, ω_y and ω_z.

To measure the accuracy of pose, two scenarios are considered: in the first scenario, the template is created from the first frame of the video sequence and is kept constant for the rest of the video; in the second scenario, the template is

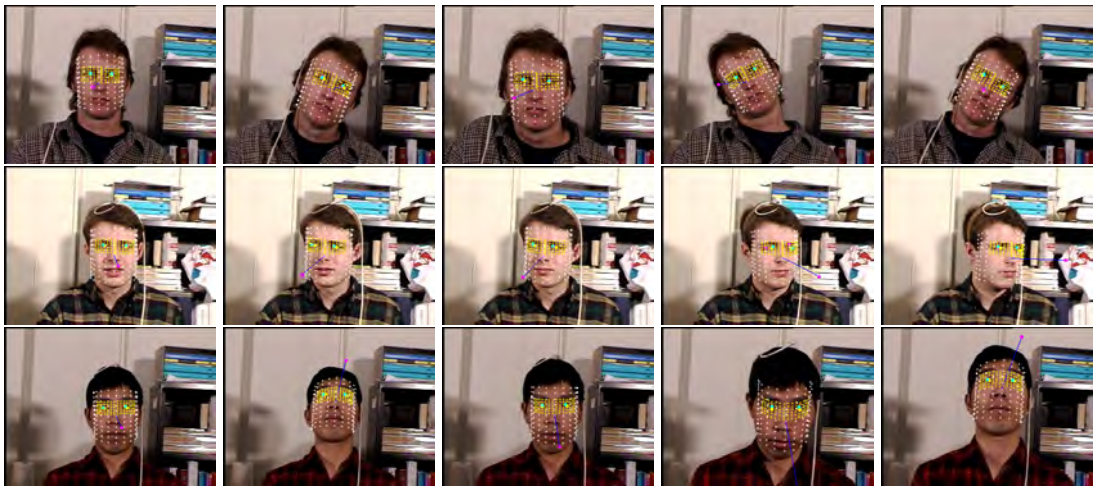


Figure 3.13: Qualitative examples of result on roll, yaw and pitch angles on videos showing extreme head poses

updated at each frame, so that the tracking is always performed between two successive frames. Table 3.2 shows the improvement in RMSE and STD given by using eye location cues in both scenarios. Note that, without using the eye cues, the updated template gives the best results. On the other hand, if the eye cues are considered, the accuracy of the fixed template becomes better than the updated one. This due to the fact that by using the eye cues while updating the template might introduce some errors at each update, which cannot be recovered at later stages. However, for both scenarios, the use of eye cues presents an improvement in estimation of the pose angles. Some challenging examples of the results obtained by our implementation of the CHM head pose tracker are represented in Figure 3.13 for challenging roll, yaw and pitch rotations. The graphs with values for the ground truth and for the accuracy of the tracker for the respective videos are shown in Figure 3.14. It can be derived that the system is able to cope with these extreme head poses.

In the last two columns of Table 3.2, we compare our results with two other methods in the literature, which use the same database. Similar to our method, Sung *et al*. [100] propose a hybrid approach combining active appearance models and cylinder head models to extend the operating range of AAM. An *et al*. [2] propose to replace the traditional CHM with a simple 3D ellipsoidal model. They provide comparison of accuracy with planar and cylindrical models. Here, we consider the accuracy reported by Sung *et al*. and from An *et al*. on the cylindrical head model [2]. From Table 3.2, it is shown that our method provides comparable or better results with respect to the compared methods.

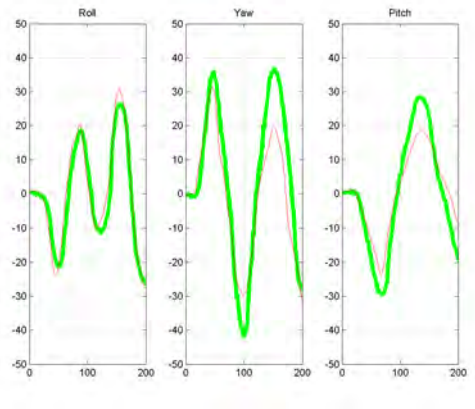


Figure 3.14: Examples of quantitative result on roll, yaw and pitch angles on videos showing extreme head poses. The ground truth is shown in green, the tracking results in red

Hence, our experiments show that using eye cues has an overall positive effect on the average RMSE. However, it is important to note that by enhancing the head tracker using the eye cues to fix the transformation matrix does not have a direct effect on the accuracy. The main effect is obtained by the re-initialization of the cylinder in a position which allows for a correct convergence once the pose tracker converges to a local minimum. In fact, by closely analyzing the results it can be derived that by using the eye cues the accuracy of the pose is decreased for particular subjects showing extreme head poses.

This issue is related to the approach used to fix the transformation matrix. In our approach, we assume that the eye located given the correct pose are the correct ones, but this will not be true in the presence of highlights, closed eye or very extreme head poses (*e.g.* when the head is turned by 90° and only one eye is visible). In these specific cases, averaging by the transformation matrix suggested by the eye location might negatively affect an otherwise correct transformation matrix given by the head tracker. Fortunately the eye locator can be considered quite accurate and therefore these cases do not occur very often, and the track is recovered as soon as the difficult condition is resolved or a semi-frontal face is detected again.

3.5.3 Visual Gaze Estimation

This section describes the experiments performed to evaluate the proposed gaze estimation system. To this end, a heterogeneous dataset was collected, which

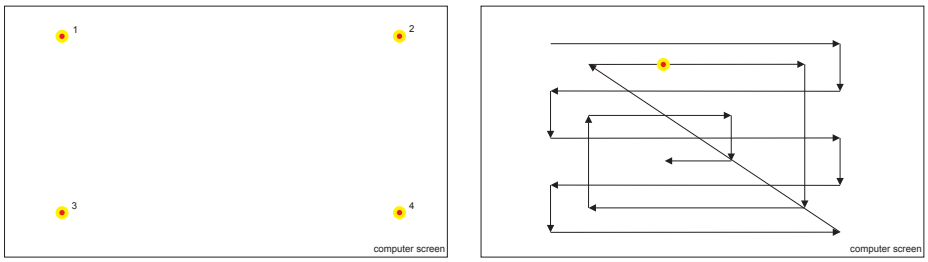
includes 11 male and female subjects with different ethnicity, with and without glasses and different illumination conditions. Figure 3.15 shows some examples of the subjects in the dataset. The data was collected using a single webcam and without the use of a chin-rest. The subject sits at a distance of $750mm$ from the computer screen and the camera. The subject's head is approximately in the center of the camera image. The resolution of the captured images is 720×576 pixels and the resolution of the computer screen is 1280 $\times$ 1024 pixels. To test the system under natural and extreme head movements, the subjects were requested to perform two set of experiments:

Figure 3.15: Some of the test subjects

(1) The first task, named *static dot gazing*, is targeted at evaluating how much the head pose can be compensated by the eye location. The subjects are requested to gaze with their eyes at a static point on the computer screen (see Figure 3.16(a)) and move their head around while still looking at the specific point. The point is displayed at certain locations on the screen for about 4 seconds each time. When the point is displayed on the screen, the subject is asked to look at it and then to rotate his/her head towards the point's location. When the desired head position is reached the subjects are asked to move their head while their eyes are still gazing at the displayed point. The location and the order in which the points are displayed is shown in Figure 3.16(a). (2) The second task, named *dot following*, is targeted at evaluating the gaze estimation performance while following a dot on the screen. The test subjects are requested to look and follow a moving point on the screen in a natural way, using their eyes and head if required. The path followed by the dot is shown in Figure 3.16(b).

The ground truth is collected by recording the face of the subject and the corresponding on-screen coordinates where the subjects are looking.

In order to test the performance of the proposed approach, three different meth-



(a) (b)

Figure 3.16: The two tasks performed during the data collection: (a) dot gazing and (2) dot following

ods are tested:

- **Eyes Only Gaze Estimator**: This estimator uses the anchor-pupil vectors directly into the mapping as in the system proposed in [111]. Hence, when the user moves his head from the calibration position, the mapping is bound to fail. This experiment is performed to evaluate the limitations of the classic mapping approaches in presence of head movements.
- **Pose-Normalized Gaze Estimator**: This estimator uses the information about the position of the user's head to pose-normalize the anchor-pupil vectors. During the calibration step, the displacement vectors between the anchor point and the location of the eyes are calculated from the pose-normalized cylindrical head model. These vectors are used together with the coordinates of the corresponding points on the computer screen for training. Then, the estimator approximates the coefficients of the underlying model by minimizing the error measure of misfit of the generated estimates by a candidate model and the train data. When a certain threshold is reached, the model is accepted and used for the estimation of the point of interest when a future displacement vector is constructed;
- **Pose-Retargeted Gaze Estimator**: This is the approach proposed in Section 3.4.2, which treats the 3D gaze estimation problem as a superset of 2D problems. Also this estimator uses pose-normalized displacement vectors. The main difference between the Pose-Retargeted estimator and the Pose-Normalized one is that, when the user moves his/her head, the set of known points points is retargeted using the head pose information. The new coefficients of the underlying model are then approximated and used for the estimate of the new point of interest.

Table 3.3 shows the mean errors of the three gaze estimators in the first task (static dot gazing) for each of the tested subjects. Due to the big changes in

Subject	Eyes Only		Pose-Normalized		Pose-Retargeted	
	Mean	Std	Mean	Std	Mean	Std
1	688.91, 281.29	1090.71, 552.12	338.97, 366.42	181.04, 280.57	186.11, 223.51	191.32, 212.11
2	633.08, 187.74	464.23, 143.99	310.17, 226.07	182.89, 201.83	134.91, 191.77	197.01, 153.41
3	2285.11, 301.45	4929.71, 436.26	359.97, 246.45	184.91, 216.05	161.21, 255.71	201.04, 168.91
4	1202.11, 2664.01	2537.21, 7280.43	346.56, 330.79	164.11, 246.59	190.12, 164.84	199.37, 154.58
5	1388.72, 276.79	1073.91, 630.76	428.04, 327.46	222.63, 287.75	239.25, 215.67	200.78, 225.44
6	874.85, 239.81	726.24, 491.89	429.17, 265.61	215.31, 190.13	232.84, 234.02	175.45, 177.93
7	710.24, 328.63	443.08, 224.25	449.44, 217.21	240.39, 175.33	242.87, 162.01	188.21, 140.37
8	666.58, 257.17	376.31, 194.65	397.56, 236.25	226.65, 181.82	196.38, 152.45	191.22, 131.81
9	623.68, 316.73	412.11, 209.09	395.59, 337.47	206.61, 246.04	204.87, 220.94	197.46, 202.01
10	750.93, 332.06	947.91, 1462.83	430.95, 319.41	247.44, 263.63	272.23, 223.17	231.42, 205.61
11	924.62, 398.24	2297.01, 297.93	443.03, 580.46	229.99, 412.91	252.93, 320.81	186.16, 255.22

Table 3.3: Mean pixel error and standard deviation comparison on the static dot gazing task

head pose while keeping the eyes fixed, the Eyes Only estimator has a significantly larger error and standard deviation with respect to the other methods which include pose normalized displacement vectors. The Pose-Normalized estimator, in fact, has a mean error of (393.58, 313.96) pixels, corresponding to an angle of ($8.5°$, $6.8°$), while the Pose-Retargeted estimator has a mean error of (210.33, 214.99) pixels, corresponding to an angle of ($4.6°$, $4.7°$) in the x and y direction, respectively. The proposed Pose-Retargeted estimator improves the method with a factor of approximately 1.87 in x direction and a factor of about 1.46 in y direction compared to the Pose-Normalized system.

Subject	Eyes Only		Pose-Normalized		Pose-Retargeted	
	Mean	Std	Mean	Std	Mean	Std
1	3461.68, 938.55	1931.83, 567.68	238.88, 112.91	159.53, 69.42	75.95, 117.01	71.02, 76.82
2	3125.42, 361.55	5874.41, 253.68	229.78, 104.72	137.44, 73.58	79.16, 115.87	58.79, 82.01
3	3531.19, 564.11	7725.46, 353.82	253.51, 103.87	162.11, 77.08	78.73, 128.01	67.48, 77.07
4	2380.21, 400.89	2002.35, 608.31	277.71, 134.53	180.27, 139.32	99.29, 115.16	108.59, 150.38
5	3554.94, 656.51	2799.42, 468.44	268.51, 105.09	165.54, 77.78	85.77, 101.13	78.03, 72.79
6	2365.84, 472.37	1574.86, 336.32	254.63, 95.47	165.61, 62.83	80.25, 78.27	78.67, 57.13
7	3606.85, 729.86	3414.06, 1730.97	282.74, 101.62	179.79, 76.36	93.81, 104.64	84.21, 81.32
8	3332.96, 573.66	6989.07, 625.28	278.79, 188.84	189.94, 137.01	92.25, 92.77	85.31, 65.44
9	11958.67, 775.88	21508.32, 752.94	250.25, 200.03	180.95, 139.38	92.52, 99.32	72.27, 71.75
10	5082.26, 731.92	9972.58, 719.33	295.22, 168.61	190.91, 115.41	91.92, 92.21	86.09, 59.38
11	8482.27, 693.31	13728.11, 832.69	303.83, 179.69	201.23, 162.51	89.36, 98.12	109.38, 138.84

Table 3.4: Mean pixel error and standard deviation comparison on the dot following task

Table 3.4 shows the results of the second task (dot following). In this task, due to the fact that the head significantly shifts from the calibration position to allow the eyes to comfortably follow the dot on the screen, the mapping in the Eyes Only estimator completelly fails. However, the Pose-Normalized estimator achieves a mean error of (266.71, 135.94) pixels, which corresponds to

an angle of ($5.8°$, $3.0°$), while the Pose-Retargeted estimator has a mean error of (87.18, 103.86) pixels, corresponding to an angle of ($1.9°$, $2.2°$) in the x and y direction, respectively. When compared to the Pose-Normalized estimator, the Pose-Retargeted estimator improves the accuracy with a factor of approximately 3.05 in x direction and with a factor of about 1.31 in y direction.

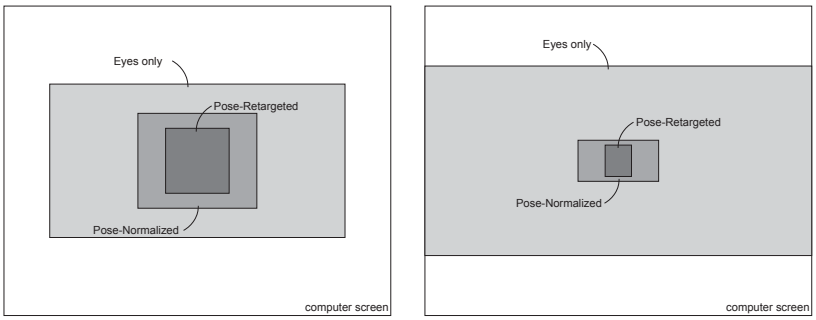
The differences between the accuracy obtained by the different systems in both tasks is visually represented in Figure 3.17.

Although the average error obtained by the proposed system seems high at first, one should consider that the human fovea covers $\sim 2°$ of the visual field, in which everything can be seen without requiring a saccade. Therefore, when asking a subject to gaze at a specific location, there is always an inherent error on the gaze ground truth. In fact, assuming that the test subjects are sitting at a distance of $750mm$ from the computer screen, the projection of the foveal error $\epsilon_f = 2°$ on the target plane corresponds to a window of about 92 $\times$ 92 pixels, which is in the same magnitude of the results obtained by the proposed system. By analyzing the causes for the errors (and the big standard deviation) we note that, in most cases, the results in the y direction are worse than the results in x direction. There are two main reasons for this: (1) the camera is situated on top of the computer screen so when the test subject is gazing at the bottom part of the screen, the eyelids obscure the eye location and significant errors are introduced by the eye locator, (2) the eyes move less in y direction than in x direction. Futhermore, errors in the eye center locator seriously affect the system, as just a few pixels error on the eye estimation result in significant displacements at a distance of $750mm$.

However, it is clear that the proposed Pose-Retargeted estimator outperforms the other tested approaches in all the experiments, while the Pose-Normalized estimator cleary outperforms the method based on eyes only. This clearly indicates that it is beneficial to combine head pose and eye information in order to achieve better, more natural and accurate gaze estimation systems.

3.6 Conclusions

In this chapter, we proposed a deep integration of a CHM based head pose tracker and an isophote based eye locator in a complementary manner, so that both system can benefit from each other's evidence. Experimental results showed that the accuracy of both independent systems is improved by their combination. The eye location estimation of the unified scheme achieved an improve-



(a) (b)

Figure 3.17: Errors for the three tested estimators compared to the computer screen for the (a) static dot gazing task and (b) dot following task

ment in accuracy from 16% to 23%, while the pose error has been improved from 12% to 24%. Besides the improvements in accuracy, the operating range of the eye locator has been extended (more than $15°$) by the head tracker and the ineffectiveness of the previously reported eye location methods against extreme head poses was compensated. Furthermore, automatic quality control and re-initialization of the head tracker was provided by the integration of the eye locator, which helps the system in recovering to the correct head pose. Consequently, the proposed unified approach allows for an autonomous and self-correcting system for head pose estimation and eye localization. Finally, the information obtained by the proposed system is combined in order to project the visual gaze of a person on the target scene by retargeting a set of known points using the head pose information. The evaluation using the collected dataset proved that joint eye and head information results in a better visual gaze estimation, achieving a mean error between $2°$ and $5°$ on different tasks without imposing any restraints on the position of the head.

4

Image Saliency by Isocentric Curvedness and Color[1]

4.1 Introduction

Visual saliency is a very important part of our vision: it is the mechanism that helps in handling the overload of information that is in our visual field by filtering out the redundant information. It can be considered as a measure determining to what extent an image area will attract an eye fixation. Unfortunately, little is known about the mechanism that leads to the selection of the most interesting (salient) object in the scene such as a landmark, an obstacle, a prey, a predator, food, mates etc. It is believed that interesting objects on the visual field have specific visual properties that makes them different than their surroundings. Therefore, in our definition of visual saliency, no prior knowledge or higher-level information about objects is taken into account.

In this chapter, we are interested in the computational simulation of this mechanism, which can be used in various computer vision scenarios, spanning from algorithm optimization (less computation spent on uninteresting areas in the image) to image compression, image matching, content-based retrieval, etc.

General *context-free* (*i.e.* without prior knowledge about the scene) salient point detection algorithms aim to find distinctive local events in images by focusing on the detection of corners, edges [45, 121, 94, 70] and symmetry [101, 71, 46].

[1]Published as R. Valenti, N.Sebe, and T. Gevers, "Image Saliency by Isocentric Curvedness and Color", IEEE International Conference on Computer Vision, 2009.

These methods are very useful to find locally salient points, however globally salient regions are usually computed by partitioning the images into cells and by counting the number of salient descriptors which fall into them. The above techniques lack the ability to infer the location of global structures as an agreement of multiple local evidences. Therefore, to infer global salient regions, we propose a framework that combines isophotes properties (Section 4.2.1) with image curvature (Section 4.2.2) and color edges information (Section 4.2.3). The contributions are the following:

- Instead of using the widely adopted edge information, we propose to use the gradient slope information to detect salient regions in images.
- We use an isophote symmetry framework to map local evidence close to the centers of image structures.
- We provide an enabling technology for smarter, saliency aware, segmentation algorithms.
- We solve the problem of defining the size of unknown interesting objects by segmentation and subwindow search.

4.2 The Saliency Framework

By analyzing human vision and cognition, it has been observed that visual fixations tend to concentrate on corners, edges, along lines of symmetry and distinctive colors. Therefore, previously proposed saliency frameworks in the literature [68, 51, 33, 60, 76, 103, 95] often use a combination of intensity, edge orientation and color information to generate saliency maps. However, most interest detectors focus on the shape-saliency of the local neighborhood, or point out that salient points are “interesting" in relation to their direct surroundings. Hence, salient features are generally determined from the local differential structure of images.

In this work, the goal is to go from the local structures to more global structures. To achieve this, based on the observation that the isophote framework (previously proposed in [109] for eye detection) can be generalized to extract generic structures in images, we propose a new isophote-based framework which uses additional color edges and curvature information. As with many other methods proposed in the literature, our approach is inspired by the feature integration theory [104]. Therefore, we compute different salient features which are later combined and integrated into a final saliency map. In the next sections, the

principles of each of the used salient features and how they are combined are described.

4.2.1 Isocentric Saliency

Isophotes are lines connecting points of equal intensity (curves obtained by slicing the intensity landscape). Since isophotes do not intersect each other, an image can be fully described by its isophotes both on its edges and on smooth surfaces [67]. Furthermore, the shape of each isophote is independent of changes in the contrast and brightness of an image. Due to these properties, isophotes have been successfully used as features in object detection and image segmentation [67, 56]. To formulate the concept of isophote, a local coordinate system is defined at every point in the image, which points in the direction of gradient. Let the gauge coordinate frame be $\{v, w\}$, the frame vectors can be defined as

$$\hat{w} = \frac{\{L_x, L_y\}}{\sqrt{L_x^2 + L_y^2}}, \quad \hat{v} =\perp \hat{w},$$

where L_x and L_y stand for the first-order derivatives of the luminance function $L(x, y)$ in the x and y dimensions, respectively. Since by definition there is no change in intensity on an isophote, the derivative along v is 0, whereas the derivative along w is the gradient itself. Thus, an isophote is defined as $L(v, w(v)) = constant$.

At each point of the image, we are interested in the displacement of the center of the osculating circle to the isophote, which is assumed to be not far away from the center of the structure to which the isophote belongs. Knowing that an isophote is a curvilinear shape, the isophote curvature, κ, is computed as the rate of change, w'', of the tangent vector, w'. In Cartesian coordinates, this is expressed as:

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

The magnitude of the vector (radius) is simply found as the reciprocal of the above term. The information about the orientation is obtained from the gradient, but its direction indicates the highest change in luminance. The duality of the isophote curvature is then used in disambiguating the direction of the vector: since the sign of the isophote curvature depends on the intensity on the

outer side of the curve, the gradient is simply multiplied by the inverse of the isophote curvature. Since the gradient is $\frac{\{L_x,L_y\}}{L_w}$, the displacement coordinates $D(x,y)$ to the estimated center are obtained by

$$\begin{aligned} D(x,y) &= \frac{\{L_x,L_y\}}{L_w}\left(-\frac{L_w}{L_{vv}}\right) = -\frac{\{L_x,L_y\}}{L_{vv}} \\ &= -\frac{\{L_x,L_y\}(L_x^2+L_y^2)}{L_y^2L_{xx}-2L_xL_{xy}L_y+L_x^2L_{yy}}. \end{aligned} \tag{4.1}$$

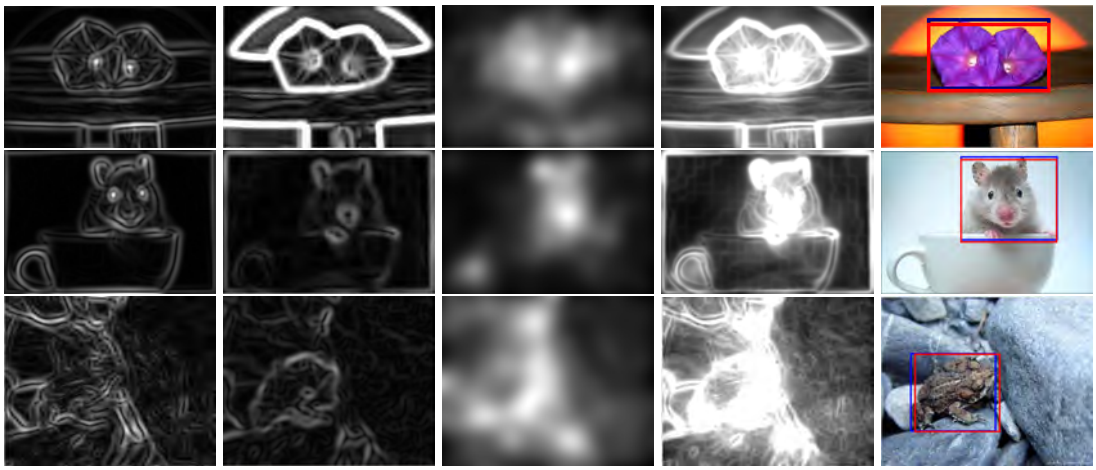
In this manner every pixel in the image gives an estimate of the potential structure it belongs to. In order to collect and reinforce this information and to deduce the location of the objects, $D(x,y)$'s are mapped into an accumulator, which is in turn convolved with a Gaussian kernel so that each cluster 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 (isocenters). By applying this framework to natural images, many votes can be affected by noise or are generated by uninteresting cluttered parts of the image. To reduce this effect, each vote is weighted according to its local importance, defined as the amount of image curvature (Section 4.2.2) and color edges (Section 4.2.3).

4.2.2 Curvature Saliency

A number of approaches use edge information to detect saliency [121, 94, 70]. The amount of information contained in edges is limited if compared to the rest of the image. Instead of using the peaks of the gradient landscape, we propose to use the slope information around them. To this end, an image operator that indicates how much a region deviates from flatness is needed. This operator is the curvedness [59], defined as

$$curvedness = \sqrt{L_{xx}^2+2L_{xy}^2+L_{yy}^2}.$$

The curvedness can be considered as a rotational invariant gradient operator, which measures the degree of regional curvature. Since areas close to edges will have a high slope and since isophotes are slices of the intensity landscape, there is a direct relation between the curvedness and the density of isophotes. Hence isophotes with higher curvedness are more appropriate for our goal of mapping from local structures to global structures, as they are likely to follow



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

Figure 4.1: An example of the conspicuity maps and their combination: (a) Curvedness, (b) Color boosting, (c) Isocenters clustering, (d) Combined saliency map, (e) Area with highest energy in the saliency map (red: detection, blue: ground truth).

object boundaries and thus belong to the same shape. An example of the effect obtained by applying the curvedness to natural images can be seen in Figure 4.1(a).

4.2.3 Color Boosting Saliency

While determining salient image features, the distinctiveness of the local color information is commonly ignored. To fully exploit the information coming from the color channels, both shape and color distinctiveness should be taken into account.

The method proposed by van de Weijer *et al*. [112] is based on the analysis of the statistics of color image derivatives and uses information theory to boost the color information content of a color image. Since color boosting is derivative based and its outcome is enhanced color edges, the method can easily be integrated in our framework. According to information theory, rare events are more important than normal events. Therefore, the quantity of information I of a descriptor v with probability $p(v)$ can be defined as

$$I(v) = -\log(p(v)).$$

In order to allow rare color derivatives to have equal values, image derivatives are mapped to a new space using a color saliency boosting function g. The function g can be created by analyzing the distribution of image derivatives. In fact, it can be derived that the shape of the distribution is quite similar to an ellipse that can be described by the covariance matrix M. This matrix can be decomposed into an eigenvector matrix U and an eigenvalue matrix V. The color boosting function g will be the transformation with which the ellipses are transformed into spheres: $g(L_x) = V^{-1}U^T L_x$, where the eigenvectors U are restricted to be equal to the opponent color space, and $V = diag(0.65, 0.3, 0.1)$ as was found in the distribution of the data in the Corel dataset [112]. After the mapping, the norm of the image derivatives is proportional to the information content they hold. An example of the effect obtained by applying this operator to natural images can be seen in Figure 4.1(b).

4.3 Building the Saliency Map

In this section, the previously described saliency features are combined into a saliency framework. Since all the features sections make use of image derivatives, their computation can be re-used in order to lower the computational costs of the final system. Furthermore, the three features were selected as both curvedness and color boosting enhance edges, and isophotes are denser around edges. Therefore, the saliency features can be nicely coupled together to generate three different conspicuity maps (Figure 4.1(a), (b) and (c)): At first, the maps obtained by the curvedness and the color boosting are normalized to a fixed range $[0, 1]$ so that they can easily be combined. The linear combination of these maps is then used as weighting for the votes obtained from Eq. 4.1 to create a good isocenter clustering of the most salient objects in the scene. In this way, the energy of local important structures can contribute to find the location of global important structures. An example of isocenter clustering is given in Figure 4.1(c), obtained by weighting the votes for the isocenters using the curvature and color boosting conspicuity maps in Figure 4.1(a) and (b). The main idea is that if a region of the image is relevant according to multiple conspicuity maps, then it should be salient, therefore the normalized conspicuity maps are linearly combined into the final saliency map (Figure 4.1(d)). However, multiple objects or components could be present in an image and hence receive higher saliency energy from the conspicuity maps. For instance, in the example on the second line of Figure 4.1, the mouse and the handle of the cup are receiving the most of the energy, but what is the real salient object? The full mouse, its face, the cup, the handle or all of them together? This question raises the problem

of scale and size of the object that we are looking for. Depending on the application, the size of the object might be known (*e.g.* the size of the silhouette of a person seen from a specific security camera, the size of the object on which we are performing visual inspection for quality control *etc.*). In the experimental section we will show that, if the size is known, the saliency map obtained with the described procedure is already enough to obtain a good location estimate for the salient object. This is shown in Figure 4.1(e), where the red box represents the area with the maximum energy (the salient region in the image), and the blue box the ground truth annotation. On the other hand, if the size of the object is unknown, additional information about the persistence of the object in scale space and information about its boundaries are required. These topics are discussed in the following sections.

4.3.1 Scale Space

The scale selection is an important problem that must be considered when defining what a salient object is. The scale problem is commonly solved by exhaustively searching for the scale value that obtains the best overall results on a homogeneous dataset. Given the heterogeneity of the size of images and the depicted objects, we want to gain scale independence in order to avoid adjustments of the parameters for different situations.

To increase robustness and accuracy, a scale space framework is used to select the results of the conspicuity maps that are stable across multiple scales. To this end, a Gaussian pyramid is constructed from the original color image. The image is convolved with different Gaussians so that they are separated by a constant factor in scale space. In order to save computation, the image is downsampled into octaves. In each octave the conspicuity maps are calculated at different intervals: for each of the image in the pyramid, the proposed method is applied by using the appropriate $sigma$ as a parameter for the size of the kernel used to calculate image derivatives. This procedure results in two saliency pyramids (Figure 4.2), one retaining the color saliency and the other the curvature saliency. These two pyramids are then combined together with isophote information to form a third saliency pyramid, containing isocentric saliency. The responses in each of the three saliency pyramids are combined linearly, and then scaled to the original image size to obtain a scale space saliency stack. Every element of the saliency stack is normalized and therefore considered equally important, hence they are simply accumulated into a single, final saliency map. The areas with the highest energy in the resulting saliency map will represent the most scale invariant interesting object, which we will consider to be the ob-

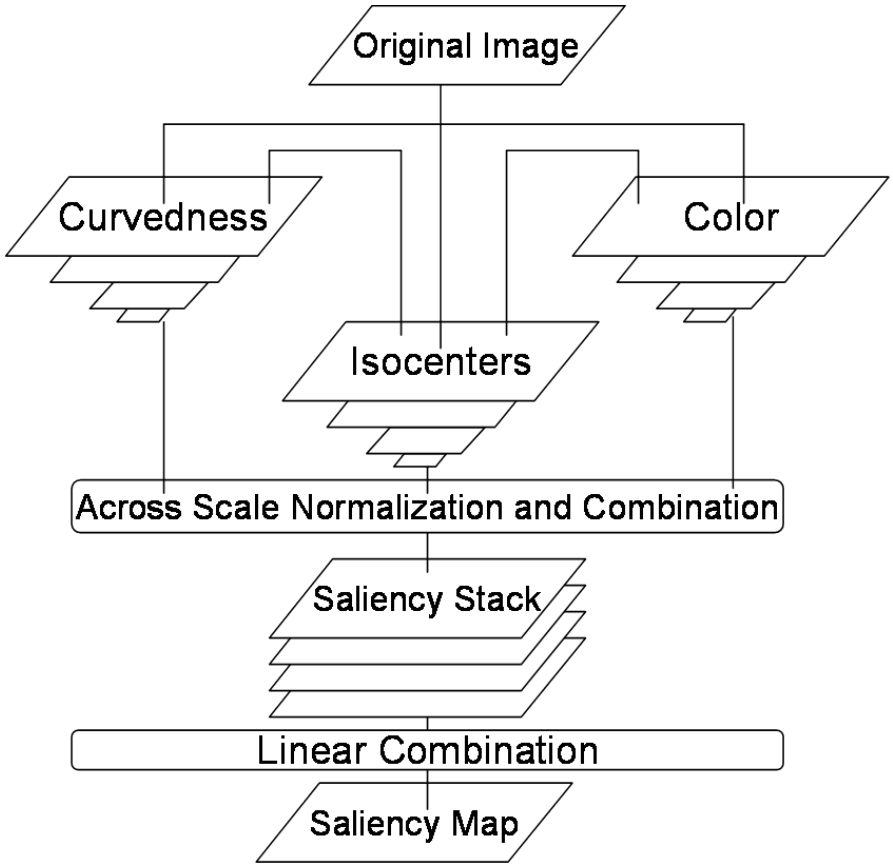


Figure 4.2: The proposed saliency scale space framework

ject of interest in the image.

4.3.2 Graph Cut Segmentation

Although the obtained saliency map has most of its energy at the center of image structures, the successful localization of the most salient object can only be achieved by analyzing its edges. In fact, since curvedness and color boosting are combined together with isocentric saliency in the final saliency map, a great part of the energy in it will still lie around edges. To distribute the energy from the center and the edges of the salient structure to connected regions, a fast and reliable segmentation algorithm is required. The method proposed in [35] addresses the problem of segmenting an image into regions by using a graph-based representation of the image. The authors propose an efficient segmentation method and show that it produces segmentations that satisfy global prop-

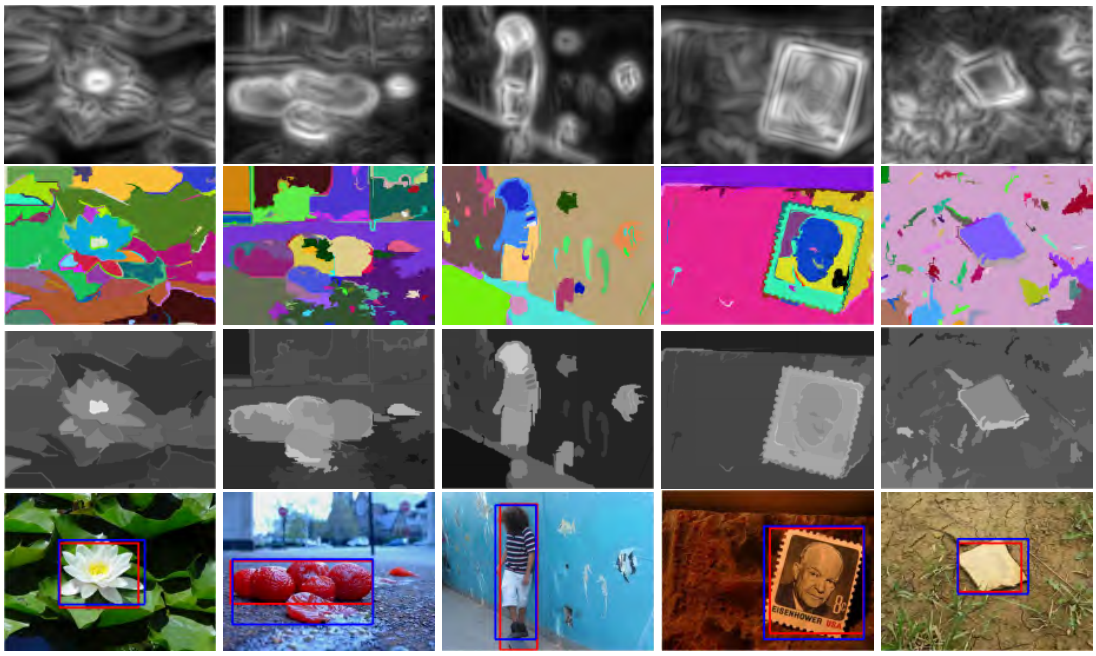


Figure 4.3: An example of the obtained results. From top to bottom: Saliency map, graph-cut segmentation, segmentation weighted by saliency, ESS result (red: detection, blue: ground truth).

erties. This method was chosen because its computational complexity is nearly linear in the number of graph edges and it is also fast in practice. Furthermore, it can preserve the details in low-variability image regions while ignoring them in high-variability regions.

As shown in [107], it is possible to extract a salient object in an image by means of a segmentation algorithm and eye fixations. Since the saliency map obtained by our system can be considered as a distribution of potential fixation points, it can be used to enhance the segmentation algorithms to extract connected salient components. The graph cut segmentation results for a set of images are shown in the second row of Figure 4.3. For each of the segmented components, the average energy covered in the saliency map is computed. Therefore, if the component has higher energy, it will be highlighted in the saliency weighted segmentation.

The third row of Figure 4.3 shows the effect of weighting the segmentation components on second row of Figure 4.3 by the saliency map in the first row of Figure 4.3. Note that, in this case, the brightness indicates the level of saliency of the region (brighter = more salient) and that, if the results are thresholded, it is possible to obtain a binary map of segmented salient regions. This opens

the possibility for saliency based segmentation algorithms that would join segmented components based on their saliency other than their similarity.

4.4 Experiments

In this section, the accuracy of the proposed algorithm and of its components is extensively evaluated on a public dataset.

4.4.1 Dataset and Measures

The used saliency dataset is the one reported in [68]. The dataset contains 5000 high quality images, each of them hand labeled by 9 users requested to draw a bounding box around the most salient object (according to their understanding of saliency). The provided annotations are used to create a saliency map $S = \{s_x | s_x \in [0, 1]\}$ as follows:

$$s_x = \frac{1}{N} \sum_{n=1}^{N} a_x^n$$

where N is the number of users and a_x^n are the pixels annotated by user n. In this way, the annotations are combined into an average saliency map. In order to create a binary ground truth saliency map, only the bounding box of the area annotated by more than four users is kept as annotation s_x of the most salient object in the scene. Given the ground truth annotation s_x and the obtained detection d_x of the salient region in an image, the precision, recall, and F-measure are calculated. The precision and recall measures are defined as:

$$Precision = \frac{\sum_x s_x d_x}{\sum_x d_x} \quad Recall = \frac{\sum_x s_x d_x}{\sum_x s_x}.$$

The F-measure is the weighted harmonic mean of precision and recall, therefore is an overall performance measure. It is defined as

$$F\text{-}measure = \frac{(1 + \alpha) \times Precision \times Recall}{\alpha \times Precision + Recall},$$

where α is set to 0.5 as in [68] and [74]. All measures are then averaged over all the 5000 images in the dataset to give overall figures.

4.4.2 Methodology

The task of determining the location and size of an unknown object in an image is very difficult. The proposed system is tested against two scenarios: one in which the size of the interesting object is known and one where no assumptions on the size are made.

Sliding Window

The purpose of this test is to verify whether or not the location of an interesting object can be retrieved if its scale is known. Therefore, in this scenario, the size of the object is known, and it corresponds to the size obtained from the ground truth. The location of the object in the image, however, is unknown. With the help of integral images (as defined in [114]), the saliency map is exhaustively searched for the region d_x which obtains the highest energy by sliding the ground-truth window over it.

Efficient Subwindow Search

In this scenario, both the size and the location of the relevant object in the image are unknown. To solve this problem, an exhaustive search of all possible subwindows in the saliency map could be performed, in order to retain the one which covers the most energy. However, this would be computationally unfeasible. The Efficient Subwindow Search (EES) is an algorithm which replaces sliding windows approaches to object localization by a branch-and-bound search [64]. It is a simple yet powerful scheme that can extend many existing recognition methods to also perform localization of object bounding boxes. This is achieved by maximizing the classification score over all possible subwindows in the image. The authors show that it is possible to efficiently solve a generalized maximum subwindow problem in many situations. However, even if an efficient search of the best subwindow could be performed, not knowing the size of the object will result in many subwindows with high energy (as in the mouse example in Section 4.3). To obviate this problem, the ESS algorithm is applied on the integral image of the saliency weighted segmentation (third row of Figure 4.3), obtained as described in Section 4.3.2.

Method	Size Known			Size Unknown		
	Precision	Recall	F-measure	Precision	Recall	F-measure
Curvedness	77.55%	77.11%	77.40%	72.47%	50.74%	49.95%
Isocenters	79.95%	79.49%	79.79%	84.23%	66.39%	72.44%
Color	80.91%	80.45%	80.75%	81.63%	37.29%	44.41%
Curvedness + Color	83.79%	83.31%	83.63%	71.50%	71.73%	67.29%
All	85.77%	85.28%	85.61%	84.91%	76.19%	79.19%

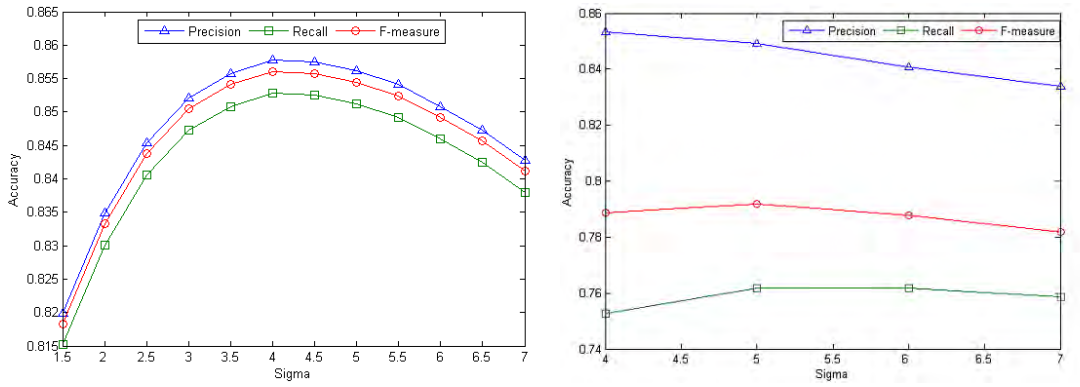
Table 4.1: Contribution of each of the used features and their combination, when the size of the object is known or unknown.

4.4.3 Evaluation

In order to estimate the partial contribution of each of the conspicuity maps to the final saliency map, they are evaluated independently. The sliding window methodology is used to obtain results that are independent of segmentation. Therefore, only the discriminant saliency power of the features is evaluated. By simply using the curvedness, the method can already achieve a good estimate of the salient object in the image (F-measure 77.40%). However, curvedness alone fails in cluttered scenes, as the number of edges will distract the energy from the salient object. The plain isocenters clustering (without weighting) obtains better performance (F-measure 79.79%), similar to color boosting alone (F-measure 80.75%). The linear combination of color boosting and curvedness provides an improved result over the two features considered independently (F-measure 83.63%). This indicates that the two features are somewhat complementary (one succeeds when the other fails). Finally, the proposed combination of all the features achieves the best result (F-measure 85.61%).

In the second scenario, the used edge features are expected to fail as they do not contribute to the center of the components. In fact, curvedness and color boosting achieve an F-measure of 49.95% and 44.41%, while their combination only improves this figure to 67.29%. However, given its capability to distribute the boundary energy to the center of image structures, the isocenter saliency alone has an F-measure of 72.44%. The combination of all the features achieves an F-measure of 79.19%. A summary of the obtained precision, recall and F-measure accuracy for each of the features in both scenarios is shown in Table 4.1.

The graphs in Figure 4.4 show how the accuracy changes with respect to the used standard deviation of the Gaussian kernel ($sigma$) in both scenarios. In the first scenario the $sigma$ parameter can be fine tuned to obtain the best results. Note that, since the size of the object corresponds to the size in the ground truth, there is a relation between precision, recall and F-measure and they are



(a) (b)

Figure 4.4: Effect of changing the standard deviation of the Gaussian kernel when the object size is known (a) and unknown (b).

therefore very similar. In the second scenario, when using the ESS search over the saliency weighted segmentation, changing the parameter has virtually no effect on the accuracy of the system as it has the sole effect of slightly modify the energy in each the segmentation components.

We compared our results with the ones obtained by other methods in the literature: The method from Ma *et al*. [72] uses fuzzy growing, the framework proposed by Itti *et al*. [51] uses multiscale color, intensity and orientation features, and the method from Liu *et al*. [68] uses multi-scale contrast, center-surround histogram and color spatial distribution, combined by a learned Conditional Random Field. To compare the quality of the obtained result with respect to a human performance, we computed the worst performing human annotation with respect to the ground truth annotation (obtained by agreement of at least four subjects). This corresponds to a precision of 87.99%, a recall of 76.74% and an F-measure of 80.29%.

A summary of the precision, recall, and F-measure achieved by the cited methods is displayed in Figure 4.5. Note that our first scenario (column 4) has prior knowledge about the size of the object and is therefore not directly comparable with the other methods. However, without using any prior knowledge (column 5), our method outperforms the classical approaches [72, 51] on the same dataset, while achieving comparable results with the state of the art method [68] without requiring any learning.

Furthermore, as already discussed in [68], the precision measure is much more important in saliency than the recall (*e.g*. if all the image is selected as the salient region, the recall is 100%). In our case, we obtain the highest precision when

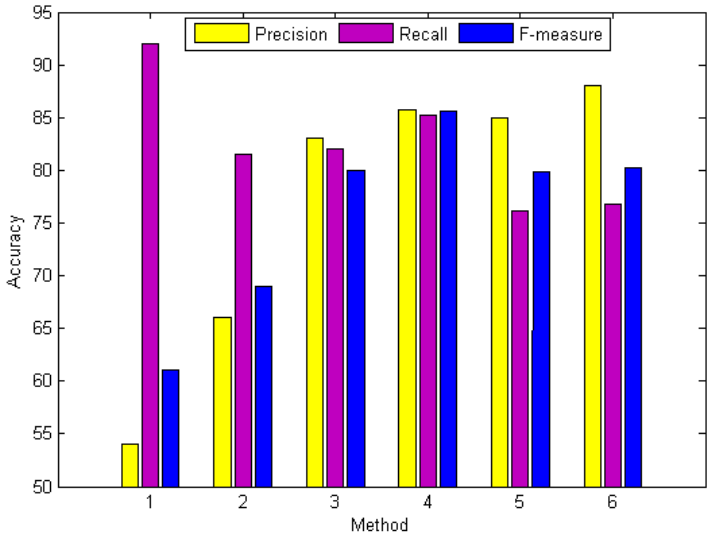


Figure 4.5: A comparison with other methods: 1) Ma *et al*. [72] 2) Itti *et al*. [51] 3) Liu *et al*. [68] 4) Our method on the first scenario (size known) 5) Our method on the second scenario (size unknown) 6) Worst human annotation.

compared to the state of the art methods, while achieving the same F-measure as the best computational methods and as the worst human annotation.

4.4.4 Visually Salient vs. Semantically Salient

In order to discriminate if an object is visually or semantically interesting, we illustrate in Figure 4.6 a qualitative comparison between the obtained saliency map and heat maps obtained by analyzing eye fixations on the same images. By analyzing the painting example in Figure 4.6(a) it is clear that there is a similarity between the eye fixations (second row) and the detected salient regions (third row). It can be seen that, even if they are equally visually salient, the subject appears to mainly focus on faces as they are more semantically salient and less on lower areas, which hold less semantic information (like knees). The same reasoning can be done for the website example in Figure 4.6(b): while every line of the navigation menu is equally salient, the subject focuses only on the top entries. Also, while the items in the middle of the page have similar visual saliency, the user seems to focus only on few of them. This is a clear difference between visual saliency and higher levels of reasoning (*e.g*. knowledge and interest), which can be used to understand if an object is semantically interesting versus visually interesting. It can be seen, however, that eye fixations are

(a) (b)

Figure 4.6: A comparison with eye fixations. From top to bottom: Original image, recorded eye fixations, saliency map obtained by our method superimposed to the original image.

always directly related with salient regions in the image. Therefore, if eye fixation information is available, our method could be used to differentiate between salient regions and the subject's interest. This information can be very valuable as it could be used in a multitude in applications (*e.g.* to tailor user interfaces or commercial ads in minimizing the possible elements of distraction in the visual field).

4.4.5 Discussion

By analyzing our results, we found that the main reason for the low recall lies on the manner that the dataset is annotated: by considering the fruit example in the second column of Figure 4.3, it can be seen that the detected region is smaller than the annotated one. However, the saliency weighted segmentation of the

salient object is nearly optimal. By including this last part of the object in the detected subwindow, a big part of the background would get included as well, lowering the overall energy covered by the subwindow, which in turn would be discarded by the ESS algorithm. The same happens in many of the images in the dataset, explaining our low recall: appendices of object are often not considered as they would decrease the overall energy in the detected subwindow.

As the proposed system focuses on images, we are aware that it is not suitable for locating the salient object in all situations, especially involving changes and movements (as in [73]). Given the flexibility of the framework, the conspicuity maps from other saliency operators can easily be added to the system in order to cover additional saliency cues. However, this will probably require some learning to correctly integrate the different conspicuity maps, and thereby reduce the attractiveness of our method as, contrary to other systems, the actual creation of the saliency map is computationally fast (only a combination of few image derivatives is needed) and does not require any training.

4.5 Conclusions

In this chapter, we have presented a computational bottom-up model to detect visual saliency in common images. The method is based on the assumption that interesting objects on the visual field have specific structural properties that makes them different than their surroundings, and that they can be used to infer global important structures in the image.

The system performs well as it is able to correctly locate or give maximum energy to the same object annotated by humans with an F-measure of 85.61% if the size of the object is known. If the size of the object is unknown, our method is used to enhance a segmentation algorithm. An efficient subwindow search on the saliency weighted segmentation shows that the algorithm can correctly locate an interesting object with an F-measure of 79.19%, while keeping a high precision. The obtained results are very promising as they match the worst human annotation. Furthermore, since no learning is required but only calculation of image derivatives, the system is fast and it can be used as a preprocessing step in many other applications.

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.

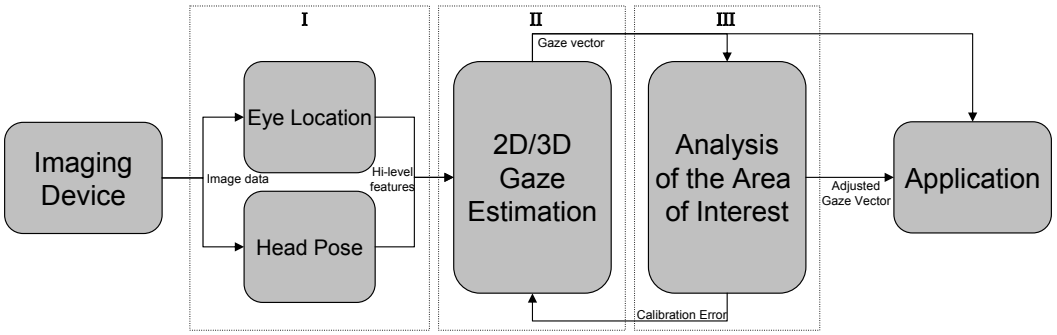


Figure 5.1: The visual gaze estimation pipeline, extended as proposed in this chapter.

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

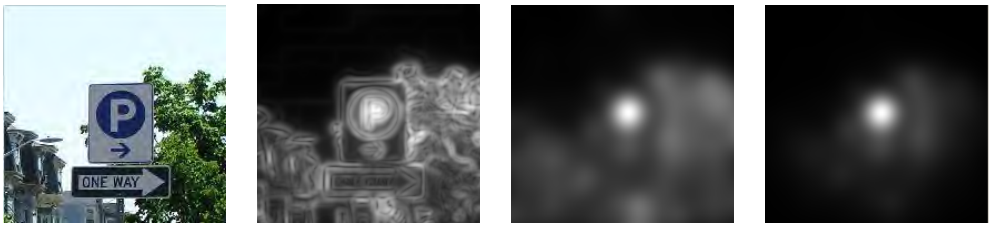
This error is attributed to the resolution of the visual gaze starting from the features extracted in the first component. Eye gaze trackers often use a mapping between the position of the eye and the corresponding locations on the screen. Therefore, the tracking system needs to be calibrated. In case 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

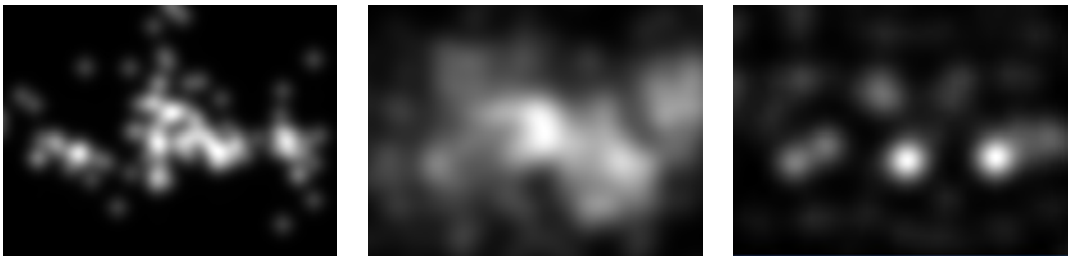
Table 5.1: Correlation results for the eye gaze tracker experiment

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

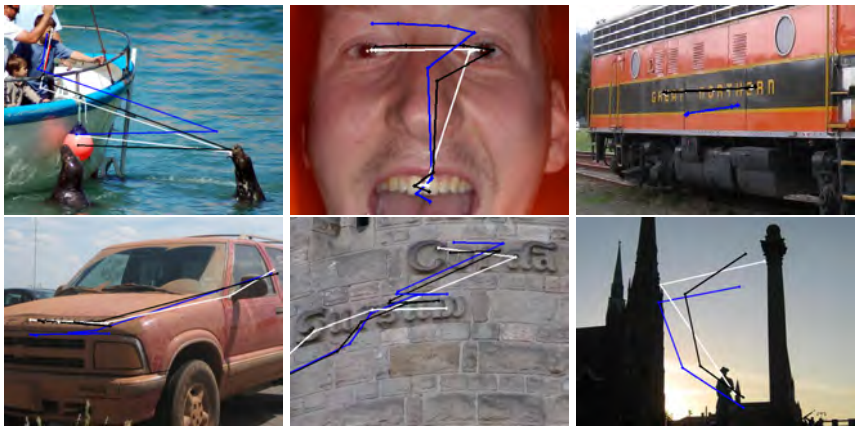


Figure 5.5: Example of success and failure while adjusting the fixations on the eye gaze tracking dataset. The blue line indicates the original fixations, the white line are the fixations corrected by the proposed method, while the black line represent the location of the original fixations transformed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

6

Summary and Conclusions

6.1 Summary

In this thesis, we have analyzed important techniques for gaze estimation and explored ways to improve each of them. We have proposed a gaze estimation pipeline consisting of the following components:

- **Detection:** - Accurately detect and extract eye centers and head pose information.
- **Estimation:** - Combine head and eye location information to obtain a multi-cue visual gaze estimation system.
- **Inference:** - Use information about the gazed environment to infer the most probable gazed object in the scene.

The results obtained in this thesis are discussed in the following paragraphs:

Chapter 2: Accurate Eye Center Location through Invariant Isocentric Patterns.
In Chapter 2 we argue that, although accurate eye center location can already be determined using commercial eye-gaze trackers, additional constraints and expensive hardware make these existing solutions unattractive and impossible to be used on standard (*i.e.* visible wavelength), low resolution images of eyes. Systems based solely on appearance are proposed in literature, but their accuracy does not allow to accurately locate and distinguish eye centers movements in these low resolution settings. Therefore, we investigate a fast method which uses circular symmetry based on isophotes, where centers of the oscu-

lating circles of the isophotes are computed from smoothed derivatives of the image brightness, so that each pixel can provide a vote for its own center. The use of isophotes in the proposed method yields low computational cost (which allows for real-time processing) , robustness to rotation and linear illumination changes (contrast and brightness), and achieve (in-plane) rotational invariance. Furthermore, a scale space framework is used to improve the accuracy of the proposed method and to gain robustness to scale changes.

The proposed approach is extensively tested for accurate eye center location and robustness to changes in illumination, occlusion, eye rotation, pose and scale by using the BioID database, the color FERET database, the Yale Face B database and the CMU Multi-PIE database.

Chapter 3: Combining Head Pose and Eye Location Information for Gaze Estimation.
In Chapter 3, we discussed that head pose and eye location for gaze estimation are closely related topics, but their combination has not been studied in the literature. We argue that, in presence of non frontal faces, eye locators are not adequate to accurately locate the center of the eyes. On the other hand, head pose estimation techniques are able to deal with these conditions, hence they may be suited to enhance the accuracy of eye localization. Therefore, in Chapter 3, we propose a proper integration of a head pose tracker and the isophote based eye locator discussed in Chapter 2 in a complementary manner, so that both system could benefit from each other's evidence.

In the proposed method, the transformation matrix obtained from the head pose is used to normalize the eye regions, while the transformation matrix generated by the extracted eye locations is used to correct the pose estimation procedure. The scheme is designed to (1) enhance the accuracy of eye location estimations, especially in low resolution videos, (2) to extend the operative range of the eye locators and (3) to improve the accuracy of the head pose tracker. The experimental results show that the proposed unified scheme improves the accuracy of eye estimations by 16% to 23%. Further, it considerably extends its operating range by more than $15°$, by overcoming the problems introduced by extreme head poses. Also the accuracy of the head pose tracker is improved by 12% to 24%.

These enhanced estimations are then combined to obtain a gaze estimation system which uses both eye location and head information in order to project the visual gaze of a person on the gazed scene, by retargeting a set of known points on the gazed scene using the head pose information and pose-normalized eye locations. This allows the proposed method to estimate locations gazed by the

eyes at different head locations. The experimentation on the proposed combined gaze estimation system shows that it is accurate (with a mean error between 2° and 5°) and that it can be used in cases where classic approaches would fail without imposing restraints on the position of the head.

Chapter 4: Image Saliency by Isocentric Curvedness and Color.
To locate the most probable gazed object, we need a way to find the most salient objects in the scene. Therefore, in Chapter 4, we propose a computational bottom-up model to detect saliency in images. The method is based on the idea that salient objects should have local characteristics that are different than the rest of the scene, being edges, color or shape. By using a novel operator, these characteristics are combined to infer global information. The obtained information is used as a weighting function for the output of a segmentation algorithm so that the salient object in the scene can easily be distinguished from the background.

The proposed approach is fast and it does not require any learning. The experimentation shows that the system can enhance interesting objects in images and it is able to correctly locate the same object annotated by humans with an F-measure of 85.61% when the object size is known, and 79.19% when the object size is unknown, improving the state of the art performance on a public dataset.

Chapter 5: Improving Gaze Estimation by Saliency.
Using the salient objects detected by the method discussed in Chapter 4, in Chapter 5, we argue that raw gaze estimates should be combined with the saliency information about the scene, to infer the most likely gazed object. Hence, we propose 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 might have occurred during the gaze estimation procedure. Depending on the amount of confidence of the gaze estimate, the saliency framework discussed in Chapter 4 is used to locally define a probability density function, which can be used to find the best adjustment of the raw gaze estimate. The best solution which minimizes the error between the candidate adjustments is then taken to correct the current and future gaze estimates.

The system is tested on three scenarios: using eye tracking data, enhancing a low-accuracy webcam-based eye tracker, and using a head pose tracker. The experimentation shows that the correlation between the subjects in the commercial eye tracking data is improved by an average of 13.91%. The correlation on the low accuracy eye gaze tracker is improved by 59.85%, and for the head pose tracker we obtain an improvement of 10.23%. These results show the potential of the system as a way to enhance and self-calibrate different visual gaze

estimation systems.

6.2 Conclusions

The goal of this thesis was to build an appearance based gaze estimation system which would combine eyes, head pose and information indicating important objects in the gazed scene. Therefore, we have investigated, improved and extended the steps necessary to achieve gaze estimation using a normal webcam. To this end, as the first objective (detection) we targeted the accurate detection of eye location and head pose information in normal, low resolution images. We proposed a fast eye location method which can be used in a variety of situations, and we showed that it is robust to changes in illumination, pose, scale, eye rotations and occlusions. Due to its high accuracy, the method was combined with an appearance based head pose tracker and used to adjust erroneous pose estimations, while the head pose estimations were used to improve the eye location estimation in non-frontal head poses. We showed that this synergetic combination provides improved results in both head pose estimation and eye location.

The second objective of the thesis (estimation) was to combine the obtained head pose and eye information into a framework which would allow restraint-free gaze estimation. To this end, instead of considering the two cues independently, we proposed a method in which the head pose information is used to recalibrate the mapping between the eye positions and known points on the gazed area. This allowed the system to reduce the 3D gaze estimation problem to a set of 2D problems, allowing the eye gaze mapping to be computed at different head pose settings. Although the resulting estimate gives a good approximation of the gaze of the user, this estimate might still be noisy or erroneous due to the resolution of the off-the-shelves hardware and to image noise.

Therefore, the third objective of the thesis (inference) was to adjust these possibly erroneous estimates by using information about the gazed scene. To this end, we firstly proposed a computational method which extracts early saliency in images. The algorithm analyzes local image structures to determine global structures, and proved to achieve higher accuracy than other state of the art methods. Then, the errors which influenced the accuracy of gaze estimation system were taken into account to define a maximum area in which errors were plausible. Using the described saliency framework on the candidate gaze estimates, a possible overall adjustment was computed. This adjustment was then

used to correct the previous and future gaze estimates, obtaining a self calibrating and self adjusting system.

Overall, the methods and techniques proposed in this thesis can be used by devices with a low-resolution camera to address the question "What are you looking at?", in a user-friendly way, without requiring specialized hardware or enforcing specific restraints on the user.

I believe that the ideas discussed in this thesis and the results obtained by their combination are able to provide enabling technology with which computers, starting from a sequence of frames, could understand the interest of the user and react consequently. I foresee the proposed methods and their combination to be actively used in fields such as human behavior analysis and human-computer interfaces, with direct application to assistive interfaces, gaming, interactive advertisement, market research, and many more. Nowadays, for instance, more and more portable devices with integrated cameras are becoming available. Smartphones with front and back facing cameras and with sufficient computational power are already pervasive and, consequently, interesting new computer vision applications and techniques are emerging to take advantage of these new platforms. Using the discussed techniques, it is possible to envision applications in which one camera studies the user, while the other studies the environment. This information could then be combined to understand the interests of the user, and augmented information could flow through the screen of the smartphones, like in an information-magnifier lens. In this way, the proposed technology could enable these devices to effectively become our third eye, pointed at the world of information.

Bibliography

[1] J. S. Agustin, J. P. Hansen, and J. Mateo. Gaze beats mouse: hands-free selection by combining gaze and EMG. In *Computer Human Interaction*, 2008.

[2] K. H. An and M. Chung. 3D head tracking and pose-robust 2D texture map-based face recognition using a simple ellipsoid model. In *Intelligent Robots and Systems*, 2008.

[3] M. Argyle and M. Cook. Gaze and mutual gaze. *Cambridge University Press*, 1976.

[4] S. Asteriadis, N. Nikolaidis, A. Hajdu, and I. Pitas. An eye detection algorithm using pixel to edge information. In *International Symposium on Control, Communication and Signal Processing*, 2006.

[5] S. Ba and J. Odobez. A probabilistic framework for joint head tracking and pose estimation. In *International Conference on Pattern Recognition*, 2004.

[6] S. Ba and J. Odobez. From camera head pose to 3D global roomhead pose using multiple camera views. In *International Workshop Classification of Events Activitis and Relationships*, 2007.

[7] L. Bai, L. Shen, and Y. Wang. A novel eye location algorithm based on radial symmetry transform. In *International Conference on Pattern Recognition*, pages 511–514, 2006.

[8] D. A. Baldwin. Understanding the link between joint attention and language. *Joint attention: Its origins and role in development*, pages 131–158, 1995.

[9] R. Bates, H. Istance, L. Oosthuizen, and P. Majaranta. Survey of de-facto standards in eye tracking. In *COGAIN Conf. on Comm. by Gaze Inter.*, 2005.

[10] BioID Technology Research. The BioID Face Database. http://www.bioid.com, 2001.

[11] P. Bloom. Mindreading, communication and the learning of names for things. *Mind and Language*, 17:37–54, 2002.

[12] M. Bohme, A. Meyer, T. Martinetz, and E. Barth. Remote eye tracking: State of the art and directions for future development. In *Conference on Communication by Gaze Interaction*, 2006.

[13] L. Brown. 3D head tracking using motion adaptive texture-mapping. In *Computer Vision and Pattern Recognition*, 2001.

[14] G. Butterworth. The ontogeny and phylogeny of joint visual attention. *Natural theories of mind: Evolution, development and simulation of everyday mindreading*, pages 223–232, 1991.

[15] G. Butterworth and N. Jarrett. What minds have in common is space: Spatial mechanisms serving joint visual attention in infancy. *British Journal of Developmental Psychology*, 9:55–72, 1991.

[16] P. Campadelli, R. Lanzarotti, and G. Lipori. Precise eye localization

through a general-to-specific model definition. In *British Machine Vision Conference*, 2006.

[17] E. L. Cascia, S. Sclaroff, and V. Athitsos. Fast, reliable head tracking under varying illumination: An approach based on registration of texture-mapped 3D models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(4), 2000.

[18] M. L. Cascia, S. Sclaroff, and V. Athitsos. Fast, reliable head tracking under varying illumination: An approach based on registration of texture-mapped 3D models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(4):322–336, 2000.

[19] COGAIN. Communication by gaze interaction: Gazing into the future. http://www.cogain.org, 2006.

[20] C. Colombo, D. Comanducci, and A. del Bimbo. Robust tracking and remapping of eye appearance with passive computer vision. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 3(4), 2007.

[21] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):564–577, 2003.

[22] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685, 2001.

[23] V. Corkum and C. Moore. Development of joint visual attention in infants. *Joint attention: Its origins and role in development*, pages 61–83, 1995.

[24] D. Cristinacce and T. Cootes. Feature detection and tracking with constrained local models. In *British Machine Vision Conference*, 2006.

[25] D. Cristinacce, T. Cootes, and I. Scott. A multi-stage approach to facial feature detection. In *British Machine Vision Conference*, pages 277–286, 2004.

[26] E. B. Dam and B. ter Haar Romeny. *Front End Vision and Multi-Scale Image Analysis*. Kluwer, 2003.

[27] N. A. Dogson. Variation and extrema of human interpupillary distance. In *SPIE*, 2004.

[28] A. Doshi and M. M. Trivedi. On the roles of eye gaze and head pose in predicting driver's intent to change lanes. *IEEE Transactions on Intelligent Transportation Systems*, 10(3), 2009.

[29] A. T. Duchowski. *Eye Tracking Methodology: Theory and Practice*. Springer, 2007.

[30] S. Duffner. *Face Image Analysis With Convolutional Neural Networks*. PhD thesis, Albert-Ludwigs-Universitat Freiburg, 2008.

[31] R. P. W. Duin. Prtools version 3.0: A matlab toolbox for pattern recognition. In *SPIE*, 2000.

[32] W. Einhauser, M. Spain, and P. Perona. Objects predict fixations better

than early saliency. *Journal of Vision*, 8(14), 11 2008.

[33] L. Elazary and L. Itti. Interesting objects are visually salient. *Journal of Vision*, 8(3):1–15, 3 2008.

[34] T. Farroni, M. Johnson, M. Brockbank, and F. Simion. Infants' use of gaze direction to cue attention: The importance of perceived motion. *Visual Cognition*, 7:705–718, 2000.

[35] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2), 2004.

[36] B. Froba and A. Ernst. Face detection with the modified census transform. *Automatic Face and Gesture Recognition*, pages 91–96, 2004.

[37] W. S. Geisler and M. S. Banks. *Handbook of Optics: Fundamentals, Techniques and Design*, volume 1. McGraw-Hill, Inc., 1995.

[38] A. Georghiades, P. Belhumeur, and D. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):643–660, 2001.

[39] J. Geusebroek, A. Smeulders, and J. van de Weijer. Fast anisotropic gauss filtering. *IEEE Transactions on Image Processing*, 12, 2002.

[40] G. Ghinea, C. Djeraba, S. Gulliver, and K. P. Coyne. Introduction to the special issue on eye-tracking applications in multimedia systems. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 3(4), 2007.

[41] C. C. Gordon, B. Bradtmiller, T. Churchill, C. E. Clauser, J. T. McConville, I. O. Tebbetts, and R. A. Walker. Anthropometric survey of us army personnel: Methods and summary statistics. Technical report, United States Army Natick Research, 1988.

[42] R. Gross, I. Matthews, J. F. Cohn, T. Kanade, and S. Baker. Multi-pie. In *Automatic Face and Gesture Recognition*, 2008.

[43] M. Hamouz, J. Kittlerand, J. K. Kamarainen, P. Paalanen, H. Kalviainen, and J. Matas. Feature-based affine-invariant localization of faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(9):1490–1495, 2005.

[44] D. Hansen and Q. Ji. In the eye of the beholder: A survey of models for eyes and gaze. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(3), 2010.

[45] C. Harris and M. Stephens. A combined corner and edge detection. In *Alvey Vision Conference*, pages 147–151, 1988.

[46] G. Heidemann. Focus-of-attention from local color symmetries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(7):817–830, 2004.

[47] W. R. Hendee and P. N. Wells. *The Perception of Visual Information, 2e*. Springer, 1997.

[48] Y. Hu, L. Chen, Y. Zhou, and H. Zhang. Estimating face pose by facial asymmetry and geometry. In *Automatic Face and Gesture Recognition*, 2004.

[49] J. Huang and H. Wechsler. Visual routines for eye location using learning and evolution. *Evolutionary Computation*, 4(1), 2000.

[50] K. Huang and M. Trivedi. Robust real-time detection, tracking and pose estimation of faces in video streams. In *International Conference on Pattern Recognition*, 2004.

[51] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11), 1998.

[52] O. Jesorsky, K. J. Kirchbergand, and R. Frischholz. Robust face detection using the Hausdorff distance. In *International Conference on Audio- and Video-based Biometric Person Authentication,*, pages 90–95, 1992.

[53] Q. Ji, H. Wechsler, A. Duchowski, and M. Flickner. Special issue: eye detection and tracking. *Computer Vision and Image Understanding*, 98(1), 2005.

[54] Q. Ji and X. Yang. Real-time eye, gaze and face pose tracng for monitoring drivr vigilance. In *Real Time Imaging*, 2002.

[55] T. Judd, K. Ehinger, F. Durand, and A. Torralba. Learning to predict where humans look. In *International Conference on Computer Vision*, 2009.

[56] C. Kervrann, M. Hoebeke, and A. Trubuil. Isophotes selection and reaction-diffusion model for object boundaries estimation. *International Journal of Computer Vision*, 50:63–94, 2002.

[57] S. Kim, S.-T. Chung, S. Jung, D. Oh, J. Kim, and S. Cho. Multi-scale gabor feature based eye localization. In *World Academy of Science, Engineering and Technology*, 2007.

[58] C. Kleinke. Gaze and eye contact: A research review. *Psychological Bulletin*, 100:70–100, 1986.

[59] J. Koenderink and A. J. van Doorn. Surface shape and curvature scales. *Image and Vision Computing*, pages 557–565, 1992.

[60] L. Kovacs and T. Sziranyi. Focus area extraction by blind deconvolution for defining regions of interest. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1080–1085, 2007.

[61] B. Kroon, S. Boughorbel, and A. Hanjalic. Accurate eye localization in webcam content. In *Automatic Face and Gesture Recognition*, 2008.

[62] B. Kroon, A. Hanjalic, and S. M. Maas. Eye localization for face matching: is it always useful and under what conditions? In *ACM International Conference on Image and Video Retrieval*, 2008.

[63] A. Lablack, F. Maquet, and C. Djeraba. Determination of the visual field of persons in a scene. In *International Conference on Computer Vision Theory and Applications*, January 2008.

[64] C. Lampert, M. Blaschko, and T. Hofmann. Beyond sliding windows:

Object localization by efficient subwindow search. *Computer Vision and Pattern Recognition*, 2008.

[65] S. Langton, R. Watt, and V. Bruce. Do the eyes have it? cues to the direction of social attention. *Trends in Cognitive Sciences*, 4:50–59, 2000.

[66] S. R. Langton, H. Honeyman, and E. Tessler. The influence of head contour and nose angle on the perception of eye-gaze direction. *Perception & Psychophysics*, 66(5), 2004.

[67] J. Lichtenauer, E. Hendriks, and M. Reinders. Isophote properties as features for object detection. In *Computer Vision and Pattern Recognition*, volume 2, pages 649–654, 2005.

[68] T. Liu, J. Sun, N. N. Zheng, X. Tang, and H. Y. Shum. Learning to detect a salient object. *Computer Vision and Pattern Recognition*, 2007.

[69] X. Liu, N. Krahnstoever, T. Yu, and P. Tu. What are customers looking at? In *Advanced Video and Signal Based Surveillance*, 2007.

[70] D. Lowe. Distinctive image features from scale-invariant keypoints. In *International Journal of Computer Vision*, volume 20, pages 91–110, 2003.

[71] G. Loy and A. Zelinsky. Fast radial symmetry for detecting points of interest. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):959–973, 2003.

[72] Y. F. Ma and H. J. Zhang. Contrast-based image attention analysis by using fuzzy growing. In *ACM Multimedia*, 2003.

[73] S. Marat, T. Phuoc, L. Granjon, N. Guyader, D. Pellerin, and A. Guerin. Modelling spatio-temporal saliency to predict gaze direction for short videos. *International Journal of Computer Vision*, 82:231–243, 2009.

[74] D. R. Martin, C. C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):530–549, 2004.

[75] Y. Matsumoto and A. Zelinsky. An algorithm for real-time stereo vision implementation of headpose and gaze direction masurement. In *Automatic Face and Gesture Recognition*, 2000.

[76] O. L. Meur, P. L. Callet, D. Barba, and D. Thoreau. A coherent computational approach to model bottom-up visual attention. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(5):802–817, 2006.

[77] C. Moore and V. Corkum. Infant gaze following based on eye direction. *British Journal of Developmental Psychology*, 16:495–503, 1998.

[78] L. Morency and T. Darrell. Head gesture recognition in intelligent interfaces: The role of context in improving recognition. In *Intelligent User Interfaces*, 2006.

[79] L. Morency, A. Rahimi, and T. Darrell. Adaptive view based apperance models. In *Computer Vision and Pattern Recognition*, 2003.

[80] C. H. Morimoto and M. R. M. Mimica. Eye gaze tracking techniques for interactive applications. *Computer Vision and Image Understanding*, 98(1),

April 2005.

[81] L. Moses, D. Baldwin, J. Rosicky, and G. Tidball. Evidence for referential understanding in the emotions domain at twelve and eighteen months. *Child Development*, 72:718–735, 2001.

[82] E. Murphy-Chutorian and M. Trivedi. Head pose estimation in computer vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009.

[83] R. Newman, Y. Matsumoto, S. Rougeaux, and A. Zelinsky. Real-time stereo tracking for head pose and gaze estimation. In *Automatic Face and Gesture Recognition*, 2000.

[84] Z. Niu, S. Shan, S. Yan, X. Chen, and W. Gao. 2D cascaded adaboost for eye localization. In *International Conference on Pattern Recognition*, 2006.

[85] J. Panero and M. Zelnik. *Human Dimension and Interior Space: A Source Book of Design Reference Standards*. Watson-Guptill, 1979.

[86] R. J. Peters, A. Iyer, C. Koch, and L. Itti. Components of bottom-up gaze allocation in natural scenes. *Journal of Vision*, 5(8), 9 2005.

[87] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss. The FERET evaluation methodology for face recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:1090–1104, 2000.

[88] D. Reisfeld, H. Wolfson, and Y. Yeshurun. Context free attentional operators: the generalized symmetry transform. *International Journal of Computer Vision*, 14:119–130, 1995.

[89] B. Repacholi. Infants' use of attentional cues to identify the referent of another personons' emotional expression. *Developmental Psychology*, 34:1017–1025, 1998.

[90] N. Robertson and I. Reid. Estimating gaze direction from low-resolution faces in video. In *European Conference on Computer Vision*, 2006.

[91] E. A. Rossi and A. Roorda. The relationship between visual resolution and cone spacing in the human fovea. *Nature Neuroscience*, 13, 2009.

[92] D. Russakoff and M. Herman. Head tracking using stereo. *Machine Vision and Applications*, 13:164–173, 2002.

[93] M. Scaife and J. Bruner. The capacity for joint visual attention in the infant. *Nature*, 253:265–266, 1975.

[94] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530–534, 1997.

[95] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio. Robust object recognition with cortex-like mechanisms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(3):411–426, Feb 2007.

[96] K. Smith, S. O. Ba, J.-M. Odobez, and D. Gatica-Perez. Tracking the visual focus of attention for a varying number of wandering people. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(7), July 2008.

[97] M. Spain and P. Perona. Some objects are more equal than others: Measuring and predicting importance. In *European Conference on Computer Vision*, 2008.

[98] D. Stavens and S. Thrun. Unsupervised learning of invariant features using video. In *Computer Vision and Pattern Recognition*, 2010.

[99] R. Stiefelhagen and J. Zhu. Head orientation and gaze direction in meetings. In *Conference on Human Factors in Computing Systems*, 2002.

[100] J. Sung, T. Kanade, and D. Kim. Pose robust face tracking by combining active apperance models and cylinder head models. *International Journal of Computer Vision*, 80(2), 2008.

[101] H. Tek and B. B. Kimia. Symmetry maps of free-form curve segments via wave propagation. *International Journal of Computer Vision*, 54(1-3):35–81, 2003.

[102] M. Tomasello. Joint attention as social cognition. *Joint attention: Its origins and role in development*, pages 103–130, 1995.

[103] A. Torralba, A. Oliva, M. S. Castelhano, and J. M. Henderson. Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search. *Psychological Review*, 113(4):766–786, 2006.

[104] A. M. Treisman and G. Gelade. A feature-integration theory of attention. *Cognitive Psychology*, 12(1):97–136, 1980.

[105] J. Tu, T. Huang, and H. Tao. Accurate head pose tracking in low resolutio video. In *Automatic Face and Gesture Recognition*, 2006.

[106] M. Türkan, M. Pardás, and A. Çetin. Human eye localization using edge projection. In *International Conference on Computer Vision Theory and Applications*, 2007.

[107] T. Urruty, S. Lew, N. Ihadaddene, and D. A. Simovici. Detecting eye fixations by projection clustering. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 3(4):1–20, 2007.

[108] R. Valenti. Uvaeyes: eye annotations for the boston university head pose dataset.

[109] R. Valenti and T. Gevers. Accurate eye center location and tracking using isophote curvature. In *Computer Vision and Pattern Recognition*, 2008.

[110] R. Valenti, N. Sebe, and T. Gevers. Image saliency by isocentric curvedness and color. In *International Conference on Computer Vision*, 2009.

[111] R. Valenti, J. Staiano, N. Sebe, and T. Gevers. Webcam-based visual gaze estimation. In *International Conference on Image Analysis and Processing*, 2009.

[112] J. van de Weijer, T. Gevers, and A. D. Bagdanov. Boosting color saliency in image feature detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1):150–156, 2006.

[113] M. van Ginkel, J. van de Weijer, L. van Vliet, and P. Verbeek. Curvature

estimation from orientation fields. In *Annual Conference of the Advanced School for Computing and Imaging*, 1999.

[114] P. Viola and M. J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

[115] M. Voit and R. Stiefelhagen. Deducing the visual focus of attention from head pose estimation in dynamic multi-view meeting scenarios. In *International Conference on Multimodal Interfaces*, 2008.

[116] J. Wang, L. Yin, and J. Moore. Using geometric properties of topographic manifold to detect and track eyes for human-computer interaction. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 3(4), 2007.

[117] P. Wang, M. B. Green, Q. Ji, and J. Wayman. Automatic eye detection and its validation. In *IEEE Workshop on Face Recognition Grand Challenge Experiments*, page 164, 2005.

[118] P. Wang and Q. Ji. Multi-view face and eye detection using discriminant features. *Computer Vision and Image Understanding*, 105(2), 2007.

[119] J. Xiao, T. Kanade, and J. Cohn. Robust full motion recovery of head by dynamic templates and re-registration techniques. In *Automatic Face and Gesture Recognition*, 2002.

[120] Y. Zhang and C. Kambhamettu. 3D head tracking under partial occlusion. *Pattern Recognition*, 35:1545–1557, 2002.

[121] Z. Zheng, H. Wang, and E. K. Teoh. Analysis of gray level corner detection. *Pattern Recognition Letters*, 20(2):149–162, 1999.

[122] Z. H. Zhou and X. Geng. Projection functions for eye detection. In *Pattern Recognition*, pages 1049–1056, 2004.

[123] J. Zhu and J. Yang. Subpixel eye gaze tracking. In *Automatic Face and Gesture Recognition*. IEEE Computer Society, 2002.

[124] Z. Zhu and Q. Ji. Robust real-time eye detection and tracking under variable lighting conditions and various face orientations. *Computer Vision and Image Understanding*, 98(1):124–154, 2005.

Samenvatting

In dit proefschrift analyseren we relevante technieken ten aanzien van het automatisch schatten van kijkrichting in video. In ons onderzoek naar het verbeteren van deze technieken hebben we een kader ontwikkeld bestaande uit de volgende componenten:

- **Detectie:** Het accuraat detecteren en extraheren van oog locaties en de pose van het hoofd.
- **Schatting:** Het combineren van informatie betreffende oog locaties en hoofd pose in een systeem voor het schatten van de kijkrichting.
- **Gevolgtrekking:** Het identificeren van het aanschouwde object op basis van omgevinginformatie.

De in dit proefschrift behaalde resultaten worden hieronder behandeld:

Hoofdstuk 2: Accurate Pupil Lokalisatie op basis van Invariante Isocentrische Patronen.
Het lokaliseren en volgen van de pupillen kan op accurate wijze worden uitgevoerd met commerciële oog-trackers die veelal op infrarood licht opereren. Echter, de kostbaarheid van de benodigde hardware en de ontoepasbaarheid van deze oplossing op normale, vaak voorkomende digitale afbeeldingen (van lage resolutie en zonder infrarood) maken dat de technologie niet breed inzetbaar en dus onaantrekkelijk is. Hiertoe zijn systemen ontwikkeld die opereren op basis van informatie in het zichtbare spectrum. Deze oplossingen zijn echter niet accuraat genoeg om data in lage resolutie te verwerken. Om deze redenen onderzoeken we in dit proefschrift het gebruik van circulaire symmetrie op basis van isophoten voor het lokaliseren van de pupillen. De middelpunten van de osculerende isophoot cirkels worden berekend op basis van afgeleiden van

de intensiteit waarden in het beeld. Op deze manier kan elke pixel "stemmen" op zijn eigen middelpunt. Het gebruik van isophoten in onze methode leidt tot zeer snelle (real-time) data verwerking, robuustheid tegen rotatie en lineaire veranderingen van de belichtingsomstandigheden, en volledige invariantie onder rotaties in het beeldvlak. De accuraatheid van de methode wordt verder verbeterd door de beeld data op meerdere observatieschalen te verwerken.

We testen de door ons voorgestelde methode uitvoerig in termen van accuraatheid van de pupil lokalisatie en robuustheid tegen veranderingen in de belichting, occlusie, rotatie, pose en schaal. De hiervoor gebruikte databases zijn BioID, color FERET, Yale Face B en CMU Multi-PIE.

Hoofdstuk 3: Het Schatten van Kijkrichting door het Combineren van Hoofd Pose en Pupil Locaties.
Het gebruik van de pose van het hoofd of de pupil locaties voor het schatten van de kijkrichting zijn twee nauw verwante onderwerpen. Desondanks heeft het combineren van deze twee informatiebronnen weinig tot geen aandacht ontvangen in de literatuur. We observeren dat het detecteren van de ogen in afbeeldingen van niet-frontale gezichten leidt tot foutieve schattingen ten aanzien van de locatie van de pupil Ű het middelpunt van het oog. Terwijl aan de andere kant de technieken die de pose van het hoofd schatten juist goed kunnen omgaan met deze omstandigheden. We stellen daarom in hoofdstuk 3 een methode voor waarmee de pose van het hoofd bepaald kan worden door te combineren met onze methode om ogen te detecteren uit hoofdstuk 2. We doen dit op complementaire wijze, zodat beide methodes kunnen profiteren van elkaars bevindingen.

Het schatten van de pose van het hoofd levert een transformatie matrix op. Deze transformatie wordt gebruikt om de gebieden waarin de ogen zich bevinden te normaliseren. Het lokaliseren van de pupillen levert op zijn beurt ook een transformatie matrix op, welke wordt gebruikt om de schatting van de pose van het hoofd te corrigeren. Dit stelsel is ontworpen om 1) de schatting van de pupil locaties te verbeteren, voornamelijk in videoŠs van lage resolutie, 2) het operationele gebied van de pupil detector te verwijden en 3) het schatten en volgen van de pose van het hoofd te verbeteren. De experimentele resultaten tonen aan dat het door ons voorgestelde unificerende stelsel de detectie van de pupillen met 16% tot 23% verbetert. Door het corrigeren voor situaties waarin de pose van het hoofd extreem varieert, wordt het operationele gebied waarin pupillen kunnen worden gedetecteerd verwijd met meer dan 15°. Bovendien kan de pose van het hoofd met 12% tot 24% beter worden geschat.

Op basis van de verkregen resultaten bouwen we een systeem waarmee de kijkrichting kan worden bepaald. We doen dit door de geschatte kijkrichting te projecteren op een omgeving waarvan we enkele 3D coördinaten kennen, zodat het systeem gekalibreerd kan worden. Als een resultaat hiervan kunnen we op basis van de geschatte informatie ten aanzien van de pose van het hoofd en de locaties van de pupillen bepalen waar iemand naar kijkt. De gemiddelde fout in het schatten van de kijkrichting ligt tussen 2° en 5°. Het door ons ontwikkelde systeem is dus zeer nauwkeurig, en blijft overeind in situaties waarin meer klassieke benaderingen falen.

Hoofdstuk 4: Object Saillantie op basis van Isocentrische Kromming en Kleur.
We onderzoeken welke objecten in de omgeving het meest waarschijnlijk zijn om aanschouwd te worden. Hiertoe stellen we in hoofdstuk 4 voor om op basis van een computationeel bottom-up model de saillantie van objecten in het beeld te bepalen. De methode is gebaseerd op het idee dat de lokale karakteristieken van saillante objecten verschillen van de rest van het beeld in termen van randen, kleur of vorm. Gebruikmakend van een door ons ontwikkelde operator, worden deze karakteristieken gecombineerd om te komen tot globale informatie over de objecten. De verkregen informatie wordt gebruikt in een mechanisme om het resultaat van een segmentatie algoritme te wegen. Op deze manier worden saillante objecten onderscheiden van de achtergrond.

De voorgestelde methode is snel en heeft geen voorbeelden nodig. Experimentele resultaten tonen aan dat dezelfde objecten die ook door mensen als saillant zijn aangemerkt kunnen worden gedetecteerd met een accuraatheid (F-measure) van 85.61%, gegeven dat de afmetingen van de objecten bekend zijn. In het geval dat de afmetingen onbekend zijn is het resultaat 79.19%, waarmee de state-of-the-art op een publiekelijk beschikbare dataset is verbeterd.

Hoofdstuk 5: Verbeterde Kijkrichting op basis van Object Saillantie.
We onderzoeken of de schatting van de kijkrichting gecombineerd kan worden met object saillantie om te kunnen bepalen welk object in de omgeving aanschouwd wordt. Als derde stap in ons raamwerk voor het bepalen van de kijkrichting, stellen we voor om de kijkrichting te corrigeren op basis van saillantie van objecten in de nabijheid van de schatting. Hiertoe wordt object saillantie gerepresenteerd als een kansverdeling over het beeld, en afhankelijk van de zekerheid ten aanzien van de schatting van de kijkrichting meegenomen in de optimalisatie.

Het door ons ontwikkelde systeem wordt getest in drie verschillende scenarioŠs, waarin data wordt gebruikt van 1) een commerciële infrarood eye-tracker, 2) een eye-tracker die opereert op beelden zoals verkregen uit een normale we-

bcam en 3) een methode om de pose van het hoofd te schatten en te volgen. De experimentele resultaten tonen aan dat de correlatie tussen de subjecten in de commerciële eye-tracking data met gemiddeld 13.91% wordt verhoogd. In het tweede scenario wordt de correlatie verbeterd met 59.85%, en met 10.23% in het derde scenario. Met deze resultaten is het potentieel van het door ons ontwikkelde systeem voor het verbeteren (en automatisch kalibreren) van verschillende systemen voor het schatten van kijkrichting aangetoond.

Aknowledgements

This thesis would not have been possible without the help and support of my supervisors Nicu Sebe and Theo Gevers. Their guidance was impeccable, as they gave me the freedom to wonder in my own thoughts, make my own mistakes and work at my own pace, while still being very supportive and keeping a close watch on my work. It was amazing to have had these two special persons guiding me in my research, but also in my life: you guys really know that being a good supervisor does not end at fixing the English on a paper! My deep gratitude also goes to all the people with whom I closely worked on topics related to my research: Jacopo Staiano, Zeynep Yücel, Enver Sangineto, Hamdi Dibeklioğlu, Alejandro Jaimes, Adel Lablack and Kalin Stefanov.

Together with them, I would also like to thank Arnold, and all my colleagues at the University of Amsterdam for the nice exchange of ideas and for their valuable feedback during my presentations: Each of your comments was invaluable to my research! A particular thank you (and sorry) goes to the people who shared the office with me during these years, and thus had to cope with all the meetings, phone calls, guests and supervisors constantly popping in and out of the room: Ivo (the dude) Everts, Gosja Migut, Vladimir (Kurac) Nedović, Athanasius (are you crazy?) Noulas, Jasper Uijlings, Koen van de Sande, Arjan Gijsenij, Hamdi (the man) Dibeklioğlu. You all are really great and talented people, and I really had a great time working around all of you!

Special words of gratitude go to Virginie for always being so helpful, and to my Paranimfs, Ivo Everts and Jan van Gemert, who were always there ready to help me with anything I needed, sometimes even without asking for it! I also would like to thank Jasper van Turnhout for passionately designing the cover of this thesis, and all the guys at ThirdSight for proactively turning this work (or part thereof) into a product.

Furthermore, I would like to thank all the people who indirectly made this thesis possible: First of all my parents for always giving me the strength to keep going, and always sending me tons of love with a sprinkle of nice weather forecast from Ascoli. I know that you are mentioning it because you would like me to be there with you. I miss you too!

I would also like to thank my early abductors: Elin, Hetty and Harry. They really helped me during my first years in the Netherlands. Without them, I would probably be back in Italy right now and none of this work would have been possible.

I also want to thank all my friends for the good times we had together and for the balance they brought into my life, allowing me to enjoy Amsterdam with such a wonderful, family-like feeling. Thank you for being my awesome dysfunctional international family, I Love You All!

A personal "I love you all" goes to Laura, who I thank for having the strength to cope with me in stressful periods and for always being able to cheer me up. You really are an amazing woman.

I guess I should also thank the Coen brothers for giving us the Big Lebowski. That movie definitively signed this part of my life, as half of the research group was not able to have a normal conversation without quoting it!

Last but not least I would really like to thank everyone who had the patience and the will of reading through every single page of this work in an orderly fashion. You truly are a special person.