Graduate School ETD Form 9
(Revised 12/07)

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Harikrishna K. Rajabather

Entitled An Adaptive Eye Gaze Tracking System Without Calibration for Use in an Automobile

For the degree of Master of Science in Electrical and Computer Engineering

Is approved by the final examining committee:

Chair	
Sarah Koskie	
Yaobin Chen	
Lauren Christopher	

To the best of my knowledge and as understood by the student in the *Research Integrity and Copyright Disclaimer (Graduate School Form 20)*, this thesis/dissertation adheres to the provisions of Purdue University's "Policy on Integrity in Research" and the use of copyrighted material.

Approved by Major Professor(s): Sarah Koskie

Sarah Koskie

Approved by: Yaobin Chen 3/24/2011

Head of the Graduate Program Date

Graduate School Form 20
(Revised 9/10)

PURDUE UNIVERSITY
GRADUATE SCHOOL

Research Integrity and Copyright Disclaimer

Title of Thesis/Dissertation:

An adaptive eye gaze tracking system without calibration for use in an automobile

For the degree of Master of Science in Electrical and Computer Engineering

I certify that in the preparation of this thesis, I have observed the provisions of *Purdue University Executive Memorandum No. C-22,* September 6, 1991, *Policy on Integrity in Research.**

Further, I certify that this work is free of plagiarism and all materials appearing in this thesis/dissertation have been properly quoted and attributed.

I certify that all copyrighted material incorporated into this thesis/dissertation is in compliance with the United States' copyright law and that I have received written permission from the copyright owners for my use of their work, which is beyond the scope of the law. I agree to indemnify and save harmless Purdue University from any and all claims that may be asserted or that may arise from any copyright violation.

Harikrishna K. Rajabather

Printed Name and Signature of Candidate

12/15/2010

Date (month/day/year)

*Located at http://www.purdue.edu/policies/pages/teach_res_outreach/c_22.html

AN ADAPTIVE EYE GAZE TRACKING SYSTEM FOR USE IN AN AUTOMOBILE

A Thesis

Submitted to the Faculty

of

Purdue University

by

Harikrishna K. Rajabather

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science

May 2011

Purdue University

Indianapolis, Indiana

To my loving parents

ACKNOWLEDGMENTS

First and foremost I would like to thank my advisor Dr. Sarah Koskie for her advice and guidance. I would also like to thank the members of my graduate committee, Dr. Yaobin Chen and Dr. Lauren Christopher for taking the time and effort to help me with my masters thesis.

I would like to acknowledge and thank the members and researchers at the Transport Active Safety Institute, whose help and input was vital to my research work. I would like to specifically thank Dr. Robert Dufour, Ray Prieto, Dr. Nicola Fricke, Heather Wisdom, Nate Bruce, and Kreg Sweeney.

Finally I want to thank my mother, my father and Sindhu for the support they have provided over the years, without which I would have been unable to finish my masters research work.

PREFACE

Currently there are no eye gaze tracking systems in automobiles, primarily because such systems require gaze calibration. This is not very user friendly and commercial car manufacturers shy away from such a solution. Furthermore there are a lot of other technical issues such as efficient placement of cameras, camera cost, robust tracking, feature extraction, excessive head movements of drivers.

As a result, in this thesis we strive to build and design a robust eye gaze tracking system. Over the course of the research we have examined all the various modules that comprise of such a system and have addressed a variety of issues to improve the system robustness to the extent possible.

TABLE OF CONTENTS

Page

Page

Page

LIST OF TABLES

LIST OF FIGURES

SYMBOLS

GENERAL SYMBOLS

r	Row in Image Frame
c	Column in Image Frame
P_{ix}	Pixel in an Image Frame
$P_{ix}s^{-1}$	Pixels per second; Unit measure for velocity
$P_{ix}s^{-2}$	Pixels per second square; Unit measure for acceleration
σ	Smoothing factor in the Gaussian kernel unless otherwise noted

STATE VECTOR MACHINES

x	input feature vector
$\phi(x)$	Kernel Function used to map input vectors to a higher dimension
ω_o	Distance to Hyperplane from the origin
ω	normal to the hyperplane
C	arbitrary upper limit on Lagrangian
α_i	coefficients for the feature vectors

MEAN-SHIFT TRACKER

h	window size
k	kernel density function
y	pixel at the center of window
x_i	pixel at location i
$b(x_i)$	index of histogram bin at location x_i
u	histogram bin index

$\hat{p}_u(y)$	intensity probability distribution of target image centered at pixel y
$\hat{q}_u(y)$	intensity probability distribution of current image centered at pixel y
δ	Kronecker delta function

KALMAN PREDICTION

K	Kalman gain
Σ	Covariance matrix
X_t	State transition matrix
W_t	System perturbation
Z_t	Current input vector
V_t	Process noise
R	Measurement error covariance

GAZE MAPPING AND GENERALIZED REGRESSION NEURAL NETWORKS

$\vec{V_p}$	Visual axis
$\vec{V_o}$	Optical axis
M	Orthonormal matrix for gaze mapping
X	Input gaze vector for GRNN
Y	Enumeration of target Gaze regions

GLOSSARY

blob	A region having a constant pixel intensity value, usually found in binary images.
dilation	An image processing technique that enhances bright regions in a image.
erosion	An image processing technique that shrinks brights regions in an image.
glint	Purkinje image due to a light.
generalized regression	
neural network	A class of Neural Networks.
Kalman filter	An algorithm that is used to predict the location of an object in the next image frame based on previous measurements.
K-nearest neighbor	A pattern recognition algorithm that classifies objects based on closest training samples.
mean-shift algorithm	Computer vision algorithm that locates the maximum of a density function. It can be used to estimate the location of an object the next frame.
neural networks	Mathematical model based on biological neural networks used to process information.
Purkinje image	A reflection of an external object from structure of the eye.
regions of interest	Areas that traditionally provide information about the driving scene to the driver.

state vector machines	A two-state hyperplane-based pattern classification algorithm.

ABSTRACT

Rajabather, Harikrishna K. M.S.E.C.E., Purdue University, May 2011. An Adaptive Eye Gaze Tracking System for Use in an Automobile. Major Professor: Sarah Koskie.

One of the biggest hurdles to the development of an effective driver state monitor is the that there is no real-time eye-gaze detection. This is primarily due to the fact that such systems require calibration. In this thesis the various aspects that comprise an eye gaze tracker are investigated. From that we developed an eye gaze tracker for automobiles that does not require calibration. We used a monocular camera system with IR light sources placed in each of the three mirrors. The camera system created the bright-pupil effect for robust pupil detection and tracking. We developed an SVM based algorithm for initial eye candidate detection; after that the eyes were tracked using a hybrid Kalman/Mean-shift algorithm. From the tracked pupils, various features such as the location of the glints (reflections in the pupil from the IR light sources) were extracted. This information is then fed into a Generalized Regression Neural Network (GRNN). The GRNN then maps this information into one of thirteen gaze regions in the vehicle.

1. INTRODUCTION

1.1 Background Overview

Currently a lot of research aims to allow the commercialization of autonomous vehicles. This research ranges from control systems for autonomous motion, to sensors, to Human Machine Interaction (HMI).

HMI research on autonomous vehicles is very important because the driver determines the success of a commercial autonomous vehicle. If the driver does not have a pleasant experience he/she will not purchase such a system. Secondly due to the current state of highway infrastructure these systems will be limited in their abilities, hence they are called semi-autonomous systems. As a result the driver will have to be in the control loop of the system and will have to be aware of the road conditions. Finally, in order to avoid liability, commercial car companies want systems that keep the driver in the loop and systems that are aware of the driver's state.

This has lead to a lot of research that seeks to make the HMI experience pleasant, efficient and natural. There is also research on the control algorithms for the autonomous vehicles to work in a manner that provide a comfortable driving experience for the driver. However the twin constraints of keeping the driver constantly in the loop and alerting the driver to the driving situation, while providing the driver with a pleasant experience makes the development of such systems very hard. This is because such a HMI system must be aware of the driver's current status (with respect to drowsiness, focus on the road, alertness, etc.). Such a system would need some sort of feedback from a very robust Driver State Monitor (DSM) system. Such a system should identify drowsiness, eye gaze and other measures of driver state through a computer vision based system.

Unfortunately there is currently no robust DSM system that can provide all eye gaze information. One reason is that eye gaze mapping currently requires calibration. In this thesis the problem of eliminating calibration for eye gaze mapping is explored. Background research on existing DSM systems, eye gaze mapping and tracking systems and computer vision technology are reviewed. From that a robust eye gaze tracking system is developed.

1.2 Existing Research on Driver State Monitoring

The literature describes many attempts to quantify driver state. There have also been attempts to develop computer vision based DSM systems. One easily computable driver state metric is the PERCLOS index [1], which is the gold standard for driver drowsiness prediction. [2], [3] and [4] attempt to quantify face-pose (forward or not). Furthermore they have developed algorithms that can discern face pose without prior calibration. [5] investigated the correlation between head pose, head movement and eye gaze, in particular for lane change.

Most DSM systems in the literature determine face-pose and calculate eye closure which can be used for drowsiness detection [2], [3] and [6]. [4] developed a stereo camera system that can perform eye gaze detection in a laboratory setup, but their system requires extensive prior calibration. Two main reasons that there is no reliable commercial DSM system that incorporates eye gaze tracking are:

- **Calibration:** Almost all eye tracking systems perform calibration to make them accurate. Commercial automotive companies shy away from any solution that requires calibration.
- **Image quality:** Gaze calibration requires high definition camera systems. However with today's camera costs decreasing, this may not be a problem much longer.

1.3 Thesis Overview and Areas of Focus

In order to build a robust gaze tracking system for use in vehicles, the following topics are of interest:

- **Vehicle Regions of Interest:** Based on functionality, regions in the car were identified as being of interest to the HMI.
- **Human Facial Anatomy:** Unique properties of the human face include spatial relationships among various facial features, including eyes, nose and mouth, as well as the anatomy of the human eye and its optical properties. The relation between gaze, head pose and head movement patterns in an automobile are also of interest.
- **Image Acquisition:** Various image acquisition systems were examined, such as monocular, stereo, multiple camera systems. The minimum resolution required to extract pupil features was determined through experimentation. Constraints on locations where these image acquisition systems could be placed were quantified. Finally based on these constraints, the ideal locations to place image acquisition systems in the simulator (and vehicles in general) were identified.
- **Initial Eye Detection:** Algorithms to detect initial eye candidates were researched.
- **Eye Tracking:** Various computer vision tracking techniques were examined in order to determine the most robust eye tracking algorithm under various head movements.
- **Face Pose:** The relation to face pose and gaze was studied.
- **Gaze Detection and Calibration:** Many different eye gaze techniques and calibration techniques were studied. From that an algorithm that required no calibration was developed.

The next few sections discuss the current state of research in each of the above mentioned areas of focus. Furthermore the work done in this thesis with respect to these areas will also be briefly discussed in these sections.

1.3.1 Vehicle Regions of Interest

Early in the research it was found that it is more useful to provide to the HMI systems a region of interest (ROI) than a coordinate in the 3D space. Furthermore it became evident that if gaze is classified by ROIs then gaze mapping requires less accuracy, making calibration easier. The main regions of interest are indicated in Figure 1.1.

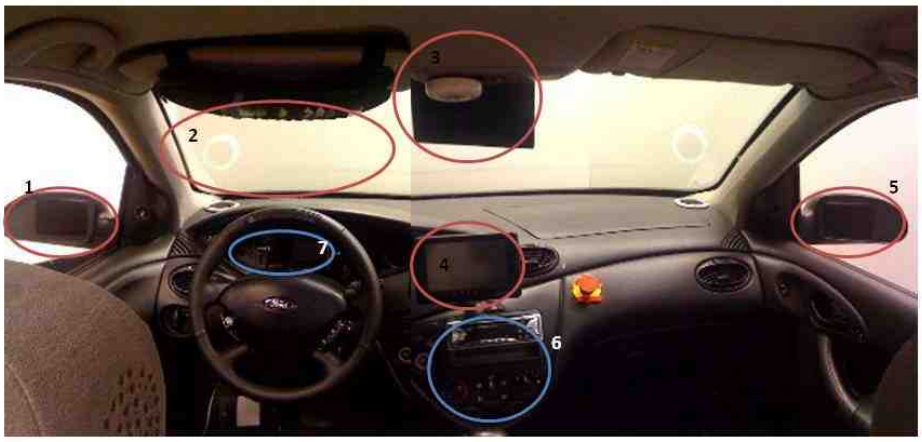


Fig. 1.1. Regions of interest in the car

The ROIs are areas that traditionally provide information about the driving scene to the driver, for example forward view, the dashboard cluster and the mirrors. From

subject testing in the simulator it was found that these were also the regions that drivers tends to look at the most. If the driver was looking at any point other than those specified in Figure 1.1 then the driver was generally not paying any attention. The areas indicated in red are the primary ROIs, the areas indicated in blue are secondary ROIs. Secondary ROIs are regions that are harder to map using the gaze mapping function.

1.3.2 Human Facial Anatomy

A lot of eye trackers and eye gaze detection systems use the unique features of the human eye and the human face to improve efficiency, so there is a wealth of research on how to utilize these features for gaze calibration and detection.

The human eye and the face have many interesting features. [7] talks about the optical properties of the human eye in great detail. [8] and [4] take advantage of the bright eye effect for eye tracking. Meanwhile [9] did research on how lighting and facial orientations affect the intensities of bright eyes. [4] discusses the relationship between eyes and how this can be used to discern face pose.

[10] developed robust algorithms to extract glints from the human eyes based the unique intensity patterns of the human eye that allows them to perform such an operation. [4] also uses this property for initial eye detection. [11] uses the relationship between the human eye and other facial features to develop a gaze tracking algorithm.

In this thesis, we combine these techniques to design eye detection, tracking, face pose discerning and gaze mapping algorithms.

1.3.3 Image Acquisition

There are two ways to perform image acquisition. The first way is to use a single monocular camera, then perform 3D projection during image processing [12]. The other way is to use a stereo camera. The main difference between these two systems is that the 3D stereo camera makes it easier to establish the real world coordinates

of the eye pupil candidates and the resultant gaze mapping functions are easier to compute. However in vehicles the potential locations where stereo cameras can be placed are limited.

Apart from the gaze mapping difference, the rest of the eye gaze tracking system is similar for both techniques. Both these techniques use an even/odd image acquisition system [8]. The even image induces the bright eye effect in one image and the odd image leaves the pupil black. The subtraction of these two images make the extraction of the eye pupil regions easier. Hence the stereo camera system is just an extension of the monocular camera system when it concerns image processing and eye tracking.

1.3.4 Initial Eye Detection

There has been a lot of past work using geometric constraints of the eye, [13], [14], [15] and [16]. These researchers use appearance-based methods such as circle and ellipsoid matching, where a generic eye model is used to sequentially search the image for eyes. [16] used deformable templates to increase accuracy during sideways motion. [17], [18] used a Radial Basis Function (RBF) Neural Network (NN) classifier, but it is trained for the frontal view face image only.

Literature specific to gaze tracking systems points to several downsides to eye detection techniques that use geometric features of the eye such as length of iris, size of sclera and other spatial features. These features are not accurate under excessive head movements, during eye closure, during occlusion and these features are varied across the population. [19], [20] used State Vector Machines (SVM) to detect the eye, taking advantage of the unique intensity pattern around the eye to perform pattern classification.

In this thesis we compare the SVM [19] and K-Nearest Neighbor(KNN) [21] techniques using a variety of training data.

1.3.5 Eye Tracking

Most eye tracking techniques use brute force methods such as pattern recognition or template matching from frame to frame [2], [8] and [15]. This is not very efficient and takes a lot of computational processing time.

The other approach is to use some sort of image tracking technique. [22] used a Kalman filter [23] to track eye features from frame to frame. [19] used a hybrid of the Kalman filter and Mean-shift tracking [24] techniques to track eye features from frame to frame. The Kalman filter helped predict the position of the eye in the next frame. If the Kalman filter did not work then Mean-shift was used. This technique speeds up processing and is fairly robust compared to generic tracking algorithms. However, under quick head movements this technique does not work effectively.

We built and compared various eye tracking techniques. We modified [19]'s algorithm by using a second order dynamic system for the Kalman state equation.

1.3.6 Face Pose

Face pose estimation is important because it is used for gaze calibration and for providing driver state information, when gaze mapping is not available. There are a number of techniques that quantify and measure face pose. Face pose can be estimated using model-based techniques, appearance-based techniques or feature-based techniques. Model-based approaches typically recover the face pose by recreating the 3D face model [25] and [4]. Appearance-based techniques use view-interpolation to try to create a relationship between face appearance and orientation [26]. According to [4], appearance-based techniques are less accurate than model-based techniques because appearance-based techniques use pattern recognition from a set of sample images to interpolate gaze. Feature-based approaches determine face pose using a subset of facial features. A lot of feature-based techniques use the Viola-Jones technique [27]. The following correlations were established between 3D face pose and the properties of pupils [4]:

- The distances between pupils decrease as the pupils rotate away from the frontal orientation.
- The ratio of the pupil intensities is approximately 1 when the face pose is frontal. This increases or decreases as the pupils rotate away from the frontal orientation.
- The size of the pupil decreases as it rotates away or as the head moves vertically up or down.

In our thesis we incorporated some of the work done by [4] to perform simple face pose estimation.

1.3.7 Eye Gaze Detection and Calibration

There has been a lot of work surrounding eye-gaze tracking and eye-gaze mapping for general HMI. However little work has been done on eye gaze trackers for use in automobiles. [4] developed a DSM system that included eye gaze detection; however, this technique required calibration. [28] gives an overview of eye-gaze tracking and the most common techniques used today.

The most common eye tracking techniques are either 2-D mapping based gaze estimation techniques (see [4], [11], [29] and [30]) or 3-D gaze estimation techniques (see [12], [31] and [32]). Calibration for most of these techniques involves looking at predefined points in the real world space.

2. ANATOMY OF THE HUMAN EYE AND FACE

2.1 Introduction

Unique characteristics of the human eye include the intensity patterns of the eyes [19], the structural relationships between the eyes and nose [33], [34] and the optical properties of human eyes [35]. This chapter discusses unique properties of the human face, the anatomy of the human eye, its optical properties, how it reacts under infra-red (IR) light, and finally how computer vision can be used to take advantage of these properties.

2.2 Anatomy of the Human Eye

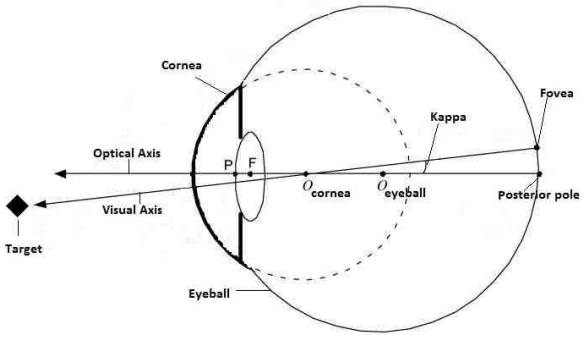


Fig. 2.1. Anatomy of the Human Eye

Figure 2.1 shows the anatomy of the human eye and its optical properties. The eye ball is made up of two spheres [35]. The smaller segment called the anterior is transparent and constitutes one-sixth of the volume of the eyeball. Its radius is about 8 mm. The larger segment, is called the posterior, is opaque and forms five-sixths of the volume of the eye ball. Its radius is about 12 mm.

The anterior pole of the eye is the center of curvature of the transparent segment or cornea. The posterior pole is the center of the posterior curvature of the eyeball. The optic axis is defined as a line connecting these two poles, as shown in Figure 2.1.

The fovea is the center of the retina. This is the region that has very high color sensitivity. The eye ball orients itself so that the light from the object it is currently viewing falls on the fovea. The visual axis is the line connecting the object being viewed to the center of the fovea. This axis goes through O_{cornea}, the eye's nodal point, as shown in Figure 2.1. The fovea is located slightly away from the anterior pole, thus the optical and visual axes meet at the point of intersection (O_{cornea}), making an angle κ between them. This angle should be the same for both eyes and is approximately 5^o [12], [35].

It is important to note that the optical axis will not be measured. Instead the papillary axis, which connects the center of the pupil with O_{cornea}, will be measured. This axis is very close to the optical axis and can be used to approximate the optical axis [12].

2.2.1 Glint Formation

When light passes through the eye, the boundary of the cornea acts like a reflective surface. Therefore, if a light source is placed in front of the eye, the reflection from the external surface of the cornea is captured as a very bright spot in the eye image as shown in Figure 2.2. This is also known as the glint.

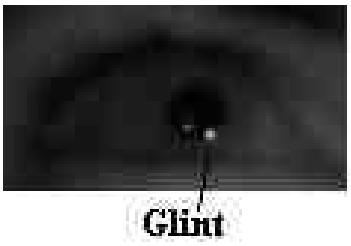


Fig. 2.2. Glints in the Eye

To better understand the orgin of the glint, the optical properties of the human eye need to be discussed. The cornea can be modeled as a convex mirror with radius R. The focal point is F, and the center of curvature, O_{cornea} (also center of the cornea), is shown in Figure 2.3. If a light source is placed in front of the eye, the cornea will produce a virtual image of the light source.

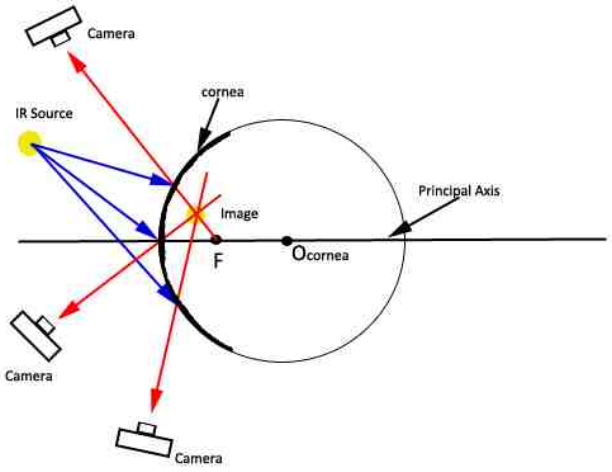


Fig. 2.3. Formation of the virtual image

The virtual image is the imaginary location from which the light diverges. Thus the glints seen in Figure 2.2 are the virtual images of the light source. The reflection law of convex mirrors establishes that the virtual light position (behind the cornea) is determined only by the location of the mirrors and the actual position of the lights. Hence it is independent of the observer.

The importance of the glints is that they help estimate the location of the O_{cornea} which determines the papillary axis. From that we can estimate the visual axis.

Bright Eye Effect

The bright eye effect shown in Figure 2.4 is created when the eye is illuminated by an IR source emitting light along the camera's optical axis [19]. At the near IR

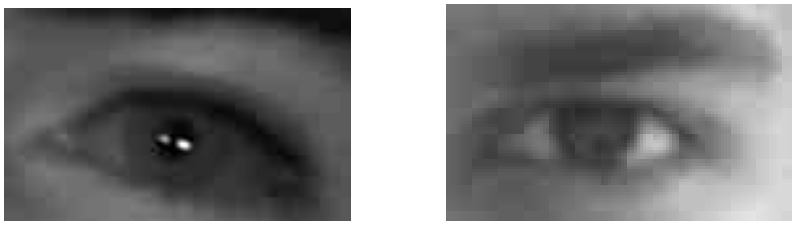
wavelength, pupils reflect almost all IR light they receive along the same path. Since this path is parallel to the camera's optical axis, the camera captures the bulk of the light, if the light source is placed very close to the camera as in Figure 3.5 producing the bright eye effect.

Fig. 2.4. Bright-eyes effect

2.3 Effect of IR Light on Human Eyes

IR light is below the visual spectrum for the human eye, hence it does not cause discomfort to the driver. Furthermore, we can block visual light using an IR bandpass filter, making image processing and feature extraction much easier.

The more important effect of IR light on human eyes is that under IR illumination the iris has an unique intensity pattern. Thus regardless of the color of the iris, eyes have a similar intensity under IR light. This helps with feature extraction. Figure 2.5(a) shows an eye under IR illumination while Figure 2.5(b) shows an eye under visible light.



(a) Eye pattern under IR illumination

(b) Eye pattern under visible light illumination

Fig. 2.5. Eyes under IR and visible light illuminations

3. SYSTEM OVERVIEW

3.1 Introduction

This chapter gives an overview of the software and hardware models and how the various sections are related to each other. The adaptive eye gaze tracking software system consists of three major parts: the pattern recognition module that extracts the eye features, the eye tracking module, and finally the gaze mapping module (which consists of the calibration and mapping). The hardware system consists of the camera for image capture, the LEDs for gaze mapping and the embedded system module.

3.2 Software Model Overview

As seen in Figure 3.1 the system performs software initialization upon start up. Once the software is initialized, it calls the pattern recognition module. In this module the system scans each frame for potential eye candidates. Once at least a positive eye candidate is identified we initialize the tracker.

The tracker initializes the hybrid Kalman/mean-shift algorithm. Details about the eye and its position relative to other facial features are taken into account. If both the eye candidates are found then the tracker tries to figure out which is the left eye and which is the right eye. Once the tracking module is initialized the system moves to the next phase. It is important to note that the system will not return to the pattern recognition and tracker initialization modules unless it loses track of both the eye candidates.

The eye tracking module consists of the eye tracker algorithm and also the robust feature extraction algorithm. The eye tracker tracks the eye candidates from frame to frame. In the meantime the feature extractor uses a robust technique to get the

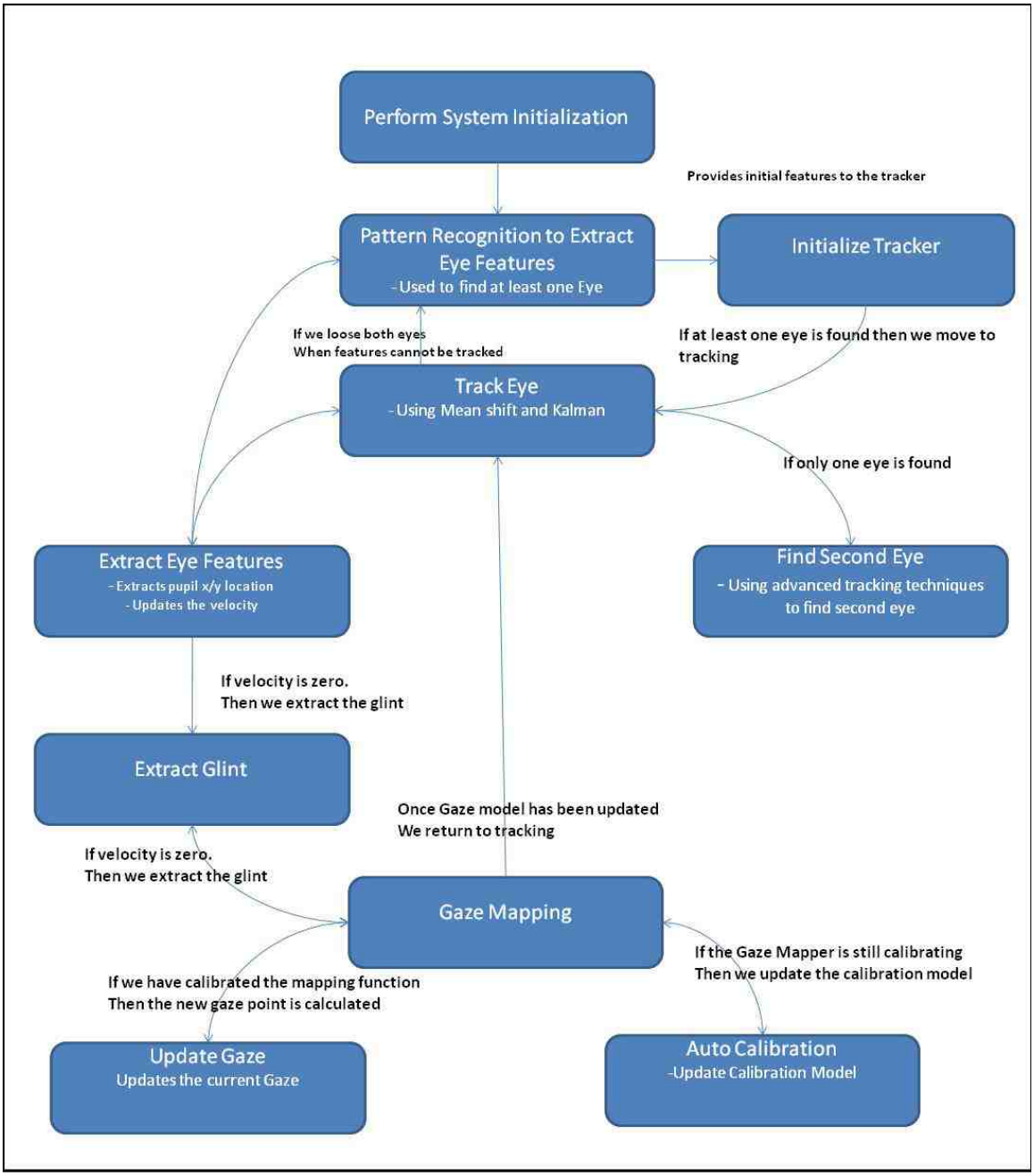


Fig. 3.1. Software System Overview

locations of the pupil centers, the velocity and acceleration of the eyes, the iris radii, and the X/Y locations of all the gli nts in the pupils. The Eye Tracker outputs one

region within which the pupil of a particular eye could potentially be, and this region is passed to the feature extractor. If both eyes are being tracked then the left eye will be used as the primary eye for gaze mapping, else the right eye will be used.

If only one eye is tracked then after extracting the features and mapping the gaze using that eye, a quick eye detection technique is used to find the second eye. Furthermore this eye detection technique also updates the locations of other facial features such as the nose and mouth as in [33].

The gaze mapping system has two parts: the calibration module and the mapping module. The calibration is done only once each time the system is activated. During calibration, gaze mapping is not performed, although face pose information can still be used to discern driver state information. Once the system is calibrated, gaze mapping commences. The eye tracker module passes glint positions, pupil positions and the papillary diameter to the gaze mapping module. Based on this information the gaze mapper can provide the real world Cartesian coordinates (X,Y,Z) of where the driver is looking. This information is matched with one of the ROIs.

3.2.1 Key System Requirements

The key requirements for the system are listed below:

- **Real-Time performance:** The system has to be real-time. Hence the need to make sure that the image processing is efficient. The system currently processes 30 Frames Per Second (Fps) and the size of the frame is 1200×720 pixels.

- **Robustness:** The system has to be robust; it has to work with large head movements, extreme poses, and eye region occlusion.

- **Accuracy:** As described in the introduction, this system does not have to be as accurate as gaze mapping systems for other purposes; however, it has to correctly identify the ROI in the vehicle.

- **Modularity:** The eye tracker and gaze calibration systems are modular in software design to allow the easy switch between 2D and 3D gaze mapping. Furthermore the gaze mapper can be easily integrated with other DSM (Driver State Monitor) modules such as eye closure and distraction meters.

- **No need for Prior Calibration:** This is the most important feature which has been implemented. The biggest hurdle to commercial implementation of Eye Gaze Trackers is the requirement for calibration.

3.2.2 Integration with other Vehicle Systems

This system can easily be integrated with other vehicular systems via the Controller Area Network (CAN). Supervisory control and the lane centering systems would benefit the most from this eye gaze tracker. Table 3.1 below shows the information this system will provide periodically over CAN.

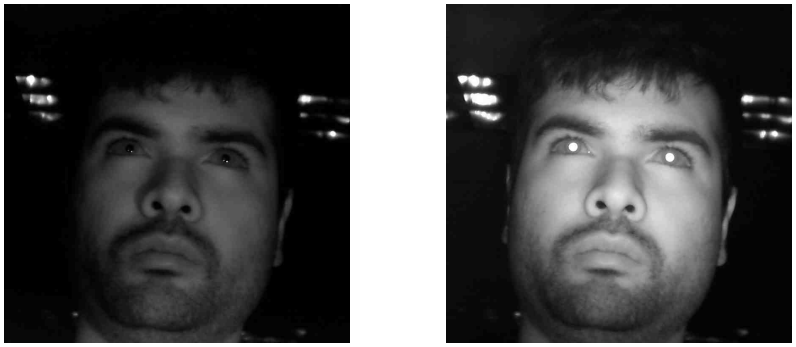
Table 3.1
Data Provided by Eye Gaze Tracker

#	Data
1	ROI viewed by Driver
2	Eyes currently tracked
3	Face Pose Direction
4	Eye Closure Data

It is important to note that it is easy to add other DSM modules to the eye gaze tracking system and create a complete DSM system, which would provide and inform the vehicle the driver's status with respect to driver state.

3.3 Hardware Model Overview

The camera model (for 3D stereo vision, two cameras using the same configuration are used) is based on the work done by [8] and [19]. This model allows the creation of bright and dark eye images as shown in Figure 3.2. The ability to create even and odd image alternatively reduces image processing complexity and makes the system more accurate.



(a) Odd Image (b) Even Image

Fig. 3.2. Even/Odd Image

Morimoto's design shown in Figure 3.3, involved two rings of LEDs [8]. When the outer layer is lit up it creates the odd image. When the inner layer is lit up it creates the even image. The inner layer of LEDs is aligned with the Z-axis of the camera. As a result the pupil reflects the light back directly to the camera creating the bright pupil effect.

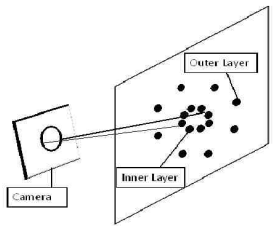


Fig. 3.3. Morimoto's Camera Design

3.3.1 Camera Circuit Design

The setup used in this research is slightly different as seen in Figure 3.4. In Morimoto's design the outer ring of LEDs was used to light up the face. However, the design implemented in this research does not utilize the outer ring, instead LEDs are placed at strategic locations around the vehicle. The placement of the LED's in different locations also help with gaze calibration. This is similar to the method of by [36], who placed four sets of LEDs at the corners of a target area and then used the glints from the LEDs to help with eye gaze detection. The experimental hardware system is shown in Figure 3.4 below. Figure 3.5 below shows the camera in greater detail.

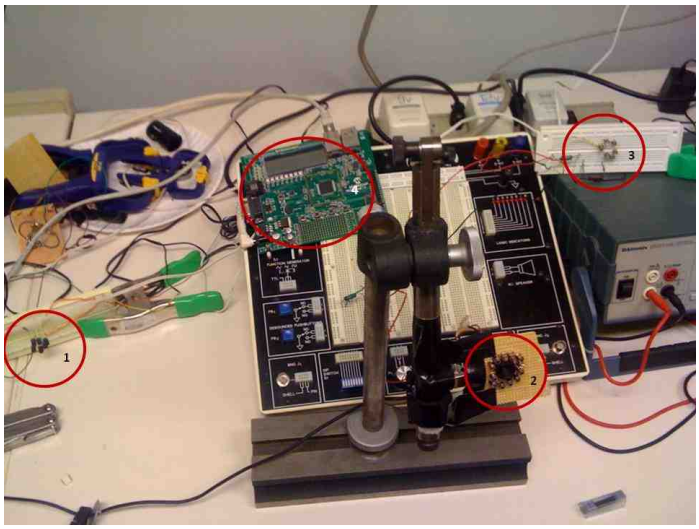


Fig. 3.4. Lab test bed with external LEDs

The four red circles in Figure 3.4 represent the camera, embedded hardware system to control the even/odd oscillations and the two reference LEDs that are used for gaze mapping. Circles 1 and 3 represent LEDs; circle 2 is the camera system; Circle 4 is the hardware that communicates serially with the software system, via an opto-coupler that turns the LED's around the camera on and off to create the even/odd image sequence.

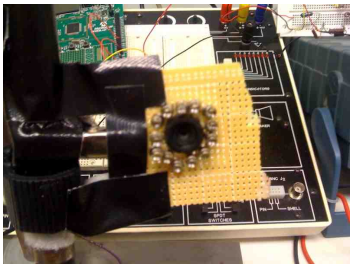


Fig. 3.5. Camera Used in Research

If a 3D stereo-vision system is used, making gaze mapping easier, but restricting areas where the cameras can be placed in the vehicle, both cameras have to have the circuit set up as shown if Figure 3.5. This will enable both the camera systems to observe the bright pupil effect.

The LEDs used have an IR-wavelength of 860nm. [8] suggested using different wavelength LEDs to enhance different types of glints. As a result different LEDs ranging from 750nm to 920nm were tested. However no difference in pixel intensity or pattern was observed.

3.3.2 Camera and LED Placement in Vehicle

LEDs were placed on each of the three mirrors in the vehicle. In the test-bed the LEDs were in the center of the mirrors as shown in Figure 3.6. This location was found to give the most accurate results. In the simulator, LEDs were placed in each of the three mirrors as shown in Figures 3.7 to 3.9

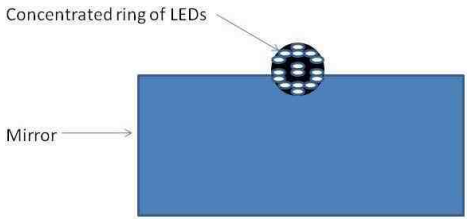


Fig. 3.6. Ideal placement of LEDs in the mirrors

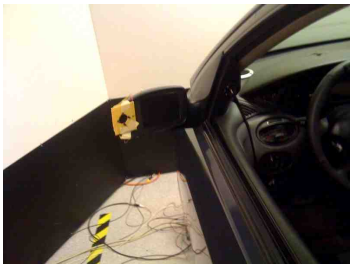


Fig. 3.7. LEDs on the driver side mirror

Fig. 3.8. LEDs on the passenger side mirror

Fig. 3.9. LEDs on the rear view mirror

Placement of the camera is tricky. The most important constraints to consider when placing the camera in the vehicle are:

- **Camera should ALWAYS be positioned below the driver's head:** This is done to reduce occlusion. We noticed that if the camera is placed above the driver's head then eyelashes greatly obstruct the view of the eye. In addition, if the driver is looking down the eye will not be seen. Figure 3.10 shows where the camera should be placed with respect to the human head. The Z-axis of the

camera should ideally make an incident angle of $\theta = 30^o$ with the X/Y plane of the human head.

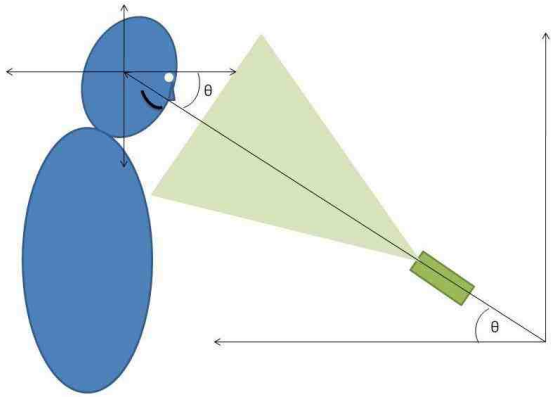


Fig. 3.10. Angle between the Z-axis of the camera and the X/Y Plane of the human head. Ideally $\theta = 30^o$

- **Camera should be far enough from the eye to allow for the bright eye effect to work:** This point is relevant to the image acquisition system used in this research, which uses the bright eye effect to extract the eyes. Sometimes if the driver is too close to the camera system, then the bright eye effect does not work very well because some LEDs take a longer time to diffuse the IR light waves until they are traveling parallel to the Z-axis of the camera.

- **Camera should be able to capture full length of driver's lateral head movement:** The driver's movements are restricted in a car due to seating arrangements. The maximum sideways movement of the human head can be quantified. The camera should have a wide angle lens (or be positioned far enough from the eyes) to take advantage of this.

- **Camera should be able to track at least one eye for all the ROI's:** For every ROI in the vehicle at least one of the eyes should be visible to the camera when the driver is looking at that particular ROI.

- **Minimize occlusion from hands and steering wheel:** The camera must be placed in an area where the hands cannot block its view. If this is not possible then at least interference must be minimized, and quantified.

- **Stereo Camera Constraint:** The stereo cameras must be parallel to each other in order to have a good solution. This adds a further constraint on where they can be placed.

Figure 3.11 shows camera locations in the simulator that satisfy the constraints above. The advantages and disadvantages of each of these positions are as follows:

- **Location 1:** Entertainment console
  Advantages: All ROIs can be viewed, minimum interference from hands
  Disadvantages: Interference from hands on rare occasions, no stereo camera implementation

- **Location 2:** Far side of the vehicle dash
  Advantages: All ROIs can be viewed, no interference from hands, can be used for stereo camera solution
  Disadvantages: If the driver is too short their face may not be visible very well

- **Location 3:** Dashboard cluster
  Advantages: All ROIs can be viewed, driver is always visible, can be used for stereo camera solution
  Disadvantages: Lots of interference from hands, dashboard cluster is usually a priority for other HMI indicators

- **Location 4:** On top of dashboard cluster
  Advantages: Good frontal and upward view, Minimum interference from

hands, can be used for stereo camera solution

Disadvantages: No downward view for short people, interference from hands on rare occasions

- **Location 5:** Two cameras on the side frames (multiple view solution)
  Advantages: All ROI can be viewed, no interference from hands
  Disadvantages: Multiple cameras add computational complexity

Fig. 3.11. Potential locations to place the camera in the simulator

4. IMAGE PROCESSING

The main objective of image processing in this research is to make it easier to highlight the eye regions and extract other features. This chapter covers the initial set of image processing techniques performed on the image frames after they are acquired from the image acquisition system.

4.1 Overview

Figure 4.1 is the image processing state diagram, it can be seen that image processing is performed sequentially. Initially the even and odd images are extracted, then the odd image is subtracted from the even image. The odd, even, and subtracted images are shown in Figure 4.2. The subtracted image goes through a morphological operation called top-hat, which brings out the bright eye regions in the image frame. Next the image is thresholded. The thresholding operation is followed by a rectangular dilation operation creating a binary image with blobs. These blobs can be used as masks for eye tracking or as input regions for eye detection.

4.1.1 Image Subtraction and Top-hat Morphology

After the odd image is subtracted from the even image, the subtracted image, shown in Figure 4.2(c), displays the bright pupil region, making it stand out compared to the rest of the face. The subtraction operation also reduces background noise reducing the need for further image processing to clean up the image. So [8] and [19] directly performed adaptive thresholding on the subtracted image to extract the binary image with blobs. In general, adaptive thresholding is sufficient but in the rare case the bright eye effect is not sufficiently prominent, the resulting subtracted image

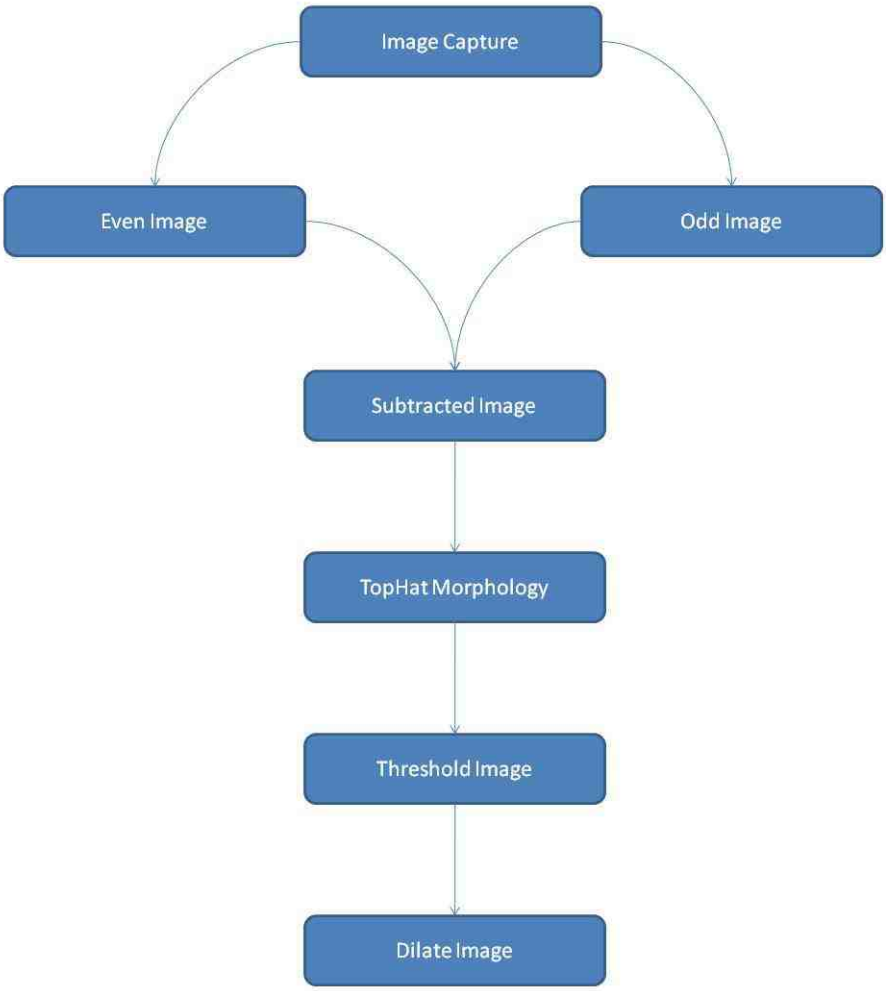
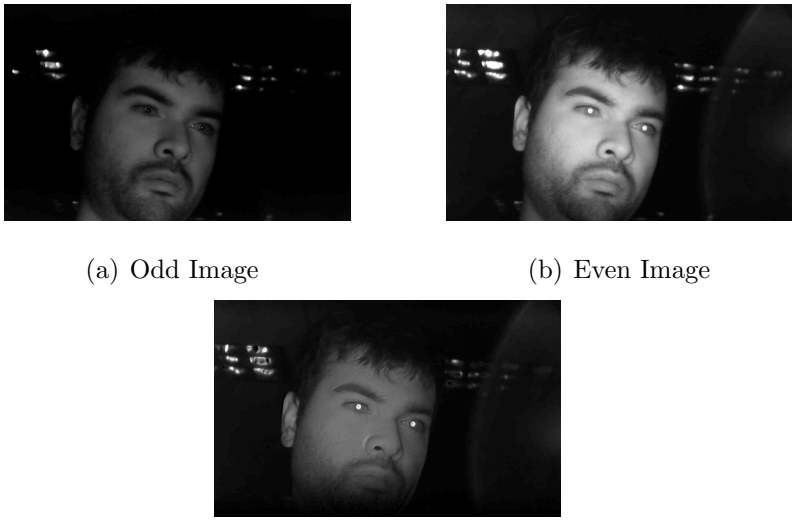


Fig. 4.1. Eye region extraction maximization

will not have a distinct bright white region. Hence a top-hat morphology operation is employed with a cross-shaped kernel of size 10x10 pixels. Top-hat morphology is a image processing technique that enhances regions that are brighter than their surrounding regions. Figure 4.3 shows the image after passing through the morphology top-hat operation.



(a) Odd Image (b) Even Image

(c) Subtracted Image

Fig. 4.2. Even/Odd and Subtracted Image

Fig. 4.3. Eye region after extraction maximization using the top-hat operation

The main drawback with the top-hat operation is that even relatively dark regions will get brighter if their surrounding regions are even darker. Hence the image must be thresholded to reduce noise and to convert it to a binary image.

4.1.2 Thresholding and Dilation

The morphed image is thresholded in order to create a binary image that will contain the eye blobs (along with a few other similar looking blobs). After dilation this image can be used as a mask for eye tracking or for eye detection. Another reason

to threshold the image is to eliminate noise, especially noise pixels generated by the top-hat operation. The thresholded image for Figure 4.3 is shown in Figure 4.4.

Fig. 4.4. Thresholded Image

The thresholded image then goes through a dilation operation. This operation uses a square kernel of size 10x10 pixels. Blob regions that are equal to or greater than the kernel size are grouped in a rectangle, while smaller regions are ignored as shown in Figure 4.5.

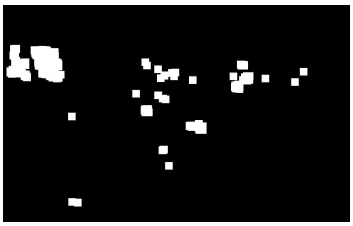


Fig. 4.5. Final Dilated blob Image

4.2 Summary

The techniques employed above highlight the eye regions used for eye detection and tracking. They are very reliable and robust under most conditions. Thus they provide an effective eye gaze detection system.

It is important to note that this procedure is performed on all image frames, hence it takes the bulk of the computational processing time. There are numerous ways to reduce this processing time. Examples include performing background subtraction and decreasing the resolution.

5. INITIAL EYE DETECTION

5.1 Introduction

Initial eye detection is very important in any eye tracking system. It is important not only for initializing the system by getting the initial eye candidates, but also to obtain new candidates if the tracker loses track of the eyes. This algorithm has to be robust enough that if the image processing module provides a positive eye candidate, the module should be able to identify it as an eye despite variation in head pose or partial occlusion of the eyes.

We replicated the State Vector Machine (SVM) based model of [19] and compared their approach with a baseline pattern recognition technique called the K-Nearest Neighbor (KNN) approach. Input data was collected from a variety of candidates, under various face poses and occlusions.

5.2 State Vector Machines

SVM [20] is a classification technique meant to distinguish between just two groups. Input vectors that are not linearly separable are non-linearly mapped to a higher-dimensional feature space. In this feature space a linear decision surface is constructed. Special properties of the decision surface ensure high generalization ability of the learning machine. Kernels that map input vectors to a higher dimension can be chosen arbitrarily, however the most common kernels used are Gaussian and polynomial. The SVM creates a discriminant function

$$G(x) = \omega \cdot \phi(x) + \omega_o, \tag{5.1}$$

where ω_o is the distance to the hyperplane from the origin, and $\phi(x)$ is the kernel function used to map a set of input feature vectors x to a higher dimensional space, and ω is the normal of the hyperplane.

If the decision function is greater than zero, then it is in class one (positive class), else if it is less than zero it is in class two (negative class). The function should ideally never equal zero, since a hyperplane should always maintain a minimum distance from the learning vectors that are at the extremities as showing in Figure 5.1.

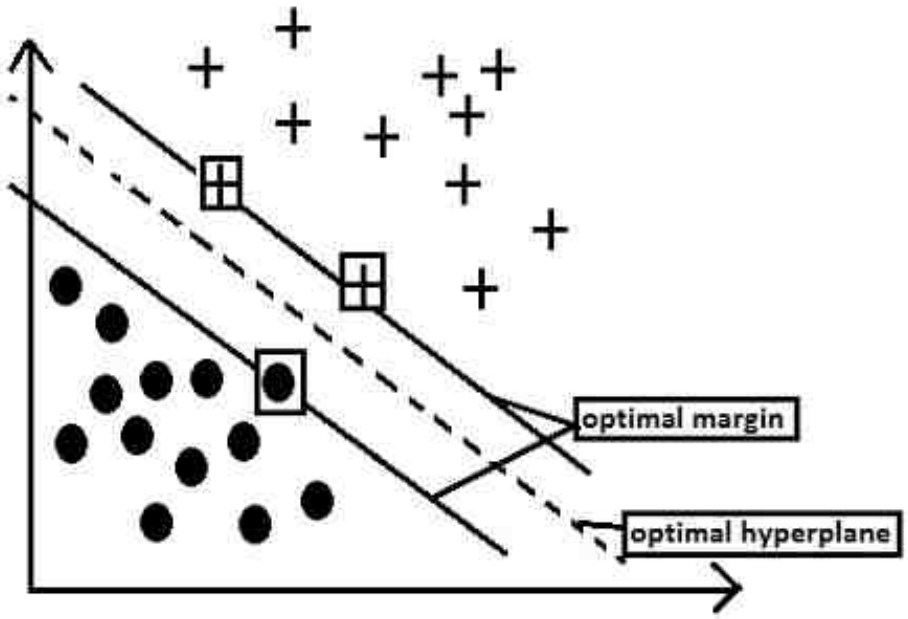


Fig. 5.1. SVM Hyperplane

During training the objective is to increase the optimal margin on either side of the optimal hyperplane. The distance between the optimal margin and the optimal hyperplane is

$$G(x) = \frac{1}{|\omega|}. \tag{5.2}$$

Thus by minimizing ω the margin can be increased, thereby reducing generalization errors. ω can be minimized by constructing the functional

$$L(\omega, \alpha) = \frac{1}{2}\omega\omega^T - \sum_i \alpha_i(y_i(\omega^T\phi(x_i) + \omega_o) - 1), \tag{5.3}$$

which is minimized with respect to ω and maxmimized with respect to α. Equation 5.3 can be reformulated as

$$L(\alpha) = \sum_{k=1}^{n} \alpha_i - \frac{1}{2}\sum_{k,j}^{n} \alpha_i\alpha_j y_i y_j \phi^T(x_i)\phi(x_j), \tag{5.4}$$

maximizing α, subject to the constraints

$$\sum \alpha_i y_i = 0, \tag{5.5}$$

and

$$C \geq \alpha_i \geq 0. \tag{5.6}$$

C is an arbitrary upper limit on the Lagrangian. The purpose of this limit is to act as a relaxation limit for the optimization algorithm. In general the higher the value of C, the higher the error rates. In practice the value of C is usually chosen by trial and error.

5.3 K Nearest Neighbor

The K Nearest Neighbor (KNN) technique (see [21], [37]) is one of the simplest and most robust pattern classification techniques. It works by comparing a potential test sample with its K nearest neighbors in the feature space. Usually euclidean distance is used as a distance metric.

Given a vector x_0, the task of the classifier is to predict the class of x_0. The KNN algorithm tries to find x_0's nearest neighbors and then uses a majority vote to determine the class of x_0. This algorithm is very useful for binary classification. The main thing to notice is that unlike SVM, the KNN technique does not have arbitrary weights that can be modified.

The main drawback of the KNN system is that all samples must be cached in memory, creating a big burden on system resources.

5.3.1 3-NN Algorithm

The 3-NN attempts to locate the 3 closest neighbors, then it uses a majority vote to classify the input vector x_0. Since this algorithm uses the weights of only the first three closest neighbors, it is faster than the regular KNN; however, it is generally less accurate.

5.4 Image Processing and Module Overview

Figure 4.1 shows how to extract eye candidate regions. The binary image is passed to the eye detection module. Geometric constraints are applied to all blobs in the binary image. Successful candidate images are normalized to a 40×40 pixel area around the corresponding odd image. Pattern recognition is performed for each of the eye candidate images. If they pass these two steps then we perform feature extraction on these images. This process is repeated until we have gone through all potential blobs or until we have extracted the position of both the eye images. Figure 5.2 gives an overview of this process.

Figure 5.3 shows a sample image with potential eye candidates. After applying geometric constraints, only the four blobs highlighted in blue circles are passed to the pattern recognition algorithms.

5.5 Training Data

A total of 100 training images were acquired from 10 subjects. There were 50 positive data samples and 50 negative data samples. The positive eye images were 40x40 pixel areas centered around the pupils. The images were taken with various head poses. The negative images were non-eye image data that could potentially be mistaken for an eye after image processing. This included noses and other facial features. Figure 5.4 shows some of the positive eye image samples. Figure 5.5 shows the corresponding negative eye image samples.

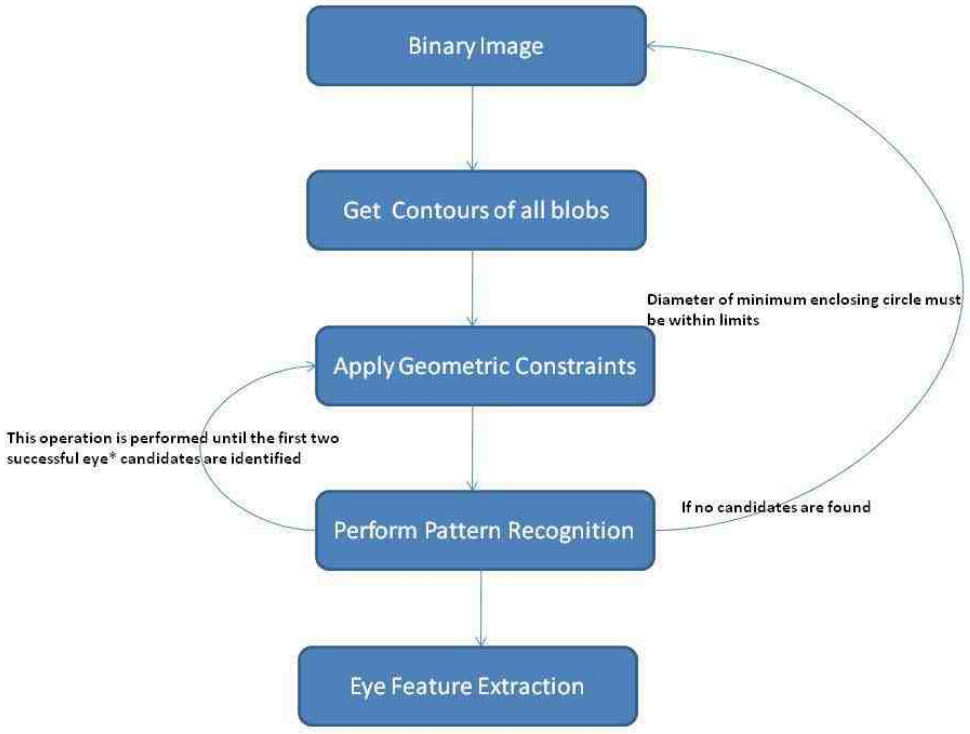
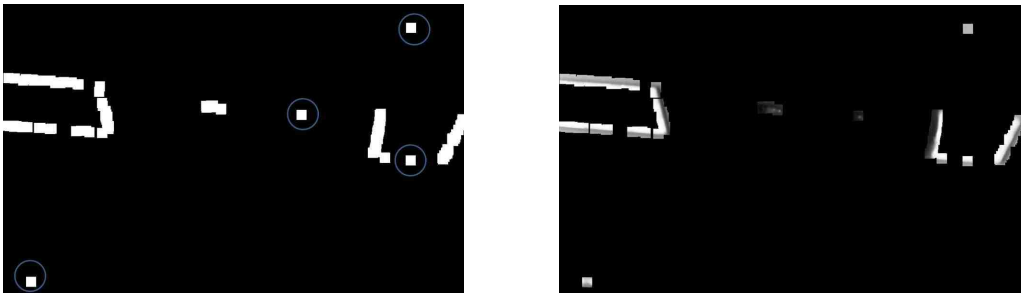


Fig. 5.2. Pattern Recognition Module Overview



(a) Binary Image (b) Corresponding Odd Image

Fig. 5.3. Initial Eye candidates

These 1600 pixels are the input features. These pupils are fed into the SVM as a 1600 input vector feature space. The training algorithm takes two matrices, X and Y as inputs.

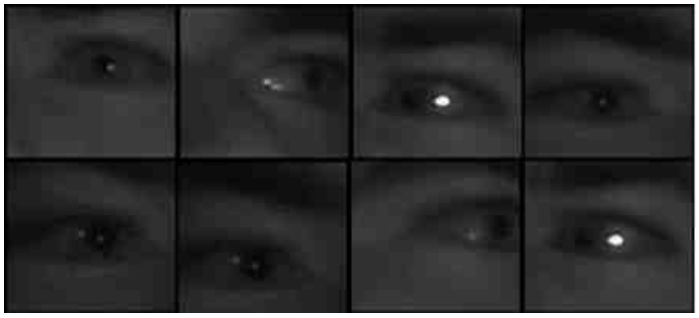


Fig. 5.4. Positive Eye Sample

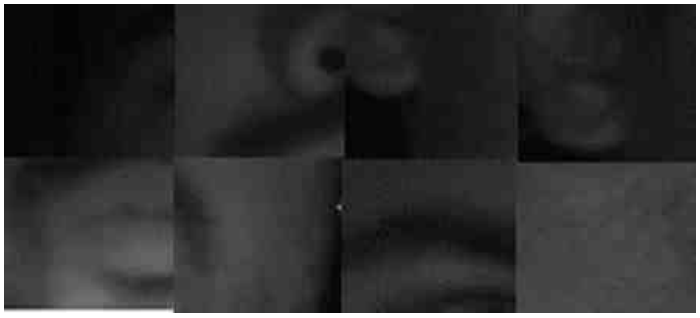


Fig. 5.5. Negative Eye Sample

- X is the set of positive and negative eye images. Each row is a separate input vector (eye image). Thus this matrix has 1600 columns. The number of rows depends on the size of the training sample.
- Y is a 1-dimensional matrix. Each corresponding row in the X matrix, is assigned either $+1$ representing a positive sample or -1 representing a negative sample.

Based on these values, and the equations and constraints provided, a quadratic programming based algorithm is used to train the SVM.

5.6 Results

The objective of the main SVM experiments was to figure out the optimal C and σ values that would minimize the number of negative images being classified as positive

eye images. A number of C and σ values, were tested on a sample of 70 images (35 positive and 35 negative). Table 5.6 shows how many negative eye images were rejected by the SVM classifier with the indicated C and σ value. Table 5.6 shows how many positive eye images were accepted by the SVM classifier. From the tables it can be seen that the classifier with values $C = 0.1$ and $\sigma = 3$ rejects the most non eye images and also classifies the most positive eye images.

Table 5.1
Number of Negative Eye Images Rejected, with sample size 35

$\frac{C}{\sigma}$	$\sigma = 1$	$\sigma = 2$	$\sigma = 3$
$C = 0.05$	29/35	28/35	27/35
$C = 0.1$	28/35	29/35	35/35
$C = 0.2$	28/35	28/35	20/35

Table 5.2
Number of Eye Images Accepted, with sample size 35

$\frac{C}{\sigma}$	$\sigma = 1$	$\sigma = 2$	$\sigma = 3$
$C = 0.05$	29/35	35/35	34/35
$C = 0.1$	29/35	34/35	34/35
$C = 0.2$	29/35	34/35	35/35

The KNN and 3-KNN classifiers were trained using the same input X and Y matrices. The test samples were run on both. Table 5.6 shows that a high number of negative eye images were classified as positive eye images. For the KNN to work more accurately, more training sample images are required.

The KNN gives an accurate picture of the efficiency of the system because the accuracy of its classification represents how well it was trained. This is unlike the SVM whose weights can be modified arbitrarily to maximize rejection on the training

Table 5.3
KNN & 3-KNN results. Sample Size is 35

	Negative eye images rejected	Positive eye images accepted
3NN	29/35	29/35
KNN	28/35	29/35

samples. Unfortunately the KNN would not be a very efficient solution for a real-time system; however, it can be used to determine the optimal SVM hyperplane coefficients.

In order to choose ideal SVM coefficients that would be practical for a real-world eye-detection application the training samples and the test samples should include:

- eyes from subjects of a variety of ethnicities,
- images taken as close to and as far as possible from the camera, and
- eyes looking at each of the ROIs for the three different face poses (forward, left and right).

A KNN classifier would then be trained and tested on these samples. The SVM coefficients that give a similar result to the KNN provide a practical classifier. Ideally the system should be trained for a large number of eye samples satisfying the criteria above.

5.6.1 Summary

In this chapter a SVM algorithm was developed for initial eye detection. We noticed that the SVM hyperplane coordinates that produced similar results to the KNN algorithm during training of the sample set, had improved performance on the larger dataset compared to SVM hyperplane coordinates that were optimized on the

training dataset. We also presented a set of criteria for choosing optimal training and testing samples.

6. EYE TRACKING

6.1 Introduction

There are numerous image processing techniques for tracking objects. The hybrid Kalman/Mean-shift algorithm of [19] was found to be efficient for large data sets, because its running times were faster than other techniques and tracked well under facial deformations. The other methods used to track the eyes were either brute force (shape and template matching techniques) or pattern recognition techniques are not very efficient when the data sets are large, for example 1200×720 as in our case.

We used [19]'s hybrid Kalman/mean-shift algorithm as a baseline, to which we compared, various computer vision tracking algorithms such as mean-shift and continuously adaptive mean-shift (cam-shift). Based on these comparisons we concluded that the hybrid Kalman/mean-shift algorithm was an efficient tracking scheme. However our Kalman filter differed from that of [19]. The Kalman's state equations were modified to a second order system (inclusion of acceleration parameters), which increased the efficiency and accuracy of the tracker.

6.2 Computer Vision Tracking Techniques

Computer vision techniques can be split between tracking algorithms such as Mean-shift tracking and prediction algorithms such as the Kalman Filter. Tracking algorithms find the most probable region that represents the object being tracked in the next consecutive frame. They are usually recursive solutions. Prediction algorithms usually predict the location of an object based on knowledge about the system, e.g. equations of motion. They need a distance measure that can locate the exact

region being tracked in the next consecutive frames. The sub sections below describe the different tracking and prediction techniques.

6.2.1 Mean-shift Algorithm

The mean-shift algorithm [24] is a robust, iterative, appearance-based object-tracking algorithm. It uses the mean-shift analysis technique to find the target candidate in the next consecutive frame ($\hat{p}$) that has similar pixel density probability to the eye model region in the current frame ($\hat{q}$).

The mean-shift analysis technique is very similar to kernel density estimation [38], which uses a kernel function, e.g. Gaussian distribution, that assigns weights to a set of data points (pixel values in a window). Mean shift then seeks to estimate the gradient (direction of change) of the data distribution. To create a mean-shift vector that will point to where the mean-shift window will move to next and then reiterate this process. When the difference in pixel density probability between the eye model image region $\hat{q}$ and the target region $\hat{p}$ is below a threshold, we say that their probability distributions match.

The probability of the intensity for a particular bin in a histogram of the region in a particular image frame, is computed by

$$\hat{p}_u(y) = \frac{\sum_{i=1}^{n_h} k(\|\frac{y - x_i}{h}\|^2)\delta[b(x_i) - u]}{\sum_{i=1}^{n_h} k(\|\frac{y - x_i}{h}\|^2)}, \tag{6.1}$$

where h is the window size, δ is the Kronecker delta function, k is a kernel function (e.g. Gaussian or rectangular), $b(x_i)$ is the index of the histogram bin for the particular pixel intensity value x_i. u identifies the bin for which the probability is being calculated, and y represents the pixel at the center of the Mean-shift window. Both

$\hat{q}_u(y)$ and $\hat{p}_u(y)$ are calculated using Equation 6.1. From the probability distributions of the m bin histogram the Bhattacharyya coefficient $\rho(y)$ is calculated as

$$\rho(y) = \rho[\hat{p}_y, \hat{q}] = \sum_{u=1}^{m} \sqrt{\hat{p}_u \hat{q}_u}. \tag{6.2}$$

Then the distance between the two distribution is defined as

$$dist[\hat{p}_y, \hat{q}] = \sqrt{1 - \rho[\hat{p}_y, \hat{q}]}. \tag{6.3}$$

The objective is to bring the distance below some threshold value ϵ, as described below. The Mean-shift algorithm consists of the following steps:

1. Calculate the eye model in the current region $\hat{q}$ using Equation 6.1.

2. Initialize the target location y_0 in the next image frame by centering it at the same location as the current eye model region. Then compute $\hat{p}_y$ from Equation 6.1.

3. Assign weights w_i to each pixel location x_i in the region based on its current intensity according to
$$w_i = \sum_{u=1}^{m} \delta[b(x_i) - u] \sqrt{\frac{\hat{q}_u}{\hat{p}_u}}. \tag{6.4}$$

4. The new location of the eye target (y_1) is
$$y_1 = \frac{\sum_{i=1}^{n_h} x_i w_i g(\| \frac{y_0 - x_i}{h} \|^2)}{g(\| \frac{y_0 - x_i}{h} \|^2)}, \tag{6.5}$$
where the mean-shift vector $g(x)$ is obtained from the kernel gradient function as $g(x) = -k'(x)$.

5. If the distance between the two windows is less than the chosen threshold then stop, else go back to step 2.

Mean shift is a very robust algorithm, especially if the movement of the eye is not very fast. The practical usage of this algorithm depends on many factors that

weigh accuracy vs. efficiency, including the threshold ϵ, the maximum number of iterations allowed, and the search window size. Using a larger search window and a larger number of iterations usually improves tracking. However, with a large search window there is a possibility that the mean-shift algorithm will track a non-eye region that has a similar intensity profile.

Cam-shift

The cam-shift algorithm is very similar to the mean-shift algorithm. The difference is that the ROI dynamically changes from frame to frame as the size of the object changes. Cam-shift is used widely for face tracking and in applications where the region tracked can change in size significantly, but still occupy a significant portion of the image frame.

6.2.2 The Kalman Filter

Kalman filtering is a method that predicts the most probable next state. The Kalman Filter was first used in image contour tracking by [23]. The basic idea is that the next estimate (location of the object being tracked) is a weighted combination of the previous measurement based on their uncertainties.

The theory behind the Kalman filter can be explained using a 2D scalar process. Assume that x_1 and x_2 are uncertain measurements, with a Gaussian distribution, with means $\bar{x}_1$ and $\bar{x}_2$, and standard deviations σ_1 and σ_2. Hence $\bar{x}_{12}$, the prediction of the next state,is a weighted combination of the two prior input measurements,

$$\bar{x}_{12} = \left(\frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2}\right) x_1 + \left(\frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2}\right) x_2. \tag{6.6}$$

Once the next value is measured the posterior value $\hat{x}_2$ is

$$\hat{x}_2 = \hat{x}_1 + K(x_2 - \hat{x}_1), \tag{6.7}$$

where

$$K = \frac{\hat{\sigma}_1^2}{\hat{\sigma}_1^2 + \sigma_2^2}, \tag{6.8}$$

$\hat{x}_2$ is a weighted sum of the current measurement and all previous measurements. If the uncertainty of the current measurement is low then its contribution is large. If the current measurement is noisy then the covariance σ_2^2 will be high and the contribution of the current measurement to the sum will be low.

The contribution of the current uncertainty covariance $\hat{\sigma_2}^2$ to the system uncertainty $\hat{\sigma}_1^2$ is

$$\hat{\sigma}_2^2 = (1 - K)\hat{\sigma}_1^2. \tag{6.9}$$

Kalman filters in motion tracking can be modeled as dynamic systems, [4], [19] and [22] used first order Kalman filters for eye tracking. To improve accuracy we used a second order system. The horizontal and vertical components of the eye's position in the next frame are

$$c_{t+1} = c_t + u_{ct} + \frac{1}{2}a_{ct}, \tag{6.10}$$

and

$$r_{t+1} = r_t + u_{rt} + \frac{1}{2}a_{rt}. \tag{6.11}$$

c_t and c_{t+1} are the vertical components or column positions of the eye in the current and next image frame. u_{ct} is the vertical component of the velocity (column axis). u_{rt} is the horizontal component of the velocity (row axis), a_{ct} and a_{rt} are the vertical and horizontal components of the acceleration.

The vertical and horizontal components of the eye's velocities in the next frame are

$$u_{c(t+1)} = u_{ct} + a_{ct}, \tag{6.12}$$

and

$$u_{r(t+1)} = u_{rt} + a_{rt}. \tag{6.13}$$

The dynamic state equation representing the second order system is

$$X_{t+1} = \Phi X_t + W_t, \tag{6.14}$$

X_{t+1} is the state of the system in the next time instant (next frame), X_t is the current state of the system (current frame), Φ is the state transition matrix, and W_t is the system perturbation. The current state of the system can be described as

$$X_t = \begin{pmatrix} c_t \\ r_t \\ u_{ct} \\ u_{rt} \\ a_{ct} \\ a_{rt} \end{pmatrix}, \tag{6.15}$$

and the state transition matrix as

$$\Phi = \begin{pmatrix} 1 & 0 & 1 & 0 & 1/2 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1/2 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \tag{6.16}$$

The system perturbation is normally distributed as

$$W_t \approx N(0, Q), \tag{6.17}$$

and the system noise covariance as

$$Q = \begin{pmatrix} 16 & 0 & 0 & 0 & 0 & 0 \\ 0 & 16 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{pmatrix}. \tag{6.18}$$

The system noise is due to the random nature of the driver's head movement and the assumptions made in the second order dynamics equations that represent the system.

These values were chosen by trial and error based on the test subjects. The estimated row and column values have an uncertainty of 16 pixels, while the horizontal and vertical velocities have an uncertainty of 4 pixels. Finally the horizontal and vertical accelerations have an uncertainty of 2 pixels.

Since the camera system provides the row/column location of the eye feature in each image frame, the feature extraction model estimates the row/column (X/Y axis) position of the eye. We denote the measured input vector by

$$Z_t = (\hat{c}_t, \hat{r}_t). \tag{6.19}$$

Then the measurement model for the Kalman Filter is

$$Z_t = HX_t + V_t, \tag{6.20}$$

where the current state X_t, current input vector Z_t are related by H, which is

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{6.21}$$

V_t is the measurement uncertainty which is normally distributed

$$V_t \approx N(0, R), \tag{6.22}$$

where the measurement error covariance matrix is

$$R = \begin{pmatrix} 6 & 0 \\ 0 & 6 \end{pmatrix}. \tag{6.23}$$

This noise is due to inefficiencies in the image capture and image processing techniques. The value of R which means the camera system has a measurement error of 6 pixels was chosen by trail and error.

Furthermore in more than one dimension, the covariance σ_i of the state x_i becomes the covariance matrix Σ_i for state vector X_i. The covariance matrix was initialized to

$$\Sigma_0 = \begin{pmatrix} 100 & 0 & 0 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 & 0 & 0 \\ 0 & 0 & 25 & 0 & 0 & 0 \\ 0 & 0 & 0 & 25 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 \end{pmatrix}, \tag{6.24}$$

assuming no cross-correlation. The diagonal elements can be viewed as indicating the relative magnitudes of uncertainty in the states.

The goal of Kalman filtering is to estimate the tracked pupil location in the next frame. The Kalman prediction involves two steps: a prediction step followed by an update step.

- **Prediction Step**

  In the prediction step, the Kalman filter estimates both the location of the pupil in the next frame and the uncertainty associated with the estimate. The estimated state is

  $$X_{t+1}^{-} = \Phi X_t, \tag{6.25}$$

  and the error in the estimation is

  $$\Sigma_{t+1}^{-} = \Phi \Sigma_t \Phi^t + Q_t. \tag{6.26}$$

  The feature extraction algorithm will now use the row and column values in X_{t+1}^{-}, along with a search window whose dimensions are

  $$Width = 50\Sigma_{t+1}^{-}(1,1), \tag{6.27}$$

  and

  $$Height = 50\Sigma_{t+1}^{-}(2,2). \tag{6.28}$$

  The minimum search window size used is 50×50 pixels. The larger the uncertainty, the larger the resulting search window.

- **Update step**

  Once we extract the pupil locations we update the Kalman model, creating the posterior state estimate X_{t+1}. This estimate incorporates both prior estimate X_{t+1}^{-} and the measured pupil position z_{t+1}. The Kalman gain K_{t+1} determines the extent to which each of the two values contribute to the calculation of X_{t+1}.

  The new gain is

$$K_{t+1} = \frac{\Sigma_{t+1}^{-} H^T}{H\Sigma_{t+1}^{-} H^T + R}, \tag{6.29}$$

  where the estimated error measurement is Σ_{t+1}^{-} and the actual measurement noise is R. The posterior state estimate and the posterior error covariance matrix estimate are then given by

$$X_{t+1} = X_{t+1}^{-} + K_{t+1}(Z_{t+1} - HX_{t+1}^{-}), \tag{6.30}$$

  and

$$\Sigma_{t+1} = (I - K_{t+1}H)\Sigma_{t+1}^{-}. \tag{6.31}$$

The beauty of the Kalman filter is that each new measurement is used to improve the existing model. After the Kalman filter has run for a long time, new measurements that are not in the historical motion path are considered noise and do not change the estimate very much,

6.2.3 Modeling System Noise

In our application the system noise is essentially the uncertainty associated with the driver's motion. The system has no knowledge of where the eyes are going to move next, thus we use a constant noise in Equation 6.18.

Of course this is an oversimplification: the driver's facial and eye movements over time do follow a path. There are two potential ways in which the system noise model could be improved:

1. **Expressing noise as a function of velocity:** Under normal circumstances the head follows a path. Thus the uncertainty in predicting its location in the next time instant is much less than when the face is stationary. The faster the eye moves, the more likely its location in the next time instant will lie on the current path, because it is harder to change direction at a higher velocity.

   We attempted to model the noise as a function of velocity; however the frame rate of (30Fps) the camera system used was too low. We estimate that a frame rate of 60 Fps would suffice.

2. **Noise estimation based on movements among ROIs:** A signification portion of head movements were from one ROI to another, especially the three mirrors and the front driving scene. Certain paths among ROIs are more common than others. This probability could be modelled.

6.3 Kalman/Mean-shift Hybrid

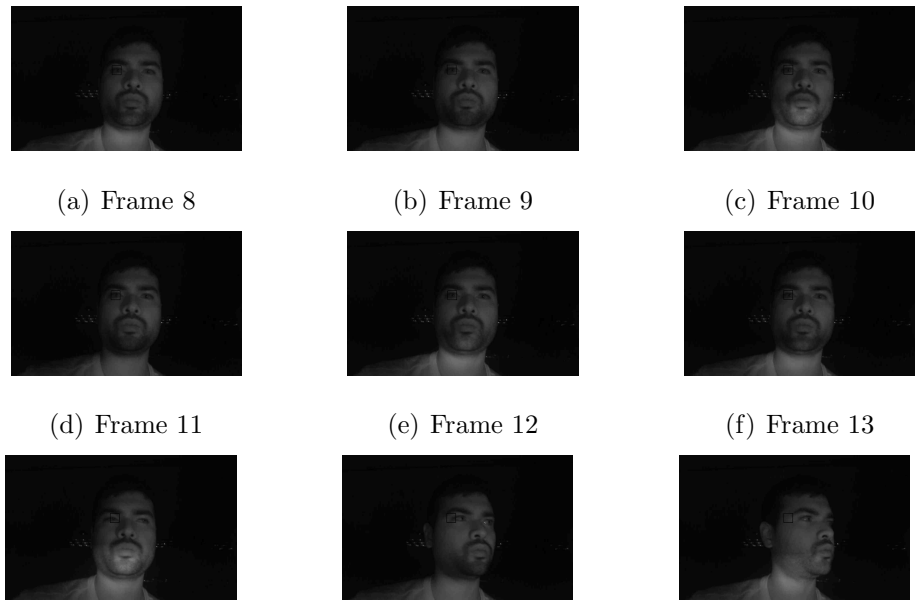
The basic idea behind the hybrid algorithm is that if the search window is larger than a pre-determined minimum area, in this case 70×70 pixels, then we augment the tracking system by including a Mean-shift.The result of the estimation step is fed into the mean-shift algorithm so that when there are large random head movements we can easily keep track of the eyes (and track face pose).

6.4 Results

The mean-shift, cam-shift and Kalman/mean-shift algorithms were tested on a set of images that contained both small and large head movements. Specific tests are described below.

6.4.1 Mean-shift under Various Search Window Sizes

This section presents results from using two search windows; 50×50 and 100×100 pixels. Figure 6.1 shows a set of frames under the 50×50 pixel search window. Figure 6.2 shows the same set of frames with the larger 100 ×100 pixel search window. It can be seen that the 50×50 search window works fine when the eye movements are small; however, during large eye movements the tracker loses track of the eye. With the larger window of 100 ×100 pixels the tracker does not lose the eye. However the 100×100 pixel requires more computational processing time, not only for the mean-shift tracking but also during feature extraction. Furthermore the larger window occasionally tried to jump to the other eye, or track a non-eye region that had an intensity similar to the eye region.



(a) Frame 8 (b) Frame 9 (c) Frame 10

(d) Frame 11 (e) Frame 12 (f) Frame 13

(g) Frame 14 (h) Frame 15 (i) Frame 16

Fig. 6.1. Mean-shift with a window of 50×50 pixels

(a) Frame 8 (b) Frame 9 (c) Frame 10

(d) Frame 11 (e) Frame 12 (f) Frame 13

(g) Frame 14 (h) Frame 15 (i) Frame 16

Fig. 6.2. Mean-shift with a window of 100×100 pixels

6.4.2 Cam-shift

Next, a cam-shift tracking algorithm was applied to the same set of image frames as in Figures 6.1 and 6.2. As mentioned before the tracking window in the cam-shift algorithm changes dynamically. Figure 6.3 shows the results of the cam-shift operation. It can be seen from Figure 6.3 that cam-shift does not work well under large eye movements. Furthermore the window of the eye region is different in each frame. The main purpose of using cam-shift was to provide a reliable dynamic window around the papillary/iris region of the eye. However cam-shift does not do this, because the window size is too small for the cam-shift algorithm. Also it often mistakes larger black areas as the pupil.

(a) Frame 8 (b) Frame 9 (c) Frame 10

(d) Frame 11 (e) Frame 12 (f) Frame 13

(g) Frame 14 (h) Frame 15 (i) Frame 16

Fig. 6.3. Cam-shift algorithm

6.4.3 Hybrid Kalman/Mean-shift algorithm

Figure 6.4 shows the hybrid Kalman/Mean-shift algorithm tracking eyes. It can be seen from Figure 6.4 that the algorithm is robust to head movement. Both eyes are tracked and the search windows are generally centered on the eyes. This is very helpful for the feature extraction algorithms.

Table 6.4.3 below shows the number of corrections required for each of the three techniques, applied to a set of 120 frames. These frames had large head movements and small head movements. All three systems lost track of the eye occasionally. However, the cam-shift technique required more corrections that the others. This is a result of the tendency of the cam-shift algorithm expecting the size of the region tracked to change from frame to frame.

(a) Frame 1 (b) Frame 2 (c) Frame 3

(d) Frame 4 (e) Frame 5 (f) Frame 6

(g) Frame 7 (h) Frame 8 (i) Frame 9

(j) Frame 10 (k) Frame 11 (l) Frame 12

Fig. 6.4. Hybrid Kalman/Mean-shift under regular head movement

Table 6.1
Tests on eye frames tracked, with sample size 120

Tracking Technique	No of corrections	First missed frame
Mean-shift (100 pixel window)	26	45
Cam-shift	55	15
Hybrid Kalman/Mean-shift	12	49

6.5 Chapter Summary

The purpose of the eye tracker is to effectively track the eyes from frame to frame using available processing time. A hybrid Kalman/mean-shift algorithm was

developed and its performance compared with that of baseline mean-shift, cam-shift algorithms. The three systems were compared based on the effectiveness of tracking eyes from frame to frame.

The experiments showed that the hybrid Kalman/mean-shift algorithm, using a second order dynamic system was the most efficient of those tested. This algorithm minimized the mean-shift tracking window, reduced the number of mean-shift iterations and provided windows to the feature extractor that made it easy to extract eye features.

7. FEATURE EXTRACTION

7.1 Overview

Feature extraction involves two steps: extracting pupil features and extracting glints (if any). The pupil features that are extracted are shown in Table 7.1. Glint features that are extracted are shown in Table 7.2.

Table 7.1
Features extracted and calculated from the pupil

#	Feature
1	Location of the pupil center
2	Length of eye region
3	Radius of the iris
4	Velocity of the pupil
5	Acceleration of the pupil

Table 7.2
Glint features extracted

#	Feature
1	location of each of the glints
2	radius of the glints

The pupil feature-extraction module is given a potential region within which the pupil may exist. The module has two separate techniques through which it can try and locate a pupil. The first technique takes advantage of the bright pupil (even)

image. The second technique uses the dark pupil (odd) image. Once a pupil's features have been extracted, the iris region (using the odd image) is thresholded in order to extract a set of bright image blobs (the glints). A flow chart for this module is shown in Figure 7.1.

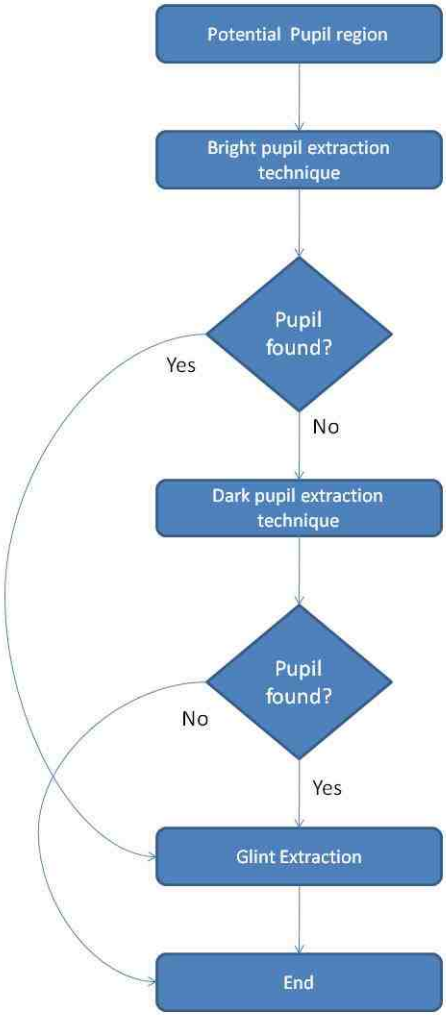


Fig. 7.1. Pupil/Glint feature extraction overview

There are two reasons, that glint features should be extracted only when the velocity of the eye is below a threshold:

1. The gaze vector can only be defined when the eyes are fixed.

2. The bright (even) and dark (odd) pupil's locations do not match.

7.2 Pupil Extraction

The eye tracker module provides the pupil extraction module a region in which the pupil could potentially be. There are two ways of extracting the location of the pupil; if the bright pupil exists then a series of image processing steps can extract its location. Otherwise the dark pupil can be extracted. However using the dark pupil tends to return much higher false positives.

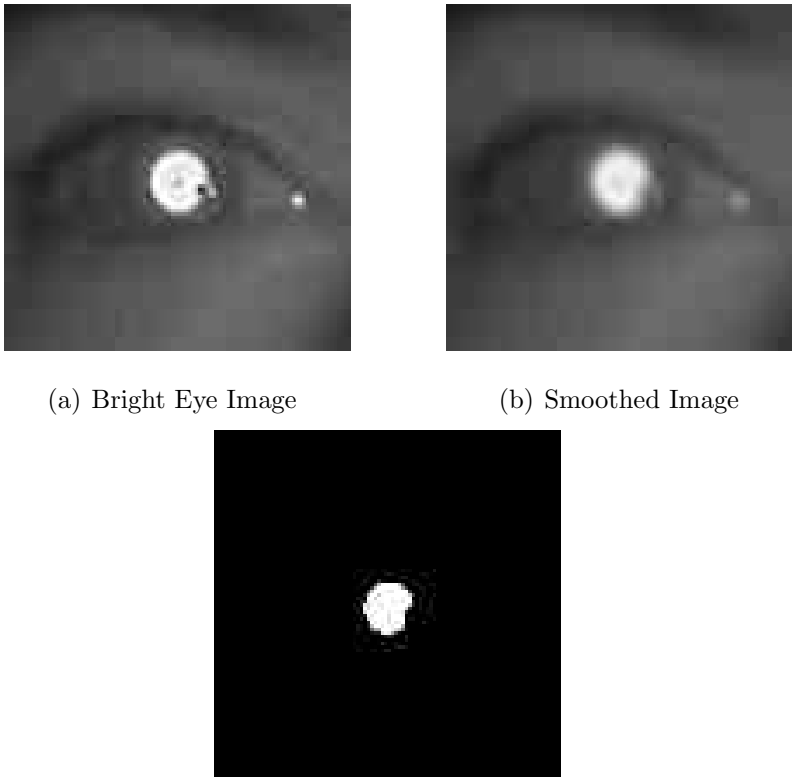
7.2.1 Bright Pupil Extraction Technique

In bright pupil extraction, the image region undergoes a smoothing process. This blurs the image, which reduces noise along with the resolution of the image, and the papillary region has a single intensity value.

Once the image has been smoothed, it is thresholded, based on the histogram profile of the region. This creates a binary image containing a single blob, which is usually the pupil region. After applying the geometric constraints shown in Table 7.3, the eye region is extracted. The geometric constraint is used because in rare circumstances there are multiple blobs corresponding to multiple light reflections on the sclera. The information extracted are the center and the radius of the pupil. From this we can calculate the current velocity and the acceleration of the pupil. Figure 7.2 shows the various steps involved in extracting the pupil region. Figure 7.3 shows the efficiency of the same process despite a bad frame capture.

7.2.2 Dark Pupil Extraction

If the bright eye pupil extraction module cannot find a suitable pupil candidate, then the dark eye pupil extraction technique is applied.



(a) Bright Eye Image (b) Smoothed Image

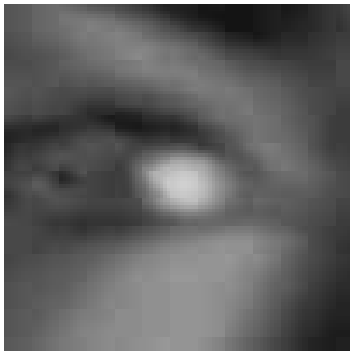
(c) Thresholded Image

Fig. 7.2. Bright eye pupil extraction

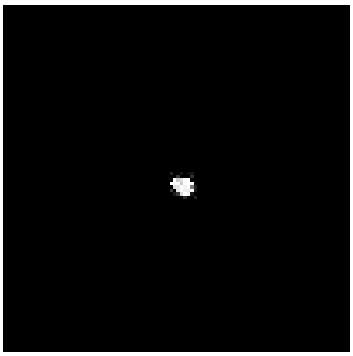
The dark pupil extraction technique works by putting the dark pupil image through an erosion process. Erosion is an image processing operation which finds the local minimum over a region defined by a kernel. Then it replaces all the pixels in a given region with that minimum value. This brings out the dark regions by suppressing the lighter brighter regions. In the context of dark pupil extraction, the dilation operation enhances all dark regions (pupil, eyebrows and nasal regions) as shown in Figure 7.4(b).

The eroded image is then put through a histogram equalization process. This further increases the contrast in the image as shown in Figure 7.4(c). The next step is to threshold the image, then extract the blobs and perform edge detection. These

(a) Poor quality bright Eye Image



(b) Smoothed Image



(c) Thresholded Image

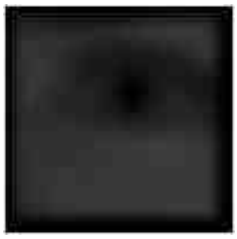
Fig. 7.3. Bright eye pupil extraction on a poor quality image

blobs candidates are put through geometric tests to be described in section 7.2.3. The most successful candidate's location and radius are extracted.

7.2.3 Geometric Constraints Applied to Pupil Candidates

Numerous geometric constraints are applied to the pupil candidate blobs. These include the baseline max/min pupil radius constraints (static constraints) and the dynamic constraints based on the face pose, the distance between the pupils, and the pupil radius in the previous frame. The baseline minimum and maximum pupil radii are calculated based on parameters of the image acquisition system used such as camera resolution, lens, maximum and minimum distance between pupil and cam-

(a) Dark eye image



(b) Eroded and Smoothed Image

(c) Histogram Equalized Image

(d) Image with eye circle

Fig. 7.4. Dark eye pupil extraction

era. The default values used in our analysis are shown in Table 7.3. The dynamic constraints are itemized below:

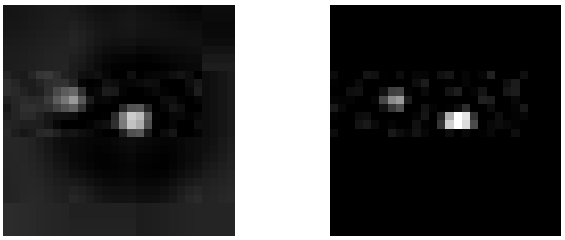
- If the horizontal velocity is positive and the face pose is either forward or left, then the current iris radius must be less than or equal to its previous value.
- If the horizontal velocity is positive and face pose is right, then the current iris radius must be greater than or equal to its previous value.
- If the horizontal velocity is negative and face pose is either forward or right, the current iris radius must be less than or equal to its previous value.
- If the horizontal velocity is negative and face pose is left, the current iris radius must be greater than or equal to its previous value.

Table 7.3
Default static constraints for pupil radius

Constraint	Value
Max pupil radius	30 pixels
Min pupil radius	4 pixels

7.3 Glint Extraction

If pupil features are successfully extracted and the velocity of the pupil is less than 5 pixels per frame, then the glint extraction module is called. The glint extraction module uses technique similar to those of [10]. The dark eye image is thresholded and all blobs having the radius of the iris are extracted. Figure 7.5(a) shows the initial iris region with two glints and Figure 7.5(b) shows the same region after thresholding.



(a) Iris region with glints

(b) Glints after thresholding

Fig. 7.5. Glint extraction

The reason the system does not extract glints unless the velocity of the eye is less than 5 pixels per frame is that the papillary location of the bright and dark eye pupils do not match from the two images. This is because the driver is moving their head from one gaze point to another. The velocity constraint value of 5 pixels per frame was chosen by trail and error.

7.4 Chapter Summary

The feature extraction module extracts the various pupil features required for eye detection, tracking and gaze mapping. If the velocity of the pupil is less than the threshold velocity, then the glint extraction module extracts the features of all the glints in the eye.

8. GAZE MAPPING

8.1 Introduction

As indicated in the literature review numerous eye gaze mapping systems have been developed in the context of human computer interaction. Most of these algorithms are used for assisted reading on computer screens. This requires high accuracy and the level of head movements are small. In the context of automotive applications, less accuracy is required but head motion may be large.

We selected three algorithms that seemed suited to automotive applications. These were the 3D gaze mapping technique of [7], the one circle algorithm of [11] and the ROI mapping algorithm of [39]. Based on the work of [12] and [11] a calibration sequence was devised, in which LEDs were placed on each of the three mirrors and as the driver looked at each mirror, the calibration sequence would update itself based on a new point of view.

After testing a sample of subjects it became evident that glint patterns in the eye were similar for all subjects on each ROI. In general there was a correlation between the glint patterns and where the subjects were looking. Using this information, a 2D neural network based approach was developed that maps the gaze based on face pose and glint positions on the pupil. This is very similar to the work done by [39]. This system can be used to generate an automatic calibration sequence for a proposed 3D gaze mapping system developed by [7].

8.2 Overview of Gaze Mapping Algorithms

The one circle, the 3D gaze mapping, and the ROI mapping algorithms are suited for automotive applications because they allow head movements and the calibration procedures are simple enough that we can try and eliminate them.

8.2.1 One Circle Algorithm

The one circle algorithm uses a 2D back projection technique very similar to that used in most other monocular eye gaze systems. Such a system requires calibration if accuracy is important. However the one circle algorithm differs from other 2D back projection algorithms in that it does not use glints to map gaze. Instead the one circle algorithm uses the distance between the pupil/iris center and the edges of the eye as shown in Figure 8.1. $P1$ and $P3$ are corner points; $P2$ and $P4$ are local maxima on the eyelids. Since the angle between the visual axis and the optical axis is constant, the relative position of the pupil center can provide an adequate estimate of the optical axis.

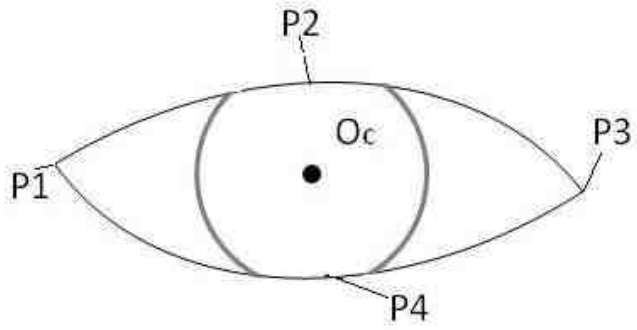


Fig. 8.1. Eye with corners highlighted

The main problem with this algorithm is that it is hard to extract the contour of the eyelids via image processing, so it is hard to extract the points $P1$, $P2$, $P3$ and $P4$. Furthermore under facial rotation the shapes of the facial features change. Another

disadvantage of this algorithm is that it is not very robust under head movement, because it requires recalibration.

However it was noticed that the ratio of the X and Y axes of the ellipse that models the pupil indicates whether the driver is looking up or down, left or right with respect to facial pose. This can be used to enhance gaze region mapping and will be discussed in further detail below.

8.2.2 3D Gaze Mapping Algorithm

3D gaze mapping works by calculating the visual axis $\vec{V_p}$ from the optical axis $\vec{V_o}$ using

$$\vec{V_p} = M^T \vec{V_o}, \tag{8.1}$$

where M^T is the transpose of the orthonormal rotation matrix. Nine predefined points are required to calibrate M. Various techniques for calculating ($\vec{V_o}$) can be found in [7].

8.2.3 ROI Mapping Algorithm

[39] devised a Generalized Regression Neural Network (GRNN) algorithm that mapped eye gaze into eight adjacent eye regions. They used an IR light source to create a glint. Based on the glint's position and other eye features, the GRNN maps the eye gaze into one of the eight regions. This algorithm is very robust to head movements.

8.3 Gaze Mapping in Automobiles

The forward view of the vehicle was split into 13 non-uniform Eye Gaze Regions, as shown in Figure 8.2, Table 8.3 details which part of the forward view each gaze region represents. The blue rectangles are the primary ROIs that we attempt to successfully identify. It is important to notice the the blue rectangles also include a

margin of two inches around the ROIs as an error margin. The red rectangles include all other regions.

Fig. 8.2. Eye Gaze Regions in the Simulator

Table 8.1
Gaze Regions

Region	Area it represents
1	Driver Side mirror - ROI1
2	Region above driver side mirror
3	region between front view and driver side mirror
4	Front view - ROI2
5	Steering wheel and dashboard
6	Rear view mirror - ROI3
7	console view
8	Passenger front view
9	Area above passenger side mirror
10	Passenger Side Mirror - ROI4
11	Area below passenger side mirror
12	Area below driver side mirror

Eye glint patterns under various face poses were observed for the four main ROIs. It soon became evident that the glints form a unique pattern for each ROI.

The initial idea was to train a simple pattern recognition algorithm such as a feed forward NN or KNN classifier to obtain a calibration sequence. The output of the pattern recognition algorithm would be the gaze point that could be used to update one of the generic gaze mapping algorithms. The 3D gaze mapping algorithm would work for four calibration points, but would not be very accurate.

It soon became apparent that the GRNN mapping technique similar to that of [39] could be used to map the various gaze regions in the vehicle. The gaze was mapped to regions of interest using a trained GRNN, while the calibration process involved retraining the algorithm for each new user.

8.3.1 Generalized Regression Neural Networks

Generalized Regression Neural Networks (GRNNs) [40] are probabilistic neural networks. The GRNN uses the training sample to estimate the response at new points. The output is estimated using a weighted average of the training cases, where the weighting is related to the distance of the training point from the current point being estimated, so that points nearby contribute most heavily to the estimate, as with the KNN classifier. The general GRNN structure, shown in Figure 8.3 consists of four layers: the input layer, the hidden layer, the summation layer and the output layer. The input layer has 14 nodes, one for each input. The hidden layer has one node for each training sample. There are 2 nodes in the summation layer and 12 nodes in the output layer. The primary advantages of GRNNs over other NNs are that they do not need a lot of training samples and that training samples can be clustered into nodes in the hidden layer; so that only 12 nodes are needed.

The objective of GRNN is to estimate the dependent variable Y based on the observed values of independent variable X. In this case the independent variable is the input gaze information vector shown in Equation 8.4, while the dependent variable is the scalar gaze region to which it maps. The dependent and independent variables are related by a Probability Density Function (PDF). The beauty of a GRNN is that

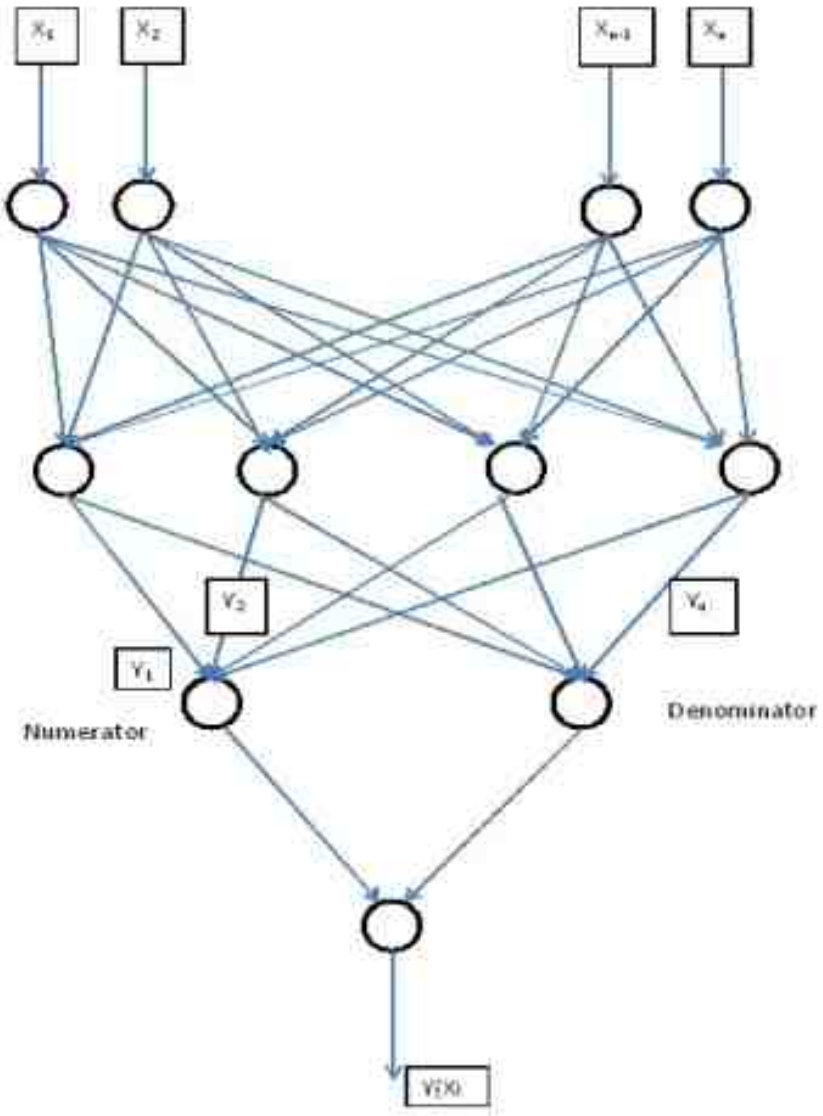


Fig. 8.3. Structure of a GRNN

the underlying PDF does not need to be known. Instead, by using estimators similar to the ones proposed by Parzen, the PDF can be estimated from the training samples.

In this system the distance between the input vectors and the training samples is

$$D_i^2 = (X - X^i)^T (X - X^i). \tag{8.2}$$

The weight of each sample input X to a trained input X_i is calculated by finding the exponential value of the distance between them, which is calculated using Equation 8.2. The estimated independent variable $\hat{Y}$ is calculated from an observation X according to

$$\hat{Y}(X) = \frac{\sum_{i=1}^{n} Y^i e^{\left(-\frac{D_i^2}{2\sigma^2}\right)}}{\sum_{i=1}^{n} e^{\left(-\frac{D_i^2}{2\sigma^2}\right)}}. \tag{8.3}$$

The value of σ is estimated by trial and error. If the estimate for σ is very large, then the dependent value $\hat{Y}$ will be the mean of all the training data (mean of observed Y_i). As σ goes to zero the estimated dependent value $\hat{Y}$ takes the value of the closest Y_i. We choose σ to be 0.4.

8.3.2 GRNN Training and Gaze Calibration

The input data vector provided to the gaze calibration is

$$X = \begin{pmatrix} \Delta r_{gd} \\ \Delta c_{gd} \\ \Delta r_{gr} \\ \Delta c_{gr} \\ \Delta r_{gp} \\ \Delta c_{gp} \\ R_{pupils} \\ \theta \\ r \\ c \end{pmatrix}, \tag{8.4}$$

where Δr_{gd} and Δc_{gd} are the relative positions of the glint due to the driver side mirror, Δr_{gr} and Δc_{gr} are the relative positions of the glint due to the rear view mirror, Δr_{gp} and Δc_{gp} are the relative positions of the glint due to the passenger side mirror, R_{pupils} gives the ratio of the radii of the two eyes (a measure of head pose), r is

the row value of the pupil center, and c is the column value of the pupil center. Thus the input vector incorporates the relations between the glints and face orientation.

Gaze Mapping

When input vector X is fed into the trained GRNN, it returns the region Y where the gaze is currently fixated. The regions are identified by number as shown in Table 8.3 so

$$Y \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}. \tag{8.5}$$

The relative positions of the glints with respect to the pupil center and the other glints are more important than the actual locations of glints in the image frame. In order to adapt the GRNN for each user, the gaze mapping algorithm retrains itself using correctly mapped X_{input} vectors. This retraining is performed once for each ROI.

Initial GRNN Training

From Figure 8.4 it can be seen that most of the training vectors X_i and Y_i are sampled in the gaze regions that are not ROIs. Furthermore most of the vectors represent gaze points that are in the relative center of these regions. By training heavily on the non-ROI regions false positives are reduced. In addition, because the system retrains itself based on the first few positive ROI classifications, the system maximizes positive classification on the ROI.

During training, the mean of each of the twelve regions is found using the K-means algorithm that is the basis of the KNN classifier. The K-means algorithm provides the twelve sample vectors X_i corresponding to gaze region Y_i. Each one of these twelve sample vectors corresponds to a hidden node in the GRNN.

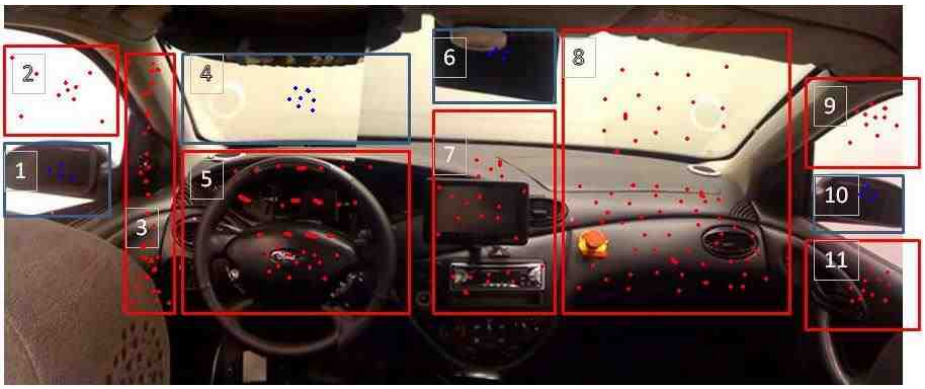


Fig. 8.4. Sample gaze points taken for training

8.4 Results

Overview of Results

Tests were performed on four ROIs. The first test was to determine how well a sample of gaze points (8 for each ROI) could be mapped to its corresponding ROI. The results are shown in Table 8.4.

The strong positive results are due to the fact that there is an IR light at three of the four ROIs. Hence the relative positions of the glints along with face pose can show a strong correlation to that ROI. Most of the training was done using gaze points from other regions. As a result the distances to these points were high. On top of that the first point from each corresponding ROI was used to retrain the GRNN as mentioned above.

This technique has a margin of error of about 2 inches from the boundaries. Eight gaze points are taken that are within 2 inches of the boundary of each ROI. Figure 8.4 shows the results. It is important to notice at the ROI with the highest error is the front view. This is due to two reasons, the first being that the area is large, as a result more likely to produce errors. The second reason is that it does not have a distinct IR light to create a strong glint feature set as do the other ROIs, the ROI requires all the features to have a relative set of glint reflections.

Table 8.2
Mapping Gaze Points to their corresponding ROI

ROI	Correctly classified	Misclassified
ROI1 - Driver Side Mirror	8	0
ROI2 - Front View	8	0
ROI3 - Rear view Mirror	7	1
ROI4 - Passenger Side Mirror	6	2

Due to the position of the IR LEDs, the error margins on the boundaries of each of the ROIs are not uniform. Boundary regions that are closer to the IR LEDs have a significantly smaller error margin. In order to show this, two gaze points were taken from each of the boundaries of the ROIs (hence 8 points). Due to the position of the LEDs in each of the ROIs, gaze points above the LEDs are correctly classified. Gaze points that are on the far side of each mirrors could not be mapped (eyes were out of range due to extreme facial rotation) so they are not classified. As for the other points they have a high level of misclassification.

Table 8.3
Gaze Points within 2 inches from corresponding ROI boundaries

ROI	Correctly classified	misclassified
ROI 1 - Driver Side Mirror	3	5
ROI 2 - Front View	1	7
ROI 3 - Rear view Mirror	5	3
ROI 4 - Passenger Side Mirror	3	5

The final set of tests was a series of gaze points that were greater than 2 inches from a particular ROI. About 16 points were used for each ROI. The objective of the test was to see how many of those points would be misclassified as being in the ROI.

Table 8.4
Gaze points more than 2 inches from the corresponding ROI boundary

ROI	correctly classified	misclassified
ROI 1 - Driver Side Mirror	13	3
ROI 2 - Front View	15	1
ROI 3 - Rear view Mirror	11	5
ROI 4 - Passenger Side Mirror	12	4

8.5 Discussion

The GRNN based eye gaze mapping system provides several benefits but has certain limitations.

Benefits of current solution

Our solution maps to the correct ROI with a high degree of accuracy, uses a monocular system, and is robust to head movements.

The system can also be used as a calibration procedure for other eye tracking systems. Every time a new point in a ROI is provided, it can be used to recalculate the mapping matrix M used in Equation 8.1.

Limitations of current solution

One of the primary limitations of the current system is that there is a relatively large uncertainty, especially when the driver is looking at the boundaries between the regions. There are numerous ways in which this uncertainty can be reduced, for example by improving the camera system to capture images more quickly and clearly. Also the LEDs can be placed in a manner that optimizes their radiance on the driver face region, helping create more precise glints. Of course using a higher number of

training samples would also improve performance. These additional samples should ideally be chosen from boundary regions.

8.6 Summary

Four regions of interest were identified and LEDs placed in each of these regions. Based on their placement and facial rotation a GRNN was used to map the gaze into each of those regions. Tests indicated a strong positive match.

9. SUMMARY

Every aspect of developing a computer vision based gaze mapping system was explored, including camera placement in vehicles, image processing, eye detection, eye tracking, feature extraction and gaze mapping. Eye-gaze mapping systems have not been used in vehicles primarily because they need calibration which is time-consuming and difficult. In this thesis we present an eye-gaze mapping system suitable for use in a DSM system. Contributions include the insight that it is more important to map eye-gaze to regions in the forward view rather than to specific points and the development of a GRNN based system that allows eye-gaze to be mapped to twelve such regions.

The image acquisition system consisted of a monocular camera with multiple light sources. The mapping system could be applied using multiple cameras, but at additional cost. The camera system created bright pupils that allowed eye regions to be easily extracted with simple image processing. The IR light sources created glints used for gaze mapping.

A SVM classification technique, based on a hybrid Kalman/mean-shift tracker, was used to track eye images from frame to frame for initial eye detection. A feature extractor extracts the glints and other eye information to be used for gaze mapping.

The gaze mapping function uses a GRNN to map the input eye feature information into one of twelve regions identified as part of the research process. This system retrains itself for the first few successfully mapped gaze points, thus it optimizes itself for the current user. The system can also be used as a calibration procedure for a 3D mapping function.

10. RECOMMENDATIONS

The system can be improved in many ways. Several simple hardware upgrades would increase the efficiency of the system, especially with respect to feature extraction. To improve eye detection and tracking, better training and system noise modeling could be incorporated. These would also improve the GRNN system used in gaze mapping. Finally the system as a whole can be improved by incorporating standard face pose modeling techniques.

10.1 Hardware Recommendations

A good hardware system can reduce noise during image processing and improve efficiency. The camera system used directly affects the quality of the image frames. In addition, stereo camera systems can be used making eye gaze and face pose mapping computationally easier. Apart from the camera systems, the LEDs used can also be upgraded, using better quality LEDs will produce sharper glint reflections, improving gaze mapping.

10.1.1 Camera System

We used an off-the-shelf Microsoft View-cam. This model has limited capabilities. A camera, with a higher frame count, more exposure and better zoom lens capabilities, would improve performance.

The second recommendation would be to use a stereo imaging system. With a 3D system, the face pose and gaze mapping become much easier. Numerous such systems are available and a lot of work has been done on stereo camera calibration **??**.

10.1.2 LEDs and Embedded Circuit

In this thesis the LEDs were hand fashioned and the serial communication system was not optimized for the frame speed of 30 Fps. The LED circuits should be designed to optimize their illumination of the facial region to produce the best quality glint images.

10.2 Eye Detection and Tracking

Improving eye detection and eye tracking will result in better gaze mapping by providing more reliable eye region images. Also improving eye tracking also reduce overall computational time. Improving gaze mapping will reduce the number of false positives making the system more efficient.

10.2.1 Eye Detection

The SVM-based system for eye detection needs a much higher number of training subjects. Having three separate classifiers, one for frontal view and two for side views would also reduce the rejection rates. The only drawback to adding these classifiers would be the tripling of the run time; but the SVM is a fast algorithm so this would not be prohibitive.

10.2.2 Eye Tracking

Dynamic modeling of system noise would make the tracking algorithm more efficient. Also a higher sampling rate would make the hybrid Kalman/Mean-shift algorithm more efficient. Even doubling the rate would be expected to yield noticeable improvement.

10.3 Gaze Mapping

The GRNN based system can also be improved significantly if boundary areas are reclassified as new regions and if more training samples are used.

LIST OF REFERENCES

LIST OF REFERENCES

[1] D. F. Dinges, M. M. Mallis, G. Maislin, and J. W. Powell, "Final report: Evaluation of techniques for ocular measurement as an index of fatigue and as the basis for alertness management." Report for National Highway Traffic Safety Administration, 2006.

[2] R. I. Hammoud, G. Witt, R. Dufour, A. Wilhelm, and T. Newman, "On driver eye closure recognition for commercial vehicles," *SAE International Journal of Commercial Vehicles*, vol. 1, no. 1, pp. 454–463.

[3] C. Cudalbu, B. Anastasiu, R. Radu, R. Cruceanu, E. Schmidt, and E. Barth, "Driver monitoring with a single high-speed camera and IR illumination," in *International Symposium on Signals, Circuits and Systems, 2005*, vol. 1, pp. 219 – 222, 2005.

[4] Q. Ji and X. Yang, "Real-time eye, gaze, and face pose tracking for monitoring driver vigilance," *Real-Time Imaging*, vol. 8, no. 5, pp. 357–377, 2002.

[5] A. Doshi and M. Trivedi, "On the roles of eye gaze and head dynamics in predicting driver's intent to change lanes," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 10, no. 3, pp. 453 –462, 2009.

[6] U. Bueker, R. Schmidt, and S. Wiesner, "Camera-based driver monitoring for increased safety and convenience," *SAE Technical Paper Series*, 2007.

[7] Z. Zhu and Q. Ji, "Novel eye gaze tracking techniques under natural head movement," *Biomedical Engineering, IEEE Transactions on*, vol. 54, no. 12, pp. 2246 –2260, 2007.

[8] C. Morimoto, D. Koons, A. Amir, and M. Flickner, "Pupil detection and tracking using multiple light sources," *Image and Vision Computing*, vol. 18, pp. 331 –335.

[9] K. Nguyen, C. Wagner, D. Koons, and M. Flickner, "Differences in the infrared bright pupil response of human eyes," in *ETRA '02: Proceedings of the 2002 symposium on Eye tracking research & applications*, (New York, NY, USA), pp. 133–138, ACM, 2002.

[10] S. Goñi, J. Echeto, A. Villanueva, and R. Cabeza, "Robust algorithm for pupil-glint vector detection in a video-oculography eyetracking system," in *Proceedings of the 17th International Conference on Pattern Recognition*, vol. 4, pp. 941–944, IEEE Computer Society, 2004.

[11] J.-G. Wang, E. Sung, and R. Venkateswarlu, "Eye gaze estimation from a single image of one eye," *IEEE International Conference on Computer Vision*, pp. 136 – 143, 2003.

[12] Z. Zhu and Q. Ji, "Novel eye gaze tracking techniques under natural head movement," *Biomedical Engineering, IEEE Transactions on*, vol. 54, no. 12, pp. 2246 –2260, 2007.

[13] A. L. Yuille, P. W. Hallinan, and D. S. Cohen, "Feature extraction from faces using deformable templates," *International Journal of Computer Vision*, vol. 8, no. 2, pp. 99–111, 1992.

[14] X. Xie, R. Sudhakar, and H. Zhuang, "On improving eye feature extraction using deformable templates," *Pattern Recognition*, vol. 27, no. 6, pp. 791 – 799, 1994.

[15] K.-M. Lam and H. Yan, "An improved method for locating and extracting the eye in human face images," in *Proceedings of the 13th IEEE International Conference on Pattern Recognition*, vol. 3 of *ICPR 96*, pp. 411 – 415, 1996.

[16] G. C. Feng and P. C. Yuen, "Multi-cues eye detection on gray intensity image," *Pattern Recognition*, vol. 34, no. 5, pp. 1033 – 1046, 2001.

[17] M. Reinders, R. W. C. Koch, and J. Gerbrands, "Locating facial features in image sequences using neural networks," in *In 2nd International Conference on Automatic Face and Gesture Recognition*, pp. 230–235, 1997.

[18] J. Huang and H. Wechsler, "Eye detection using optimal wavelet packets and radial basis functions," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 13, no. 7, pp. 1009–1026, 1999.

[19] 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*, vol. 98, no. 1, pp. 124–154, 2005.

[20] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995.

[21] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. O'Reilly, 2008.

[22] X. Xie, R. Sudhakar, and H. Zhuang, "Real-time eye feature tracking from a video image sequence using Kalman filter," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 25, no. 12, pp. 1568 –1577, 1995.

[23] A. Blake, R. Curwen, and A. Zisserman, "A framework for spatiotemporal control in the tracking of visual contours," *Int. J. Comput. Vision*, vol. 11, no. 2, pp. 127–145, 1993.

[24] K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *Information Theory, IEEE Transactions on*, vol. 21, no. 1, pp. 32–40, 1975.

[25] T. Horprasert, Y. Yacoob, and L. S. Davis, "Computing 3-D head orientation from a monocular image sequence," in *Proceedings of the 2nd International Conference on Automatic Face and Gesture Recognition*, pp. 242–247, 1996.

[26] R. Rae and H. Ritter, "Recognition of human head orientation based on artificial neural networks," *Neural Networks, IEEE Transactions on*, vol. 9, no. 2, pp. 257–265, 1998.

[27] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, pp. 511–518, 2001.

[28] A. Poole and L. J. Ball, "Eye tracking in human-computer interaction and usability research: Current status and future prospects," in *Encyclopedia of Human-Computer Interaction. Pennsylvania: Idea Group, Inc*, 2005.

[29] Y. Zhou, X. Zhao, S. Zhang, and Y. Zhang, "Design of eye tracking system for real scene," *IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, vol. 1, pp. 708–711, 2008.

[30] F. Coutinho and C. Morimoto, "Free head motion eye gaze tracking using a single camera and multiple light sources," in *19th Brazilian Symposium on Computer Graphics and Image Processing, 2006*, pp. 171 –178, 2006.

[31] S. W. Shih and J. Liu, "A novel approach to 3-D gaze tracking using stereo cameras," *IEEE Transactions on Syst. Man and Cybern., part B*, vol. 34, pp. 234–245, 2004.

[32] D. Beymer and M. Flickner, "Eye gaze tracking using an active stereo head," in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, vol. 2, pp. 451–458, 2003.

[33] J. Batista, "Locating facial features using an anthropometric face model for determining the gaze of faces in image sequences," in *Image Analysis and Recognition*, vol. 4633 of *Lecture Notes in Computer Science*, pp. 839–853, Springer Berlin / Heidelberg, 2007.

[34] A. C. Guyton and J. E. Hall, *Textbook of Medical Physiology*. Elsevier Saunders, 2006.

[35] C. Oyster, *The Human Eye: Structure and Function*. MA: Sinauer Associates, Inc., 1999.

[36] J. X. Dong, C. Y. Suen, and A. Krzyzak, "A fast SVM training algorithm," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 17, no. 3, pp. 367–384, 2003.

[37] V. Athitsos, J. Alon, and S. Sclaroff, "Efficient nearest neighbor classification using a cascade of approximate similarity measures," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 486–493, 2005.

[38] G. Bradski and A. Kaehler, *Learning OpenCV*. O'Reilly, 2008.

[39] Q. Ji and Z. Zhu, "Eye and gaze tracking for interactive graphic display," *Machine Vision and Applications*, vol. 15, no. 3, pp. 139–148, 2004.

[40] D. Specht, "A general regression neural network," *IEEE Transactions on Neural Networks*, vol. 2, no. 6, pp. 568–576, 1991.