Gaze Estimation Based on Multi-view Geometric Neural Networks

Devarth Parikh
dp5116@rit.edu

Follow this and additional works at: https://scholarworks.rit.edu/theses

Recommended Citation
Parikh, Devarth, "Gaze Estimation Based on Multi-view Geometric Neural Networks" (2020). Thesis. Rochester Institute of Technology. Accessed from

This Thesis is brought to you for free and open access by RIT Scholar Works. It has been accepted for inclusion in Theses by an authorized administrator of RIT Scholar Works. For more information, please contact ritscholarworks@rit.edu.
Gaze Estimation Based on Multi-view Geometric Neural Networks

by
Devarth Parikh

A Thesis Submitted
in
Partial Fulfillment
of the
Requirements for the Degree of
MASTER OF SCIENCE
in
Imaging Science

Approved by:

Professor
Dr. Guoyu Lu (Thesis Advisor)

Professor
Dr. Jeff Pelz (Thesis Committee Member)

Professor
Dr. Carl Salvaggio (Thesis Committee Member)

Chester. F. Carlson Center for Imaging Science

COLLEGE OF SCIENCE
ROCHESTER INSTITUTE OF TECHNOLOGY
ROCHESTER, NEW YORK
JULY 2020
Gaze Estimation Based on Multi-view Geometric Neural Networks

by

Devarth Parikh

I, Devarth K. Parikh, hereby grant permission to the Wallace Library, of Rochester Institute of Technology, to produce this document in whole or in part, with the knowledge that any reproduction will not be for commercial use or profit.

Devarth Kaushik Parikh

07/27/2020

Devarth K. Parikh

Date
I dedicate this master's thesis to my family—Kaushik, Shilpa, Samarth, Pravin, Minaxi, Urmil and Miti who have been a constant source of motivation and support to me.
Acknowledgement

Firstly, I’d like to express my sincere gratitude to Dr. Guoyu Lu for giving me the opportunity to work with the Intelligent Vision and Sensing (IVS) group at Chester F. Carlson College of Imaging Science. Dr. Lu has always backed me and motivated me at every step of my master’s program. His guidance throughout my research journey has helped me shaped the way I approach and tackle problems. Additionally, I have also learnt time management and interpersonal skills from working closely with him. I have always admired his ability to work diligently with a never give up attitude that will keep me inspired throughout my career. Working with the group I have had several opportunities to work with corporations from the industry on several interesting projects and interact with multiple people. I have taken several courses with Dr. Lu that has helped me develop a strong foundation of the subject which will be the founding platform to my career. It has truly been a privilege working with him.

I would like to thank Dr. Jeff Pelz and Dr. Carl Salvaggio, my thesis committee members, for their guidance and suggestions through the course of my master’s thesis. I am thankful to Dr. Pelz for giving me the opportunity to work in his lab in Spring 2019. I learnt a lot about eye tracking working in his lab, which has been pivotal to my thesis.

I take this opportunity to thank all Imaging Science professors and staff members for all the knowledge I gained by taking Imaging Science courses. I want to thank all of them for their constant support and treating me like a part of the family. This includes, but not limited to, Dr. Charles Bachmann, Dr. Roger Easton, Dr. Jan Van Aardt, Dr. Zoran Ninkov, Dr. David Messinger.

I am thankful to my lab-mates, Yawan and Yuxing for the countless discussions we have had when we got stuck on problems. You made the lab feel like my second home. I would also like to thank my classmates of the Imaging Science department that made my master’s journey fun and engaging.

I would like to thank my parents and my brother for being my pillar of strength who have always supported me and have been a constant source of motivation.
Abstract

Gaze and head pose estimation can play essential roles in various applications, such as human attention recognition and behavior analysis. Most of the deep neural network-based gaze estimation techniques use supervised regression techniques where features are extracted from eye images by neural networks and regress 3D gaze vectors. I plan to apply the geometric features of the eyes to determine the gaze vectors of observers relying on the concepts of 3D multiple view geometry. We develop an end-to-end CNN framework for gaze estimation using 3D geometric constraints under semi-supervised and unsupervised settings and compare the results. We explore the mathematics behind the concepts of Homography and Structure-from-Motion and extend it to the gaze estimation problem using the eye region landmarks. We demonstrate the necessity of the application of 3D eye region landmarks for implementing the 3D geometry-based algorithms and address the problem when lacking the depth parameters in the gaze estimation datasets. We further explore the use of Convolutional Neural Networks (CNNs) to develop an end-to-end learning-based framework, which takes in sequential eye images to estimate the relative gaze changes of observers. We use a depth network for performing monocular image depth estimation of the eye region landmarks, which are further utilized by the pose network to estimate the relative gaze change using view synthesis constraints of the iris regions. We further explore CNN frameworks to estimate the relative changes in homography matrices between sequential eye images based on the eye region landmarks to estimate the pose of the iris and hence determine the relative change in the gaze of the observer. We compare and analyze the results obtained from mathematical calculations and deep neural network-based methods. We further compare the performance of the proposed CNN scheme with the state-of-the-art regression-based methods for gaze estimation. Future work involves extending the end-to-end pipeline as an unsupervised framework for gaze estimation in the wild.
## Table of Contents

<table>
<thead>
<tr>
<th>CHAPTER NO.</th>
<th>CHAPTER TITLE</th>
<th>PAGE NO.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td></td>
<td>v</td>
</tr>
<tr>
<td>TABLE OF CONTENTS</td>
<td></td>
<td>vi</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td></td>
<td>ix</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td></td>
<td>xi</td>
</tr>
<tr>
<td>CHAPTER-1</td>
<td>INTRODUCTION AND RELATED WORK</td>
<td>1.</td>
</tr>
<tr>
<td>1.1</td>
<td>Introduction</td>
<td>1.</td>
</tr>
<tr>
<td>1.2</td>
<td>Related Work</td>
<td>2.</td>
</tr>
<tr>
<td>1.2.1</td>
<td>Deep Learning Based Gaze Estimation</td>
<td>2.</td>
</tr>
<tr>
<td>1.2.2</td>
<td>Deep Learning Based Homography</td>
<td>3.</td>
</tr>
<tr>
<td>1.2.3</td>
<td>Single Image Depth Estimation</td>
<td>4.</td>
</tr>
<tr>
<td>1.3</td>
<td>Per Chapter Synopsis</td>
<td>5.</td>
</tr>
<tr>
<td>CHAPTER-2</td>
<td>DATASETS AND DATA-PREPROCESSING</td>
<td>7.</td>
</tr>
<tr>
<td>2.1</td>
<td>Need for Synthetic Dataset</td>
<td>7.</td>
</tr>
<tr>
<td>2.1.1</td>
<td>U2Eyes Dataset</td>
<td>7.</td>
</tr>
<tr>
<td>2.1.2</td>
<td>UnityEyes Dataset</td>
<td>8.</td>
</tr>
<tr>
<td>2.1.3</td>
<td>MPIIGaze Dataset</td>
<td>9.</td>
</tr>
<tr>
<td>2.1.4</td>
<td>GI4E Dataset</td>
<td>10.</td>
</tr>
<tr>
<td>2.2</td>
<td>Data-Preprocessing for Eye Region</td>
<td>11.</td>
</tr>
<tr>
<td>2.2.1</td>
<td>Cropping the Eye Region</td>
<td>11.</td>
</tr>
<tr>
<td>2.2.2</td>
<td>Effect of Cropping on Intrinsic Parameters</td>
<td>12.</td>
</tr>
<tr>
<td>2.2.3</td>
<td>Normalizing the Eye Images</td>
<td>13.</td>
</tr>
<tr>
<td>2.2.4</td>
<td>Plotting the Gaze Vector in the U2Eyes Dataset</td>
<td>14.</td>
</tr>
<tr>
<td>CHAPTER-3</td>
<td>CONCEPTS OF MULTIVIEW-GEOMETRY</td>
<td>16.</td>
</tr>
<tr>
<td>3.1</td>
<td>Homography Matrix</td>
<td>16.</td>
</tr>
<tr>
<td>3.2</td>
<td>Using Co-Planar points for estimating the pose of the camera</td>
<td>18.</td>
</tr>
<tr>
<td>3.3</td>
<td>Decomposing homography matrix into 3D rotation</td>
<td>19.</td>
</tr>
<tr>
<td>3.4</td>
<td>Perspective-n-Point Problem</td>
<td>22.</td>
</tr>
<tr>
<td>3.5</td>
<td>Using the Perspective-n-Point problem to determine the gaze of the Observer</td>
<td>25.</td>
</tr>
<tr>
<td>CHAPTER-4</td>
<td>SUPERVISED DEEP LEARNING BASED GAZE ESTIMATION</td>
<td>27.</td>
</tr>
</tbody>
</table>
4.1 Predicting Eye Region Landmarks Using Deep Neural Networks 27.

4.2 Deep Learning Based Supervised Eye Region Landmark Prediction 27.
  4.2.1 Network Architecture for Predicting Eye Region Landmarks 28.
  4.2.2 Training Details 28.
  4.2.3 Quantitative Analysis of the Network for Predicting eye landmarks 29.

4.3 Supervised Depth Prediction for the Eye Region Landmarks 31.
  4.3.1 Depth Normalization 31.
  4.3.2 Training Details of the Depth Network 32.
  4.3.3 Performance of the Depth Estimation on the U2Eyes Dataset 34.
  4.3.4 Performance of the Depth Estimation on the MPIIGaze Dataset 35
  4.3.5 Quantitative Evaluation of Depth Network 36.

4.4 Homography Based Gaze Estimation 37.
  4.4.1 Using the Concept of Homography decomposition for Gaze Estimation 37.
  4.4.2 Performance of Homography decomposition on U2Eyes Dataset 38.
  4.4.3 Performance of Homography decomposition on MPIIGaze Dataset 39.
  4.4.4 Quantitative Evaluation for performance of Homography Decomposition method 40.

4.5 Deep Learning Based Homography Prediction 42.
  4.5.1 4-Point Parameterized matrix 42.
  4.5.2 Deep Neural Network for Homography Estimation 42.
  4.5.3 Performance of the Homography Estimation network 45.

CHAPTER-5

UNSUPERVISED GAZE ESTIMATION FROM SFM 48.

5.1 Problem Statement 48.

5.2 Unsupervised Learning Framework for Gaze Estimation 50.
  5.2.1 Projection of 2D iris landmarks into 3D space 51.
  5.2.2 Convolutional Neural Network for 2D Iris landmarks Depth Estimation 52.
  5.2.3 Pose Estimation for 2D iris landmarks 53.
  5.2.4 Convolutional Neural Network for relative pose estimation 55.
  5.2.5 Framework for Unsupervised Gaze Estimation 55.

5.3 Training Details 57.
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.3.1</td>
<td>Loss Functions for unsupervised pose and depth estimation</td>
<td>57.</td>
</tr>
<tr>
<td>5.3.2</td>
<td>Training Details for the Unsupervised Pose and Depth Estimation network</td>
<td>58.</td>
</tr>
<tr>
<td>5.3.3</td>
<td>Performance of the unsupervised pose and depth estimation network</td>
<td>58.</td>
</tr>
<tr>
<td>5.3.4</td>
<td>Visual Performance Analysis of the Unsupervised Pose and Depth network on U2Eyes Dataset</td>
<td>59.</td>
</tr>
<tr>
<td>5.3.5</td>
<td>Visual Performance Analysis of the Unsupervised Pose and Depth network on MPIIGaze Dataset</td>
<td>60.</td>
</tr>
<tr>
<td>5.3.6</td>
<td>Training Details for supervised Depth and unsupervised Pose estimation network</td>
<td>62.</td>
</tr>
<tr>
<td>5.3.7</td>
<td>Performance of the supervised Depth and Unsupervised Pose Estimation Network</td>
<td>62.</td>
</tr>
<tr>
<td>5.3.8</td>
<td>Visual Performance Analysis of the supervised depth and unsupervised pose</td>
<td>63.</td>
</tr>
</tbody>
</table>

**CHAPTER-6**

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1</td>
<td>CONCLUSION AND FUTURE WORK</td>
<td>65.</td>
</tr>
</tbody>
</table>
## List of Figures

<table>
<thead>
<tr>
<th>FIGURE LABEL</th>
<th>PAGE NO.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fig. 2.1: Few samples of the U2Eyes dataset.</td>
<td>8</td>
</tr>
<tr>
<td>Fig. 2.2: 2D Eye region landmarks from a sample in U2Eyes Dataset.</td>
<td>8</td>
</tr>
<tr>
<td>Fig. 2.3: Visualizing the Optical axis and the Visual Axis of the human eye</td>
<td>8</td>
</tr>
<tr>
<td>Fig. 2.4: UnityEyes Dataset</td>
<td>9</td>
</tr>
<tr>
<td>Fig. 2.5: a) Shows the eye image captured from the laptop (left)</td>
<td>10</td>
</tr>
<tr>
<td>Fig. 2.5: b) Normalized eye images (right)</td>
<td>10</td>
</tr>
<tr>
<td>Fig. 2.6: Shows some of the images from the MPIIGaze dataset.</td>
<td>10</td>
</tr>
<tr>
<td>Fig. 2.7: Samples from GI4E dataset.</td>
<td>10</td>
</tr>
<tr>
<td>Fig. 2.8: Raw Images from U2Eyes Dataset</td>
<td>11</td>
</tr>
<tr>
<td>Fig. 2.9: Cropped Images from the U2Eyes Dataset with annotated 2D iris</td>
<td>12</td>
</tr>
<tr>
<td>landmarks.</td>
<td></td>
</tr>
<tr>
<td>Fig. 2.10: Intrinsic Parameters</td>
<td>13</td>
</tr>
<tr>
<td>Fig. 2.11: (a) Raw Images from U2Eyes Dataset</td>
<td>14</td>
</tr>
<tr>
<td>Fig. 2.11: (b): Cropped grayscale eye images</td>
<td>14</td>
</tr>
<tr>
<td>Fig. 2.11: (c): Normalized eye images</td>
<td>14</td>
</tr>
<tr>
<td>Fig. 2.12: Demonstrates the gaze vector computed from the ground truth</td>
<td>15</td>
</tr>
<tr>
<td>3D iris landmark points</td>
<td></td>
</tr>
<tr>
<td>Fig. 3.1: Homographic Relationship between two image planes</td>
<td>16</td>
</tr>
<tr>
<td>Fig. 3.2: Demonstrates geometric relationship between two image planes</td>
<td>19</td>
</tr>
<tr>
<td>Fig. 3.3: The left camera C is first rotated to align orientation with camera C’ and then translated to align the camera center with camera C’.</td>
<td>19</td>
</tr>
<tr>
<td>Fig. 3.4: Demonstrates the rotation between two image planes</td>
<td>20</td>
</tr>
<tr>
<td>Fig. 3.5: Image represents the reprojection error between the image point p and the projected 3D point p'</td>
<td>24</td>
</tr>
<tr>
<td>Fig. 3.6: Projection of 3D Points onto the image plane.</td>
<td>24</td>
</tr>
<tr>
<td>Fig. 3.7: Results from the Solve PnP method</td>
<td>25</td>
</tr>
<tr>
<td>Fig. 4.1: Architecture for eye region landmark detection</td>
<td>28</td>
</tr>
<tr>
<td>Fig. 4.2: Performance of the landmark detection network on U2Eyes dataset</td>
<td>30</td>
</tr>
<tr>
<td>Fig. 4.3: Performance of the landmark detection network on MPIIGaze dataset</td>
<td>30</td>
</tr>
<tr>
<td>Fig. 4.4: Depth Regression network for predicting the depth of the eye region landmarks</td>
<td>31</td>
</tr>
<tr>
<td>Fig. 4.5: Normalized depth for the U2Eyes Dataset</td>
<td>32</td>
</tr>
<tr>
<td>Fig. 4.6: Ground truth depth for the U2Eyes Dataset</td>
<td>32</td>
</tr>
<tr>
<td>Fig. 4.7: Plot for the depth loss</td>
<td>33</td>
</tr>
<tr>
<td>Fig. 4.8: Plot for the heatmap loss</td>
<td>33</td>
</tr>
<tr>
<td>Fig. 4.9: Plot for the landmark loss</td>
<td>33</td>
</tr>
<tr>
<td>Fig. 4.10: Plot for the overall training loss</td>
<td>33</td>
</tr>
<tr>
<td>Fig. 4.11: Performance of depth estimation network on U2Eyes dataset.</td>
<td>34</td>
</tr>
<tr>
<td>Fig. 4.12: Performance of depth estimation network on MPIIGaze dataset</td>
<td>35</td>
</tr>
</tbody>
</table>
Fig. 4.13: Feature Matching between 2D iris landmarks
Fig. 4.14: Performance of the Homography Decomposition on the U2Eyes Dataset
Fig. 4.15: Performance of the Homography Decomposition for MPIIGaze Dataset
Fig. 4.16: Neural architecture for Homography Estimation
Fig. 4.17: The regression module for predicting the parametrized homography matrix
Fig. 4.18: Warping of the reference image onto the target image using the homography obtained from 2D iris feature matches.
Fig. 4.19: Plot describing the training process
Fig. 4.20: Performance of Homography Network
Fig. 5.1: Example of Structure from Motion.
Fig. 5.2: Demonstrating each eye pose having unique ellipse shape
Fig. 5.3: Elliptical 2D projection of a 3D circular feature on the image plane
Fig. 5.4: 2D projection of a 3D circular object on a 2D plane
Fig. 5.5: Projecting 2D image point into 3D space.
Fig. 5.6: Network Architecture for depth estimation.
Fig. 5.7: Relative pose between two iris orientations.
Fig. 5.8: Pose Network Architecture.
Fig. 5.9: Framework for unsupervised Gaze Estimation.
Fig. 5.10: (a) RGB eye Image
Fig. 5.10: (b) Segmented Iris mask
Fig. 5.10: (c) Segmented Eye Region mask
Fig. 5.10: (d) Segmented RGB Iris
Fig. 5.11: Training Loss for Unsupervised Depth and Pose Estimation
Fig. 5.12: Plot of the gaze error for unsupervised depth and pose estimation network for U2Eyes dataset
Fig. 5.13: Performance of the unsupervised depth and unsupervised pose estimation network for gaze estimation on the U2Eyes Dataset
Fig. 5.14: Performance of the unsupervised depth and unsupervised pose estimation network for gaze estimation on the MPIIGaze Dataset
Fig. 5.15: Training Loss for Supervised Depth and Unsupervised Pose estimation
Fig. 5.16: Plot of the gaze error for Supervised Depth and Unsupervised Pose estimation network for U2Eyes dataset
Fig. 5.17: Results for gaze estimation with Supervised Depth and Unsupervised Pose estimation.
List of Tables

<table>
<thead>
<tr>
<th>TABLE TITLE</th>
<th>PAGE NO.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 4.1: Performance of eye region landmark detection.</td>
<td>29</td>
</tr>
<tr>
<td>Table 4.2: Performance of the supervised depth estimation network.</td>
<td>36</td>
</tr>
<tr>
<td>Table 4.3: Performance of the Homography Decomposition method.</td>
<td>40</td>
</tr>
<tr>
<td>Table 5.1: Performance of the Unsupervised Gaze estimation network on U2Eyes Dataset</td>
<td>64</td>
</tr>
</tbody>
</table>
Chapter 1.
Introduction and Related Work.

1.1 Introduction.

Eye tracking is a powerful tool for understanding human behavior and analyze their actions. Eye tracking is an active research domain with its applications in Virtual Reality, tracking attention of automobile drivers, understanding consumer behavior etc. Most of the gaze estimation research is carried out using eye trackers to track the eye movements using local features of the eye. Eye trackers have cameras close to the eye, which enables them to capture high resolution eye images for accurate detection and tracking of the eye features which correlates to higher accuracy in gaze tracking. However, accurate gaze tracking from mobile devices such as mobile phones, laptop and webcam is difficult because of the low-resolution eye images with uncontrolled illumination and occlusion. With advancements in deep learning, increased accuracies in gaze estimation has been observed using convolutional neural networks. In Section 1.2.1 we discuss most state-of-the-art deep learning based, supervised gaze estimation methods. However, we see that most of the state-of-the-art algorithms for gaze estimation using deep learning use supervised learning for regressing the gaze angle from eye images. Firstly, it is difficult to collect quality ground truth data of eye images with accurate gaze of the observers carrying out their everyday activities in the wild conditions outside laboratory settings. It is still not clear about the eye features that the network learns while regressing for the 3D gaze vector. These supervised networks usually perform with high accuracies on the training dataset but are not robust on cross datasets and the accuracy drops. In our method, instead of completely relying on unknown features we want to leverage the eye region landmarks as features and concepts of 3D multiview geometry for gaze estimation. We demonstrate the importance of depth information of the iris landmarks to implement the multiview geometry algorithms. Later, we explore the use of neural networks to estimate the depth of iris landmarks using supervised learning algorithms. Further, we explore the use of eye region landmarks as features to develop an unsupervised learning architecture to estimate the 3D gaze of
the observer. We also look into the traditional concepts of multiview geometry such as homography and pose estimation for determining the gaze of the observer. Later we compare the performance of traditional and neural network-based methods.

1.2 Related Work.

1.2.1 Deep Learning based Gaze Estimation.

With the advancement in neural networks, we have observed an improved performance in gaze estimation. As we see in this section, most of the recent state of the art algorithms for gaze tracking still rely heavily on ground truth information of 2D gaze vector to use supervised learning algorithms for training the gaze estimation networks. These supervised algorithms are known to perform well on the training dataset or subject, but their performance drops significantly on different data. Zhang et. al [1] proposed the first appearance-based gaze estimation technique using neural networks based on supervised learning. They developed the MPIIGaze dataset for estimating the gaze of the observer in the wild settings. The dataset consists of 213,659 images from 15 participants with varying illumination, head pose, gaze direction and appearance, which makes the gaze estimation on this dataset much more challenging compared to other datasets. They were among the first to propose a deep learning-based gaze estimation method utilizing the head pose and eye images. They used a VGG-Net based architecture with 16 convolutional layers, 2 fully connected layers and a classification layer to regress a 2D gaze vector, which constitutes of two angles, the pitch and yaw. Head pose information was added to the network to demonstrate improved performance. It has been claimed that the performance of the gaze estimation can be improved with the use of full-face images. Zhang et. al [6] use full face images with the AlexNet architecture and spatial weights to regress the 2d and 3d gaze vectors of the observer.

Attempts have been made to use intermediate image features using CNNs rather than directly using regression for gaze vector. Park et. al [2] proposed a method to estimate the eye region landmarks in the wild settings. They use UnityEyes, a synthetic eye image dataset [3] for training the stacked hourglass network to localize the eye region landmarks using the ground truth labels. They further use the predicted landmark region and model fitting methods to determine the 2D gaze angles,
which are pitch and yaw. Alternatively, Park et. al [4] proposed a method where they break down the gaze estimation method into intermediate supervision step. They prove that it improves the performance of the gaze estimation. Researchers have explored the impact of using different modalities in combination with the Convolutional Neural Networks (CNNs) to determine the gaze of the observer. Palmero et. al [5] leverage the time domain information for estimating the gaze of an observer. They use the EYEDIAP [8] dataset for training and include temporal information in the form of full-face image, eye image and facial landmarks to determine the 2D gaze vector. The network consists of three stages, with the first stage named ‘ Individual’ which learns the appearance of the observer using the normalized full-face image and normalized eye images. Later, in the “Fusion” stage the appearance-based features are combined with the normalized facial landmark. The last is the “Temporal” stage where Recurrent Neural Networks are used to determine the 2D gaze vector of the observer. They prove that they improve the accuracy using geometric features and temporal information along with the appearance-based features. Lian et. al [7] explored the use of depth modality along with RGB images with Convolutional Neural Networks for head pose estimation and eyeball pose. However, depth images have lot of noise and holes, which needs to be dealt with for robust gaze estimation. They propose a multitask network which includes a GAN based network for refining the depth and a CNN based network to predict the gaze, where both the networks are trained jointly. They show that the refinement of the depth map results in a highly robust gaze estimation. They train the network on EYEDIAP [8] dataset; however, they also generate their own dataset since the number of images in the EYEDIAP dataset was not enough for training the networks.

1.2.2 Deep Learning based Homography.

Homography Estimation using traditional computer vision algorithms depends heavily on the features / key point detection. However, if the feature matches are not accurate, the homography estimation may be inaccurate. There have been several attempts to use Convolutional Neural Networks to determine the homography with supervised and unsupervised learning techniques. Daniel DeTone. et. al. [27] solve the problem of determining the homography matrix using an end to end Convolutional Neural Network (CNN). They present a two-fold network-based solution, Regression based, and Classification based. A feed-forward network takes in two grayscale images
stacked upon each other as input, with 10 layers to predict the Homography matrix (H) with 8 DOF. Homography matrix encodes both the rotation and translation terms, which is difficult to predict directly, and so they use the 4- point parameterized matrix as labels and reduce the L2 loss between the predicted parameterized matrix and the ground truth labels. Nguyen et. al [29] present an unsupervised deep learning-based approach for estimating the Homography between the two images. They claim that the approach achieves state of the art performance over traditional methods under certain conditions. They compare the performance of both the supervised and unsupervised approaches. They use the photometric error as an unsupervised loss constraint.

1.2.3 Single Image Depth Estimation.

Some of the initial work in supervised single image depth estimation using neural networks was done by Eigen. Et. al [9]. They developed two deep networks each for global and local depth estimation. They first train the global depth estimation network followed by the local depth estimation network for refining the predicted depth locally, utilizing the ground truth depth as supervision using the L2 loss. Iro Laina and Christian Rupprecht. et. al [10] use the concept of residual learning using the convolutional neural networks to map the function between the depth and the RGB images. Resolution of the predicted depth map is improved by using the proposed reverse Huber loss, with the network independent of post processing.

Because of the limitations of the labeled data, researchers have focused on unsupervised techniques for single image depth estimation, depending on the concepts of multiview geometry for constraining the unsupervised loss functions. Zhou et. al [11] use the concepts of Structure from Motion to develop unsupervised loss constraints for estimating the depth and also the pose between two consecutive frames. They train a depth network and a pose network simultaneously based on a combined loss function. However, while testing, depth and pose network can be used independently. Clément et. al [12] proposed novel method for training unsupervised depth estimation network. They claim that the photometric reconstruction loss was helpful to predict a low-quality depth map, which was further improved by leveraging the consistency between the disparity maps of both the left and the right images along with the smoothness loss to avoid
discontinuities. The authors claim their network can perform better than the supervised learning methods trained on ground truth depth values under certain conditions.

1.3 Per Chapter Synopsis.

In Chapter 1, first we introduce our work on gaze estimation using neural networks and concepts of multiview geometry. We discuss the motivation for the work and the need for developing an unsupervised learning algorithm with eye landmarks for estimating the gaze of an observer, captured from remote devices. We also introduce the related work for gaze estimation using supervised neural networks, homography based gaze estimation and single image depth estimation using neural networks.

In Chapter 2, we explore the details of the various synthetic and real eye image datasets used throughout the work. We discuss the essential and unique features of each dataset along with its pros and cons. We further discuss the details of determining the normalized eye images from the raw image datasets essential for training the neural networks. We also discuss the effects of normalization on the intrinsic parameters of the camera.

In Chapter 3, we introduce the concepts of 3D multiview geometry and develop a mathematical understanding of the traditional concepts. We discuss the concept of camera projection matrix, homography, pose estimation using perspective-n-point problem and epipolar geometry. We further discuss in detail, how the geometry can be leveraged specifically for gaze estimation of the observer. Later, we demonstrate the importance of the depth information of the eye region landmarks to implement the above discussed concepts.

In Chapter 4, we first introduce some existing work on estimating the eye region landmarks using CNN’s [30]. We then extend this work to estimate the depth of the iris landmarks using neural networks. We want to explore the performance of the network trained on synthetic images, to determine the depth of the eye region landmarks on the real eye images in the wild settings. We later explore the concept of homography decomposition for estimating the relative gaze for both
real eye images and synthetic eye images. Lastly, we explore the concept of homography estimation using convolutional neural networks and demonstrate its performance for estimating the gaze of the observer.

In Chapter 5, we introduce and explore a geometry-based method for gaze estimation using the concepts of Structure from Motion and deep learning. Inspired from the work of Zhou et. al [33], we explore an unsupervised learning framework for simultaneously estimating the depth of the iris landmarks and the relative pose between the 3D iris landmarks for consecutive eye images to determine the relative gaze. We discuss the unsupervised loss functions used to train the network. Later we evaluate the performance of the network on synthetic eye images as well as real eye images. Lastly, we discuss the results and drawbacks of the method for gaze estimation.

In Chapter 6, we summarize and discuss the results of the experiments throughout the work. We further discuss the potential work for the future that can be explored to improve upon the current performance.
Chapter 2.
Datasets and Data Pre-Processing.

2.1 Need for Synthetic Dataset.

Most of the publicly available datasets for gaze estimation provide with manually annotated eye region landmarks, however for in the wild settings it is very difficult to obtain ground truth data. For using the concepts of multiview geometry, we require the 3D eye region landmarks. However, the depth information is missing in most of the real-world dataset. We also further explore the idea of CNN’s to see if we can improve the performance of the traditional multiview geometry techniques. This requires large dataset with annotated accurate ground truth eye region landmarks, which allows us to explore the idea further. We also want to explore the performance of the neural networks to predict the gaze of the observer on real images, when trained on synthetic dataset [20].

2.1.1 U2Eyes Dataset.

U2Eyes dataset has been brought to us by Porta et. al [13]. This dataset consists of synthetic images created using UnityEyes for 1000 users consisting of 5875 images and 503 files. The images have a resolution of 3820x2160 pixels and were captured by a camera with a focal length of 3.67 mm. This is the only dataset that provides with the focal length of the camera, which is essential for using the concepts of multiview geometry for gaze estimation. The dataset consists of 3D head pose and gaze directions for each user. The head pose information consists of its orientations in terms of roll, yaw and pitch and head center in terms of cartesian coordinates. They also consist of 3D and 2D landmarks of the pupil, iris, eyelids, caruncle and eye corners. The dataset consists of 20 skin colors and 5 eye textures distributed uniformly throughout the dataset. Each observer with a specific head pose orientation is asked to look onto the grid points as shown in the Fig. 2.1 below.
Fig. 2.3 represents the 3D model of the eye. The 3D model of the eye rotates around the eyeball center. The visual axis is shown as the red vector that passes through the eyeball center. The angle \( \kappa \) represents the horizontal and vertical offset of the visual axis from the optical axis by 3-7° and 2-3° respectively [13]. The optical axis is represented by the vector \((0, 0, 1)\) in green. It passes through the pupil center and eyeball center. Throughout the work, we consider the optical axis as the gaze vector as the angle \( \kappa \) is fixed for an observer and can be used to compute the visual axis.

2.1.2 UnityEyes Dataset.

UnityEyes dataset has been developed by Wood et. al [14]. UnityEyes dataset is a synthetic dataset for gaze estimation in the wild settings, where the pupil suffers occlusion for large gaze angles. They provide the 2D/3D ground truth landmarks for caruncle, iris and interior eye region. They also provide with the 3D gaze vector and head pose vector in camera-space. They allow the user to set the range of gaze orientation
of the dataset by setting the Yaw, Pitch and Roll parameters of the eyeball. They also allow the user to control the rotation of the camera facing the eye. We weren’t able to find the camera intrinsic matrix for this dataset, which made it difficult to use it for multiview geometry applications.

2.1.3 MPIIGaze Dataset.

The MPIIGaze dataset is developed by Zhang et. al. [15] which consists of 213,659 images collected from 15 participants. The dataset was collected over a time period of 10 months. 20 on-screen marks are randomly positioned on the screen for the subjects to observe, under everyday settings. The concentration of the subject was verified as they were asked to press the spacebar as they fixate on the marker on the screen. The dataset has varying appearance and illumination properties. The dataset consists of 3D left and right eye centers, 3D gaze target, 4 eye corner landmarks and 2 mouth landmarks, head pose orientation from 6 facial landmarks. They also present the camera intrinsic parameters which consists of the focal length and distortion coefficients (radial and tangential distortion) along with extrinsic parameters (rotation & translation) along with the laptop screen dimensions in pixels, which was used to collect the dataset. Along with the raw images the authors also provide normalized eye images in which they use perspective transformation which counteracts the effects of rotation and scaling. The images have been captured under everyday settings and hence this in the wild dataset is more challenging for gaze estimation processes.
The GI4E dataset is developed by Villanueva et. al [16], which consists of 1339 images, 12 each from 103 subjects. The dataset was collected using a webcam with a resolution of 800x600 pixels. The dataset also provides 2D iris and eye corner landmark points in pixels. The dataset is helpful for evaluation as they have a large number of participants with high resolution images, which is helpful in evaluating the performance of the algorithms.
2.2 Data-Preprocessing for the eye region.

The datasets such as MPIIGaze, U2Eyes and UnityEyes usually provide a high-resolution image with the intrinsic parameters, which consists of an entire scene along with the eyes. However, we are only interested in the eye region for the analysis and not the rest of the scene as it adds to computational time and resources. Thus, we want to crop out the extra information from the image and consider only the eye region for further analysis to estimate the gaze of the observer.

2.2.1 Cropping the Eye Region.

Since we are only interested in the eye region, we need to crop the image. However, cropping the image, results in the change in intrinsic parameters (K) of the camera, which are very important for implementing the concepts of multiview geometry algorithms. The gaze results are very sensitive to the value of K. So, we refrain from warping and scaling the image to preserve the intrinsic properties of the camera. We need the intrinsic matrix in the multiview geometry calculations. The cropped images are shown in the Fig. 2.9 below.

Figure 2.8: Raw images from the U2Eyes dataset. Porta et. al [13].
One way of cropping is to keep the iris in the center of the frame and the other is keeping the eye region center as the center of the frame. If we keep the center of iris (mean of iris landmarks) in the center of the frame, we may lose the translation term and only keep the rotation term. If we keep the eye region center (mean of interior eye landmarks) as the center of the frame, we can keep both the translation and rotation terms of the iris. For instance, using the U2Eyes dataset, the raw image is shown in the Fig. 2.8. We can see the cropped eye images in Fig. 2.9 with the annotated eye region landmarks.

![Cropped Images from the U2Eyes Dataset with annotated 2D iris landmarks.](image)

### 2.2.2 Effect of Cropping on Intrinsic Parameters.

In the above image we observe the effects of cropping on the focal length \((f_x, f_y)\) and principle points \((p_x, p_y)\) of the image. As shown in Fig 2-10 below, the intrinsic parameters comprise of the intrinsic matrix and distortion parameters which details the properties of the camera used to capture the observer. The intrinsic matrix comprises of the focal length, principle points and the skew parameter. The distortion parameters consist of the tangential and radial distortion.
Here, \( K \) is the intrinsic matrix,

\[
K = \begin{bmatrix}
  f_x & 0 & p_x \\
  0 & s & p_y \\
  0 & 0 & 1
\end{bmatrix}
\]  \hspace{1cm} (2.1)

Where, \( f_x \) and \( f_y \) are the focal length along x and y axis respectively and \( p_x \) and \( p_y \) are the principle points along the x and y axis respectively.

Since we deal with synthetic dataset, it doesn’t suffer from radial and tangential distortion. It has been known that the image in the dataset have dimension of 3840x2160 pixels and the focal length is 2820 pixels. Since we crop the image to dimension 320x172, the new dimension is (1/12) of the original image dimension. We again assume the new image center to be the principle points which in this case is \((p_x = 160, p_y = 88)\). The focal length is kept the same.

### 2.2.3 Normalization of the eye images.

This normalization method has been derived from the work of Park. et. al \[30\], which was applied to the network for estimating the eye region landmarks. Firstly, the eye image is required to be cropped from the entire full resolution image. Interior eye region landmarks, caruncle and the iris landmarks are used to determine the left, right and the center of the eye for cropping. Later, the
cropped image is scaled to constrain the width of all eye images to be the same size. Then these images are translated such that the eye center is the center of the image. For augmentation, a small translation and rotation can be added to the dataset, for the network to show robust performance. Once the geometric augmentations are dealt with, the images are converted to grayscale and blurred with randomized standard deviation. Finally, histogram equalization is applied to spread out the most frequently occurring intensity values. Fig 2.11* demonstrates the normalization.

![Original Eye Image - Cropped Left Eye Image - Processed Left Eye Image](image)

*Figure 2.11: Demonstrates the normalization of the eye images for the neural networks. The first row shows the raw images from the U2Eyes dataset, the second column shows the cropped grayscale images and the third column shows the normalized eye images.*

### 2.2.4 Plotting the Gaze Vector in the U2Eyes Dataset.

Since we could not obtain the 2D gaze vector directly from the dataset, we use the 3D iris landmarks and use Principle Components Analysis to determine the 3D gaze. Firstly, we crop the

* Reference for eye image normalization: https://github.com/david-wb/gaze-estimation
image as described in the normalization process where the landmark points are scaled appropriately. Then we back project the 2D iris points into the 3D space with the known scaled depth and obtain the 3D iris landmarks for the scaled image. We, then use Principal Components Analysis, widely known as PCA to determine the gaze vector, which is then back projected onto the image plane.

As an alternate way to determine the gaze vector, we can use the concept of pose estimation using the 3D and 2D iris landmarks available to us. The process of pose estimation using the Perspective-n-Point problem is defined in Section 3.4.
Chapter 3.
Concepts of Multiview-Geometry

3.1 Homography Matrix.

Homography is essentially a projective transformation that maps a point \( x (u, v, 1) \) from one plane to the corresponding point \( x' (u', v', 1) \) on the other plane. Mathematically, the 3x3 \( H \) matrix can be represented as,

\[
H = \begin{bmatrix}
H_{11} & H_{12} & H_{13} \\
H_{21} & H_{22} & H_{23} \\
H_{31} & H_{32} & H_{33}
\end{bmatrix}
\] (3.1)

The point \( x \) in a given plane can be projected onto the other plane at \( x' \) as shown in the Fig. 3.1\(^\circ\) using homography matrix (H) from Eq 3.1.

\[
x' = H x
\] (3.2)

*Fig 3.1: Demonstrates the homography relation between two planes [22].

\(^\circ\) Fig 3.1: Reference: https://docs.opencv.org/3.4/d9/dab/tutorial_homography.html
Given, a point \( x = [x_1, y_1, 1] \) in a given plane is projected to a point \( x' = [x'_2, y'_2, 1] \) on another plane as shown in Fig 3.1. Mathematical equations shown in the following section have been derived from “Multiview Geometry in Computer Vision” by Richard Hartley and Andrew Zisserman [39] and “Computer Vision: Algorithms and Applications” by Richard Szeliski [40].

\[
x' = \begin{bmatrix}
    x'_1 \\
    x'_2 \\
    x'_3
\end{bmatrix} = \begin{bmatrix}
    H_{11} & H_{12} & H_{13} \\
    H_{21} & H_{22} & H_{23} \\
    H_{31} & H_{32} & H_{33}
\end{bmatrix} \begin{bmatrix}
    x_1 \\
    x_2 \\
    x_3
\end{bmatrix}
\] (3.3)

The points \((x'_2, y'_2)\) in the second plane are given as,

\[
x'_2 = \frac{x'_3}{x'_3} = \frac{H_{11}x_1 + H_{12}x_2 + H_{13}}{H_{31}x_1 + H_{32}x_2 + H_{33}}
\] (3.4)

\[
y'_2 = \frac{x'_3}{x'_3} = \frac{H_{11}x_1 + H_{12}x_2 + H_{13}}{H_{31}x_1 + H_{32}x_2 + H_{33}}
\] (3.5)

We need a minimum of such 4-point correspondences to determine homography matrix within a scale factor.

Further, we can represent the rows of 3x3 homography (H) matrix as \( h_1, h_2 \) and \( h_3 \) respectively. We can rearrange the equations as follows,

\[
\begin{bmatrix}
    0^T & x^T & y'x^T \\
    x^T & 0^T & -x'x^T
\end{bmatrix} \begin{bmatrix}
    h_1 \\
    h_2 \\
    h_3
\end{bmatrix} = \begin{bmatrix}
    0 \\
    0
\end{bmatrix}
\] (3.6)

The above equation is of the form \( Ah = 0 \), where \([h_1, h_2, h_3]^T\) is a 9-element vector which we want to solve for. As seen from the Eq. 3.6, we can obtain the above two equations corresponding to a point correspondence between the two images. We can obtain 2nx9 matrix, with \( n \) to be the number of point correspondences between the two images. We can then solve for \( h \) in the above Eq. 3.6 considering it to be nonzero. To solve for \( H \), eigenvector that consists of the lowest eigenvalue is
computed. Singular value decomposition of A matrix gives the decomposed form $UDV^T$. H matrix can be determined from the rearmost column of the V matrix. Thus, it can be shown that a 2D point on a plane can be projected onto the second plane using the homography H matrix.

### 3.2 Using Co-Planar Points for estimating the pose of the camera.

We refer to the OpenCV’s blog on basic concepts of homography for determining the following equations [22]. A given 3D point $X(x, y, z)$ in World Coordinate System, can be projected onto the image plane at a point $x(u, v)$ using the camera projection matrix $P$ defined as,

$$ x = K [R|T] X $$  \hspace{1cm} (3.7)

We assume that all the 3D points in the World Coordinate System lie in the same plane. Since the depth of all the points is same, we can say,

$$ X = (x_1, y_1, 0, 1) $$  \hspace{1cm} (3.8)

Substituting the value of $X$ in Eq. 3.9.

$$ x = K [R_1 R_2 R_3 T] \begin{bmatrix} x_1 \\ y_1 \\ 0 \\ 1 \end{bmatrix} $$  \hspace{1cm} (3.9)

$$ P = K [R_1 R_2 T] \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} $$  \hspace{1cm} (3.10)

Here, the homography matrix ($h$) is defined as the transformation described in Eq. 3.9 is planar,

$$ h = \emptyset K [R_1 R_2 T] $$  \hspace{1cm} (3.11)
\[ x = h \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} \] (3.12)

The camera projection matrix is then given by,

\[ P = K [R_1 \ R_2 \ (R_1 \times R_2) \ T] \] (3.13)

Since we know that the \( R_3 \) is perpendicular to both \( R_1 \) and \( R_2 \). Thus, if we have a minimum of 4-point correspondences between the points in the image plane and object plane, we can determine the pose of the camera in the World-Coordinate System. The equations have been derived from [22].

![Figure 3.2: The image demonstrates the relationship between the two image planes. Yang et. al [24].](image)

3.3 Decomposing homography matrix into 3D rotation.

![Figure 3.3: The left camera \( C \) is first rotated to align orientation with camera \( C' \) and then translated to align the camera center with camera \( C' \). Richard Hartley and Andrew Zisserman. 2003 [39].](image)
The following concept has been derived from the book by Richard Hartley and Andrew Zisserman [39]. The camera C and C’ are aligned by a two-step process. From the above figure we observe that the camera C is first rotated to align with the orientation of camera C’. Later the camera C is translated to align the camera centers of camera C and camera C’. Here, we observe that the rotation of the image plane in the first step is a homographic transformation (H). This means that the rotation information is encoded in the homography term. We can decompose the homography matrix (H) to obtain the 3D rotation term (R). The 3D rotation term is given by,

\[ R = K'^{-1}H K \]  

(3.14)

Here, R is the 3x3 rotation term and K and K’ represent the 3x3 intrinsic camera matrix belonging to camera C and camera C’ respectively.

![Figure 3.4: Demonstrates the rotation between two image planes. Szeliski R. (2011) [26].](image)

Now we derive the relationship between the 3x3 homography matrix (H) and 3x3 rotation matrix (R). As shown in the above figure, assuming that there is only rotational motion between the two cameras C and C’ and we can determine the homographic (H) relation between the two camera planes using the feature matches as discussed above. Here, camera C and camera C’ have a common center.
Given a set of 3D points \((X, Y, Z)\) in the WCS, we can project them into the image plane of camera C as shown in the Eq. 3.15. The projection of the 3D point onto the image plane of camera C is \((u, v)\).

\[
\begin{bmatrix}
  u \\
  v \\
  1
\end{bmatrix} = K \begin{bmatrix} I & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X \\
  Y \\
  Z \\
  1
\end{bmatrix}
\]  
(3.15)

Since there is only rotational motion between the two cameras C and C’, the 3D points can be projected into the image plane of camera C’ as shown in the Eq. 3.16. The projection of the 3D point onto the image plane of camera C’ is \((u’, v’)\).

\[
\begin{bmatrix}
  u' \\
  v' \\
  1
\end{bmatrix} = K' \begin{bmatrix} R & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X \\
  Y \\
  Z \\
  1
\end{bmatrix}
\]  
(3.16)

Here, K and K’ are the camera intrinsic matrix corresponding to camera C and C’

Equation 3.15 can be rewritten as,

\[
K^{-1} \begin{bmatrix}
  u \\
  v \\
  1
\end{bmatrix} = \begin{bmatrix} X \\
  Y \\
  Z \\
  1
\end{bmatrix}
\]  
(3.17)

The 3D point \((X, Y, Z)\) is same in both Eq. 3.15 and Eq. 3.16. Thus substituting \([X, Y, Z, 1]^T\) with \(K^{-1} \begin{bmatrix} u \\
  v \\
  1
\end{bmatrix}\) in Eq. 3.16,

\[
\begin{bmatrix}
  u' \\
  v' \\
  1
\end{bmatrix} = K' \begin{bmatrix} R & 0 \\ 0 & 1 \end{bmatrix} K^{-1} \begin{bmatrix}
  u \\
  v \\
  1
\end{bmatrix}
\]  
(3.18)
From the previous section we recall that the homography (H) between the two camera planes can be determined using a minimum of 4 feature matches.

\[
\begin{bmatrix}
    x'_n \\
    y'_n \\
    1
\end{bmatrix} = \begin{bmatrix}
    H_{11} & H_{12} & H_{13} \\
    H_{21} & H_{22} & H_{23} \\
    H_{31} & H_{32} & H_{33}
\end{bmatrix} \begin{bmatrix}
    x_n \\
    y_n \\
    1
\end{bmatrix}
\]  
(3.19)

From Eq. 3.18, we have

\[
\begin{bmatrix}
    u' \\
    v' \\
    1
\end{bmatrix} = K' [R] K^{-1} \begin{bmatrix}
    u \\
    v \\
    1
\end{bmatrix}
\]  
(3.20)

From Eq. 3.19 and Eq. 3.20, we have

\[
H = K' [R] K^{-1}
\]  
(3.21)

Rearranging Eq. 3.21, we have

\[
R = K'^{-1} H K
\]  
(3.22)

Given the camera calibration matrix, we can form a relation between the homography and rotation between the two image planes.

3.4 Perspective-n-Point Problem.

Problem of estimating the pose of the camera in the 3D space is called the Perspective-n-Point (PnP) problem. We can compute the pose of the camera if we have the 3D points in the World Coordinate System (WCS) with its corresponding 2D points in the Image Coordinate System (ICS) along with the intrinsic camera parameters (K). The pose of the camera in the WCS is given by a 3 DOF rotation matrix R, and 3 DOF translation matrix. We derive the following mathematical formulation from the blog “Learn OpenCV” [43].

As we have discussed earlier, we can project the 3D points in the 2D image plane in two steps as follows,
We can transform the 3D points from the World Coordinate System to the Camera Coordinate system using the extrinsic parameters (Rotation and Translation). Mathematically it is given by,

\[
\begin{bmatrix}
    x \\
    y \\
    z \\
    1
\end{bmatrix} = [R|T]
\begin{bmatrix}
    X \\
    Y \\
    Z \\
    1
\end{bmatrix}
\] (3.23)

Here, R is the 3x3 Rotation matrix and T is the 3x1 translation matrix.

\[
[R|T] = \begin{bmatrix}
    r_{11} & r_{12} & r_{13} & t_x \\
    r_{21} & r_{22} & r_{23} & t_y \\
    r_{31} & r_{32} & r_{33} & t_z
\end{bmatrix}
\] (3.24)

\[
R = \begin{bmatrix}
    r_{11} & r_{12} & r_{13} \\
    r_{21} & r_{22} & r_{23} \\
    r_{31} & r_{32} & r_{33}
\end{bmatrix}; \quad T = \begin{bmatrix}
    t_x \\
    t_y \\
    t_z
\end{bmatrix}
\] (3.25)

(X, Y, Z, 1) is the 3D point in world coordinate system and (x, y, z, 1) is the 3D point in CCS.

We can transform the 3D point from the camera coordinate system to the image plane with the camera intrinsic matrix (K).

\[
\begin{bmatrix}
    u \\
    v \\
    1
\end{bmatrix} = s \ K \begin{bmatrix}
    x_1 \\
    y_1 \\
    z_1
\end{bmatrix}
\] (3.26)

where, K is the intrinsic matrix

\[
K = \begin{bmatrix}
    f_x & 0 & p_x \\
    0 & f_y & p_y \\
    0 & 0 & 1
\end{bmatrix}
\] (3.27)
Where, \( f_x \) and \( f_y \) are the focal length along x and y axis respectively and \( p_x \) and \( p_y \) are the principle points along the x and y axis respectively. \((u, v)\) are the image points and \( s \) is the scale factor introduced because of unknown depth.

Because of the scale factor, the Equation 3.26 is now non-linear,

\[
\begin{bmatrix}
X \\
Y \\
Y \\
1
\end{bmatrix} = s \begin{bmatrix} R & T \end{bmatrix} \begin{bmatrix} X \\
Y \\
Z \\
1
\end{bmatrix}
\]  
(3.35)

Similar to homography estimation, we can use DLT to solve for the scale factor. However, DLT is not very accurate. We want to use an optimization scheme that solves the problem iteratively by reducing the error. Reprojection error can be used to iteratively solve the problem. A given 3D point can be correctly projected onto its corresponding 2D image point if the extrinsic parameters
(R and T) are accurate. The reprojection error can be used as a metric to evaluate the accuracy of the extrinsic parameters and can be iteratively corrected.

3.5 Using the Perspective-n-Point problem to determine the gaze of the Observer.

Here, we have the 2D cropped images and its corresponding 2D iris landmarks. If we have the 3D iris landmarks, or essentially the depth of the eye region, we can determine the pose of the camera and essentially the gaze of the observer. Here, the focus is to demonstrate the importance of the depth of the iris landmarks for determining the gaze of the observer from geometry. Using the above mathematics, we determine the following camera pose of the observer.

![Image](image_url)

*Figure 3.7: Results from using the PnP method. Here the pose of the camera is shown by the blue (z-axis), green (y-axis), red (z-axis).*

We conduct an experiment to demonstrate the importance of the depth of the 2D iris landmarks. We can use neural networks to determine the 3D facial landmarks from a 2D face image. We use Face Alignment Network developed by Bulat. et. al [42] to determine the 2D facial landmarks and its corresponding 3D facial landmarks for a given face image. The network provides landmarks for interior eye region but not the 2D iris region. However, we can use the 2D and 3D correspondences of the facial landmarks to obtain the pose of the camera with respect to the 3D facial landmarks using the concept of Perspective-n-Point problem. We can later use state of the art eye region landmarks prediction network [30] to determine the 2D iris landmarks. We can then project the 2D iris points in 3D space onto the known 3D facial landmarks with a scale factor. Since we do not know the depth scale, we use the mean depth of the interior eye landmarks as the depth scale for the iris landmarks. We can use PCA on the determined 3D iris landmarks to determine the gaze vector. However, we observe that the gaze vector is not accurate as the depth
scale of the 2D iris landmarks is not accurate as we use an approximated scaled depth for projecting it in 3D space. From this experiment we understand the importance of the depth information of the 2D iris landmarks.
Chapter 4.
Supervised Deep Learning Based Gaze Estimation.

4.1 Predicting Eye Region Landmarks using Deep Neural Networks.

It is a difficult task to determine the eye region landmarks from unlabeled eye images captured in the wild settings outside the laboratory settings, using the concepts of traditional computer vision. Generalized algorithms do not perform robustly on images with significant variations in appearance and illumination as shown in the MPIIGaze dataset. However, the Convolutional Neural Networks (CNNs) can be used to determine the eye region landmarks robustly as shown by Park et. al [30] with varying conditions. We use the technique presented by Park et. al [30] for determining the eye region landmarks. They provide a deep learning-based approach for estimating them. The network is trained on synthetic UnityEyes dataset; however, it performs extremely well on real eye images as well, as demonstrated on the MPIIGaze dataset. We demonstrate their approach and performance in this section and later utilize this algorithm to predict the depth of the iris landmarks and thus obtain the 3D eye region landmarks.

4.2 Deep Learning Based Supervised Eye Region Landmark Prediction.

We use the pre-trained network developed by Park. et. al [30] to determine the eye-region landmarks to further implement the multiview geometry algorithms. It is especially useful for images that do not have the labelled ground truth eye region landmarks. It is shown that the supervised learning architecture trained on synthetic eye image dataset, can determine the eye region landmarks accurately on real eye images in the wild settings. In this section we describe the training process of the neural network for predicting the eye region landmarks and later evaluate its performance on datasets such as MPIIGaze and U2Eyes.
4.2.1 Network Architecture for Predicting Eye Region Landmarks.

The solution to the problem of determining eye region landmarks has been inspired from the human pose estimation using the stacked hourglass network [37]. The stacked hourglass network is used to determine the 2D landmarks that represent the joints of the subject’s body. For the pose estimation problem, both the local features as well as global features of the subject are essential to accurately estimate the pose. Higher order features are further evaluated by stacking the hourglass networks to improve the overall performance of the network.

The network inputs a normalized grayscale eye image and can predict 32 eye region landmarks. This includes, 16 iris landmarks, 16 interior eye region landmarks.

4.2.2 Training Details.

The pretrained network\(^1\) that uses similar concepts to [30] for determining eye landmarks, that has been trained on the synthetic eye images using the UnityEyes image dataset. The images are normalized as discussed in Chapter 2. before training. Three hourglass networks are stacked together to predict the heatmaps for the eye region landmarks. The heatmaps represent the confidence of the presence of the eye landmarks. The eye region landmarks consist of the 16 iris landmarks and 16 interior eye landmarks as shown in the Fig. 4.2. The network predicts 32 heatmaps, which are later passed through the soft-argmax layer to determine the spatial landmarks corresponding to the heatmaps. The peak intensity of the heatmap is determined using the soft-argmax layer to determine the position of the landmarks. The pretrained network also predicts the

\(^1\) Reference for the pre-trained network: https://github.com/david-wb/gaze-estimation
iris center which we do not utilize. Supervised loss functions are utilized to train the network. First one is the heatmap loss and the second is the landmark loss.

1. Heatmap loss: Heatmaps represent the confidence of an eye region landmark at a given pixel location in the image. A gaussian heatmap is generated at each landmark location which represents the ground truth heatmap. L2 distance loss between the ground truth heatmaps ($H_{gt}$) and the predicted heatmaps ($H_{pred}^*$) is minimized.

$$L_{Heatmaps} = ||H_{gt} - H||^2$$ (4.1)

2. Landmark Loss: The predicted heatmaps are passed through the soft-argmax layer to determine the eye landmarks. L2 distance loss is minimized between ground truth landmark position ($L_{gt}$) and predicted landmark position ($L_{pred}$).

$$L_{Landmarks} = ||L_{gt} - L_{pred}||^2$$ (4.2)

For the training details, Adam Optimizer is used, with L2 regularization, batch size of 16, learning rate to of 0.0001 and ReLU activation function between the hidden layers. It can be demonstrated that the network is able to predict eye region landmarks on real eye images. This is particularly useful for datasets which do not have 2D eye landmark labels.

4.2.3 Quantitative Analysis of the Network for Predicting eye landmarks.

<table>
<thead>
<tr>
<th></th>
<th>U2Eyes</th>
<th>MPIIGaze</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris Landmarks (pixels)</td>
<td>2.03</td>
<td>N/A</td>
</tr>
<tr>
<td>Interior Eye Region (pixels)</td>
<td>2.27</td>
<td>N/A</td>
</tr>
</tbody>
</table>

*Table 4.1: The table demonstrates the accuracy of the neural network for predicting the 2D landmarks for the iris region and interior eye regions. Mean error of the prediction is shown in pixels.*

Here, the network has been trained on UnityEyes dataset. However, we see that the network performs reasonably well on the U2Eyes synthetic dataset as well with a mean error of 2.03 and
2.27 pixels for iris landmarks and interior eye region landmarks respectively. The N/A in MPIIGaze column suggests that we do not have ground truth iris landmarks for the dataset, but the visual results are shown in Fig. 4.3. The performance of the network for determining eye region landmarks in shown in the Fig. 4.2 and Fig. 4.3 below.

**Figure 4.2**: Shows the performance of the network on the U2Eyes Dataset. Left Column: Shows the high-resolution image from U2Eyes dataset. Middle column: Shows the preprocessed eye images cropped from the high-resolution image with ground truth eye region landmarks (green). Right Column: Shows the eye region landmarks (blue) predicted by the neural network.

**Figure 4.3**: Shows the performance of the network on the normalized eye images of MPIIGaze Dataset. Left Column: Shows the normalized eye images. Right Column: Shows the predicted iris landmarks (Red) and predicted interior eye landmarks (green).
4.3 Supervised Depth Prediction for the Eye Region Landmarks.

In Chapter 3, we discuss the importance of 3D eye landmark labels for implementing the multiview geometry algorithms. We want to use the above discussed network to determine the depth of the 2D eye landmarks. Thus, we add a depth regression module to the above discussed network as shown in the figure below. We want to use the 2D positional features of the predicted eye region landmarks along with the global features of the eye as priors to the added depth network for predicting the depth of the iris landmarks. We want to determine if the depth network trained on the synthetic eye images can accurately predict the depth of eye landmarks for real eye images captured in the wild settings. We use the network described in Fig. 4.4 to determine the depth of the 2D eye landmarks. The depth regression module consists of fully connected layers.

![Depth Regression Network](image)

*Figure 4.4: Depth Regression network for predicting the depth of the eye landmarks. The idea for the network has been inspired from Zhou. et. al [38] and Park et. al. [30].*

4.3.1 Depth Normalization

The ground truth depth of the iris landmarks is relative, as shown in Fig. 4.5. The variance of ground truth depth values is small as seen from the plot. Regression for a network is a difficult problem and small variance in the depth values add to the difficulty. We normalize the depth of the 16 iris landmark values for improved depth prediction. Given the depth $(D)$ of 16 iris landmark points, $D_i = \{D_1, D_2, \ldots, D_{16}\}$, we can normalize the values such that the relative depth now varies from 0 to 1.

$$D_{\text{norm}} = \frac{\max(D_i) - D_i}{\max(D_i) - \min(D_i)}$$  \hspace{1cm} (4.3)
Once the depth is normalized ($D_{\text{norm}}$), it is used as ground truth for the depth regression module.

![Normalized depth for the U2Eyes Dataset](image1)

![Ground truth Depth for the U2Eyes Dataset](image2)

Once the depth values are predicted by the network, we use the original depth scale to correct for the predicted depth scale. We can use the below equation to recover the original depth scale.

$$D_i = \max(D_i) - D_{\text{norm}} \cdot (\max(D_i) - \min(D_i))$$  \hspace{1cm} (4.4)

We have the depth scale for only the synthetic dataset with ground truth depth labels. For real eye images with unknown depth, we assume the depth scale to be similar to the synthetic images which adds to the error in depth prediction as discussed in section 4.3.5.

### 4.3.2 Training Details of Depth Network:

We train this network on U2Eyes dataset, as it is the only dataset with ground truth depth labels of iris landmarks. We want to determine the accuracy of depth estimation on the U2Eyes itself and more importantly test if the predicted depth can be extended to real images as well. We later test the trained network on MPIIGaze dataset. The training plots are as shown below. We train the network with 20,000 eye images from U2Eyes dataset, with batch size 32 and learning rate to be $10^{-4}$. The loss functions for training the network are as follows,

$$\text{Loss Function} = \lambda_1 \cdot L_{\text{heatmaps}} + \lambda_2 \cdot L_{\text{landmarks}} + \lambda_3 \cdot L_{\text{depth}}$$  \hspace{1cm} (4.5)
Here, $L_{landmarks}$ and $L_{Heatmaps}$ are discussed in the Section. 4.2.2 and $L_{depth}$ is defined as follows,

Depth Loss: The depth loss is defined as the L2 distance between the ground truth depth ($D_{gt}$) of the iris landmarks and the predicted depth ($D_{pred}$).

$$L_{depth} = \| D_{gt} - D_{pred} \|^2$$

(4.6)

The training process can be understood from the following plots as shown below,
4.3.3 Performance of the Depth Estimation Network on the U2Eyes Dataset.

![Eye Image](image)

Figure 4.11: Column 1 shows the gaze error, using the predicted depth of the 16 iris landmarks. The second column demonstrates the depth prediction error using the neural network.

Here, the light vector in the eye image in Fig. 4.7 is the ground truth gaze vector and the dark vector is the predicted gaze vector. As seen in the second column, the blue points represent the ground truth depth and the orange points represent the predicted depth. We use Principle Components Analysis to determine the gaze vector from the predicted depth and 2D iris landmarks.
4.3.4 Performance of the Depth Estimation Network on the MPIIGaze Dataset.

Since, we do not have the ground truth depth labels for the MPIIGaze dataset, we cannot plot the 3D iris landmark points. Here, the green vector represents the ground truth gaze vector and the red vector represents the predicted gaze vector. Also, due to the absence of the ground truth depth labels for MPIIGaze dataset, the normalized predicted depth for the iris landmarks is scaled using the scale provided by the synthetic dataset. Again, Principle Components Analysis is used to determine the gaze vector from the predicted depth and 2D iris landmarks.

Figure 4.12: Performance of the depth network on the real eye images.
4.3.5 Quantitative Evaluation of the Depth Network.

We use cosine similarity to determine the error in gaze prediction. The gaze prediction error defines the angular error between the ground truth gaze vector and the predicted gaze vector.

\[
\text{Similarity} \left( g_{gt}, g_{pred} \right) = \frac{g_{gt} \cdot g_{pred}}{\|g_{gt}\| \|g_{pred}\|} = \frac{\sum_{i=1}^{n} w_{i,gt} w_{i,pred}}{\sqrt{\sum_{i=1}^{n} w_{i,gt}^2} \sqrt{\sum_{i=1}^{n} w_{i,pred}^2}} \tag{4.7}
\]

Here, we compute the cosine of the angles between the predicted \( g_{pred} \) and the ground truth \( g_{gt} \) gaze vectors. The vector lengths are normalized in the denominator.

<table>
<thead>
<tr>
<th></th>
<th>U2Eyes</th>
<th>MPIIGaze</th>
</tr>
</thead>
<tbody>
<tr>
<td>Depth Error (%)</td>
<td>4.32</td>
<td>N/A</td>
</tr>
<tr>
<td>Gaze Error (°)</td>
<td>7.21</td>
<td>17.96</td>
</tr>
</tbody>
</table>

*Table 4.2: Performance of the supervised depth estimation network.*

The supervised depth estimation network trained on the U2Eyes dataset performs considerably with an average error in depth estimation of 4.32% on the U2Eyes dataset. Using the depth information, gaze vector can be obtained. The average gaze error on the U2Eyes dataset is 7.2 degrees. However, when the network trained on U2Eyes dataset is tested on the MPIIGaze dataset, the performance drops significantly. One reason is the unknown scale factor for the real eye images in the MPIIGaze dataset. Since, we do not know the depth scale for the real eye images, we use the scale obtained from the synthetic eyes. However, we see that small difference in the scale factor can significantly affect the predicted depth and hence the gaze vector. We also observe a larger movement of the iris in the U2Eyes dataset, compared to the MPIIGaze dataset, indicating a significantly different scale for both the datasets.
4.4 Homography Based Gaze Estimation

4.4.1 Using the concept of Homography decomposition for Gaze Estimation.

Since, we can determine the 2D iris landmark points using the CNN for eye images in the wild, we can determine the feature matches and hence the homographic transformation between the two image planes. Once we obtain the homography (H), we can determine the 3D rotation and translation between the two image planes by decomposing the homography matrix. Thus, we can determine the relative pose of the iris in the two frames and hence the relative gaze. We discuss the method to determine the homography matrix in Chapter 3. We can use the 2D iris landmarks as feature points to determine the homography between the 2 images using the concept in Direct Linear Transform (DLT) discussed in Section 3.1. Considering two different poses of the iris as shown in the Fig 4.13 below, the 3x3 Homography matrix can be computed as discussed on Section 3.1.

![Figure 4.13: Matching the 2D iris landmarks between two poses to determine the homography matrix.](image)

From Section 3.1, we recall that a minimum of 4 feature matches are required to determine the homography matrix (H). We have 32 iris landmark points to accurately determine the homographic relation between the two planes.

\[
\begin{bmatrix}
    x'_n \\
    y'_n \\
    1
\end{bmatrix} =
\begin{bmatrix}
    H_{11} & H_{12} & H_{13} \\
    H_{21} & H_{22} & H_{23} \\
    H_{31} & H_{32} & H_{33}
\end{bmatrix}
\begin{bmatrix}
    x_n \\
    y_n \\
    1
\end{bmatrix}
\]

(4.8)

Here \((x_n, y_n)\) are iris landmarks in first image and \((x'_n, y'_n)\) are iris landmarks in the second image. As mentioned above, minimum 4 such feature matches can be used to solve for the (3x3) H matrix. We have 32 iris landmark points to accurately determine the homography matrix.
between the two iris poses. Further, we can decompose the homography matrix to obtain the relative rotation between the two image planes. The decomposition is described in detail in Section 3.3 to determine the relation between the relative rotation matrix and homography matrix.

\[ R = K'^{-1} H K \]  \hspace{1cm} (4.9)

Here, \( K' \) and \( K \) are the intrinsic matrix for camera 1 and camera 2 respectively.

### 4.4.2 Performance of homography decomposition on U2Eyes dataset.

![Figure 4.14: Performance of the Homography Decomposition on the U2Eyes Dataset.](image)
4.4.3 Performance of homography decomposition on MPIIGaze dataset.

Figure 4.15: Homography Decomposition for MPIIGaze Dataset.
In the above images, first column represents the reference image and second column shows the target image. Given the reference gaze vector $g_{\text{ref}}$, we multiply the computed rotation ($R_{\text{ref}}$) matrix obtained from the homography decomposition to determine the predicted target gaze ($g_{\text{pred}}$) vector (shown in red).

\[ g_{\text{pred}} = R_{\text{ref}} g_{\text{ref}} \quad (4.10) \]

The third column shows the warped reference image using the homography computed between the reference and the target image with respect to the 2D iris landmark points. The warped reference image represents the change in orientation of the reference image plane to align with the target image plane. In the fourth column, we observe the ground truth target gaze vector (green) and the predicted target gaze vector (red) obtained from the computed relative rotation matrix.

### 4.4.4 Quantitative Evaluation for Performance of the Homography Decomposition Method

The Table 2. shows the gaze error, using the method of homography decomposition.

<table>
<thead>
<tr>
<th>Gaze Error (degrees)</th>
<th>U2Eyes</th>
<th>MPIIGaze</th>
</tr>
</thead>
<tbody>
<tr>
<td>11.67</td>
<td>16.36</td>
<td></td>
</tr>
</tbody>
</table>

*Table 4.3: Table demonstrates the error in relative gaze estimation using homography decomposition method.*

Comparing the results of the obtained relative gaze upon decomposing the homography matrix, we observe that the gaze error for MPIIGaze dataset is higher than the U2Eyes dataset. One reason is that the U2Eyes dataset is a synthetic dataset that has accurate ground truth iris landmarks. From visual inspection, we observe that the error in landmark prediction for the MPIIGaze dataset is higher than U2Eyes dataset. Landmark detection error is further propagated when they are used to compute the homography matrix and hence the relative rotation matrix. We also observe that there is an error in the relative gaze estimation for the synthetic dataset as well. The homography estimated from the 2D iris landmark points is very accurate, however there is an error in computing the relative rotation upon decomposing the homography matrix. Reason for the error in homography decomposition may be because of the assumption made in Equation 3.18 in Chapter.
3, which is used to compute the relative rotation matrix. An assumption is made that the 3D position of the iris points remains the same, when the camera is assumed to be rotated. In reality, there is a change in the position of the 3D iris points for any two given samples and the assumption may not hold true, leading to an error in the relative gaze estimation upon homography decomposition. For a given 2D projection of 3D iris points, there exist two possible orientations of the iris points in 3D space. In other words, both the orientations in the 3D space have the same 2D projection on the image plane. Therefore, there are two possible values of the homography that can be obtained from the 2D iris feature matches used to fit the solution. Out of the two only one solution, that is the orientation of the 3D iris points is correct. In the above results to compute accuracy, we assume that the correct orientation of the 3D iris is obtained in every case. One possible way to disambiguate from the two solutions is to use the 2D position of the iris in the image plane to determine the orientation of the 3D iris points.

4.5 Deep Learning Based Homography Prediction.

We can explore the application of deep learning for estimating the homography between any two input eye images, based on the 2D iris landmarks. We have seen that homography binds the information of the relative orientation (rotation and translation) of the iris pose between any two eye images, which can be extracted upon decomposing homography matrix (H). We want to explore existing deep learning framework that takes in two sequential eye images namely, target ($I_{t+1}$) and reference ($I_t$) images. The network outputs the predicted homography based on the orientation of the iris in the captured images. We require the 2D labeled iris landmarks for computing the homography between the two input eye images. Since, we want an end to end framework, we can combine the landmark prediction and homography estimation pipeline to directly decompose the homography matrix to obtain relative rotation and translation (R and T). The work is heavily dependent on ‘Deep Image Homography Estimation’ [27] by Daniel DeTone. et. al. and ‘Unsupervised Deep Homography’ [29] by Nguyen et. al.
4.5.1 4-Point Parameterized matrix.

We use the idea proposed by [27] to use the parameterized form of the homography matrix. The positional difference between the landmark correspondences of the two input eye images is used to determine a 4x2 parameterized matrix. Given a 2D point (x, y) in image A and its corresponding 2D point (x’, y’) in image B, we can calculate the positional difference as $\Delta x = x - x'$ and $\Delta y = y - y'$. Similarly, if we have a minimum of 4 such points, we can create a 4x2 parameterized matrix as shown in equation 4.12.

$$
\begin{bmatrix}
    x'_1 \\
    y'_1 \\
    1
\end{bmatrix}
= 
\begin{bmatrix}
    H_{11} & H_{12} & H_{13} \\
    H_{21} & H_{22} & H_{23} \\
    H_{31} & H_{32} & H_{33}
\end{bmatrix}
\begin{bmatrix}
    x_1 \\
    y_1 \\
    1
\end{bmatrix}
$$

(4.11)

$$
H_{4\text{pt}} = 
\begin{pmatrix}
    \Delta x_1 & \Delta y_1 \\
    \Delta x_2 & \Delta y_2 \\
    \Delta x_3 & \Delta y_3 \\
    \Delta x_4 & \Delta y_4
\end{pmatrix}
$$

(4.12)

It is difficult for a neural network to directly regress a 3x3 homography matrix. This is because the homography matrix consists of rotation ($H_{11}$, $H_{12}$, $H_{21}$, $H_{22}$) and translation terms. A small error in the rotational term may not significantly affect the L2 loss term during optimization, however, it may result in a large error in determining the homography (H) matrix. There is also chance of mixing up the rotation and translation terms [29]. The 2D correspondences between the features of the two images can be used to determine the 4-point parameterized matrix.

4.5.2 Deep Neural Network for Homography Estimation.

We use the architecture proposed by Nguyen et. al [29] and Daniel DeTone. et. al. [27] for determining the homography between the two eye images with respect to the 2D iris, which can later be decomposed to determine the relative gaze. The network architecture is shown in the Figure 4.16 below. The network inputs two grayscale images stacked together and outputs the 4-point parameterized matrix, which can be used to determine the homography matrix.
Here, the architecture for regression model is as shown in the figure 4.17 below,

Since, we want to determine the relative pose of the observer based on the orientation of the iris, we use the iris landmarks as features to determine the homography between the target and reference image. We consider two cases. In the first case we consider only 4 points for computing the parameterized matrix (4x2) and in another case we consider all 32 iris landmarks for computing the parameterized matrix (32x2) for homography.

From Eq 4.12 we compute the $\Delta x_n = x_{tn} - x_{rn}$ and $\Delta y_n = y_{tn} - y_{rn}$, where n is the number of iris landmarks which vary from 1 to 32. $(x_{t}, y_{t})$ refers to the (x, y) coordinate of the iris landmarks in the target image $(x_{r}, y_{r})$ refer to the (x, y) coordinate of the iris landmarks in the reference image.

The input to the network is the 2-channel target image $(I_{t+1})$ and reference image $(I_t)$ stacked together. The output is the $H_{\text{npt}}$ (nx2) matrix, which can be further used to determine the 3x3 homography matrix (H).
We use the following loss functions to train the proposed neural network,

1. **Parameterized Matrix Loss**: We reduce the L2 regression loss \( L_{parameterized} \) between the ground truth parameterized matrix \( H_{4pt} \) and the predicted parameterized matrix \( H_{4pt}^* \) by the network.

\[
L_{parameterized} = \| H_{4pt} - H_{4pt}^* \|^2 \tag{4.13}
\]

2. **Binary Photometric Loss**: We reduce the L1 binary photometric loss \( L_{photometric} \) between the target image and the warped reference image using the homography matrix \( (H) \) predicted by the neural network.

\[
L_{photometric} = \| I_{t+1} H_{pred} - I_t \| \tag{4.14}
\]

3. **Landmark Loss**: Here, the 2D iris landmarks \( P_{reference}(x_r, y_r) \) in the reference image \( (I_t) \) are projected onto the target image \( (I_{t+1}) \) using the homography matrix predicted by the neural network. L2 distance loss \( L_{landmark} \) is reduced between the ground truth 2D iris landmarks \( P_{target}(x_t, y_t) \) in the target image \( (I_{t+1}) \) and the projected 2D iris landmark points from the reference image \( (I_t) \) onto the target image \( (I_{t+1}) \).

\[
L_{landmark} = \| (P_{target}(x_t, y_t) - H_{pred} \cdot P_{reference}(x_r, y_r)) \|^2 \tag{4.15}
\]

Here, the binary photometric loss uses the shape of the iris as supervision for training the network. Here, as seen from the example below, we find the homography and hence, rotation of the image plane in 3D space to align the iris landmarks. However, we cannot use the photometric error for each pixel within the iris region because of the occlusion. When we warp the reference image to the target, we can overlay the reference iris landmarks onto the target iris landmarks with the help of homography, which represents the iris shape. However, as seen from Figure. 4.18, the per pixel photometric error cannot be utilized.
Spatial transformation module is used to ensure that the warping process is differentiable, and backpropagation can take place, allowing the network to learn. [29]

### 4.5.3 Performance of the Homography Estimation network.

We train the network with 10,000 images from the UnityEyes dataset. We use Adam Optimizer with learning rate of $10^{-4}$ and batch size of 16. The input to the network is the stacked grayscale target and reference images and the output is the 4-point parameterized matrix, which can subsequently be used to determine the homography (H) matrix.

The plot below describes the training process for the network,

![Training loss](image)

*Figure 4.19: Plot describing the training process.*
Training Loss = \lambda_1 \cdot L_{\text{parameterized}} + \lambda_2 \cdot L_{\text{photometric}} + \lambda_3 \cdot L_{\text{landmark}} \quad (4.16)

Here, the image demonstrates the visual performance of the neural network tested on the Unity Eyes dataset.

Figure 4.20: Performance of Homography Network. a) Reference Image; b) Target Image; c) Reference image warped using predicted Homography; d) Reference image warped using Homography matrix computed from traditional method.
We observe that the network is unable to determine the correct homographic relation between the reference and the target image. The reason for the failure of the network may be following. Firstly, while training we input the two stacked images to the network. We use the iris landmarks information for the ground truth parameterization matrix and the regression module tries to predict the parameterized matrix. The network does not have any prior positional information of the 2D iris landmarks. While testing, the two stacked images are input to the network and the 4-point parameterization matrix is predicted. Network may not be able to learn the positional information of the iris features from only the input images without the prior landmark information. We also observe that the binary photometric loss does not contribute significantly to the loss function of the network. Also, our inability to use the per pixel photometric error because of the occlusion discussed in Section 4.5.2 is a major drawback.
Chapter 5.
Unsupervised Gaze Estimation from SFM

5.1 Problem Statement.

Here, we want to use the concept of Structure from Motion to solve the problem of gaze estimation for an observer. In Structure from Motion, the 3D scene is determined from a sequence of 2D images of the scene with overlapping field of view where no prior information of the camera poses, or 3D scene is required. Firstly, the pose of the camera is determined, which is later used to triangulate the matched feature points in the images into the 3D space using the concepts of epipolar geometry to determine the 3D points in the scene. The problem of estimating 3D camera poses, and the 3D point cloud estimation is tackled simultaneously. Further, Bundle Adjustment is used to refine the computed camera poses, which is obtained by minimizing the nonlinear least square error. Usually in Structure from Motion the scene is static, and the camera is moved around in the 3D scene. The static objects in the scene are utilized to determine the point cloud of the 3D scene, while the dynamic objects are determined and masked out of the algorithm.

![Structure from Motion](image)

*Figure 5.1: Demonstrating the concept of Structure from Motion (SfM). Yilmaz et. al [41].*

However, considering the scenario of gaze tracking, the camera in the remote device such as a laptop is fixed, and the pose of the observer varies. The pose of the observer can be determined by their head orientation with respect to the camera and the gaze of the observer can be determined
by the orientation of the iris with respect to the camera. The head pose of the observer gives a good estimate of where the person is looking. However, if we want to accurately determine the gaze, we need to extract the orientation of the iris. We can look at the task of gaze estimation as an inverse problem to Structure from Motion, where the camera is static, and the scene is dynamic. Now, we can make an assumption that the camera is moving, and the scene is static, if we focus only on the iris region and mask out the rest of the objects in the scenes. As we discuss below, each elliptical shape of the iris corresponds to a unique camera pose. Thus, masking out the rest of the eye region as shown in the Fig. 5.2, we can leverage this property.

The movement of the observer involves both the movement of the head and the movement of the iris according to where the observer is looking. An intuitive way to think about it is as follows. If the person is looking straight directly into the line of sight of the camera device, the projection of the iris onto the image plane is circular. However, if the observer is looking away from the screen, at an angle gamma to the line of sight, the projection of the iris onto the camera plane would be elliptical. Each elliptical projection is unique to the direction in which the observer is looking. Thus, we can determine the pose of the observer by extracting the shape of the iris projection onto the image plane. Some of the images below, explain the proof of concept for the above assumption.

![Figure 5.2: Demonstrating each eye pose having unique ellipse shape.](image)

![Figure 5.3: Elliptical 2D projection of a 3D circular feature on the image plane. Shiu. et. al [35].](image)
The images in Fig. 5.4 describe the projection of a 3D circular object onto a 2D plane.

5.2 Unsupervised learning framework for gaze Estimation.

In this section we demonstrate an unsupervised end to end gaze estimation architecture using the concepts of multiview geometry for constructing the unsupervised loss functions. This method is heavily influenced from the works of Zhou et. al. [33]. We want to jointly train a depth and a pose network, using the 3D geometry constrains to develop unsupervised loss functions to train the network. The network inputs consecutive eye images with different iris poses and predicts their relative pose from using both the depth and the pose network. To understand the architecture, we first need to develop a mathematical foundation.

The entire pipeline for unsupervised gaze estimation is as follows:

---

\[ \text{Reference for Fig. 5.4: http://www.grad.hr/geomteh3d/Monge/11rotacija/rotacija_eng.html} \]
1. Firstly, we consider two consecutive eye images with different iris pose. The first frame namely target frame ($I_t$) and the second, the reference frame ($I_{t+1}$).

2. We compute the 2D iris landmarks for both the reference and the target frames as proposed in Section 4.2 using the neural networks.

3. As seen in the Chapter 3, we need to determine the depth of the 2D iris landmarks in the target image. We propose to use a depth network to determine the depth of the 2D iris landmarks and use it to project them from target image into the 3D space.

4. We want to determine the pose of the reference iris points with respect to the target iris points. If we have the prior information of the 3D pose between the target image and the reference image, we can back project the 3D iris landmarks from the world space onto the reference image plane. If, the pose information is accurate, the 2D iris landmarks projected from the target image, would coincide with the 2D iris landmarks on the reference image after projection. In this case the reprojection error is 0. However, since we are not aware of the pose, we use the Convolutional Neural Networks to predict the 3D relative pose (3 DOF Rotation and 3 DOF Translation) of the iris in both the frames and use the reprojection error as a metric to determine the loss function to train both the depth and pose network. We discuss each step of the entire pipeline in detail in the following sections.

5.2.1 Projecting the 2D iris landmarks into 3D space.

We have 32 iris landmarks for the eye as shown in the Fig. 4.3. From Chapter 3, we understand the mathematics of projecting a 3D point into the camera space and then into the image plane using the following equation.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K [R|T] \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}$$  \hspace{1cm} (5.1)

Here, $(x_1, y_1, z_1)$ is the 3D point in WCS, which is converted to the CCS using the extrinsic parameters (R and T) and $(u, v, 1)$ is the corresponding 2D image point. In this case we consider the 3D iris points to be in the CCS system and hence R = I and T = 0.
\[
\begin{bmatrix}
\bar{u} \\
\bar{v} \\
1
\end{bmatrix}
= K \begin{bmatrix} 1 & 0 \end{bmatrix}
\begin{bmatrix}
x_1 \\
y_1 \\
z_1
\end{bmatrix}
\] (5.2)

Similarly, we can project the 2D points from the image plane into the camera space using the following equation to a scale factor.

\[
\begin{bmatrix}
x_1 \\
y_1 \\
z_1
\end{bmatrix}
= \lambda K^{-1}\begin{bmatrix} \bar{u} \\
\bar{v} \\
1
\end{bmatrix}
\] (5.3)

As seen from the Equation 5.2, we can project the 2D iris points into the 3D space, but the depth is unknown to a scale factor as shown in the Fig. 5.5 below. We want to explore if the depth can be learnt using neural networks. As shown in Section 4.3 we can use a supervised learning approach to determine the depth of the 2D iris landmarks or use an unsupervised approach as discussed below.

**5.2.2 Convolutional Neural Network for 2D Iris Landmark Depth Estimation.**

We use an encoder-decoder based network, inspired from the ideas of Zhou et. al [33] for estimating the depth of the 32 iris landmarks. The network has skip connections, with 7 convolutional and deconvolutional layers. All layers use ReLU activation function. The final depth prediction layer uses sigmoid activation. The input to the network is the target image ($I_t$) and the output is the depth (D) of 32 iris landmarks points in the target image.

---

*Reference for Fig. 5.5: https://support.pix4d.com/hc/en-us/articles/202559089-How-are-the-Internal-and-External-Camera-Parameters-defined*
5.2.3 Determining the pose of the iris in the Target image ($I_t$) with respect to the iris in Reference frame ($I_{t+1}$).

Once we have the depth, we want to determine the pose of the iris in the target image with respect to the iris in the reference image. If the pose between the reference and target images is known, we can back-project the 3D iris landmarks determined in Section 5.2.2 from the reference image onto the target image. Given the correct pose, the geometry dictates that after back projecting the 3D iris landmarks of the target image onto the reference image, both the target and reference landmark points coincide. It is mathematically demonstrated by the equations below.

Let the 2D iris points in the target image be $(u_t, v_t)$ and the 2D iris points in the reference image be $(u_r, v_r)$.

\[
\begin{bmatrix}
    x_1 \\
    y_1 \\
    z_1
\end{bmatrix} = D K^{-1} \begin{bmatrix}
    u_t \\
    v_t \\
    1
\end{bmatrix}
\]

Thus from Eq. 5.4, $(x_1, y_1, z_1)$ is the projected target point in the 3D space and D is the depth of the projected 3D landmark points.

If we have the relative pose ($P_{t\rightarrow r}$) between the target and the reference frame, we can back-project the 3D target iris points $(x_1, y_1, z_1)$ onto the 2D reference image.
Here, $P_{t\rightarrow r}$ is the relative rotation (R) and translation (T) from the target to the reference frame and $K$ is the camera intrinsic matrix.

Thus, we can mathematically observe that the 2D iris points in the target image can be projected onto the reference image given the depth (D) and the relative pose ($P_{t\rightarrow r}$) between the two frames.

\[
\begin{bmatrix}
    u_r \\
    v_r \\
    1
\end{bmatrix} = K P_{t\rightarrow r} D K^{-1}
\begin{bmatrix}
    x_t \\
    y_t \\
    z_t
\end{bmatrix}
\]  

(5.6)

If the iris points from target image are projected into 3D space at a correct depth scale and then back projected into the reference image using the accurate relative pose, the projected 2D target iris points and 2D reference iris points should coincide. This geometric property can be leveraged to develop the unsupervised loss functions.

Figure: 5.7: Relative Pose between the two iris orientations.
5.2.4 Convolutional Neural Network for Relative Pose Estimation

In section 5.2.3 the mathematics is only valid if we have the pose between the target and reference image. Since we do not have the ground truth pose, we can use the following CNN to determine the relative pose of the two frames. The network takes in the stacked input target and reference images and outputs a 6 DOF pose (3 DOF rotation and 3 DOF translation matrix.)

![Figure 5.8: Pose Network Architecture design, inspired from the work of Zhou et. al [33].](image)

The network architecture has been inspired from the work of Zhou et. al [33]. The network has 7 convolutional layers and ReLU activation has been used for all layers. No activation function is used for the final layer, where the pose is predicted. The network outputs a 6x1 matrix, which represents the relative pose between the reference and target image.

5.2.5 Framework for Unsupervised Gaze Estimation.

In this section, we summarize the previous discussion and describe the entire unsupervised gaze estimation framework as described in image below.

We take two sequential images from the dataset as the target view ($I_{t+1}$) and the reference view ($I_t$) as shown in the Figure 5.9. The depth network takes in the target image and predicts the depth (D) of the iris landmarks.
Subsequently, the reference and target image are stacked together and input to the pose network. The pose network predicts the pose between the target and the reference image. We can use the predicted pose to project the points back onto the reference image.

\[
\begin{bmatrix}
  x_1 \\
  y_1 \\
  z_1 
\end{bmatrix}
= D K^{-1}
\begin{bmatrix}
  u_t \\
  v_t \\
  1
\end{bmatrix}
\]

(5.5)

We want to train the pose and the depth network simultaneously using the geometric loss constraints developed in the previous sections. If the depth and the pose values are predicted accurately, the 2D iris landmark points in the target image, projected on the reference image would perfectly coincide with 2D reference iris points. If there is some error in depth or pose estimation, there will be a large reprojection error between projected 2D target iris landmarks and reference iris landmarks. We can use the reprojection error as one of the metrics for evaluating the training loss of the neural network. We see that the training of the depth and the pose networks are interdependent on each other. If the depth prediction is inaccurate, the pose network would not be able to predict the correct pose. Similarly, if the pose prediction is incorrect, even though the depth
predicted by the depth network is accurate, the projection of the iris landmarks will have error, giving erroneous gaze estimation results.

5.3 Training Details.

We train the depth and pose network simultaneously, with common loss functions. We use 5000 images for training, with Adam Optimizer and a learning rate of $10^{-4}$, with a batch size of 4. The batch size has a limitation because of the simultaneous computations necessary for the equations.

5.3.1 Loss Functions for Unsupervised Pose and Depth Estimation.

1. Landmark Loss: As discussed in the previous section, we can project the 32 iris landmarks into the 3D space using the depth network, and then back-project it onto the adjacent frame (reference) using the pose determined from the pose network. If the depth and pose predictions are accurate, the 32 iris landmarks in the reference image coincide with the iris landmarks projected from the target image onto the reference image, which means the reprojection error is 0. We want to reduce the reprojection loss.

$$L_{\text{landmarks}} = ||x_{\text{reference}} - KP_{t\rightarrow r}DK^{-1}x_{\text{target}}||^2$$ (5.7)

Here, $x_{\text{target}}$ represents the iris co-ordinates of the target image $[u_t, v_t, 1]^T$, D is the predicted depth, K is the camera matrix, $P_{t\rightarrow r}$ is the predicted pose from the pose network and $x_{\text{reference}}$ represents the iris co-ordinates of the reference image.

2. Binary Photometric Loss: As discussed in chapter 4, we can use the unique shape of the ellipse for supervision of the pose and the depth network. We mask out the 2D iris from the eye image and use the binary mask for supervision. We call the binary iris mask of the reference image as reference mask and binary iris mask of the target image as target mask. The reference mask is warped onto the target mask using the predicted pose and depth. If the pose and depth values are predicted accurately, the warped iris shape of the reference mask is similar to the iris shape of the target mask.
\[ L_{binary\_photometric} = \| I_{target} - I'_{target} \|^2 \] (5.8)

Here, \( I'_{target} \) is the reference iris mask that has been warped, based on the predicted pose and depth network.

Loss Function = \( \lambda_1 \cdot L_{binary\_photometric} + \lambda_2 \cdot L_{landmarks} \) (5.9)

Figure 5.10: Eye image along with cropped iris region and interior eye region. a) RGB eye Image, b) Segmented Iris mask, c) Segmented Eye Region mask, d) Segmented RGB Iris image.

5.3.2 Training Details for unsupervised Pose and Depth estimation network.

The network is trained on the synthetic U2Eyes dataset. It is trained on 10,000 training images and tested on 500 test images. The network is trained with Adam Optimizer with a learning rate of 0.0001. We observe that the contribution of the binary photometric loss is not significant for training the network.

Figure 5.11: Training Loss for Unsupervised Depth and Pose Estimation.

5.3.3 Performance of the Unsupervised Pose and Unsupervised Depth Estimation Network.

We observe that the performance of the unsupervised depth and unsupervised pose estimation
network is not accurate with a mean error of 23.17 degrees. From the training loss, we observe that the landmark loss is not effective to train the network. More efficient loss function is necessary to constrain the loss to effectively train the network. Further analysis shows that after few epochs the unsupervised depth network is not able to predict the correct depth of the eye-region landmarks. The depth values saturate with all the landmark points having the same depth value. The depth network is unable to determine the small relative depth of the iris landmarks. Since the depth values are incorrectly predicted, the predicted pose is also incorrect.

Unsupervised Pose, Unsupervised Depth

![Plot of the gaze error for unsupervised depth and pose estimation network for U2Eyes dataset](image)

**Figure 5.12**: Plot of the gaze error for unsupervised depth and pose estimation network for U2Eyes dataset

5.3.4 Visual Performance Analysis of the Unsupervised Pose and Depth Network on U2Eyes Dataset.
Figure 5.13: Performance of the unsupervised depth and unsupervised pose estimation network for gaze estimation on the U2Eyes Dataset.

5.3.5 Visual Performance Analysis of the Unsupervised Pose and Depth Network on MPIIGaze Dataset.
Figure 5.13 and 5.14 demonstrate the performance of the unsupervised pose and depth network on the synthetic and real eye images. In Fig. 5.13, the first column represents the target image and the red vector represents its gaze vector label. The second column shows the reference image and the green vector represents its gaze vector label. The third column shows the ground truth gaze vector for the reference image in green and the predicted gaze vector in red. Here, the relative rotation between the reference and the target image is obtained from the 6 DOF predicted pose. The red gaze vector in the third column is the predicted gaze vector, obtained by multiplying the ground truth gaze vector of the target image and the relative rotation matrix predicted by the network.

Similar approach is followed for representing the predicted gaze vector for the MPIIGaze dataset in Fig. 5.14. Here, we train the above network on MPIIGaze dataset on 10,000 images. While testing, we observe that the unsupervised depth and pose estimation network performs poorly as shown in Fig. 5.14. We observe that the rotation matrix predicted by the network is close to identity in every case. One reason for the inability of the network to determine the relative gaze vector is the fact that the iris movement in the MPIIGaze dataset is considerably small.

One reason for the error in both the datasets is the fact that depth values predicted by the depth network is same. Since both the pose and depth network are trained on common loss functions, the pose network cannot predict accurate relative pose and hence the gaze.
5.3.6 Training Details for Supervised Depth and Unsupervised Pose Estimation Network.

We observe from Section 5.3.5 that the depth network cannot estimate the correct depth values. The depth network cannot differentiate between the small differences in the depth values of the iris landmarks as shown in the Chapter 4. Because of the error in depth prediction, the pose prediction is erroneous. To investigate the effectiveness of the landmark loss function, we supervise the depth network using the ground truth depth values.

![Landmark Loss](image)

*Figure 5.15: Training Loss for Supervised Depth and Unsupervised Pose Estimation.*

5.3.7 Performance of the Supervised Depth and Unsupervised Pose Estimation Network.

![Gaze Error Density](image)

*Figure 5.16: Plot of the gaze error for Supervised Depth and Unsupervised Pose estimation network for U2Eyes dataset.*
We observe that the mean gaze error is reduced by a small factor when the depth is supervised. The plot in Fig. 5.16 demonstrates the performance of supervised gaze estimation network with the mean gaze error to be 21.75 degrees. The density in the plot represents the normalized number of images.

5.3.8 Visual Performance Analysis of the Supervised Depth and Unsupervised Pose.

Figure 5.17: Results for gaze estimation with supervised depth and unsupervised pose estimation network.
From the training plots above, we observe that the landmark loss is the major contributor for the training of the network. The effectiveness of the photometric binary loss is insignificant. It may end up inhibiting the network’s ability to learn. We also observe that upon depth supervision, the accuracy does not increase significantly. This indicates that the unsupervised constraints are not effective in training the network robustly. The landmarks constraint is the most important contributor; however, the network cannot reduce the landmark error below 50 pixels even after supervising the depth network. In many cases, the mean movement of the iris between two consecutive images may be less than 50 pixels. Thus, the network may not be able to accurately predict the pose in such cases. For improving the performance of the network, more effective geometric loss functions need to be developed. Our inability to utilize the per pixel photometric information for the loss function is a major drawback as discussed in Section 4.5.2. Also, we use the depth of just 32 iris landmark points as supervision. We hypothesize that the performance of the network can be improved if we have per pixel depth of the iris.

| Table 5.1: Performance of the Unsupervised Gaze estimation network on U2Eyes Dataset. |
|-------------------------------------------|-----------------|-----------------|
|                                          | 23.17           | 21.75           |
| Gaze Error (∘)                            |                 |                 |

From the training plots above, we observe that the landmark loss is the major contributor for the training of the network. The effectiveness of the photometric binary loss is insignificant. It may end up inhibiting the network’s ability to learn. We also observe that upon depth supervision, the accuracy does not increase significantly. This indicates that the unsupervised constraints are not effective in training the network robustly. The landmarks constraint is the most important contributor; however, the network cannot reduce the landmark error below 50 pixels even after supervising the depth network. In many cases, the mean movement of the iris between two consecutive images may be less than 50 pixels. Thus, the network may not be able to accurately predict the pose in such cases. For improving the performance of the network, more effective geometric loss functions need to be developed. Our inability to utilize the per pixel photometric information for the loss function is a major drawback as discussed in Section 4.5.2. Also, we use the depth of just 32 iris landmark points as supervision. We hypothesize that the performance of the network can be improved if we have per pixel depth of the iris.
Chapter 6.
Conclusion and Future Work.

The aim of the thesis was to explore the concepts of 3D geometry along with deep learning to determine the gaze of the observer from remote cameras. Most of the state-of-the-art algorithms rely on supervised deep learning-based algorithms for estimating the gaze of the observer. It is difficult to generate high quality ground truth gaze labels of the observer in the wild settings. Since these supervised networks rely on the CNN to extract the high-level features, their performance drops upon evaluation on images significantly different from the training set. We want to explore the use of the geometric eye features to develop an unsupervised learning architecture for estimating the relative gaze between the two poses of the observer. We start by demonstrating the importance of the depth information of the 2D iris landmarks for implementing the multiview geometry algorithms. First, we explore the supervised learning architecture for predicting the depth of the 2D iris landmarks, which can be further used to determine the gaze vector. Later, we explore the performance of the unsupervised depth and pose estimation network that uses the concept of Structure from Motion and deep learning for estimating the gaze. To further understand the shortcomings of the above network, we supervise the depth and predict the unsupervised relative pose of the 3D iris landmarks for consecutive eye images. We also dig deeper into the concept of homography decomposition for gaze estimation using traditional epipolar geometry concepts. Further, we explore the use of neural networks for homography estimation, which can later be decomposed into rotation and translation to determine the relative gaze. We implement all the above discussed algorithms on both synthetic and real eye images and further compute and evaluate the results.

First, we explore the performance of the pre-trained network [30] developed by Park et. al to determine the eye region landmarks for both real and synthetic eye images. It was shown by [30] that with appropriate normalization and pre-processing, neural networks can be trained on synthetic dataset with ground truth eye region landmarks, to accurately predict the eye landmarks.
for real eye images in the wild settings. We further extend the idea to determine the depth of the computed 2D iris landmarks. We fuse a supervised regression network [38] with the eye landmark prediction network [30]. Reason for fusing both the networks is that we can use the prior 2D positional information from the landmark prediction network, which can help the depth prediction network for robust depth estimation. The supervised learning algorithm is trained on synthetic U2Eyes dataset using the ground truth depth labels. We see that the network performs well on the synthetic U2Eyes dataset with a depth error of 4.32 %. However, it cannot accurately predict the depth of the iris landmarks for the MPIIGaze dataset. One reason for the inaccuracy in depth prediction for the real eye images may be the unavailability of the ground truth depth scale for the MPIIGaze dataset. The normalized depth predicted by the network has a range of 0 to 1, which needs to be scaled back to the ground truth scale. However, since we do not know the ground truth depth scale of the real eye images in the MPIIGaze dataset, we use the depth scale of the synthetic dataset. However, we see that the movement of the iris in the synthetic dataset is more compared to the real eye images in the MPIIGaze dataset, indicating a different depth scale for both the datasets. One hypothesis is the assumption of the depth scale for real eye images, may result in a higher error in depth prediction for MPIIGaze dataset compared to U2Eyes synthetic dataset.

We explore the traditional computer vision algorithms and multiview geometric constraints, that can be leveraged to determine the relative gaze of the observers. We dive deeper into the use of homography decomposition to determine the change in orientation of the 3D planar iris landmarks to determine the relative gaze. We see that the performance of the synthetic images is more robust compared to the real eye images. This is because of the more accurate ground truth information of the eye region landmarks for the synthetic U2Eyes dataset. Since we do not have the ground truth 2D iris landmarks for the MPIIGaze dataset, we use the 2D iris landmarks predicted by the network [30]. However, we can see that the error in landmark detection for the images of MPIIGaze dataset is higher than the U2Eyes dataset. This error in landmark detection is carried forward to the homography detection and rotation matrix extracted from the homography decomposition. We also observe an error in the relative gaze estimation after homography decomposition for the synthetic U2Eyes dataset, even though the predicted landmarks are accurate. On further investigation we see that the predicted homography from the 2D iris landmarks is very accurate. One hypothesis for the error in the homography decomposition is the assumption made in the Eq. 3.18 that the position
of the corresponding 3D iris points is same for both the 2D iris poses in the two consecutive eye images. However, in reality the 3D position of the iris landmarks may change for the two eye images.

Later, we explore the use of the deep learning to solve the problem of gaze estimation using the concepts of traditional computer vision for developing the loss functions. Inspired from the ideas of Nguyen et. al [29] and Daniel DeTone. et. al. [27] we use a CNN to predict the homography between two eye images to determine the change in orientation of the 3D planar iris and hence the relative gaze of the observer. We can observe that the performance of the network in determining the homography from iris landmarks is poor and cannot achieve the accuracy similar to the traditional methods.

Later, we leverage the idea proposed by Zhou et. al. [33] to use the concept of Structure from Motion and develop an unsupervised network architecture that uses unsupervised geometric constraints. The depth and the pose network are trained in an unsupervised manner with the loss constraints discussed in Chapter 4. We demonstrate the performance of the network with an accuracy of 23.17 degrees on the synthetic U2Eyes dataset. We observe that the depth network saturates and predicts similar depth values for all the iris landmarks. It is unable to differentiate the small depth difference between the iris landmarks. As seen from the loss functions, with an error in depth estimation network, the pose network can’t be trained correctly as both the networks are trained based on a common loss function. To further investigate, we supervise the depth network with the ground truth depth from the U2Eyes dataset. We observe that the performance of the pose estimation network still does not improve significantly with an error of 21.75 degrees. One hypothesis is the inability of the network to reduce the landmark loss, since it is the major contributor in the training of the network. There is a need to develop more robust and accurate unsupervised loss constraints by further diving deeper into the concepts of 3D geometry.

To overcome the challenges, as discussed above we want to explore more efficient loss constrains that can be more effective to train the unsupervised network robustly. In-depth knowledge of neural networks can be leveraged to understand the performance of the depth network in the unsupervised learning framework in Section 5.3.3. Large public datasets with ground truth depth
for the eye region are not easily available. There are datasets that capture the depth of entire face images; however, they cannot be used for the purpose of depth estimation for the eye region. This is because there is significant head movement in the datasets, but no significant eye movement to robustly learn the depth for different poses of the iris. We further propose to capture our own dataset using depth camera to obtain the ground truth depth information for the eye region. The ground truth depth information can be used to train the supervised depth and unsupervised pose network as shown in section 5.3.4. We also want to explore the use of the 3D Morphable Models (3DMM) of human faces to determine the depth of the eye region. These depth values can be used for supervision, for accurate pose prediction. In-depth understanding of the 3DMM models is necessary to determine the robustness of using the models of different face images to determine the depth of the eye region. Either way, from the ground truth depth information or 3DMM models can be used to train networks for estimating the depth of the eye region for large number of observers. Later this pretrained network depth network can further be jointly trained with the unsupervised pose network for estimating the relative pose of the iris as the observer looks in different directions.
References


[22] https://docs.opencv.org/3.4/d9/dab/tutorial_homography.html


[34] http://www.grad.hr/geomteh3d/Monge/11rotacija/rotacija_eng.html


[43] https://www.learnopencv.com/tag/solvepnp/  
