NOVEL INTUITIVE HUMAN-ROBOT INTERACTION
USING 3D GAZE

by
Songpo Li
A thesis submitted to the Faculty and the Board of Trustees of the Colorado School of Mines in partial fulfillment of the requirements for the degree of Doctor of Philosophy (Mechanical Engineering).

Golden, Colorado
Date ______________________

Signed: ______________________
   Songpo Li

Signed: ______________________
   Dr. Xiaoli Zhang
   Thesis Advisor

Golden, Colorado
Date ______________________

Signed: ______________________
   Dr. Greg Jackson
   Professor and Head
   Department of Mechanical Engineering
ABSTRACT

Human-centered robotics has become a new trend in robotic research in which robots closely work around humans or even directly/indirectly make contact with humans. Human-centered robotics not only requires the robot to successfully and safely accomplish the given task, but also requires it to establish a rapport with humans by considering human factors. Human-robot interaction (HRI) has been an essential component in human-centered robotics due to the fundamental information exchange between the human and the robot, which plays an essential role in the task success and rapport establishment. In this dissertation, human gaze, which indicates where a person is looking, is scientifically studied as an intuitive and effective HRI modality. The gaze modality is natural and effortless to utilize, and from gaze modality, rich information about a user’s mental state can be revealed. Despite the promise of gaze modality, applying gaze as an interaction modality is significantly limited by the virtual gaze tracking technology available and low-level gaze interpretation.

Three-dimensional (3D) gaze tracking in real environments, which measures the 3D Cartesian location of where a person is looking, is highly desirable for intuitive and effective HRI in human-centered robotics. Employing 3D gaze as an interaction modality not only indicates the manipulation target, but also reports the location of the target and suggests how to perform the manipulation on it. The goal of this dissertation is to achieve the novel 3D-gaze-based HRI modality, with which a user can intuitively express what tasks he/she wants the robot to do by directly looking at the object of interest in the real world. In working toward this goal, the investigation concentrates on 1) the technology to accurately sense where a person is looking in real environments and 2) the method to interpret the human gaze and convert it into an effective interaction modality. This new interaction modality is expected to benefit users who have impaired mobility in their daily living as well as able-bodied users who need an additional hand in general working scenarios.
TABLE OF CONTENTS

ABSTRACT ................................................................. iii
LIST OF FIGURES .......................................................... xii
LIST OF TABLES ............................................................ xix
ACKNOWLEDGMENTS ......................................................... xxi
DEDICATION ................................................................. xxii
CHAPTER 1 INTRODUCTION ................................................. 1

CHAPTER 2 A NOVEL 3D GAZE ESTIMATION METHOD WITH
THEORETICAL ANALYSIS AND EXPERIMENTAL EVALUATION . . 6

2.1 Abstract ................................................................. 6
2.2 Introduction .............................................................. 6
2.3 Related Work ............................................................ 9
2.4 Gaze Vector Method for 3D Gaze Estimation ......................... 12
  2.4.1 Gaze Vector Method for 3D Gaze Estimation .................... 12
  2.4.2 Characteristics of Gaze Vector Method ......................... 14
2.5 Theoretical Analysis .................................................. 16
  2.5.1 Simulation Setup .................................................. 16
  2.5.2 Theoretical Analysis of Gaze Vector Estimation ............... 17
  2.5.3 Theoretical Analysis of 3D Gaze Estimation ................... 22
2.6 Experimental Validation .............................................. 28
  2.6.1 3D Gaze Tracking System ....................................... 29
3.4.4 Estimation based on Neural Networks ........................................... 56
3.5 Experiments ................................................................................. 56
3.6 Experiment Results and Discussion ............................................. 57
  3.6.1 Overall Error ........................................................................... 57
  3.6.2 Comparison of Gaze Vector Method with Visual Axes Intersection Method ................................................................. 59
  3.6.3 Gaze Vector Error Reduction .................................................. 59
  3.6.4 Various Settings of the Neural Network ................................. 60
  3.6.5 Comparison with Existing 3D Gaze Tracking Systems ............ 61
3.7 Conclusion ................................................................................... 62
3.8 Acknowledgment .......................................................................... 64
3.9 Appendix: Computation of Gaze Vector From Visual Axes .......... 64

CHAPTER 4 3D-GAZE-BASED ROBOTIC GRASPING THROUGH MIMICKING HUMAN VISUOMOTOR FUNCTION FOR PEOPLE WITH MOTION IMPAIRMENTS ........................................... 65

  4.1 Abstract .................................................................................... 65
  4.2 Introduction ................................................................................ 66
  4.3 Related Work ............................................................................. 69
    4.3.1 3D Gaze Tracking in Real Environments .............................. 69
    4.3.2 Human-Robot Interaction based on 2D Gaze ....................... 71
    4.3.3 Eye-Hand Coordination in Object Grasping ....................... 72
  4.4 Methods .................................................................................... 73
    4.4.1 Accurate 3D Gaze Tracking in a Real Environment ............ 74
    4.4.2 3D Gaze Interpretation - Mimicking the Human Visuomotor Function during Grasping .................................................. 77
5.5.1 Gaze Data Filter .................................................. 114
5.5.2 Experiment Results .............................................. 115
5.5.3 Questionnaire Evaluation ...................................... 116
  5.5.3.1 Overall System Usability .................................. 116
  5.5.3.2 User Experience Over Repetitive Tests ................ 116
  5.5.3.3 Comparison of User Experience between Two Methods 117
5.6 Discussion ......................................................... 119
5.7 Conclusion ......................................................... 120

CHAPTER 6 IMPLICIT INTENTION COMMUNICATION IN HUMAN-ROBOT
INTERACTION THROUGH VISUAL BEHAVIOR STUDIES ........ 121

6.1 Abstract .......................................................... 121
6.2 Introduction ...................................................... 122
6.3 Related Work ...................................................... 124
6.4 Method — Gaze-based Implicit Intention Communication Framework .... 126
  6.4.1 Attention-based Object Recognition ....................... 126
    6.4.1.1 Intentional Gaze Recognition for Attention Detection .... 127
    6.4.1.2 Equivalent Representation of Visual Attention ........... 127
    6.4.1.3 Attention-based Object Detection and Recognition ....... 128
  6.4.2 Human Intention Inference .................................. 128
    6.4.2.1 Knowledge of Human Intention .......................... 128
    6.4.2.2 Human Intention Inference using Gaze Cues ............. 130
  6.4.3 Gaze-based Intention Communication ....................... 131
    6.4.3.1 Trigger and Complete the Gaze-based Communication .... 131
CHAPTER 7 CONCLUSION AND FUTURE WORK ........................................ 153

7.1 Conclusion .................................................................................. 153

7.2 Future Work ................................................................................ 153

7.2.1 Multimodal Interaction ................................................................. 154

7.2.2 Validation on Users with Special Needs .......................................... 154

7.2.3 Gaze-Enlightened Robot Task Knowledge ...................................... 155

REFERENCES CITED ..................................................................... 156
LIST OF FIGURES

Figure 1.1 Demonstration of 3D-gaze-based human-robot interaction. AP# is the abbreviation of Action Plan #, indicating how to perform a task.  

Figure 1.2 Overview of the dissertation with technologies developed shown as blocks. P# indicates Procedure #; AP# indicates Action Plan #; and O# indicates Object #.  

Figure 2.1 Demonstration of the gaze vector method, which decouples the estimation of a 3D gaze point into the estimation of the gaze vector and the gaze distance along the gaze vector, respectively.  

Figure 2.2 The gaze vector method can accurately estimate the gaze vector even when the estimated visual axes have large errors. The estimated visual axes and gaze vector are indicated with dashed lines.  

Figure 2.3 The gaze vector method can accurately estimate the gaze vector even when the estimated visual axes have large errors. The estimated visual axes and gaze vector are indicated with dashed lines.  

Figure 2.4 The simulated visual axes $V_L$ with a certain estimation error forms a cone surface around the ideal visual axis $V_i$.  

Figure 2.5 Point tests at $[0, 0, 40]$ cm, $[0, 0, 70]$ cm, and $[0, 0, 100]$ cm respectively. (a) The pictorial view of all possible intersections of two visual axes. Each marker represents a possible intersection. All the possible gaze vectors are divided into three categories according to their angular errors, and their corresponding markers are marked with different colors. (b)-(d) The top, side, and left views of these markers at each point. (e)-(m) The top, side, and left views of each error category at a single point $[0, 0, 70]$ cm.  

Figure 2.6 The error distribution of the estimated gaze vector in point test. The base error of the visual axis is $0.5^\circ$. Each point test results in 400 possible gaze vectors where their errors are ordered from low to high. Three points along the depth direction have the same average error of $0.318^\circ$.  

Figure 2.7 The error of the estimated gaze vector in plane tests.  

xii
Figure 2.8 Space test of the gaze vector estimation with various errors of visual axis, which is annotated. (a) The angular error of the gaze vector when visual targets are on different depth. (b) The error reduction ratio of the gaze vector’s angular error when visual targets are on different depths. The potential 3D gaze error is plotted on the left axis with black lines, and the error reduction ratio is plotted on the right axis with color lines. Different markers are used to annotate different visual axis errors.

Figure 2.9 The distribution of the estimated 3D gaze using the gaze vector method (green squares) and using the visual axes intersection method (purple squares), respectively, at point test of [0, 0, 70] cm. Note that the Z-axis has a very different scale from the X-axis and Y-axis.

Figure 2.10 Results from plane tests of the visual axes intersection method (VAIM) and the gaze vector method (GVM).

Figure 2.11 Results from space tests of the visual axes intersection method (VAIM, red lines) and gaze vector method (GVM, blue lines). Different error conditions of the visual axes are annotated with different markers.

Figure 2.12 Space test results of gaze vector method when gaze distance have various errors. The various gaze distance’s errors are annotated with different markers. The 3D gaze errors are shown as the green lines, and the error reduction ratios are blue lines.

Figure 2.13 The binocular eye tracker for 3D gaze tracking.

Figure 2.14 The extended space tests of the visual axes intersection method for 3D gaze. In this simulation, the left and right visual axes have an error of 0.5°, 0.7°, 0.9°, 1.1°, and 1.3°, respectively.

Figure 2.15 The demonstration of the gaze error overshoot when estimated visual axes intersect at the backward direction. $EG_L$ and $EG_R$ are left and right eye gaze angles, respectively. $e_L$ and $e_R$ are angular error of the estimated left visual axis and right visual axis. $\vec{V}_L$ and $\vec{V}_R$ are the estimated left visual axis and right visual axis, respectively, which consist of errors. $P$ is the location of the visual target.

Figure 3.1 The binocular eye tracker with coordinate system.

Figure 3.2 Steps of pupil identification from the image: (a) raw color image, (b) grayscale image, (c) binary image, (d) binary image after noise elimination, and (e) original image with the identified pupil.
Figure 3.3 Illustration of the gaze vector method with the decoupled gaze vector \( \mathbf{v}_g \) and gaze distance \( l_g \). \( \mathbf{e}_L \) and \( \mathbf{e}_R \) are the position of left and right eyes, \( \mathbf{v}_L \) and \( \mathbf{v}_R \) are the left and right visual axes, \( \mathbf{i}_L \) and \( \mathbf{i}_R \) are the intersection of the two visual axes with their common normal. The middle point of \( \mathbf{i}_L \) and \( \mathbf{i}_R \) is considered as the intersection of two visual axes. ........................................ 43

Figure 3.4 The gaze vector method can accurately estimate the gaze vector even when the visual axes have large errors. The estimated visual axes and gaze vector are indicated with dashed lines. ........................................ 45

Figure 3.5 Estimating 3D gaze by combining the gaze vector and the gaze distance in gaze vector method. ........................................ 46

Figure 3.6 X and Y components of the pupil location when one subject is viewing the 4 \( \times \) 4 grid of visual stimuli on a plane. ........................................ 47

Figure 3.7 Demonstration of pupil rotation when one subject is viewing four visual stimuli that lays in a row. (a)-(d) correspond to the four stimuli. ....... 48

Figure 3.8 Plot of pupil rotation angle when one subject is viewing a set of visual stimuli organized as a 4 \( \times \) 4 grid. ........................................ 48

Figure 3.9 Plot of pupil size when one subject is viewing the 4 \( \times \) 4 grid of visual stimuli on a plane. ........................................ 49

Figure 3.10 Plot of axis ratio of the pupil ellipse when one subject is viewing the 4 \( \times \) 4 grid of visual stimuli on a plane. The data in this plot has been centered. ........................................ 49

Figure 3.11 Demonstration of pupil distance changes when the visual stimuli are further away. Two green dashed lines are reference and aligned vertically with the pupil locations when the visual stimulus is at 35 cm away. The image are flipped, which makes it like the pupil distance is declining when the visual stimulus is further away. ................. 50

Figure 3.12 Plot of pupil distance with respect to the number of the collected data. The visual stimuli are along three different gaze vectors, left axis, central axis, and right axis. ........................................ 51

Figure 3.13 Illustration of visual stimuli that are lying in vertical and horizontal lines. ........................................ 52
Figure 3.14  Plot of pupil position when a user looked at the horizontal and vertical visual stimuli, which were placed at different depths. (a) is at depth of 50 cm; (b) is at depth of 80 cm; and (c) is at depth of 110 cm.

Figure 3.15  Simulation results when using different numbers of calibration point to capture the pupil position pattern. Plot (a), (b), and (c) are results when there are two, three, and four calibration points, respectively.

Figure 3.16  Plot of pupil distance with respect to the distance of the visual stimulus. The visual stimuli are along three different gaze vectors, left axis, central axis, and right axis.

Figure 3.17  Illustration of the experimental setup for 3D gaze tracking in a real environment (a) and the layout of visual stimuli used for calibration (b).

Figure 4.1  Architecture of the 3D-gaze-based human-robot interaction system.

Figure 4.2  The binocular eye tracker with coordinate definition.

Figure 4.3  The illustration of the gaze vector method with the decoupled gaze vector and gaze distance. In the illustration, $e_L$ and $e_R$ are the positions of left and right eyes, $v_L$ and $v_R$ are the left and right visual axes, $i_L$ and $i_R$ are the intersections of the two visual axes with their common normal, and the middle point of $i_L$ and $i_R$ is considered as the intersection of two visual axes.

Figure 4.4  Adoption of human eye-hand coordination model in grasping to the robot.

Figure 4.5  Demonstration of the 3D gaze calibration procedure. The red dots are those 4×4 defined visual targets that the participants need to concentrate on. This plane was placed on four different depths ranging from 60 cm to 100 cm.

Figure 4.6  Object layouts and experiment setup for building the grasping model.

Figure 4.7  Experiment setup of using 3D gaze to command an assistive robot for grasping.

Figure 4.8  Demonstration of successful estimation of object center and pose using 3D gaze. The big green dot is the object center and the blue arrow that passes through this dot is the object pose. Four small red dots represent the four fixations generated by the subject.
Figure 4.9  Raw grasping data of the contact point and the hand approaching angle from one of the participants. $AiDj$ corresponds to the 12 testing locations, where $i \in [1, 4]$, and $j \in [1, 3]$.

Figure 4.10  The visuomotor model for grasping. The top plot is the GMM from the raw grasping data; and the bottom is the generated GRM.

Figure 4.11  Grasping planning using the visuomotor grasping model with a virtual robot at four grasping points.

Figure 4.12  The distribution of the estimated 3D gaze using the gaze vector method (green squares) and using the visual axes intersection method (purple squares), respectively, at visual target $[0, 0, 70]$ cm (blue dot). Note that the Z-axis has a different scale from the X-axis and Y-axis.

Figure 4.13  Theoretical error distributions of object pose estimation when the 3D gaze has different levels of error, which are indicated by lines with different colors. The black dashed line is the relationship between the 3D gaze error (right Y-axis) and the expected error of the object pose.

Figure 4.14  3D gaze in a real environment for assistive robot control. Our goal is to achieve a novel paradigm that a user can intuitively command an assistive robot for certain services by naturally gazing at the target object in the real world.

Figure 5.1  The flowchart of the attention-aware gaze-guided robotic laparoscope system.

Figure 5.2  The setup of the gaze-guided automated robotic laparoscope system. The image of the artificial muscle in the surgery simulator is projected on a monitor through the robotic laparoscope.

Figure 5.3  Two kinds of historical eye-gaze behaviors in a surgical operation.

Figure 5.4  The overall procedure of the fuzzy inference engine.

Figure 5.5  Fuzzy logic membership functions.

Figure 5.6  AScore and SSscore.

Figure 5.7  The experiment setup with a virtual simulator.
Figure 5.8  Comparison between the raw gaze data (left) with refined gaze data (right). The small squares are the gaze points. The big circles are the targets that the subject gazed upon on the monitor. The lines are the trajectory of the eye movements. 115

Figure 5.9  Comparison of the response time in the dwell-time method and the fuzzy inference method. 116

Figure 5.10  The summary of system usability score in two different methods. The higher score means that the system has a better usability. ASoFI: average score of the fuzzy inference method. ASoDT: average score of the dwell-time method. 117

Figure 5.11  The summary of user experience over repetitive tests for each method. The average score of the dwell-time method is 38.5, which is comparable to 38 for the fuzzy inference method. 118

Figure 5.12  User experience comparison of the fuzzy inference method over the dwell-time method. The total score ranges between 0 and 40 (the value closer to 0 means that the dwell-time method is superior to the fuzzy inference method and the value closer to 40 means that the fuzzy inference method is better than the dwell-time method). 118

Figure 6.1  Detection of intentional gaze using a SVM classifier. 127

Figure 6.2  MST clustering for visual attention representation. Gaze point cluster is presented by a dashed-line circle with its center shown as a red dot. The visual attention is the gaze cluster with the maximal amount of gaze points. The gaze path is shown as the arrow direction. The clustered gaze points are shown as green and the rest are shown as black. 128

Figure 6.3  A Naive Bayes model for representing object-intention knowledge database. $O_j$ is the $j^{th}$ object and $I_i$ is the $i^{th}$ intention. Each linkage indicates the conditional appearance probability ($p_{ji}$) of an object to one type of intention. A longer link means a lower probability, and vice versa. 129

Figure 6.4  Gaze-based intention communication framework illustration. 132

Figure 6.5  The artificial kitchen image with labelled objects. 134

Figure 6.6  Experiment setup for gaze-based intention communication. 135
Figure 6.7  Heat map of gaze patterns during intentional gaze and intention-free gaze in a kitchen environment. The height of each peak is proportional to the duration of gaze dwell time. .................................................. 137

Figure 6.8  Plot of pupil size samples from the subjects. At transition point from the intention-free gaze to intentional gaze, the pupil size increases. The horizontal axis is time, and about 3 seconds of pupil data was trimmed in each sample. .............................................................. 137

Figure 6.9  The performance of the SVM attention classifier in the experiment. The performance is evaluated by correct detection rate $R_{crct}$ shown as a white bar pointing up and fake detection rate $R_{fk}$ shown as a gray bar pointing down. .......................................................... 139

Figure 6.10 Correlation plot between intentions and objects. The correlations are clustered into high, medium and low strength levels and are notated using dots with different sizes. A larger dot represents a greater correlation strength. ..................................................... 140

Figure 6.11  Intention inference correctness rate with different knowledge fusion strategies illustrated as 6.2-6.4. The white bar is the correctness rate of the intention inference using recorded objects and the gray bars are those using the SVM detected objects. ............................... 141

Figure 6.12 Intention inference confusion matrix with linearly fused knowledge. For intention "Prepare a cup of coffee", it is correctly inferred in 92.9% cases. And the overall correctness rate is 75.0%. ...................... 143

Figure 6.13 Questionnaire score summary of gaze (blue) and mouse (red) modalities using box plot. ................................................................. 144

Figure 6.14 Summary of the first object came into subjects’ mind when they had a type of intention to express. .................................................. 146

Figure 6.15 The order of $O_{mind}$ being gazed by subjects when they expressed a type of intention. ............................................................... 147

Figure 6.16 Summary of the order of each dominant object being looked at in each type of intention. ......................................................... 148
Table 2.1 Error summarization of the point tests using the gaze vector method (GVM) and the visual axes intersection method (VAIM). P1 is [0, 0, 40] cm; P2 is [0, 0, 70] cm; P3 is [0, 0, 100] cm. Ave stands for average. Max stands for maximal. Min stands for minimal. e stands for error.

Table 2.2 Average error summarization of the estimated left and right visual axes and the computed gaze vector.

Table 2.3 Error summarization of 3D gaze estimation using the gaze vector method and the visual axes intersection method. Max: maximal, Min: minimal, Ave: average, and e: error.

Table 3.1 Eye tracking resolution of the presented eye tracker at depth 50 cm, 80 cm, and 110 cm. The unit of the eye tracking resolution is pixel/cm.

Table 3.2 Error summarization of 3D gaze tracking in a real environment using gaze vector method. The error with the standard deviation along each axis and the overall error are summarized for the entire test space and for each test depth. The unit of the gaze error is cm and the gaze vector error is in degree.

Table 3.3 Error comparison between gaze vector method and the visual axes intersection method for 3D gaze estimation. Max: maximal error, Ave: average error, Min: minimal error.

Table 3.4 Error summarization of the estimated visual axes, \( v_L \) and \( v_R \), and the computed gaze vector, \( v_g \).

Table 3.5 Error summarization when the visual axes estimator has various settings in Neural Networks.

Table 3.6 Error summarization when the gaze distance estimator has various settings in Neural Networks.

Table 3.7 Comparison of our 3D gaze tracking system using gaze vector method with existing systems.

Table 4.1 CARTESIAN ERRORS OF 3D GAZE ESTIMATION USING THE PRESENTED SYSTEM. MAX: MAXIMAL, MIN: MINIMAL, AND AVE: AVERAGE.
ACKNOWLEDGMENTS

First, I would like to thank my advisor, Dr. Xiaoli Zhang, for her excellent guidance and support not only on my research but also on my development in all professional aspects. Her enthusiasm and dedication have greatly influenced me in my work. I thank her for allowing me to participate in so many different research projects that have broadened my vision and strengthened my capabilities for considering and solving a complex problem.

I would like to thank Dr. Nelson Carl, Dr. Fernando J. Kim, Dr. Rodrigo Donalisio da Silva, Diedra Gustafson, Wilson R. Molina, Jeremy D. Web, Natalia Kalin, and Ian Coberly for their contributions to my work. I also thank the funding agencies of my work: NSF and Colorado School of Mines. I thank all these participants of my project for allowing me to validate and evaluate my ideas and work.

I would like to thank Dr. Hao Zhang, Dr. William Hoff, Dr. John Steele, and Dr. Hua Wang for their help in my study and work.

I would like to thank all my friends for their kind friendship and support of my life and work while accomplishing my doctoral degree.

I am highly grateful to my wonderful family for their unconditional love and care throughout my life.
To my lovely family.
CHAPTER 1
INTRODUCTION

In recent decades, research in robotics has transitioned to a new scenario in which the robot is not repeating fixed motions or under complete control of a human user but rather functions as an independent, intelligent agent that provides timely and appropriate assistance to the human user in various tasks. In this new scenario, a new interaction relationship has been developed between the human and the robot, and applications related to it have been termed human-centered robotics.

In the domain of human-centered robotics, robots work closely with the human user, not only from the aspect of physical distance, but also from the aspect of relation. The robot may work around a human user with direct or indirect contact. In addition, they may work together on the same task or separate task components for the same goal. With more and more robots that work closely with the human, how the human and robot should interact with each other becomes a question that researchers have to investigate and scientifically answer.

In human–robot interaction, one fundamental question is ’How should or can users express themselves so that the robot can understand what the users want?’ In order to express themselves, the user needs to select one interaction modality. Meanwhile, the robot should be capable of sensing and interpreting signals from the selected modality. To facilitate the expression, various interaction modalities have been investigated, including speech, facial expression, body gesture, electromyography (EMG) signals from muscles, and electroencephalogram (EEG) signals from the brain. Some of these interaction modalities are commonly used by a person to ’read’ another person in human–human interaction. One primary motivation of this investigation is to achieve intuitive and effective interaction. Substantial research has been conducted on these modalities to investigate the development and us-
ability. However, due to the limitations of the current technology level or the modality self, obstacles have been posed during the interaction on either the human utilization or the robot interpretation. Novel interaction modalities and interpretation technologies for intuitive and effective human-robot interaction are greatly needed.

In this dissertation, three-dimensional (3D) gaze is introduced as a human-robot interaction modality, which measures where a person is looking in the real 3D world. It explicitly measures 3D Cartesian locations of 3D gaze points in a real environment. Employing 3D gaze as an interaction modality can bring in multiple advantages, which makes 3D gaze a natural and effective means for HRI. Looking at an object or a set of objects reflects a person’s high-level desires. For example, when a person is thirsty and wants to drink, he/she will naturally look at drinking-related objects, such as a bottle of water or a water fountain. Naturally, this object being looked at will be the manipulation target. This natural link between human gaze and human mind makes gaze an intuitive modality to express what a person wants. Moreover, from 3D gaze, the location information of the object being looked at can be retrieved, which is an essential piece of information for any object operation. Looking at an object is a natural and effortless behavior, which makes 3D gaze modality natural and effortless for utilization. Even persons with motion disabilities remains the capability of looking at particular objects. This remaining functionality could make gaze modalities the last potential means for a disabled user to interact with robots or even other humans. Gaze has been naturally used in human-human interaction; for example, one uses his/her gaze to guide another to an object of interest in order to build joint attention, which makes gaze easily adaptable in human-robot interaction.

Although 3D gaze has great potential for HRI, research on 3D gaze tracking is rare. Gaze tracking has a long history, and commercial gaze tracking systems are available on the market. However, the available gaze tracking technologies are mainly for two-dimensional (2D) environments, which tracks where a person is looking on a 2D computer screen or image. 2D gaze tracking may be good for 2D human-computer interaction but cannot provide
sufficient functionalities for HRI. To interact with robots in the physical 3D world, 3D gaze tracking technology in real environments is highly desirable. Another critical problem in gaze-based interaction is the simple interpretation technique. Currently, gaze has only been used as a command trigger to activate a predefined event when gaze dwells on an onscreen button for a while, which is like the button click action derived from a mouse. It has been used to activate a step motion of a robot along a certain direction or about a certain joint. Interpreting gaze as a simple action trigger results in low task efficiency and high user workload. Thus, accurate tracking and effective interpretation technologies of 3D gaze are greatly needed for gaze-based HRI.

In this dissertation, work that focuses on promoting the gaze modality for intuitive and effective HRI is presented. The goal is to achieve a novel interaction modality with which a user can intuitively express what tasks he/she wants the robot to do by naturally looking at the object of interest in the real world. One example of 3D-gaze-based HRI can be shown in Fig. 1. When a user is looking at a coffee pot and thinking about drinking some coffee, the robot can sense where and what the user is looking at and infer what he/she wants. Moreover, an execution plan can be determined based on the hints from user’s gaze to facilitate the robotic operation. This new interaction modality is expected to benefit users who have impaired mobility in their daily living and able-bodied users who need an additional hand in general working scenarios.

In working towards the goal, research has concentrated on the following two areas: 1) technologies for accurate 3D gaze tracking in real environments and 2) technologies for effective 3D gaze interpretation. Figure 2 is an overview of this dissertation, which demonstrates the technologies that are developed in this dissertation. Those technologies have been separately developed in five chapters, and the corresponding chapter numbers are shown with light blue circles. A binocular eye tracking system is developed to track both eyes’ movements, and based on visualization behavior studies, representative eye features are selected to qualify the eye movement. With a novel gaze vector method, 3D gaze is estimated with
The user wants to drink some coffee, and I should prepare it.

- **Target**: Kettle @ location XX, ...
- **Procedures**: Grip(AP1) →
  Four(AP2) → ...

**Interpretation**

Figure 1.1: Demonstration of 3D-gaze-based human-robot interaction. AP# is the abbreviation of Action Plan #, indicating how to perform a task.

A high degree of accuracy using the selected eye features as inputs. The 3D gaze is passed through a particularly designed adaptive sliding window filter to remove the noise caused by unconscious eye movements, blinks, or visual distractions. Then, the visual attention, a location where a user is looking with the intention of performing manipulation, is detected using implicit methods based upon natural visualization behaviors, which can provide faster responses and better user experiences. The object being looked at are detected to infer the user’s high-level intention. Meanwhile, the object location can be determined and the pose can be estimated from the 3D gaze points. A human visuomotor model is built to describe the correlation between human gaze and hand grasping. This model is adopted by an assistive robotic arm to function as a third arm for the human user.
Figure 1.2: Overview of the dissertation with technologies developed shown as blocks. $P\#$ indicates Procedure #; $AP\#$ indicates Action Plan #; and $O\#$ indicates Object #.
CHAPTER 2
A NOVEL 3D GAZE ESTIMATION METHOD WITH THEORETICAL ANALYSIS AND EXPERIMENTAL EVALUATION

A paper to be submitted to Sensors
Songpo Li\(^1\) and Xiaoli Zhang\(^2\)

2.1 Abstract

Three-Dimensional (3D) gaze tracking is to track where a person is looking in a 3D environment and report the 3D coordinates of this location. It has started to draw researchers’ attention in the last decade due to its broad applications of human natural behavior studies and assistive technologies, especially for home healthcare of the elderly and disabled people. However, currently there is not an effective method that can accurately estimate the 3D gaze. Moreover, the lack of a means to theoretically analyze a given estimation method results in that the performance has to be experimentally evaluated by implementing the estimation method into a physical eye tracking system. To overcome these challenges, in this chapter, a novel method, gaze vector method, is developed, which can accurately estimate a person’s 3D gaze in a relatively large workspace. In addition, theoretical analysis is introduced as a means to reveal insights of a given estimation method and evaluate the method before physical implementation.

Index Terms: 3D gaze tracking, gaze vector, visual axis, theoretical analysis

2.2 Introduction

Gaze tracking to tell where a person is looking has been an active research area for a very long time [1–3]. It has been widely used as a tool to support research in neurobiology

\(^1\)Primary researcher and author, graduate student, Department of Mechanical Engineering, Colorado School of Mines.
\(^2\)Author for correspondence, Assistant Professor, Department of Mechanical Engineering, Colorado School of Mines.
[4, 5], psychology [6, 7], computer science [8, 9], human factors [10, 11], etc. Moreover, it has also been adopted as an intuitive means to interact with computers [8] and robots [12–16], which functions as an alternative to or extension of human hands. It is very valuable for the disabled and elderly people allowing them to command an assistive robot by simply looking at an object that needs to be operated. However, the tracking technology is only available in two-dimensional (2D) environments which tracks where a person looks on a 2D display surface. Only in last decade has effort been put forth into 3D gaze tracking, and currently research focus is still on the development of accurate and reliable methods for 3D gaze estimation. 3D gaze tracking is to track where a person is looking in a 3D environment and report the 3D coordinates of this location. This 3D environment could be virtual, like stereo display, or real as the real world. The need for 3D gaze tracking has become apparent as new requirements arise in modern technologies and research which cannot be satisfied by traditional 2D gaze tracking. Stereo image technology has become particularly favorable in computer science where the development of 3D gaze tracking is needed to investigate how humans perceive in and interact with virtual 3D environments. Because humans live and interact in a physical 3D world, 3D gaze tracking is preferable in order to conduct the studies of natural human visual behaviors. Furthermore, the gaze’s 3D coordinates will be a useful modality for designing the next generation of human-machine interaction paradigm.

However, 3D gaze tracking is still underdeveloped, and the accuracy and the workspace achieved in current research are very limited, especially in the real 3D environment. Currently, the employed methods that estimate the 3D gaze are mainly adopted from existing 2D gaze tracking methods or proposed from researchers’ subjective understanding about gaze, which have not been fully validated. Common 2D gaze estimation methods include regression method for direct mapping [17, 18] and the geometrical method for indirect construction [1, 19]. The regression method relies on a mapping relationship between the gaze location on a tracking surface (could be a monitor screen or image) and the eye movement features. This mapping relationship is built through a calibration procedure. Commonly,
this relationship can be built as polynomial functions [20] or more complex models using Neural Networks [18]. The geometrical method solves the gaze location as the intersection point of a visual axis with a known display screen. The visual axis is defined as a vector that emits from an eye pointing to where the user is looking. The visual axis can be reconstructed based on an anatomic eye model or estimated using a regression-based mapping relationship. Researchers attempted to directly adopt the aforementioned methods in 3D gaze tracking, however, the achieved accuracy was poor and the workspace was small.

Another issue that arises in 3D gaze tracking research is the lack of a means to judicially evaluate a 3D gaze estimation method. Currently, the only way to evaluate a 3D gaze estimation method is to integrate it with a physical eye tracking system and test the overall performance. During this procedure, the performance is more likely to be affected by other components in the system such as the hardware limitations, the implementation algorithm, the testing subjects, or the testing procedure. Thus, it is not fair and rigorous to access a method based on its performance in a certain physical system. A means that can evaluate an estimation method without physical interference is greatly needed. It will not only judicially evaluate the merits of the method but also reveal some underlying characteristics of the method, which can give insights of the potential workspace, the potential error, the error distribution and change trend, requirements or limitation of its application, or even more.

In this chapter, a novel method for 3D gaze estimation is presented. This new method is named as gaze vector method (GVM), which constructs the 3D gaze in a way that is different from any previous methods. Particularly, our contributions in this chapter include:

1. A novel method, gaze vector method, is developed for 3D gaze estimation in 3D environments. The gaze vector method has three valuable characteristics that can reduce estimation errors and perform more accurate estimation. The gaze vector method is validated thoroughly with theoretical analysis and practical tests to demonstrate its outstanding capability for accurate 3D gaze estimation in a relatively large workspace.
2. Theoretical analysis is introduced as a means to evaluate a given 3D gaze estimation method without interferences from the physical system, the testing subjects, or the testing procedure. It is a generic analysis method that can be applied to most of the geometrical gaze estimation methods. In this chapter, theoretical analysis is carried out to evaluate the gaze vector method and visual axes intersection method (VAIM, the commonly used method for 3D gaze estimation that is used as a base line for comparison in this chapter), respectively. Moreover, insights of the methods are revealed from the analysis, including the potential errors, the error distribution and change trend, and the potential workspace.

3. The severe error of the visual axes intersection method systematically revealed from theoretical analysis and practical tests showed that it is an impractical method for 3D gaze estimation. We demonstrate that the visual axes intersection method is very sensitive to the error of visual axes, resulting in a large error of 3D gaze even in a small workspace, and the error of 3D gaze gets worse when the visual target is further away. In contrast, the gaze vector method outperformed the visual axes intersection method in error tolerance for the visual axis and the gaze distance in a relatively large workspace.

2.3 Related Work

There is very limited research on 3D gaze tracking in virtual or real 3D environments. The few available methods had poor performance or lack of solid validation. In this review, to best of the authors’ knowledge, all works that can explicitly estimate the 3D coordinates of the gaze location are covered. We exclude works that can estimate or compute only the gaze’s direction but not its 3D coordinates in a 3D environment like [13, 21–25].

The methods that have been used for 3D gaze estimation can be categorized into three kinds: direct mapping method, visual axes intersection method, and depth plane method. Each method will be introduced in detail.
In the direct mapping method, the estimation of 3D gaze is performed with a mapping function, which is a function of a set of features of eye movements. In other words, eye movements are directly mapped to the gaze location using the mapping function. This method is comparable to the regression method in 2D gaze tracking. To build this mapping function, a user calibration is needed, in which the user needs to look at a set of visual targets with known positions and the eye features at each visual target are recorded. One research group from Germany has used this method to estimate where a user is looking in a stereo display [26, 27]. Their mapping relationship was created based on a Neural Network algorithm, and an average error of 2.79 cm was achieved in a virtual 3D environment that was 22 cm deep. Though they attempted to apply the same implementation in a real 3D environment in 2012, there was no quantitative data reported [28]. In [29], one UK research group used a polynomial mapping function to map the eye location to the gaze location in a virtual 3D environment. However, no quantitative data about the workspace or accuracy were available in their paper. Compared with 2D environments, eye movements for visualizing a 3D environment are much more complex. Directly learning the mapping relationship becomes extremely difficult for 3D gaze estimation, especially for achieving a high degree of accuracy in a large workspace.

The visual axes intersection method considers two eyes function as a stereo camera system. It assumes that, when a person visually concentrates on an object or a location in the 3D environment, the visual axes from the left eye and right eye intersect at that location. Thus the intersection point of the left visual axis with the right visual axis represents the location of the 3D gaze. However, in reality, two estimated visual axes may not intersect due to the error of the visual axes. Thus, the middle point of two visual axes’ common normal is defined as the location of the 3D gaze, which has the shortest squared distance to both visual axes. The visual axis of each eye can be estimated using a mapping function built through a calibration process or reconstructed using the anatomic eye model. The visual axes intersection method is the most popular method used to estimate the 3D gaze.
and has been used in both visual and real 3D environments. The German research group [26, 27] and another group from Japan [30, 31] have both implemented this method in the stereo virtual 3D environment. Errors of 6.18 cm and 4.53 cm were achieved, respectively, in a virtual environment that was 22 cm deep. In 2009, Hennessey and Lawrence [32] first investigated the feasibility of 3D gaze tracking in a real environment using the visual axes intersection method. An average accuracy of 3.93 cm was achieved across a 30 cm × 23 cm × 25 cm (width×height×depth) workspace. Later in 2012, Abbott and Faisal [33] reported their implementation of the visual axes intersection method in a testing depth range from 54 cm to 108 cm within a 47 cm wide and 27 cm high workspace. They achieved an average error of 5.8 cm. Though the intersection method seems reasonable for 3D gaze estimation, it is very sensitive to errors on the visual axes as the errors severely propagate during the intersection procedure. This method is used as the baseline for comparison in the theoretical analysis and experimental tests.

The depth plane method solves the gaze location in a way similar to the geometrical method in 2D gaze tracking, which intersects a visual axis with the display screen. In this method, the visual target is assumed to be on a virtual plane that is facing the user straight-on, and the vertical distance from the user to the virtual plane is defined as the depth from the user to the visual target. The depth plane method has two steps: 1) to estimate the visual axis and 2) to estimate the depth plane, which is based on two separate mapping relationships. One research group of Kwon from Korea used this method to estimate users’ gaze in a stereo display in which the depth plane was linearly related to a single eye feature: the distance between two pupils [34–37]. However, this method was poorly validated in their system, and there was a lack of results in their chapter. In 2012, Lee et al. [38] attempted this method in a real environment within a pyramid-shaped workspace whose base was a 10 cm × 10 cm square, the height was 50 cm, and the apex was at one eye’s center. Only one single eye was tracked in their system. However, even in a setup where the testing points were the same as the calibration points, the experimental results showed low accuracy. The
average error was 4.59 cm along the depth direction and there were large deviations in other
directions too.

In above research, eye movements are the only inputs to the estimation methods. Some
researchers used extra devices to assist the measurement of gaze in 3D environments instead
of performing the estimation using only eye information. However, due to the additional
hardware, their systems were bulky and cumbersome. Additionally, the system’s complexities
of setup and usage are increased. In [39], an RGB-D camera was used to project 2D gaze into
the 3D environment, and the 2D gaze was estimated using a traditional 2D gaze tracking
method. The same setup has also been used by another research group of Paletta [40–
42]. Stereovision technology has also been used to reconstruct the 3D gaze as it build the
3D environment using stereo cameras [43, 44]. The RGB-D camera-assisted system and
the stereovision-assisted system are variants of the depth plane method and visual axes
intersection method, respectively. In addition, the infrared-light-based (IR-based) motion
tracking system has also been used to cooperate the gaze tracking method to find out which
IR marker’s adjacent area was visually attended to [45]. However, this system can only track
the locations labeled with IR markers.

2.4 Gaze Vector Method for 3D Gaze Estimation

2.4.1 Gaze Vector Method for 3D Gaze Estimation

The gaze vector method decouples the 3D gaze estimation into the estimation of the gaze
vector and then the estimation of the gaze distance along the gaze vector as shown in Fig.
2.1. The gaze vector is defined as the vector emitting from the middle point between two
eyes to the visual target. The gaze distance is the absolute distance from the eyes’ middle
point to the visual target. From the captured eye images, the left visual axis and right visual
axis, $\mathbf{V}_L$ and $\mathbf{V}_R$, are estimated through mapping relationships $M_{va,L}$ and $M_{va,R}$ using
(2.1) and (2.2), respectively, where $x_L$ and $x_R$ are sets of eye movement features from the left
or right eyes. Instead of directly using their intersection to estimate the 3D gaze point, the
gaze vector method uses their intersection to compute the gaze vector $\mathbf{V}_{gz}$ through (2.3) and
\( \overline{C} \) is the intersection point of the left visual axis with the right visual axis, which is the middle point of two axes’ common normal, and the computation process, \( I(\overline{V}_L, \overline{V}_R) \), is explained in appendix. Then the computed gaze vector and selected eye features, \( x'_L \) and \( x'_R \), which have strong indications to the distance of the visual target, are used to estimate the scalar gaze distance \( D_{gz} \) through \( M_D(\overline{V}_{gz}, x'_L, x'_R) \) as (2.5). Thus, the Cartesian coordinates of the 3D gaze point, \( \overline{G}_3 \), can be calculated as in (2.6). A calibration process is needed to generate the mapping relationships \( M_{va,L}, M_{va,R}, \) and \( M_D \).

\[ \overline{V}_L = M_{va,L}(x_L) \quad (2.1) \]

\[ \overline{V}_R = M_{va,R}(x_R) \quad (2.2) \]

\[ \overline{C} = I(\overline{V}_L, \overline{V}_R) \quad (2.3) \]

\[ \overline{V}_{gz} = \frac{\overline{C}}{|\overline{C}|} \quad (2.4) \]

\[ D_{gz} = M_D(\overline{V}_{gz}, x'_L, x'_R) \quad (2.5) \]

\[ \overline{G}_3 = \overline{V}_{gz} * D_{gz} \quad (2.6) \]
2.4.2 Characteristics of Gaze Vector Method

The gaze vector method has three valuable characteristics that make it outperform the existing methods for 3D gaze estimation. Those characteristics are: (1) it can accurately estimate the gaze vector even when the visual axes have large errors; (2) the error of the 3D gaze mainly depends on the error of the gaze distance when the gaze vector has a small error; and (3) once the gaze vector has been accurately determined, the relationship between pupil features with the gaze distance becomes clear, and the gaze distance can be accurately estimated. Due to these characteristics, the 3D gaze location can be accurately estimated from the gaze vector and gaze distance.

The procedure for gaze vector estimation in the gaze vector method is illustrated in Fig. 2.2, which is computed from visual axes' intersection. In the illustration, the ideal visual axes and gaze vector are shown using solid lines, while the estimated terms are represented with dashed lines. Both of the estimated visual axes $\mathbf{V}_L$ and $\mathbf{P}_R$ have angular errors, which can lead to a significant deviation from their intersection $\mathbf{C}$ to the actual visual target $\mathbf{P}$. However, the gaze vector can be accurately estimated using the gaze vector method even when two visual axes have large angular errors. As shown in Fig. 2.2, the estimated $\mathbf{V}_{gz}$ only has a small offset from the ideal gaze vector $\mathbf{V}_{i_{gz}}$ even when the visual axes deviate largely from the ideal ones.

If we place the origin of the coordinate system (the middle point between two eyes), the visual target, and the estimated 3D gaze point using the gaze vector method onto one plane, it can be shown as Fig. 2.3. The estimated gaze vector $\mathbf{V}_{gz}$ has an angular error $e_v$, which is small ($< 1^\circ$), and the estimated gaze distance $D_{gz}$ has an error $e_D$. If the distance from the visual target to the origin is $D$ ($D_{gz} = D + e_D$), the error $E$ between the estimated 3D gaze with the visual target can be expressed in (2) based on the cosine theorem. As the angle error $e_v$ is small, equation (2.7) can be approximated as (2.8), and (2.9) can be derived from it. It mathematically demonstrates that the error of 3D gaze estimation depends mainly on the error of the estimated gaze distance and is approximately equal to the error of the
Figure 2.2: The gaze vector method can accurately estimate the gaze vector even when the estimated visual axes have large errors. The estimated visual axes and gaze vector are indicated with dashed lines.

estimated gaze distance when the gaze vector has a small error.

\[ E^2 = D^2 + (D + e_D)^2 - 2D(D + e_D)\cos e_v \] \hspace{1cm} (2.7)

\[ E^2 = D^2 + (D + e_D)^2 - 2D(D + e_D) \] \hspace{1cm} (2.8)

\[ E \approx e_D \] \hspace{1cm} (2.9)

Figure 2.3: The gaze vector method can accurately estimate the gaze vector even when the estimated visual axes have large errors. The estimated visual axes and gaze vector are indicated with dashed lines.
When a person looks at visual targets that have the same distance to the person but are along different gaze vector directions, different eye orientations are required to focus on those targets, which result in different eye features. In such situations, it is extremely difficult to build a single mapping relationship that can map these various features to the same distance output. However, along one certain gaze vector direction, the mapping relation between the distance of the visual target and the eye features becomes clear, and the estimation of this distance becomes easier and more accurate. Thus, the gaze vector method, which estimates the gaze vector first and then uses the estimated gaze vector as an important input feature for the gaze distance estimation, can effectively improve the accuracy of 3D gaze estimation.

In the following section of this chapter, we further validate and explore those three characteristics of the gaze vector method. Theoretical analysis and practical tests are both carried out, and further insight about the gaze vector method is revealed.

2.5 Theoretical Analysis
2.5.1 Simulation Setup

Theoretical analysis was performed to investigate the performance and the characteristics of our gaze vector method under various error conditions. The left, right visual axes and gaze distance are all simulated, which contain unavoidable errors. How well the gaze vector method can handle those errors significantly affects the accuracy of 3D gaze estimation, and it is the major objective of this theoretical analysis. As a comparison, the existing visual axes intersection method was also analyzed.

The origin of the coordinate system is assigned at the middle point of the two eyes, and the Y-axis is along two eyes (Fig. 2.1). The Z-axis points straight forward along the depth direction. An overall space (60 cm × 60 cm, width × height) on the XY-plane and 20 cm to 200 cm depth along the Z-axis was investigated. Accuracy analysis is carried out for gaze tracking at single points, vertical planes with points arranged on them, and the entire space. Correspondingly, they are referred to as the point test, plane test, and space test. The analysis starts with an angular error of 0.5° for each visual axis. The reason for selecting 0.5°
as the testing reference was that, in the reported 2D gaze tracking articles and commercial 2D gaze tracking products, 0.5° was the best accuracy that had ever been achieved for the gaze direction (typical error was 0.5° ~1°).

1) **Point Test:** A single point $\mathbf{P}$ is assumed to be the visual target. The visual axes and gaze distance are simulated with specific errors, respectively. Then they are inputted to the 3D gaze estimation method to compute the estimated gaze vector and then the 3D gaze. The estimated 3D gaze is compared with the actual position of the visual target to evaluate the estimation accuracy. The point test demonstrates all possible estimations of the gaze vector and 3D gaze point for a given visual target.

2) **Plane Test:** The plane test consists of a set of point tests whose visual targets are on a plane that is perpendicular to the Z-axis (the depth direction) and at a certain depth. On the plane, there are $21 \times 21$ visual targets assigned in the grid and the distance between two adjacent visual targets is 3 cm. Point test is carried out for each visual target on that plane, and the average error of the point test is used to present the potential error of that visual target.

3) **Space Test:** The space test studies the error deterioration along the depth direction, which is composed of multiple plane tests at different depths. The average error of each plane test is used to represent the error of that plane. The space test is carried out from a depth of 20 cm to a depth of 200 cm with 10 cm as the interval. In addition, more error conditions of the visual axes are simulated to explore the estimation method’s capability of handling larger errors.

2.5.2 Theoretical Analysis of Gaze Vector Estimation

In this analysis, the estimated left visual axis $\mathbf{V}_L$ and right visual axis $\mathbf{V}_R$ are assumed to have an error of 0.5°, which forms a cone surface around each actual visual axis, respectively, shown in Fig. 2.4. The cone has its apex at the eye’s center where the visual axis starts. Twenty possible estimated visual axes are sampled on each cone surface, which result in 400 possible visual axis pairs. These pairs are inputted to the gaze vector method to compute
the intersection and then to estimate the gaze vector.

Figure 2.4: The simulated visual axes $\vec{V}_L$ with a certain estimation error forms a cone surface around the ideal visual axis $\vec{V}_L^i$.

1) Point Test: At visual targets, [0, 0, 40] cm, [0, 0, 70] cm, and [0, 0, 100] cm, the point tests are performed respectively. For each visual target, there are 400 intersections that resulted from the simulated 400 pairs of left and right visual axes, which are shown in Fig. 2.5(a) with different color markers. Each intersection point represents an estimated gaze vector that emits from the origin. The intersection points have different deviations from the visual target, which result in different angular errors for the estimated gaze vectors. Based on this error, the estimated gaze vectors are divided into three groups and their corresponding intersection points are marked with different colors as shown in Fig. 2.5(b)-2.5(d). The gaze vectors with an error between $[0^\circ, 0.2^\circ]$ are cyan; $(0.2^\circ, 0.4^\circ]$ are purple; and $> 0.4^\circ$ are green. Even though the intersection points have different deviations for different visual targets and deviate more when the visual target is further, the resulted gaze vectors have similar distributions. For each visual target, 25% of its estimated gaze vectors have an error that is within $[0^\circ, 0.2^\circ]$, and 30% of the gaze vectors have an error that is in $(0.2^\circ, 0.4^\circ]$. More interestingly, if we sort the errors of the estimated gaze vector for each visual target, they have almost exactly the same distribution pattern shown in Fig. 2.6. The average error is $0.318^\circ$ for each visual target, which means that if the estimated visual axes for a visual target have an error of $0.5^\circ$, the estimated gaze vector would potentially have an error of
0.318°. Moreover, in 95% of the cases, the estimated gaze vector has an error that is less than 0.5°. For the other 5% of the cases, the gaze vector has an error that is close or equal to the error of the visual axes. The zoom-in area in Fig. 2.6 shows that the maximum error of the gaze vector is 0.5038°.

Moreover, another finding made from the colored visual axes’ intersections is that the intersections with large deviations from the visual target could result in small errors of the gaze vectors, shown in Fig. 2.5(e)-2.5(m). In each test of three visual targets, the cyan markers have the largest offsets from the visual target, but result in the most accurate gaze vectors, which have the smallest angular errors that are in [0°, 0.2°].

2) **Plane Test:** Plane test is performed separately on planes that are at depths of 40 cm, 70 cm, and 100 cm. The potential errors of the estimated gaze vector when the visual targets are on those planes are shown in Fig. 2.7. The distributions are V-shaped with valleys are along the X-axis. As two eyes lie on the Y-axis, the error distributions suggest that when the visual target is away from the user in the left or right direction, the estimated gaze vector turns to have a slightly larger error. On three planes, the errors of the gaze vectors are all smaller than the error of the visual axes, which is similar to the results in point tests. Moreover, the error decreases when the visual targets are on a plane that is further away.

3) **Space Test:** The results from the space test are plotted in Fig. 2.8. When visual axes have an error of 0.5°, the potential gaze vector error at a depth of 20 cm is 0.3252° and it reduces slightly to 0.3178° at a depth of 200 cm. Both of these errors are less than the error of visual axes. Similar error reductions are also observed when visual axes have larger errors at 0.7°, 0.9°, 1.1°, and 1.3°. In addition, it is observed that the curves of the gaze vectors’ error have a similar slope, which inspires the investigation of error reduction ratio\(^3\). Error reduction ratio measures how much smaller the error of gaze vector is compared to the errors of the visual axes. Figure 2.8 shows how the error reduction ratio varies along the depth.

\(^3\)Error reduction ratio is calculated as \((\text{Mean of Visual Axis Error} - \text{Gaze Vector Error}) / \text{Mean of Visual Axis Error}\).
Figure 2.5: Point tests at \([0, 0, 40]\) cm, \([0, 0, 70]\) cm, and \([0, 0, 100]\) cm respectively. (a) The pictorial view of all possible intersections of two visual axes. Each marker represents a possible intersection. All the possible gaze vectors are divided into three categories according to their angular errors, and their corresponding markers are marked with different colors. (b)-(d) The top, side, and left views of these markers at each point. (e)-(m) The top, side, and left views of each error category at a single point \([0, 0, 70]\) cm.
Figure 2.6: The error distribution of the estimated gaze vector in point test. The base error of the visual axis is 0.5°. Each point test results in 400 possible gaze vectors where their errors are ordered from low to high. Three points along the depth direction have the same average error of 0.318°.

Figure 2.7: The error of the estimated gaze vector in plane tests.
direction for different error conditions of visual axes. The error reduction ratio is consistent\textsuperscript{4} regardless of the different errors of visual axes, which strongly proves that the gaze vector method can accurately estimate the gaze vector. With this consistent error reduction ratio, the error of the gaze vector can be predicted approximately as (2.10) when the accuracy of the mapping relationship is known. The $e_{gv}$ is the expected error of gaze vector, $e_{va}$ is the visual axis error, and $r$ is the error reduction ratio conservatively taken as 35.0%.

$$e_{gv} = (1 - r) \times e_{va} \tag{2.10}$$

\textit{Summary:} The theoretical results show that estimating the gaze vector using the gaze vector method can achieve an accurate gaze vector even when the visual axes have large errors (proof of the first characteristic of gaze vector method). It shows that the gaze vector method can obtain an accurate gaze vector method in 95% of cases, and the average error reduction is expected to be above 35%. The capability of error reduction is very stable regardless of the error conditions of the visual axes, which makes it possible to predict the error of the gaze vector when the errors of the visual axes are approximately known.

\subsection*{2.5.3 Theoretical Analysis of 3D Gaze Estimation}

The estimated gaze vectors in the previous section are used to estimate the 3D gaze point with simulated gaze distances. Based on our experimental observations, the base error of gaze distance $e_D$ is assumed to be $\pm 2$ cm. As the sign of the gaze distance’s error does not cause significant difference to the overall estimation error of the 3D gaze point, for the sake of simplification in this chapter, it is assumed to be positive. In addition, various error conditions of gaze distance are investigated. For each visual target, there are 400 estimations due to the 400 possible gaze vectors resulting from the visual axes pairs. As a comparison, the visual axes intersection method is also analyzed with the same visual axes pairs. The point test, plane test, and space test are performed, respectively.

\textsuperscript{4}The small jerks that happen when visual axes have errors of $1.1^\circ$ and $1.3^\circ$ are caused by a term called error overshooting and will be further discussed in the section Discussion.
Figure 2.8: Space test of the gaze vector estimation with various errors of visual axis, which is annotated. (a) The angular error of the gaze vector when visual targets are on different depth. (b) The error reduction ratio of the gaze vector’s angular error when visual targets are on different depths. The potential 3D gaze error is plotted on the left axis with black lines, and the error reduction ratio is plotted on the right axis with color lines. Different markers are used to annotate different visual axis errors.
1) Point Test: The point test of the gaze vector method and visual axes intersection method are separately performed at visual targets [0, 0, 40], [0, 0, 70], and [0, 0, 100] cm. As the visual axes intersection method uses the intersection point as the 3D gaze point directly, point test results of the visual axes intersection method are those in Fig. 2.5, where each dot represents a possible estimation. The estimated gaze points scatter around the visual target and form an interesting distribution shape. Moreover, the distribution shapes are similar for different visual targets but scatter more when the visual target is further away, which suggests larger estimation errors. The point test results (green squares) of the gaze vector method at point [0, 0, 70] cm are plotted in Fig. 2.9, in which results (purple squares) of the visual axes intersection method at this visual target are also displayed for a comparison. The estimated 3D gaze points using the gaze vector method are roughly on one plane, as the error differences are very small. The same observation is made for the other two visual targets. Please note that the Z-axis has a very different scale from the X-axis and Y-axis.

Figure 2.9: The distribution of the estimated 3D gaze using the gaze vector method (green squares) and using the visual axes intersection method (purple squares), respectively, at point test of [0, 0, 70] cm. Note that the Z-axis has a very different scale from the X-axis and Y-axis.

The qualitative results from the point tests of both methods are summarized in Table 2.1, which contains the average and maximum errors on each axis and overall errors. The gaze vector method has potential errors of 2.02 cm, 2.05 cm, and 2.09 cm at three tested visual targets, respectively, where the error differences are negligible. However, the error using
the visual axes intersection method quickly grows to 10.27 cm. Moreover, the maximum possible error in each point test is significant and reaches 33.26 cm in the worst case. For both methods, the major error happens to be on the Z-axis.

Table 2.1: Error summarization of the point tests using the gaze vector method (GVM) and the visual axes intersection method (VAIM). P1 is [0, 0, 40] cm; P2 is [0, 0, 70] cm; P3 is [0, 0, 100] cm. Ave stands for average. Max stands for maximal. Min stands for minimal. e stands for error.

<table>
<thead>
<tr>
<th>Unit: cm</th>
<th>Gaze vector method</th>
<th>Visual axes intersection method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P1</td>
<td>P2</td>
</tr>
<tr>
<td>Max e on X</td>
<td>0.37</td>
<td>0.63</td>
</tr>
<tr>
<td>Ave e on X</td>
<td>0.15</td>
<td>0.25</td>
</tr>
<tr>
<td>Max e on Y</td>
<td>0.37</td>
<td>0.63</td>
</tr>
<tr>
<td>Ave e on Y</td>
<td>0.15</td>
<td>0.25</td>
</tr>
<tr>
<td>Max e on Z</td>
<td>2.00</td>
<td>2.00</td>
</tr>
<tr>
<td>Ave e on Z</td>
<td>2.00</td>
<td>2.00</td>
</tr>
<tr>
<td>Max Euclidean e</td>
<td>2.03</td>
<td>2.09</td>
</tr>
<tr>
<td>Ave Euclidean e</td>
<td>2.02</td>
<td>2.05</td>
</tr>
</tbody>
</table>

2) Plane Test: The plane tests are conducted at depths of 40, 70, and 100 cm, respectively, for both 3D gaze estimation methods. The results are demonstrated in Fig. 2.10. The error results from the gaze vector method are so close that they overlap significantly in the plot, which is close to 2 cm. In contrast, there is a large error increase for the visual axes intersection method. In addition, the error distribution on a testing plane is bowl-shaped, which means the error grows when the visual target are further from the center of the test plane. Similarly, the errors of the gaze vector method also are also bowl-shaped. However, the error change on a testing plan is negligible, resulting in results in an error distribution appearing like a plane.

Space Test: The errors from space tests of both methods are plotted in Fig. 2.11, in which the errors are logarithmically scaled (the Y-axis) for display convenience. The red lines represent the results from the visual axis intersection method, the blue lines are from the gaze vector method, and various error conditions of visual axes are indicated with different
Figure 2.10: Results from plane tests of the visual axes intersection method (VAIM) and the gaze vector method (GVM).

markers. With the visual axis intersection method, the potential error of 3D gaze is large and rapidly grows up when the visual target is further away. Specifically, when the visual axes have an error of $0.5^\circ$, the potential error of 3D gaze using the visual axes intersection method is $1.40 \text{ cm}$ at depth of $20 \text{ cm}$, while it grows up to $42.62 \text{ cm}$ at depth of $200 \text{ cm}$. Moreover, the potential error when visual axes have a $1.1^\circ$ or $1.3^\circ$ error is outside the display range of $100 \text{ cm}$. In contrast, the errors of the gaze vector method are quite stable across the depth range and different visual axes error conditions. The worst error condition of using the gaze vector method is $3.67 \text{ cm}$ at depth $200 \text{ cm}$ when the visual axes have an error of $1.3^\circ$.

In previous analysis, the gaze distance is assumed to have an error of $2 \text{ cm}$, which is the ideal and expected condition. It is possible for an estimation of gaze distance with larger errors. With this consideration, space tests of the gaze vector method when the simulated gaze distance has various errors are performed and their results are plotted in Fig. 2.12.
Figure 2.11: Results from space tests of the visual axes intersection method (VAIM, red lines) and gaze vector method (GVM, blue lines). Different error conditions of the visual axes are annotated with different markers.

The results demonstrate that the error of the 3D gaze (green lines with solid markers) is tightly related to and approximately equal to the gaze distance’s error regardless of where the visual target is located. Interestingly, it is observed that when the gaze distance has no error, the 3D gaze has the greatest error increase when the visual target is further away. This inspires the investigation of the gaze error ratio\(^5\), which is a comparison between the error of the estimated 3D gaze point to the gaze distance’s error. The gaze error ratios at different error conditions are plotted in Fig. 2.12 as blue lines with hollow markers. The plot shows that when the gaze distance has a larger error, the error ratio is smaller. This means the gaze vector method would not enlarge the final 3D gaze’s error when the gaze distance has a large error.

Summary: The above analysis shows that the gaze vector method can accurately estimate the 3D gaze location, and this capability is stable across the entire workspace even when the visual axes have significant errors. It also shows that the final 3D gaze error is mainly

\(^5\text{Gaze error ratio is calculated by } \left\{ \frac{(3D \text{ gaze’s error}) - (gaze distance’s error)}{(gaze distance’s error)} \right\}.\)
Figure 2.12: Space test results of gaze vector method when gaze distance have various errors. The various gaze distance’s errors are annotated with different markers. The 3D gaze errors are shown as the green lines, and the error reduction ratios are blue lines.

dependent on the error of the gaze distance (the second characteristic of the gaze vector method). In contrast, the error of the visual axes intersection method is large and increases quickly when the visual target is further away and when the visual axes have a large error.

2.6 Experimental Validation

From theoretical analysis, we have proved and further demonstrated the first two characteristics of the gaze vector method: first, that it can obtain an accurate gaze vector even when the visual axes have large errors and, second, the error of the 3D gaze mainly depends on the error of the gaze distance when the gaze vector has a small error. In this experimental validation, we intend to answer the most important question about whether the gaze distance can be accurately estimated with the gaze vector method (the third characteristic of the gaze vector method). To answer this question, the gaze vector method was integrated on a physical eye tracking system and tested to estimate where a user is looking in a real environment.
2.6.1 3D Gaze Tracking System

A 3D gaze tracking system was built with the gaze vector method integrated on a binocular eye tracker, which was based on a head-mounted frame [46], shown in Fig. 2.13. In this system, both eyes were tracked with image sensors, and from the captured images eye movement features were extracted. Those features were pupil position, pupil rotation, pupil dimension, and pupils’ distance. A Neural Network (NN) was utilized to implement the mapping functions $M_{va,L}(x_L)$, $M_{va,R}(x_R)$ and $M_D(V_{gz}, x'_L, x'_R)$ for estimating the left visual axis, right visual axis and gaze distance, respectively. Following the gaze vector method, a person’s 3D gaze in a real environment can be computed as the product of the gaze vector and the gaze distance.

![Image of the binocular eye tracker for 3D gaze tracking.](image)

Figure 2.13: The binocular eye tracker for 3D gaze tracking.

A calibration process was carried out to build the mapping relations with NNs. During the calibration, each participant wore the eye tracker and was asked to look at a set of markers in a $4 \times 4$ grid on a plane that was 27 cm wide and 27 cm tall. The plane was placed at four different depths from 60 cm to 100 cm. The visual axes, the gaze vector, and the gaze distance are computed and used to build the correlation with the pupil features. During the testing stage, subjects were asked to view another set of markers. For judicial evaluation, only 22% of the testing markers belonged to the markers for calibration. At those testing locations, the person’s 3D gaze was estimated and compared to the actual position of the
markers. Experiments with thirty subjects were carried out in accordance with the Code of Ethics of the World Medical Association. Prior to participating in the study, a short introduction was provided to the participants, including technologies involved, the system setup, and the purpose of the study.

### 2.6.2 Experiment Results

The average errors of the left and right visual axes from the NN estimators were $1.08\pm0.55^\circ$ and $0.63\pm0.32^\circ$, respectively, and the average error of the gaze vector was $0.56\pm0.31^\circ$, summarized in Table 2.2. This was consistent with the finding in the theoretical analysis that the gaze vector method can accurately estimate the gaze vector even when the visual axes have large errors. The average reduction ratio of this method in the experimental tests was 34.50%.

Table 2.2: Average error summarization of the estimated left and right visual axes and the computed gaze vector.

<table>
<thead>
<tr>
<th></th>
<th>Gaze vector</th>
<th>Left visual axis</th>
<th>Right visual axis</th>
<th>Error reduction ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error</td>
<td>$0.56\pm0.31^\circ$</td>
<td>$1.08\pm0.55^\circ$</td>
<td>$0.63\pm0.32^\circ$</td>
<td>34.50%</td>
</tr>
</tbody>
</table>

The 3D gaze estimation errors using the gaze vector method are summarized in Table 2.3 with a comparison to the visual axes intersection method. The gaze vector method achieved an average error of 2.4 cm with the worst case being 8.9 cm. The visual axes intersection method had a much larger error with an average error was 6.7 cm and a worst scenario of 31.9 cm. The experimental results prove that the gaze vector method is an accurate method for 3D gaze estimation and can provide accurate 3D estimation in practice. Moreover, it also shows that the gaze distance can be accurately estimated in practice following the gaze vector method.
Table 2.3: Error summarization of 3D gaze estimation using the gaze vector method and the visual axes intersection method. Max: maximal, Min: minimal, Ave: average, and e: error.

<table>
<thead>
<tr>
<th>Unit: cm</th>
<th>Gaze vector method</th>
<th>Visual axes intersection method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Max</td>
<td>Min</td>
</tr>
<tr>
<td>e on X</td>
<td>2.4</td>
<td>0.2</td>
</tr>
<tr>
<td>e on Y</td>
<td>1.0</td>
<td>0.1</td>
</tr>
<tr>
<td>e on Z</td>
<td>6.6</td>
<td>0.6</td>
</tr>
<tr>
<td>Overall</td>
<td>6.7</td>
<td>0.8</td>
</tr>
</tbody>
</table>

2.7 Discussion

2.7.1 Overshooting of Visual Axes Intersection Method

In our theoretical analysis of the visual axes intersection method, severe errors occurred when the visual axes have an error of 1.1° or 1.3° (Fig. 2.11). With further investigation by extending the depth range, error severity increases and eventually lead to error overshooting. Figure 2.14 shows the space tests with the extended depth range where severe errors occur when visual axes have an error of 0.7°, 0.9°, 1.1°, or 1.3°. The error overshooting occurs at a shorter depth when the error of visual axes is larger. This suggests that an error overshooting is expected to happen even if the visual axes have an error of 0.5° or smaller when the investigation depth range is further. Further investigation of this overshooting issue reveals that the major error always appears on the Z-axis (the depth direction).

There are two conditions that result in the error overshoot of the visual axes intersection method. One condition is that the two estimated visual axes are close to being parallel, which causes the intersection at a point that is infinitely far away. Another condition is that the estimated visual axes intersect in the backward direction (which is infeasible in reality), shown in Fig. 2.15, due to errors in the visual axis estimation. Though it is very difficult to predict the overshooting as it happens by chance, which is dependent on the location of the visual target and also the errors of the left and right visual axes, the larger errors of the estimated visual axes do increase the chances of the overshoot and cause it to happen at a
shorter depth. The gaze vector method avoids these two unrealistic estimation conditions. Even with error overshooting conditions, the gaze vector can still be accurately estimated following the gaze vector method shown in Fig. 2.8, where the effects of the error overshooting is negligible. With an accurate gaze vector, the gaze distance estimation would not be affected by the previous error overshooting. Moreover, the experimental tests demonstrated that the gaze distance can be accurately estimated along an accurately estimated gaze vector.

2.7.2 Workspace of the Visual Axis-based Methods

To avoid the gaze error overshoot, the error of the visual axes and the workspace need to be considered at the same time. With specific errors of the visual axes, a safe workspace under these errors can be defined, in which the chance of the error overshooting is significantly low. Recall Fig. 2.15 where, to avoid both conditions of error overshooting, the angle \((e_L + EG_L + e_R + EG_R)\) should be much less than 180°. The relation can be expressed as (2.11) where a new parameter \(\omega \ (\omega \geq 0)\) is introduced which can further tighten the
Figure 2.15: The demonstration of the gaze error overshoot when estimated visual axes intersect at the backward direction. \( EG_L \) and \( EG_R \) are left and right eye gaze angles, respectively. \( e_L \) and \( e_R \) are angular error of the estimated left visual axis and right visual axis. \( \mathbf{V}_L \) and \( \mathbf{V}_R \) are the estimated left visual axis and right visual axis, respectively, which consist of errors. \( \mathbf{P} \) is the location of the visual target.

constraint of the governing function. The angles \( EG_L \) and \( EG_R \) can be computed from the eye locations \( \mathbf{E}_L, \mathbf{E}_R \) and the visual target location \( \mathbf{P} = [x, y, z]^T \), which results in (2.12).

In the defined coordinate system, two eyes are lying on the Y-axis and their distance is \( l \).

Thus, the governing function can be rewritten as (2.13). To avoid the error overshooting, the visual target has to satisfy this relation with two visual axes’ errors. In other words, equation (2.13) defines the workspace of the visual axes intersection method, which is a function of the left and right visual axes’ angular errors. In this analysis, the estimated visual axes are assumed to be on the same plane with two eyes and the visual target. This is the boundary case before more complex overshooting conditions, like when the estimated visual axes are not on the eyes-visual-target plane or even any plane. By increasing \( \omega \), the chance of error overshooting can be significantly reduced, but it results in a much smaller workspace.

\[
e_L + EG_L + E_R + EG_R < 180^\circ - \omega
\]  

\[
\cos^{-1}\left(\frac{(\mathbf{E}_R - \mathbf{E}_L) \cdot (\mathbf{T} - \mathbf{E}_L)}{||\mathbf{E}_R - \mathbf{E}_L|| \cdot ||\mathbf{T} - \mathbf{E}_L||}\right) + \cos^{-1}\left(\frac{(\mathbf{E}_L - \mathbf{E}_R) \cdot (\mathbf{T} - \mathbf{E}_R)}{||\mathbf{E}_L - \mathbf{E}_R|| \cdot ||\mathbf{T} - \mathbf{E}_R||}\right)
< 180^\circ - \epsilon - e_L - e_R
\]
\[
\cos^{-1}\left(\frac{y + l/2}{\sqrt{x^2 + (y + l/2)^2 + z^2}}\right) + \cos^{-1}\left(\frac{-y + l/2}{\sqrt{x^2 + (y + l/2)^2 + z^2}}\right) < 180^\circ - \epsilon - e_L - e_R
\] (2.13)

2.8 Acknowledgments

This material is based on work supported by the US NSF under grant 1414299. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the National Science Foundation. The authors would like to thank Jeremy Webb for assisting with preparation of the experiment setup.

2.9 Appendix: Computation of Gaze Vector From Visual Axes

The gaze vector $\mathbf{V}_{gz}$ is calculated using equations (2.14)-(2.20) from the intersection of two visual axes. Two visual axes intersect with their common normal at $\mathbf{I}_L$ and $\mathbf{I}_R$, which can be defined parametrically as (2.14) and (2.15). The direction vector of the common normal, $\mathbf{V}$, can be calculated as the cross product of the left and right visual axes using (2.16). In addition, $\mathbf{I}_L$ and $\mathbf{I}_R$ have the relationship as described in (2.17). $s$, $t$, and $m$ are the scalar coefficients.

\[
\mathbf{I}_L = \mathbf{E}_L + s \cdot \mathbf{V}_L
\] (2.14)

\[
\mathbf{I}_R = \mathbf{E}_R + t \cdot \mathbf{V}_R
\] (2.15)

\[
\mathbf{V} = \mathbf{V}_L \times \mathbf{V}_R
\] (2.16)

\[
\mathbf{I}_L = \mathbf{I}_R + m \cdot \mathbf{V}
\] (2.17)

From above definitions, a linear relationship can be formulated as (2.18). Solving it produces a unique solution set of $s$, $t$, and $m$, correspondingly, the unique intersection of the common normal with each visual axis can be calculated as (2.14) and (2.15). Thus, the center of the common normal $\mathbf{C}_c$ and the gaze vector can be calculated using (2.19) and
(2.20), respectively.

\[ s \cdot \mathbf{V}_L - t \cdot \mathbf{V}_R + m \cdot \mathbf{V} = E_L - E_R \]  
\[ (2.18) \]

\[ \overline{C_c} = \frac{I_L + I_R}{2} \]  
\[ (2.19) \]

\[ \mathbf{V}_{gz} = \frac{\overline{C_c}}{|\overline{C_c}|} \]  
\[ (2.20) \]
CHAPTER 3
ACCURATE 3D GAZE TRACKING IN REAL ENVIRONMENTS BASED ON EYE MOVEMENT STUDIES

A paper submitted to *Advanced Robotics*
Songpo Li\(^6\) and Xiaoli Zhang\(^7\)

3.1 Abstract

Tracking a person’s 3D gaze in real environments is fundamentally necessary for human visual behavior studies. Only in this way can valid data reflecting actual visual behaviors of a human be collected. Moreover, 3D gaze can also be a potential communication channel for intuitive and effortless human-machine interactions, such as navigating a mobile robot to fetch objects. However, current 3D gaze tracking technologies are still underdeveloped, and it is very challenging to achieve a high degree of accuracy in real 3D environments. In this chapter, a novel 3D gaze tracking system is presented with a focus on developing new technologies to improve tracking accuracy in real environments. In working towards this goal, a novel gaze vector method for 3D gaze estimation is developed. Fundamental studies of eye movement patterns were performed to investigate how a human views visual stimuli in a real 3D environment. From this study, the representative features that could stably characterize eye movements were selected. The gaze vector method with input of the selected features was tested, and a high degree of 3D gaze tracking accuracy in a real 3D environment is achieved. Moreover, hardware evaluation is introduced to evaluate the eye tracker design aiming for accurate 3D gaze tracking and large tracking space.

\(^6\)Primary researcher and author, graduate student, Department of Mechanical Engineering, Colorado School of Mines.
\(^7\)Author for correspondence, Assistant Professor, Department of Mechanical Engineering, Colorado School of Mines.
Keywords: 3D gaze tracking, real environments, gaze vector method, eye feature selection, eye tracker hardware evaluation

3.2 Introduction

Being able to tell where a person is looking is an important area of research and refers to a type of technology called gaze tracking. Gaze is naturally and deeply linked with a human’s attention and the cognition process, and this correlation has been widely used to support studies in various researches including neurobiology [4, 5], psychology [6, 7], computer science [8, 9], human factors [10, 11], etc. Moreover, researchers have also investigated the potential of using gaze as a communication tool to interact with computers [47, 48] or robots [13, 14]. However, technologies of gaze tracking are mainly available for virtual environments such as computer display screens, two-dimensional (2D) images and stereo image displays [1–3].

Recently, particular attention has been given to three-dimensional (3D) gaze tracking in real environments, which tracks where a person is looking in the 3D real world and reports this location’s absolute coordinates (X, Y, Z). This technique provides essential 3D information of a person’s gaze for human behavior studies and cognition studies. For example, it provides information to build a fundamental understanding of how humans visually perceive or visually interact with the surroundings, which can further reveal the mental process. Because humans live and interact in a physical 3D world, it is preferable to perform human behavior studies in 3D surroundings so a person can naturally act. Only in this way can valid data reflecting actual visual behaviors of a human be collected. Moreover, the absolute location of an object in a real environment that is being gazed at can be retrieved from the 3D gaze. Thus, 3D gaze can be employed as a natural communication channel in real environments, which allows a user to intuitively and efficiently interact with surrounding machines by looking at the objects. For example, it could be used to communicate with assistive robots for object retrieval.

However, it is very challenging to achieve a high degree of accuracy in a relatively large workspace for practical applications. In the limited previous studies, researchers tried to
directly adopt the hardware design and methods used for virtual gaze tracking into the real 3D environment. However, the achieved accuracy was low and the workspace was small. Moreover, fundamental research regarding this topic has not been conducted. This research includes how to design an eye tracking system suitable for 3D gaze tracking, what features should be measured in order to capture eye’s movement during visualization of a real 3D environment, and how to estimate the 3D gaze in a real environment.

In this chapter, a novel 3D gaze tracking system for real environments is presented. The main contributions of this chapter are as follows:

1. A novel gaze vector method for 3D gaze estimation is developed in the presented system, which constructs the 3D gaze in a way that is different from any previous research. This new method decouples 3D gaze estimation into the estimation of the gaze vector and then the estimation of the gaze distance along the gaze vector. The gaze vector is the vector emitting from the middle point between two eyes to the visual target (the actual 3D gaze point). The gaze distance is the absolute distance from the eyes’ middle point to the visual target. This new method has been validated in our experiments to: a) effectively attenuate the common error propagation issue in current 3D gaze tracking techniques, and b) offer accurate 3D gaze estimation in a relatively large workspace for practical usage.

2. Fundamental visual behavior studies are carried out to capture the eye movements during visualization of a 3D real environment. We investigated different eye features including pupil location, pupil size, pupil dimension, pupil rotation angle (the pupil appears as an ellipse on the captured image, and this ellipse’s orientation is defined as pupil rotation angle), and pupil distance (the distance between the centers of two eyes). From this study, the representative and stable eye features were selected for 3D gaze estimation.
3. Hardware evaluation for eye tracking is introduced for the first time, and particular design parameters are introduced for 3D gaze tracking in real environments. For 3D gaze tracking, the distance from the visual target to the user varies, and eye movement features become subtle when the visual target is further. How well the eye tracker can distinguish those subtle movements significantly affects its tracking accuracy and workspace. Accordingly, two criteria, eye tracking resolution and effective region of pupil distance, are introduced to evaluate a tracking system.

3.3 Related Work

3.3.1 Eye Movement Tracking and Gaze Estimation in Virtual Environments

Extensive research has been conducted regarding eye movement tracking and gaze estimation in virtual environments. Various techniques that monitor eye movements have been explored, including contact lens, electrooculogram (EOG) [14, 49] and optical method [19, 50]. Nowadays, video-based optical tracking has become common both in research and on the market. It offers noninvasive tracking of the eyes, whereas the other two methods require direct contact with the eyeballs or the skin around the eyes. Moreover, better accuracy can be achieved with optical tracking methods. Based on different configurations of an eye tracker’s video cameras, there is a table-stand setup (remote) and a head-mounted setup (goggle-like, wearable). In the head-mounted setup, normally there is a scene camera facing what the user is facing toward, and a pixel point on the scene image is used to represent where the user is looking. The head-mounted setup provides the users with more freedom by allowing them to move around. In contrast, the table-stand systems are mainly designed to track users’ gaze points on a monitor or a projector display.

Gaze estimation methods that have been employed for 2D gaze estimation in virtual environments can be categorized into regression method or geometrical method. Regression method directly maps the gaze location to a set of eye movement features, and the mapping relation is built based on a calibration process [18, 51]. In geometrical method, the gaze position is solved as the intersection of a visual axis with the display screen [1, 52]. The visual
axis can be reconstructed based on an anatomic eye model or from a mapping relationship. This method has also been used to estimate the gaze in a virtual 3D environment with one more step that estimates where the virtual display screen is located [27, 53, 54]. The pupil distance has commonly been used to indicate the location of that virtual display screen.

3.3.2 3D Gaze Tracking in Real Environment

3D gaze tracking technology is still in its infancy, and all reported researches are limited by their accuracy and workspace. In this review, works that only estimate the gaze direction are excluded, and we focus on these that can explicitly report the 3D coordinates of a gaze point in real 3D environments. In some literatures, researchers converted a 2D gaze point to its corresponding 3D location using an RGB-D camera [39, 42] or stereo cameras [43, 44]. However, these methods have limitations on system size, hardware complexity, and setup complexity, which are caused by the necessary additional hardware for 3D position reconstruction.

In 2009, Hennessey reported the first system for 3D gaze tracking in a real environment, which only required eye information from a table-stand binocular eye tracking system [32]. Over the entire workspace of 30 cm × 23 cm × 25 cm (width×height×depth) (workspace volume 17250 cm³), an average accuracy of 3.93 cm was achieved, and this workspace was at a general distance for human-computer interaction (it was estimated to be 17.5 cm away in [33]). In their system, the hardware specifications have to be calibrated, and a user calibration is also required to identify the personal eye parameters. Their estimation is based on the intersection of two visual axes (named as visual axes intersection method, a visual axis is defined as an axis pointing from the eye’s center to the gazed object/location), in which the intersection point is considered to be the 3D gaze location. In 2012, Abbott and Faisal reported their 3D gaze tracking system [33], which only required a user calibration. Their estimation was also based on the visual axes intersection method. An overall error of 5.8 cm (with a 5.1 cm error in depth) was achieved among a testing distance from 54 cm to 108 cm in a 47 cm wide and 27 cm high workspace (workspace volume 47 cm × 27 cm × 54 cm³).
The visual axes intersection method is based on the assumption that when a person visually concentrates on a target or a location in the 3D environment, the visual axes intersect at that location. However, due to the fact that the estimated visual axes contain a certain degree of error, two visual axes do not intersect most of the time. Thus, in practical usage, the point that has the shortest distance to both visual axes is considered as the intersection of two visual axes and then the estimated gaze point, which is the middle point of two visual axes’ common normal. During this process, the angular error of each visual axis will cause a significant error of 3D gaze as the intersection significantly deviates from the visual target (error propagation). Moreover, the condition of the error propagation gets worse, which produces much larger 3D gaze error when the visual target is further.

Another 3D gaze tracking system was reported by Lee in 2012, which was based on a monocular eye tracker [38]. In this system, the method that had been used for 3D gaze in virtual environments was employed, which estimates and intersects the visual axis and a virtual display screen. This system was tested in a pyramid-shape workspace, whose apex was at the eye center. The pyramid’s base was a 10 cm $\times$ 10 cm square, and the height was 50 cm (workspace volume $\approx$ 1667 cm$^3$). However, even in a setup where the testing points were the same as the calibration points and they were collected in the same data collection period, the experimental results showed low accuracy. The average errors were 0.96° along the X-axis, 1.60° along the Y-axis, and 4.59 cm along the depth Z-axis.

### 3.4 3D Gaze Estimation in Real Environments

To achieve a high degree of accuracy for 3D gaze tracking in real environments, several efforts from different perspectives are carried out. These efforts include a novel gaze vector method based on visual behavior pattern, eye feature selection based on fundamental eye behavior studies, and Neural-Network-based estimation.
3.4.1 Binocular Eye Tracker

To be able to track a person’s 3D gaze in a real environment, a binocular eye tracker is built. It is designed to be light-weight, compact, and wearable so that the system can be practically used in our later studies of human behavior or human-robot intention, in which the user may need to walk around wearing it and perform tasks naturally. The hardware platform, shown in Fig. 3.1, is built on a head-mounted frame [46]. It has two extendable mounts for image sensors, which can be adjusted to face each eye respectively for individuals. The image sensors offer a maximum frequency of 30 frames per second and a resolution of 640×480. Live videos from the image sensor are streamed to a computer through USB 2.0 connections. A visible light filter is laid on the top of the image sensor and four near infrared (IR) LEDs are mounted around each image sensor to illuminate the pupil. Under the IR light, the pupil appears to be dark, and there are reflections of the light sources on the iris, which are called glints. The estimation reference frame is defined in Fig. 3.1, in which the X-axis points down vertically, the Y-axis points to the left side horizontally, and the Z-axis (depth direction) points forward horizontally. The origin of the coordinates (the red dot) is at the middle of two eyes.

Figure 3.1: The binocular eye tracker with coordinate system.

OpenCV image processing library [55] in C++ is used to process eye images and track the eye position. The IR LEDs and the visible light filter are used to increase the contrast between the pupil and iris. This high contrast effectively simplifies the procedure of pupil segmentation with an intensity threshold, which converts a greyscale image (Fig. 3.2(b))
to a binary one (Figs. 3.2(c)). On the image the pupil appears as an ellipse. Following a standard image processing procedure shown in Fig. 3.2, the pupil region is recognized by a shape-based classifier. An ellipse is fitted to the pupil region as the representation of the pupil (Fig. 3.2(e)) and its center position, dimension, size, and rotation angle are reported for later gaze estimation.

Figure 3.2: Steps of pupil identification from the image: (a) raw color image, (b) grayscale image, (c) binary image, (d) binary image after noise elimination, and (e) original image with the identified pupil.

3.4.2 Gaze Vector Method for 3D Gaze Estimation

Figure 3.3: Illustration of the gaze vector method with the decoupled gaze vector $v_g$, gaze distance $l_g$, $e_L$ and $e_R$ are the position of left and right eyes, $v_L$ and $v_R$ are the left and right visual axes, $i_L$ and $i_R$ are the intersection of the two visual axes with their common normal. The middle point of $i_L$ and $i_R$ is considered as the intersection of two visual axes.

a. Gaze Vector Method Gaze vector method decouples the 3D gaze estimation into the estimation of the gaze vector through visual axes intersection and the estimation of the gaze distance along the vector, shown in Fig. 3.3. Along one gaze direction, there is a relatively
clear eye movement changes when the visual target is closer or further. Left and right visual axes, $v_L$ and $v_R$, are estimated first from each eye’s movements using pre-trained mapping relationships $M_L(x_L)$ and $M_R(x_R)$. $x_L$ and $x_R$ are a set of eye features of the left eye and right eye separately. The gaze vector, $v_g$, is defined as a vector emitting from two eyes’ middle point to the visual target and its estimation using visual axes intersection is introduced in Appendix. Then the computed gaze vector, combined with selected eye features, $x'_L$ and $x'_R$, is used to estimate the distance from the human to the visual target, the gaze distance $l_g$, through another pre-trained mapping relationship $M_{DL}(v_g, x'_L, x'_R)$. Thus, the Cartesian coordinates of the 3D gaze point, $g_3$, can be expressed as (3.1). A calibration process is required to build the mapping relations $M_L$, $M_R$, and $M_{DL}$.

$$g_3 = v_g \cdot l_g$$

\hspace{1cm} (3.1)

b. Characteristics of Gaze Vector Method
Gaze vector method has several characteristics that make it outperform the existing methods for 3D gaze estimation. Those characteristics are (1) it can accurately estimate the gaze vector even when the visual axes have large errors; (2) the error of the 3D gaze mainly depends on the error of the gaze distance when the gaze vector has a small error; and (3) after the gaze vector has been accurately determined, the relationship between pupil features with the gaze distance becomes clear and the gaze distance can be accurately estimated. At the end, the 3D gaze location can be accurately estimated from the gaze vector and gaze distance.

The gaze vector estimation using gaze vector method is illustrated in Fig. 3.4, which is computed from visual axes’ intersection. In the illustration, actual visual axes and gaze vector are shown using solid lines, while the estimated items are represented with dashed lines. Both of the estimated visual axes $v_L$ and $v_R$ have an angular error, which can lead to a significant error if their intersection is considered as the 3D gaze location following the visual axes intersection method. In contrast, an accurate gaze vector can be estimated using gaze vector method even with two visual axes that has errors. As shown in Fig. 3.4, the estimated $v_g$ only has a small offset from the actual gaze vector $v^a_g$ even that the visual axes
deviates largely from the actual ones.

Figure 3.4: The gaze vector method can accurately estimate the gaze vector even when the visual axes have large errors. The estimated visual axes and gaze vector are indicated with dashed lines.

If we place the origin of the coordinate system, the visual target, and the estimated 3D gaze point using gaze vector method onto one plane, it can be shown as Fig. 3.5. The estimated gaze vector \( \mathbf{v}_g \) has angular error \( \omega \), which is small \((<1^\circ)\), and the estimated gaze distance \( l_g \) has error \( \epsilon \). If the distance from the visual target to the origin is \( l \ (l_g = l \pm \epsilon) \), the error \( \rho \) between the estimated 3D gaze with the visual target can be expressed in (3.2) based on the cosine theorem. As angle \( \omega \) is small, (3.2) can be approximated as (3.3), and (3.4) can be derived from it. It mathematically demonstrates that the error of 3D gaze depends mainly on the error of the estimated gaze distance and is approximate to it when the gaze vector has a small error.

\[
\rho^2 = l^2 + (l \pm \epsilon)^2 - 2l(l \pm \epsilon) \cos \omega \quad (3.2)
\]
\[
\rho^2 \approx l^2 + (l \pm \epsilon)^2 - 2l(l \pm \epsilon) \quad (3.3)
\]
\[
\rho \approx \epsilon \quad (3.4)
\]

When a person looks at visual targets that have the same distance to the person but along different directions, different eye orientations are required to focus on those targets,
Figure 3.5: Estimating 3D gaze by combining the gaze vector and the gaze distance in gaze vector method.

which results in different eye features. It is extremely difficult to build a function that can map those various features to the same distance output. However, along one gaze direction, there is a relatively clear eye movement changes when the visual target is closer or further. Thus, the mapping relation between the distance of the visual target and the eye features becomes clear and the estimation of this distance becomes easier and more accurate.

3.4.3 Eye Movement Studies for Feature Selection

How eyes behave and appear on a captured image when the user is looking at a set of stimuli in a 3D real environment is investigated. In this investigation, the visual stimuli are laid in different patterns to make the patterns of the eye movement and pupillary response more apparent. The purpose of this investigation is to select the representative and stable features that can characterize the eye movements. The data in plots have been centered by subtracting the mean.

3.4.3.1 Stimuli on a Plane

A $4 \times 4$ grid of visual stimuli was arranged on a plane, and each visual stimulus was 9 cm away from those adjacent to it in the horizontal and vertical directions. The plane was placed at four different depths to study the eye movement features, and the sample data presented here were collected when the depth plane was 40 cm away from the user. Four
subjects participated in this study, and each subject viewed those stimuli for a short period of time, one by one following an order of left to right and top to bottom. Various pupil appearance features on the image were investigated, including pupil location, pupil rotation angle, and pupil size. For each feature, an example set of the collected data is plotted in the following figures.

**Pupil position:** The pupil position, in terms of the X and Y coordinates, is shown in Fig. 3.6. The vertical axis is the X, Y value in pixel. The horizontal axis is the number of the collected data. Both X and Y components show a stair-like shape, and each stair surface corresponds to a visual stimulus. The stair surfaces are roughly planar, which means the pupil remains at the same position while viewing each stimulus. The height of the stair represents the pupil position change when the subject shifts the visual concentration from one stimulus to another. It is observed that pupil location is stable when the user is viewing a visual stimulus, and the difference is distinct when the user is viewing different stimuli. In conclusion, the pupil location is an effective feature for characterizing eye movements.

![Figure 3.6: X and Y components of the pupil location when one subject is viewing the 4 × 4 grid of visual stimuli on a plane.](image)

**Pupil rotation angle:** When a person is looking at visual stimuli that are at different locations, the pupil rotation angles are different. One sample set of the pupil rotation when
the user is looking at four stimuli in a row is shown in Fig. 3.7. For convenience, green dashed lines are used to indicate the rotation. Figure 3.8 is the plot of the pupil rotation angle when a user was viewing the $4 \times 4$ visual stimuli. The plot also has a stair-like shape with a planar surface, and the difference from one stair to the next is distinct. This means the pupil ellipse has a stable rotation angle when the subject is viewing a certain stimulus, and it is distinctive among different stimuli.

Figure 3.7: Demonstration of pupil rotation when one subject is viewing four visual stimuli that lays in a row. (a)-(d) correspond to the four stimuli.

Figure 3.8: Plot of pupil rotation angle when one subject is viewing a set of visual stimuli organized as a $4 \times 4$ grid.

_Pupil size:_ Pupil size when a user is viewing the $4 \times 4$ visual stimuli is plotted in Fig. 3.9. The plot also has a stair-like shape, which means the pupil size distinctly changes toward the visual stimuli location. However, the stairs are steep rather than flat. This means that
the pupil size is changing while the user is viewing a certain visual stimulus. This change is caused by many factors like the projection angle to the image, light condition, emotion condition, and cognition condition. Thus, it is impractical to use pupil size as a feature to indicate the difference of the visual stimuli.

Even though the pupil size varies during visualization of a certain visual stimulus, we found that the ratio between the major axis and the minor axis of the pupil ellipse remains constant. This makes it a proper feature to characterize the eye movement. The axis ratio of the pupil ellipse is plotted in Fig. 3.10 where the data has been centered. The plot has a stair-like shape with a distinctive stair height and roughly planar stair surface. Note that even the feature quality of the axis ratio is much better than the original pupil size. There are cases where it fails to characterize the eye movements, such as in row 4.

3.4.3.2 Stimuli along a Radial Line

In this study, the visual stimuli were arranged along a line. This line started from the middle of two eyes. Thus, those visual stimuli can be approximately treated as along one gaze vector. Visual stimuli on three lines (gaze vectors) were examined. One was the central
gaze vector when the subject looked straight forward, and the other two were generated by horizontally rotating the central gaze vector 7.25° to the left and right, respectively. The visual stimuli were arranged from 35 cm to 265 cm along the gaze vector direction. A specific feature, pupil distance, is examined in this study.

Pupil distance has a strong indication of the distance from the user to the visual stimuli when the visual stimuli are aligned on one gaze vector. When the visual stimulus is close to the user, the eyes move toward each other to focus on it which results in a smaller pupil distance. In contrast, when the user gazes at an object further, eyes move in the opposite way and the pupil distance gets larger. This phenomenon is illustrated with pupil images in Fig. 3.11. The pupil distance collected in this scenario is plotted in Fig. 3.12.

Figure 3.11: Demonstration of pupil distance changes when the visual stimuli are further away. Two green dashed lines are reference and aligned vertically with the pupil locations when the visual stimulus is at 35 cm away. The image are flipped, which makes it like the pupil distance is declining when the visual stimulus is further away.

In the plot of pupil distance (Fig. 3.12), it shows a stair-like shape with a planar stair surface, which means the pupil distance remains stable during the period when the user is looking at each visual stimulus. However, the stairs are only distinctive at a close range, and it becomes difficult to distinguish them when the stimuli are further away. This suggests that, within a close range, the pupil distance can strongly and distinctively indicate the distance of the visual stimulus; however, this indication becomes weaker and more ambiguous when the
user is looking somewhere further away. This result can also prove the third characteristic of the gaze vector method that there is a relatively clear relationship between pupil features with the 3D when the gaze vector is determined, which can improve the estimation accuracy.

3.4.3.3 Stimuli Lying Horizontally and Vertically

Two sets of visual stimuli lying horizontally and vertically, respectively, on a plane were viewed by the subjects, shown in Fig. 3.13. The plane was placed facing the user at three depths: 50 cm, 80 cm, and 110 cm, respectively. The distance of two adjacent visual stimuli is 7.62 cm (3 inches). The pupil positions on the image when the subject was viewing those stimuli are plotted in Fig. 3.14. Even when the visual stimuli were aligned along a straight line, the pupil positions formed a curve on the captured image due to the nature of eye ball rotation and the camera projection effect. In addition, it is noticed that when the stimuli are further away, the change of the pupil position from one stimulus to another gets smaller. This curved pupil position pattern makes it difficult to model and estimate the visual target position using the pupil captured on image. To model this nonlinear relation, a sufficient amount of data is required to precisely characterize the eye movement pattern.
The minimal amount of data that is required to capture the pupil movement pattern defines the number of points needed in the calibration process. If calibration points are not enough, the data cannot capture the pupil movement pattern and will lead to a large estimation error. However, increasing this number increases the user’s workload and calibration time, which results in the system being cumbersome and impractical. To find the appropriate number of calibration points that balances the accuracy and calibration complexity, simulations with various point numbers to capture the pupil pattern were conducted, and the results are illustrated in Fig. 3.15. The condition in which there are only two calibration points is demonstrated in Fig. 3.15(a). The pupil position pattern is treated as a linear relation, which is far from its actual condition, especially for the horizontal visual stimuli. With more calibration points, the captured pupil pattern gets closer to the actual condition. When the number of calibration points reaches four, it can closely represent the pupil position pattern. Thus, for the best tracking accuracy and minimal calibration efforts, four calibration points are selected for our gaze tracking system.

3.4.3.4 Eye Tracking Resolution

It has been observed that the changes of pupil features become subtle when the visual stimuli are further away. This phenomenon is caused by the motion nature of the eye ball.
When the visual target is further away, the rotation of the eye ball required for concentration becomes smaller, and the changes of eye features captured by the eye tracker become more subtle. Whether the eye tracker is capable of distinguishing those subtle eye movements and how distinctive the eye movements can be reflected from the eye tracking data significantly affect the 3D gaze tracking accuracy and tracking space. With respect to this consideration, \textit{eye tracking resolution} is introduced to quantitatively evaluate an eye tracker’s capability of measuring subtle eye movements.

The eye tracking resolution $R$ is defined as a ratio of the pupil position change on the captured image (in pixel) over the position change between two visual stimuli (cm). At a certain depth $d$ (cm), when eyes shift from a visual stimulus $A$ to $B$, on the captured
image the pupil shifts from location $a$ to location $b$ correspondingly. Accordingly, the eye tracking resolution $R_d$ at depth $d$ can be computed using (3.5). The larger this ratio is, the better tracking accuracy that can be achieved. Generally, the pupil position changes are different in horizontal and vertical directions. Thus, there are $r_d^V$ and $r_d^H$ corresponding to the vertical eye tracking resolution and horizontal eye tracking resolution, respectively. The eye tracking resolutions of the current prototype are calculated and summarized in Table 3.1. It shows that if the visual stimulus moves 1 cm horizontally at a depth of 50 cm, the pupil on the image will move 4.6 pixels correspondingly. However, $r_d^V$ declines to 1.5 pixel/cm at a depth of 110 cm, which is not sufficient to provide distinctive eye movement tracking, due to effects of microsaccades (small, jerk-like, involuntary eye movements during visual fixation at a specific position). Therefore, for accurate gaze estimation, the eye tracking resolution should be distinctive.

$$r_d = \frac{|a - b|}{|A - B|} \quad (3.5)$$

<table>
<thead>
<tr>
<th>Unit (pixel/cm)</th>
<th>Depth of 50 cm</th>
<th>Depth of 80 cm</th>
<th>Depth of 110 cm</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_d^H$</td>
<td>4.6</td>
<td>2.9</td>
<td>2.1</td>
</tr>
<tr>
<td>$r_d^V$</td>
<td>3.4</td>
<td>2.1</td>
<td>1.5</td>
</tr>
</tbody>
</table>

3.4.3.5 Effective Range of Pupil Distance

To detect the subtle eye movements when the visual targets are further away, a high-resolution image sensor is required to amplify the subtle eye movements on the image. However, given a high-resolution image sensor, there is always a range limit in which only the eye tracker can capture distinctive eye movements. This range can be illustrated from the change of pupil distance when the visual stimulus moves further away. In Fig. 3.16, the pupil distances against the distance of the visual stimuli are plotted. As shown in Fig. 3.16,
when the visual stimuli are closer (<100 cm), the pupil distance change from one position to another is relatively more evident, while this change becomes smaller when the stimulus moves further away. Thus, effective range of pupil distance is defined to indicate the range limit of an eye tracker, which defines an eye tracker’s potential workspace for 3D gaze tracking. For our current prototype, its effective range of pupil distance is considered to be 100 cm. Upgrading the image sensors to have a higher resolution could increase the range.

![Figure 3.16: Plot of pupil distance with respect to the distance of the visual stimulus. The visual stimuli are along three different gaze vectors, left axis, central axis, and right axis.](image)

3.4.3.6 **Glints**

In the current system, features related to glints are not considered in the estimation of the visual axes or the gaze distance. Glints are widely used in virtual gaze tracking systems as a reference to measure pupil movements. In most of those systems, glints were assumed to appear at the same location on the cornea, which is based on an assumption that the eye ball is an ideal sphere. Further visual behavior studies are needed to identify the quality of features related to glints (the degree of distinction and stability) for 3D gaze tracking.
3.4.4 Estimation based on Neural Networks

Neural Networks (NNs) [38, 56, 57] are used to build the mapping relations $M_L$, $M_R$, and $M_{DL}$ in order to estimate the left and right visual axes and gaze distance. NN with one hidden layer was chosen because of its advanced capability of modeling highly nonlinear relationships. Based on previous eye behavior studies, pupil position (X, Y component), axis ratio of the pupil ellipse, and pupil rotation angle are selected as the input features for estimating each eye’s visual axis. Afterward, the gaze vector is computed following (3.6)-(3.12) introduced in Appendix. Then the computed gaze vector is used as a new feature to estimate the gaze distance along with other eye features including both eyes’ pupil positions, axis ratios of pupil ellipse, pupil rotation angle, and pupil distance. Before the NNs can perform the estimation, they need to be trained with the data collected in a calibration process.

3.5 Experiments

Thirty subjects participated in the evaluation experiment. A calibration process was performed where the subjects were asked to look at $4 \times 4$ grid of visual stimuli on a plane, which was placed on 4 different depths. This setup is demonstrated in Fig. 3.17. The calibrated space was 27 cm wide (along the Y-axis) and 27 cm high (along the X-axis) with a depth (along the Z-axis) ranging from 60 cm to 100 cm (volume $29160 \text{ cm}^3$). Four calibration points were used to capture the curved pattern of the pupil movement. The experiment was conducted in accordance with the Code of Ethics of the World Association. Prior to participating in this study, a short introduction was provided to the subjects, including the involved technology, the system setup, and the purpose of this study.

For thorough testing, 3D gaze estimation inside and outside of the calibration space were carried out, respectively. Within the calibrated space a set of visual stimuli with known positions were viewed one by one by the participants. The estimated 3D gaze using the gaze vector method was compared with their actual locations. For judicial evaluation, during the
inner testing there was only a small portion (22%) of the visual stimuli that belonged to those used for calibration, and they were collected in a revisit manner.

In a real environment, the visual field of a human is open. For all of the reported 3D gaze tracking methods based on regression, the calibration was only carried out on a small visual space. Situations where the user looks outside the calibration space happen frequently. How well the 3D gaze tracking system can handle those situations is a critical criterion for evaluating the system’s robustness. In light of this consideration, a set of visual stimuli outside the calibrated space was tested. The extra testing visual stimuli were placed on the left side 9 cm away, on the right side 9 cm away, and 10 cm further beyond the calibration space, which made the tested space to be 45 cm wide, 27 cm high, and 50 cm long (workspace volume 60750 cm$^3$).

3.6 Experiment Results and Discussion

3.6.1 Overall Error

The average error of the tests inside the calibration space was 2.4 cm, and for the tests outside it was 7.5 cm. The errors are summarized in Table 3.2. It includes the gaze vector error, 3D gaze error on each axis and its overall error, and each error’s standard deviation.
Table 3.2: Error Summarization of 3D gaze tracking in a real environment using gaze vector method. The error with the standard deviation along each axis and the overall error are summarized for the entire test space and for each test depth. The unit of the gaze error is cm and the gaze vector error is in degree.

<table>
<thead>
<tr>
<th>Depth (cm)</th>
<th>Tests with points inside the calibration space</th>
<th>Tests with points outside the calibration space</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Gaze vector (degree)</td>
<td>X (cm)</td>
</tr>
<tr>
<td>Average</td>
<td>0.56±0.31</td>
<td>0.8±0.5</td>
</tr>
<tr>
<td>60</td>
<td>0.44±0.17</td>
<td>0.4±0.3</td>
</tr>
<tr>
<td>70</td>
<td>0.49±0.38</td>
<td>0.5±0.5</td>
</tr>
<tr>
<td>80</td>
<td>0.54±0.31</td>
<td>0.8±0.4</td>
</tr>
<tr>
<td>90</td>
<td>0.72±0.59</td>
<td>1.4±1.4</td>
</tr>
<tr>
<td>100</td>
<td>0.76±0.25</td>
<td>0.9±0.6</td>
</tr>
<tr>
<td>110</td>
<td>1.03±0.69</td>
<td>2.5±1.4</td>
</tr>
</tbody>
</table>
For both tests inside and outside of the calibration space, the major error of the estimation was along the depth direction (Z axis). Inside the calibration space, the error of the estimated 3D gaze had a clear deterioration tendency when the visual stimuli were further away.

Outside the calibration space, the testing error was much greater than that inside of the calibration space, and the worst case was 11.9 cm offset from the actual visual stimuli positions at depth 80 cm.

3.6.2 Comparison of Gaze Vector Method with Visual Axes Intersection Method

The gaze vector method was compared with the traditional visual axes intersection method, and their performance within the calibration space is summarized in Table 3.3. The gaze vector method had an average error of 2.4 cm, while the intersection method had an average error of 8.9 cm. The results show that both methods have major error on the Z-axis, which is along the depth direction. In addition, the maximum error of the gaze vector method was 6.7 cm, which was much less than the maximum error of 31.9 cm of the intersection method.

Table 3.3: Error comparison between gaze vector method and the visual axes intersection method for 3D gaze estimation. Max: maximal error, Ave: average error, Min: minimal error.

<table>
<thead>
<tr>
<th>Unit: cm</th>
<th>Gaze vector method</th>
<th>Visual axes intersection method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Max</td>
<td>Min</td>
</tr>
<tr>
<td>e on X</td>
<td>2.4</td>
<td>0.2</td>
</tr>
<tr>
<td>e on Y</td>
<td>1.0</td>
<td>0.1</td>
</tr>
<tr>
<td>e on Z</td>
<td>6.6</td>
<td>0.6</td>
</tr>
<tr>
<td>Overall</td>
<td>6.7</td>
<td>0.8</td>
</tr>
</tbody>
</table>

3.6.3 Gaze Vector Error Reduction

The average error of the visual axes and gaze vector are summarized in Table 3.4. The gaze vector error was less than the mean of the visual axes error. The error reduction rate\(^8\)

\(^8\)Error reduction rate is calculated by $1 - \{\text{gaze vector error}\}/\{\text{mean visual axis error}\}$. 

59
is 34.50%. This means the gaze vector has 34.50% less error compared to the visual axes, which are used to compute the gaze vector. This finding is consistent with our previous analysis demonstrated in Fig. 3.4.

Table 3.4: Error summarization of the estimated visual axes, $v_L$ and $v_R$, and the computed gaze vector, $v_g$.

<table>
<thead>
<tr>
<th></th>
<th>$v_L$</th>
<th>$v_R$</th>
<th>$v_g$</th>
<th>Error Reduction Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error</td>
<td>1.08±0.55°</td>
<td>0.63±0.32°</td>
<td>0.56±0.31°</td>
<td>34.50%</td>
</tr>
</tbody>
</table>

3.6.4 Various Settings of the Neural Network

The hidden layer in the NN model is an important component that can significantly affect the estimation of the gaze vector and gaze distance. To investigate its effect and find a better NN configuration, hidden layers with different numbers of hidden units were tested. Table 3.5 summarizes the performance of the system when the visual axis estimator has various hidden units. The changes of the hidden units first affect the estimation of the visual axes and then the gaze vector, which eventually affects the estimation accuracy of the 3D gaze. When the number of hidden units is between 4 and 7, there is not much difference on the gaze vector error for both testing cases. When the hidden layer has 13 hidden units or more, the gaze vector error increases. It may indicate the occurrence of overfitting during the training of NNs. Thus, for the optimal estimation, the number of hidden units can be chosen between 4 and 7. Though the error of gaze vector increases, it does not much affect the final 3D gaze estimation. This is because of a characteristic of the gaze vector method that, when the gaze vector has a small error, the error of the 3D gaze mainly depends on the error of the gaze distance. Table 3.6 summarizes the performance of the system when the gaze distance estimator has various numbers of hidden units. The results indicate changing the hidden units does not have distinctive effects to the gaze distance estimation inside the calibration space. However, it does cause greater gaze errors outside the calibration space.
Moreover, the error of the 3D gaze is very close to the error of the gaze distance, which is consistent with the analysis of the gaze vector method.

Table 3.5: Error summarization when the visual axes estimator has various settings in Neural Networks.

<table>
<thead>
<tr>
<th>Hidden Units #</th>
<th>Inside the calibration space</th>
<th>Outside the calibration space</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ω of gaze vector</td>
<td>ρ of 3D gaze (cm)</td>
</tr>
<tr>
<td>2</td>
<td>0.70°</td>
<td>3.2</td>
</tr>
<tr>
<td>4</td>
<td>0.56°</td>
<td>2.4</td>
</tr>
<tr>
<td>5</td>
<td>0.57°</td>
<td>3.0</td>
</tr>
<tr>
<td>7</td>
<td>0.61°</td>
<td>2.5</td>
</tr>
<tr>
<td>10</td>
<td>0.67°</td>
<td>3.0</td>
</tr>
<tr>
<td>13</td>
<td>0.90°</td>
<td>3.5</td>
</tr>
<tr>
<td>15</td>
<td>0.91°</td>
<td>3.1</td>
</tr>
<tr>
<td>18</td>
<td>0.74°</td>
<td>3.0</td>
</tr>
<tr>
<td>21</td>
<td>0.73°</td>
<td>3.1</td>
</tr>
<tr>
<td>25</td>
<td>0.86°</td>
<td>3.7</td>
</tr>
</tbody>
</table>

3.6.5 Comparison with Existing 3D Gaze Tracking Systems

A comparison between our 3D gaze tracking system and the reported systems is summarized in Table 3.7. This comparison focuses on hardware specification, tracking workspace, tracking accuracy, and the estimation method. Our gaze vector method does not require sophisticated knowledge of eye geometrical shape or precise camera parameters, which require large amounts of time and special hardware to achieve. The presented system is validated in a large workspace, and a high degree of accuracy is achieved. Within the calibrated space, the average error is only 2.4 cm. Over the entire tested workspace, combining the calibrated space and non-calibrated space, our system achieved an average error of 4.6 cm. Its accuracy outperforms all the reported 3D gaze tracking systems. Note that the only reported system that performs closely to our system is the one in [33]. However, a calibration over the entire workspace was required in [33], while only a partial of the entire workspace was calibrated in our system.
Table 3.6: Error summarization when the gaze distance estimator has various settings in Neural Networks.

<table>
<thead>
<tr>
<th>Hidden Units #</th>
<th>Inside the calibration space (cm)</th>
<th>Outside the calibration space (cm)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\epsilon$ of gaze distance</td>
<td>$\rho$ of 3D gaze</td>
</tr>
<tr>
<td>2</td>
<td>2.7</td>
<td>2.9</td>
</tr>
<tr>
<td>4</td>
<td>2.4</td>
<td>2.7</td>
</tr>
<tr>
<td>5</td>
<td>2.1</td>
<td>2.1</td>
</tr>
<tr>
<td>7</td>
<td>2.5</td>
<td>2.7</td>
</tr>
<tr>
<td>10</td>
<td>3.0</td>
<td>3.2</td>
</tr>
<tr>
<td>13</td>
<td>3.4</td>
<td>3.6</td>
</tr>
<tr>
<td>15</td>
<td>3.2</td>
<td>3.4</td>
</tr>
<tr>
<td>18</td>
<td>3.0</td>
<td>3.2</td>
</tr>
<tr>
<td>21</td>
<td>3.0</td>
<td>3.2</td>
</tr>
<tr>
<td>25</td>
<td>3.0</td>
<td>3.2</td>
</tr>
</tbody>
</table>

3.7 Conclusion

In this chapter, a novel 3D eye tracking system is introduced. It includes a wearable binocular eye tracking platform and a novel gaze vector method for 3D gaze estimation in a real environment. This system does not need sophisticated eye geometry or precise camera parameters, which effectively simplifies the calibration and usage procedures. Based on our hardware platform, we explicitly studied the pupil patterns on an image when a human is viewing the visual stimuli in a real 3D environment. From this study, the representative and stable eye features are selected to characterize the eye movements. Meanwhile, eye tracking resolution and effective range of pupil distance are introduced as two new criteria for evaluating an eye tracker hardware, which are critical for 3D gaze tracking accuracy and tracking space. Neural Networks are employed to estimate the visual axes and gaze distance from those selected eye features. With all these efforts, the best 3D gaze tracking accuracy among reported systems is achieved. We hope that our work can inspire more research efforts on 3D gaze tracking and provide guidance on how to design the eye tracker, how to select features, and how to create the estimation algorithm.
Table 3.7: Comparison of our 3D gaze tracking system using gaze vector method with existing systems.

<table>
<thead>
<tr>
<th>Unit (cm)</th>
<th>The presented system using gaze vector method based on learning</th>
<th>Nennessey etc. using visual axes intersection based on 3D gaze model [32]</th>
<th>Abbott etc. using visual axes intersection based on regression method [33]</th>
<th>Lee etc using intersection of single visual axis and a depth plane [38]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inside the calibration space</td>
<td>Overall (inside plus outside)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sophisticated eye geometry</td>
<td>Not needed</td>
<td>Needed</td>
<td>Not needed</td>
<td>Not needed</td>
</tr>
<tr>
<td>Precise camera parameters</td>
<td>Not needed</td>
<td>Needed</td>
<td>Not needed</td>
<td>Not needed</td>
</tr>
<tr>
<td>Tested workspace (Width × Height × Depth)</td>
<td>27 × 27 × 40 (29160 cm³)</td>
<td>45 × 27 × 50 (60750 cm³)</td>
<td>30 × 23 × 25 (17250 cm³)</td>
<td>47 × 27 × 54 (68526 cm³)</td>
</tr>
<tr>
<td>Depth range from the user</td>
<td>60 - 100</td>
<td>60 - 110</td>
<td>17.5 - 42.5 (estimated by [33])</td>
<td>54 - 108</td>
</tr>
<tr>
<td>Euclidean distance error (overall in the entire workspace or at a certain depth)</td>
<td>2.4 (overall)</td>
<td>3.93 (overall)</td>
<td>5.8 (overall)</td>
<td>0.96° on the X-axis</td>
</tr>
<tr>
<td>1.3 (60)</td>
<td>3.62 (17.5)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.9 (70)</td>
<td>3.35 (22.5)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.3 (80)</td>
<td>3.75 (27.5)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.6 (90)</td>
<td>3.98 (32.5)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.5 (100)</td>
<td>4.28 (37.5)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.61 (42.5)</td>
<td>4.59 cm on the Z-axis (overall)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
3.8 Acknowledgment

Support for this research from US NSF award 1414299 is gratefully acknowledged.

3.9 Appendix: Computation of Gaze Vector From Visual Axes

The gaze vector $v_g$ is calculated using (3.6)-(3.12) from the intersection of two visual axes. Two visual axes intersect with their common normal at $i_L$ and $i_R$, which can be defined parametrically as (3.6) and (3.7). The direction vector of the common normal, $v$, can be calculated as the cross product of the left and right visual axes using (3.8). In addition, $i_L$ and $i_R$ have the relationship as described in (3.9). $s$, $t$, and $m$ are the scalar coefficients.

\begin{align*}
  i_L &= e_L + s \cdot v_L & (3.6) \\
  i_R &= e_R + t \cdot v_R & (3.7) \\
  v &= v_L \times v_R & (3.8) \\
  i_L &= i_R + m \cdot v & (3.9)
\end{align*}

From above definitions, a linear relationship can be formulated as (3.10). Solving it produces a unique solution set of $s$, $t$, and $m$, correspondingly, the unique intersection of the common normal with each visual axis can be calculated as (3.6) and (3.7). Thus, the center of the common normal $c_c$ and the gaze vector can be calculated using (3.11) and (3.12), respectively.

\begin{align*}
  s \cdot v_L - t \cdot v_R + m \cdot v &= e_L - e_R & (3.10) \\
  c &= \frac{i_L + i_R}{2} & (3.11) \\
  v_g &= \frac{c}{|c|} & (3.12)
\end{align*}
CHAPTER 4
3D-GAZE-BASED ROBOTIC GRASPING THROUGH MIMICKING HUMAN VISUOMOTOR FUNCTION FOR PEOPLE WITH MOTION IMPAIRMENTS

A paper published at IEEE Transactions on Biomedical Engineering

Songpo Li\textsuperscript{10}, Xiaoli Zhang\textsuperscript{11}, and Jeremy D. Webb\textsuperscript{12}

4.1 Abstract

Objective: The goal of this chapter is to achieve a novel 3D-gaze-based human-robot-interaction modality, with which a user with motion impairment can intuitively express what tasks he/she wants the robot to do by directly looking at the object of interest in the real world. Toward this goal, we investigate 1) the technology to accurately sense where a person is looking in real environments and 2) the method to interpret the human gaze and convert it into an effective interaction modality. Looking at a specific object reflects what a person is thinking related to that object, and the gaze location contains essential information for object manipulation.

Methods: A novel gaze vector method is developed to accurately estimate the 3D coordinates of the object being looked at in real environments, and a novel interpretation framework that mimics human visuomotor functions is designed to increase the control capability of gaze in object grasping tasks.

Results: High tracking accuracy was achieved using the gaze vector method. Participants successfully controlled a robotic arm for object grasping by directly looking at the target object.

\textsuperscript{9}Reprinted with permission of IEEE Transactions on Biomedical Engineering, 2017.
\textsuperscript{10}Primary researcher and author, graduate student, Department of Mechanical Engineering, Colorado School of Mines.
\textsuperscript{11}Author for correspondence, Assistant Professor, Department of Mechanical Engineering, Colorado School of Mines.
\textsuperscript{12}Graduate student, Department of Mechanical Engineering, Colorado School of Mines.
**Conclusion:** Human 3D gaze can be effectively employed as an intuitive interaction modality for robotic object manipulation.

**Significance:** It is the first time that 3D gaze is utilized in a real environment to command a robot for a practical application. 3D gaze tracking is promising as an intuitive alternative for human-robot interaction especially for disabled and elderly people who cannot handle the conventional interaction modalities.

Index Terms: 3D gaze tracking, assistive robot, robotic grasping, gaze control

### 4.2 Introduction

Assistive robots have drawn significant research attention in recent decades. These assistive robots can be employed to facilitate and enhance a person’s daily living by providing assistance with tasks like cooking [58, 59], doing laundry [60, 61], performing bed baths [62], assisting with walking [63, 64], etc. Because assistive robots are designed with advanced capabilities, they can accomplish a large number of complex tasks. Conducting these systems, however, becomes cumbersome with traditional control interfaces consisting of buttons, switches, knobs, and joysticks. Moreover, new challenges arise in designing HRI due to the fact that a large portion of the target users is disabled or elderly. The question of how the human user can effectively and efficiently control these systems has drawn significant attention in robotic research.

To facilitate the interaction between human users and assistive robots, researchers have been exploring natural, interpersonal communication signals and biosignals employed inside the human body, and are attempting to enable the user to intuitively convey control commands to assistive robots using these natural signals. Substantial research has been conducted on signals like speech [65, 66], facial expression [67, 68], body gesture [69, 70], electromyography (EMG) signal from muscles [71, 72], and electroencephalogram (EEG) signal from the brain [73–75]. However, there is another promising natural signal, gaze signal, which has not been given sufficient attention in HRI.
Gaze is defined as where a person is looking, which is estimated from eye movements. It is natural, effortless, and rich in information to be employed as a communication signal between a human and a robot. Humans naturally look at objects that reflect what they are thinking and wanting. For example, when a person is thirsty and wants to drink, he/she will naturally look at drinking-related objects such as a bottle of water or a water fountain. Naturally, this object being looked at will be the manipulation target [76, 77]. This natural link between human gaze and human mind makes gaze an intuitive modality to express what a person wants. Gaze has been naturally used in human-human interaction; for example, one uses his/her gaze to guide another to an object of interest in order to build joint attention, which makes gaze an easily adaptable modality in human-robot interaction. Moreover, intentionally looking at an object is almost effortless, which is easily achievable for most human beings and especially so for those with severe disabilities [18, 78].

Although gaze is promising for intuitive HRI, the investigation of gaze-based HRI is limited by gaze tracking technology. In the 3D real world, the 3D coordinates of a person’s gaze needs to be explicitly estimated. This 3D gaze location can directly represent the location of the object and command a robotic system for certain types of operation. However, research of 3D gaze tracking in real environments is very rare, and the achieved accuracy is too low for practical applications. Currently, the major research of gaze estimation is still focused on estimating where a person is looking on either a two-Dimensional (2D) display screen or a 2D scene image that captures the view in front of the person. This 2D gaze position in image pixel loses the valuable information about the absolute coordinates of that visualized object.

Another problem of using gaze for HRI is the difficulty in interpreting the gaze signal for robot operation [16, 79]. Without appropriate interpretation, the 2D gaze has only been used as a simple trigger to trigger a predefined action of the robot, such as driving to the left one step or rotating a robotic joint certain degrees. This level of interpretation may be sufficient to express commands to change a state of the robot like opening and closing its grasper.
However, with the same interpretation method, commanding a robotic arm to reach a point becomes extremely frustrating due to continuous triggering of step commands. Moreover, the utilization difficulty extremely increases for complex tasks that need to specify an execution plan and results in it to be inapplicable. For example, for successful robotic grasping, the robot is necessary to know the object location and orientation and a sophisticated grasping plan; however, none of this information can be delivered by interpreting the gaze using previous methods. The 3D gaze contains rich information about the task and user’s high-level desire. How to extract this information and use it to facilitate robotic operation is still an open problem.

To adopt the gaze modality for intuitive HRI, in this chapter, we present our work on how to accurately sense where a person is looking and how to interpret this information to convert human gaze into an effective interaction modality. The goal of this project is to achieve a novel interaction modality with which a user can intuitively express what tasks he/she wants the robot to do, which include object grasping task, object retrieval task, or more complex manipulation tasks related to an object’s functionality, by naturally looking at that object in the real world. It is expected to benefit those users who have impaired mobility in their daily living and able-body users who need an additional hand in general working scenarios. We selected object grasping to demonstrate the usability of the 3D due to robotic grasping being a fundamental but very complex task; in fact, object grasping is associated with many control parameters [80] that need to be specified for successful grasping, and, in this chapter, we studied the grasping problem of a cuboid object. Particularly, the contributions of this chapter are:

1. A novel 3D gaze estimation method is developed and integrated into a binocular eye tracking system to accurately track a person’s 3D gaze in a real environment. The improved accuracy enables us to investigate 3D gaze for HRI, which is the first time 3D gaze is utilized in a real environment for a practical application.
2. A new framework is developed to interpret the 3D gaze in order to extract useful information that can facilitate the task. Unlike previous work where 2D gaze can only be used to trigger a command, the 3D gaze of humans can convey the location and pose of the target object that the user wants to manipulate as well as how the user wants the object to be manipulated. This procedure is similar to the human visuomotor function of using visual perception to localize an object, determine the pose of the object, and eventually carry out a plan for operation based on the perceived information.

3. A mathematical visuomotor grasping model is built, which models the coordination of human hand grasping motion and visual perception. Although there are some qualitative observations about the role of humans’ eye-hand coordination during object grasping, no mathematical models have been built to quantify their relationship. Our visuomotor grasping model can be used to predict how a user will grasp an object when he/she looks at a particular portion of the object. This model is validated to generate a proper grasping configuration that is human-like.

In addition, we experimentally evaluated the individual modules and the overall interaction framework to assess its usability and user acceptance. Furthermore, theoretical analysis was performed to assess the usability of some key modules systematically.

4.3 Related Work

4.3.1 3D Gaze Tracking in Real Environments

Even though gaze tracking has a long history and has been in a highly active phase of development for the past two decades [1, 3], most of this work focuses on gaze tracking for a 2D display or image, like the computer screen. Research on tracking 3D gaze in a real environment is very rare and has low accuracy, which makes it hard to put it into practical applications.

In 2009, Hennessey reported the first system for 3D gaze tracking in a real environment, which was based on a binocular table-stand eye tracking system [32]. They individually
estimated each eye’s visual axis, which was defined as a vector emitting from an eye’s center to the visual target. The intersection point of two visual axes was considered to be the location of 3D gaze (visual axes intersection method). In their test, over the entire workspace of 30 cm × 23 cm × 25 cm (width×height×depth) (workspace volume 17250 cm³), an average accuracy of 3.93 cm was achieved. In 2012, Abbott reported their 3D gaze tracking system [33], which was also based on the visual axes intersection method. An average error of 5.8 cm was achieved in a testing depth ranging from 54 cm to 108 cm in a 47 cm wide and 27 cm high workspace (workspace volume 68526 cm³).

The visual axes intersection method is very sensitive to the errors of the visual axes, which will severely propagate and result in a significant error for the 3D gaze when the visual target is far away. This phenomenon is referred to as the error propagation problem. The visual axes intersection method is based on the assumption that when a person visually concentrates on a target or a location in the 3D environment, the visual axes intersect at that location. However, due to the fact that the estimated visual axes contain certain angular errors, two visual axes do not intersect most of the time. Thus, in practical usage, the point that has the shortest squared distance to both visual axes is considered to be the intersection point of two visual axes and thus the estimated gaze point, which is the middle point of two visual axes’ common normal. During this process, the angular error of each visual axis will significantly deviate the intersection point from the visual target (error propagation problem), which causes large 3D gaze errors. Moreover, the error propagation problem gets much worse when the visual target is further away.

Another 3D gaze estimation system in real environments was reported by Lee in 2012, which was based on a monocular eye tracker [38]. They assumed the visual targets were on imaginary planes that were vertical and facing the user. They separately estimated one eye’s visual axis and plane location, and then intersected them to compute the 3D gaze location. This method was tested within a pyramid-shaped workspace whose base was a 10 cm × 10 cm square, height was 50 cm, and apex was at one eye’s center (workspace volume ≈1667
cm³). However, even in a setup where the testing points were the same as the calibration points, the experimental results showed low accuracy. On average, the estimated 3D gaze had an error of 4.59 cm in the depth direction itself, and there were also severe shifts in the horizontal and vertical directions.

Other than estimating the 3D gaze using only eye movements, researchers have also tried to use additional measurement devices to reconstruct the 3D location of 2D gaze, such as the RGB-D camera [39–41] or stereo cameras [43, 44]. However, due to the additional hardware, their systems were bulky, and the complexities of setup and usage were increased as well.

4.3.2 Human-Robot Interaction based on 2D Gaze

Gaze tracking has been used on studying human behaviors in various disciplines, such as evaluating a patient’s mental engagement during therapy [81]; evaluating skills of pilots [11], surgeons [82], and drivers [83] during training; and evaluating the design of a website [84] or advertisement [85]. Work that utilized the human gaze as a control modality was rare and based on 2D gaze. No research that utilizes 3D gaze in a real environment has been reported due to the poor tracking accuracy.

Researchers first utilized 2D gaze as a control modality to substitute the hand input for people who have impaired or weak arms. Several researchers reported the attempts of steering a wheelchair using gaze, which roughly converted the gaze location on a scene image to control commands of driving forward, backward, left or right [16, 79, 86]. A similar strategy was also applied to control the movement direction of mobile robots [87], quadcopters [88], and even robotic laparoscope systems [89]. A setup with on-screen buttons, which could be triggered by gazing at them, was reported in [90, 91]. These buttons were associated with certain motion commands that activated a robot to move one joint or its end effector in a defined direction. In our previous works, gaze was used to directly indicate the destination instead of explicitly triggering a series of motion commands to incrementally navigate the robot. In [76, 92, 93], gaze was used to define the concentration area of a robotic laparoscope system, and in [94], gaze was used to define the destination of a mobile ground
In all above attempts, the setups were for teleoperation, in which a computer screen was used in order to provide the user vision feedback, and on this screen, 2D gaze was tracked. Though this scenario is useful for many applications of teleoperation, situations where a user directly interacts with his/her surroundings in the real environment are more common. However, due to the difficulty of tracking gaze in a real environment, there is less research reported where gaze is utilized for robot interaction in real environments. Moreover, among this sparse reported research, the gaze direction was estimated and used only to roughly select a target when a user was looking toward that direction of an object [13, 95].

In all the aforementioned research, gaze functions as a trigger to select the target from a pool of certain objects or buttons. The functionality of gaze that can be utilized for HRI has not been fully investigated, especially when it is 3D gaze. As 3D gaze lies on the object, the gaze location can be used to 1) represent the location of the visualized object and 2) command a robot for simple manipulation, such as simply reaching a point on the object. However, for complex object manipulation, such as object grasping, only knowing the location of the object is not sufficient. To successfully grasp an object, the robot needs to know the location of the object, the pose of the object, and a sophisticated grasping plan. Whether 3D gaze has the capability to provide this essential information has not been reported and is worth investigation since it has the potential to significantly facilitate complex robotic manipulation.

4.3.3 Eye-Hand Coordination in Object Grasping

Human beings pose a highly developed ability of grasping objects under many different conditions, taking into account variations in location, structure, motion and orientation. This natural ability is called eye-hand coordination. Normally, a grasping plan is initiated before the hand actually reaches the target object [96]. This plan is regulated by the interaction of the eye, hand and arm control systems. For many years, researchers have been studying this process in attempts to discover the underlying mechanism that controls eye-
hand coordination in object grasping. These studies have found that gaze fixtures provide a strong cue for predicting the hand’s grasping configuration. For example, eyes temporally lead the hands for object grasping in order to provide additional inputs for planning further movements, which is so-called the predictive gaze [97]. Moreover, in most cases, the area on an object that the user gazes at is the area where contact with the thumb or index finger will occur during grasping [98–100]. However, currently only qualitative summaries of eye-hand coordination from experimental observations exist, and there is not a single quantitative model that represents this eye-hand coordination process effectively.

4.4 Methods

In order to achieve the interaction modality where a user can intuitively command a robot by naturally looking at the object, two major works are included in the chapter: 1) a novel 3D gaze estimation method and 2) a novel framework for 3D gaze interpretation. The system architecture is demonstrated in Fig. 4.1, in which the major contributions are highlighted with colors. A gaze vector method is developed and integrated on a binocular eye tracker to accurately estimate a person’s 3D gaze in real environments. Human eyes and hands are tightly correlated during task execution, where eyes provide guidance to the hand motion both spatially and temporally. To take advantage of this correlation for HRI, we interpret the 3D gaze by mimicking human visuomotor functions. Here, the 3D gaze is the effective measure (visual attention) of the visual perception of eyes. In a grasping task, a person localizes the operation target, determines its pose from the visual perception, and initializes a grasping plan based on the perceived information. We interpret the 3D gaze following the same routine, which is to extract essential information about the object that can facilitate the task execution. With this framework, 3D gaze is used to control the complex robotic grasping task instead of only using gaze to select the operation target as previous works.
4.4.1 Accurate 3D Gaze Tracking in a Real Environment

The 3D gaze tracking system is shown in Fig. 4.2, which can provide accurate 3D gaze tracking in real environments. It is built on a hardware platform of binocular eye tracker frame for 2D gaze tracking [46], which has been modified in our project by redesigning the camera mounts and adding extra light resources. It has two extendable mounts for image sensors, which can be adjusted to face each eye respectively for individuals. The image sensors offer a maximum frequency of 30 frames per second with a resolution of 640X480 pixels. A visible light filter is added on top of the image sensor, and four near infrared (IR) LEDs are mounted around each image sensor to illuminate the pupil. Under the IR light, the pupil appears to be dark, and there are reflections of the light sources on the iris, which are called glints. A pupil tracking module has been coded with the OpenCV image processing library [55] in C++ to process live eye video streams and detect the pupil in each frame with a shape-based algorithm. An ellipse is fitted to the pupil region to represent the pupil. Later, two pupils’ center positions, dimensions, rotation angles, and left and right pupils’
distance are extracted for 3D gaze estimation. The coordinate system of 3D gaze has been drawn in Fig. 4.2, in which the X-axis points down vertically, the Y-axis points to the left side horizontally, and the Z-axis (depth direction) points forward horizontally. The origin of the coordinates (the red dot) is at the middle of the two eyes.

Figure 4.2: The binocular eye tracker with coordinate definition.

A gaze vector method is developed and applied on the binocular eye tracker to estimate the 3D gaze, which is illustrated in Fig. 4.3. This method decouples 3D gaze estimation into the estimation of the gaze vector, $\mathbf{v}_g$, and the estimation of the gaze distance, $d_g$, along the gaze vector. The gaze vector is defined as a vector that emits from the middle point of the two eyes (the origin of the coordinate system) to the visual target. The gaze distance is the Cartesian distance from the origin to the visual target. The coordinates of the 3D gaze location, $g_3$, can be calculated as the production of $\mathbf{v}_g$ and $d_g$ in (4.1).

Instead of directly estimating the gaze vector from a trained mapping relationship, in the gaze vector method, the gaze vector is calculated as the vector from the middle point of two eyes to the intersection point of the left and right visual axes (5.2)-(4.5). Although using the intersection point of two visual axes to estimate the 3D gaze location leads to large Cartesian errors [32, 33], it is a great way to compute the gaze vector. This procedure can effectively alleviate the errors of two visual axes. A gaze vector with high accuracy can be computed using this method even when two estimated visual axes have large errors. In this process, the left and right visual axes, $\mathbf{v}_L$ and $\mathbf{v}_R$, can be estimated through pre-trained mapping relationships $M_L$ and $M_R$ with each eye’s features as inputs (5.2)-(5.3). Eye features of the
pupil’s center position, dimension, and rotation angle from the left and right eye are used as $x_L$ and $x_R$, respectively. The intersection point is calculated through a standard line intersection procedure that is notated as $I(v_L, v_R)$ (4.4), and the gaze vector is calculated as (4.5).

To estimate the gaze distance, $d_g$, the computed gaze vector is combined with the pupil distance, which is the distance from the left pupil’s center to the right pupil’s center, and previous eye features $x_L$ and $x_R$ (6). The pupil distance is annotated as $ps$. Combining $v_g$ and $ps$ to estimate the gaze distance is inspired by our experimental observation that, for a set of visual stimuli along the same gaze vector, there is a clear relationship between the distance of two pupil centers and the distance of a visual stimulus. Thus, we believe that using the $v_g$ and $ps$ together as inputs to estimate the distance of a visual stimulus can achieve a better accuracy. In addition, Neural Networks [101] with two layers are used to learn the mapping functions $M_L$, $M_R$, and $M_D$, respectively, due to their decent capability of learning highly nonlinear relations from given data. A calibration process is performed in order to collect training data to train the Neural Networks.

Figure 4.3: The illustration of the gaze vector method with the decoupled gaze vector and gaze distance. In the illustration, $e_L$ and $e_R$ are the positions of left and right eyes, $v_L$ and $v_R$ are the left and right visual axes, $i_L$ and $i_R$ are the intersections of the two visual axes with their common normal, and the middle point of $i_L$ and $i_R$ is considered as the intersection of two visual axes.
\[ g_3 = v_g \cdot d_g \]  \hspace{1cm} (4.1)
\[ v_L = M_L(x_L) \]  \hspace{1cm} (4.2)
\[ v_R = M_R(x_R) \]  \hspace{1cm} (4.3)
\[ c = I(v_L, v_R) \]  \hspace{1cm} (4.4)
\[ v_g = \frac{c - 0}{|c - 0|} \]  \hspace{1cm} (4.5)
\[ d_g = M_D(v_g, ps, x_L, x_R) \]  \hspace{1cm} (4.6)

4.4.2 3D Gaze Interpretation - Mimicking the Human Visuomotor Function during Grasping

The 3D gaze interpretation includes two phases, **Phase I** and **Phase II**, which mimics the human visuomotor function during grasping. In **Phase I**, the location of the object and its pose are estimated from the 3D gaze data, and in **Phase II**, a grasping plan is generated from a visuomotor grasping model learned from humans with a grasping location that is assigned by the user’s final stable fixation on the object.

4.4.2.1 Phase I - Localization and Pose Estimation for the Operation Target

While a person is looking at the cuboid object, the estimated 3D gaze points lie on the surface of the object. Thus, the location of this object can be measured by the 3D gaze in the real environment. A user needs to look at four points on the object with one at each corner region of the object, which forms a rectangle that is approximately located at the center of the object. The mean of the 3D gaze on each set of these points is used to represent the approximate center of the object. An adaptive sliding window filter [76] is applied to the raw gaze data to filter out the noises, which is caused by unconscious eye movements. The pose of the object is estimated by fitting the gaze points to a plane \( z = ax + by + c \), where \( a, b, \) and \( c \) are plane parameters. Then, the direction of the plane is calculated to represent
the pose of the object.

4.4.2.2 Phase II - Visuomotor Grasping Planning

The aim of this section is to infer how the user wants to grasp the object when he/she intends to grasp at a particular location on the object. This inference of the grasping configuration is achieved by adopting a human visuomotor model in grasping. This visuomotor grasping planning procedure mimics human eye-hand coordination in grasping, and the underlying mechanism is learned from human grasping and adopted by the robotic system, shown in Fig. 4.4. The grasping planning process is to determine the most probable robotic grasping configuration when the user intends to grasp a particular location of the object. The output grasping configuration is expected to be a successful grasping instance and to be similar to a grasping configuration performed by a human (human-like grasping).

![Figure 4.4: Adoption of human eye-hand coordination model in grasping to the robot.](image)

During object grasping, when a person looks at \(p_g\) on the object, it indicates that he/she intends to grasp the object at \(p_g\), and the thumb of the person will be put at or around this location while grasping. This is the qualitative description of the human eye-hand coordination. It results in a grasping configuration of \(p_c^H\) and \(\theta_g^H\), where \(p_c^H\) is the contact point of the hand with the object and \(\theta_g^H\) is the approaching angle of the hand. \(p_c^H\) and \(\theta_g^H\) are described in a local coordinate system, \(\Gamma\), which is defined along the edge of the object, demonstrated in Fig. 4.4. The line that passes though \(p_c^H\) and \(p_g\) intersects the \(-\Gamma\) axis with an angle \(\theta_g^H\), and this relationship can be notated as \(R(p_g, \theta_g^H, p_c^H)\).
A Gaussian Mixture Model (GMM) is used to model this eye-hand coordination behavior while grasping. GMM is a probabilistic model, which can handle the variances and uncertainty in object grasping. It models the correlation between \( p_c^H \) and \( \theta_g^H \) as (5.7), which is the probability of grasping the object with a hand approaching angle \( \theta_g^H \) and coming in contact at point \( p_c^H \). In the case that the most probable configuration is not applicable in an actual situation, the GMM model can easily find other alternatives. Moreover, with the learned GMM, a Gaussian Regression Model (GRM) can be easily obtained through a standard routine, which can both clearly demonstrate the correlation and be understandable for human researchers. This GMM contains \( K \) Gaussian models, and \( \omega_i, \mu_i, \) and \( \Sigma_i \) are the mixture weight, mean and variance of each Gaussian model \( (i \in [1, K]) \), respectively. The number \( K \) is determined empirically, while other parameters are learned through an Expectation-Maximization (EM) process. \( p_c^H \) and \( \theta_g^H \) are results of the human intent, grasping the object at point \( p_g \). When the user is looking at \( p_g \) on the object in order to command the robot for grasping, the final grasping configuration is determined by (5.8), which results in a combination of \( p_c^R \) and \( \theta_g^R \) that has the greatest probability. Then, the contact point and hand approach direction in the local coordinate system is converted to the global frame for the robot to execute the grasping task. An appropriate approaching trajectory is carefully designed so that the robot can reach the object with the given contact point and hand approach direction without accidentally knocking the object away.

\[
p(p_c^H, \theta_g^H) = \sum_{i=1}^{K} \omega_i N(\mu_i, \Sigma_i) \tag{4.7}
\]

\[
(p_c^R, \theta_g^R) = \text{argmax}(p(p_c^H, \theta_g^H)|R(p_g, \theta_g^H, p_c^H), p_g) \tag{4.8}
\]

### 4.4.3 System Integration

The 3D-gaze-based HRI framework was implemented based on Robotic Operation System (ROS), which allows different modules to be tested, modified, and integrated easily. The major modules in this project were the following: the binocular tracking module for
processing the eye images to detect the pupil and extracting the pupil information; the 3D gaze estimation module for estimating the 3D gaze point using the gaze vector method and filtering the 3D gaze points to remove the noise; the grasping planning module which generates the grasping plan based on the visuomotor grasping model and a designated grasping point selected by the user’s gaze; and the robot driving module to control the motion of the robot to properly approach the object and grasping it with the grasping plan, as shown in Fig. 4.1. A Mico robotic arm from Kinova Robotics was used to perform the grasping task under the control of 3D gaze.

4.5 Experiments

Individual modules were separately evaluated before evaluating the overall 3D-gaze-based HRI framework. Thirty subjects in total participated in the experiments. Prior to the experiments, a short introduction was provided to those subjects, including technologies involved, the system setup, and the purpose of the study. Throughout the experiment, a head stand was used to hold the subject’s head still. In practical applications, a user tracking module could be added to track the movement of the user and user’s head, which is not covered in this chapter.

4.5.1 3D Gaze Estimation

The performance of the 3D gaze tracking system was first examined. A calibration process was carried out with 64 calibration points for training the mapping relationships $M_L$, $M_R$, and $M_D$. During the calibration, each participant wore the eye tracker and was asked to look at a set of points in a 4×4 grid on a plane that was 27 cm wide and 27 cm tall, as shown in Fig. 4.5. The plane was placed at four different depths ranging from 60 cm to 100 cm, which gave a total workspace of $29160 \text{ cm}^3$. The visual axes, the gaze vector, and the gaze distance were computed and used to build the correlation with the pupil features. In the testing, subjects were asked to view another set of points. At those testing points, the participants’ 3D gaze was estimated and compared to the actual position of the points. For
comparison purposes, the visual axes intersection method was also applied to the recorded eye feature data to estimate the 3D gaze.

![Diagram of 3D gaze calibration procedure](image)

Figure 4.5: Demonstration of the 3D gaze calibration procedure. The red dots are those 4×4 defined visual targets that the participants need to concentrate on. This plane was placed on four different depths ranging from 60 cm to 100 cm.

### 4.5.2 Object Localization and Pose Estimation

The performance of using 3D gaze to estimate the object location and pose was second to be examined. After the 3D gaze calibration, the participants were asked to view the cuboid object that was 23.5 cm long, 18 cm wide, and 3.5 cm thick. The object was placed at four random locations within the calibrated space. The 3D gaze points were estimated when the participants were viewing different points on the object. These 3D gaze points were used to estimate the object location and the pose with the methods presented in 3D gaze interpretation Phase I. The estimations were compared with the actual values to measure the estimation performance.

### 4.5.3 Building the Visuomotor Grasping Model

In this experiment, the participants were asked to grasp the cuboid object using their right hand at nine different areas on the object to simulate the natural eye-hand coordination of looking at and grasping the location $p_g$. Each grasping area was repeated six times by each participant. The object was placed on the right side of each participant at 12 different
positions with different orientations, as illustrated in Fig. 4.6. Each position is annotated as \( AiDj \), where \( i \in [1, 4] \), and \( j \in [1, 3] \). For each participant, all these locations were reachable. The hand contact points and hand approaching angles were recorded to build the grasping model. The visuomotor grasp planning using the grasping model was evaluated on a virtual robot. At four designated grasping points, the generated grasping configurations were generated, and their human-like level and satisfaction level were subjectively evaluated by the participants.

![Figure 4.6: Object layouts and experiment setup for building the grasping model.](image)

4.5.4 Overall Evaluation of the Interaction Framework Using 3D Gaze

The presented HRI framework using 3D gaze was validated with a setup shown in Fig. 4.7. The participants who wore the 3D gaze tracking system controlled the Mico robotic arm to grasp the cuboid object by directly looking at it. The participants needed to look at five points on the object in each grasping trial (four points are for *Phase I* to determine the object location and pose and one point is for *Phase II* to indicate the grasping plan). Then the Mico arm was activated to grasp the object using the generated plan. At least four trials were performed by each of the participants. The evaluation was quantified from two aspects: 1) the success rate of the grasping task and 2) subjective evaluation using questionnaires. Subjective evaluation was used to evaluate the overall interaction framework using 3D gaze.
from the aspects of ease of use (A1) and ease of learning (A2), and the robotic grasping performance. The evaluation of A1 and A2 was performed using a questionnaire adopted from the USE questionnaire [102], which is a widely used questionnaire designed for usability evaluation\textsuperscript{13}. The evaluation of the grasping performance consisted of two questions: 1) I am satisfied with the grasping performance (A3) and 2) The robotic grasping configuration is human-like (A4). These items in the questionnaire (A1 to A4) were scored by the participants from 0 to 4 with 4 as the most positive assessment and 0 as the most negative.

![Figure 4.7: Experiment setup of using 3D gaze to command an assistive robot for grasping.](image)

An additional experiment of using 3D gaze to perform the robotic grasping task was conducted. The experiment setup and procedure were the same as the previous one. The only difference was that the pose of the cuboid object was not determined from 3D gaze but instead was pre-known, which simulated the case that additional technologies (e.g., RGB-D camera) were used to measure the object pose after having the target object localized in the 3D environment. The grasping results were evaluated by the success rate and the same questionnaires (A3) and (A4).

\textsuperscript{13}Questions 9, 11, 12, 14, 15, 16, and 19 were selected from USE questionnaire for A1 evaluation, and questions 20, 21, 22, and 23 were selected for A2 evaluation.
4.6 Results

4.6.1 Accuracy of the 3D Gaze Estimation

The Cartesian errors of the estimated 3D gaze on each axis and the overall Cartesian error are summarized in Table I. The presented system using the gaze vector method achieved an average error of 2.4±1.2 cm within a depth range of 100 cm. This accuracy is better than any of those achieved in the previous work (an error of 3.93 cm in [32]; an error of 5.8 cm in [33]; 4.59 cm error along the depth direction only in [38]), which makes it possible to build practical robotic applications with the 3D gaze. As a further comparison, the visual axes intersection method was also tested using the recorded data and the presented hardware system, and the average error was 8.9±7.9 cm, as show in Table I, which is much greater than the gaze vector method. The experimental results prove that the gaze vector method is a practical method for accurate 3D gaze estimation in real environments.

Table 4.1: CARTESIAN ERRORS OF 3D GAZE ESTIMATION USING THE PRESENTED SYSTEM. MAX: MAXIMAL, MIN: MINIMAL, AND AVE: AVERAGE.

<table>
<thead>
<tr>
<th>Unit: cm</th>
<th>Gaze vector method</th>
<th>Visual axes intersection method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Max</td>
<td>Min</td>
</tr>
<tr>
<td>$e$ on X</td>
<td>2.4</td>
<td>0.2</td>
</tr>
<tr>
<td>$e$ on Y</td>
<td>1.0</td>
<td>0.1</td>
</tr>
<tr>
<td>$e$ on Z</td>
<td>6.6</td>
<td>0.6</td>
</tr>
<tr>
<td>Overall</td>
<td>6.7</td>
<td>0.8</td>
</tr>
</tbody>
</table>

The average errors of the left and right visual axes from the mapping relationships $M_L$ and $M_R$ were 1.08±0.55° and 0.63±0.32°, respectively, and the average error of the gaze vector calculated from these two visual axes was 0.56±0.31°. This demonstrates that the gaze vector method can effectively alleviate the error propagation problem and achieve an accurate gaze vector. Moreover, the average error of the gaze distance estimated by $M_D$ was 2.1±1.3 cm, which proves our belief that the gaze distance can be accurately estimated along the gaze vector.
4.6.2 Object Localization and Pose Estimation

In Fig. 4.8, a successful estimation of the object location and pose is demonstrated by plotting them in a point cloud captured by Kinect, in which the estimated items have been transferred into the Kinect’s coordinate system. The big green dot represents the object center and the blue arrow that passes through the center represents the object pose. The 3D gaze estimated when the participant was looking at the four corners are represented by four small red dots. The average error of the object location and pose were $2.6 \pm 2.0$ cm and $29.4 \pm 6.1^\circ$, respectively. As the subject was looking at the surface of the object, the dots that represent 3D gaze and the object’s center are on the object’s surface.

Figure 4.8: Demonstration of successful estimation of object center and pose using 3D gaze. The big green dot is the object center and the blue arrow that passes through this dot is the object pose. Four small red dots represent the four fixations generated by the subject.

4.6.3 Visuomotor Model for Robotic Grasping

The recorded grasping data of the contact point and hand approach angle from one of the participants are plotted in Fig. 4.9. A clear grasping pattern can be observed from the plot, and this pattern can be observed in the recorded data of all other participants. Within the reachable region, this grasping pattern is not affected much by the location and pose of the object. This suggests that there is a generic visuomotor grasping model for humans, and this model is mainly related to the object. Small variations of this model were observed
among different participants, which could be attributed to the individual differences between participants’ heights, arm lengths, and hand sizes. The learned GMM grasping model with $K = 3$ is shown in Fig. 4.10, and the generated GRM is also shown to better visualize the relationship between the hand contact point and the hand approach angle.

![Figure 4.9: Raw grasping data of the contact point and the hand approaching angle from one of the participants. $A_iD_j$ corresponds to the 12 testing locations, where $i \in [1, 4]$, and $j \in [1, 3]$.](image)

The generated grasping configurations of the virtual robot using the visuomotor grasping model are shown in Fig. 4.11. Participants all agreed the grasping configurations were satisfactory and human-like (the average score of A3 is 3.6 out of 4 and the score of A4 is 3.6 out of 4). This shows that the learned visuomotor grasping model is representative and that a human-like and functional grasp plan can be generated using this model.

### 4.6.4 Evaluation of the 3D-Gaze-Based Human-Robot Interaction

A success rate of 55.6% was achieved in the first set of experiments (the object pose were determined by the 3D gaze), and a success rate of 74.2% was achieved in the second set (the object pose was pre-known). In the first set of experiments, a major failure was the robot’s hand smashed into the object or knocked it away. Both the location error and pose error
Figure 4.10: The visuomotor model for grasping. The top plot is the GMM from the raw grasping data; and the bottom is the generated GRM.

Figure 4.11: Grasping planning using the visuomotor grasping model with a virtual robot at four grasping points.
contributed to this failure. In the second set of experiments, the success rate was better as it was only affected by the location error.

The successful grasping instances in both sets of experiments were subjectively evaluated by the participants. The results show that the grasping outcomes were satisfactory and human-like (the average score of A3 was 2.5 out of 4 for the first set and 2.9 for the second set, and the average score of A4 was 3.2 for the first set and 3.2 for the second set). These scores were less than those in the planning simulation with the virtual robot, and scores of the second set of experiments were better than those of the first one. In real experiments, the estimation errors in object localization, pose estimation, and the grasping point assignment affected the final robotic grasp configuration and thereby reduced the similarity of the robotic grasping configuration to a human grasping configuration. In addition, higher estimation errors had greater effects on the final robotic grasping configuration and further lowered the evaluation scores.

From the overall framework evaluation, we can draw the conclusion that the 3D-gaze-based HRI is very easy to learn and use (the average scale of A1 was 3.5 out of 4, and the scale of A2 was 2.9). From the interviews of participants, they were excited to see their 3D gaze could be tracked in a real environment and be used to command a robotic arm for grasping. They thought it was easy to learn and use the 3D-gaze-based interaction system to command the robotic arm. However, they also mentioned that explicitly looking at four corners of the object was not convenient and slowed the whole procedure.

4.7 Discussions

4.7.1 Theoretical Analysis of the Gaze Vector Method

The gaze vector method was theoretically analyzed with simulated visual axes and gaze distance. At a specified visual target, the desired left and right visual axes and gaze distance were computed following the definition. With a specified angular error, all possible estimated visual axes that had this angular error could be simulated, which formed a cone around the desired left and right visual axes, respectively, with the apex of each cone located at the
corresponding eye. The opening angle of such a cone is twice as large as the angular error. Similarly, the estimated gaze distance could be simulated, but it would be a single item as it was a scalar. Inputting one pair of the estimated left and right visual axes into the gaze vector method with the estimated gaze distance, a 3D gaze point could be estimated and the deviation of this estimation from the visual target could be computed. Performing the gaze vector method using all combinations of the left and right visual axes, a distribution of the estimated 3D gaze locations for that visual target could be obtained. In addition, the mean of the deviations of all estimated 3D gaze points from the visual target could be treated as the error expectation of this visual target under this specified error condition. Following the same procedure, the error distribution for any visual target in the workspace could be obtained. For comparison purposes, the visual axes intersection method was also analyzed.

Figure 4.12 demonstrates the 3D gaze distribution that was respectively estimated using the gaze vector method (green squares) and the visual axes intersection method (purple squares). The visual target was at [0, 0, 70] cm, shown as a blue dot. The simulated left and right visual axes both had 0.5° angular error, and the simulated gaze distance had +2 cm error. Twenty left and right visual axes were respectively sampled, which resulted in 400 estimated 3D gaze points. It clearly demonstrated that the 3D gaze points estimated using the gaze vector method had smaller deviations than those using the visual axes intersection method. In addition, the error expectation using the gaze vector method was 2.05 cm, while it was 5.05 cm for the visual axes intersection method.

After applying the theoretical analysis on the entire workspace, more evidence was found to support the gaze vector method. There was a chance of 95% that the error of the computed gaze vector was less than the average error of two visual axes, and on average the error of the gaze vector was only 65% of the visual axes’ average error. This proves that an accurate gaze vector could be computed in the gaze vector method.
Figure 4.12: The distribution of the estimated 3D gaze using the gaze vector method (green squares) and using the visual axes intersection method (purple squares), respectively, at visual target \([0, 0, 70]\) cm (blue dot). Note that the Z-axis has a different scale from the X-axis and Y-axis.
4.7.2 Analysis of 3D Gaze for Object Pose Estimation

Though we have improved the accuracy of 3D gaze tracking, we still encountered the low accuracy issue in object pose estimation using the 3D gaze during the experiment. A possible reason was that these estimated 3D gaze points had different errors with different magnitudes and directions pointing from the visual target to the estimated 3D gaze location. This error difference in direction and magnitude could largely affect the pose determination. In this section, we further theoretically analyze how the object pose estimation is affected by these estimated 3D gaze points.

In this theoretical analysis, we assigned four points as visual targets corresponding to the four points viewed by the participants in the experiment. For each visual target, a set of estimated 3D gaze points (50 in number) were simulated, which had a Cartesian error $\epsilon$ and were uniformly distributed around the visual target on a sphere’s surface whose radius was $\epsilon$. Four simulated 3D gaze points were picked, one from each set. Then, the object pose was computed using these four gaze points, until all combinations of four points had been picked once (number of $50^4$). The errors of the estimated object pose were statistically analyzed by fitting them with a normal distribution, as shown in Fig. 4.13, (the left Y-axis). Cases with different $\epsilon$ are distinguished by different colors. For example, when the 3D gaze has an error of 1.0 cm (shown as the orange line), the estimated object pose has an error distributed in a range from $0^\circ$ to $9.5^\circ$ (along the X-axis), and the expected error is $3.8^\circ$, which is the peak of the distribution curve.

This analysis shows that the object pose estimation is very sensitive to the error of 3D gaze. When 3D gaze has an expected error of 0.5 cm, the possible maximal pose error could be $4.8^\circ$, and it increases to $32.2^\circ$ when the 3D gaze has an expected error of 3.5 cm. This suggests the object’s pose can only be roughly estimated from the 3D gaze, and it needs to be further improved with alternative technologies for robust robot control.

Another relationship can be obtained when plotting the 3D gaze error against the expectation error of the object pose, shown as the black dashed line in Fig. 4.13, (the right
Figure 4.13: Theoretical error distributions of object pose estimation when the 3D gaze has different levels of error, which are indicated by lines with different colors. The black dashed line is the relationship between the 3D gaze error (right Y-axis) and the expected error of the object pose.

This line can be used to predict the expected error level of the object pose when the error of 3D gaze is known. More importantly, it suggests the accuracy requirement of the 3D gaze estimation for a practical application. As an example, for a practical application, if the error tolerance of the object pose is $5^\circ$, it suggests the error of the 3D gaze tracking should be less than 1.5 cm. Similar analysis could also be performed for the object location estimation to get the prediction relationship between the error of 3D gaze with the error of the object location.

4.7.3 Potential of 3D Gaze in HRI

In this chapter, we demonstrate how a user can use gaze to express what he/she wants to effectively and intuitively command a robot for object manipulation. In the experiments, the participants managed the 3D-gaze-based HRI without any particular learning efforts. They successfully conducted the robot to accomplish the grasping task with their gaze as the control input. This chapter demonstrates possible ways to interpret and convert the 3D gaze into an effective control modality of robots. We are looking forward to applying
this 3D-gaze-based HRI in the homecare scenario, which allows disabled and older adults to effectively and intuitively interact with assistive robots or other assistive systems in their daily living with a setup shown in Fig. 4.14.

Instead of explicitly driving a mobile by specifying its moving direction or controlling a robotic arm by specifying a joint’s rotation degree as previous 2D-gaze-based HRI and most EEG-based HRI [103], using 3D gaze as the interaction modality can directly indicate the manipulation target by simply looking at the object. After defining the manipulation target, the intelligent robot can autonomously perform the task or some procedures of the task. For example, for transferring an object, the user can simply look at the object to indicate it as the manipulation target and look at a location on a table to indicate the spot as the transfer destination; then, the robot can perform the task autonomously. Commanding an assistive robot in a manner like this is much more desirable than explicitly steering the robotic arm. Similarly, the 3D gaze can also be used to indicate the destination of a drone or a mobile. Moreover, the gaze modality can be combined with other modalities, such as gesture, speech, EEG, or EMG, to achieve multimodal interaction for more intuitive and accurate communication. Combinations of multiple modalities for HRI can solve some inherent problems of an individual modality. For example, combining with EEG or EMG modality can be helpful to solve the Midas touch problem of the gaze modality. A command derived from the EEG or EMG signal can be used as a confirmation action to a gaze command by distinguishing the intentional gaze command from the natural eye-gaze movement data without manipulation intention. On the other hand, the gaze modality can directly localize the target of interest, which is helpful to regulate the motion of an assistive robot arm controlled using the EMG signal. However, effectively combining various modalities is still an open problem and worth of investigation. In this chapter, we focused on object grasping. The more complex human intention like making breakfast or washing a cup can be inferred from gaze as presented in our earlier work [104], which, however, require advanced task knowledge for the robot to accomplish the task autonomously.
Figure 4.14: 3D gaze in a real environment for assistive robot control. Our goal is to achieve a novel paradigm that a user can intuitively command an assistive robot for certain services by naturally gazing at the target object in the real world.

On the other hand, though the tracking accuracy of 3D gaze was improved in our tracking system, the errors still caused some failures of the grasping tasks. In addition, the questionnaire results show that the interpretation of the 3D gaze could be more implicit. Alternative ways to interpret the 3D gaze, which can implicitly and accurately localize the object and determine the pose, need further investigation. Further improvement of the 3D gaze tracking accuracy, interpretation of the 3D gaze, and combinations with other existing technologies to compensate for the 3D gaze accuracy and build more robust interaction modality will be the focus of our future work. For example, by making use of the techniques of computer vision, the accuracy of the target object’s location and pose can be further improved, and the identity of the object can be recognized.

4.8 Conclusion

In this chapter, we investigated and validated using human 3D gaze in real environments to command a robotic arm for complex task execution. A gaze vector method was developed to accurately estimate a person’s 3D gaze, which enabled gaze-based HRI in real environments. Moreover, we designed a novel interaction framework to interpret the 3D gaze, which
mimics humans’ visuomotor functions and takes advantage of humans’ eye-hand coordination. In this framework, the location and pose of the visualized object were determined from the 3D gaze data, and a grasping plan was generated using the learned visuomotor model. In the experiments, the participants managed the 3D-gaze-based interaction to command a robotic arm for object grasping. The experimental results demonstrated the effectiveness and intuitiveness of gaze-based HRI. The presented framework can increase the control capability of the gaze signal in relatively complex robotic tasks. This 3D gaze modality is expected to benefit those users during HRI, who suffer from impaired mobility in their daily living or need an additional hand in a working scenario.

4.9 Acknowledgement

This material is based on work supported by the US NSF under grant 1414299. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the National Science Foundation.
CHAPTER 5
ATTENTION-AWARE ROBOTIC LAPAROSCOPE BASED ON FUZZY
INTERPRETATION OF EYE-GAZE PATTERNS

A paper published at *Journal of Medical Devices*\(^\text{14}\)

Songpo Li\(^\text{15}\), Xiaoli Zhang\(^\text{16}\), Fernando J. Kim\(^\text{17}\), Rodrigo Donalisio da Silva\(^\text{17}\), Diedra
Gustafson\(^\text{17}\), and Wilson R. Molina\(^\text{17}\)

5.1 Abstract

Laparoscopic robots have been widely adopted in modern medical practice. However, explicitly interacting with these robots may increase the physical and cognitive load on the surgeon. An attention-aware robotic laparoscope system has been developed to free the surgeon from the technical limitations of visualization through the laparoscope. This system can implicitly recognize the surgeon’s visual attention by interpreting the surgeon’s natural eye movements using fuzzy logic and then automatically steer the laparoscope to focus on that viewing target. Experimental results show that this system can make the surgeon-robot interaction more effective, intuitive, and has the potential to make the execution of the surgery smoother and faster.

5.2 Introduction

5.2.1 Laparoscopic Surgery

Laparoscopic surgery has been widely adopted in modern medical practice. For example, over 96% of the approximate \(1 \times 10^6\) cholecystectomies are performed each year in the U.S.

---

\(^{14}\)Reprinted with permission of *Journal of Medical Devices*, 2015, 9(4):041007.

\(^{15}\)Primary researcher and author, graduate student, Department of Mechanical Engineering, Colorado School of Mines.

\(^{16}\)Author for correspondence, Assistant Professor, Department of Mechanical Engineering, Colorado School of Mines.

\(^{17}\)Department of Urology, Denver Health Medical Center.
using laparoscopic surgical techniques [105]. The great distinction is drawn at the employment of slender surgical instruments and laparoscopes compared with traditional surgeries. Instruments and laparoscopes are inserted into the patient’s body through small incisions (typically 5-10 mm wide). During the surgery, they are manipulated and positioned separately by the primary surgeon and/or a human assistant to carry out appropriate operations on and/or visualization of the surgical site. Performing the surgery laparoscopically can significantly reduce the trauma to patients’ tissue and consequently reduce blood-loss, post-operative pain, hospitalization, and recovery time. As a result, laparoscopic surgery has achieved great popularity among surgeons and patients.

Visualization and focus of the surgical site can be challenging, however, which may increase the operating time and frustration, especially when re-adjusting of the laparoscope is required during a critical part of surgery. The laparoscope must be frequently re-adjusted and is usually done by the surgeon’s assistant via oral communication to choose the correct operating site [106, 107]. In light of the fact that the surgical field is being projected remotely from the patient’s body on a monitor and the assistant is watching the monitor from a different angle relative to the surgeon, it can be difficult for the assistant to fully understand which area of the surgical field the surgeon would like to view through oral communication. Some other common issues with using a human assistant to position the laparoscope include hand tremor, fatigue, and a fulcrum effect at the trocar insertion point [107].

5.2.2 Robotic Laparoscope Systems

To overcome the problems caused by manually positioning the laparoscope, robotic laparoscope systems have been developed to substitute human assistants. Such robotic systems have fine manipulation capabilities including scalable, steady, tremor-free motion, and enhanced dexterity. By 2009, at least 27 kinds of robotic laparoscopes have been commercialized or published in referred articles [108] and this number has been growing rapidly in recent years. These robotic systems include AESOP made in the U.S. by Computer Motion, Inc. [109], EndoAssist made in the UK by Armstrong Healthcare Ltd. [110] which is now
known as Freehand system from Prosurgics, Inc. [111], LapMan made in Belgium by Medsys s.a. [112], and Naviot made in Japan by Hitachi Co., Ltd. [113].

Although these systems remove the need for human assistants, interacting with these robots may increase physical and cognitive load on the surgeon. Explicit control of these robots through a control interface, such as joystick, foot pedal, voice command controller, or head/face motion-activated system, could be an additional task that distracts the surgeon’s attention from the surgical site and may result in frustration and prolonged surgical time. For example, in joystick/button control, the surgeon needs to remove his/her hand from the manipulation of the surgical instruments to re-adjust the laparoscope [114]. Voice-recognition software may accept wrong commands and may limit what the surgeon can say to others in the operating room (OR) [115]. Using head movement to control the robotic laparoscope requires the surgeon to move his/her head an additional amount while performing a surgery, which may complicate the surgery [116] and tire the surgeon, especially during long procedures. Therefore, reducing the surgeon’s control burden by increasing the level of the autonomy in robotic laparoscopic systems is a necessity to simplify their use for surgeons and ensure their smooth and fast operation.

5.2.3 Automatic Laparoscopy

5.2.3.1 Automatic Instrument Following

To free the surgeon from explicitly steering the laparoscope, the development of several automatic laparoscope adjustment systems has been attempted [117–120]. In those systems, the laparoscope is automatically adjusted to focus on the primary surgical instrument. The hypothesis of this strategy is that the position of the instrument’s tip represents the surgeon’s region of interest in the laparoscopic images. Several techniques have been adopted to track the position of the instrument’s tip, including pattern recognition [121], color identification [117, 118], and optoelectronic barcode identification [122]. However, none of these has really gained broad clinical acceptance because the position of the instrument’s tip cannot always stand for the surgeon’s visual need [123]; there are cases that the surgeon needs to observe
a specific location without manipulating it.

### 5.2.3.2 Gaze-Controlled Laparoscopes

Gaze tracking is a technique that continuously estimates where the user is looking by monitoring his/her eye movements. Nowadays, optical eye tracking devices have gained wide acceptance and usage [17, 18, 124, 125] due to their noninvasiveness and inexpensiveness. Eye-gaze tracking has been widely used as a tool to study cognitive science, psychology, neurology, and visual behaviors. Recently, using gaze as a novel modality to explicitly control or implicitly interact with a computer [47, 126] or robot [87, 127] has drawn much attention due to its unique characteristics including affordability, ease of usage, unobtrusiveness, etc. [1, 128]. In particular, gaze has been used as a unique communication means to assist people who have motor impairment or need an extra hand in human-machine interaction. For example, disabled people can use gaze for typing through an onscreen keyboard [129], triggering buttons on a monitor [130], and controlling a wheelchair [131] and even a robotic arm [13].

In 2010, Yang and coworkers attempted to use the surgeon’s gaze to steer a robotic laparoscope system by gazing at a button or a particular screen portion which represented a motion command for a robot joint or a certain step movement toward one direction [91]. Staub et al. [89] established another gaze-controlled method in ARAMIS surgical robotic system. In their system, gaze was used to define the laparoscope’s moving direction, and a foot pedal was needed to activate the robot as well as to stop the robot’s motion along that direction.

In our previous work, we proposed a direct gaze-guided robotic laparoscope system [92, 93] in which the surgeon could directly indicate the visual target using his/her gaze. The rational of our research is what you are looking at indicates your viewing attention. The surgeon just needs to naturally look at the site that he/she is interested in and the robot automatically follows the surgeon’s viewing attention. Compared with the indirect, incremental control strategies in previous robotic laparoscope systems using eye gaze, the direct control in our
system effectively reduced the surgeon’s direct interaction with the robotic laparoscope. The learning curve for the surgeon to become familiar with the robot operation was significantly shortened, as there was no need for the surgeon to know the control techniques behind the laparoscopic image. Therefore, by using this gaze-guided laparoscope system, the surgeon can focus more on the exploration and manipulation at the surgical site, instead of dealing with operating the robot.

5.2.4 Challenges in Gaze-based Interaction

The common issue with gaze-based interaction, not only existing in gaze-based laparoscope control but also in general computer/robot interaction, is the lack of an effective interpretation method to recognize the user’s point of attention from his/her natural eye-gaze data. As eye movements can never be off [8], wherever a person looks a command may get activated without the user’s intention to do so, which is widely known as the Midas touch problem [132, 133]. The most common solution to overcome this problem is to add an extra confirmation produced by the user. Some popular confirmations include a prolonged fixation (dwell-time method), intentional blink, and additional physical confirmation (such as a button or a foot pedal). However, those factitious and imposed rules may bring extra mental and physical burdens upon the user.

To identify the surgeon’s visual attention to steer the laparoscope from the gaze data is even more challenging because of the uncertainty caused by the complex surgical environment. Unexpected situations like bleeding or failure in one specific operation may occur during a procedure. In addition, the surgeon may look at objects or areas unrelated to the needs of surgical operation. These challenges make the common confirmation strategies inappropriate for the OR. Because of those effects, an appropriate value of the dwell-time can hardly be defined for an entire surgery procedure. When the dwell-time is not long enough to distinguish the normal eye movements, the laparoscope’s motion errors will increase due to the misinterpretation which will require more time to correct. At the same time, the errors will increase mental burden on the surgeon as he/she may be always worrying about
accidentally triggering the robotic motion. If the dwell-time is too long, the surgeon has to unnaturally maintain his/her gaze at the target for an additional amount of time, which may limit the surgeon’s natural viewing behaviors and easily result in fatigue. Correct recognition of surgeon’s visual attention becomes more critical in the OR due to safety consideration, as any unauthorized movement of the laparoscope caused by the incorrect interpretation of the surgeon’s gaze can lead to serious consequences for the patient. Therefore, a more natural and intuitive but also effective, methodology is urgently needed to recognize the correct visual attention from the surgeon’s natural eye-gaze data. This would allow the surgeon to use his/her gaze to communicate and interact with the computer or machine without extra tedious eye gestures.

To address the problems mentioned above, we presented a novel attention recognition method by analyzing the surgeon’s natural eye-gaze movement patterns. Compared with the traditional dwell-time method, we investigated the gaze-based robot control by interpreting the eye-gaze movements at the pattern level, which considered the surgeon’s current and historic visual behaviors, to better understand the surgeon’s viewing attention. A fuzzy logic inference engine was developed to infer the surgeon’s visual attention using the extracted eye-gaze movement patterns. Using the presented attention recognition method, the robotic laparoscope system has the potential to shift from the automatic level to the autonomous level, as shown in Fig. 5.1.

5.3 Methods: Visual Attention Inference Using Fuzzy Logic Interpretation of Visual Behavior for Robotic Laparoscope

5.3.1 Gaze-Guided Laparoscopic View Interaction

Our gaze-guided robotic laparoscope system, shown as Fig. 5.2, allows the surgeon to directly control the laparoscopic view using eye gaze [92, 93]. When a new visual attention of the surgeon is recognized, the system can automatically steer the robotic laparoscope to focus on the region of interest. The advantage of this system is to introduce a direct control approach in which the visual attention on the image in two-dimensional pixel coordinates is
Figure 5.1: The flowchart of the attention-aware gaze-guided robotic laparoscope system.

directly mapped to the robot’s three-dimensional motion in Cartesian space into gaze-based robotic laparoscope systems. The kinematics transformation can be expressed as Eq. 5.1, where $p^{atten}$ is the position of the surgeon’s visual attention on the laparoscopic image in pixel; $[m]$ is a mapping relationship from the laparoscopic image in pixel to the Cartesian position in terms of the laparoscope frame; $[T]$ is a coordinate transformation matrix from the laparoscope frame to the robot base frame; and $P^c$ is the target position of the laparoscope so that it can focus on the surgeon’s new visual attention.

$$P^c = [T][m] \cdot p^{atten} \quad (5.1)$$

The gaze-guided laparoscope system comprises a robotic laparoscope system and one eye tracking system. Compact bevel-gared robot for advanced surgery (CoBRASurge), which was built in our previous research [134], is used as the robotic laparoscope holder. Its capability as a robotic laparoscope holder has been validated in animal tests [135, 136]. CoBRASurge is based on a spherical bevel-gared mechanism consisting of three gear pairs and six turning pairs. It creates a mechanically constrained remote center of motion (RCM), which contains three rotational degrees of freedom (DOFs) and one translational DOF passing through it. During surgery, the RCM is aligned with the surgical entry port which
Figure 5.2: The setup of the gaze-guided automated robotic laparoscope system. The image of the artificial muscle in the surgery simulator is projected on a monitor through the robotic laparoscope.

This system is hosted on a laptop running LABVIEW software (National Instruments, Austin, TX). In our previous work, the gaze control adopted a traditional dwell-time method. A sliding window filter is first used to remove involuntary eye movements as well as saccades from raw gaze data. From the refined gaze data, the surgeon’s visual attention is recognized by using the dwell-time method which is derived from the ground truth that a human’s eye-gaze focuses on an object when he/she is interested in it. A dwell-time threshold is defined, and when the surgeon’s gaze focuses on a small area longer than this threshold, visual attention is recognized. The visual attention, represented as a point or an area on the screen, is converted into a series of motion commands to drive the robot toward the
visual attention. Although this dwell-time method was adopted in this previous study, it could undermine the usability of the whole system in practical applications as it forces the surgeon to stare at the visual targets for a long time to distinguish the attention from normal visual behaviors. It could affect the efficiency of the system and lead to extra fatigue for the surgeons.

5.3.2 Eye-Gaze Behavior Studies

5.3.2.1 Superimposed Eye Movements

Human eyes make many different movements, including the convergence for visual focusing, pursuit motion for visual tracking, saccades for attention shifting, etc. It is more important to note that some movements are involuntary, for example, rolling, nystagmus, drift, and microsaccades along with physiological nystagmus. Also, visual distractions, blank stares, and blinks may occur. Due to these superimposed eye movements, the gaze data estimated from human eye movements will consist of noise, and must therefore be filtered. After being filtered, the eye-gaze data representing the surgeon’s attention processes can be recognized.

5.3.2.2 Vision Center and Visual Acuity

In general, a human tends to put visual targets at his/her vision center. This is related to a term called visual acuity, which describes the extent to which a human can clearly see a target within the visual field. Visual acuity exponentially decreases as the object shift farther from the vision center [137]. This means that for clear perception, it is better to directly look at it than looking askance at it. Additionally, looking askance at an object for a long time will lead to incredible fatigue and stress on human eyes. Another very important aspect is that when a target is viewed directly, its surroundings are also visible within human split vision, in other words, even if the human is concentrating on the visual target, he/she can also stay aware of the surroundings, which can eliminate the emergencies and incidents caused by blind vision. In the humanmonitor scenario, the visual target or manipulation
target is usually located at the center of the monitor for clear, comfortable visualization, and awareness of the surroundings. In cases like this, we say the display is focusing on the manipulation or visualization target.

Based on these components of vision characteristics, an elliptical central is defined at the center of the monitor displaying live video (shown in Fig. 5.3) and is called the focusing area. It is assumed that during stable manipulation and/or visualization, the manipulation target and visual target should be within this elliptical central area and the surgeon’s gaze should be within it as well. In this condition, the robotic laparoscope will stand still to provide stable view to the surgeon. When the surgeon’s gaze falls out of this focusing area, it may indicate that the surgeon has lost interest (manipulation or visualization interest) to the current focus area, and the laparoscope should be adjusted to focus on surgeon’s new attention area. The size of this elliptical central area is adjustable based on the surgeon’s preference. In default, its major axis is half long as the screen’s width and its minor axis is three fifths of the screen’s height, which covers about 23% of the entire screen.

![Figure 5.3: Two kinds of historical eye-gaze behaviors in a surgical operation.](image)

### 5.3.2.3 Eye-Gaze Movement Patterns

As mentioned previously, the acquired gaze data are superimposed with noise and it is very difficult to recognize the surgeon’s visual attention from eye movement data. Currently, most of the attention recognition methods are still at the fixation level, such as the dwell-time method which only considers how long a human’s gaze has been maintained on an object or area. In this method, the historical eye movements before gaze starts settling are discarded.
Further exploration of the historical eye movement data at the pattern level should provide more clues about how human attention is shifted from one location and generated at another.

By observing the surgeon’s eye-gaze behaviors, two common eye-gaze patterns are identified in our study as shown in Fig. 5.3. The dots are clusters of gaze points and the arrows show the path of these gaze clusters. In pattern A, the gaze cluster starts from the inside of the focus area and returns back after moving out once. In this instance, the gaze cluster 2 may be caused by a visual distraction or a quick glance. It suggests that when the gaze cluster first moves out of the focus area after staying in it for a long time, there is a possibility that the new gaze cluster is caused by some visual distractions and the surgeon will look back in the focus area shortly. Therefore, the robot needs to be aware of this situation and be conservative in its activation to reduce the chance of misinterpretation. In contrast, pattern B shows that the gaze cluster is shifting outside of the focus area, which means there is a high possibility that the surgeon has lost his/her interest in the current vision center and is searching for the next visual target. Given this situation, the robot should be more sensitive to the duration of gaze maintaining on an object or a small area and responsively adjust the laparoscope to focus on surgeon’s new visual attention.

### 5.3.3 Noise Filter

An adaptive sliding window filter is developed for filtering the raw gaze data, illustrated in Eq. 5.2 and 5.3. $N$ is the size of the sliding window. $P_i$ and $\tilde{P}_i$ are the position of the $i$th raw gaze point and the gaze point after being processed by the filter, respectively. $E_i$ is an influence coefficient derived by Eq. 5.3, which indicates whether this new received gaze point affects the output or not. This filter first assigns the influence coefficient, either 0 or 1, to each received gaze point, based on the relative distance from the gaze point to the centers of the points in the current sliding window. The output of the filter is the mean position of all the gaze points, whose influence coefficients are 1, in the sliding window. This filter is developed to remove the noisy eye movements caused by blinks, attention shifts, and tracking failure. Additionally, it can remove involuntary eye movements such as rolling,
nystagmus, drift, and microsaccades to smooth the gaze points.

\[
\tilde{P}_i = \frac{1}{\sum_{k=1}^{N} \{ \sum_{j=1}^{N-1} \tilde{P}_{i-j} * E_{i-j} + P_i * E_i \}}
\]  

(5.2)

\[
E_i = \begin{cases} 
0, & \text{if } \| P_i - \frac{1}{\sum_{k=1}^{N} E_{i-k}} \sum_{j=1}^{N} \tilde{P}_{i-j} * E_{i-j} \| \leq \text{threshold} \\
1, & \text{if } \| P_i - \frac{1}{\sum_{k=1}^{N} E_{i-k}} \sum_{j=1}^{N} \tilde{P}_{i-j} * E_{i-j} \| > \text{threshold}
\end{cases}
\]  

(5.3)

5.3.4 Fuzzy Logic Interpretation of Eye-Gaze Data

In 1965, Zadeh published the fuzzy set theory to better model the uncertainty in the real world [138]. It has the potential to be a suitable tool for surgical applications to better handle uncertainties during a surgical operation caused by the complexity of the surgery and the surgeon’s eye movement behavior. Some successful instances of using the fuzzy logic theory in machine control include the Sendai railway (1987) in Japan, the Mitsubishi air conditioner [139], research trials such as intelligent behaviors of humanoid robots [140] and mobile robot navigation [141, 142].

A fuzzy logic inference engine is developed to infer the surgeon’s visual attention by interpreting his/her eye-gaze behavior data. It measures the likelihood of the current gaze cluster being a new visual attention rather than other visual behaviors like searching, glancing, etc. The inference output will be control strategy, ”standstill” or ”activation” which the laparoscope robot should adopt in order to keep the current focusing area or move to surgeon’s new visual attention.

The elliptical focus area is defined at the screen center as the boundary to differentiate between the current view is good and the current view has expired. As mentioned earlier, normally the surgical target being manipulated as well as the surgeon’s visual attention would be inside the focus area where it indicates that the current view is good, therefore, the laparoscope should stand still and focus on this area. When the surgeon’s visual attention falls out of the focus area, it means the current view has expired.
The fuzzy inference engine, illustrated in Fig. 5.4, consists of three components: membership functions, IF-THEN inference rules, and inference conclusion. There are two membership functions which model how the current gaze points focus on one location and how the historical gaze behaved before focusing on that location. The membership functions convert the measurement of gaze behaviors into three fuzzy descriptions, with degrees of truth, indicating how likely the situation is fitted. The IF-THEN rules tell, at different situations, what control strategy (standstill or activation) should be adopted. All the likelihoods of these control strategies can then be fused and the one with the higher likelihood will be the inference output.

![Diagram of fuzzy inference engine](image)

**Figure 5.4:** The overall procedure of the fuzzy inference engine.

### 5.3.4.1 Membership Functions

The surgeon’s historical eye-gaze behaviors are modeled by two fuzzy logic membership functions, new interest (NI) and keep focus (KF), respectively. NI function measures how
likely a small area outside of the focus area, concentrated on by the current gaze points, is
the surgeon’s new visual interest based on the duration the gaze points have been maintained
in that area. It is described as possibility level low, medium, and high with degrees of truth,
which are calculated, respectively, using the duration of gaze points. KF function is the
measurement of the surgeon’s interest in the current focus view. The input for KF is the
period that gaze stays within the focus area before falling outside of it, and the outputs are
the degrees of truth for each possibility level (low, medium, and high). The duration inputs
are first normalized by T, which is an empirically determined duration of a human’s attention
on a visual target. In both membership functions, 0.33, 0.66, and 0.99 are selected as the
cutoff values for the possibility low, medium, and high levels. Two membership functions
are illustrated in Fig. 5.5. The degree of truth for each level is calculated using Eqs. 5.4-5.6.

![Fuzzy logic membership functions](image)

Figure 5.5: Fuzzy logic membership functions.

\[
S_{\text{low}} = \begin{cases} 
1 & t \in [0, 0.33) \\
(0.66 - t)/0.33 & t \in [0.33, 0.66) \\
0 & t \in [0.66, 0.99) \\
0 & t \in [0.99, \infty) 
\end{cases} \quad (5.4)
\]

\[
S_{\text{medium}} = \begin{cases} 
0 & t \in [0, 0.33) \\
(t - 0.33)/0.33 & t \in [0.33, 0.66) \\
(0.99 - t)/0.33 & t \in [0.66, 0.99) \\
0 & t \in [0.99, \infty) 
\end{cases} \quad (5.5)
\]
5.3.4.2 IF-THEN Inference Rules

Two control strategies are considered in the fuzzy inference engine to activate the laparoscope or to make it stand still. The IF-THEN inference rules formulate which control strategy is appropriate based on the situation described by the NI and KF possibility levels. Nine inference rules are designed to determine how the laparoscope should be steered, activated, or fixed. The nine rules are listed as follows and also summarized in 5.1. In each inference rule, a logic AND operation is used to deduce the conclusion, as shown in Eq. (7), where $O_i$ is the output of $i$th inference rule, and $L_{NI}$ and $L_{KF}$ are the degrees of truth for the corresponding membership levels.

Rule 1: IF \( NI \) is low AND \( KF \) is low, THEN laparoscope stand still

Rule 2: IF \( NI \) is low AND \( KF \) is medium, THEN laparoscope stand still

Rule 3: IF \( NI \) is low AND \( KF \) is high, THEN laparoscope stand still

Rule 4: IF \( NI \) is medium AND \( KF \) is low, THEN activate the laparoscope

Rule 5: IF \( NI \) is medium AND \( KF \) is medium, THEN laparoscope stand still

Rule 6: IF \( NI \) is medium AND \( KF \) is high, THEN laparoscope stand still

Rule 7: IF \( NI \) is high AND \( KF \) is low, THEN activate the laparoscope

Rule 8: IF \( NI \) is high AND \( KF \) is medium, THEN activate the laparoscope

Rule 9: IF \( NI \) is high AND \( KF \) is high, THEN laparoscope stand still

<table>
<thead>
<tr>
<th>$NI_{low}$</th>
<th>$KF_{low}$</th>
<th>$KF_{medium}$</th>
<th>$KF_{high}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standstill</td>
<td>Standstill</td>
<td>Standstill</td>
<td></td>
</tr>
<tr>
<td>$NI_{medium}$</td>
<td>Activation</td>
<td>Standstill</td>
<td>Standstill</td>
</tr>
<tr>
<td>$NI_{high}$</td>
<td>Activation</td>
<td>Activation</td>
<td>Standstill</td>
</tr>
</tbody>
</table>

Table 5.1: IF-THEN fuzzy inference rules.
\[ O_i = \min(L_{NI}, L_{KF}) \]  

### 5.3.4.3 Inference Output

Among the nine IF-THEN rules, those inference results with the same control strategy are fused using a square accumulation method, as shown in 5.8 and 5.9, and the square root of the resultant is considered to be the likelihood score of that control strategy, which are the activation score (AScore) and standstill score (SScore), respectively. AScore measures how likely it is that the current gaze position is the surgeon’s visual attention and whether the laparoscope needs to be activated to focus on this new area. In contrast, SScore measures how likely it is that the laparoscope should stand still. AScore and SScore from the fuzzy engine across the entire input domain are shown in Fig. 5.6. When AScore is greater than SScore and also greater than 0.8 (a threshold value of attention confirmation), the system considers that visual attention of the surgeon outside the focus area has been recognized and then the robot is activated to follow the surgeon’s visual attention. The threshold value 0.8 is selected based on the analysis of the AScore and SScore plot in Fig. 5.6.

\[
\text{AScore} = \sqrt{O_4^2 + O_7^2 + O_8^2} 
\]

\[
\text{SScore} = \sqrt{O_1^2 + O_2^2 + O_3^2 + O_5^2 + O_6^2 + O_9^2} 
\]

![Figure 5.6: AScore and SScore.](image)
5.4 Experiments

Both simulations and benchtop experiments were performed to evaluate the fuzzy inference system. Simulation results were to evaluate the accuracy and time response of the fuzzy gaze interpretation method as the simulation environment eliminated the potential disturbances introduced by other components of the physical system. In the benchtop experiments with clinical participants, we focused on evaluation of the user experience of the overall system.

5.4.1 Visual Exploration Simulation

A visual exploration simulator was created which included an image with numbered blocks on it. A part of the image was displayed to the participants, and the participants were asked to explore the rest of the image using his/her gaze. When participants’ visual attention fell out of the focus area, the display was updated and the new display was the image centered at the attended block. Different attention recognition methods, the fuzzy logic inference method and the dwell-time method were tested using this simulator. The setup is shown in Fig. 5.7.

Figure 5.7: The experiment setup with a virtual simulator.
5.4.2 Experiment Trials

Benchtop experiments were performed at the Department of Urology, Denver Health Medical Center, Denver, CO. Five subjects (two were experienced surgeons and the other three were clinical/surgical researchers) participated in the experiments. They were asked to perform target exploration tasks laparoscopically in the virtual simulator first and then in a surgical training box with numbered blocks inside. A short introduction and demonstration were given first, and then they started to use the system directly without practice.

In the simulation tests, subjects were asked to change the field-of-view at least ten times using the fuzzy inference method and another at least ten times with the dwell-time method. Therefore, at least 50 view change samples for each method were collected. In the benchtop tests, the experiments were separated into two trials to avoid fatigue and other side effects caused by continuous and intensive view changing. Subjects were asked to focus on different areas in the surgical training box by steering the presented laparoscope system using their eyes. Both the fuzzy inference method and the dwell-time method were integrated into the system and subjects were asked to use both methods, respectively, in two trails to finish the exploration. During the trials, eye movement data and the system response time were recorded.

A threshold of 2.0 s was used for the dwell-time method based on our previous studies [92, 93]. Threshold values shorter than 2.0 s could easily cause faulty recognition of the surgeon’s visual attentions and led to incorrect movements of the laparoscope. The $T$ value for normalization in fuzzy membership functions was taken as 1.5 s.

After each view update, the subject was asked to verbally report all the numbered blocks in the focus area to force the subjects to perform a certain observation of the central area. After that, the subject was asked to change the view to focus on another numbered block outside of the focus area by looking at that block. This block could be one that the subject had seen earlier or had not previously seen. This simulated both conditions in which the surgeon knows the position of the target object before gazing upon it and in which the
surgeon is unaware of the object’s location and must search for it.

In the interaction with the robot, the subjects were asked to explore in the surgery simulator by steering the camera to focus on different areas using the presented system. Fuzzy inference method and dwell-time method were both integrated and the subjects needed to use both methods separately in two trails to finish the exploration.

5.4.3 Questionnaires

After the experiments, a questionnaire was given to the subjects. The questionnaire was customized based on system usability scale (SUS), a widely used global assessment of system usability [143], and new items specific for gaze studies were added in. SUS is a ten-item Likert scale which ranges from 0 to 100. Twelve new evaluation criteria specific for gaze studies were added and they were designed following the same scaling rule of SUS. Twenty-two criteria were scored separately as three subsections. Section 4.3.1 includes the original ten items and three new ones for evaluating the overall system usability. Section 4.3.2 constitutes five items for evaluating the user experience over repetitive tests, and Sec. 4.3.3 is for a comparison of user experience between the two attention recognition methods.

5.5 Results

5.5.1 Gaze Data Filter

The raw gaze data and refined gaze data using the adaptive sliding window filter are shown in Fig. 5.8 (the raw gaze data plot on the left and the refined gaze data on the right). The gaze data were collected when the subject looked at five visual targets on the screen and small red squares were the gaze points. It shows that even when a human tries to focus on a visual target, gaze data can still be quite noisy due to the involuntary rolling, nystagmus, drift, blink, and microsaccades. Gaze points are scattered around a visual target instead of concentrated on it, and there are gaze points that are shifted a great amount from the visual targets, which may be caused by blinks or eye tracking failure. For the refined gaze data, the noises are eliminated and the data concentrated on the visual targets, which is very helpful.
for the visual attention recognition.

![Figure 5.8: Comparison between the raw gaze data (left) with refined gaze data (right). The small squares are the gaze points. The big circles are the targets that the subject gazed upon on the monitor. The lines are the trajectory of the eye movements.](image)

5.5.2 Experiment Results

All the subjects successfully accomplished the exploration tasks using both attention recognition methods, which confirmed that the gaze-based interaction for laparoscopic view change is easy to use and participants can manage it effectively.

The response time (how long the subject needed to maintain his/her gaze on a visual target before his/her visual attention was recognized) was summarized. Using the dwell-time method, the five subjects made 57 view changes and the average response time was 2.063 s, with the maximum of 2.283 s and the minimum of 2.0 s. This result is consistent with the dwell-time threshold of 2.0 s. The slight variation is because the participants needed a little time to stabilize their eyes in order to focus on one target. In the fuzzy inference method, 58 view changes were generated by the participants and the average response time was 1.456 s, with the maximum of 2.517 s and the minimum of 0.833 s. The average response time is very close to the cutoff value of NIs high level ($1.5 \times 0.99 = 1.485$ s). The summarized response time is illustrated as a boxplot in Fig. 5.9. The red crosses were outliers whose
values were relatively far from the medium. The median for the dwell-time method was 2.05 s, and the median was 1.367 s for the fuzzy inference method.

![Box plot comparing dwell time and fuzzy inference methods](image)

Figure 5.9: Comparison of the response time in the dwell-time method and the fuzzy inference method.

### 5.5.3 Questionnaire Evaluation

#### 5.5.3.1 Overall System Usability

The extended SUS with three added evaluation questions has a score that ranges from 0 to 130 (a higher score means the system has a better usability). The three new questions focus on the system’s efficiency and accuracy. Each participant’s scores in two control methods are shown in Fig. 5.10. The average score of the dwell-time method is 88.5 and the fuzzy inference method is 100.

#### 5.5.3.2 User Experience Over Repetitive Tests

This section was to see if the user experience varies much after familiarization. It is composed of five questions. The total score is 50, and the results are shown in Fig. 5.11.
Figure 5.10: The summary of system usability score in two different methods. The higher score means that the system has a better usability. ASoFI: average score of the fuzzy inference method. ASoDT: average score of the dwell-time method.

The average score of the dwell-time method is 38.5, which is comparable to 38 for the fuzzy inference method.

5.5.3.3 Comparison of User Experience between Two Methods

This section is for the comparison of user experience on the fuzzy inference method over that in the dwell-time method. It mainly focused on user experience on the response time, comfort level, and intensity level of each method. The total score ranges between 0 and 40 (The value closer to 0 means that the dwell-time method is superior to the fuzzy inference method and the value closer to 40 means that the fuzzy inference method is better than the dwell-time method). In the results, the average is 28, with the maximum of 30 and the minimum of 22.5. The results are summarized in Fig. 5.12.
Figure 5.11: The summary of user experience over repetitive tests for each method. The average score of the dwell-time method is 38.5, which is comparable to 38 for the fuzzy inference method.

Figure 5.12: User experience comparison of the fuzzy inference method over the dwell-time method. The total score ranges between 0 and 40 (the value closer to 0 means that the dwell-time method is superior to the fuzzy inference method and the value closer to 40 means that the fuzzy inference method is better than the dwell-time method).
5.6 Discussion

In the experiments, the response time using the fuzzy inference method varied widely from 0.833 s to 2.517 s, based on the subjects’ historic visual behaviors in the last 3 s. The majority of the response times in the fuzzy inference method were less than those of the dwell-time method (1.367 s versus 2.05 s). The system using the fuzzy inference method was more responsive and activated more quickly when it could clearly tell that the current location gazed upon was the subject’s visual attention based on his/her historic visual behaviors. Otherwise, it was more conservative and waited to collect more data to confirm a new visual attention. In this way, the robot can respond efficiently without undermining its accuracy. These results show that the fuzzy inference method has advantages over the traditional dwell-time method in terms of efficiency. While both methods have good accuracy, the minimal threshold for the dwell-time method was 2.0 s and the fuzzy inference method had an average response time of 1.456 s.

From the results of questionnaire Sec. 4.3.1, it primarily shows that gaze is a promising interaction modality to allow the surgeons to naturally and effectively interact with the laparoscopic view because both methods have relatively high scores. The results also show that the fuzzy inference method has a better usability than the traditional dwell-time method. Specifically in both methods, the participants highly agree that the gaze-guided robotic laparoscope system is simple, straightforward, and easy to use (average score is 3.1 out of 4); people can learn to use it very quickly (average score is 2.8 out of 4) and they need to learn little before using it (average score is 3.4 out of 4). In the mental aspect, the subjects felt confident using this system (average score 2.9 out of 4).

In questionnaire Sec. 4.3.2, fuzzy inference method earned comparable score as dwell-time method. This shows that each method is consistent and repeatable. The participants started to use the system without practice and his/her user experience did not deteriorate as they became more familiar with the system in the second trial. This means that the empirical dwell-time threshold, the fuzzy inference’s cutoff value, and the minimization combination
operation are reasonable to provide a good user experience to the surgeons. And from the
results in questionnaire Sec. 4.2.3, it concludes that the participants preferred the fuzzy
inference method over the dwell-time method, as it can better understand users’ intent and
correctly and efficiently recognize users’ visual attention.

From our experiments, 57 samples for the dwell-time method and 58 samples for the
fuzzy inference method from five clinical participants were collected. These participants
had professional or closely related experience on the laparoscopic surgery and understood
the need of view change during the surgery. Analysis of these results clearly showed the
advantages of the fuzzy inference method in response time and user experience. Further
tests are needed to evaluate effectiveness of the presented system in animal operations or
with a bigger population of clinical participants.

5.7 Conclusion

In this chapter, a novel visual attention recognition method is introduced which is based
on fuzzy logic interpretation of eye-gaze movements at the pattern level. It is the first
time fuzzy logic has been used to recognize the surgeon’s visual attention based on eye-
gaze patterns. It can effectively and efficiently recognize the surgeon’s visual attention
from his/her natural eye-gaze behaviors. Integrating this fuzzy logic interpretation method
into the laparoscopic view interaction will constructively enhance the surgeon’s performance
during the operation. The entire system can effectively and correctly recognize the surgeon’s
attention on the surgical site and then automatically steer the robotic laparoscope to focus
on it. Therefore, intervention of the laparoscope can be eliminated from the surgeon’s to
do list. It can completely free the surgeon’s hands to manipulate the surgical instruments
while naturally adjusting the laparoscopic view using his/her eyes at the same time. Using
this system, the surgeon will be able to accomplish the operation more smoothly, as it can
eliminate switching back and forth between different control modes of the surgical instrument
and laparoscope. Consequently, the operation time can be reduced as well as the cost and
risk related to it.
CHAPTER 6
IMPLICIT INTENTION COMMUNICATION IN HUMAN-ROBOT INTERACTION THROUGH VISUAL BEHAVIOR STUDIES

A paper published at IEEE Transactions on Human-Machine Systems\textsuperscript{18}

Songpo Li\textsuperscript{19} and Xiaoli Zhang\textsuperscript{20}

6.1 Abstract

The emergence of assistive robots presents the possibility of restoring vital degrees of independence to the elderly and impaired in Activities of Daily Living (ADL). However, one of the main challenges is the lack of a means for effective and intuitive Human-Robot Interaction (HRI). While humans can express their intentions in different ways (e.g., physical gestures or motions, or speech or language patterns), gaze-based implicit intention communication is still under-developed. In this study, a novel, nonverbal implicit communication framework based on eye gaze is introduced for HRI. In this framework, a user’s eye-gaze movements are proactively tracked and analyzed to infer the user’s intention in ADL. Then the inferred intention can be used to command assistive robots for proper service. The advantage of this framework is that gaze-based communication can be handled by most people as it requires very little effort, and most of the elderly and impaired retain visual capability. This framework is expected to simplify HRI, consequently enhancing the adoption of assistive technologies and improving users’ independence in daily living. The testing results of this framework confirmed that a human’s subtle gaze cues on visualized objects could be effectively used for human-intention communication. Results also demonstrated that the gaze-based intention communication is easy to learn and use. In this study, the relationship

\textsuperscript{19}Primary researcher and author, graduate student, Department of Mechanical Engineering, Colorado School of Mines.
\textsuperscript{20}Author for correspondence, Assistant Professor, Department of Mechanical Engineering, Colorado School of Mines.
of visual behaviors with the mental process during human intention expression was studied for the first time to build a fundamental understanding of this process. These findings are expected to guide further design of accurate intention inference algorithms and intuitive HRI.

Index Terms: Gaze-based communication, implicit intention inference, assisted activities of daily living, human-robot interaction

6.2 Introduction

Since the early 1990s, robotic caregivers have become more accepted as caretakers for the elderly and infirm. Tremendous progress of assistive robots has been achieved in the commercial market [144–146] and in research laboratories [62, 147, 148]. Although these assistive technologies bring hope of independent daily living for the elderly and impaired, one obstacle is the lack of a suitable communication means to allow the elderly and impaired to effectively and intuitively interact with those assistive systems [149, 150]. This obstacle between human users and assistive robots may considerably undermine the acceptance of these assistive technologies. Some existing human-machine communication modalities include speech [151–153], joystick [154], gesture [106, 107, 111, 116], physical contact [155–157], and electromyography (EMG) [158, 159]. However, to use them, users must generate explicit service requests, which often involve additional motor motions. This complication is likely to become more severe with the development of assistive systems with higher function capabilities. To achieve effective and intuitive HRI, new techniques are needed that are easy to learn and are easy and effective to use for users with very limited motion or verbal capability.

Ideally, a robot should be able to proactively and implicitly understand a user’s intentions and automatically provide the desired services. Satisfying assistance should, therefore, be based on timely, accurate, and natural intention inference technologies. Researchers have investigated different human behaviors such as motion/gesture [70, 160], speech/language [161, 162], and electroencephalography (EEG) [163, 164] to infer a user’s intention in HRI. Another natural signal that can be used to infer a person’s intention in ADLs is the eye gaze, which has not been well-studied.
Studies have shown that visual attention is strongly linked to a human’s cognitive processes [165]. When a person visually focuses on an object, he/she is processing it in mind, and the duration of the focusing is closely related to the duration of the cognitive processes. Research has shown that a human’s mental state transformation can be reflected in the changes of visual behaviors [166]. Furthermore, the objects that are focused upon reflect a person’s certain desires. For example, a person who is thirsty and wants to have some water would naturally and intentionally look at drinking-related objects, such as a water machine, a cup, or a bottle of water, all of which are helpful to fulfill this desire. This strong and intuitive link to the intention of gaze makes it a promising signal for intention understanding. Furthermore, managing gaze requires very little effort for the users, and gaze inherently includes location information, which can be used to localize the visualized object and thus provide critical position information for robot execution. This advantage is manifested when multiple objects are involved for complex ADLs.

In this chapter, a novel, gaze-based implicit intention communication framework is introduced to enable users to intuitively and effortlessly express their intention (service requests in ADLs) using their eyes in HRI. For robots, this framework enables them to infer the user’s needs proactively as well as to take the corresponding action at the right time automatically. The main contributions of the research in this chapter are as follows:

1. A novel HRI framework is developed to enable implicit and intuitive gaze-based intention communication. This framework allows the human to intuitively express his/her intention to an assistive robot using natural gaze cues. Experiments were performed to demonstrate the effectiveness, accuracy, and intuitiveness of the presented approach.

2. In the human intention inference using gaze, semantic knowledge and behavioral knowledge are established and fused to improve the inference accuracy. The fused knowledge results in better accuracy over using either semantic or behavioral knowledge. Different fusion strategies were investigated in this study.
3. Human visual behaviors while humans are expressing their needs with gaze in complex ADLs are studied. The results reveal the relation between a human’s mental process with visual behaviors, which explains how visual attention is initialized and shifted from one object to another during intention expression with gaze.

6.3 Related Work

Gaze tracking is a technique that persistently estimates where a person is looking (gaze) based on monitoring his/her eye movements. Gaze estimation on a 2D display has been well-studied [1, 17, 18, 125, 167]. Although gaze-based Human-Computer Interaction (HCI) has been extensively studied, in which gaze is used as a pointing device to replace the function of the mouse for controlling the cursor [168, 169], gaze-based HRI is rare, especially at the intent level. HRI is fundamentally different from HCI, because robotic systems, which exhibit autonomy and cognition, are more complex, and are operated in dynamic, real-world environments [170, 171]. This difference brings new challenges for gaze-based HRI.

Currently, the attention of using gaze in HRI has been focused on using gaze location or direction to control a robot’s motion. Gaze has been used to steer a wheelchair by simply converting the gaze direction up, down, left, and right to the corresponding control commands of driving forward, backward, left, and right [14–16]. A similar strategy has also been applied to remotely steer mobile robots [172, 173], quadcopters [88, 174], and robotic laparoscope systems [76, 89, 91]. Attempts of activating the control command of a robot by gazing at an on-screen button have also been done [91, 175]. In [13], the gaze direction served as a pointing line to the visualized object along which the robot could search for the visualized object and retrieve the first-found object.

Little effort has been put forth to investigate how gaze can be utilized for high-level intention communication in HRI. In our preliminary work, we demonstrated how a person’s complex intention, which involves multiple objects, can be inferred by monitoring his/her gaze [176]. Similar feasibility studies of gaze were also reported in [177–179]. However, how to integrate the gaze-based intention communication into effective, practical, and intuitive
interaction between the human and the robot is still an open problem. To achieve this, the following issues have to be answered:

1. A responsive and effective HRI framework specifically designed for implicit gaze-based intention communication is missing. How to apply the gaze-based intention inference practically and naturally in HRI in the real world needs to be investigated and evaluated.

2. Fundamentally, using gaze for intention inference requires the differentiation of visual attention from normal behavioral eye movements, which is known as the Midas touch problem [132] (in other words, everywhere you look gets activated. Additional confirmation is needed to confirm visual attention). The common solution is to ask the user to deliberately blink or prolong the gaze, which acts as confirmation of the visual attention. To reduce the deliberate effort using gaze modality required from the user, a new approach that can implicitly detect the visual attention is needed.

3. A sophisticated knowledge base is needed to support robust intention inference using eye gaze. This knowledge represents how the visualized objects are correlated with certain types of intention. It should be generic and have practicability such that it can support robust human intention inference. Commonly, the knowledge is built through either questionnaire or observation of the practices, which overlooks the important difference between them. How this difference affects the inference results and how it can be effectively used to improve the performance are requirements to study.

4. There is a lack of understanding on how humans visually express their intention with respect to objects. Fundamental studies are needed to explore what objects humans think about and look at during intention expression and how they are related. This relationship can guide interaction design and inference algorithm design.
6.4 Method — Gaze-based Implicit Intention Communication Framework

The application scenario of this gaze-based intention communication HRI framework includes a homecare setting with an assistive robot, where a user who is confined to a bed or chair has to command a mobile assistive robot to perform ADLs for him/her. The mobile robot moves around in different rooms and feeds back live scene video to the user. The user’s gaze on the scene image is tracked and analyzed in order to infer his/her intention. The confirmed intention is then sent to the robot as action commands. The steering of the robot is not covered in the scope of this study. This framework consists of multiple main components: visual attention detection with detecting the visualized object, human intention knowledge, intention inference and the interaction with the human user. The following sections elaborate on each component.

6.4.1 Attention-based Object Recognition

When a person’s eyes focus on an object, the visual attention can be manifested as a cluster of gaze points on that object. To detect the visual attention, a visual-behavior-based classifier was developed to differentiate a specific eye-gaze pattern of visual attention from natural visual behaviors. This specific eye-gaze pattern is named ‘intentional gaze’, which means the user is looking at the visual target with intention of manipulating it, while the rest are called ‘intention-free gaze’ (e.g., searching or exploring, looking at an object without manipulatory intention). Compared with the common prolonged gaze method and deliberate blink, attention detection using natural visual behaviors does not require the users to learn and memorize special instructions or perform artificial behaviors, therefore reducing the deliberate effort required from the user. Following the detection of each attention, a backward Minimal Spanning Tree method (MST) [180] is used to cluster gaze points on the visualized object, and an equivalent circle is calculated to represent the cluster of gaze points. Lastly, from the overlap of the human attention region (represented by the equivalent circle) with the scene image, the visualized objects can be detected and recognized.
6.4.1.1 Intentional Gaze Recognition for Attention Detection

The visual-behavior-based classifier for detecting intentional gaze is based on a Support Vector Machine (SVM) classifier [181], and the classification procedure is shown in Fig. 6.1. This classifier utilizes eye-gaze features extracted from natural visualization, which does not force the users to perform any unnatural visual behavior, such as the prolonged gaze or a deliberate blink. The features used for this classifier include gaze dwelling time, pupil size/gradient variation, gaze speed and gaze concentration. These features were selected based on experimental observation and review of literature [182]. The classifier is trained before it can distinguish the intentional gaze from the intention-free gaze. During utilization, the extracted eye-gaze features are input into the classifier, and the output is its best guess of which visualization condition the user is in. Only when the intentional gaze is detected is the visual attention detected.

![Feature Extraction](https://example.com/feature_extraction.png)

**Figure 6.1:** Detection of intentional gaze using a SVM classifier.

6.4.1.2 Equivalent Representation of Visual Attention

After the detection of visual attention, a backward MST method is used to cluster the gaze points. MST uses the minimal spatial distance from one gaze point to all the other points of one cluster as the criterion to determine whether the gaze point belongs to the cluster or not. The clustering procedure is done backward, starting from the gaze point, at which an intentional gaze is detected. The raw gaze points are filtered using an adaptive sliding window filter [76] to eliminate the influence of superimposed saccades and involuntary eye movements such as rolling, nystagmus, drift or microsaccades. Afterward, a circle that can cover the gaze points in a cluster is calculated to represent that cluster, as shown in Fig.
6.2. The cluster with the maximal amount of gaze points will be considered as the attention location.

Figure 6.2: MST clustering for visual attention representation. Gaze point cluster is presented by a dashed-line circle with its center shown as a red dot. The visual attention is the gaze cluster with the maximal amount of gaze points. The gaze path is shown as the arrow direction. The clustered gaze points are shown as green and the rest are shown as black.

6.4.1.3 Attention-based Object Detection and Recognition

By overlapping the attention circle with the scene image, the object that the user has gazed at intentionally can be detected and recognized. To facilitate this procedure, the scene image can be segmented first using a Normalized Cut method [183]. A Scale-Invariant Feature Transform (SIFT) method [184] can be used along with a multiclass SVM classifier [185] to recognize the objects.

6.4.2 Human Intention Inference
6.4.2.1 Knowledge of Human Intention

To be able to recognize human intention from the objects that a person has looked at intentionally, the system needs knowledge about the correlation between objects and intention. This knowledge is a representation of how objects are related to a certain type of intention or how a type of intention can be fulfilled with objects. This knowledge is
modeled as a Naive Bayesian Graphic Probabilistic model [186] as shown in Fig. 6.3. Objects ($O_j, j = 1 \sim M$, where $M$ is the total number of objects) are linked to various types of human intention ($I_i, i = 1 \sim N$, where $N$ is the number of total possible types of intention) with different strengths of correlations ($p_{ji}$, correlation strength between $O_j$ to $I_i$), which represent the possibility that $O_j$ appears when $I_i$ happens. In the Naive Bayes model demonstration (Fig. 6.3), strong correlations are shown with short linkages, while weak correlations are shown with longer linkages. No connection means no correlation. The strength of correlation $p_{ji}$ is determined by the appearance of this correlation using 6.1, where $A_{i,j}$ is the appearance of correlation between $I_i$ with $O_j$, and $A_i$ is the appearance of $I_i$

\[ p_{ji} = \frac{A_{i,j}}{A_i} \]  

(6.1)

Two intention knowledge bases are established using surveys and experimental observations, respectively. In the survey, a set of objects and a set of various types of intentions

![Figure 6.3: A Naive Bayes model for representing object-intention knowledge database. $O_j$ is the $j^{th}$ object and $I_i$ is the $i^{th}$ intention. Each linkage indicates the conditional appearance probability ($p_{ji}$) of an object to one type of intention. A longer link means a lower probability, and vice versa.](image)
are provided to each subject, and the subject selects a subset of objects to accomplish one type of intention. The knowledge built from the survey is named as semantic knowledge \( V_s \), which represents how objects are logically/literally related to a type of intention. The experimental observation method is used to experimentally observe how a subject expresses certain intention by gazing at the objects, and it builds the behavioral knowledge \( V_b \). In both methods, participants are free to select any number of objects. The reported intention and the corresponding objects are formatted in intention-object pairs as: \( I_i: O_1, O_2, O_3, \ldots, O_j \), indicating one intention instance \( I_i \) has occurred with associated objects \( O_1, O_2, \) and \( O_3, \) etc.

Knowledge bases \( V_b \) and \( V_s \) are two views of how a certain type of intention is related to the objects. These two views, \( V_b \) and \( V_s \), are fused at the utilization stage using different fusion strategies illustrated as 6.2-6.4. In the linear summation fusion strategy 6.2, \( \alpha \) is a fusion coefficient and varies from 0 to 1. In 6.3 and 6.4, the final conditional probability is taken as the maximum or minimum, respectively.

\[
V = \alpha V_b + (1 - \alpha) V_s \tag{6.2}
\]

\[
V = Min(V_b, V_s) \tag{6.3}
\]

\[
V = Max(V_b, V_s) \tag{6.4}
\]

### 6.4.2.2 Human Intention Inference using Gaze Cues

At the utilization stage, human intention is inferred using a Naive Bayes classifier from the visualized objects. With visualized objects, each possible intention’s conditional possibility can be calculated, and the one with the highest possibility is considered the most possible intention. The classifier is formulated as 6.5, in which \( 1, \ldots, j \) is a list of objects that the user has gazed upon.

\[
\hat{y} = \arg \max_{i \in 1, \ldots, N} p(I_i|O_j; j \in 1, \ldots, k) \tag{6.5}
\]

130
6.4.3 Gaze-based Intention Communication

Robotic assistance is provided to the human user, so it is essential to consider how to design the interaction to be intuitive for the user. In this gaze-based HRI framework, several simple and intuitive rules are designed to interact with the user. One physical button is added to control the interaction. This button is clicked a limited number of times only when a user initializes, finishes or confirms the intention, which remains in the capability of the elderly and impaired. Though the physical button is used in the current prototype, it could be replaced by a virtual button displayed on the screen that could be activated by human gaze.

6.4.3.1 Trigger and Complete the Gaze-based Communication

In real-world applications, it is critical for a robot to know when the user begins to express intention and when the expression is complete. The button will be pressed to notify the system before the user starts to express intention, and will be pressed again after the expression is completed. The system uses the recognized visual attention between these two button clicks to perform the intention inference.

6.4.3.2 Confirm or Deny an Inference Result

It is necessary to check with the user about the correctness of the inferred result, as there are chances that the inference results are wrong due to distractive environments or faulty gaze tracking. The inferred result is conveyed to the user by displaying it on the middle of the screen. To confirm the inferred intention, the user simply clicks on the button. If the system does not receive the confirmation within five seconds (this duration can be adjusted for an individual), it considers the inferred result to be incorrect and proceeds to check the second possible intention with the user. Otherwise, the user can simply deny all the inferred results by holding the button, and re-express it after the system prompts to re-express. In the second inference, the previously denied results are eliminated from the possible candidates.
6.4.4 Gaze-based Implicit Intention Communication Framework

The aforementioned modules are integrated to form the entire framework, which is illustrated in Fig. 6.4. The user is looking at the live video fed back from the robot side. During the intention expression procedure, the robot is assumed to hold a stable scene image containing objects for the user. Once the user triggers the intention inference engine by clicking the button, the system starts to analyze the eye gaze data to extract the gazed objects and infer the user’s intention from them. The inferred result is presented to the user on the monitor for confirmation. For the confirmed result, one command is then sent to the assistive robot following the format: "Do task ####, with the object M at location XX, and the object N at location YY." The location of the visualized object can be calculated from the object’s location in the scene image as a pointing vector from the scene camera to the object.

Figure 6.4: Gaze-based intention communication framework illustration.
6.5 Experiments

6.5.1 SVM Classifier Training for Visual Attention Detection

The SVM classifier was trained before it could successfully detect the visual attention. As a prerequisite of the experiments, the SVM classifier for visual attention detection was trained with 14 volunteers. These 14 volunteers also participated in experiments for evaluation of the framework. To collect the training data, participants were asked to observe an artificial image with numbered blocks. When a certain block was being viewed intentionally, a button was pressed to indicate to the system that currently there was visual attention on that concentrated block. Afterward, the display was automatically updated to focus on that visualized block. Eye-gaze features were extracted from this confirmed visual attention and used as positive training data. Negative training data was the recorded gaze data that was not confirmed as attention. During data collection, we asked the participants to relax and observe the image naturally.

6.5.2 Establishment of Gaze-based Intention Knowledge

Two small-scale knowledge bases were separately built with surveys and experiments, which involved four types of service-related intention and 14 manipulatory objects. An identification (ID) number was given to each object, as shown in Table 6.1. Some of them might not be related to the studied intention and were used as visual distractions. All the manipulatory objects were included in an artificial kitchen image, shown in Fig. 6.5. The intention knowledge was established with 25 subjects (Twenty subjects participated in the survey to build the semantic knowledge, and another five subjects participated in the experimental observation to build the behavioral knowledge). The collected intention-object pairs were used to build the semantic knowledge and behavioral knowledge, respectively.

The four potential intentional tasks were:

T1: "Prepare a cup of coffee" (Pre. Cof.);

T2: "Prepare oatmeal for breakfast" (Pre. Brf.);
Figure 6.5: The artificial kitchen image with labelled objects.

Table 6.1: The ID numbers for 14 manipulatory objects

<table>
<thead>
<tr>
<th>Object</th>
<th>ID</th>
<th>Object</th>
<th>ID</th>
<th>Object</th>
<th>ID</th>
<th>Object</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cup</td>
<td>O1</td>
<td>Coffee pot</td>
<td>O2</td>
<td>Milk box</td>
<td>O3</td>
<td>Medicine container</td>
<td>O4</td>
</tr>
<tr>
<td>Dishwashing liquid</td>
<td>O6</td>
<td>Spoon</td>
<td>O7</td>
<td>Bowl</td>
<td>O8</td>
<td>Oatmeal</td>
<td>O9</td>
</tr>
<tr>
<td>Kettle</td>
<td>O11</td>
<td>Microwave</td>
<td>O12</td>
<td>Cleaning sponge</td>
<td>O13</td>
<td>Washable object</td>
<td>O14</td>
</tr>
</tbody>
</table>
T3: "Take medicines" (Tk. Med.);

T4: "Wash a washable target (the target could be a cup, coffee pot, spoon, or bowl.)" (Wash).

6.5.3 Experiment Setup and the Procedure

The experiment focused on visual attention detection and intention inference, which are critical parts of the overall framework. The homecare scenario was simulated with the feedback scene image displayed to the user, as shown in Fig. 6.6. The scene image showed a kitchen with visible objects. A table-stand GP3 eye tracking system from Gazepoint was used to track where the subjects were looking on the monitor.

During the experiment, the subjects sat in front of a monitor that was displaying the kitchen scene image. Participants tried to express their intention to the assistive robot by looking at those objects. Their eye movements were recorded along with the gaze positions. From the eye-gaze data, their visual attention was detected using the SVM classifier, and then the visualized objects were recognized. From the gazed objects, participants’ intention was inferred using the Nave Bayes classifier. During the experiment, subjects’ actual visual attention and intention were reported and recorded as ground truth to evaluate the perfor-
formance of the system. To demonstrate the gaze-based modality is effective and intuitive, after
the gaze modality, participants were also asked to use a mouse as an alternative modality
to indicate the objects associated with each particular intention by clicking on them. In
authors’ previous work [76], which used gaze to control a robotic laparoscope holder, the
authors have demonstrated that interpreting a person’s natural visual behaviors for atten-
tion detection can result in a more intuitive and user-friendly interaction interface than that
using the prolonged gaze. Thus, we compared our gaze modality using natural visual be-
haviors for attention detection with the mouse modality, which is one of the most popular
interaction modalities and has a great usability. Even though the mouse modality is not
applicable for most disabled users, it is still a very effective baseline to show the usability
of the gaze modality due to its popularity and great usability. The rest of the setup using
the mouse was the same as using gaze. After the experiments, questionnaires were provided
to the participants to evaluate their user experience in both modalities. The well-validated
USE questionnaire [102] was used for usability evaluation. It includes 30 items covering four
dimensions of usability: usefulness, ease of use, ease of learning, and satisfaction.

6.6 Results

6.6.1 SVM Classifier Training for Visual Attention Detection

Different eye-gaze characteristics are shown at visualization stages of the intentional gaze
and intention-free gaze. Two particular eye-gaze features are shown here to demonstrate
the difference. During the intentional gaze, the gaze of a person dwells relatively longer and
concentrates more than the gaze distribution during the intention-free gaze, which is shown
in Fig. 6.7. At the transition from the intention-free gaze to intentional gaze, the size of the
pupil tends to increase, as shown in Fig. 6.8.

Ninety-three sets of positive training data and 68 negative training datasets were collected
for the SVM classifier training. Those training data sets were used to train the classifier,
and the overall training success rate was 79.5%. More detailed training performance is
summarized in Table 6.2. The formulated classifier tends to more easily miss real attention
Figure 6.7: Heat map of gaze patterns during intentional gaze and intention-free gaze in a kitchen environment. The height of each peak is proportional to the duration of gaze dwell time.

Figure 6.8: Plot of pupil size samples from the subjects. At transition point from the intention-free gaze to intentional gaze, the pupil size increases. The horizontal axis is time, and about 3 seconds of pupil data was trimmed in each sample.
than produce fake detections.

Table 6.2: SVM classifier for attention detection training performance.

<table>
<thead>
<tr>
<th></th>
<th>Classified as positive</th>
<th>Classified as negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive training data</td>
<td>76.3%</td>
<td>23.7%</td>
</tr>
<tr>
<td>Negative training data</td>
<td>16.2%</td>
<td>83.8%</td>
</tr>
</tbody>
</table>

The SVM classifier was then used in the experiment to detect each participant’s visual attention. Note that in the experiment, gaze data was not pre-labeled according to the visual attention. Two criteria were used for performance evaluation: correct detection rate and fake detection rate. The correct detection rate $R_{crct}$ was calculated using 6.6, while the fake detection rate $R_{fk}$ using 6.7. In the equations, $N_{va}$ is the actual visual attention number produced by the participants, $M_{va}$ is the visual attention number detected by the classifier, $m_c$ is the correctly detected visual attention among $M_{va}$, and $m_f$ is the fake detection number among $M_{va}$. The overall correct detection rate is 72.3%, which means that, among the visual attention produced by the subjects, 72.3% was recognized successfully. The overall fake detection rate of 32.3% means that, among the detected attention, 32.2% was fake. $R_{crct}$ and $R_{fk}$ of each participant are plotted in Fig. 6.9. $R_{crct}$ is plotted as a white bar and the $R_{fk}$ is plotted as a gray bar.

$$R_{crct} = \frac{m_c}{N_{va}} \quad (6.6)$$

$$R_{fk} = \frac{m_f}{M_{va}} \quad (6.7)$$

The performance of the SVM attention classifier varied significantly among different subjects. The second subject, S2, had the best $R_{crct}$ at 100%, which means all of his visual attention was detected. The results demonstrated the SVM classifier and the selected eye-gaze features were able to detect users’ visual attention during the observations under natural visual behaviors. The third subject, S3, had the worst $R_{fk}$, 62.5%. To achieve the best performance for each user, individual training of the classifier is recommended.
Figure 6.9: The performance of the SVM attention classifier in the experiment. The performance is evaluated by correct detection rate $R_{crd}$ shown as a white bar pointing up and fake detection rate $R_{fk}$ shown as a gray bar pointing down.

### 6.6.2 Visual Object-Intention Knowledge Base

Eighty intention-object pairs were collected for the semantic knowledge building and 20 pairs for the behavioral knowledge, which were fed into the Naive Bayes models for training the correlations separately. The trained models were fused using the three fusion strategies 6.2-6.4, respectively. One example of the correlations between visualized objects with each type of intention is shown in Fig. 6.10 (it is a fused Naive Bayes model using 6.2 with $\alpha=0.9$). In the correlation plot, the horizontal axis is the object and the vertical axis is the possible intention. A dot represents the correlation strength of one object to a specific type of intention. For the convenience of display and perception, the correlations were clustered into high, medium, and low strength levels and were notated using different sized dots.

Even though different participants might have visualized different objects for particular intention, common objects were widely observed among most of the participants. For example, in “Prepare a cup of coffee,” coffee pot, cup, and milk were selected by most of the participants. Those objects selected with high possibilities are called dominant objects (strong correlation with one type of intention). In contrast, objects with low possibilities
are called optional objects. One type of intention has a unique set of dominant objects, as summarized in Table 6.3. For example, "Prepare oatmeal for breakfast" has the dominant objects oatmeal, bowl, and kettle, which are different from those of "Prepare a cup of coffee." This difference infers that a specific type of intention can be distinctively represented by its dominant objects. On the other hand, although the dominant objects were shared among the participants, there were notable variances for the optional objects, such as tap, spoon, bowl and kettle in "Prepare a cup of coffee." This variety was mainly caused by specific situations, individual differences, or preferences, which increased inference difficulty.

Table 6.3: Summary of dominant objects for each intention

<table>
<thead>
<tr>
<th>Intentions</th>
<th>Dominant objects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prepare a cup of coffee (Pre. Cof.)</td>
<td>Cup, coffee pot, milk</td>
</tr>
<tr>
<td>Prepare oatmeal (Pre. Brf.)</td>
<td>Bowl, oatmeal, kettle</td>
</tr>
<tr>
<td>Take medicine (Tk. Med.)</td>
<td>Medicine</td>
</tr>
<tr>
<td>Wash an object (Wash)</td>
<td>Tap, dishwashing liquid, washable target</td>
</tr>
</tbody>
</table>

6.6.3 Intention Inference

Intention inference results based on three different fusion strategies of the semantic knowledge and behavioral knowledge are shown in Fig. 6.11. The horizontal axis is different
fusion strategies, and the vertical axis is the success rate of intention inference. The first 11 histograms on the left are the linear summation fusion strategy 6.2 with different fusion coefficients. Max and Min correspond to fusion strategies 6.3 and 6.4, respectively. The gray bars show the intention inference results of the whole framework, which combine the performance of both SVM-based attention detection and intention inference engine. The white bars show the success rate of the intention inference engine only, in which visualized objects were from manually recorded visual attention during the experiment.

Figure 6.11: Intention inference correctness rate with different knowledge fusion strategies illustrated as 6.2-6.4. The white bar is the correctness rate of the intention inference using recorded objects and the gray bars are those using the SVM detected objects.

From the results, three observations were made:

1. The SVM attention classifier significantly affected the overall performance. The performance of the gaze-based intention inference framework was based on two parts: user attention detection and intention inference. The attention detection module produced inputs for intention inference. If the attention extraction module failed to detect the real attention and produced fake detections, the performance of the intention inference was undermined. This can be observed by the comparison between the overall success rate (gray) and the success rate of intention inference only (white). Therefore, visual attention detection is critical to the whole framework. The results showed the main
failure was caused by the false attention detection. On the other hand, as multiple objects were used in intention inference, failures of attention detection were tolerated to some extent by the inference process.

2. Both semantic knowledge and behavioral knowledge affected the inference results. In the linear summation strategy, the success rate varied from 66.07% to 75.00%. With the increase of the behavioral knowledge’s contribution, the success rate improved. The semantic knowledge collected from the survey can establish a generic relationship between various types of intention with objects; however, it is too idealistic to appropriately model how a human would visually behave in practice, which are results of individual adaptation to specific situations. The behavioral knowledge has its own drawbacks. For example, it cannot capture complete information of the optional objects for a given intention, and it is costly in time and human labor for the establishment. Considering both knowledge resources, intention inference contributed greatly to improve inference results and practical utilization.

3. The Min strategy had comparable results with the best performance of the linear summation strategy. The Max strategy had the worst performance among all the different fusion strategies. Linear combination strategy is recommended for knowledge fusion in this specific application as the performance is predictable.

One detailed inference performance of the whole system is illustrated as a confusion matrix shown in Fig. 6.12. The linear combination strategy with $\alpha = 0.9$ was used for knowledge fusion. The horizontal axis is the target intention and the vertical axis is the inferred intention. The correctness for each type of intention and overall correctness is summarized at the bottom row. The intention ”Prepare a cup of coffee” was inferred correctly in 92.9% of the cases, and ”Prepare oatmeal for breakfast” was inferred 100% of the time. ”Wash a washable target” was inferred correctly in 78.6% of the cases, and ”Take medicine” was inferred correctly 28.6% of the time. The overall correctness rate was 75.0%. Notably,
71.43% of the inference error occurs for the intention "Take medicine", which only has one dominant object. With fewer dominant objects, it is more difficult to characterize the intention from the object aspect, and the inference engine has less tolerance for the mistakes of the SVM attention detection classifier.

Figure 6.12: Intention inference confusion matrix with linearly fused knowledge. For intention "Prepare a cup of coffee", it is correctly inferred in 92.9% cases. And the overall correctness rate is 75.0%.

### 6.6.4 Questionnaire for Usability Evaluation

Usability of the gaze modality using the USE questionnaire from 14 subjects is summarized in Fig. 6.13. In the questionnaire, there are four subsections to separately evaluate usefulness, ease of use, ease of learning, and overall satisfaction. *p-values* were performed between two modalities on each subsection, and the p-values were also plotted in Fig. 6.13. The full usability score for each section is 7.0, and the greater the values, the better the usability. The gaze modality earned a relatively high score in all the sections. It can be concluded that the gaze-based intention inference for HRI is useful (Usefulness median score is 4.63). The system is easy to learn and use for intention expression (Ease of use median score is 4.68, and ease of learning score is 6.0). Also, the participants are satisfied with using
it (Satisfaction score is 5.29).

![Box plot of questionnaire scores for gaze and mouse modalities.]

**Figure 6.13:** Questionnaire score summary of gaze (blue) and mouse (red) modalities using box plot.

The scores of the gaze modality were lower and had a larger variation than those of the mouse modality. However, the differences between two modalities are not statistically significant as only in subsection usefulness and ease of use the p-values are less than 0.05. The lower usability scores of the gaze modality were expected for two main possible reasons. One is that gaze is a relatively new interaction modality, especially at the intention level, and is underdeveloped when compared to the well-developed and widely accepted mouse. It is a common fact that humans need time to get familiar with and accept new technologies by continuously encountering them. Before this occurs, the understanding of those new technologies may contain bias which also results in evaluation bias. Even though practices of the gaze modality was provided before the experiments, insufficient training may still exist in the experiments and result in lower evaluation scores compared to the mouse modality. Different people have different perceptions of the same thing, and the perception difference can vary more greatly when they are facing a newborn technology. Another possible reason for having lower scores is that all subjects were asked to use the gaze-based interface before the mouse-based interface. The learned experience about the testing task in gaze modality
might have helped them in the mouse modality, which resulted in a better user experience. This ordering might have caused both lower scores and a larger variation of the gaze modality. Which reason had more influences in the final result is left as an open problem.

Aside from the lower usability scores compared to those of the mouse modality, the subjects successfully managed the gaze-based interface to accomplish the given task with ease, although it was their first time using the gaze modality. Moreover, the scores of the gaze modality had high usability result marks (all the median scores were higher than 4.63, and the ease of learning measure was as high as 6). Those results have strongly proved the promising potential of gaze modality in HRI. One of our research focuses in the future will be the investigation of methods and designs that can further improve gaze modality’s usability during practical usage.

6.6.5 Mental Processes Revealed by Visual Behaviors during Intention Expression with Gaze

During the experiment, special attention was paid to understanding human’s mental processes and visual behaviors during intention expression using gaze. Participants were asked to report the first object that came up in their minds, \( O_{\text{mind}} \), while they started to express intention. The objects that the participants actually gazed at are notated as \( O_{gaze} \). The relationships between \( O_{\text{mind}} \), \( O_{gaze} \) and the dominant objects were studied separately.

6.6.5.1 \( O_{\text{mind}} \) vs. Dominant Objects

For each type of intention, the frequency of a certain object acting as \( O_{\text{mind}} \) is displayed in Fig. 6.14. Take "Prepare a cup of coffee" as an example. When the subjects tried to express the intention, 71.43% of subjects initially thought about the dominant object coffee pot, 21.43% of the subjects initially thought about the dominant object cup, and the rest of them thought about some other optional objects.

From the plot, two observations were made:
1. It shows that when humans try to express intention, the first thing that comes into mind will very likely be one dominant object of that intention which has a strong correlation with the intention.

2. Multiple dominants of one intention do not appear as $O_{\text{mind}}$ with similar possibilities. Instead, there is a priority among the dominant objects. For example, ”Prepare a cup of coffee” has three dominant objects: coffee pot, cup, and milk. However, the possibility of coffee pot is much greater than cup, and milk did not appear in the experiment as $O_{\text{mind}}$. This priority was not reflected in the semantic knowledge or behavioral knowledge, as coffee pot, cup and milk had very close strengths for ”Prepare a cup of coffee” in the knowledge (Fig. 6.10).

6.6.5.2 $O_{\text{mind}}$, vs. $O_{\text{gaze}}$

The relation between $O_{\text{mind}}$ with $O_{\text{gaze}}$ is plotted in Fig. 6.15, in which it summarizes the order that $O_{\text{mind}}$ was gazed at. Take ”Prepare a cup of coffee” as an example. When subjects were expressing the intention, the object that first came into the users’ mind was gazed at first in 42.86% of the cases, second in 42.86% of cases, third in 7.14% of cases and
later in 7.14% of cases.

![Graph showing gaze order of objects](image)

Figure 6.15: The order of $O_{\text{mind}}$ being gazed by subjects when they expressed a type of intention.

The results demonstrate that the object that first comes into a person’s mind will be gazed at as early as possible and very likely will be in the first two. Generally, the subjects will look for the $O_{\text{mind}}$ first from the scene. In most of the cases, they could find it first. However, there is still a chance the first object gazed at is something else, which can be caused by the cluttered environment, obviousness of an object, or the relative distance from the current gaze point to the object. Even if the search process of $O_{\text{mind}}$ can be interrupted, the user will come back to find it sooner or later.

### 6.6.5.3 $O_{\text{gaze}}$ vs. Dominant Objects

For each intention, the order of each dominant object being gazed at is summarized in Fig. 6.16. Take ”Prepare a cup of coffee” as an example. When the subjects expressed the intention, the dominant object cup was looked at first in 35.72% cases, second in 28.57% cases and third in 28.57%, so that it appeared in 92.86% of intention expression cases.

The plot shows that dominant objects will be frequently looked at during the expression and often be in the first three. The dominant objects can strongly indicate one specific
intention, and people try to look at them in the expression at the earlier stage. In addition, other related objects will also be looked at based on a user’s preference and the environment’s setup. It also shows that even though all dominant objects are tightly linked with a kind of intention, different priorities are demonstrated among them during the intention expression, like the milk box for “Prepare a cup of coffee.”

The plot shows that dominant objects will be frequently looked at during the expression and often be in the first three. The dominant objects can strongly indicate one specific intention, and people try to look at them at the earlier stage. In addition, other related objects will also be looked at based on a user’s preference and the environment’s setup. It also shows that even though all dominant objects are tightly linked with a kind of intention, different priorities are demonstrated among them during the intention expression, like the milk box for “Prepare a cup of coffee.”

In summary, when a subject has intention and he/she wants to express it, an object that has a strong correlation to this intention will come into his/her mind initially, and this object is very likely to be one of the dominant objects of that intention. Then the subject starts to look for it in the environment. Because of environmental effects, the subject may notice
some other dominant objects earlier than the object in his/her mind, but the subject will come back to search for it again. In addition, most of the dominant objects will be looked at by the subject along with some optional objects to specify the intention.

6.7 Discussion

Promising results and valuable findings are obtained from the experiments. In the future, these findings will be used to redesign the intention inference algorithm for better accuracy.

6.7.1 Personalized SVM for Attention Detection

The performance of the visual-behavior-based attention detection varies for different subjects due to the individual differences in their visual behaviors. To achieve accurate attention detection for each individual, the SVM classifier will be trained personally in later work. The procedure will be similar to the procedure presented in this chapter.

6.7.2 Specially Treating Intentions with Less Dominant Objects

In the experiment, the intention "Take medicine" has a smaller number of dominant objects and its inference had the worst accuracy. For a type of intention with a lower number of dominant objects, the inference is more sensitive to the error produced in the attention detection procedure and distractions from the environment, as the first case misses the relevant inputs and the second one produces more irrelevant ones. The personalized SVM classifier will improve the accuracy of attention detection, and thus relieve this problem to a certain degree. Furthermore, those kinds of intention with less dominant objects can be summarized into a particular group. During intention inference, a second inference engine will be developed to perform a separate inference, which only considers intention belonging to this particular group. In the second inference engine, the potential intention pool can be narrowed down, and it can significantly improve the accuracy of the intention with less dominant objects. This second inference engine’s results will be combined with those of the inference engine presented in this chapter as the final output. And how to combine these
two inference engines will be left as future work.

6.7.3 Weighting the Dominant Object

From the built knowledge, it is noted that for each intention there is a particular set of objects (dominant objects) that are linked to the intention with particularly high strengths and are distinctive from other types of intention. Those dominant objects strongly imply the occurrence of a certain type intention. In future work, a particular weight $\gamma(j|i)$ will be assigned to the object to indicate if $O_j$ is a dominant object for $I_i$. $\gamma(j|i)$ will be 1 if $O_j$ is not a dominant object for $I_i$; otherwise it takes a value that is greater than 1. Then the human intention will be inferred through 6.8 using gaze cues. $\gamma$ will increase the relative possibility of $I_i$ when its dominant objects are viewed during intention expression. The value of $\gamma$ will be learned with testing trails.

$$\hat{y} = \arg \max_{i \in 1, \cdots, N} \gamma(j|i)p(I_i|O_j); j \in 1, \cdots, k$$ (6.8)

6.7.4 Weighting the Object Viewed Earlier

During intention expression, the human turns to look at those objects earlier, which are tightly linked to the intention. In most cases, they are the objects that first come into the user’s mind and/or the dominant objects. A set of weights could be assigned to the first three objects based on the order they are viewed, $\rho_1$, $\rho_2$, and $\rho_3$. They can highlight the significance of the first three viewed objects in intention inference. The implementation of $\rho_1$, $\rho_2$, and $\rho_3$ is left as future work.

6.7.5 Potentials of Gaze in Robotic Operation

In this chapter, we focused on the intention (object manipulatory task) expression of a human user to an assistive robot, which is a key component in commanding an assistive robot for daily activity services. To achieve a fully functioning assistive system, related research needs to be investigated. The work of using gaze to navigate a mobile robot or drone, reported in [88, 172–174], could be integrated with the research presented in this
chapter. Such integration will allow the user to steer the assistive robot using gaze to approach objects before performing any manipulatory task on them. The integration will require certain modifications of function modules, including driving module and intention expression module, in order to achieve a seamless transition between them. Accordingly, a new round of evaluation is required for this new gaze-based control system.

After the intention has been expressed to the robot, there are still many open problems that need investigation in order to automatically perform the task for the robot. Task learning is one of them [187, 188], which is learning the object-oriented sequential procedure and reproduce the task step-by-step according to various situations. Note that during intention expression, dominant objects are given particular attention by being visualized earlier. Whether those dominant objects and the viewing order could enlighten the task execution sequence is a question worth future study. If human gaze can provide this information, it can significantly reduce the technical challenge and increase the success rate of task execution.

6.8 Conclusion

In this chapter, a novel, gaze-based implicit intention communication framework was presented in which human intention is inferred by monitoring a user’s gaze. This framework requires very little physical effort from the user. The framework was validated with experiments. In the experiment, two small-scale intention knowledge bases were created, and based on this knowledge, the participants’ intention was recognized correctly from the objects they gazed on. In this study, two knowledge bases were built and different fusion strategies were studied to improve the inference accuracy. This study also revealed the relationship between the mental process and the visual process during the intention-expression procedure. This relationship could help to design new inference algorithms to improve accuracy and help to design more intuitive interaction procedures to reduce user workload. This framework is expected to simplify HRI, consequently enhancing the adoption of assistive technologies and improving users’ independence in daily living. With the purpose of facilitating the daily living of disabled people, testing the presented interaction framework with target users is the

151
goal of this project, and it is the authors' next plan for further validation and evaluation.
7.1 Conclusion

In this dissertation, human gaze is scientifically studied as an intuitive and effective interaction modality between human users and assistive robots. The goal is to enable a user to intuitively express what tasks he/she wants the robot to do by naturally looking at the object of interest in the real world, which has been accomplished with the investigation in this dissertation. The work presented in this dissertation can be summarized from the following concentration areas:

1. developing the gaze vector method for accurate 3D gaze estimation and investigating how to practically implement it (Chapters 2 and 3),

2. investigating the Midas touch problem to distinguish the intentional gaze, which indicates the gaze command, from the gaze data generated during natural visualization (Chapter 5 and 6),

3. interpreting the gaze as a control signal to command a robot for object grasping (Chapter 4), laparoscope holding and orientating (Chapter 5), or complex daily living activities (Chapter 6).

From this investigation, 3D gaze has been successfully converted into an interaction modality and utilized by users to command a robot to accomplish various tasks. Moreover, subjective evaluations demonstrate that the gaze modality is effective and easy to learn and use.

7.2 Future Work

Although promise of using 3D gaze as an intuitive and effective interaction modality has been demonstrated through the investigation in this dissertation, some open problems have
also been revealed. These open problems will be discussed here to expose potential ways in which the gaze-based interaction modality can be further improved.

7.2.1 Multimodal Interaction

In this dissertation, the single interaction modality, gaze modality, is studied to explore its usability and effectiveness. In contrast, multiple modalities are utilized simultaneously during common human-human interaction. Multimodal interfaces are generally preferred over unimodal alternatives due to better flexibility and reliability. Research has shown that multimodal interfaces can speed up information processing and increase understanding precision. Although gaze modality is natural and effective, solving the Midas touch problem often produces additional workload on the user. Combining the gaze modality with other interaction modalities is expected to leverage the workload of the gaze modality. Moreover, it can also improve the command accuracy as gaze is relatively unstable due to visual distractions and unconscious eye movements. However, which modalities are more compatible and how to divide the work among the combined modalities are still open problems.

7.2.2 Validation on Users with Special Needs

One advantage of the gaze modality is its extensive applicability as most users can easily handle it, including those users with severe disability. Moreover, there is an urgent need of an intuitive and effective interaction modality between disabled users and assistive robots. In the experiments, the presented gaze-based interaction systems were validated with able-bodied participants. When these systems are applied to users with disabilities, customization is foreseeable by taking characteristics of the user and assistive task into consideration. It is necessary to validate the present gaze-based interaction systems with this specific user group and investigate the customization strategies that are generally applicable.
7.2.3 Gaze-Enlightened Robot Task Knowledge

Through the gaze modality, a person can command a robot for complex tasks, which can involve multiple stages and complicated operations. Robots autonomously performing tasks has been a challenging problem, which requires specific task knowledge and support of advanced sensing, reasoning, and action capabilities. Although great effort has been put forth, there is still a long way to go due to the highly unstructured environment when performing a task. One key issue in task automation is reasoning the sequential procedure from learned knowledge in order to reproduce the task. However, due to the unstructured environment and flexibility of a task, reasoning the appropriate procedure is challenging. Note that during intention expression, dominant objects of a task are given particular attention by being visualized earlier. Whether those dominant objects and the object viewing order could enlighten the task execution sequence is a question worth further investigation. If human gaze can provide this information, it can significantly reduce the technical challenge and increase the success rate of autonomous task execution.
REFERENCES CITED


Muhammad Imran Shahzad and Saqib Mehmood. Control of articulated robot arm by eye tracking, 2010.


