Human Visual Intelligence-based Multi-Modal Multi-Dimensional Analytics for Metaverse and First-Responder Technology

A dissertation submitted by

Srijith Rajeev

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

In

Electrical Engineering

Tufts University

May 2022

Advisor: Dr. Karen Panetta

Abstract

The metaverse is a massive interoperable network of real-time rendered three-dimensional (3D) virtual worlds that can be experienced synchronously and persistently by an unlimited number of users with an individual sense of presence and data continuity.

Visual intelligence (VI), defined as the 'ability of a machine to see and understand things the same way humans do,' plays an essential role in enhancing a human's experience in the Metaverse. VI is primarily achieved in the metaverse using Artificial Intelligence (AI), including Computer Vision (CV). Despite several advancements, most AI and CV implementations face several challenges in mobile deployment due to their large memory and power requirements. Moreover, the scarcity of real and synthetic datasets further hinders the performance and limits of current AI architectures. Despite several advancements in multi-modal, multi-dimensional sensors, current AI systems are primarily designed for the visible spectrum. Moreover, as the price of spectral sensors is reducing, there is a dire need to develop novel AI architectures that suit them.

This dissertation aims to create novel, robust, and efficient algorithms and architectures specially designed for Visual AI in the metaverse. It further proposes the intersection of Computer Vision, Human Cognition, and Artificial Intelligence for advancing machine VI. By developing Human Visual Intelligence (HVI)-based multi-modal, multi-dimensional methods and systems, this dissertation addresses the following hypothesis: 1) The metaverse will embed human cognition in its visual AI systems. 2) It will become necessary to efficiently and in real-time transform multi-dimensional, multi-modal data into virtualized environments.

To achieve HVI in machines, advanced gaze prediction, analysis, and visualization systems are explored. These AI architectures incorporate human sensory traits such as eye movements and

visual imagery to foster advancements in machine perception for navigation in real and metaverse technologies. To overcome the limitations of visible spectrum-based systems, AI algorithms such as multi-modal semantic segmentation and depth estimation and transformation are developed.

New public databases were created and annotated with benchmark results for training and validation. Novel algorithms developed in this dissertation are validated using quantitative and qualitative analysis. Finally, the scholarly contributions in this dissertation are generalized by applying the developed techniques to numerous fields, including psychology, 3D biometrics, and first-responder technologies.

DEDICATION

To all my loved ones, thank you for motivating me to stay strong and constantly inspiring me throughout my life.

Acknowledgements

Nothing is accomplished alone, and no document tells the full story of the foundations, motivations, and support that made it possible. I am honored and humbled to have received so much unwavering and unconditional support from so many wonderful people during this long journey.

I am deeply grateful to my advisor Dr. Karen Panetta for all the valuable lessons you have taught me, inspiring me to pursue bigger goals, and supporting and believing in my potential to do impactful research. These experiences have taught me many lessons and given me many valuable tools I will carry throughout my career. I want to thank Dr. Sos Agaian for giving me this opportunity and for believing in me. Thank you for always motivating me and providing such treasured guidance. Both Dr. Panetta and Dr. Agaian have made the greatest impact on my academic path, and I am a better researcher today because of you.

I would like to express my sincere gratitude to Dr. Ron Lasser and Dr. James Intriligator for being part of my dissertation committee. Your feedback and guidance have been very valuable and deeply appreciated. I would like to express my sincere gratitude to Dr. Holly Taylor for providing me the opportunity to work on the Center for Applied Brain and Cognitive Sciences (CABCS) projects. These projects showed me the directions to this dissertation.

From the bottom of my heart, I would like to say a big thank you to all the members of the Panetta Lab. Dr. Qianwen Wan (Wendy), Ozan Erat, Dr. Long Bao, Dr. Arash Samani, Dr. Victor Oludare, Dr. Landry Kezebou, Abigail Stone, and Obafemi Jinadu for their knowledge, energy, entertainment, understanding, and help throughout my research.

Rahul Rajendran, Dr. Shishir Rao, and Dr. Shreyas Kamath; It is because of you three that I am here today. We can achieve much more, and I believe the ‘fantastic four’ can do it. A special acknowledgment to Shreyas for all the collaborations we shared.

I would like to specially acknowledge Kim Ellwood, who has been a constant source of support and one of the friendliest faces in the Dean’s office. I would like to thank Michael Minafo, Miriam Santi, and other faculty and staff members of the Electrical and Computer Engineering Department and Computer Science Department for their help during my stay at Tufts. This research would have truly never been possible without the team managing the Tufts High-Performance Computing Clusters. I would especially like to thank Delilah Maloney and Shawn Doughty from the Tufts Research Technology team for your constant technology support.

I would like to thank my mother, father, and brother for always believing in me and supporting me throughout my life. I would also like to thank my sister-in-law, who has been a new source of inspiration. Finally, I would like to thank Tanya and her family for their constant encouragement. Thank you all for your love and support.

Finally, I gratefully acknowledge the sponsorship granted by:

1) The Last Call Foundation, MA under the award 103248-00001. The author would like to thank the Last Call Foundation, the Boston Fire Department, the Medford Fire Department, and the Malden Fire Department for their valuable input and cooperation. The views and conclusions in this document are fthose of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Fire Department or the U.S. Government.

2) the U.S. Army Combat Capabilities Development Command was accomplished under Cooperative Agreement Number W911QY-15-2-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Combat Capabilities Development Command, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon,

3) the National Science Foundation and was accomplished under Award No 1942053. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

4) the US Department of Transportation, Federal Highway Administration (FHWA), under the grant contract: 693JJ320C000023. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the US Department of Transportation or the FHWA.

Table of Contents

List of Tables

List of Figures

List of Publications

Journal papers:

1. K. Panetta, K. M. Shreyas Kamath, S. Rajeev and S. S. Agaian, "FTNet: Feature Transverse Network for Thermal Image Semantic Segmentation," in IEEE Access, vol. 9, pp. 145212-145227, 2021, doi: 10.1109/ACCESS.2021.3123066
2. K. Panetta et al., "ISeeColor: Method for Advanced Visual Analytics of Eye Tracking Data," in IEEE Access, vol. 8, pp. 52278-52287, 2020, doi: 10.1109/ACCESS.2020.2980901
3. K. Panetta, S. Kamath K. M, S. Rajeev and S. S. Agaian, "LQM: Localized Quality Measure for Fingerprint Image Enhancement," in IEEE Access, vol. 7, pp. 104567-104576, 2019, doi: 10.1109/ACCESS.2019.2931980
4. K. Panetta, S. Rajeev, K. M. S. Kamath and S. S. Agaian, "Unrolling Post-Mortem 3D Fingerprints Using Mosaicking Pressure Simulation Technique," in IEEE Access, vol. 7, pp. 88174-88185, 2019, doi: 10.1109/ACCESS.2019.2925605
5. K. Panetta et al., "A Comprehensive Database for Benchmarking Imaging Systems," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 3, pp. 509-520, 1 March 2020, doi: 10.1109/TPAMI.2018.2884458
6. K. Panetta, K. S. Kamath, S. Rajeev, and S. S. Agaian, " No-Reference Fingerprint Image Quality Measure," (Status: Submitted. Awaiting review from the Journal of Surveillance, Security and Safety, 2022)

7. K. Panetta, S. Rajeev, S. S. Agaian, “Tufts I-Drive Dataset: Multi-modal, Multi-dimensional dataset with Eye-tracking data.” (Status: In preparation for the IEEE transactions on pattern analysis and machine intelligence, 2022)

Conference papers:

1. S. Rajeev, S. K. KM, Q. Wan, K. Panetta, and S. S. Agaian, "Illumination invariant NIR face recognition using directional visibility," Electronic Imaging, vol. 2019, no. 11, pp. 273-1-273-7, 2019, doi: 10.2352/ISSN.2470-1173.2019.11.IPAS-273
2. S. Rajeev, A. Samani, K. Panetta and S. Agaian, "3D Navigational Insight using AR Technology," 2019 IEEE International Symposium on Technologies for Homeland Security (HST), 2019, pp. 1-4, doi: 10.1109/HST47167.2019.9033006
3. S. Rajeev, Q. Wan, K. Yau, K. Panetta, and S. Agaian, "Augmented reality-based vision-aid indoor navigation system in GPS denied environment," in Mobile Multimedia/Image Processing, Security, and Applications 2019, 2019, vol. 10993, p. 109930P: International Society for Optics and Photonics, doi: 10.1117/12.2519224
4. S. Rajeev, K. Panetta, and S. S. Agaian, "No-reference color-depth quality measure: CDME," in Mobile Multimedia/Image Processing, Security, and Applications 2018, 2018, vol. 10668, p. 106680L: International Society for Optics and Photonics, doi: 10.1117/12.2304980
5. Q. Wan, S. Rajeev, A. Kaszowska, K. Panetta, H. A. Taylor, and S. Agaian, "Fixation oriented object segmentation using mobile eye tracker," in Mobile Multimedia/Image Processing, Security, and Applications 2018, 2018, vol. 10668, p. 106680D: International Society for Optics and Photonics, doi: 1117/12.2304868

6. S. Rajeev, S. K. KM, K. Panetta, and S. S. Agaian, "Forensic print extraction using 3D technology and its processing," in Mobile Multimedia/Image Processing, Security, and Applications 2017, 2017, vol. 10221, p. 102210L: International Society for Optics and Photonics, doi: 10.1117/12.2262307
7. K. K. M. Shreyas, S. Rajeev, K. Panetta and S. S. Agaian, "Fingerprint authentication using geometric features," 2017 IEEE International Symposium on Technologies for Homeland Security (HST), 2017, pp. 1-7, doi: 10.1109/THS.2017.7943449
8. S. Rajeev, K. K. M. Shreyas, K. Panetta and S. Agaian, "3-D palmprint modeling for biometric verification," 2017 IEEE International Symposium on Technologies for Homeland Security (HST), 2017, pp. 1-6, doi: 10.1109/THS.2017.7943450
9. S. K. KM, S. Rajeev, K. Panetta, and S. S. Agaian, "Comparative study of palm print authentication system using geometric features," in Mobile Multimedia/Image Processing, Security, and Applications 2017, 2017, vol. 10221, p. 102210M: International Society for Optics and Photonics, doi: 10.1117/12.2262309
10. S. Rajeev, K. K. M. Shreyas, A. Stone, K. Panetta and S. Agaian, 'Gaze-FTNet: A feature transverse architecture for predicting gaze attention.' (Status: Manuscript accepted, SPIE 2022)
11. S. K. KM, S. Rajeev, K. Panetta, and S. S. Agaian. 'DTTNet: Depth Transverse Transformer Network for monocular depth estimation.' (Status: Manuscript accepted, SPIE 2022)

Patents:

1. Karen Panetta, Aruna Ramesh, Shishir Rao, Shreyas Kamath, Srijith Rajeev, and Rahul Rajendran, and Sos Agaian, ‘Fusion-Based Sensing Intelligence and Reporting’, PCT application: PCT/2021/064000, filed: Dec 2021
2. Karen Panetta, Shishir Rao, Shreyas Kamath, Rahul Rajendran, Srijith Rajeev, Erin Hennessy, Christina Economos, and Eleanor Tate Shonkoff, and Sos Agaian, ‘Food and Nutrient Estimation, Dietary Assessment, E valuation, Prediction and Management’, PCT application: PCT/US2021/063994 filed, Dec 2021
3. Karen Panetta, Shreyas Kamath, Shishir Rao, Srijith Rajeev, Rahul Rajendran, and Sos Agaian, ‘Deep Perceptual Image Enhancement’, application: PCT/US2021/063999, filed: Dec 2021
4. Karen Panetta, Srijith Rajeev, Rahul Rajendran, Shreyas Kamath, Shishir Rao, Sos Agaian, and Aruna Ramesh. ‘Methods and Apparatus for Dental Disease Detection and Isolation’, (Provisional application filed, Dec 2021)
5. Karen Panetta, Sos Agaian, Shishir Rao, Srijith Rajeev, Rahul Rajendran, Shreyas Kamath, and Jessica White ‘Systems, Methods and Program Products for Detection and Identification of Defects using Artificial Intelligence Analysis of Multi-Dimensional Information Data’, (Provisional application filed, Dec 2021)

Chapter 1: Introduction

1.1 Overview

The term metaverse is a combination of the prefix "meta," implying transcending with the word "universe," which describes a virtual environment linked to a real-world [1]. Reckoned as the future of the internet, it can be defined as a massively-scaled, persistent, interactive, and interoperable real-time platform comprised of interconnected virtual worlds [2, 3]. People can interact, work, transact, and create in the metaverse. In its complete form, the metaverse will consist of a network of decentralized, interconnected virtual worlds with a fully functional economy, allowing users to do virtually anything they can do in the real world [4-7].

While virtual reality and metaverses have existed for decades, they have remained abstract and underwhelming. Before Covid-19, Metaverse was viewed as a platform for community activities such as socializing and gaming. However, the virtual universe has taken on new significance following the pandemic with applications in education, retail industry, and defense [3, 8].

Figure 1.1 depicts the focused areas under the two categories, where the technology supports the metaverse and its ecosystem as a gigantic application [1]. The technology aspect has eight pillars for the metaverse, including CV, AI, and robotics that can work with the user to handle various activities inside the metaverse through user interactivity (manipulating virtual objects) and extended reality (XR) [1].

Visual data and CV are critical for processing, analyzing, and comprehending visuals such as digital images or videos to make meaningful decisions and take appropriate actions [9, 10]. CV enables XR devices to recognize and understand visual information about users' activities and

physical surroundings, enabling the creation of more accurate and reliable virtual and augmented environments [11]. CV algorithms need to tackle several challenges to build more reliable and accurate 3D virtual worlds in the metaverse. The algorithms need to be more precise and computationally effective. Moreover, the metaverse will also require understanding and perceiving the user's environment based on scene understanding techniques.

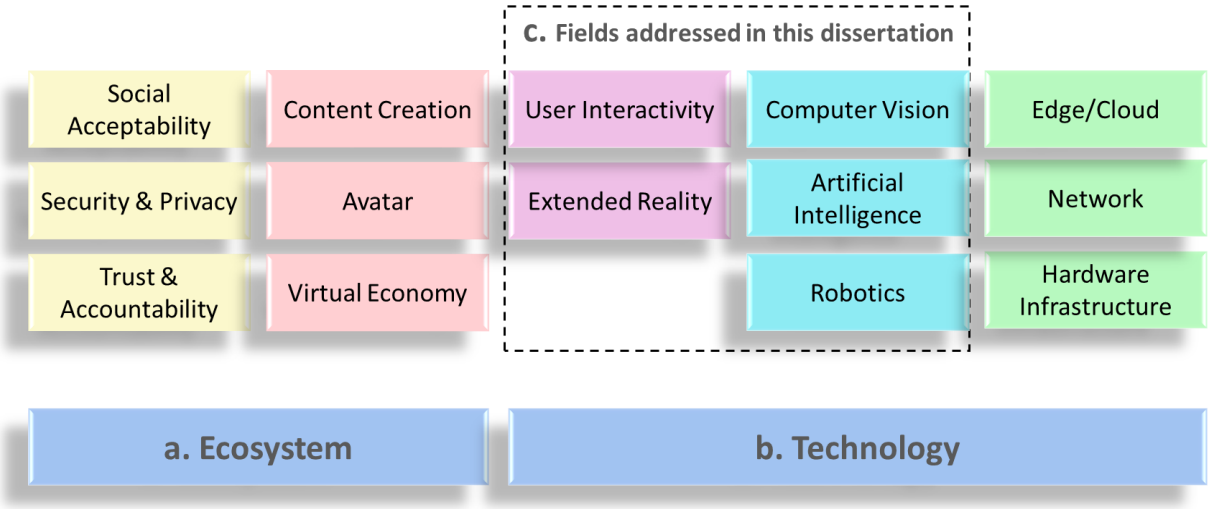


Figure 1.1: The fourteen focused areas, under two key aspects of (a) ecosystem and (b) technology for the metaverse. (c) This dissertation addresses the fields of CV, AI, Robotics (Navigation), User Interactivity, and Extended Reality in Metaverse Technology.

Artificial Intelligence in the Metaverse: Bridging the Virtual and Real Worlds. AI is a technology that attempts to simulate human intelligence by enabling computer applications to learn from experience through iterative processing and algorithmic training [10]. AI systems become smarter with each successful round of data processing, as each interaction enables the system to evaluate and test solutions and develop expertise in the task at hand [12]. This makes AI a compelling and valuable technology. Machine intelligence has far-reaching implications for nearly

every aspect of contemporary life, from financial markets to surveillance to multimedia consumption [13].

AI systems outperform humans significantly in many cases and across various applications, which is why AI technology has become so critical to the modern economy [14]. While virtual reality worlds can exist without AI, their combination provides an entirely new level of realism.

Eye-tracking can facilitate the integration of the real and virtual worlds. However, there are still several challenges in eye tracking in both worlds. First, there is a lack of accurate methods to ensure precise distance estimation. The inclusion of eye-tracking opens up new research directions, such as understanding human behavior accurately in the 3D immersive environment. Early research in AI was influenced by cognitive scientists and inspired by human intelligence [15-20]. AI architectures are often loosely inspired by natural forms of cognition, such as convolutional architectures. In turn, the improvement of these algorithms and architectures enabled more advanced models of human cognition that can replicate and enlighten our understanding of human behavior [21-24]. This advancement inspires the development of new AI systems with improved human-system synergies [25-27].

Several computationally tractable, data-driven methods, such as deep neural networks, have emerged in machine learning to derive useful representations of complex perceptual stimuli [28]. However, they are explicitly optimized to solve certain engineering objectives [29]. This trend leads to a lack of cross-generalization of AI algorithms.

For instance, AI models such as HRNet, PSPNet, Unet, and DeeplabV3 are state-of-the-art in visible spectrum-based object segmentation [30]. However, as shown by Panetta et al. [30], they do not translate to thermal IR image segmentation. Moreover, the current progress has come with

a voracious appetite for computing power which will rapidly become technically and economically prohibitive [30, 31], particularly in mobile-device deployment. Therefore, designing lightweight (low-parameter) and efficient AI systems is essential for cross-generalization [11].

Security in the Metaverse: Addressing the problems of interoperability. With respect to security, face and hand biometrics are likely to influence payment, online identity, and access management ecosystems to change the entire landscape of the digital economy [32]. Because these technologies can identify an individual with great accuracy based on physical and behavioral attributes, they are being used more often for identification and authentication purposes [32]. However, one of the most fundamental challenges in implementing 3D biometric authentication globally is that most existing biometric data was collected in two dimensions. For instance, large-scale identification databases such as AFIS (Automated Fingerprint Identification Systems) and NGI (Next Generation Identification) system [32] are two-dimensional image datasets. This leads to a big challenge in biometric verification known as biometric interoperability. Most biometric systems work assuming that the data being compared were acquired using the same sensor. Hence, they are limited in their capacity to match or compare biometric data acquired using different sensors [33-36]. With the advent of 3D biometrics, it will be crucial to develop 3D to 2D biometric interoperability for 3D biometric verification using existing 2D biometric database.

To address these challenges, this dissertation focuses on developing algorithms in two significant areas of the metaverse: Human Visual Intelligence (HVI)-based multi-modal, multi-dimensional analytics. These solutions aim to create novel, robust, and efficient algorithms and architectures specially designed for AI in the metaverse. Specifically, the rest of the dissertation will strive to prove the following two hypotheses:

1) The metaverse will embed human cognition in its Visual Intelligence systems.

2) It will become necessary to efficiently and in real-time transform multi-dimensional, multi-modal data into virtualized environments.

A systematic block diagram with these aspects is illustrated in Figure 1.4.

Datasets and testbeds are essential for researchers and AI developers to perform training and testing using actual operational data. Most eye-movement data and 3D data are collected in a laboratory setting. AI models are often trained on data that are captured in ideal conditions. Therefore, the algorithms may work well, for instance, in clear daylight conditions when the objects and scenes are visible to the sensor but may perform poorly when the visibility of the scene is impaired by insufficient lighting or bad weather conditions such as rain, snow, haze, and fog [37]. Synthetic data generated by computer simulations or algorithms is a low-cost alternative to real-world data that is increasingly used to create more robust AI models [38]. Thus, new public databases were created and annotated with benchmark results for training and validation [39].

Visual Intelligence systems integrating cognitive science and AI have the potential to improve our understanding of human and machine cognition. These AI architectures incorporate human sensory traits such as eye movements and visual imagery to foster advancements in machine perception for metaverse technologies. This dissertation proposes various object segmentation methods for eye-tracking data to solve the challenges in segmenting the scene objects in video data collected by an eye tracker to support cognition research [40].

Further research led to developing eye-tracking data visualization and analysis systems that allow for automatic recognition of independent objects within the visual field, using deep-learning-based semantic segmentation. User insights from recorded visual stimuli were captured and visualized

using a robust and user-friendly Graphical User Interface (GUI) [41]. However, this progress has come with a voracious appetite for computing power which will rapidly become technically and economically prohibitive [31]. Therefore, a unified end-to-end trainable network is developed to capture discriminative human gaze features from multiple resolutions [30].

In order to apply this augmented navigation, an augmented reality-based indoor navigation system is developed that can dynamically map and navigate multi-level buildings in real-time. Furthermore, an AR system to provide 3-dimensional (3D) navigational insights in unfamiliar environments is developed to promote the learning of complex environments [43] virtually. These systems will use CV to identify and localize a user and AI for path planning, and AR for path visualization [42].

Furthermore, the algorithms developed in this research will be designed for multi-dimensional analytics, particularly 3D and multi-modal (RGB-Color, NIR-Near-Infrared, IR-Thermal) data [39]. These systems can cope with diverse conditions, including nonuniform and poor lighting and noise [40]. To achieve this, a novel depth quality image measure is developed such that it demonstrates a very high correlation with the qualitative evaluation [43]. New deep-learning architectures are investigated for image modalities such as thermal-IR and depth [30]. These methods capture the context of corresponding pixels, particularly edge details in thermal images.

Additionally, novel 3D transformation methods are developed to effectively solve biometric interoperability of authenticating [44, 45]. This system can further be used to transform and manipulate 3D images robustly.

Further applications in first-responder technology. Firefighters, military, police, and paramedics play a critical role in protecting and aiding people at the scene of an emergency, such

as fires, accidents, and natural disasters. These situations are dynamic and require constant monitoring of a multitude of variables. To understand the current fire-fighting technology, several interviews were conducted with fire-fighters from Boston (MA), Medford (MA), Malden (MA), Revere (MA), and Elk Grove (CA). Through these interviews, it was realized that most first responders are unaware of the environment that they are entering. Furthermore, it was discovered that when responding to an emergency, incident commanders must be able to locate and follow the movements of their unit in real-time. Currently, such operations are performed by constant information relays through radios.

Real-time assessment of disasters depends on the human operator's ability to visually identify the subject of interest through natural vision, videos, or images. Further, damaged buildings and roadways, debris, smoke, fire, and environmental conditions such as rain complicate the human observer's ability to detect victims and hazards. It is critical that rescuers can assess hazardous conditions before risking their lives and be armed with intelligent technologies that enable them to respond to dynamic situations. Various aspects of this dissertation would enhance safety and security applications across a wide range of areas, including firefighting, search-and-rescue, and biometric authentication. Topics in this dissertation will also enable better visualization of the disparate data sources. The dissertation's application to first-responder technology can be visualized in Figure 1.2.

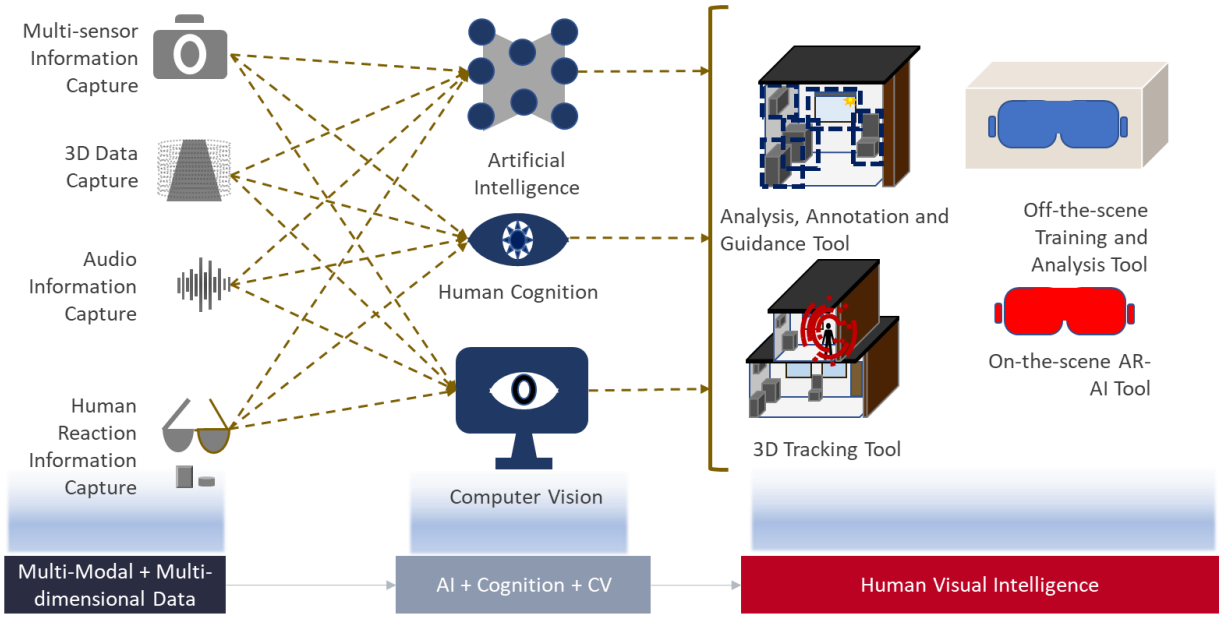


Figure 1.2: An example schematic of an HVI-based multi-modal, multi-dimensional system is proposed in this illustration. One aspect of this dissertation will aim to build a suite of artificial intelligence technologies that can be applied to various fields such as virtual reality, autonomous driving, first-responder technology, and security.

1.2 Dissertation Objectives

This dissertation aims to create novel, robust, and efficient Visual Intelligence systems integrating cognitive science and AI with multi-modal, multi-dimensional features that can power through the limitations of optical reality. This is illustrated in Figure 1.3. These aspects comprise the following tasks:

Task 1: To develop novel AI and CV algorithms that achieve some of the properties of HVI through advanced gaze prediction, analysis, and visualizations.

Task 2: To develop low-parameter, interoperable, and highly accurate algorithms such as dynamic data fusion, multi-modal segmentation, monocular depth estimation, and 3D transformations.

Task 3: To develop comprehensive and versatile multi-modal and multi-dimensional datasets for building more robust, consistent, and adaptable AI.

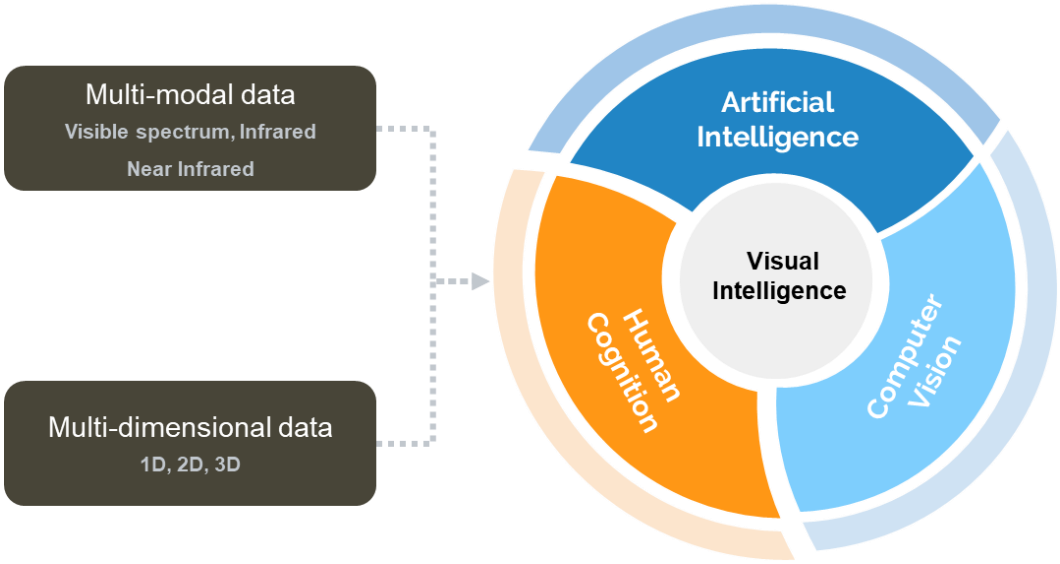


Figure 1.3: Illustrates this dissertation's key concepts, aims, and objectives. CV, AI, and human cognition will form a virtuous cycle of inquiry and discovery across all research activities to develop new architectures in HVI-based multi-modal, multi-dimensional datasets.

1.3 Dissertation Contributions

1. Developed HVI-based systems to play a significant role in the future digital world known as the metaverse.

Several human AI and CV-based eye-tracking algorithms have been explored to address this topic [40, 41]. These techniques are tailored to solve object detection and image segmentation challenges of eye-tracking video data, thus supporting cognitive research, psychology, security, and AI. A novel, stable, and task-driven multi-level fixation-oriented object segmentation method is presented to analyze where and what the observers are looking to clarify the boundary between different objects in the video frame [40]. Furthermore, a data visualization and analysis system that uses deep-learning technology and eye-tracking data to recognize objects of interest from head-mounted eye-tracking video recordings automatically is introduced [41]. A unified end-to-

end trainable network is developed that captures and optimizes gaze feature representation at the multi-scale resolution. Furthermore, an AR-based navigation system that uses CV to identify and localize a user, AI for path planning, and AR for path visualization is developed [46]. Finally, an Extended-Reality-based system that provides 3D navigational insights to users in unfamiliar environments is presented [47].

2. Interpreted and transformed multi-dimensional, multi-modal data from the real world into virtualized environments.

To fill the void in objective depth quality assessment, a novel no-reference-based depth image measure fused with an extended color quality measure is introduced [43]. Furthermore, a unified end-to-end trainable network with representations shared between the semantic segmentation and edge guidance structures is developed to capture discriminative thermal image features from multiple resolutions [30]. Furthermore, a novel network comprising an encoder network, a corresponding transverse decoder network, and a transformer layer to perform continuous depth estimation at the pixel level is introduced. To address the deficiency of 3D to 2D image mapping algorithms, novel 3D unrolling and pressure simulation methods are presented. The thrust of this work strives to develop a correspondence between 3D and 2D images [44]. The method's robustness was tested by applying the algorithm to biometric 3-D modalities such as fingerprints, palmprints, and lip-prints [45, 48, 49].

3. Developed a suite of revolutionary lifesaving AI-based technologies for search-and-rescue applications.

Using HVI-based multi-modal, multi-dimensional analytics, a 3-dimensional/2-Dimensional person localization, positioning, and navigation system was developed [42]. Further, AI networks

such as FTNet and DTTNet are used for dynamic classification, segmentation, gaze prediction, and 3D mapping [30]. The proposed framework will also enable better visualization of the disparate data sources, thus making it conducive for the human operators to detect hazards and victims. The approach of seamlessly utilizing and visualizing sensor information in a situational map to guide first-responders actions and enhance their efficiency would reduce the risk factor for first-responders.

4. Developed large datasets and testbeds so that researchers can perform training and testing on operational real and synthetic data.

Datasets and testbeds are critical for researchers and AI developers to conduct training and testing on real-world operational data. The majority of current AI systems are specialized in a single type of data: "Optical RGB Data." AI models are trained using data collected under ideal conditions. Thus, while the algorithms may perform well in bright daylight when objects and scenes are visible to the sensor, they may perform poorly when the scene's visibility is compromised by insufficient lighting or adverse weather conditions such as rain, snow, haze, or fog. Similarly, most eye-movement data are collected in a laboratory setting. AI models are trained on data that are captured in ideal conditions. To the best of our knowledge, the new Tufts I-Drive (TID) Dataset that includes visible spectrum and thermal data and eye-tracking data will make the first-of-its-kind to include multi-modal data with eye-tracking. Along with TID, a comprehensive multi-modal face database [39] and a versatile synthetic 3-D multi-modal database were also created [45].

1.4 Outline of the Dissertation

This dissertation comprises nine chapters, including the introduction and conclusion. The remaining chapters were extensively and methodically researched and crafted. A systematic block diagram with chapter numbers is illustrated in Figure 1.4.

Chapter 2 presents various object segmentation methods for eye-tracking data to support cognition research. First, a multi-level fixation-oriented object segmentation method is presented to solve the above challenges in segmenting the scene objects in video data. Then, an eye-tracking data visualization and analysis system is introduced for automatic recognition of independent objects within field-of-vision, using deep-learning-based semantic segmentation.

Chapter 3 develops an augmented reality-based system to provide 3-dimensional navigational insights in unfamiliar terrestrial environments. This is accomplished by first generating and traversing a 3D terrestrial map using path-planning navigation algorithms. This chapter further presents two novel indoor navigation systems (INS) for dynamic navigation and localization. The first approach (INS-1) develops and demonstrates an augmented-reality-based vision-aided indoor navigation system, which localizes the user without a GPS signal. The second approach (INS-2) dynamically maps and localizes one's location in multi-level buildings using LiDAR technology.

Chapter 4 introduces a novel end-to-end trainable convolutional neural network architecture that captures and optimizes feature representation at the multi-scale resolution, improving the capability to process high-resolution images and produce quality output with a lower computational cost. This network is applied to thermal infrared image segmentation and human gaze estimation.

Chapter 5 introduces the end-to-end trainable convolutional neural network architecture with vision transformers to extract local and global multi-resolution representation to perform pixel-wise depth estimation more accurately using lower parameters.

Chapter 6 presents a novel no-reference-based depth image measure and further fuses this measure with an extended color quality metric. The Color-Depth image quality measure CDME has no constraint on the 3D images being compared. Experimental results demonstrate that CDME has a very high correlation with human visual observations.

Chapter 7 introduces a suite of 3D processing algorithms, including a novel shape-independent unrolling algorithm, pressure simulation, and mosaicking algorithm. These tools are then applied to several biometric modalities.

Chapter 8 introduces three comprehensive multi-modal, multi-dimensional datasets for face and emotion recognition research, autonomous driving, and interoperable biometric security.

Chapter 9 summarizes the dissertation and discusses the future research directions.

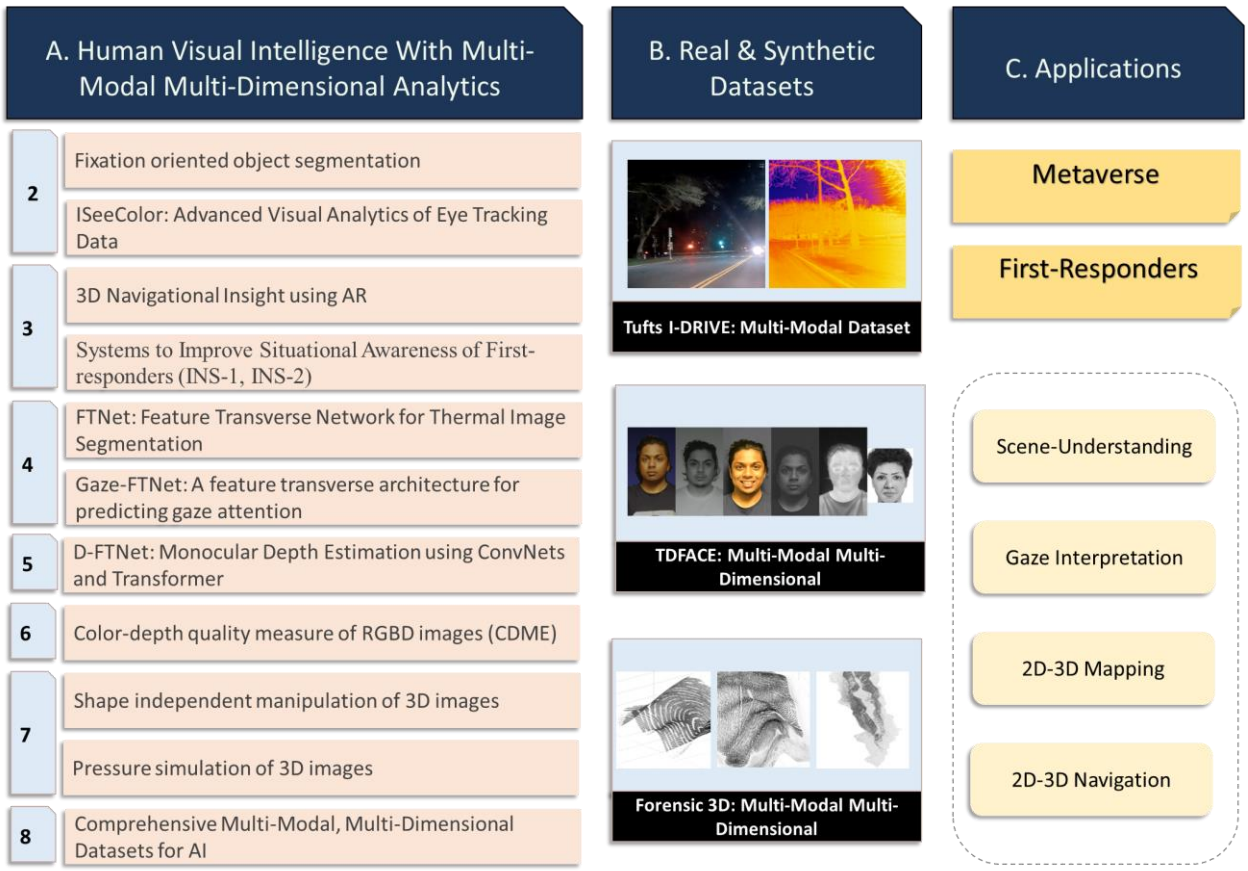


Figure 1.4: A detailed overview of the dissertation structures. (A) shows the various technological developments that are covered in this dissertation. (B) refers to the several multi-modal, multi-dimensional real and synthetic datasets developed. (C) refers to the main applications and their target technologies.

Note: Numbers (2-8) correspond to this dissertation's chapter numbers.

Chapter 2: Eye-tracking Analysis and Visualization

Abstract

Visual Intelligence systems integrating cognitive science and AI offer a deep understanding of human cognition and machine intelligence. These AI architectures incorporate human sensory traits such as eye movements and visual imagery to foster advancements in machine perception for autonomous driving and metaverse technologies. This chapter proposes two segmentation approaches for eye-tracking data to solve the challenges in segmenting the scene objects in video data collected by an eye tracker to support cognition research. Further, a data visualization and analysis system, 'ISeeColor,' that combines deep learning technology and eye-tracking data, is developed.

Index terms: eye-trackers, cognitive science, data visualization, data analysis, deep-learning.

Publications

Wan, Qianwen, Srijith Rajeev, Aleksandra Kaszowska, Karen Panetta, Holly A. Taylor, and Sos Agaian. "Fixation oriented object segmentation using mobile eye tracker." In Mobile Multimedia/Image Processing, Security, and Applications 2018, vol. 10668, p. 106680D. International Society for Optics and Photonics, 2018.

Panetta, Karen, Qianwen Wan, Srijith Rajeev, Aleksandra Kaszowska, Aaron L. Gardony, Kevin Naranjo, Holly A. Taylor, and Sos Agaian. "Iseecolor: method for advanced visual analytics of eye tracking data." IEEE Access 8 (2020): 52278-52287.

Eye-tracking technology enables researchers to monitor the eye's position and infer the direction of the gaze, which is used to better understand the nature of human attention in fields such as psychology, cognitive science, marketing, and artificial intelligence. Researchers can use commercially available head-mounted eye trackers to track pupil movements (saccades and fixations) using an infrared camera and a front-facing scene camera to capture the field of vision. While the wearable eye tracker enabled research in unrestricted environment settings, the recorded scene video typically contains uneven illumination, low-quality image frames, and moving objects.

Wearable eye trackers are small and unobtrusive, allowing cognitive psychologists to monitor observers' gaze information in naturalistic experimental paradigms [50] [51, 52]. Head-mounted eye trackers are small and unobtrusive, allowing for recording eye movements in more naturalistic experimental settings without restricting movement [50-52]. By tracking the distance between the pupil's center and the position of artificially illuminated corneal reflections, high-frequency infrared eye cameras continuously monitor and record pupil position [40, 53]. Offline analysis of this data reveals discrete eye movement events such as fixation, saccades, smooth pursuit, and scanpath. This enables researchers to conduct valid experiments that examine where and what people look at [53].

Head-mounted eye-tracking devices are used in a number of applications, including education [54], cognitive psychology [55], navigation [56], medicine [57, 58], information visualization [46, 59, 60], and assistive technology [46, 61]. However, eye-tracking analytics is made significantly more difficult by noise introduced by unpredictable weather conditions, uncontrolled lighting, and dynamic scenes with motion blur.

Finally, researchers can extract eye movement data to determine where participants looked, what they looked at (classification), and for how long they looked (duration) [62, 63]. Data visualization

and analytic approaches have been developed to understand better the nature of eye movements [64-66]. However, visualizing and analyzing eye-tracking data collected during real-world locomotion presents a unique analytical challenge for automatically determining where and what a person is looking at in a particular scene [67-70].

Recent advancements in head-mounted eye-tracking technology have enabled researchers to monitor eye movements during locomotion in natural settings, enhancing the ecological validity of research on human gaze behavior. While it is becoming easier to collect eye-tracking data, visual analytics remains challenging and time-consuming. One of the most critical tasks in analyzing video data from recorded scenes is determining the boundary between multiple objects in a single frame. A critical need exists to develop efficient visualization and analysis tools for large-scale eye-tracking data. Segmentation of objects is essential and critical as a starting point for all subsequent eye-tracking data analytic system steps.

Most eye-tracking analysis systems rely on laborious and time-consuming manual annotation systems. Few eye-tracking analysis systems can automatically categorize what people looked at without manually tagging the video's areas of interest (AOI). However, no automated visual analytic tools can present all three aspects of a dynamic scene video (Q1: location, Q2: classification, and Q3: duration). *Q1: Location* can be defined as the position of the recorded raw gaze points in the scene video frame. *Q2: Classification* can be defined as the semantic interpretation from the region-of-interest to object category within the visual field. *Q3: duration* can be defined as the duration of eye movement events falling within a specific objects-of-interest within FOV.

Table 2.1 describes the existing eye-tracking methodologies along with their descriptions and features. To the best of the author’s knowledge, there are currently no solutions to automate the

visualization of massive head-mounted eye-tracking data that combines the Q1: ‘location,’ Q2: ‘classification,’ and Q3: ‘duration’ information types in a single visualization and analysis tool.

Chapter 2 introduces two object segmentation methods for eye-tracking data. First, a multi-level fixation-oriented object segmentation method is presented to solve the above challenges in segmenting the scene objects in video data. Then, an eye-tracking data visualization and analysis system is introduced for automatic recognition of independent objects within field-of-vision, using deep-learning-based semantic segmentation.

The multi-level fixation-oriented segmentation method (M-FOS) is proposed to overcome the challenges in segmenting scene objects in video data collected via an eye tracker to aid cognition research. M-FOS demonstrates its advancements in position invariance, illumination, and noise tolerance while remaining task-driven. The proposed method is validated using real-world case studies created by our team of psychologists specializing in visual attention and human problem-solving. The extensive computer simulation validates the method's accuracy and robustness for the segmentation of fixation-oriented objects. Additionally, a deep-learning image semantic segmentation technique was explored using M-FOS results as label data to demonstrate the feasibility of online deployment of fixation-oriented object segmentation using an eye tracker.

This work develops a first-of-its-kind eye-tracking data visualization and analysis system that enables the automatic recognition of independent objects within the field of vision via semantic segmentation based on deep learning. This system recolors fixated objects of interest by integrating gaze fixation information with semantic maps. The system enables researchers to infer what objects users view and how long they view them in dynamic contexts. The architecture is evaluated

outdoors using a case study of users walking around the Tufts University campus as a navigation study conducted by research psychologists.

The contributions of the systems and methods developed in this chapter are as follows:

1) a method for multi-level fixation-oriented object segmentation;

2) a data visualization and analysis system that combines deep learning technology and eye-tracking data to recognize objects of interest from head-mounted eye-tracking video recordings automatically; and

3) a graphical user interface that displays annotations for objects of interest alongside eye-tracking data information.

This chapter is organized as follows. In section 2.1, background information on eye-tracking technology, including the hardware setup, pupil detection, calibration, and discussion about the challenges of analyzing the collected data. The proposed multi-level fixation-oriented object segmentation method is described in section 0. Moreover, to demonstrate the features of the proposed system architecture results from real-life eye tracker experimental data are shown and compared in Section 2.2. Section 2.3 shows the proposed architecture of ISeeColor. Section 2.4 presents the evaluation and comparison of the proposed ISeeColor. Lastly, section 2.5 summarizes the contributions and discusses the future directions of this work.

2.1 Related Work

Table 2.1: Related work on eye-tracking data visualization and analysis approaches, Q1 refers to ‘location, Q2 refers to ‘classification,’ and Q3 refers to ‘duration.’

Related Work	Description	Feature
Statistical metrics [71]	Display eye movement information in a numeral format, such as fixation counts for each area of interest, the total number of fixations, fixation duration, fixation duration, time to the first fixation on target, fixation density, repeat fixations, saccade/fixation ratio, saccade amplitudes, and saccadic latency.	Q1 Q3
Attention heatmap [62, 72-76]	Visual attention can also be visualized as a heat map, indicating individual fixation points and entire groups' visual attention.	Q1 Q3
Scanpath visualization [77-82]	The term "scanpath" refers to any pattern of saccades and fixations. A perfect scanpath is defined by a brief time interval and a straight-line saccade to the desired target.	Q1 Q3
Area-of-interest (AOI) visualization techniques [83, 84]	These techniques couple eye-tracking data to the visual stimulus. AOI has to consider the annotation of objects on the stimulus that is of particular interest to the researcher.	Q1 Q2
Combination of data-of-interest (DOI), AOI, and eye-movements metrics [57, 64, 65, 67, 75, 85-88]	These techniques combine several state-of-the-art eye-tracking visualizations to create diverse visualization and analysis systems that provide data points, eye movements, scanpaths, or heat maps.	*(Q1 Q2 Q3)

Eye-tracking technology enables us to deduce which aspect of a visual scene a person focuses on or where their gaze is being directed. Gaze deployment, or the direction in which one's vision is directed, is frequently used interchangeably with overt visual attention. Attention can be defined as the internal allocation of processing resources, with eye movements serving as a marker for the path of attention through a scene [89]. Eye movement and attention correlations investigate how individuals perform tasks such as reading a book, driving a car, or exploring a scene. These tasks necessitate serial processing of various scene fragments based on eye movements [90].

Eye movements can be classified into four broad categories. They are as follows: (1) Fixations: eye movements that maintain the retina's position relative to stationary objects of interest (OOI), (2) Saccades are rapid eye movements that occur in the intervals between fixations. (3) Pursuit movements: occur when a person's visual attention is drawn to a moving target; and (4) Scanpath: a series of saccades and fixations that constitute viewing.

2.1.1 Eye Tracking

Commercially available infrared eye-tracking equipment can be classified as either stationary remote eye trackers or head-mounted (or otherwise wearable) mobile eye trackers. Because remote eye trackers are stationary, their accuracy is contingent upon the system's ability to precisely locate the eyes within a given headbox (a space within which a participant can move without their eyes falling out of the tracking range). This technical constraint significantly limits participants' range of motion and frequently necessitates using chinrests to ensure data quality. Since the 1980s, remote eye trackers have garnered considerable research attention and have been extensively used in indoor human-computer interaction experiments [91, 92].

Remote eye trackers are most frequently used in computer-based tasks, effectively excluding studies where participants move their head or body, manipulate objects, or interact with other people. On the other hand, head-mounted mobile eye-tracking technology enables the tracking of people's gaze without impairing their movement, thus opening up possibilities that remote eye tracking could never approach.

The development of portable eye trackers has broadened the research possibilities to include natural tasks. In [93] [94] [95], a geology field trip and a grocery store experiment were described to illustrate possible indoor and outdoor research paradigms utilizing a head-mounted mobile eye tracker. Wearable eye trackers have been used in on-road driving applications [56] to monitor the

eye movements of a driver navigating a test route while performing various driving tasks; a head-mounted eye tracker was proposed for use in establishing evidence for predictive control of eye movements in sports [96] using two skilled squash players. Additionally, mental health monitoring through a mobile eye-tracker has been proposed and is being investigated [58].

Numerous pioneering studies have established the value of lightweight mobile eye trackers in psychology, cognition, usability, and marketing. While deploying head-mounted eye-tracking technology in academic and commercial research has been a significant success, analyzing the collected data remains a challenge. Visual analytic methods, such as manual annotation [64-66], gaze point labeling systems [67-69], and other image processing and computer vision techniques [70], have been developed to provide a more efficient and accurate way to understand the recorded eye-tracking data. The Tufts Research Psychology team used SMI Eye Tracking Glasses to conduct real-world experiments examining human problem solving and spatial learning abilities. [97].

2.1.2 Eye Tracker Hardware

An eye-tracking system typically consists of cameras, light sources, and computing capabilities. With machine learning and advanced image processing, algorithms convert the camera feed to data points. In this study, the eye-tracking glasses had a frontal scene camera capturing the field of view in front of the participant and infrared cameras directed at participants' eyes to track the distance between the center of the pupil and the position of artificially illuminated corneal reflections.

2.1.3 Eye-Tracking Data Visualization and Analysis

Generally, eye-tracking data is visualized using visual overlays such as heatmaps and scanpaths. Higher color intensity indicates a longer fixation duration in heatmaps. Scanpaths represent the

sequence of fixations and saccades as a path traversed across a scene. Additionally, when visualizing eye-tracking data, researchers must define areas of interest (AOIs), which are objects within the visual scene that are of particular interest for analysis. The following sections review the literature on eye-tracking data visualization approaches and trends in developing data visualization and analysis systems.

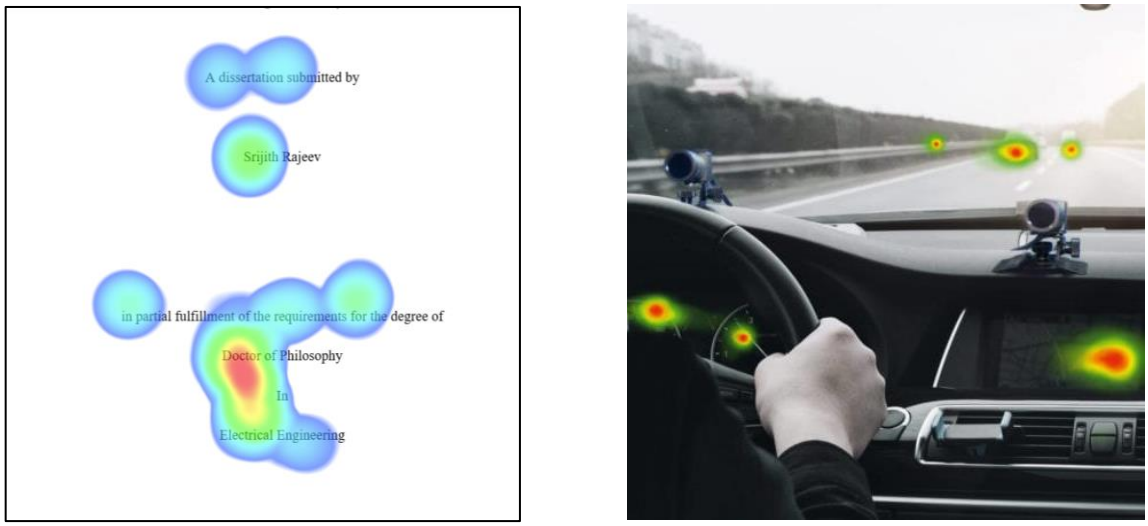


Figure 2.1. Visualizations of attention maps depict the general distribution of gaze points. Attention maps are generally displayed as a color gradient overlay on the image or stimulus that is being presented. The colors: red, yellow, and green indicate the number of gaze points directed at various parts of the image in descending order [98].

Most eye movement metrics available to researchers necessitate that data visualization techniques evolve in unison with advancements in research. Using standard software, eye movement metrics [71] such as fixation duration count, saccade/fixation ratio, and saccade amplitudes can be easily computed from raw eye-tracking data. Scanpath visualization takes fixations and saccades into account when creating a "viewing path" for a person viewing a scene. Each fixation is represented by a circle with a radius equal to the fixation duration in a typical scanpath visualization. Saccades between fixations are depicted by connecting these circles with lines, as shown in Figure 2.2 [77].

Visualization techniques based on AOI (area-of-interest) also consider the annotation of objects on the stimulus that is of particular interest to the researcher.

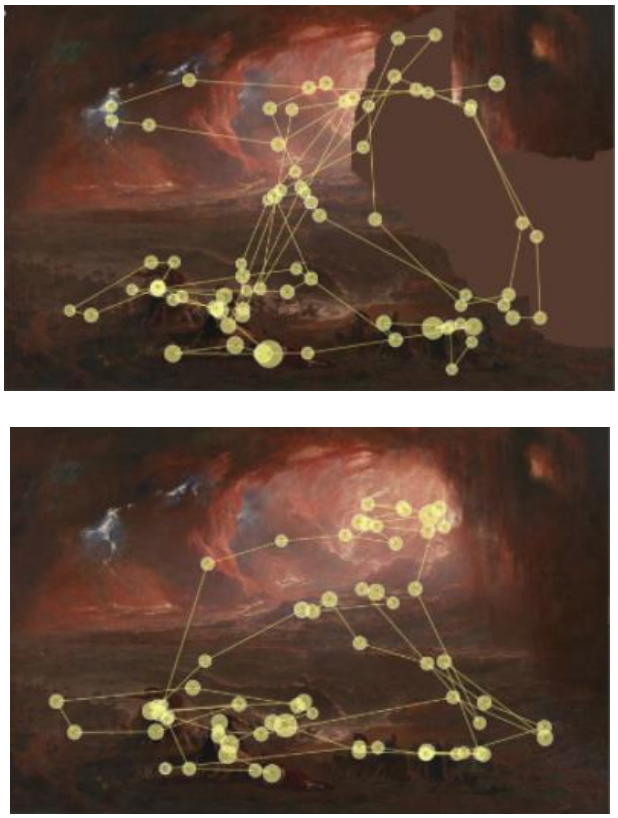


Figure 2.2: Eye-tracking scanpath for a participant viewing the painting's Neutral Infill restoration (left) and complete restoration (right) [99]. Fixations are denoted by dots, while lines denote saccades. Eye-tracking identifies which parts of the painting attract attention and how attention is shifted.

When conventional data visualization techniques, such as those previously discussed, are no longer capable of analyzing massive eye-tracking datasets, the emerging discipline of data visualization and analysis can assist in exploratory large-scale eye-tracking data analysis. ISeeCube [65] visualizes eye-tracking data using space-time cube visualization [75] (STC). Kurzhals et al. [67] developed a timeline visualization to illustrate the AOI-based scanpaths of various viewers using manually annotated AOIs. Kurzhals et al. [67] further described a method for automatically clustering eye-tracking data and integrating it into an interactive labeling and

analysis system for AOI annotation. GazeDx [57] was proposed as an interactive visual analytics tool for comparing gaze data and spatial views of radiologists' volumetric medical images. A visual analytics approach [85] was also proposed, incorporating multiple concurrent evaluation procedures, such as "thinking aloud" protocols, interaction logs, and eye-tracking. A novel technique to visualize spatial AOIs was developed by combining clustering and merging clusters that change over time[86]. Furthermore, recent eye-tracking analysis trends have led to the development of data-of-interest (DOI) based techniques [87, 88].

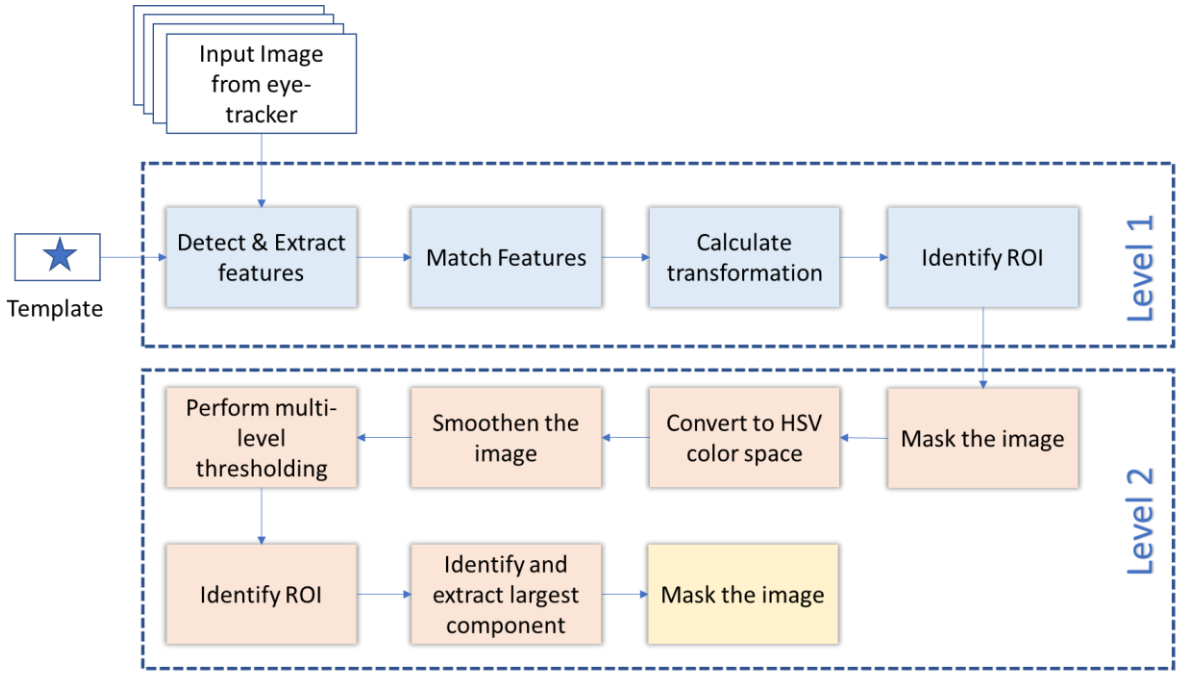


Figure 2.3: The overall system architecture for the proposed multi-level fixation-oriented segmentation M-FOS. This system requires input from an eye-tracker and a template for object matching. This system will perform well when repetitive tasks are necessary at low overhead.

2.2 Multi-level fixation oriented segmentation (M-FOS)

The first step toward developing a novel eye-tracking data visualization and analysis system that enables the automatic recognition of independent objects within the field of vision is semantic

segmentation. This section will show the development of a computer vision-based region-of-interest (ROI) segmentation based on eye-tracking data. The proposed multi-level fixation-oriented segmentation algorithm is divided into object detection and object segmentation. Figure 2.3 illustrates the overall system architecture. The following sections provide a detailed description of the proposed multi-level fixation-oriented object segmentation.

2.2.1 Object Detection

This step aims to extract the objects of interest and the orange fixation indicator. The template images and testing images depicted in Figure 2.4 serve as an example of the input data. The test images may or may not include the lego kits, an orange-colored fixation indicator, and other supplemental information.

The extraction of features is critical because the features available for discrimination directly affect the segmentation algorithm's effectiveness. The extraction task produces a set of features (feature vector), forming an image representation. A good technique for detecting features should be robust to image transformations such as rotation, scale, illumination, noise, and affine transformations. Additionally, ideal features must be highly distinctive, with a high probability of being correctly matched by a single feature. SURF, SIFT, Harris-Laplace, Gaussian Difference (DOG), and Hessian-Laplace are just a few of the detectors that can be used [100, 101]. This chapter makes demonstrative use of SURF features.

Speeded Up Robust Features (SURF): The SURF method is a fast and robust algorithm for representing and comparing images in a local, similarity-invariant manner [102]. Interest points are defined as salient features from a scale-invariant representation of an image. Convolution of the initial image with discrete kernels at multiple scales enables this multiple-scale analysis (box

filters). A Fast Hessian detector is used to detect SURF key points. Given an Image I with a point X=(x,y), its Hessian matrix at scale σ is given as

$$\mathrm{H}(\mathrm{x},\sigma) = \begin{bmatrix} L_{xx}(\mathrm{x},\sigma) & L_{xy}(\mathrm{x},\sigma) \\ L_{yx}(\mathrm{x},\sigma) & L_{yy}(\mathrm{x},\sigma) \end{bmatrix} \quad 2.1$$

Where $L_{xx}(\mathrm{x},\sigma)$ represents the convolution of the Gaussian second-order derivative of an image I(x,y) at point x. A box filter approximates Gaussian partial derivatives of second-order and integral images to accelerate convolution, significantly reducing the computational burden [102].

$$\det\left(H_{approx}\right) = D_{xx}D_{yy}\left(0.9D_{xy}\right)^2 \quad 2.2$$

Where $D_{xx},\ D_{yy}, and\ D_{xy}$ are the convolutions of each box filter and the image. By expanding the size of the box filter on the image, an image pyramid with a different scale is constructed. After calculating the approximated determinant of the Hessian matrix at each scale, the maxima are found using non-maximum suppression in a 3x3x3 neighborhood. Interpolation in both scale and image space can determine the stable location of the SURF key point and the value of the located scale [102]. SURF features can be visualized in Figure 2.5.

A coarse matching process is carried out using the nearest neighbor principle. Consider images I and J, which contain key points A1 and A2, respectively. Positively matched key points are those where the distance between them is less than the distance between any other key point in image J. This concept matches the key points identified and described [100]. Figure 2.6 illustrates the matching process.

RANSAC: RANSAC is an iterative outlier detection technique used to estimate the parameters of a mathematical model from a set of observed data that contains outliers. This step estimates the geometric transform from matched point pairs and calculates the transformation associated with

the matched points while removing outliers. The RANSAC transformation is then applied to the image to locate the object [103]. The region of interest and mask are identified from the object's location, as shown in Figure 2.7.

Figure 2.4: The example template images collected from indoor cognitive research.

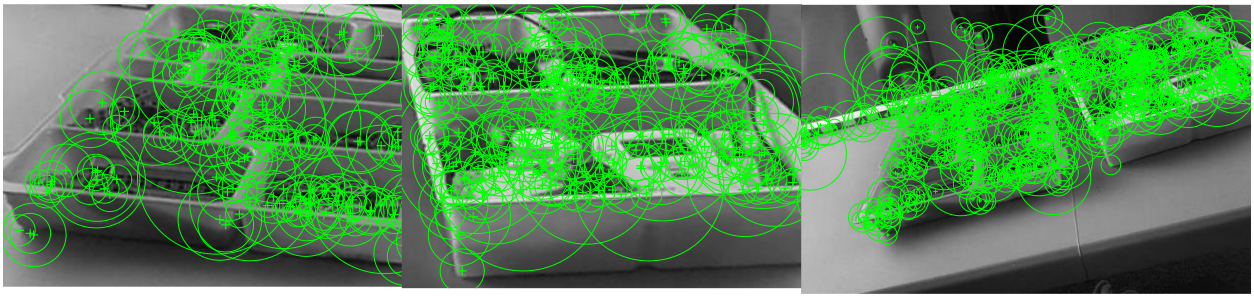


Figure 2.5: The example results of SURF feature extraction of template images and testing images.

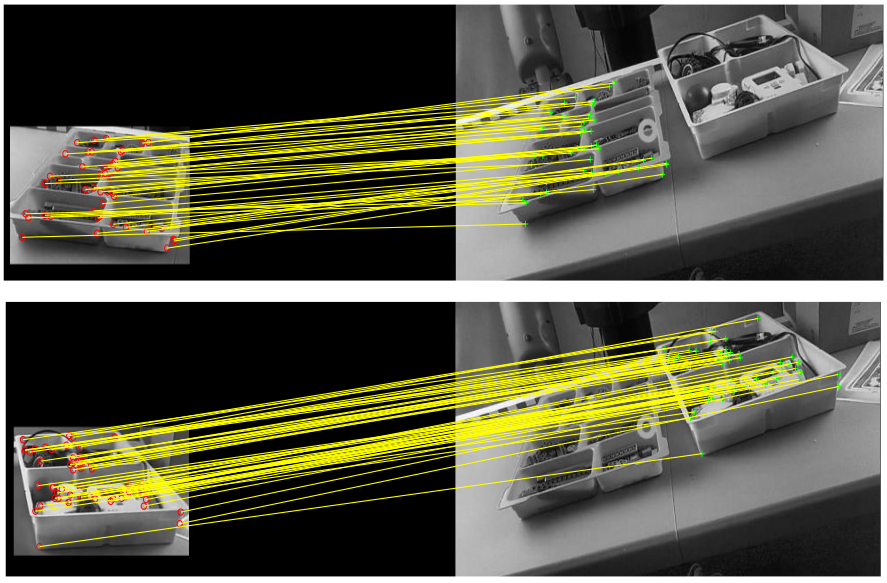


Figure 2.6: A demonstration of matching the detected featured in template images and testing images.

Figure 2.7: Shows the identified region of interest.

2.2.2 Object Segmentation

First, the masked image is converted from RGB to HSV color space. Traditional color space models such as RGB do not separate the image intensity from the color information. HSV separates *luma*, or the image intensity, from *chroma* or the color information. Therefore, the images are converted to the Hue-Saturation-Value (HSV) color space to separate color components irrespective of the intensity, thus, becoming robust to illumination and exposure variations.

Next, the resulting images are passed through a low-pass filter to reduce the noise within the masked image to produce a less pixelated image. Often images can be noisy – irrespective of the sensor, it always adds an amount of noise to the image. The statistical nature of light itself also contributes noise to the image. Thus, the low-pass filter has a more significant effect on the noise than on the image. By reducing the noise, gradual, previously invisible changes can be seen [104]. The Gaussian lowpass filter used in this approach is defined by,

$$G(u,v) = H(u,v)F(u,v) \qquad 2.3$$

where F(u,v) is the Fourier transform of the image being filtered and H(u,v) is the Gaussian lowpass filter transform function. u = 0, 1, 2…M-1 and v = 0, 1, 2…N-1

$$H(u,v) = e^{\frac{-D^2(u,v)}{2D_0^2}} \quad 2.4$$

Where D(u,v) is the distance from the origin of the Fourier Transform given as:

$$D(u,v) = [(u - M/2)^2 + (v - N/2)^2]^{1/2} \quad 2.5$$

Where $1 \leq u \leq N$, $1 \leq v \leq M$, D_0 is the cutoff frequency.

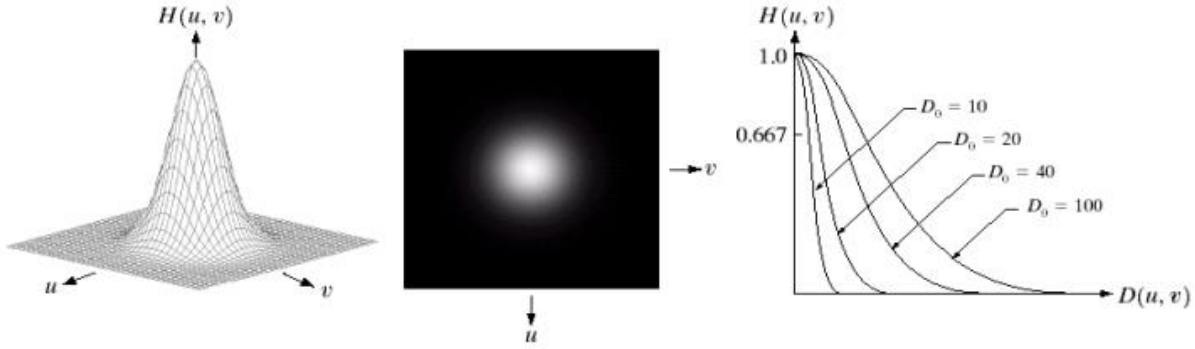


Figure 2.8: Visual illustration of the Gaussian lowpass filter transfer function [105].

Thresholding is an essential technique for image segmentation that attempts to identify and extract a target from its background. This chapter utilizes Otsu's multi-level threshold technique to perform this step [106, 107].

Consider a grayscale image L with levels (1,2,3…L). Let C_0 and C_1 be the two classes of the threshold level t, where C_0 ranges from (0, 1 … t-1) and C_1 ranges from (t, t+1 .. L-1). Let p be the probability of gray level L. They are defined as,

$$C_0 = \frac{p_0}{\sum_{i=0}^{t-1} p_i} \cdot \ldots \frac{p_{t-1}}{\sum_{i=0}^{t-1} p_i} \quad and \quad C_1 = \frac{p_t}{\sum_{i=t}^{L-1} p_i} \cdot \ldots \frac{p_{L-1}}{\sum_{i=t}^{L-1} p_i} \quad 2.6$$

$$\text{Mean levels of } C_0 \text{ and } C_1\text{: } \mu_0 = \sum_{i=0}^{t-1} \frac{ip_i}{\sum_{i=0}^{t-1} p_i} \text{ and } \mu_1 = \sum_{i=t}^{L-1} \frac{ip_i}{\sum_{i=t}^{L-1} p_i} \quad 2.7$$

$$\omega_0 = \sum_{i=0}^{t-1} p_i \text{ and } \omega_1 = \sum_{i=t}^{L-1} p_i \quad 2.8$$

$$\omega_0 + \omega_1 = 1 \quad 2.9$$

The mean intensity (μ_t) of the entire image is further given by

$$\mu_T = \omega_0\mu_0 + \omega_1\mu_1 \quad 2.10$$

Let $\sigma_0$and $\sigma_1$be the variances of C_0 and C_1, respectively. Then, Otsu's between-class is given as,

$$J(t)_{max} = \sigma_0 + \sigma_1 \quad 2.11$$

Finally, multi-level thresholding is calculated using the following method. Let $t_1, t_2, .. t_m$ be the m threshold levels. Then the objective function for thresholding is given by

$$\sigma_0 = \omega_0(\mu_0 - \mu_t)^2, \ldots \sigma_m = \omega_m(\mu_m - \mu_t)^2 \quad 2.12$$

$$J(t)_{max} = \sigma_0 + \sigma_1 + \cdots \sigma_m \quad 2.13$$

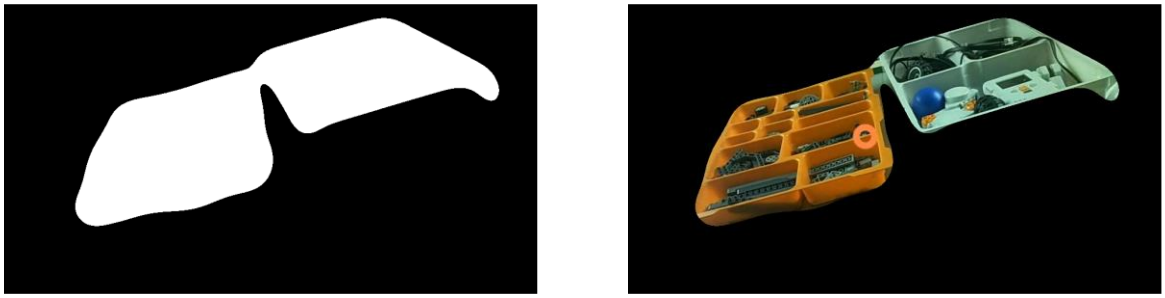


Figure 2.9: Visualized the results of the proposed M-FOS ROI identification. The left image shows the generated mask, and the right image results from applying the mask to the query image.

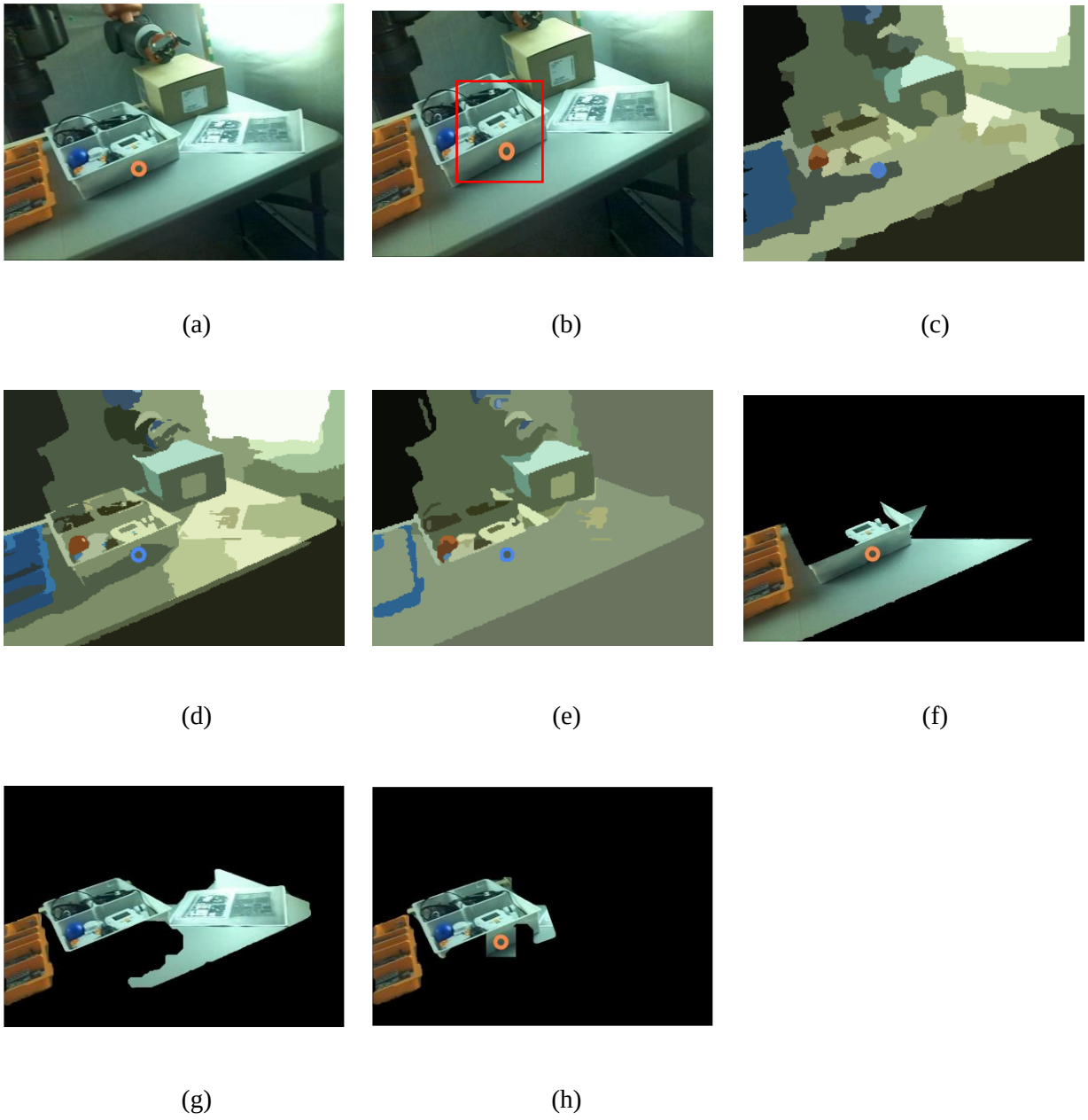
Using the multi-thresholds obtained in the previous step, the region of interest is identified by creating a mask. In image processing, connected components are frequently used to detect connected regions in digital images. As a result, the ROI image's connected components are

extracted. Then, the largest blob is identified and extracted from the resulting binary image. Finally, this component masks the image, resulting in the segmented result shown in Figure 2.9.

2.2.3 Experimental Results and Discussion

The proposed M-FOS was extensively tested using eye-tracking data collected during indoor experiments designed by a team of psychologists at Tufts University. The total duration of the video is approximately 50 hours, captured at a frame rate of 30 frames per second. Typical characteristics of recorded scene video include uneven illumination, low-quality image frames, and moving scene objects. Comparative results are included to demonstrate the performance of M-FOS in comparison to bounding box-cropping segmentation, SLIC-Graph cut normalized [52, 108-110], Quickshift [52, 111], Felzenszwalb [109], Marker-Controlled Watershed Segmentation [52, 112], and Level2 segmentation without prior object detection. The query image shown in Figure 2.10 (a) is applied to various image segmentation algorithms for demonstration purposes. Several constraints include the fact that the objects do not have a fixed size due to the participant's constant motion; this results in a zoom-in and zoom-out effect.

The output for the fixed size bounding box cropping segmentation (shown in Figure 2.10 (b)) identifies the orange dot indicator. However, it does not highlight the object the participant is fixating on due to its fixed size constraint. The outputs of SLIC, Quickshift, and Felzenszwalb are shown in Figure 2.10 [c-f]. The output of these algorithms are not suitable for segmenting individual objects and cannot distinguish objects from the redundant background information. Figure 2.10 (g) shows the output of the Level2 segmentation without prior object detection. This algorithm identifies objects based on color levels and segments the redundant background information along with the objects.



(a) (b) (c)

(d) (e) (f)

(g) (h)

Figure 2.10: Qualitative analysis of the proposed multilevel fixation oriented segmentation (M-FOS) with state-of-the-art image pixel level segmentation bounding box-cropping segmentation techniques. (a) Query image, (b) Fixed size bounding box cropping segmentation (c) SLIC- Graph cut normalized [52, 108-110], (d) Quickshift [52, 111], (e) Felzenszwalb [109], (f) Marker-Controlled Watershed Segmentation [52, 112], (g) Level2 segmentation without prior object detection, and (h) M-FOS.

Figure 2.10 (h) shows the output for the proposed multi-level fixation-oriented object segmentation method (M-FOS). M-FOS detects and segments both the objects of interest and the orange dot

indicator without redundant background information. Extensive computer simulations were performed on the rest of the dataset to test the accuracy and robustness of fixation-oriented object segmentation.

Towards Deep Learning Segmentation for Eye-Tracking Analytics: AI techniques have a widespread application in computer vision research, including image classification, object detection, image segmentation, and image enhancement. Semantic segmentation is the process of recognizing and comprehending the image's components at the pixel level. Typically, such algorithms produce regions with distinct classes as outputs. Semantic segmentation enabled by deep learning enables a slew of new applications, including smartphones and real-time video segmentation on mobile devices.

Towards creating synthetic datasets for deep-learning methods: A lack of training data impedes a promising performance even utilizing the best off-the-shelf deep learning segmentation method. The results generated by the proposed M-FOS can yield an adequate number of training images. Combining the proposed M-FOS and state-of-the-art deep learning segmentation method will benefit future research in academic and commercial sectors. A novel deep-learning-based eye-tracking visualization and analysis tool, 'ISeeColor,' was developed to achieve these goals. The following section will describe the ISeeColor methodology in detail.

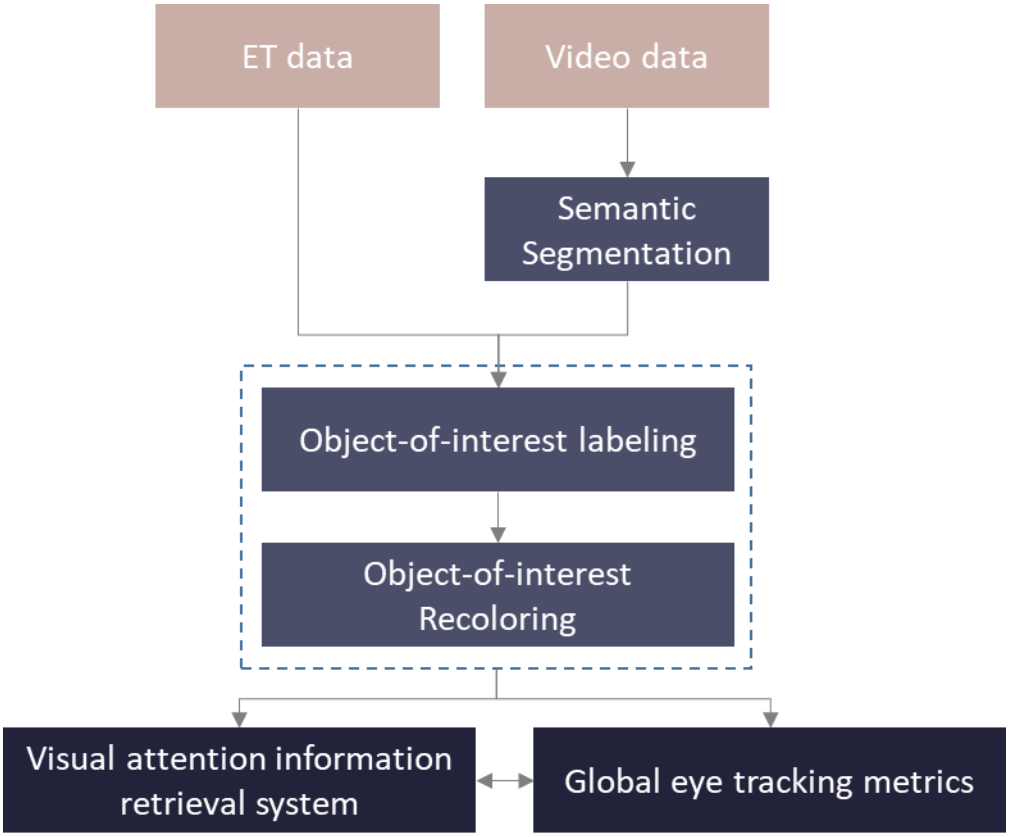


Figure 2.11: Overview of ISeeColor architecture. The key algorithmic components of the system include semantic segmentation, object classification, recoloring, and global eye-tracking analysis. This architecture overcomes the necessity of requiring template images for object recognition.

2.3 System Architecture for ISeeColor and Illustration

The proposed system architecture "ISeeColor" is illustrated in Figure 2.11. Algorithm 2.1 describes the key activities of the ISeeColor architecture.

Let V_0 denote the video captured by the eye tracker's front scene camera and D_0 denote the eye-gaze fixation data obtained from the two infrared cameras that monitor pupil movement. Let us refer to the fixation locations in each frame as $F_1^l, F_2^l, \ldots F_k^l$ $F_1^l, F_2^l, \ldots F_k^l$. Notably, additional interpolation between the generated eye movement data and the recorded front scene video is required via the eye tracker's internal clock, as the infrared and front scene cameras operate at different frequencies. Furthermore, M is used to store calculated statistical metrics for eye tracking.

For simplicity of exposition, this work quantifies and presents metrics such as fixation duration, fixation rate, and total fixation count. C is a vector that stores the category of each frame's fixated object of interest (OOI).

2.3.1 Data Acquisition

The input data for this study used SMI eye-tracking glasses with a 0.5° gaze position accuracy across all distances [113]. Calibration was performed with a gaze cursor in a three-point calibration mode. An outdoor case study collected eye-tracking data from real-world users walking around the Tufts University campus.

Algorithm 2.1: Describes the key activities of the ISeeColor architecture.

for k from 1 to K, do

Perform image semantic segmentation on I_k

Look up if F_k^l belongs to any segments of I_k

Update M_k via calculation of the eye-tracking metrics

Update C_k via automatic categorization of fixated objects-of-interest

Update I_k to I_k' via solving equation 2.14

$I_1', I_2', \ldots I'_k$; M; C and data representation ISeeColor including all the processed information

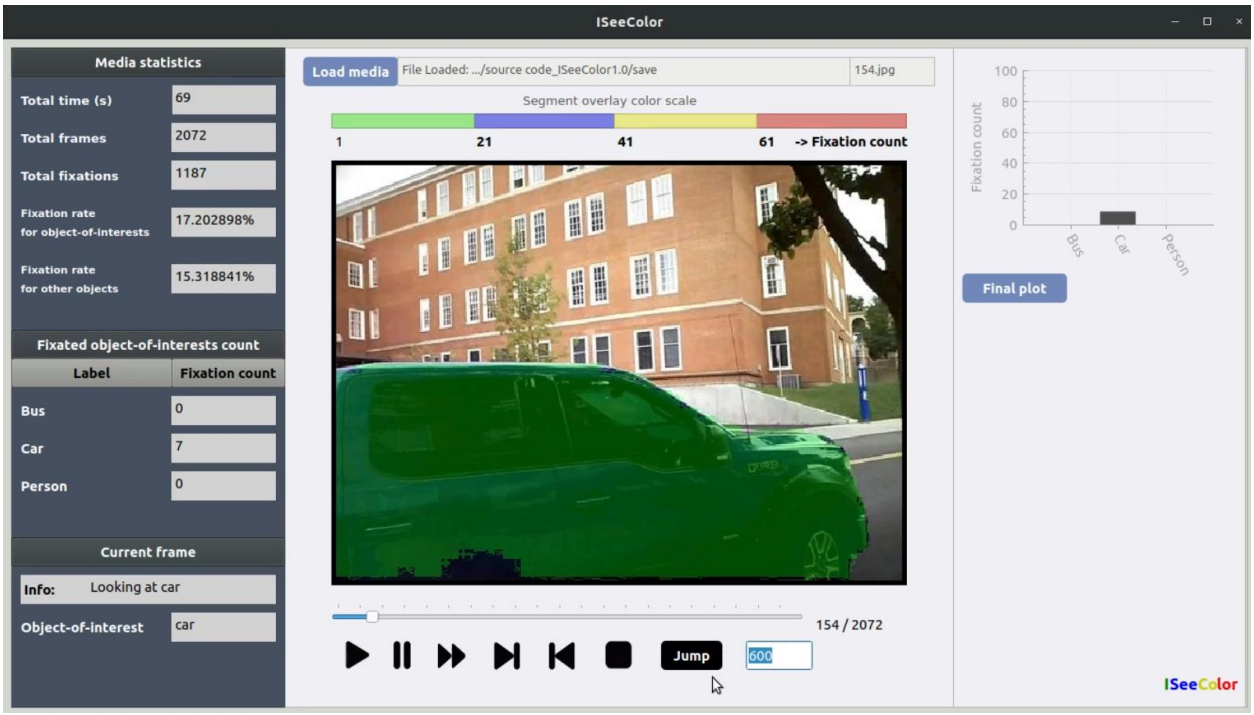
2.3.2 Image Semantic Segmentation Using Deep Learning

The authors were unable to train the dataset with eye fixations because they were unable to locate any publicly available semantic segmentation datasets that included fixation data. As a workaround, the authors used state-of-the-art segmentation methods and eye-tracker fixation masks to obtain the object-of-interest in this method.

The proposed ISeeColor can automatically detect, categorize, and highlight OOI in a dynamic scene. The ISeeColor architecture identifies the fixated OOI by correlating fixation coordinates

with segments (objects) in a frame. The fixated OOI is then overlaid with a transparent color mask whose color intensity varies as the duration of the fixation increases.

(a) Input (b) Segments (c) Recolored output based on eye gaze location and density



(d) ISeeColor Graphical User Interface (GUI)

Figure 2.12: Visualizes the ISeeColor GUI and its operations. The (a) Input image/video data is first passed through a deep-learning model to obtain (b) semantic segmentation outputs. Next, the data streams are overlayed with a composite mask based on the fixation metrics and OOI. In this case, (c) Fixation count (fc) is used as a metric to identify the color of the overlay. Finally, all of these utilities can be accessed through the ISeeColor GUI.

Segmentation of images semantically refers to the process of assigning a class label to each pixel in an image, such as a car, truck, person, or dog. Semantic segmentation using deep convolutional

neural networks groups together parts of the image that belong to the same object. This technique has garnered increasing interest from both academia and industry. For simplicity of exposition, this system will utilize DeepLabv3 [114-118] and Context Encoding Network [119].

Chen et al. [118] developed DeepLabV3, an image classification framework remodeled to perform semantic segmentation. The fully connected layers are transformed into convolutional layers, and then the feature resolution is increased via Atrous convolution. To restore the original image resolution, upsampling via bi-linear interpolation is used. This serves as the input to a fully connected Conditional Random Field [120], enhancing segmentation results.

Zhang et al. [119] developed the Context Encoding Network (EncNet), a network that captures global image information to achieve accurate scene segmentation. The model extracts feature maps using a ResNet network and then feeds them into a Context Encoding Module developed by Zhang et al. [121].

2.3.3 Object-of-Interest (OOI) Color Transform Using Alpha-Blending

The image semantic segmentation algorithm will generate a group of segments, $\{S_1, S_2, \dots, S_n\}$, where n is the number of possible OOI categories. For each video frame, a pixel-based linear operation is implemented as described in the equation below:

$$I' = \begin{cases} \alpha I + (1-\alpha) I_{color}; I(x,y) \in \bigcup_{i=1}^{n} S_i \\ I \quad ; \quad otherwise \end{cases} \qquad 2.14$$

where I' is the color composite image output. I is the original eye-tracking video frame. I_{color} is the RGB color selected by fixation duration. $0 < \alpha < 1$ is used to control the transparency of overlaid color; in this work, $\alpha = 0.3$.

2.3.4 ISeeColor Data Visualization and Analysis GUI

This section demonstrates ISeeColor's robust capabilities via a graphical user interface (GUI), in which the OOI is colored according to the fixation duration and displayed directly on a video stimulus. Users can instantly gain insights from the visual stimuli that have been recorded. For semantic segmentation, researchers will initially choose between DeepLabV3 and EncNet. Other models can be added in the future. The object categories can be customized to meet the specific needs of researchers. The analysis software for the eye-tracker provides the fixation location and duration. Longer fixation durations are thought to reflect cognitive functions such as active attentional deployment, whereas shorter durations are thought to reflect a scene's visual complexity [69]. Additionally, ISeeColor includes features for (1) viewing histogram plots of the objects viewed during video playback, (2) searching for frames containing an object, and (3) filtering objects with a long fixation duration.

The researcher may select eye-tracking metrics pertinent to the tasks and research questions. As an illustration, this chapter illustrates the ISeeColor GUI features listed in Table 2.2. Additionally, ISeeColor creates a histogram plot that is updated automatically as the frames are updated (in the forward or reverse direction). Furthermore, pertinent information regarding the presence and location of fixations in a video frame is displayed. Figure 2.13 illustrates the amount of time spent manually annotating two case study videos by an inexperienced and an experienced coder (video one is 1 minute 47 seconds, and video two is 3 minutes.

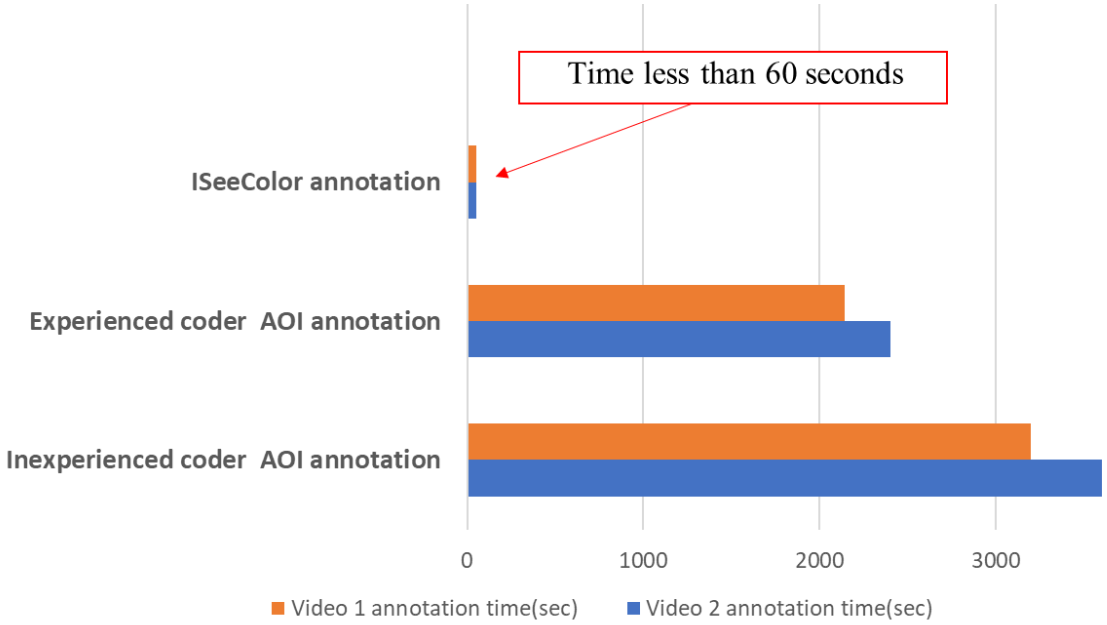


Figure 2.13: Comparison of annotation: human labor time between manually ROI annotation and ISeeColor. Notice that the ISeeColor annotation time is significantly minuscule compared to manual annotation methods. Video 1 is 1 minute 47 seconds long, and video 2 is 3 minutes long.

Finally, all of the experiments above show the robustness and utility of the proposed ISeeColor system that incorporates gaze information (location), enables automatic OOI labeling via image semantic segmentation (classification), and enables data visualization via fast-speed image recoloring based on fixation duration (duration).

2.3.5 User-Study and Comparison

In order to understand user behaviors, needs, and motivations through observation techniques, task analysis, and to obtain feedback on the ISeeColor GUI, a user study was conducted. In order to remove bias, a wide spectrum of individuals were invited to test the system. In total, 51 eye-tracking researchers and non-researchers (including individuals from cognitive psychology, electrical engineering, management studies, computer science, industry professionals, and schoolteachers) took part in the data visualization and analysis tool survey of ISeeColor. Each

Table 2.2: Media Statistics in ISeeColor [39]. ISeeColor can include most forms of eye-tracking metrics. For simplicity of exposition, the following statistical metrics were used.

Total time		A metric to record the task time
Total frames		Number of frames = time × frames /second
Total fixations		Provided by a commercial eye-tracker
Fixation rate	OOI	$\text{Fixation Rate}_{\text{objects}} = \frac{\#\text{ of fixations}_{\text{objects}}}{\text{Total time}}$
	No OOI	$\text{Fixation Rate}_{\text{No objects}} = \frac{\#\text{ of fixations}_{\text{No objects}}}{\text{Total time}}$
Number of fixations for OOI		Number of fixations the viewer spent on OOI

participant was given a short overview of eye-tracking, its challenges, and the current annotation methods. Each participant was further shown a short demo of ISeeColor, which included an overview of its functions and how it works.

Each question in the survey could be rated from 0 (extremely useless) – 10 (extremely useful). Table 2.3 shows the results of the survey. The high mean values of the results for all survey questions indicate the effectiveness of the proposed system. Furthermore, the survey also asked for suggestions that could improve the visualization and the analysis process to solve the given tasks.

Some of the feedback included interest in adding abilities to (1) click an object-of-interest and get the frames filtered from the video, (2) change the color scale to accommodate a few cases of color blindness or color aphasia, and (3) allow the user to adjust fixation count ranges to increase flexibility. The authors will include these valuable changes in future releases of the software. Overall, while wearing head-mounted eye-trackers in dynamic contexts, ISeeColor can effectively and intuitively represent what, where, and how long users view objects or areas in a scene.

Table 2.3: The ISeeColor user survey results. Researchers and non-researchers (professionals) from academia and industry participated in this evaluation. Each participant received an overview of eye tracking, its advantages and disadvantages, and a demonstration of ISeeColor.

Total number of participants	51		
Total number of participants who are familiar with eye-tracking technology	30		
Based on the ISeeColor demo, the following aspects of the software were rated as follows:	Mean	Std	Var
The information is clear, concise, and informative to the intended audience	8.32	1.28	1.64
The purpose of the software is well-defined and clearly explained to the user	8.49	1.58	2.51
The organization of the software is clear, logical, and effective, making it easy for the intended audience to understand	8.62	1.35	1.83
Computer capabilities such as graphics, color, or sound are used for appropriate instructional reasons	8.64	1.26	1.6
Based on the ISeeColor demo, the usefulness of the following parameters or control options of the software was rated as follows:	Mean	Std	Var
Color-scale setting	8.51	1.58	2.51
Video control panel	8.79	1.29	1.67
Media Statistics	8.31	1.74	3.04
Histogram plot	8.15	2.16	4.68

2.4 Extended Applications for Real-World Dynamic Contexts

ISeeColor has several potential applications for research and industry. Extracting meaningful inferences from FoV scene videos and eye-tracking data during real-world locomotion remains challenging. However, these inferences can yield enormously helpful insights.

Consider researchers who wish to evaluate how well the cognitive theories developed in controlled laboratory settings apply in real-world dynamic contexts. Alternatively, consider the narrow problem where higher fixation durations could indicate (1) deeper cognitive processing of a stimulus, (2) greater difficulty in extracting information from the stimulus, or (3) that the stimulus is highly engaging. Researchers can use ISeeColor to detect and infer these scenarios in seconds.

In another instance, ISeeColor can be used to determine the extent to which someone looks at street signs, building, roads, or their smartphone map while walking around a city. This information can then be related to their subsequent memory of the environment.

Moreover, ISeeColor functionality could be applied to large-scale studies focused on understanding differences in viewing behavior between neurotypical populations and people with learning disorders or suffering from schizophrenia or Alzheimer's. Consequently, ISeeColor has great potential for informing our understanding of human vision in naturalistic settings in an intuitive and user-friendly way.

ISeeColor can be used in product design to maximize profit, user satisfaction, and safety in the industry. For example, designers of semi-autonomous vehicles can examine the impact of system design choices on drivers' visual attention to critical OOI while driving. Another example includes market researchers determining what products catch shoppers' eyes during store visits to optimize product placement on shelves [122]. ISeeColor could open up new avenues for work in human-machine interaction and smart devices' user experience.

2.5 Conclusion and Future work

This chapter presents a multi-level fixation-oriented object segmentation method (M-FOS) to solve the challenges in segmenting the scene objects in video data collected by an eye tracker to support cognition research. M-FOS shows its advancement in position-invariance, illumination, and noise tolerance.

The proposed method utilizes (a) rotation and scale-invariant features to identify the objects and (b) multi-scale thresholding and masking on HSV color space to segment the object. The proposed

technique is tested using real-world case studies designed by a team of psychologists from Tufts University focused on understanding visual attention in human problem-solving.

Extensive computer simulation demonstrates the method's accuracy and robustness for fixation-oriented object segmentation. Moreover, incorporating the segmented results from M-FOS as label data for deep-learning image semantic segmentation was explored to demonstrate the possibility of online deployment of eye tracker fixation-oriented object segmentation.

This chapter further proposed a novel visualization and analysis tool, ISeeColor, enabling large-scale wearable automatic eye-tracking data annotation and visualization. ISeeColor is the first eye-tracking data visualization and analysis system to automatically visualize fixation events (duration and location) on a semantic interpretation of AOIs in dynamic scene video recordings. ISeeColor can provide the knowledge of what the user looked at, where they looked at, and how long they looked. ISeeColor implements efficient and low-cost objects-of-interest recoloring and colors to represent fixation duration values. Furthermore, ISeeColor is compared with a human annotation expert. ISeeColor can also be used as a gaze density information retrieval system to sort and search objects of interest based on pre-encoded color information.

Ultimately, to prove the effectiveness of the proposed method, an extensive user study was conducted with 51 participants from academia and industry, including researchers and non-researchers.

2.6 Acknowledgment

This work is supported by the U.S. Army Combat Capabilities Development Command and was accomplished under Cooperative Agreement Number W911QY-15-2-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as

representing the official policies, either expressed or implied, of the U.S. Army Combat Capabilities Development Command, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes, notwithstanding any copyright notation hereon.

Chapter 3: Systems to Improve Situational Awareness of First-responders

Abstract

Navigation, the process of accurately ascertaining one's position followed by planning and following the route, is essential for any individual or group. Navigating real and virtual environments, known and unknown, and indoor and outdoor is challenging. Traditionally, localization and navigation algorithms are limited to known environments with the help of global positioning systems. In the metaverse, dynamic localization and navigation could open doors to a new form of digital technology. In fire-fighting and defense, it could be a crucial factor in determining the success of a mission.

This chapter proposes to develop an augmented reality-based system to provide 3-dimensional (3D) navigational insights in unfamiliar terrestrial environments. Further, two novel methods for indoor navigation systems are proposed.

Index terms: Augmented-reality, GPS, HoloLens, vision, positioning system, localization, navigation, three-dimensional, image processing, mesh, mixed reality, SAR.

Publications

Rajeev, Srijith, Arash Samani, Karen Panetta, and Sos Agaian. "3D Navigational Insight using AR Technology." In 2019 IEEE International Symposium on Technologies for Homeland Security (HST), pp. 1-4. IEEE, 2019.

Rajeev, Srijith, Qianwen Wan, Kenny Yau, Karen Panetta, and Sos Agaian. "Augmented reality-based vision-aid indoor navigation system in GPS denied environment." In Mobile Multimedia/Image Processing, Security, and Applications 2019, vol. 10993, p. 109930P. International Society for Optics and Photonics, 2019.

Spatial knowledge or memory is a form of memory responsible for recording information about one's environment and spatial orientation. It is typically acquired from one or more spatial perspectives; most commonly survey or route. Though people typically navigate using spatial knowledge acquired through direct navigation or extended experience with area maps, there are many cases where people must navigate environments on short notice and without conclusive spatial knowledge. For example, military personnel and emergency first responders are projected to navigate unfamiliar environments without sufficient time to study the route. For instance, military personnel may view a two-dimensional (2D) map during a mission briefing and then be transported directly to an emergency site [123, 124]. Studies have shown the relative positive effectiveness of spatial learning perspectives during simulated transport for supporting the eventual navigation of an environment [123, 124].

The term "localization" refers to determining an individual's position in relation to a reference frame [125]. The accuracy of the localization system is used to determine its performance. It is defined as the degree to which estimated or measured data conforms to a defined reference value at a given time, which is ideally the true value. The most frequently used method of localization and navigation is to use a map in conjunction with sensor readings and some form of closed-loop motion feedback [126]. Accurate outdoor and indoor location determination is critical for enabling a variety of context-aware services and protocols to function. Figure 3.6 shows visual illustrations of common positioning systems.

Accurate localization and tracking of user position are critical for a number of applications in the metaverse, defense, and first-responder technology. First responder location and tracking are performed by constantly relaying their whereabouts through radios. Additionally, the vast majority of first responders are unaware of the environment they are entering. It is crucial that the incident

commanders locate, track their movements, and assess their vitals in real-time when first responders respond to an emergency.

To solve these challenges, this chapter proposes to develop an augmented reality-based system to provide 3-dimensional (3D) navigational insights in unfamiliar terrestrial environments. Further research developed two indoor navigation systems that enable better visualization of the disparate data sources, thus making it conducive for the human operators to detect hazards and victims.

First, in section 3.1, a 3D Navigational System that provides insight into the surroundings, including terrain and navigational paths, are introduced.

Second, in section 3.2, an indoor navigation system (INS) with AR capability is developed utilizing computer vision to identify and localize a user, AI for path planning, and AR for path visualization.

Third, in section 3.3, an advanced INS using LiDAR devices for dynamic mapping irrespective of light and environment conditions is proposed. While LiDAR-INS-based systems have been available for some time for horizontal mapping, the z-axis has proven elusive. A barometric sensor is utilized to solve the challenges of identifying altitude changes while climbing stairs or using an elevator. Finally, sections 3.4 and 3.5 provide discussion, conclusions, and next steps.

3.1 3D Navigational Insight using AR Technology

Basic map and compass land navigation training are used to continue dominance in ground warfare. The Military Grid Reference System uses a compass, map, and protractor to pinpoint regions during land navigation. However, typical map and compass land navigation training exercises do not provide the optimal spatial knowledge. The U.S. Department of Defense (DoD) developed the Global Positioning System (GPS), a satellite-based navigation system. Its capabilities are limited in remote terrains, and GPS denied regions. Other navigation techniques include Wi-Fi positioning systems (WPS), Bluetooth, and visual positioning systems [46]. However, none of these systems can provide sufficient visual stimulation for spatial learning.

Moreover, over-reliance on the GPS signal may significantly lose life and equipment if the GPS signal is lost. The current threats to GPS, ranging from anti-satellite weapons to solar storms, expose a critical vulnerability of over-reliance on GPS technology to move on the battlefield and communicate and engage targets. Currently, unfamiliar or unexplored open environments are studied using 2D imagery such as aerial satellite images or surveillance images. Although training scenarios (or camps) simulate such environments, they are not similar to the exact environment. Robot mapping is another technique used to obtain 3D/2D maps. For instance, a state-of-the-art visual SLAM algorithm tracks the camera's pose while simultaneously building an incremental map of the surrounding region [127]. There is a significant need for a system to visualize environments unknown to military and emergency first responders. Moreover, this system would automatically identify paths in these environments.

Humans can learn complex unknown environments in various guidance tasks and use the knowledge to determine near-optimal (e.g., minimum-time) performance and remain versatile and adaptive to unexpected changes in the environment. Augmented reality (AR) is a technology that

superimposes visual imagery on a user's view of the real world, thus providing a composite view. Using AR technology, users can perceive virtual and real objects as co-existing in the same environment. Moreover, it is possible to apply machine learning algorithms to AR systems. Standalone augmented reality (AR) systems have great potential for interactive three-dimensional (3D) geo-visualization. Emerging AR technologies can display rich graphical imagery of large-scale environments and permit intuitive interaction through gestural and voice inputs. Where unfamiliar environments must be navigated efficiently and without online guidance, studying interactive 3D environment visualizations with real-time navigation aids could support environmental learning and navigational efficiency. The proposed method will integrate 3D augmented reality and navigation into land navigation and unit position tracking operations. AR is a new development in the area of mobile learning. AR apps use the cameras on mobile devices to produce live views of the physical world and add digital sources, such as maps.

Studies have shown the relative positive effectiveness of spatial learning perspectives during simulated transport for supporting the eventual navigation of an environment. However, such visualization data is limited. There is a need for a system to visualize environments unknown to military and emergency first responders. Moreover, this system should automatically identify paths in these environments.

Synthetic-aperture radar (SAR) is a type of radar used to create three-dimensional models of objects such as the surface of planets. SAR can acquire imagery during the day and night as SAR provides illumination. This research proposes a proof-of-concept to construct 3D data that can be used in Augmented Reality (AR) and Virtual Reality (VR) based systems to provide 3-dimensional (3D) navigational insights in unfamiliar environments. The proposed method will create an

iteratively refined 3D landscape knowledge database that can assist personnel in navigating familiar or unfamiliar environments or assist personnel in mission planning.

The main contributions of this section are (1) to use multi-modal data fusion to generate 3D terrestrial maps leveraging Synthetic Aperture Radar (SAR) data, Google Earth imagery, and (2) to generate traversable surfaces, (3) to use AI to perform path planning operations for navigational.

3.1.1 Related Work

Kirscht et al. demonstrated that shadows in synthetic aperture radar could reconstruct 3D models of buildings and vegetation. Xu et al. [128] developed an iterative and parametric approach to model buildings as cuboids for automatic reconstruction of 3-D buildings from multi-aspect SAR images. Further postprocessing steps, such as combining, eliminating, prolonging, and classifying, extract parallelogram-like building images. Xu et al. [129] conducted an initial study on SAR-based information retrieval and scene reconstruction by introducing a parametric framework for target recognition, estimation, and reconstruction.

To overcome the challenges of existing line-based extraction methods that can only detect straight lines, Guo et al. [130], inspired by density clustering [131], proposed a point cloud extraction method using the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm [132]. This method preserved the building structure more completely by enabling the extraction of various shapes of the buildings.

Recla and Schmitt [133] developed a deep learning-based single image height prediction for the other important sensor modality in remote sensing: synthetic aperture radar (SAR) data. The authors[133] noted a disadvantage because the approach was limited to urban areas rich with height

cues [134]. Rizzi et al. [135] studied a multi-sensor-aided SAR system to obtain accurate images of driving scenarios.

3.1.2 Proposed Approach

The proposed method for developing 3D maps of open terrain and its 3D map development is described in the following steps:

1. **3D map development using existing data**

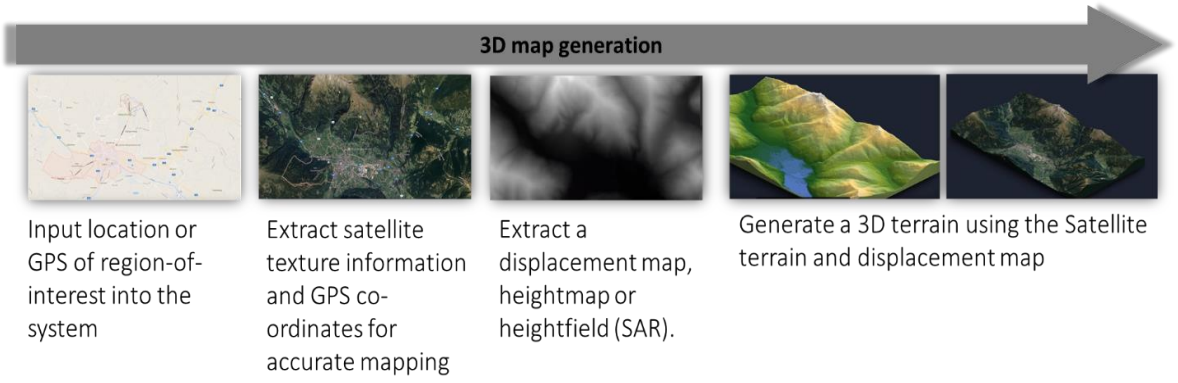


Figure 3.1: shows the proposed method for generating the 3D terrain—images obtained from [136]. GPS coordinates are used to extract the texture information and SAR information.

This project develops 3D terrain using Google Earth images and SAR data. The SAR data at the UNAVCO Data Center [137] includes satellite-transmitted and received radar scans of the Earth's surface. SAR data, analyzed using Interferometric SAR (InSAR) techniques, can model millimeter-to-centimeter scale deformation of the Earth's surface over regions tens to hundreds of kilometers across [138]. Figure 3.1 describes the method used for generating the 3D maps. SAR data is essentially a volume file without texture. The satellite information and the SAR maps are first rectified and then superimposed to obtain the terrain volume, as shown by Monterroso [139].

Finally, the texture data from the satellite maps are applied to obtain a high-quality terrestrial 3D map.

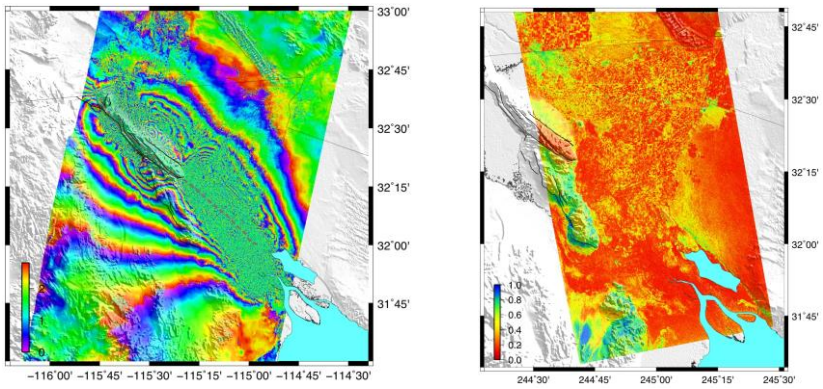


Figure 3.2: SAR data; ground deformation along the San Andreas fault, from ESA Envisat data [140, 141]. SAR data can see through the darkness, clouds, and rain, detecting changes in habitat, water and moisture levels, natural or human disturbance effects, and changes in the Earth's surface [142].

2. Artificial Intelligence (AI) based path planning using Navigational Mesh

Using AI and the 3D terrain generated, an abstract data structure can be created to aid individuals in pathfinding through complicated spaces. A navigation mesh is an abstract structure used in artificial intelligence applications to aid agents in pathfinding through complex spaces.

A navigation mesh is a collection of two-dimensional convex polygons (a polygon mesh) that define which areas of an environment are traversable by agents [143]. Adjacent polygons are connected in a graph [143, 144]. Only the traversable polygons are available to input paths to the path-finding algorithm. This reduces memory-overloading. The path between polygons can be obtained in the mesh using one of the large numbers of graph search algorithms, such as A*[145]. Therefore, paths developed using navigational meshes tend to avoid computationally expensive collision detection checks with obstacles that are part of the environment [146, 147]. This process is widely used in game development, and with the advent of augmented reality devices, a similar technique can be developed for real 3D data-based navigation, as shown in Figure 3.3.

The following is a brief demonstration of how A* works within a NavMesh. As shown in Figure 3.3, let G be a graph with P walkable polygons and B blocked polygons. First, a set of polygons C ⊆ P through which an optimal path will pass is identified. P_s (start point) must be the first polygon in C, and P_g (destination point) must be the final polygon.

To locate the remaining components of C, each polygon in G is mapped with a single node in G'. For example, P_s is mapped to N_s and P_g to N_g. Each shared edge between two polygons in G is mapped to a shared edge between two nodes in G' as shown in Figure 3.4 [148].

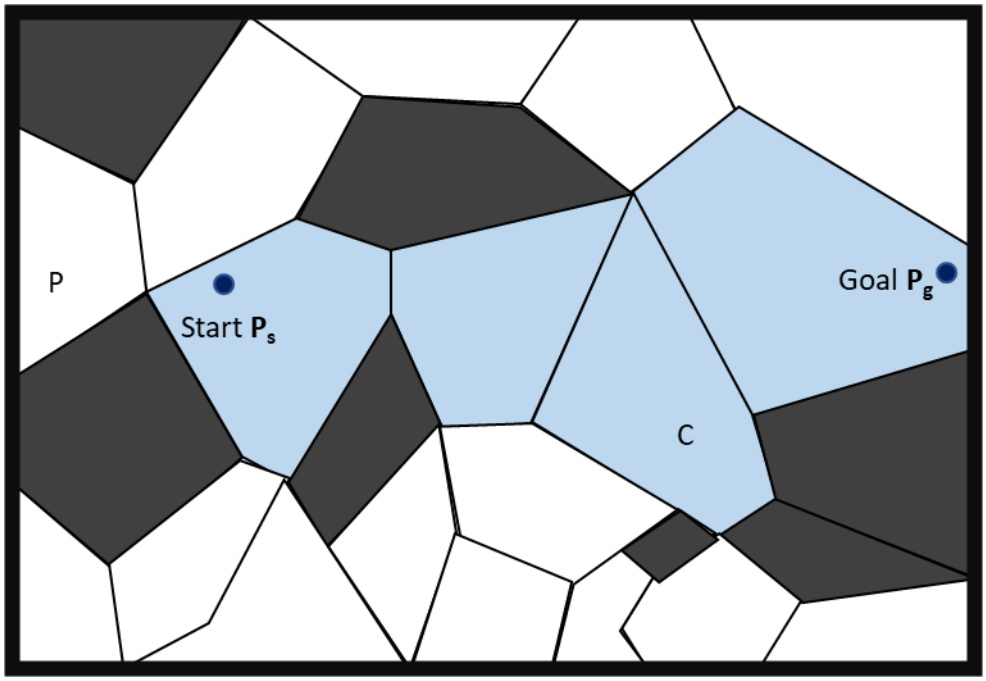


Figure 3.3: An illustration of a NavMesh with P polygons. A* will determine the polygons C in P that passes through Polygon with start point P_s and end at Polygon with point P_g.

Then, using A*, an optimal path p from Ns to Ng in G' can be determined. Each node in G' is associated with a polygon in G, and each node in p is associated with a polygon in C, as illustrated in Figure 3.5 [148].

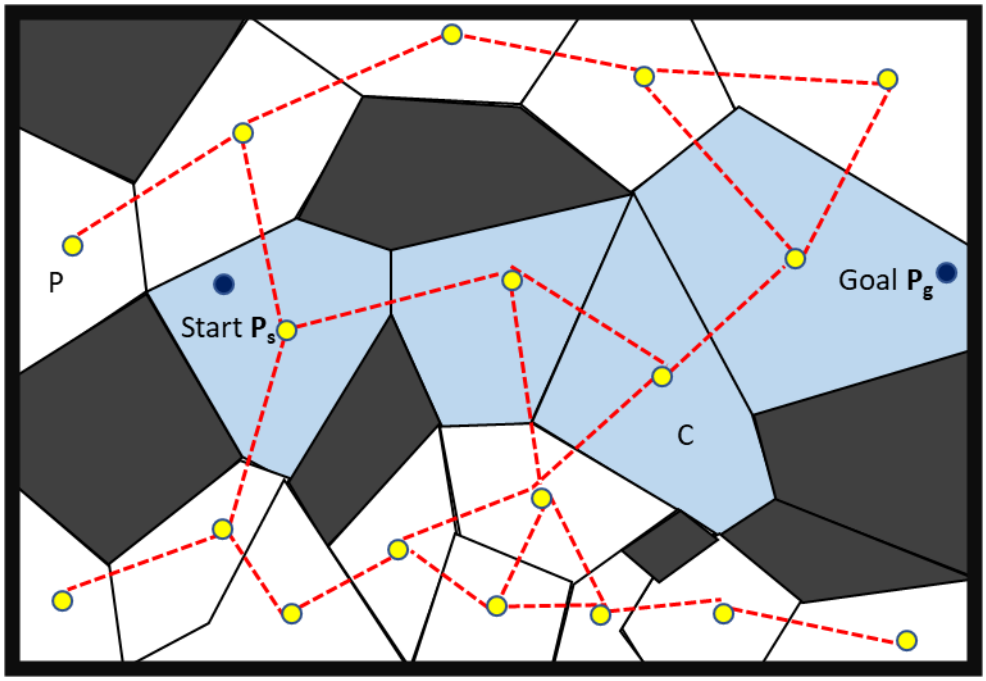


Figure 3.4: Polygons are transformed into nodes.

A* is an informed search algorithm, or a best-first search algorithm, in the sense that it is expressed in terms of weighted graphs: starting from a specific starting node in a graph, it seeks the shortest path to the specified goal node (example: least distance traveled, and shortest time). This is accomplished by maintaining a tree of paths that originate at the start node and extending them one edge at a time until the termination criterion is satisfied. A* must decide which paths to extend on each iteration of its main loop. It does so by calculating the path's cost and estimating the cost of extending the path to the goal. A* chooses the path that minimizes equation 3.1.

$$f(n) = g(n) + h(n) \qquad 3.1$$

where n is the path's next node, g(n) is the path's cost from the start node to n, and h(n) is a heuristic function that estimates the cost of the cheapest path from n to the goal. A* terminates the path it chooses to extend if no paths are eligible for extension. The heuristic function is application-

dependent. If the heuristic function is admissible and never overestimates the actual cost of reaching the goal, A* will always return the least-cost path from start to goal.

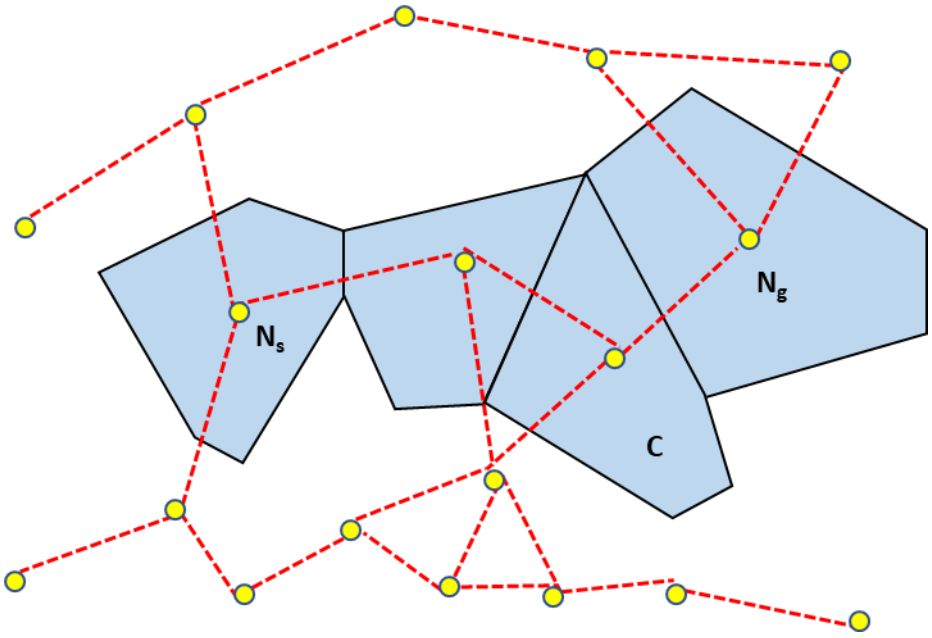


Figure 3.5: A* determines the optimal path from node N_s to N_g.

For instance, when looking for the shortest route on a map, h(x) may represent the straight-line distance to the destination, as this is the shortest distance physically possible between any two points. The Manhattan or octile distance is preferable for a grid map from a video game, depending on the available movements (4-way or 8-way).

Suppose the heuristic h satisfies the additional condition $h(x) \leq d(x, y) + h(y)$ for each edge (x, y) of the graph (where d denotes the length of that edge), h is said to be monotone or consistent. A* is guaranteed to find an optimal path without processing any node more than once when using a consistent heuristic. A* is equivalent to running Dijkstra's algorithm [149] at the reduced cost given by equation 3.2.

$$d'(x,y) = d(x,y) + h(y) - h(x) \qquad 3.2$$

As another example of a uniform-cost search algorithm, Dijkstra's algorithm can be viewed as a special case of A* with h(x) = 0 for all x [150, 151]. A* can be used to implement a general depth-first search by assuming a global counter C initialized to a very large value. Each time a node is processed, C is assigned to all newly discovered neighbors. After each assignment, one value is deducted from the counter C. Thus, the more recently discovered a node is, the greater its h(x) value. Without including the h(x) value at each node, both Dijkstra's algorithm and depth-first search can be implemented more efficiently.

3.2 Dynamic Localization and Mapping

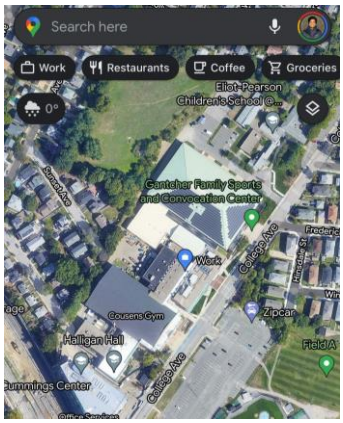
Firefighters, military, police, and paramedics play a critical role in protecting and aiding people at the scene of an emergency, such as fires, accidents, terrorism, and natural disasters. These situations are dynamic and require constant monitoring of a multitude of variables. Such constant monitoring may hinder humanitarian efforts and create lethal conditions for first responders and civilians.

While responding to the scene, disaster area assessment and search and rescue efforts rely on visual imagery captured from cell phones, aerial video, and other forms of media. Real-time assessment of disasters depends on the human operator's ability to visually identify the subject of interest through natural vision, videos, or images. Further, damaged buildings and roadways, debris, smoke, fire, and environmental conditions such as rain complicate the human observer's ability to detect victims and hazards. It is critical that rescuers can assess hazardous conditions before risking their lives and be armed with smart technologies that enable them to respond to dynamic situations.

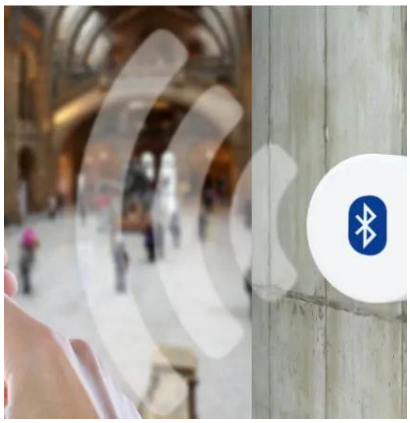
Researchers' biggest challenge is localizing the user based on their visible surroundings. Current solutions rely on the Global Positioning System (GPS) for tracking and orientation. However, GPS receivers have an accuracy of about 10 to 30 meters, which is insufficient for Augmented Reality (AR), which requires centimeter to millimeter-level precision.

There is an important need to monitor dynamic human location while reducing cognitive load and elevating human performance in dangerous and high-stress environments. The proposed framework will also enable better visualization of the disparate data sources, thus making it conducive for the human operators to detect hazards and victims. This product would also enhance safety and security applications, including firefighting, disaster relief, and search-and-rescue. The

approach of seamlessly utilizing and visualizing sensor information in a situational map to guide responders' actions and enhance their efficiency reduces the risk factor for responders.

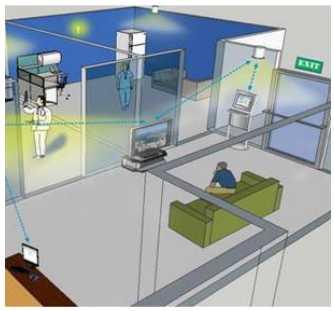


GPS [152]

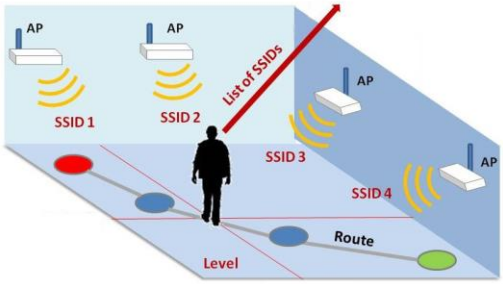


Bluetooth based positioning system [153, 154]

Visual positioning system[153, 155]



RFID [156]



Wi-Fi[157]

Figure 3.6: shows an illustrative example of the most widely used positioning systems.

To this end, this section of the chapter proposes a novel indoor navigation system (INS-1) with AR capability. INS-1 is developed utilizing computer vision to identify and localize a user, AI for path planning, and AR for path visualization. The proposed system is non-invasive. It uses the presence of a few distinctive image targets to determine the user's location. As a result, it eliminates the need for costly localization equipment such as beacons or RFID tags. It is not GPS-dependent, but the system can be extended to include GPS coordinates. Additionally, this system does not rely

on QR codes to determine its location. However, this system may not work in low-light to dark conditions. Furthermore, smoke and fire would create more challenges.

3.2.1 Related Work

The Global Positioning System (GPS) is a U.S. Department of Defense (DoD)-developed satellite navigation system [158]. While the Global Positioning System (GPS) is one of the most widely used outdoor localization systems, its capabilities are severely limited when used indoors. GPS signals require direct line-of-sight communication between transmitters (satellites) and receivers. However, as these signals bounce off buildings and penetrate walls and other enclosures, their signal intensity generally decreases [158]. These effects diminish the effectiveness of other known solutions for indoor localization that rely on electromagnetic waves (EM). Indoor positioning systems such as Wi-Fi, Bluetooth, RFID tags, and visual positioning systems are available.

Localization with wireless access points is accomplished using the Wi-Fi positioning system (WPS). WPS can be used in conjunction with or in place of GPS systems. Additionally, it can be used with GPS data to track users [159, 160]. Typical parameters that can geolocate a Wi-Fi hotspot or wireless access point are the access point's Media Access Control (MAC) address and Service Set Identifier (SSID). Possible signal fluctuations can amplify errors and inaccuracies along the user's path. Additionally, WPS is constrained by Wi-Fi connectivity and is inoperable when out of range of Wi-Fi signals.

Bluetooth operates in the ISM band at 2.4 GHz. Bluetooth is a "lighter" standard that is extremely widespread and supports various other networking protocols and IPs. Bluetooth positioning systems thrive on providing proximity information rather than precise location [161] because they were not designed to provide precise location like GPS. As a result, Bluetooth implements a geo-fence or micro-fence solution for indoor proximity.

RFID (radio-frequency identification) stores and retrieves data by transmitting it electromagnetically to a radio-frequency compatible integrated circuit [162]. However, its range is extremely constrained. RFID is a relatively new positioning technology that enables tracking objects or people's movements [163]. It uses radio waves to transmit an object's identity (e.g., a person's identity or a serial number) and other characteristics. RFID is insufficient for comprehensive localization due to its short range of less than a meter [42]. Additionally, the operability of such approaches necessitates a narrow passageway to keep them from passing out of range.

A visual positioning system determined the location of a camera-enabled mobile device by utilizing visual markers. Markers or image targets are placed at specific locations in such a system. The system determines the visual angle between the device and the marker to estimate the device's location coordinates in relation to the marker. The accuracy, coverage, and cost of common positioning systems are summarized in Table 3.1.

Table 3.1: Comparisons between popular navigation systems [164, 165].

Property	GPS	Wi-Fi	RFID	Bluetooth	Vision	Beacons	LiDAR (indoor)
Accuracy	2-20 (m)	5-15 (m)	Passive: 4m Active: 100m	1-5 (m)	1/10-1 (m)	1-8 (m)	0.5-10 mm
Coverage	Global	Depends on number of units	1/10 -100 (m)	1-30 (m)	1-10 (m)	30-75 (m)	<5m
Cost	High	Medium /High	High	Low	Medium/ High	Medium	High
Type	Outdoor	Indoor	Indoor	Indoor	Indoor	Indoor	Indoor

GPS orientation estimation is not a good solution for indoor positioning systems due to its limited ability to locate indoor subjects. Due to the high temperatures and constant environmental changes, beacons and WILP do not work well. Techniques based on IMUs are unreliable because errors accumulate exponentially over time. Due to the lack of edges and high resolution in thermal images, it is extremely difficult to obtain relative orientation using vanishing point detection [166].

Table 3.2: Vision-based Navigation systems, applications, and methods.

Vision-based Navigation systems	Category	Related work
Commonly deployed devices	UAVs	[167, 168]
	Mobile	[169]
	Wearable device	[170]
	Robots	[171]
Fields of application	Cognitive psychology	[172]
	Robot	[173]
	Autonomous Driving	[174]
	UAVs	[175]
	Human-machine interaction	[176]
	Universal Accessibility	[177]
	Medical and clinical applications	[178]
Popular methodologies	Visual Localization and Mapping, SLAM	[179-182]
	Image Classification and	
	Object Detection and Tracking	
	3D reconstruction	

Vision-based navigation systems have been a dominant approach in fields such as robotic navigation systems [171], autonomous driving products [167], and medical applications [178]. The development of vision-based navigation methods has been an emerging topic in real-life human-machine interaction consumer products as the camera calibration process has become

easier and the cost of imaging capture devices has dropped dramatically [176]. A wearable vision-based navigation system [170] was developed to assist blind and visually impaired people with local navigation by providing online feedback on obstacles and useful objects. Investigations Mapping and localization methods and image classification (global, local, bag-of-visual-words, and combinations) techniques have also been investigated in a comprehensive survey [120][183].

While AR technology allows for the creation and seamless integration of virtual digital elements into the real world, virtual information can also be displayed and used to assist navigation systems, particularly in GPS-denied or precise indoor environments [184]. A platform based on AR technology can provide people with a more natural navigation experience [185]. The combination of an AR platform and a vision-based navigation system has demonstrated its benefits in real-world scenarios, such as receiving route instructions in a shopping mall or airport to which the user has never been. Considering that an AR headset can provide a truly immersive experience for humans, this work introduces a vision-based AR navigation system. Table 3.2 summarizes the application areas and methods that are most frequently used.

3.2.2 Proposed Framework: Augmented Reality-based Vision-Aid Indoor Navigation System (INS-1)

GPS technology is ineffective indoors due to obstructive rooftops, walls, and windows. This issue, combined with the scarcity of widely available indoor maps, has posed a significant obstacle to indoor navigation capabilities. The proposed AR-based Vision-Aid Indoor Navigation System localizes the user's position using image targets, maps the features on a 3D map locally, and provides visual navigation assistance via an AR device. The components of the indoor navigation system will be used in the order listed. To begin, the application is launched. Second, the user searches the surrounding area for an image target [186]. The application then locates the user's

position on a virtual three-dimensional map. From here, the user can activate the Destination Input system by speaking the command "Set Destination." Once a destination is specified, the Navigation System performs pathfinding and displays a visual aid directing the user to the specified location. Microsoft HoloLens is used to deploy this project [187]. The proposed system's operation flow is depicted in Figure 3.7, and the following sections provide a more detailed description of this process.

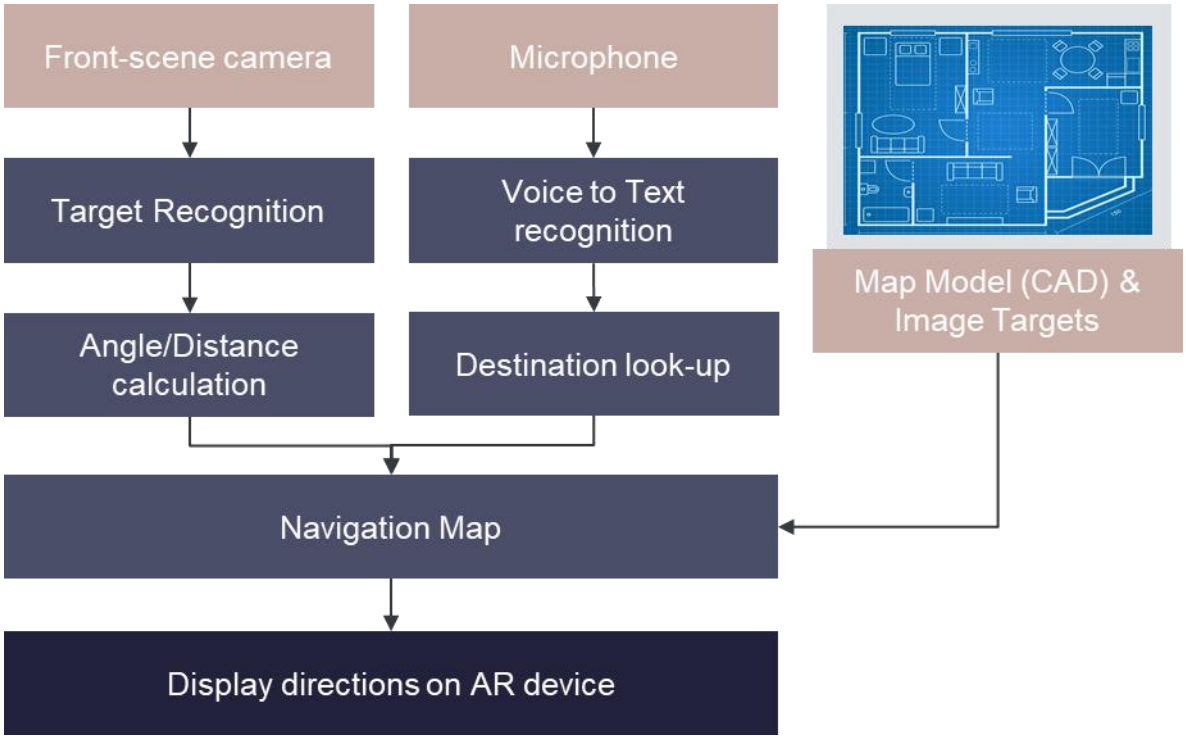


Figure 3.7: Block diagram of the proposed indoor Augmented Reality Navigation (INS-11) system utilizing computer vision to identify and localize a user, AI for path planning, and AR for path visualization.

1. A. User localization system

Localization and mapping are the computational problems involved in maintaining an up-to-date map of a known or unknown environment while tracking an agent's location. Image targets are used as markers in this system to determine the user's location and orientation in a three-

dimensional space. A front-scene camera uses a visual landmark as an image target to determine the user's position. Good image targets are detailed – busy patterns and/or images, have a high contrast – image edges correspond to sharp color transitions, lack repetitive patterns – corresponds to a unique image target – and contain a high number of natural features – these are sharp, spiked, chiseled details in the image. Additionally, the system should include only static image targets. Additionally, the user's position in the model is updated automatically via the IMU device to maintain the same real-world position for world-space virtual objects.

A set of world axes defines each scene, and each object is defined by its axes. Positions are evaluated in relation to the world's axes. Rotations of an object are determined by the rotational offset of the object's axes relative to the world axes, not by the angle formed by a ray starting at the origin and passing through the object's position relative to the world axes. A transform contains information about an object's position and rotation. Thus, the term 'transform' refers to collectively an object's position and rotation information.

After Target Recognition, the transform of the image target is calculated in comparison to the transform of the camera. The image target's location on the map is known and is a permanent feature. A direct translation of the map's position offsets to the camera's position offsets is not possible because these entities use different coordinate systems. A single set of data is collected in relation to the camera's axes (imagine three axes that stay fixed to the head's sagittal, coronal, and transverse planes, no matter in which direction the user was looking). Another set of data is gathered in relation to the map's axes. As a result, it is necessary to transform the data to align their reference axes.

The purpose of localization is to determine the camera's orientation to the map. It is the process of determining how the map is oriented relative to the camera, as shifting the physical camera is

impossible. It is possible to determine the map's orientation regarding the camera using knowledge of the image target's orientation with respect to the map and the image target's orientation with respect to the camera. The camera and map Transforms are identical when the application loads. Notably, because the image target is a child of the map, its transform remains fixed relative to the map's transform – any transformations applied to the map also apply to the image target.

To localize, first translate the map by the image target-camera position offset and rotate the map by the image target-camera rotation offset around its center. This effectively aligns the map's center with the camera's transform relative to the image target. Following that, determine the offset in rotation between the image target and the map. The final step is to apply a negative translation along the map's axes of the image target-map position offset. Consider the reverse problem of localization, which is determining the map's coordinates and rotation. Assume apostrophes (or their absence) to denote coordinates and rotations with respect to (w.r.t.) distinct sets of axes, i.e., x denotes coordinates and rotations with respect to the world space axes, x' denotes coordinates and rotations with respect to the map axes, and x'' denotes coordinates and rotations with respect to the camera's axes. When the application is initialized, the camera view and map are set to (0,0) with the orientation set to 0^o. This means that their axes are parallel to the axes of the world's space. Due to the fact that the camera is not translated or rotated, the world space axes are identical to the camera. Thus, any measurements taken relative to the world space axes will be identical to those taken relative to the camera's axes.

Let *t* refer to the image target, *c* refer to the camera, and θ refer to the rotation. Then, the rotational invariance between the different axes be calculated using the following equation:

$$\Delta\theta^{tc} = \theta^{t} - \theta^{c} = \theta^{t'} - \theta^{c'} = \Delta\theta^{(tc)'} \quad 3.3$$

The image target (trackable) has map-relative coordinates $(x,y)^{t'}$ and rotation $\theta^{t'}$. The map's relative coordinates $(x,y)^{m'}$ and rotation $\theta^{m'}$ are always (0,0) and 0^o (unchanged). The camera-relative coordinates can be calculated using the following equation:

$$(x,y)^{t''} = (x,y)^t \text{ with rotation } \theta^{t''} = \theta^t. \quad 3.4$$

Finally, the camera's (user view) position is calculated using the following equation:

$$(x,y)^{c''} = (x,y)^c = (0,0) \text{ and } \theta^{c''} = \theta^c = 0^o \quad 3.5$$

To finally localize the user, assuming $(x,y)^{t''} = (x,y)^t$ and rotation $\theta^{t''} = \theta^t$, the map can be shifted using the following equation:

$$(x,y)^{m''} \text{ and } \theta^{m''} \rightarrow (x,y)^m \text{ and } \theta^m \quad 3.6$$

Algorithm 3.1: describes the steps taken to localize the user. Summarizing equations 3.3 to equation 3.6.

Input:	$(x,y,\theta)^t$, $(x,y,\theta)^{t'}$, $(x,y,\theta)^c = (0,0, 0^o)$, $(x,y,\theta)^m = (0,0, 0^o$
Trackable-Camera Position Offset:	$\Delta(x,y)^{tc} = (x,y)^t - (x,y)^c = (x,y)^t$
Trackable-Camera Rotation Offset:	$\Delta\theta^{tc} = \theta^t - \theta^c = \theta^t$
Trackable-Map Rotation Offset:	$\Delta\theta^{(tm)'} = \theta^{t'} - \theta^{m'} = \theta^{t'}$
Map-Camera Rotation Offset:	$\Delta\theta^{mc} = \boldsymbol{\theta^m} = \Delta\theta^{tc} - \Delta\theta^{tm} = \Delta\theta^{tc} - \Delta\theta^{(tm)'} = \theta^t - \theta^{t'}$
	Transformations that can be applied to the map to get it to $(x,y)^m$ and θ^m
	Map transforms starts at position (0,0) and rotation 0^o
	Translate Map by $(x,y)^t$
	Rotate the Map around by θm
	Translate Map along its axes by $(x,y)^{t'}$
Output:	User view is localized in the map

2. Map model and features

A navigation mesh is a collection of convex polygons in two dimensions (2D) [188]. This mesh defines the user-accessible area of an environment. Additionally, the navigational mesh serves as a reference system for an area's image targets and features. The proposed indoor navigation system's successful localization and navigation are highly dependent on an accurate map model and feature placement.

A three-dimensional model of the building is used to calculate distances between points of interest and to depict walkable surfaces. Due to the application's goal of pathfinding, high-resolution detail of the structure is not required; a model that demonstrates viable traversable paths is sufficient. This includes floors, which denote walkable surfaces, and walls, which denote barriers. Two methods are used in this chapter to create the model [125]. The first method involves manually creating the walls and floors of a building using geometric Game object prefabs in Unity. This method is feasible as long as the building's critical dimensions (room widths and hallway lengths) are known. However, this procedure was lengthy and prone to error. The second preferred method is to create architectural features that correspond to a given map [189]. The first step is to obtain the building's floorplan. Following that, it is georeferenced using the free software QGIS [190].

A voice command is used to specify the destination. A preliminary list of locations and coordinates is required to incorporate this feature into the system. This system sets the destination via voice input, as speaking is a natural way to communicate information, and typing on a headset is extremely inconvenient. The voice event handler is triggered when the phrase 'Set Destination' is heard. The system will then listen to the user's speech and convert it to text. The text is then compared to an online location dictionary. This process can also be carried out using a lookup table in the proposed system.

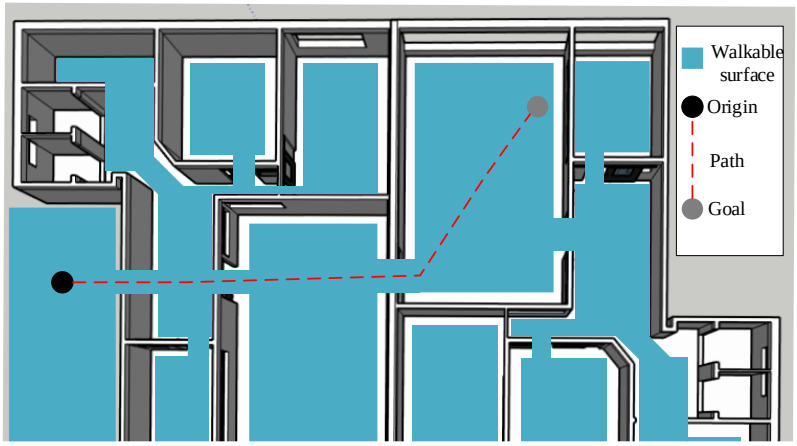


Figure 3.8: Shows a visual illustration of the navigation scenario in an indoor environment. The path planning algorithm identifies traversable paths (walkable surfaces) and identifies the shortest path from the origin to the destination.

3. Navigation system

Consider the illustration depicted in Figure 3.8. After obtaining the destination string, the application searches the list of possible destinations, identifies the correct one, and retrieves the destination's coordinates [191]. To determine the path between the initial and goal locations, the initial and goal locations must be mapped to their nearest polygons. Following that, the search algorithm begins searching (visiting) all neighboring polygons in the direction of the starting point until it reaches the destination. This system navigates using the A* (A-star) algorithm. A* employs a combination of shortest-path and heuristic searching [192]. Each cell or polygon is evaluated using the following algorithm:

$$f(n) = g(n) + h(n) \qquad 3.7$$

Where n is the path's next node, h denotes the polygon's heuristic distance (Manhattan, Euclidean, or Chebyshev) to the goal state, and g denotes the path's length connecting the initial goal states via the selected sequence of polygons. The path is visualized on the AR headset using the generated

path coordinates to aid the user in navigation.

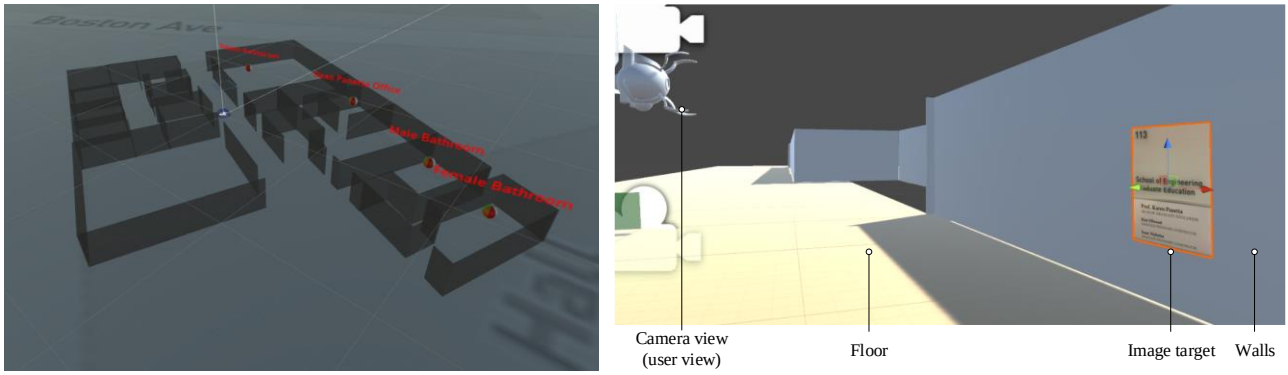
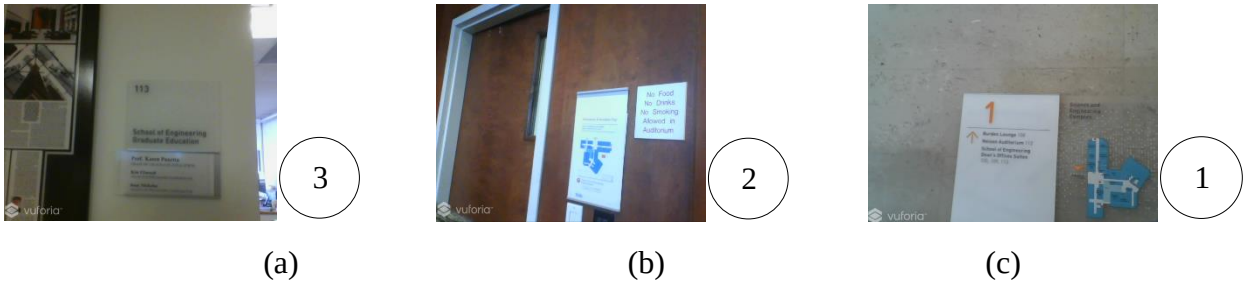
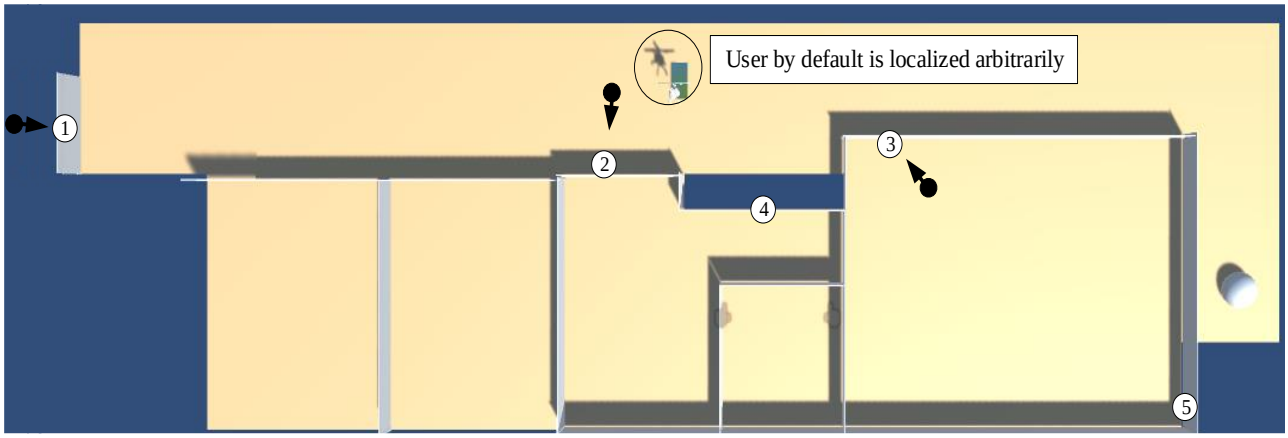


Figure 3.9: shows different views of the generated georeferenced 3D indoor map of a portion of Anderson Hall (Tufts University).



(a) (b) (c)



(d)

Figure 3.10: [a-c] depicts the images of the targets used in the computer simulation for localization; (d) depicts a section of the Tufts University Anderson Hall building with five image targets.

4. Computer Simulation

Table 3.3: shows the original and transformed position and orientation of the map with respect to the user (camera view).

w.r.t	Image Target (Figure 3.10 (a))		Image Target (Figure 3.10 (b))		Image Target (Figure 3.10 (c))	
	Original	Transformed	Original	Transformed	Original	Transformed
X	0	-2.9 ^{o}	-2.9 ^{o}	-4.0 ^{o}	-4.0 ^{o}	5.5 ^{o}
Y	0	0.6 ^{o}	0.6 ^{o}	-0.9 ^{o}	-0.9 ^{o}	-4.7 ^{o}
θ	0^{o}	204.3^{o}	204.3^{o}	44.3^{o}	44.3^{o}	283.6^{o}

A Microsoft HoloLens [193] was used to test this system in real-time. First, a Unity development environment was used to design an app based on the proposed method in C#. It was then loaded into the AR device and tested in Anderson Hall, Tufts University.

To begin testing the proposed visual indoor navigation system, the floor plan of Anderson Hall at Tufts University in Medford was obtained and converted into a navigational mesh. Floor plan features were extracted and georeferenced (shown in Figure 3.9). Notably, the proposed indoor visual positioning system does not necessitate the creation of virtual maps of the surrounding environment. Additionally, image targets were meticulously chosen and placed on the map.

User localization must be performed only once to ensure that the user is in the correct position. When the program is initialized, the user or camera is positioned at (0,0) with an orientation of 0^{o} (as illustrated in Figure 3.10 (d)). The map is not correctly oriented and positioned in relation to the user. Once the front-view camera captures an image target, the application will execute the user localization algorithm defined in the preceding section to obtain the correct user-map orientation and position. After applying and obtaining the correct transform, the application was then shown the next image target. This procedure was repeated for each of the scene's image

targets. Figure 3.10 [a-c] illustrates the successful transformation of the user's position and orientation in relation to the image targets. The user-map transformation is depicted in Table 3.3.

3.3 Dynamic 3D Mapping and Localization (INS-2)

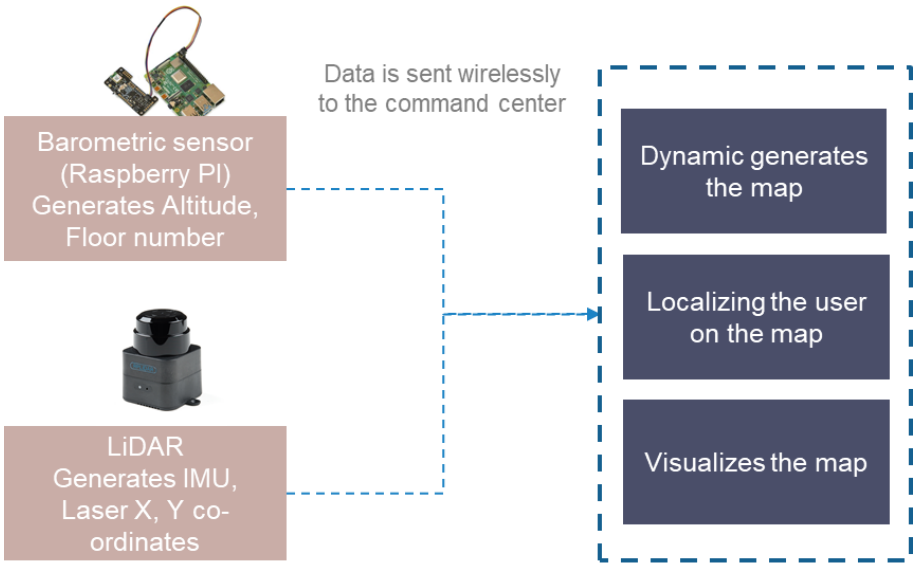


Figure 3.11: Shows the block diagram for the dynamic mapping and localization method (INS-2) to achieve multi-level building localization of a user. The LiDAR estimates the X and Y coordinates, and the barometric sensor estimates the z-coordinate.

Firefighting is a dynamic activity that involves a variety of operations occurring concurrently. Maintaining situational awareness is critical for firefighters to make accurate decisions in order to navigate a fire environment safely and successfully. While the system proposed in the previous section addresses these challenges, it may fail to work in hazardous conditions that include smoke, heat, fire, and smoke. This research utilizes recent technological advancements such as deep learning, point cloud, thermal imaging, and AR platforms to enhance a firefighter's situational awareness and scene navigation through enhanced scene understanding.

Simultaneous Localization and Mapping (SLAM) simultaneously estimates the state of a robot equipped with onboard sensors and constructs a model (the map) of the environment being

perceived by the sensors. In simple cases, the robot state is defined by its pose (position and orientation), although additional quantities such as robot velocity, sensor biases, and calibration parameters may be included in the state. On the other hand, the map represents important aspects (e.g., the location of landmarks and obstacles) of the robot's environment [194, 195].

The location of a set of landmarks is known as a priori in some robotics applications. For example, a robot operating on a factory floor can be provided with a manually created map of the environment's artificial beacons. Another instance is when the robot has access to GPS. In such cases, SLAM may be unnecessary if reliable localization with respect to known landmarks is possible. The popularity of the SLAM problem is related to the emergence of indoor mobile robotics applications. Indoor operation precludes the use of GPS to constrain the localization error; additionally, SLAM demonstrates that robot operation is possible in the absence of an ad hoc localization infrastructure [194, 195].

Apart from new algorithms, advancements in SLAM (and mobile robotics in general) have frequently been sparked by the availability of novel sensors. For example, the introduction of 2-D laser range finders facilitated the development of extremely robust SLAM systems, while 3-D lidars have been a primary focus of recent applications such as autonomous vehicles. Vision sensors have received significant research attention over the last decade, with successful applications in augmented reality and vision-based navigation.

Previous studies have shown that radar, LiDAR, and infrared cameras outperform all other rangefinders and cameras tested in these scenarios [196, 197]. While infrared cameras can be used to detect objects of interest, LiDAR systems can dynamically map and localize the firefighter as they are traversing through a building.

LiDAR-based SLAM devices can be potentially used by many applications requiring a globally consistent map, either implicitly or explicitly, for example, in various military and civilian applications, where personnel explore unknown and dynamically changing environments.

Although current systems can perform LiDAR-based SLAM estimation, they reach a failure point when tested on multi-level buildings. This is due to the inherent design of the LiDAR SLAM systems to map the layout of the paths in a 2D format. Although researchers have attempted to solve this challenge using Inertial Measuring Unit devices, they are erroneous due to the inherent drift of IMU measurements over time. 3D LiDAR devices can solve these problems, but they come at a much higher cost.

Table 3.4: Specifications of the sensors used for multi-level dynamic mapping and localization.

Sensor	Description
LiDAR [198]	Built-in IMU. Measuring range: 0.1-40 m. Accuracy <5 cm.
Barometric Sensor [199]	This device includes a GPS, Accelerometer, Gyroscope, Magnetometer (Compass), Barometric/pressure sensor (altitude), and Temperature.
Raspberry PI Zero [200]	A single-board computer with wireless and Bluetooth connectivity.

Ye et al. [201] developed a barometer-based multi-level floor localization method. While barometers have been integrated with other sensors, like inertial sensors [202], environmental sensors (example: light, temperature, and sound) [203], and location-based sensors (GPS) [199, 204, 205], they have not been used with LiDAR devices for indoor navigation and multi-level positioning.

Inspired by Ye et al. [201] and the hypothesis that LiDAR + IMU + Barometric based systems will provide real-time accurate indoor positioning in multi-levels, the proposed system utilizes an approach to combine data from a 2D LiDAR device, an IMU for local Z estimation, and a

barometric pressure sensor. A schematic diagram of the proposed multi-level dynamic mapping system is shown in Figure 3.11.

SLAM Estimation [195]: Consider an environment C. Let X be an unknown variable, which includes the trajectory of the user and landmark positions. Let measurement Z={z_i; i=1,2,..n} be a function of X as expressed in equation 3.8.

$$z_i = h_i(X_i) + \in_i \tag{3.8}$$

where $X_i \subseteq$ X is a subset of the measurement model h_i. $\in$ is random noise. The unknown variable X is now estimated by computing assignment $\hat{X}$, which have a maximum of the posterior p(X|Z).

Let p(Z|X) be the likelihood of Z given assignment X. Let p(X) be a prior probability over X. Using Bayes theorem, the aforementioned can be mathematically expressed as shown in equations 3.9, 3.10, and 3.11.

$$\hat{X} = \underset{X}{\operatorname{argmax}}\, p(X|Z) \tag{3.9}$$

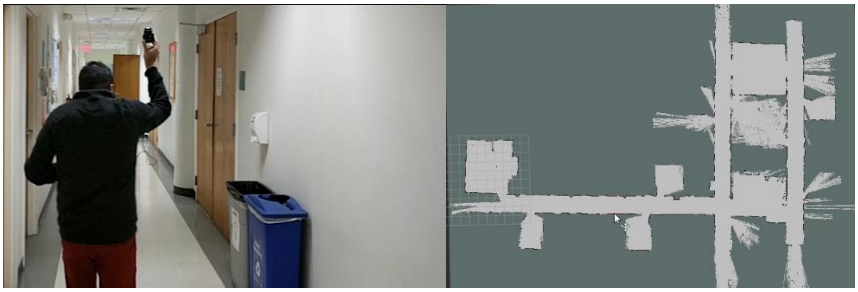
$$\hat{X} = \underset{X}{\operatorname{argmax}}\, p(Z|X)\, p(X) \tag{3.10}$$

$$\hat{X} = \underset{X}{\operatorname{argmax}}\, p(x)\ \ p\prod_{i=1}^{n}(z_i|X_i) \tag{3.11}$$

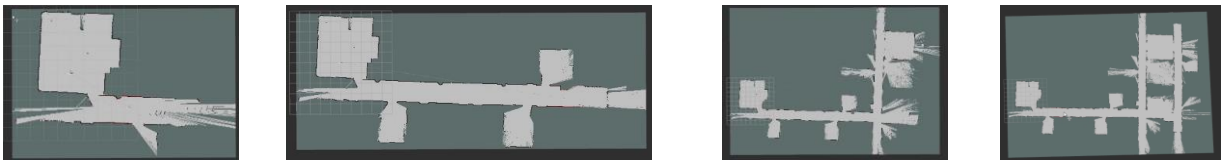
Let measurement noise $\in$ be a zero-mean Gaussian noise with inverse covariance matrix ψ_i. Then the equation further factorizes to a non-linear least square equation, as shown in 3.12.

$$\hat{X} = \underset{X}{\operatorname{argmin}} \sum_{i=0}^{n} \|h_i(X_i) - z_i\|_{\psi_i}^2 \tag{3.12}$$

With an initial prediction for the measurement as $\hat{X}$ and a quadratic cost function, the unknown measurement is solved using successive linearization [195].



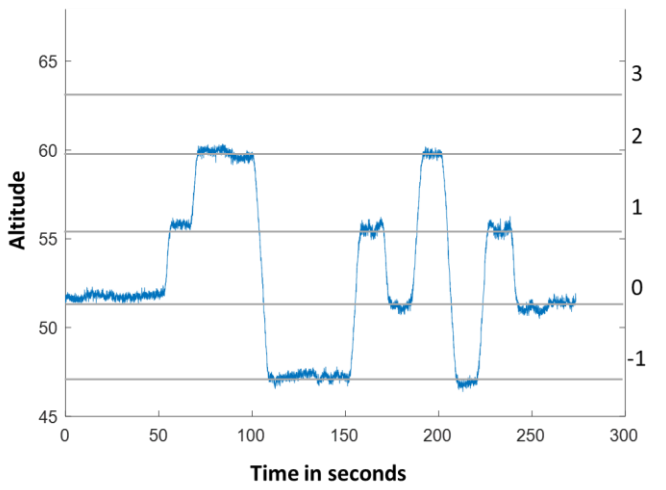
(a)



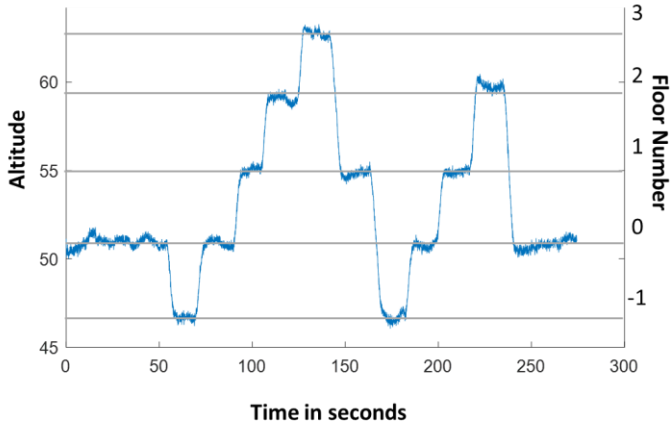
(b) (c) (d) (e)

Figure 3.12: A visualization of the dynamic real-time mapping experiment conducted at Tufts University. (a) the left image shows our researcher traversing through the building with the device in hand, and the right image displays the generated map of the floor. (b), (c), (d), and (e) are snippets of a time-lapse that show the progress of building the map in real-time as the user traverses the building.

Along with SLAM estimates, constant barometric readings estimate the building level. Although one of the pitfalls of using an environmental sensor is its sensitivity to environmental conditions such as wind gusts, these influences are minimized using the acceleration and magnetometer data. Furthermore, a thresholding system is utilized to detect a change in altitude based on the measurements of the barometric sensor. Let B(t) be a barometric measurement collected at time t. Then, the first derivative of the barometer readings B(t), is used to recognize a floor change.



Start	End	Status	Floor
0:00	1:05	Level	0
1:06	1:12	Elevator+	2
1:13	1:40	Level	2
1:41	1:30	Elevator-	-1
1:31	2:30	Level	-1
2:31	2:40	Elevator+	1
2:41	2:50	Level	1
2:51	2:53	Elevator-	0
2:54	3:06	Level	0
3:07	3:10	Elevator+	2
3:11	3:19	Level	2
3:20	3:30	Elevator-	-1
3:31	3:42	Level	-1
3:43	3:47	Elevator+	1
3:48	3:58	Level	1
3:59	4:02	Elevator-	0
4:03	4:33	Level	0



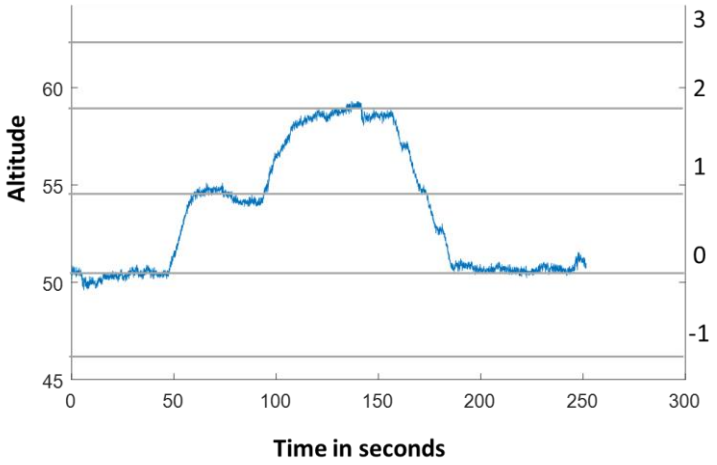
Start	End	Status	Floor
0:00	0:54	Level	0
0:55	1:03	Elevator+	2
1:04	1:11	Level	2
1:12	1:18	Elevator-	-1
1:19	1:28	Level	-1
1:29	1:40	Elevator+	1
1:41	1:46	Level	1
1:47	1:57	Elevator+	2
1:58	2:07	Level	2
2:08	2:13	Elevator+	3
2:14	2:24	Level	3
2:25	2:34	Elevator-	1
2:35	2:44	Level	1
2:45	2:56	Elevator-	-1
2:57	3:04	Level	-1
3:05	3:11	Elevator+	0
3:12	3:21	Level	0
3:22	3:29	Elevator+	1
3:30	3:38	Level	1
3:39	3:46	Elevator+	2
3:47	3:56	Level	2
3:57	4:05	Elevator-	0
4:06	4:35	Level	0

Figure 3.13: Floor-change activity determined using barometric readings. Samples were collected while the user was traversing through an elevator. This shows the significance of identifying the altitude and floor level.

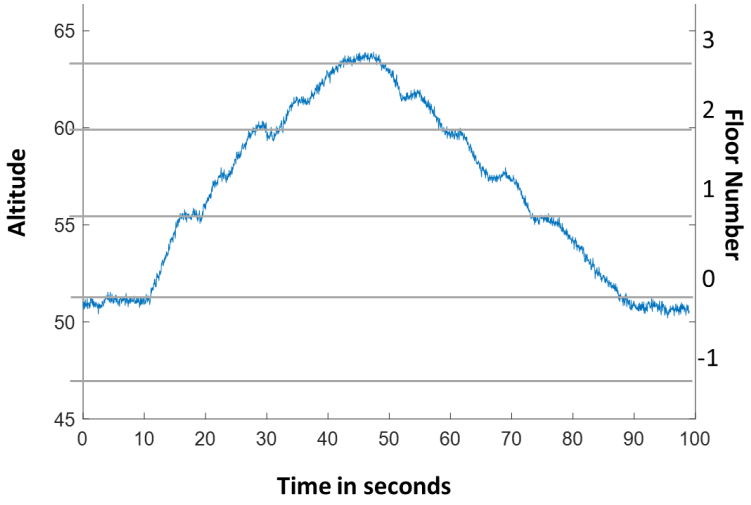
To evaluate the complete dynamic mapping and localization system, a user with the prototype walked in an indoor environment (Halligan Hall and Anderson Hall/SEC, Tufts University). The user further utilized the stairs and elevator to traverse through the building. The system was further used to wirelessly track the user throughout the experiment in real-time from a separate command center. This system was further tested in poorly illuminated areas successfully. The visualization

of the LIDAR SLAM outputs is shown in Figure 3.12. In this figure, (a) a researcher is holding the lidar sensor and moving around the campus. The data is wirelessly transmitted to the host laptop. Using SLAM technology, a real-time 2-D map is generated. This can be visualized through panels

Figure 3.12 [b-e]. The mapping visualization is performed using robot operating software (ROS).



Start	End	Diff	Status	Floor
0:00	0:21	0:21	Level	0
0:22	0:28	0:06	Stairs+	1
0:29	1:01	0:32	Level	1
1:02	1:16	0:14	Stairs+	2
1:17	1:56	0:39	Level	2
1:57	2:10	0:13	Stairs-	1
2:11	2:24	0:13	Stairs-	0
2:24	3:14	0:50	Level	0



Start	End	Status	Floor
0:00	0:11	Level	0
0:12	0:19	Stairs+	1
0:20	0:21	Level	1
0:22	0:27	Stairs+	2
0:28	0:31	Level	2
0:32	0:44	Stairs+	3
0:45	0:50	Level	3
0:51	1:01	Stairs-	2
1:02	1:04	Level	2
1:05	1:16	Stairs-	1
1:17	1:20	Level	1
1:21	1:30	Stairs-	0
1:31	1:42	Level	0

Figure 3.14: Floor-change activity determined using barometric readings. Samples were collected while the user was traversing through the stairs.

The floor-change activity can be visualized in Figure 3.13 and Figure 3.14. First, the researcher was asked to traverse the building using the elevator. The ground truth data were recorded simultaneously. Figure 3.13 shows that the altitude levels extracted from the barometric sensor are on par with the ground-truth data for floor level detection using elevators. Similarly, the plots in Figure 3.14 show that the altitude levels extracted from the barometric sensor are on par with the ground-truth data for floor level detection using stairs

3.4 Discussion

Figure 3.15 provides a general architecture of the proposed effort for the future first-responder utility system. The system includes collecting disparate sources of information, combining them into meaningful information, extracting vital features, analyzing the data, and providing outputs to the required users through different mediums.

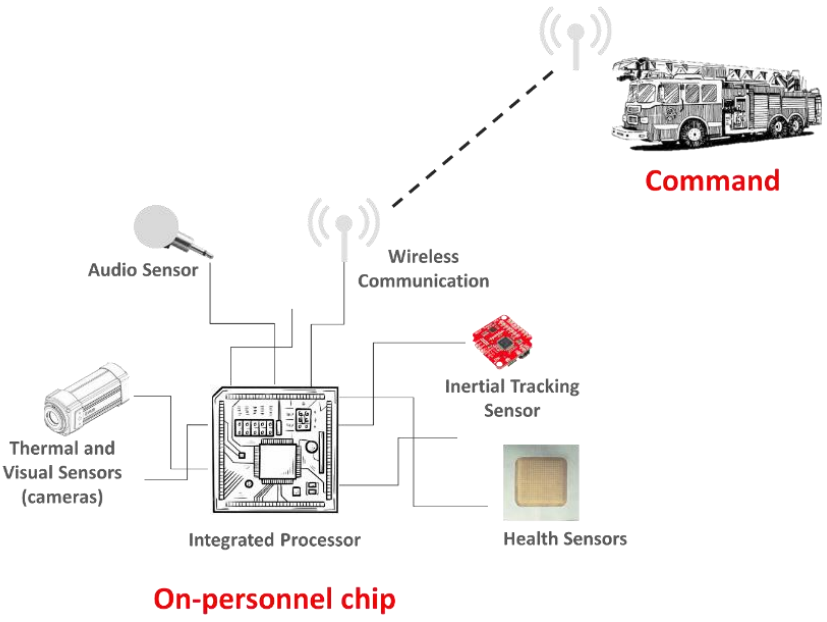


Figure 3.15: Schematic diagram of the proposed approach. The command center holds the main communication and processing system. Each user will have an on-personnel chip with one or more of the above components.

The system will transmit tracking information (Bytes to Kilobytes (KB)), Thermal (KB), visual images (MB), vitals (KB), and audio (KB) using the wireless communication technology from the firefighter to the Command. Command will monitor where the firefighters are, what they see and hear, and the on-the-go maps.

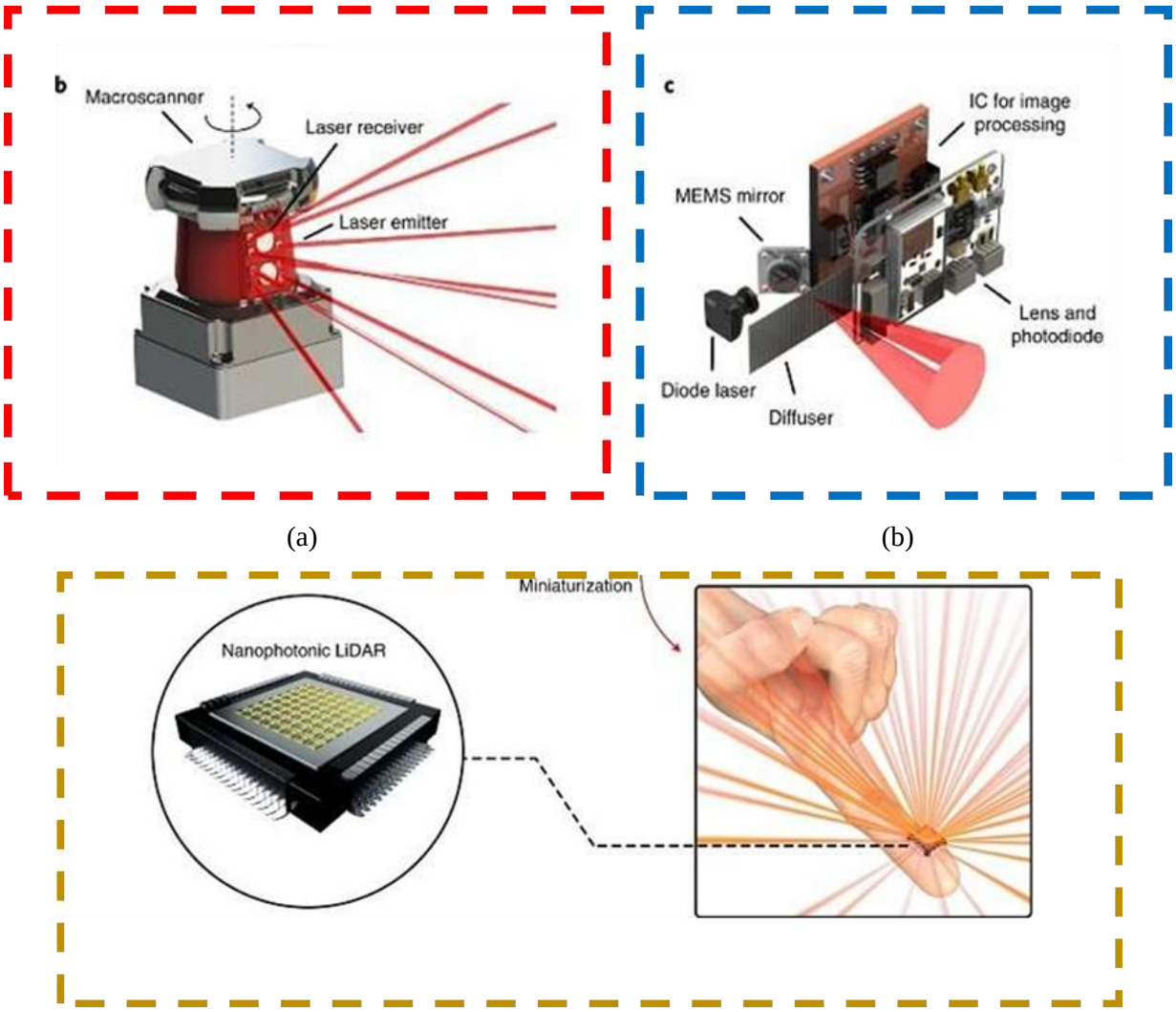


Figure 3.16: (a) Shows the LiDAR used in the study, (b) Future work will transition to smaller, non-rotating LiDARs, (c) Nanophotonic LiDARs [206] are still in development, and it will be the ideal hardware for this project.

With respect to hardware, the main challenge for research implementation lies in the cost and size of current LiDAR devices. They cost in the order of $ 500 to $ 10,000. Despite advancements in

performance, displacing current first-responder and defense methodologies will necessitate the development of new and inexpensive solutions. The advancement of miniature LiDAR, as shown in Figure 3.16, could lead to new smaller devices with widespread usage expected in industries such as autonomous driving, mobile phones, mixed reality, defense, and first-responder technologies.

3.5 Conclusion and Next Steps

This work presented a method for generating 3D traversable paths for 3D navigation. This information will create an iteratively refined 3D landscape knowledge database that can assist personnel in navigating novel environments or assist in mission planning for other operations. This research will assist military personnel and first responders in visualizing unknown terrain using AR and VR devices from any location with cardinal directions, walkable surfaces, and path-planning agents that will automatically generate paths to the desired location on the map.

Further research led to developing a head-worn augmented reality (AR) vision-aided indoor navigation system and prototype (INS-1) that does not rely on a GPS signal to locate and navigate the user. The proposed method uses computer vision to determine a user's location and then provides relevant augmented reality content based on that location. However, this system requires prior knowledge of the map and the image targets. To solve this, a system, method, and prototype for advanced INS by fusing multi-modal data such as LiDAR, IMU, and Barometric sensors for dynamic mapping and localization of users were developed (INS-2). The fusion of the sensors solved the challenges of horizontal and vertical mapping in multi-level buildings. The LiDAR sensor produces accurate horizontal maps of the region, the IMU device localizes the movement of the sensor, and the barometric sensor localizes the floor level of the user. This system overcomes

the limitations of INS-1 by dynamically creating a map in real-time and thus can be used in fields such as fire-fighting and defense.

While this chapter developed novel tools that can be used for navigation, security, and virtual reality, there is a need to introduce low-parameter, high yielding, and robust intelligent systems in these applications.

3.6 Acknowledgment

This work is supported by the Last Call Foundation, MA (Award 103248-00001). The author would like to thank the Last Call Foundation, the Boston Fire Department, the Medford Fire Department, and the Malden Fire Department for their valuable input and cooperation. The views and conclusions in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Fire Department or the U.S. Government.

Chapter 4: Feature Transverse Network for Thermal Segmentation and Gaze Estimation

Abstract

Visibility, usually associated with human vision, is generally limited to optical (visible) sensors. However, thermal imaging is superior to visible imaging with respect to its ability to operate in darkness and tolerate illumination variations. Unfortunately, current state-of-the-art image semantic segmentation methods (i) mainly concentrate on visible spectrum images, which do not adequately capture the context of corresponding pixels, particularly edge details in thermal images, and (ii) accept a trade-off between higher accuracy and lower speed, or vice-versa.

To solve these challenges, a novel end-to-end trainable convolutional neural network architecture, feature transverse network (FTNet), has been proposed. Extensive experiments were conducted to show the robustness of FTNet in thermal image segmentation. It was further extended to human gaze estimation for real-time applications.

Index terms: convolutional neural network, FTNet, semantic segmentation, thermal segmentation, edge-guidance, transverse network.

Publications

Panetta, Karen, KM Shreyas Kamath, Srijith Rajeev, and Sos S. Agaian. "FTNet: Feature Transverse Network for Thermal Image Semantic Segmentation." IEEE Access 9 (2021): 145212-145227.

Rajeev, Srijith, Shreyas Kamath KM, Abigail Stone, Karen Panetta, and Sos S. Agaian. "Gaze-FTNet: A feature transverse architecture for predicting gaze attention" Multimodal Image Exploitation and Learning 2022, International Society for Optics and Photonics, 2022 (Abstract accepted).

Image segmentation is the process of partitioning images into multiple segments [10], and it is one of the most challenging tasks in computer vision. It paved the way towards scene understanding, whose importance is highlighted by the fact that an increasing number of applications nourish from inferring knowledge from imagery, including autonomous driving [207-209], computational photography [117, 210], biomedical analysis [211, 212], and augmented reality [42, 213-216].

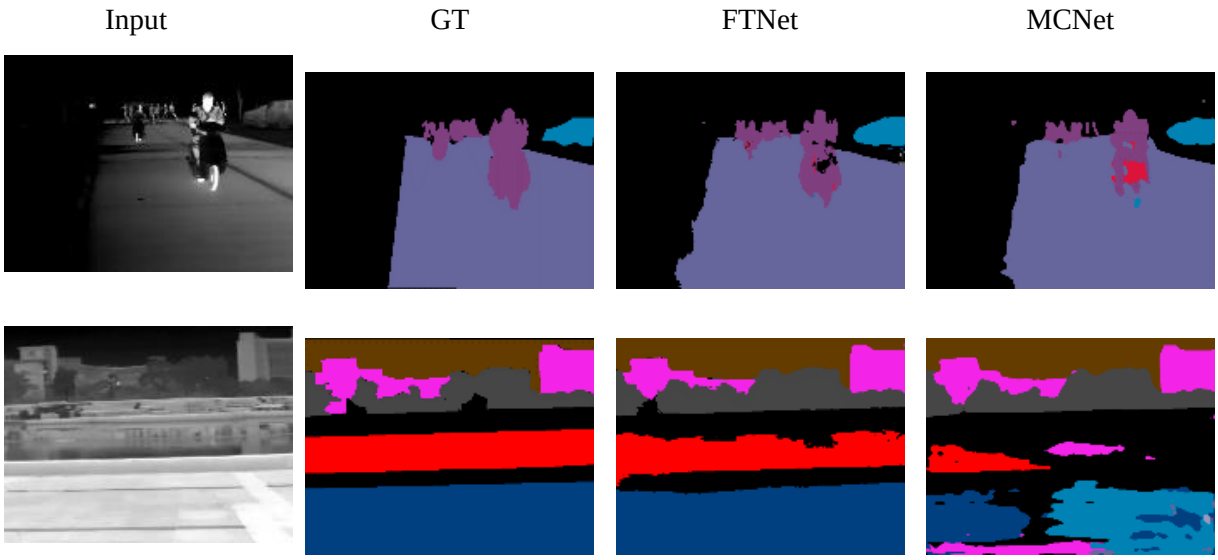


Figure 4.1: Examples demonstrating the effectiveness of the FTNet's ability to reconstruct semantic maps with higher accuracy and crisper edges. Column (a) Input thermal images, (b) Ground truth semantic maps, (c) Results produced by FTNet, and (d) Results generated by the state-of-the-art network- MCNet.

Semantic image segmentation is a high-level task formulated as a classification problem of pixels with semantic labels [215]. Semantic segmentation algorithms identify regions of different objects in the scene by grouping parts of the image together based on the same object of interest and assigning a label to each pixel of an input image. In contrast, instance segmentation treats multiple objects of the same class as distinct individual instances. Panoptic segmentation assigns two labels to each pixel of an image, namely a semantic label and an instance id. The identically labeled pixels belong to the same semantic class, and for instance, their ids distinguish their instances.

Despite significant advancements, semantic image segmentation is still considered a challenging task due to the adverse environmental conditions caused by imaging limitations of the visible spectrum. For instance, visible cameras are susceptible to lighting conditions and become invalid in total darkness. Furthermore, their imaging quality decreases significantly in adverse environmental conditions, such as rain and smog [215].

Thermal imaging is a process of utilizing infrared radiation and thermal energy to gather information about objects. It is superior to visible imaging because it operates in darkness and across illumination variations. It can penetrate smoke, aerosol, dust, and mist [217]. The global thermal imaging market for the mobility industry is expected to reach $3.22 billion by 2025 [218]. The market's growth can be attributed to growing awareness and lower prices of thermal cameras. This has led to their application in many computer vision tasks, such as detection [219, 220], tracking, segmentation [221-224], and individual and emotion identification [39, 225-227]. Thermal image-based computer vision will be instrumental in improving driver-assist systems that are increasingly quintessential in consumer car models.

These sensors offer additional information to existing autonomous driving sensory systems, which strive to improve performance in identifying objects within a vehicle's surroundings to enhance driving reactions. However, the inter-class variance of objects in thermal images is extremely low, making accurate labeling near boundaries difficult, resulting in a large amount of semantic ambiguity and intensifying the challenge of semantic segmentation. The state-of-the-art (SOTA) semantic segmentation approaches focus on diminishing semantic ambiguity using the rich context information of visible images. However, redundant and noisy semantic information from thermal images may clutter the final semantic maps.

Convolutional Neural Networks (CNN) are among the most effective and widely used deep learning (DL) architectures in computer vision, including classification, detection, and segmentation. Semantic segmentation models usually follow an encoder-decoder architecture. A deep convolutional neural network (CNN) typically computes a feature hierarchy layer by layer in the encoder stage. It develops an inherent multi-scale pyramid shape. At the decoder end-stage, a high-semantic feature map is up-sampled and fused with the previous layer feature map through lateral connections to recover higher spatial dimensions. After extracting spatial details, the network predicts the class for each pixel to complete the segmentation process.

However, this progress has come with a voracious appetite for computing power which will rapidly become technically and economically prohibitive [31]. This section presents a novel end-to-end trainable convolutional neural network architecture named feature transverse network (FTNet) to address these issues. The proposed FTNet network will be designed and optimized to perform image segmentation of thermal images. FTNet consists of two main components: a high-low feature traversing and an edge guidance part. The architecture is equipped with skip connections between these two networks to use high-resolution image details during the reconstruction. An example of the results obtained using FTNet is illustrated in Figure 4.1.

Some of the notable contributions of FTNet include:

1) a unified end-to-end trainable network that captures discriminative thermal image features from multiple resolutions and combines them in a fully connected approach;

2) a network that captures and optimizes feature representation at the multi-scale resolution, thereby improving the capability of handling high-resolution images and producing quality output at a lower computational cost;

3) a network whose main representations are shared between the semantic segmentation and edge guidance structures, which means that the FTNet simultaneously achieves semantic segmentation and edge detection without significantly increasing the model complexity;

4) an extensive computer simulation performed on challenging thermal semantic segmentation tasks on benchmarks datasets, including SODA [215] and MFNet [228], which validate the performance of the proposed model compared with state-of-the-art methods such as MCNet [229], PSPNet [230], DeepLabv3 [118], and HRNet [231];

5) open-source the architecture, code-base, and model weights for the research community.

This chapter is organized as follows. In section 4.1, the recent related literature is reviewed. A detailed description of the FTNet architecture and its analysis is provided in section 4.2. Section 4.3 presents the experimental results, including training details, ablation studies, and benchmark results. Finally, a brief discussion and conclusion are provided in sections 4.4 and 4.6, respectively.

4.1 Related Work

With significant performance improvements on popular benchmarks, deep-learning architectures have shifted the paradigm in the field of segmentation. This section provides an overview of some of the most prominent deep learning (DL) architectures for visible and thermal image semantic segmentation for the computer vision community. Table 4.1 provides a summarized list of various image segmentation methods and a brief explanation for each method.

Before the advent of deep learning, computer vision and machine learning techniques such as thresholding [232-234], histogram-based bundling, region growing [235], k-means clustering [236], watersheds [237], active contours [238], graph cuts [239], Principal Component Analysis

(PCA), Independent Components Analysis (ICA), and Canonical Correlation Analysis (CCA) were used

Table 4.1: A literature review of the state-of-art techniques for image semantic segmentation.

Author	Explanation
PSPNet [230]	This network employed a pyramid parsing module that utilized global context information by different region-based context aggregation. The local and global clues together made the final prediction more reliable.
DeepLabv3 [118]	This network aims at solving the problem of segmenting objects at multiple scales. It employed atrous convolution in cascade or parallel to capture multi-scale context by adopting multiple atrous rates.
UNet++ [240]	This network connected the encoder and decoder sub-networks through nested, dense skip pathways. The re-designed skip pathways aim at reducing the semantic gap between the feature maps of the encoder and decoder sub-networks.
ENCNet [119]	This method introduced a context encoding module, which captured the semantic context of scenes and selectively highlighted class-dependent feature maps.
HRNet [231]	This method overcomes the low-resolution representations by utilizing a high-resolution convolution stream throughout the network. It gradually adds high-to-low resolution convolution streams and connects multi-resolution streams in parallel.
MCNet [229]	This method captured the inter-class and intra-class contextual dependencies using a multi-level correction process. In addition, prior edge knowledge was combined with the context information to obtain the final feature representations.

for segmentation. However, the introduction of a Fully Convolutional Network (FCN) [241] for effective pixel-level classification changed the landscape of computer vision techniques. Convolutional layers are used in place of the final fully connected layer in FCN. With an arbitrary size input and effective reasoning, it could produce an output of the corresponding size. However, it did not make effective use of the global context information. U-Net [242], another typical segmentation network with the concatenation of multi-scale features, was initially proposed for biomedical image segmentation. Deconvolution and unpooling networks were proposed to solve the challenges of FCN.

Lin et al. [68] pioneered Global Average Pooling (GAP) in CNN models, which replaces the traditional fully connected layers. GAP calculates the mean value for each feature map without requiring additional model parameter training [243]. In another approach, Chen et al. [244] and Yu et al. [245] introduced dilated convolutions to capture intrinsic sequence information by expanding the convolution kernel's field without increasing the model's parameters.

In addition to the aforementioned networks, many practical deep learning techniques such as FPN [246, 247], PANet [248], and NASFPN [249] were introduced to improve the effectiveness of segmentation algorithms. Li et al. [215] developed a gated feature-wise transform layer for adaptively embedding edge information as guidance for a semantic segmentation network in the thermal image segmentation domain. This network extracted edges using the HED (Holistically nested Edge Detection) network [250] and embedded them into a network proposed by [118]. Xiong et al. [229] developed a method for segmenting thermal images semantically that used multi-level edge knowledge to obtain additional edge and shape features. Ha et al. [228] segmented images using RGB and thermal information. This network used two distinct encoders to extract features from visible and thermal images. It then combined them in the decoder to generate the probability map for the semantic segmentation results. Additional techniques utilizing RGB and thermal images are described in [251-253].

4.2 Proposed Method

This section presents an end-to-end trainable convolutional neural network architecture called the feature transverse network (FTNet). A high-level flow diagram of the proposed system is provided in Figure 4.2. This chapter aims to construct a function $f(I)$ developed specifically to link each pixel in an image, where I is an input image of any arbitrary size (m, n) to a class label with the same dimension. This network combines the low-level layers with poor semantic features and

strong resolution with the high-level layers with rich semantic features and low resolution. Following this theme, the novel FTNet comprises an encoder network, a corresponding transverse decoder network, and a final pixel-wise classification layer. This network aims at capturing and optimizing feature representation at the multi-scale resolution, thereby improving the capability of handling high-resolution images and producing quality output at a lower computational cost than the SOTA techniques. Additional details of these components are provided in further subsections.

4.2.1 Encoder Network

Since the main focus of the proposed network is to build a position-sensitive model capable of pixel-level classification, FTNet employs existing SOTA backbones that follow the design rule of LeNet-5 [254]. The spatial size of the features in these classification-based backbones is gradually reduced from a high-level representation to a low-level representation, thereby allowing FTNet to capture features with different representation capabilities. The basic structure of the encoder network is visualized in Figure 4.2 (a).

For simplicity of exposition, consider the case in which ResNet50 is used as a backbone. The encoder network aims at acquiring features at different resolutions by subsampling at various stages. The spatial resolution of each stage is given in Table 4.2.

As shown, the convolutions in these networks can be divided into four stages, and the output of each stage's last block from the encoder network can be represented as $\{E_i | i = 1,2,3,4\}$. This bottom-up pathway extracts and establishes a feature pyramid by incorporating features from the last convolutional layer in each stage. The extension to other backbones is similar.

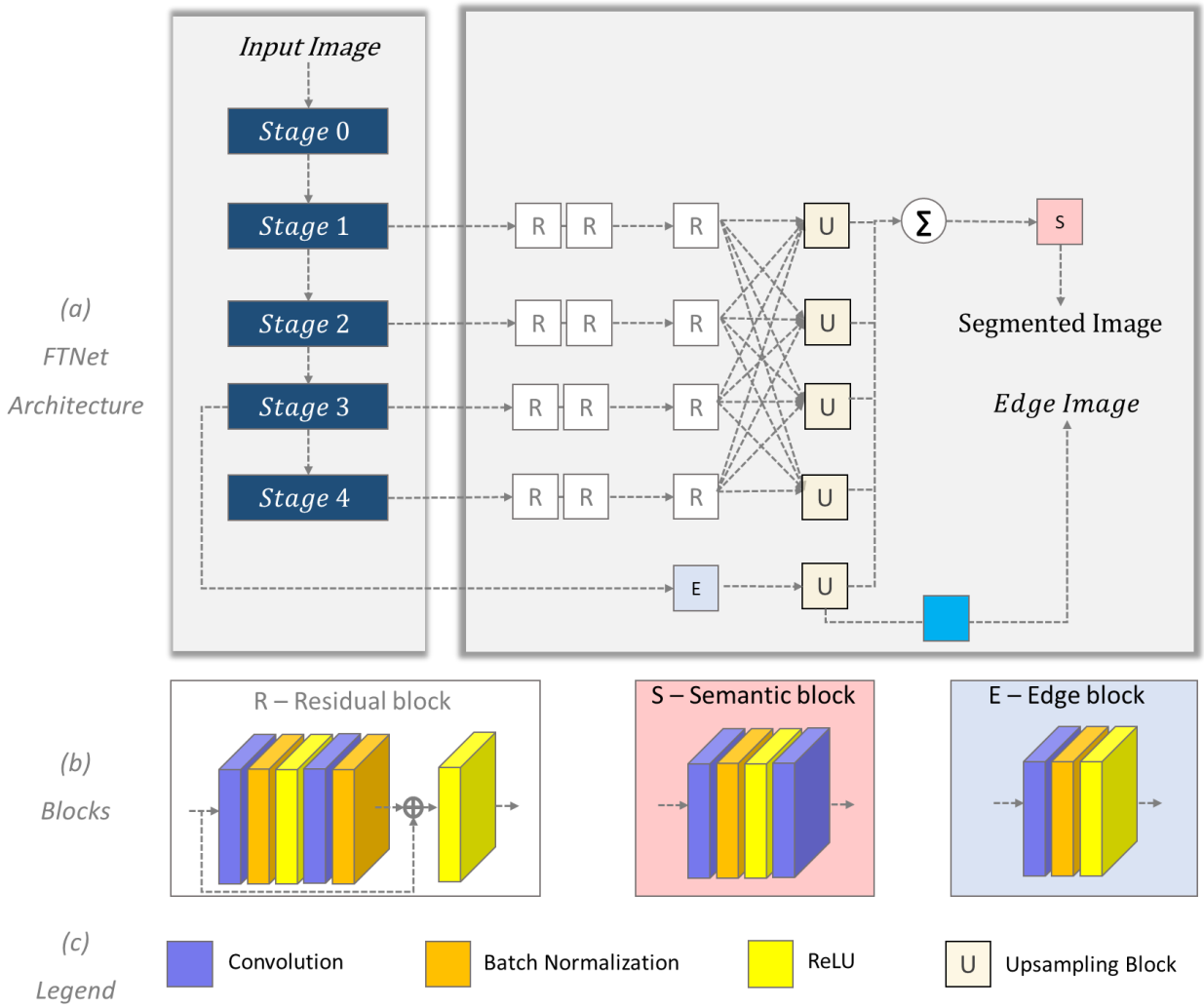


Figure 4.2: Shows the network architecture of the proposed Feature Transverse Network (FTNet), where (a) provides the FTNet Architecture. It includes an encoder-decoder structure. The decoder branches are connected transversely to one another, which are up-sampled along with edge guidance to produce a high-resolution semantic map. (b) Provides the residual, semantic, and edge block. (c) Provides the legend of each block.

Table 4.2: The spatial resolution of the encoder structure used in the FTNet architecture. H and W refer to original image dimensions.

Image Size	Encoder Stage 0	Encoder Stage 1	Encoder Stage 2	Encoder Stage 3	Encoder Stage 4
[H, W]	$\left[\frac{H}{2}, \frac{W}{2}\right]$	$\left[\frac{H}{4}, \frac{W}{4}\right]$	$\left[\frac{H}{8}, \frac{W}{8}\right]$	$\left[\frac{H}{16}, \frac{W}{16}\right]$	$\left[\frac{H}{32}, \frac{W}{32}\right]$
E_0	E_1	E_2	E_3	E_2	E_4

4.2.2 FTNet Decoder

Recent developments have demonstrated the evidence for the necessity of exploring all the multiple-resolution representations for a broad range of vision problems [231]. Following this, a transverse network with a top-down path comprising feature aggregation to produce final semantic features at high resolution is introduced. An illustration of the proposed decoder network is displayed in Figure 4.2 (c).

The proposed FTNet considers the features from the stages $\{E_i | i = 1,2,3,4\}$, which comprises multiple resolutions $\mathfrak{r} = 1/4, 1/8, 1/16,$ and $1/32$, respectively. These feature maps are passed through residual blocks $\mathfrak{U}$, as proposed by He et al. [255]. The illustration of this unit is provided in Figure 4.2 (b). Each residual block can be defined as formulated in equation 4.1.

$$\Psi_{l+1} = \mathbb{R}(\omega_s(I(\Psi_l)) + \Omega(\omega_l * \Psi_l + b_l)$$
$$|\{\omega_l = [\varpi_{l,k} : k = 1 \leq k \leq K]\} \quad (4.1)$$

where Ψ_l is the input feature map for the l^{th} residual layer, ω_l and b_l are the associated set of weights and biases, respectively, K denotes the number of weights, Ω denotes the combination of layers CONV → BN → ReLU → CONV → BN, $\mathbb{R}$ denotes the ReLU activation function, and I is the identity map that may comprise of weight ω_s when the features maps do not have the same number of channels.

A set of residual blocks along each resolution form a residual stream. The output of these residual streams can be formulated as $\{\Phi_i | i = 1,2,3,4\}$. These streams are fused in a fully connected fashion to take advantage of information exchange across multi-resolution representations. The integration of multi-resolution features maps $\mathfrak{r}(m)$ at the i^{th} stage is a summation of different feature maps with a corresponding function $f(\cdot)$. A broad formulation of this system can be defined

as shown in equation 4.2. The function $f_{ij}(\cdot)$ is dependent on feature resolutions. It can be formulated as shown in equation 4.3.

$$\mathfrak{D}_i = \sum_{j=0}^{4} f_{ij}(\Phi_{ij}) \mid i = 0,1,2,3,4 \tag{4.2}$$

$$f_{ij}(\cdot) = \begin{cases} \Phi_{ij}, & if\ i = j\ or\ r_i = r_j \\ \downarrow (\Phi_{ij}), & if\ (i < j)\ and\ r_i < r_j \\ \uparrow (\Phi_{ij}) & if\ (i > j)\ and\ r_i > r_j \end{cases} \tag{4.3}$$

To downsample ($\downarrow$) by a factor of 4, two strided convolutions with kernel size 3×3 are utilized. For upsampling ($\uparrow$), a resize convolution with a bilinear kernel and a convolutional layer of kernel size 1×1 are utilized. No function is applied when the input and output feature maps' resolutions are identical and along the same residual stream. Finally, all the feature maps are upsampled to the original resolution and passed through Θ, which comprises CONV → BN → ReLU → CONV with convolution kernel size 1×1.

4.2.3 Edge Guidance (EG)

The low resolution of thermal images results in very coarse features with little distinction between the boundaries of objects, which results in a thermal crossover. This chapter introduces an Edge Guidance (EG) mechanism to solve these challenges. The mechanism tries to regulate the quality of the edges produced by extracting features from the encoder, passing it through a convolution, batch normalization, and Leaky ReLU combination. It is regulated using an edge loss function defined in the next section. Due to these concerns, the reconstruction of the semantic maps generally depends on low-level features and edges details. The ground truth for the edges was obtained by determining the edges of the annotated edge maps. The clear distinction of the maps produces high-quality edge maps.

Considering these observations, edge map detection is introduced in the decoder section. The edges are extracted from the E_3 layer and passed through CONV → BN → ReLU. This is upsampled to the original resolution, passed through a CONV layer, and finally appended to the feature maps before applying the function Θ. It should be noted that the edge ground truth is obtained by a simple calculation of the semantic ground truth gradient, which does not require additional labeling effort. A detailed study of edges extracted from various encoder parts is provided in further sections.

4.2.4 Loss Function

An edge-based loss function is employed to ensure the prediction of crisp edges along the boundaries of semantic maps. In edge detection cases, the labels for edges and backgrounds are highly imbalanced. A binary cross-entropy with an adaptive balancing mechanism proposed by Xie and Tu [250] is utilized to overcome this issue. For an image with a ground truth that comprises of Z_+ edge pixels and Z_- background pixels, the prediction $\tilde{p}$ can be formulated as shown in equation 4.4.

$$\mathcal{L}_{edge}(\tilde{p}) = -\frac{|Z_-| \sum_{i \in Z_+} \log(\tilde{p}_j)}{|Z_+ \cup Z_-|} - \frac{|Z_+| \sum_{i \in Z_-} \log(1 - \tilde{p}_j)}{|Z_+ \cup Z_-|} \quad 4.4$$

This loss function handles the class imbalance by providing equal weights irrespective of the ratio of Z_+ and Z_- between the two classes.

A cross-entropy loss defined in equation 4.5 is utilized to supervise the semantic maps generated.

$$\mathcal{L} = -\frac{1}{N} \sum_{m}^{N} y_m \log(\Gamma_\eta(x_m|z) + (1 - y_m) \log(1 - \Gamma_\eta(x_m|z) \quad 4.5$$

where $\Gamma_\eta(x_m|z)$ denotes the probability at pixel m with the parameter η, y_m is the ground truth. The total loss adapted to train FTNet is denoted as:

$$\mathcal{L}_{total} = \alpha\,\mathcal{L}_{edge} + \beta\,\mathcal{L} \qquad 4.6$$

where α and β are continuous hyper-parameters and denote the weights for edges and semantic loss, respectively.

For experimental results, β was fixed to 1, while the α were varied. More discussions are provided in the later sections. This configuration helps obtain refined, spatially consistent, and crisp boundary-located semantic maps.

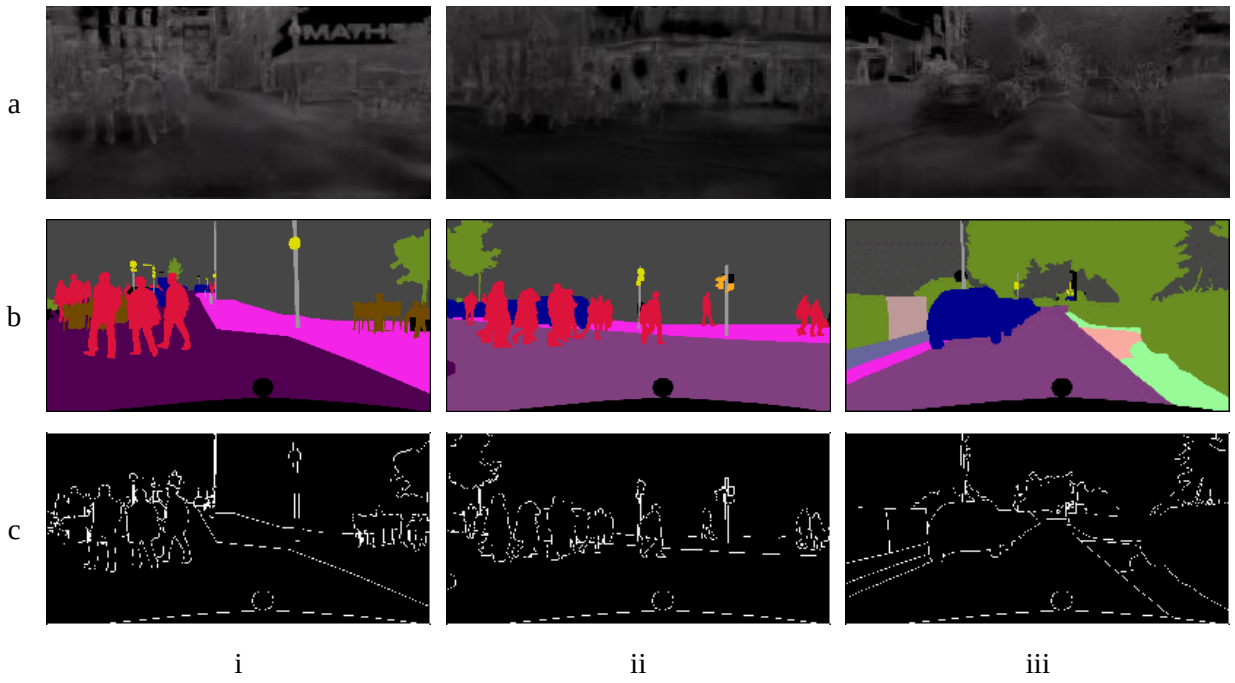


Figure 4.3: Illustration of thermal images with corresponding masks from the Cityscapes dataset. Rows [a-c] show the thermal images with the corresponding semantic and edge maps. Columns [i-iii] comprise Cityscapes images converted to the thermal domain.

4.3 Experimental Results

This section provides the performance evaluation of the FTNet. After describing the experimental settings, datasets, and training details, the performance comparisons with SOTA methods are provided to demonstrate the effectiveness and generalization ability of the proposed FTNet.

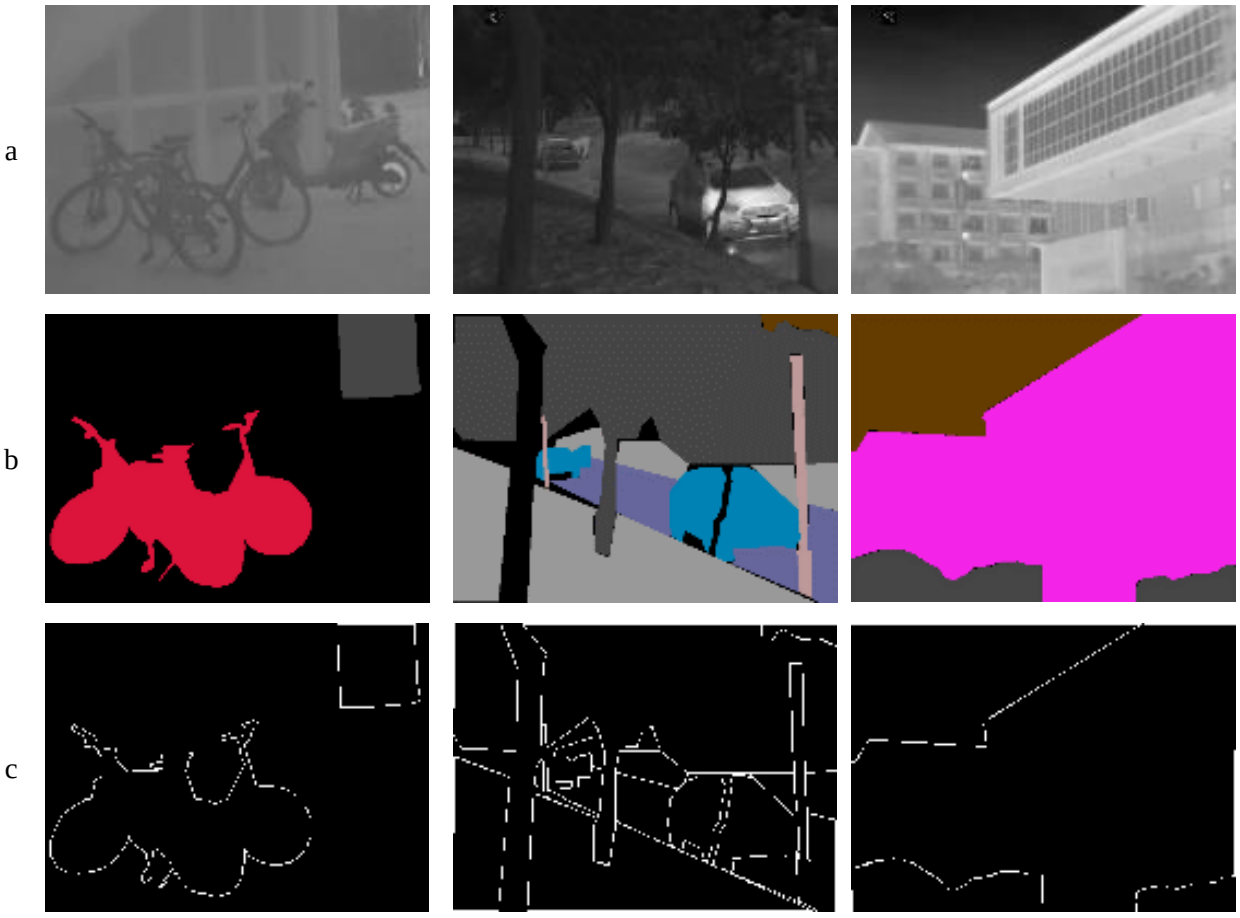


Figure 4.4: Illustration of thermal images with corresponding masks from the SODA datasets. Rows [a-c] show the thermal images with the corresponding semantic maps and edge maps. Columns [i-iii] contain images converted from the SODA dataset.

4.3.1 Dataset

The SODA dataset [215] was employed for training and testing purposes. It comprises 2,168 annotated images, among which 1,168 images are used for training, and 1,000 images are used for testing. Due to the scale of this dataset, synthetic images were utilized for pretraining the network. These synthetic images were obtained by translating the Cityscapes [256] dataset from RGB space to thermal space. An example of a set of images utilized for training is provided in Figure 4.4

The original Cityscapes dataset contained 5,000 images, including 2,975 training images, 500 validation images, and 1,525 test images. Following the training protocol described in [215], all the training, validation, and testing images were combined for pretraining purposes. The dataset does not comprise the ground truth edge maps, and they were generated following the strategy used in [257]. In this protocol, the ground truth semantic maps generate edges for each class. An example of a set of images utilized for pretraining is provided in Figure 4.3, and training is shown in Figure 4.4.

MFNet dataset is a public semantic segmentation dataset based on the driving scene with RGB-T images. It comprises 1569 images divided into 784, 392, and 393 images for training, validation, and testing.

4.3.2 Training Details

For training the network, a progressive learning algorithm is adopted. Initially, the encoder parameters are loaded with pre-trained ImageNet weights. For pretraining the model with thermal features, synthetic thermal Cityscapes images were utilized. The thermal input images were augmented by performing random cropping (from 640×480 to 480×480), random scaling in (0.5, 2), and random horizontal flipping along with the corresponding masks. The SGD optimizer [258] with a base learning rate of 0.01, momentum 0.9, and weight decay 0.0001 is employed to train the model. As the decoder section is randomly initialized, the learning rate of this network is increased by a factor of 10. The batch size was set to 16. The network was trained for 100 epochs, and a poly learning rate policy with a power of 0.9 was used to drop the learning rate.

As the Cityscapes dataset labels are different from the SODA dataset, the last layer of the network is discarded after pretraining and adjusted to match the number of classes from the SODA. The

protocol for training remains the same except for the initial learning rate, which is dropped by a factor of 10.

The experiments were conducted on the PyTorch platform [259]. The models were trained on 2 V100 GPUs, and it takes around 12 hours to complete FTNet training (Cityscapes pretraining and SODA training). To evaluate FTNet, Intersection over Union (IoU), also known as the Jaccard Index, is utilized. It provides the ratio of the intersection of the pixel-wise classification results with the ground truth to their union. A higher percentage depicts how close the predicted class maps are to the ground truth.

4.3.3 Ablation Studies

Extensive ablation studies on the various architectural components of FTNet architecture were performed. The ResNet 50 architecture was utilized as the encoder for baseline results. The number of filters in the decoder was set to 32, and the number of residual blocks $\mathfrak{U}$ for each residual stream Φ_i was set to 4. Equal weights ($\alpha = \beta = 1$) were provided to both edge and semantic maps for all the ablation studies except for loss weight analysis.

4.3.3.1 Impact of Encoder Topologies

This section aims at evaluating the performance of the FTNet decoder with various encoder architectures on the SODA dataset. The results of this study are provided in Table 4.3. It should be noted that ResNet and ResNeXt provided superior mIoU results when the filter sizes were set to 32 and 64. The ResNet and ResNeXt models were further investigated with 128 filter sizes and two residual blocks. The ResNeXt model using this setting outperformed other architectures from the ResNet. ResNeXt uses a much more parallel stacking layer structure rather than sequential layers when compared to ResNets. ResNeXt, like Inception, uses a split-transform-merge strategy,

but ResNeXt shares hyper-parameters, whereas Inception[260] uses a unique filter and size for each block. The inclusion of this cardinality helps achieve the encoder obtain results. This feature of the ResNeXt model is very effective and of central importance, in addition to the dimensions of width and depth.

Table 4.3: Impact of utilizing different encoders, filter size, and the number of residual blocks $\mathfrak{U}$ with the proposed FTNet.

Encoder	Filters	Residual blocks ($\mathfrak{U}$)	↑ mIoU(%)
ResNest - 50	32	4	55.16
	64	4	57.98
ResNet – 50 [255]	32	4	56.71
	64	4	58.82
	128	2	59.40
ResNeXt – 50 [261]	32	4	57.07
	64	4	59.45
	128	2	59.94
ResNet – 101 [255]	128	2	60.84
ResNeXt – 101 [261]	128	2	62.67

4.3.3.2 Impact of Edge Loss weights

As the defined loss from equation 4.6 comprises two components, namely, semantic loss and edge guidance loss, it is necessary to adapt them to the same order of magnitude to obtain optimum results. A small edge loss weight may lead to a failure of edge supervision, while a large weight may dominate the semantic loss. It is necessary to optimize them as semantic loss is always higher than edge loss. In this ablation study, different α values were used to empirically determine the best edge loss weight. Table 4.4 provides the complete set of variations and their corresponding mIoU scores.

When setting $\alpha = \beta = 1$, the boundaries are not crisp when compared to the results obtained with α = 20 and $\beta = 1$. This discrepancy explains that setting the magnitude of other losses is crucial for better accuracy. From the table, it can be determined that α = 20 provides superior accuracy. It should be noted that α = 0 corresponds to FTNet architecture without an edge guidance mechanism. The mIOU without EG is 55.50, while with EG and α = 20, the mIOU increased to 60.04 %. This is an 8.25 % increase in accuracy. It is also worth noting that the mIOU for all cases with EG increased, signifying the importance of edge guidance. These results clearly show both EG's impact and the EG's loss-weight.

Table 4.4: Analysis of hyperparameter- α weights in the loss function with β =1. FTNET - ResNeXt – 50 with 128 filter size and 2 residual blocks ($\mathfrak{U}$) was utilized for this ablation study.

$\boldsymbol{\alpha}$	↑ **mIoU (%)**
0	55.50
1	59.61
5	59.77
10	60.05
15	59.61
20	60.08

4.3.4 Benchmark Results

The performance was tested on the MFNet dataset [228] and SODA dataset [215] to demonstrate the generalization ability of FTNet. To show the effectiveness of the proposed FTNet, it is compared with SOTA methods such as PSPNet [230], HRNet [231], UNet++ [240], ENCNet [119], PSPNet [230], and MCNet [229]. For all these comparisons, FTNet with ResNeXt backbone was utilized. The hyperparameters used were filters=128, residual blocks=2, α = 20, and β =1.

Table 4.5: Evaluation results of semantic segmentation SODA and MFN datasets.

Model	Backbone	↓ Params (M)	↓ FLOPS (G)	↑ mIoU (%)	
				SODA Dataset	MFN Dataset
HRNet [231]	HRNet-32	29.54	53.33	58.33	46.48
DeepLabv3[118]	Dilated-ResNet- 50	39.84	56.96	58.37	45.74
ENCNet [119]	Dilated-ResNet- 50	33.71	163.87	56.00	44.17
PSPNet [230]	Dilated-ResNet- 50	46.79	207.99	58.87	46.51
UNet++ [240]	ResNet - 50	48.98	270.57	53.18	43.59
MCNet [229]	Dilated-ResNet- 50	35.67	131.87	50.32	42.28
FTNet –Filters 128, $\mathfrak{U}$=2, α = 20, β =1	ResNeXt – 50 [261]	33.44	94.55	**60.08**	**47.12**

Note: The number of params and flops are calculated for input size 640 x 480. **Bolded** numbers are higher in value.

Table 4.5 provides the mIOU accuracies for the SODA and MFN datasets. FTNet achieves 60.09% accuracy on the SODA dataset, while the top four SOTA methods were MCNet, HRNet, PSPNet, and DeepLabV3 with 50.32%, 58.33%, 58.87%, and 58.37%, respectively. Similarly, FTNet achieves 47.12% accuracy on the MFN dataset, while the top four SOTA methods were MCNet, HRNet, PSPNet, and DeepLabV3 with 42.28%, 46.48%, 46.51%, and 45.74%, respectively. These results demonstrate the performance of FTNet when compared to the SOTA methods.

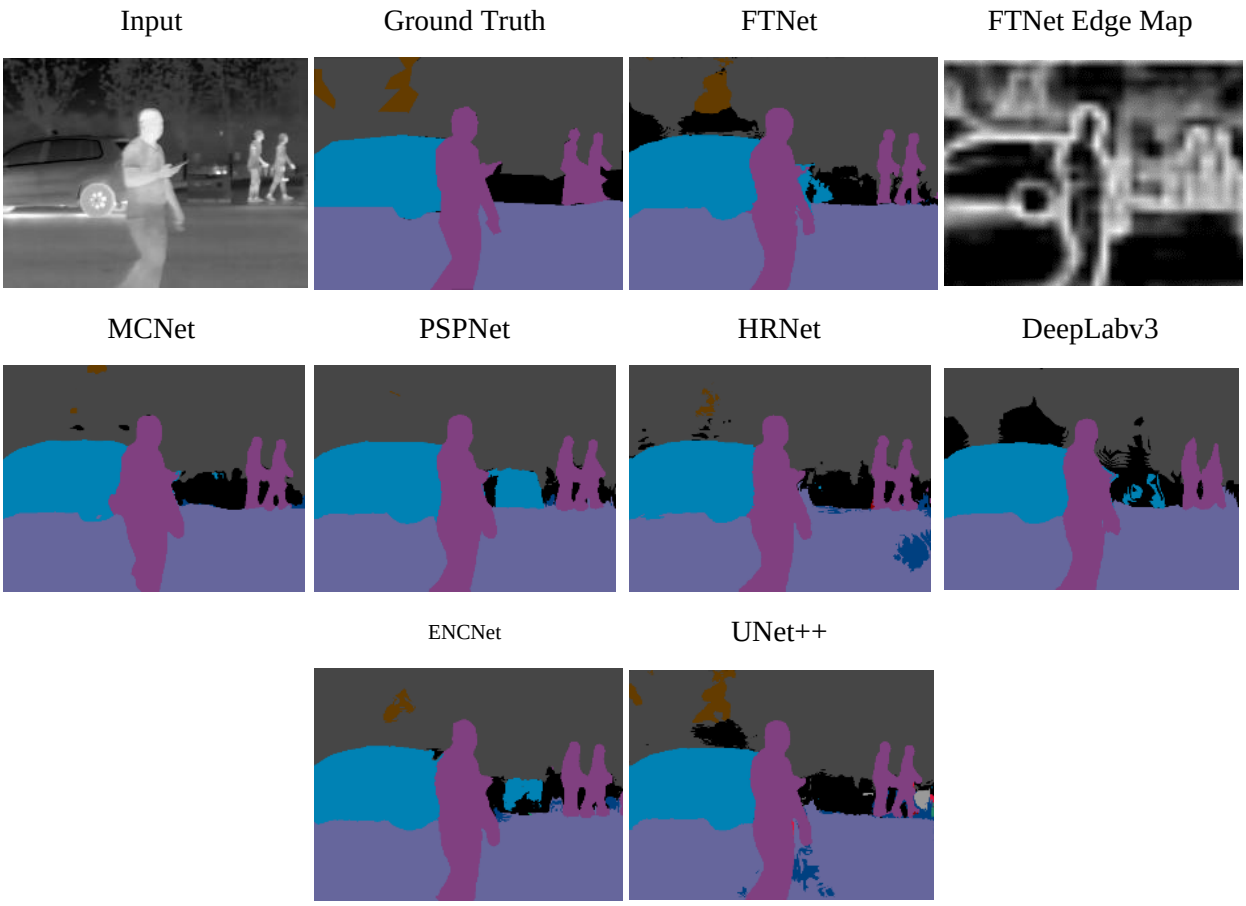


Figure 4.5: Qualitative comparison of SOTA methods for semantic segmentation of thermal images in indoor and outdoor environments. Along with the segmentation output, a reconstructed FTNet edge map is included. As can be seen from the images, FTNet has provided more defined boundaries than SOTA methods. The person class was more precisely segmented, whereas SOTA methods had an ambiguous map.

Qualitative evaluation is essential in image segmentation assessment, and the segmentation maps of FTNet and SOTA are assessed based on the human visual system. This analysis uses human observers [262-266] for subjective evaluation. Human visual analysis is critical in identifying characteristics of algorithms that quantitative metrics may not identify correctly. The qualitative comparisons are provided in Figure 4.5. These examples show complex indoor and outdoor scenarios with numerous object instances in multiple scales and partial occlusion. These scenes were also captured with diverse lighting conditions, including day and night. These outdoor

examples show challenging scenarios with various objects, such as cars and pedestrians, in close proximity to each other and far apart. FTNet effectively addresses these challenges and yields reliable semantic maps. In panel (a), The person class was more precisely segmented, whereas SOTA methods had

an ambiguous map. In panel (b), FTNet detected all semantics and distinguishable boundaries between the chair and the people, while SOTA methods were erroneous and lacked distinction and clarity.

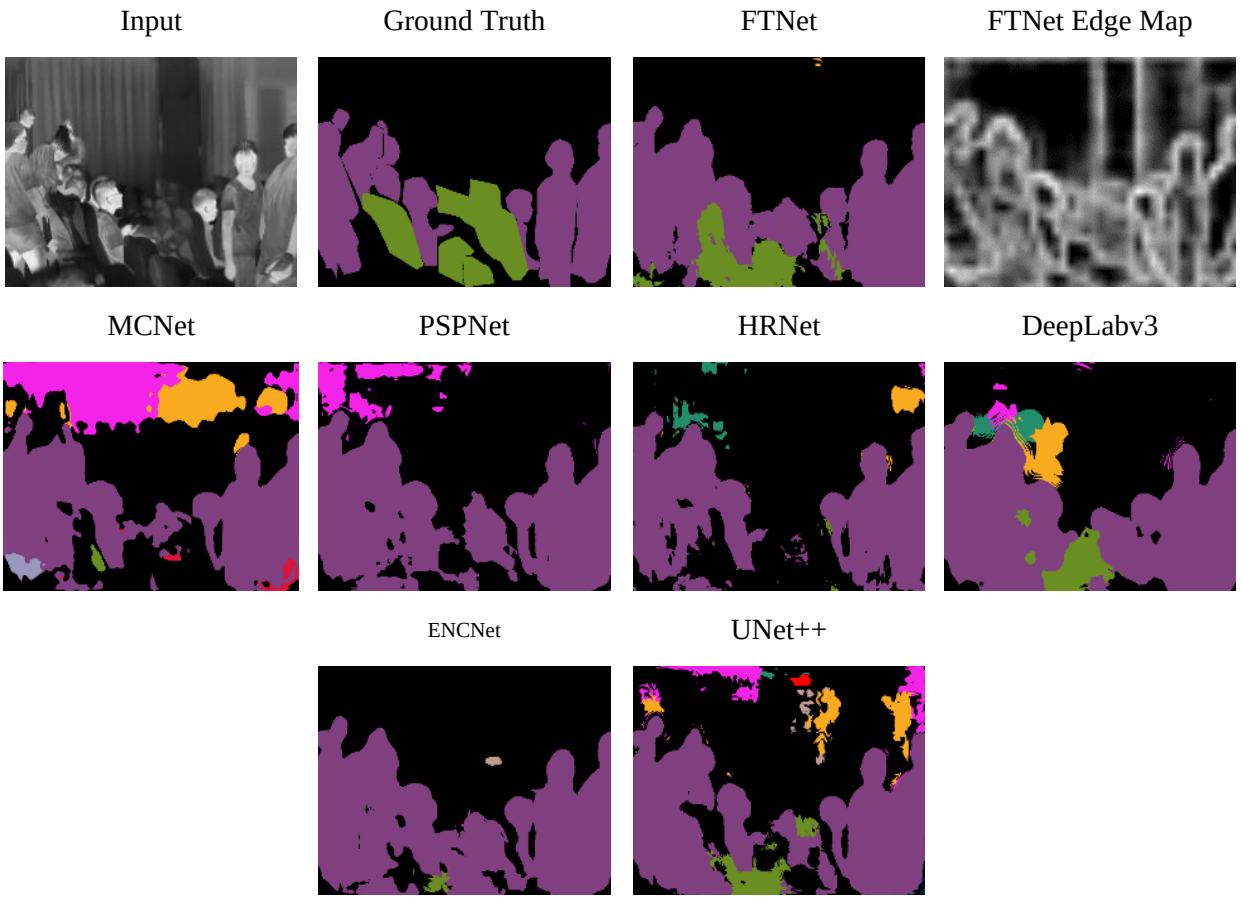


Figure 4.6: Qualitative comparison of SOTA methods for semantic segmentation of thermal images in indoor environments. Along with the segmentation output, a reconstructed FTNet edge map is included. FTNet could detect and differentiate the chair and the people, while SOTA methods were erroneous lack distinction and clarity.

Similarly, in Figure 4.5 panel b, the car has better edges, and FTNet and PSPNet, DANet, and UNet++ could detect the pole. However, FTNet detected the pole with a higher similarity to the input image. FTNet yielded acceptable results even though the SODA dataset had few indoor data representations. The proposed network reconstructs semantic maps with a higher correlation to the ground truth despite the poor quality of the thermal images. These examples illustrate the ability of the FTNet to perform better in circumstances where ambiguous object boundaries are introduced by thermal crossover compared to the other models.

4.3.5 Inference Speed

Table 4.6: Inference speed comparison with MCNet. The table shows the average execution time of 300 simulations for an image of size 640 x 480.

Model	↓ Params (M)	↓ FLOPS (G)	↓ Time (in ms)
MCNet [229]	35.67	131.87	153.57
FTNet- ResNeXt – 50, F-128, 𝔘 -2	**33.44**	**94.55**	**42.78**

The network's inference speed is calculated and tabulated. A 640 480 image was passed 300 times through the network, and the average of the results was used to calculate the computation time. This experiment was repeated ten times, and the average time for these ten runs is given in Table 4.6 as the inference time. This table summarizes the runtime performance of various approaches without optimizations and the number of parameters. The runtime was determined using an Intel i9-9900K processor at 3.60GHz and an Nvidia RTX 2080 Ti GPU. The simulation results indicate that the FTNet performs comparably to other SOTA methods in runtime performance. This model is faster than MCNet in terms of memory overhead (-2.23M) and calculation (-37.42G Flops) in terms of parameter count and FLOPS. These observations highlight FTNet's potential for use on edge devices and intelligent systems such as automated driving and video surveillance.

4.4 Discussion

To the best of the authors' knowledge, the most similar works to the FTNet are MCNet [229] and HRNet [231].

MCNet introduced multiple structures to preserve boundary information rather than post-fine-tuning the semantic segmentation results. They utilize two feature representations E_1 and E_4 (see Figure 4.2 (c)) from a dilated encoder network. It employs a loss function spanning multiple correlation matrix correction module levels. However, FTNet does not use dilated networks, thus reducing the number of parameters. FTNet exploits all the feature representations $\{E_i | i = 1,2,3,4\}$ in a transverse structure to aid the network in producing high-quality semantic maps. Furthermore, a novel edge guidance mechanism is developed to produce crisp boundaries. Finally, a weighted loss function is explored to ensure that the edge and semantic losses have the same order of magnitude. This loss function preserves the boundary information along with accurate semantic maps.

HRNet extracts feature from high-resolution feature maps in parallel with the low-resolution feature maps. The extracted feature maps from multiple parallel streams are fused to obtain high-resolution representations. However, the encoder network is not interchangeable with existing encoder backbones such as VGG, ResNet, and ResNeXt.

On the contrary, FTNet is carefully designed to transverse through multiple streams of existing serially connected encoder networks. This mechanism exploits all the feature maps at various resolutions, including the low-level feature maps containing high semantic information. The introduction of the edge guidance counterpart into this network has an immense impact on the performance shown in both quantitative and qualitative analysis. Existing SOTA encoder networks

such as Xception [267], DenseNet [268], and MobileNet [269] can be repurposed with the decoder of FTNet for various applications, including image denoising and recoloring. Furthermore, the decoder in FTNet can be readily extended to incorporate the outputs of dilated convolution in applications where it is necessary to preserve the resolution.

The benchmarking datasets used in this article [215, 228, 229] have promoted the research of semantic segmentation using thermal images. However, the annotations provided in these datasets are coarse and less detailed compared to RGB datasets. This is most likely due to the extremely low inter-class variance of objects in thermal images, making accurate labeling near boundaries difficult. Additionally, some of the labels in the datasets were misclassified in SODA Dataset. The challenges mentioned above can be visualized in Figure 4.5 (input and ground truth).

Moreover, the distribution of the semantic labels is highly imbalanced among these benchmark datasets. For example, the SODA dataset comprises 1,304 road images, whereas the number of monitor images is 75. These issues lead to a negative impact on the performance of the proposed and SOTA methods.

4.5 Extended Application: Gaze Estimation Using FTNet

Humans can rapidly direct their attention to critical regions of visual scenes. [270]. The ability to selectively focus attention on salient parts of a scene allows the human visual system to process this amount of information in real-time. For decades, identifying salient areas in a scene has been the subject of psychological research, and designing computational models for predicting salient areas is a longstanding problem in computer vision. [271]. Traditionally, saliency prediction relied on low-level features like contrast and edge. Recent saliency prediction research focuses on learning high-level semantics from eye fixation data. In recent years, there has been a surge in data-driven saliency models based on deep neural networks. However, with their high parameters, these models offset the capacity needed for small datasets such as the OSIE dataset [272] or the MIT1003 [273]. This section aims to maximize the high spatial learning ability of FTNet to build the smallest parameter model that could outperform state-of-the-art approaches on small datasets.

Figure 4.7: Sample images of the OSIE dataset or with eye-tracking fixations overlayed on them. These fixations are converted to a saliency map using a Gaussian kernel [272].

This section attempts to develop a very low-parameter Feature Transverse Network architecture geared towards saliency map predictions to enable real-time predictions. The proposed Gaze-FTNet architecture utilizes an ImageNet [274] data trained EfficientNet [187] encoder as the backbone and a feature transverse-based decoder network, which is efficient regarding the number of parameters/operations. Gaze-FTNet was trained on the OSIE dataset [272], which consists of

700 images annotated with objects and attributes. A total of 15 subjects viewed the images for 3 seconds. The dataset was split into training and validation using an 80:20 ratio.

Gaze-FTNet was qualitatively tested on the MIT1003 dataset [273]. It contains 1003 images collected from 15 different users. This database is widely used for training, fine-tuning, and testing examples to learn a model of saliency based on low, middle, and high-level image features.

It should be noted that the Edge Guidance mechanism of FTNet is not utilized in this section. For training the network, a progressive learning algorithm is adopted. Initially, the encoder parameters are loaded with pre-trained ImageNet weights. The input images were not augmented. The SGD optimizer [258] with a learning rate $1e^{-4}$, momentum 0.9, and weight decay 0.0001 is employed to train the model. Similar to Huang et al. [275], augmentations were not performed to replicate human vision tasks best. Augmentations provide the best replicates for computer vision tasks. Image sizes were set to 600x800. The batch size was set to 16. The network was trained for 200 epochs, and a poly learning rate policy with a power of 0.9 was used to drop the learning rate. The experiments were conducted on the PyTorch platform [259]. The models were trained on 2 V100 GPUs, and it takes around 5.5 hours to complete Gaze-FTNet training.

Loss function: The mean squared error (MSE) is the most commonly used loss function for regression. MSE penalizes outliers in predictions. MSE is the mean overseen data of the squared differences between true and predicted values. It can be formulated as shown in equation 4.7.

$$L(y, \bar{y}) = \frac{1}{N}\sum_{i=0}^{N}(y - \bar{y}_i)^2 \qquad 4.7$$

Where $\bar{y}$ is the predicted value.

Evaluation metrics: The results of saliency prediction are typically evaluated using a variety of metrics, including similarity, linear correlation coefficient (CC), and Kullback-Leibler Divergence

(KL-Div). The linear correlation coefficient (CC) is Pearson's linear correlation coefficient between the predicted map and ground truth. It is a numeric value between 1 and 1, and a value close to or equal to 1 indicates an ideal linear relationship between the two maps [276, 277]. The Normalized Scanpath Saliency (NSS) metric was created to quantify the saliency map values at eye fixation locations and normalize them using the saliency map variance [276, 277]. The Kullback-Leibler divergence is a frequently used metric for estimating the degree to which two distributions are dissimilar. This metric compares saliency maps and human eye fixations [277].

Table 4.7: Various evaluation metrics used for testing the proposed Gaze-FTNet.

Linear Correlation Coefficient (CC) [276]	$CC(P,QD) = \frac{\sigma(P,Q^D)}{\sigma(P)*\sigma(Q^D)}$	Where $\sigma(P, Q^D)$ is the covariance of saliency map P and fixation map QD.
Normalized Scanpath Saliency (NSS) [276]	$\bar{P} = \frac{P - \mu(P)}{\mu(P)}$ $NSS(P,QB) = \frac{1}{\sum_i Q_i^B} \sum_i \bar{P}_i * Q_i^B$	Where QB refers to the binary map of fixation locations and i indexes the ith pixel.
Kullback-Leibler Divergence (KL-Div) [277].	$\overline{SM}(x) = \frac{SM(x)}{\sum_{x=1}^{X} SM(x) + eps}$ $\overline{FM}(x) = \frac{FM(x)}{\sum_{x=1}^{X} FM(x) + eps}$ KLdiv = $\sum_{x=1}^{X} FM(x) * log\left(\frac{FM(x)}{SM(x) + eps} + eps\right)$	Where SM and FM are the saliency and fixation maps, X is the number of pixels, and eps is a small constant to avoid division by zero.

where p denotes the location of a single fixation and SM denotes the normalized saliency map with a zero mean and unit standard deviation. The final NSS score is the average of all fixations' NSS(p).

The performance of the Gaze-FTNet decoder was first evaluated across various EfficientNet encoder architectures. The results of this study are provided in Table 4.8. Extensive testing among

EfficientNet architectures revealed that EfficientNet B2 architecture produced the highest metrics with the lowest parameters.

Table 4.8: ablation study about the impact of various EfficientNet encoder architectures. The number of params and flops are for input size 640×480. Gaze-FTNet with Filters=8, Residual units (U) = 2 is used.

Encoder	↓ Parameters (M)	↓ FLOPS (G)	↑ CC	↓ KL-div	↑ NSS
EfficientNet B1	**6.65**	**3.14**	**0.6524**	**1.007**	**0.4682**
EfficientNet B2	7.85	3.59	0.6069	1.041	0.4364
EfficientNet B3	10.86	5.13	0.5833	1.09	0.4428
EfficientNet B4	17.74	7.81	0.592	1.093	0.4246
EfficientNet B5	28.55	12.06	0.6239	1.045	0.4375

The performance was tested quantitatively on the OSIE dataset [272] and qualitatively on the MIT1003 dataset [273]. For better understanding, the evaluation metric results are provided in Table 5.8. The reported numbers for SOTA are from the corresponding original paper. Gaze-FTNet achieved better results visually and obtained higher evaluation values when compared to Huang et al. [275]. It should be noted that Gaze-FTNet uses only 6.65 M parameters when compared to 29 M parameters used by Huang et al. [275].

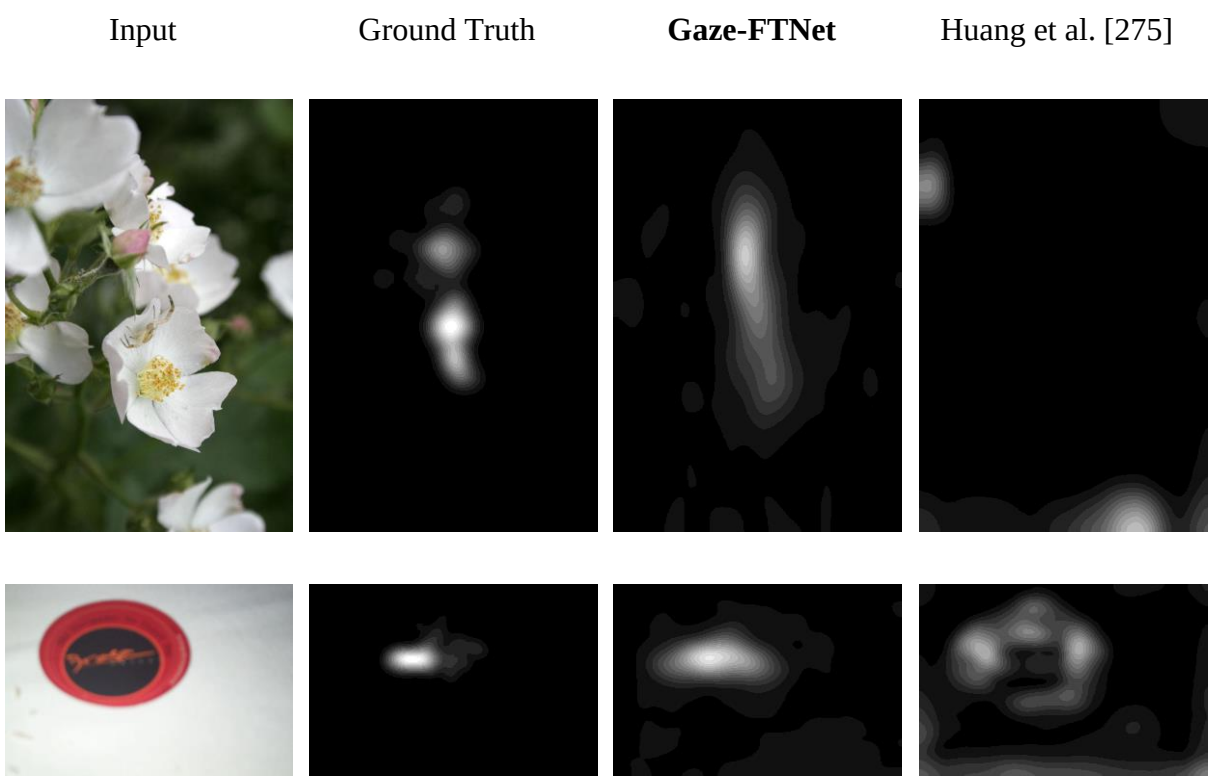


Figure 4.8: Qualitative analysis of Gaze-FTNet with state-of-the-art comparison using the MIT1003 [273] dataset. The proposed Gaze-FTNet outperformed the state-of-the-art technique qualitatively. In both instances, Gaze-FTNet estimated where the human looked at.

Table 4.9: Saliency map estimation results on OSIE dataset with Gaze-FTNet with filter size 8 and 2 residual blocks, EfficientNet B5.

	Model	Parameters (M)	CC	KL-div	NSS
OSIE dataset	Huang et al. [275]	29 M	0.4188	2.179	0.222
	Gaze-FTNet	**6.65 M**	**0.6524**	**1.007**	**0.4682**

4.6 Conclusion

A novel deep learning-based semantic segmentation network, FTNet, was presented in this work. This network aims at exploring the multi-resolution representation to perform pixel-wise classification accurately. The proposed FTNet is an end-to-end trainable architecture with a ResNeXt encoder. It employs a novel transverse-based decoder network, efficient in terms of parameters/operations and computation time. This transverse-based network captures discriminative features from multiple resolutions and combines them fully connected to achieve semantic maps close to the ground truth. An edge guidance mechanism is proposed to overcome thermal images' poor quality, single-channel, and blurry object boundary attributes. The introduction of weighted loss further improves spatial boundary information and reduces semantic ambiguity.

Extensive quantitative analysis demonstrated that FTNet achieved mIoU of 60.08%, 47.12%, and 66.73% on SODA and MFN datasets with 33.44M parameters and 94.55G FLOPS. Furthermore, the qualitative analysis showed that FTNet reconstructed rich semantic maps with crisp boundaries. These results show that FTNet can potentially optimize thermal image perception in intelligent systems such as automated driving and video surveillance applications, computational photography, biomedical analysis, and augmented reality.

The network was further successfully extended to develop an EfficientNet architecture for gaze saliency map estimation in real-time applications with very low overhead of 6.65 M. This enables real-time applications in scenarios such as fire-fighting and virtual reality. This chapter would like to recognize the lack of real eye-tracking datasets with free world movements. Most eye-tracking datasets that are available are collected in restricted and controlled environments.

4.7 Acknowledgement

This work is partially supported by the Tufts University School of Engineering, Center for Applied Brain & Cognitive Sciences (CABCS) under award numbers AR9008. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the CABCS. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Combat Capabilities Development Command, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

Chapter 5: DTTNet: Depth Transverse Transformer Network for monocular depth estimation

Abstract

Most computer vision and AI advancements are made toward 2-D image processing, such as image enhancement, object classification, and segmentation. However, the real world is three-dimensional, and there is a need to build low-cost, high-yielding, and robust AI for 3D applications. The proposed Depth Transverse Transformer Network (DTTNet) explores the multi-resolution representation to perform pixel-wise depth estimation more accurately. Moreover, this work will optimize depth perception in intelligent systems such as automated driving and video surveillance applications, computational photography, and augmented reality.

Index terms: monocular depth estimation, transverse network, convolutional neural network, deep learning.

Publications

Rajeev, Srijith, Shreyas Kamath KM, Abigail Stone, Karen Panetta, and Sos S. Agaian. " DTTNet: Depth Transverse Transformer Network for monocular depth estimation" Multimodal Image Exploitation and Learning 2022, International Society for Optics and Photonics, 2022 (Abstract accepted).

Depth estimation is crucial for inferring scene geometry from two-dimensional (2D) images [278]. It plays a vital role in computer vision since depth maps enhance the perception and understanding of real three-dimensional (3-D) scenes. Solving this task is essential to a wide range of applications such as robotics [279, 280], autonomous driving [281, 282], computational photography [283], and augmented reality [42, 213-216].

Traditional image-based depth map estimation was performed using depth cues [284, 285]; however, these methods were impractical due to many constraints. With the advent of classical hand-made features and probabilistic graph models, techniques such as scale-invariant feature transform (SIFT) [286], speeded up robust features (SURF) [287], and pyramid histogram of oriented gradient (PHOG)[288] were utilized to calculate the disparity of two 2D images through stereo matching and triangulation to obtain a depth map. These approaches assume that several scene views are available; however, this is not achievable in practice. Therefore, researchers turn their attention to monocular depth estimation.

Monocular depth estimation is the process of estimating the depth of a given scene using a single image [289]. This single image estimation technique does not require alignment and calibration compared to multiple image counterparts, decreasing time complexity [290]. With the advent of deep learning, a variety of networks, such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), variational auto-encoders (VAEs), and generative adversarial networks (GANs) [289, 291], have manifested their effectiveness to address the monocular depth estimation. Despite these advances, synthesizing the depth of a scene from a single image is highly challenging since it cannot rely on epipolar geometric constraints. Most of these networks typically have an encoder-decoder architecture. A deep CNN typically computes a feature hierarchy layer by layer in the encoder stage. It takes on a multi-scale pyramid shape by default. A higher-resolution depth

feature map is obtained on the decoder side by up-sampling the previous set of low resolution feature maps and blending with the same resolution feature maps from the encoder network through lateral connections. After extracting spatial details, the network regress continuous depth at each pixel to complete the depth map reconstruction process.

In early studies, the resolution of depth maps was limited, attributed to a series of downsampling operations performed in CNNs [292]. Several studies have been conducted to recover the degraded spatial resolution; for example up-projection was proposed by Iro et al. [293], and depth refinement by integration of CRFs into CNNs was developed by Xu et al. [294]. These networks can be computationally intensive and suffer from significant latency when processing high-resolution monocular images. Another challenge in depth estimation is the boundaries of the scene. Traditional monocular depth datasets have low feature representations at the boundaries. They may be ambiguous due to labeling errors, hardware faults, or occlusions [295].

A transformer is a novel neural network architecture that primarily employs the self-attention mechanism [296-298] to extract intrinsic features [299] and has a high potential for widespread use in AI applications. Self-attention architectures, particularly Transformers, have emerged as the preferred model for natural language processing (NLP). Researchers have recently applied transformers to computer vision (CV) tasks following the significant success of transformer architectures in natural language processing (NLP), researchers have recently applied transformers to computer vision (CV) tasks. A transformer has been used to solve a variety of other computer vision problems in addition to basic image classification, including object detection [300, 301], semantic segmentation, image processing, and video understanding [298].

This chapter presents a novel end-to-end trainable convolutional neural network architecture – depth transverse transformer network (DTTNet). The proposed network is designed and optimized

to perform monocular depth estimation. DTTNet aims to explore the multi-resolution representation to perform pixel-wise depth estimation more accurately. The proposed DTTNet architecture utilizes an existing encoder trained on ImageNet [274] data and a novel depth feature transverse-based decoder network, efficient regarding the number of parameters/operations. This transverse-based network captures discriminative features from multiple resolutions and combines them fully connected to achieve depth maps close to the ground truth. Furthermore, to leverage the efficiency of transformer networks, the mini-ViT block proposed by Bhat et al. [302] is utilized as a building block for non-local processing on the output of a CNN.

Some of the notable contributions of DTTNet include:

1) an end-to-end trainable Convolutional + Transformer network that captures discriminative depth-based features from multiple resolutions and merges them in a fully connected fashion for superior depth map estimation;

2) a network that produces high-resolution depth images at a lower computational cost by capturing and optimizing feature representation at the multi-scale resolution,

3) an extensive computer simulation performed on NYU Depth V2 [303] and KITTI [304] benchmarking dataset, which validates the superiority of DTTNet over state-of-the-art methods;

The remainder of the chapter is organized as follows. Section 5.1 provides brief literature, including a review of depth estimation methods and vision transformers. Section 5.2 describes the proposed method, and Section 5.3 presents the experimental results, including training details, ablation studies, and benchmark results. Finally, section 5.4 concludes the chapter and provides future directions.

5.1 Related Work

This section provides an overview of some of the most prominent DL architectures the computer vision community uses for monocular depth estimation.

Table 5.1: A literature review of the state-of-art techniques for depth estimation.

Author	Explanation
Eigen et al. [305]	The first monocular depth estimation problem by CNNs was proposed by introducing a global coarse-scale network and a local fine-scale network to predict the depth map from a single image end-to-end.
Eigen et al. [306]	Proposed solutions to depth and surface normal estimation using a multi-scale convolutional network.
Laina et al. [293]	Introduced a residual learning-based architecture to understand the ambiguous mapping of RGB and depth images.
Hu et al. [292]	Developed a deep-learning architecture for depth image reconstruction with high spatial resolution using a four-stage network comprising an encoder, decoder, fusion, and refinement stages.
Ranftl et al. [307].	Introduced a vision transformer architecture that could replace convolutional networks as a backbone for dense prediction tasks.

Fully convolutional networks [292, 293, 305, 306, 308] are the gold standard for dense prediction architectures. Numerous variations on this fundamental pattern have been proposed over the years. However, all existing architectures utilize convolution and subsampling as fundamental elements to learn multi-scale representations that leverage an appropriately large context [298, 307, 309]. More recently, transformer-based techniques have gained significant attention in solving various vision tasks [310, 311]. Several techniques, such as AdaBins [302] and dense prediction transformer (DPT) [312], have been proposed to leverage a Transformer encoder as a building block for non-local processing on the output of a CNN.

Vision Transformers: Inspired by the Transformer scaling successes in Natural Language Processing (NLP), a standard Transformer is applied directly to images. First, the image is split

into patches, and the sequence of linear embeddings of these patches is provided as an input to a Transformer [313]. While vision transformers have been successfully applied to various visual applications due to their capacity to capture long-range dependencies within the input, performance disparities between transformers and existing CNNs persist. One major factor could be a deficiency in extracting local data.

LeViT [314] reviewed the vast literature on CNNs and applied them to transformers, suggesting a hybrid neural network for fast picture categorization inference. BoTNet [315] substituted spatial convolutions with global self-attention in the ResNet's final three bottleneck blocks and considerably outperformed baselines on instance segmentation and object detection tasks with negligible latency overhead [298]. Yang et al. [316] suggested a Focal Transformer in which each token attends to its nearest surrounding tokens at a fine granularity and tokens further away at a coarse granularity, allowing for the efficient and effective capturing of both short- and long-range visual relationships [316].

To address the difficulties inherent in adapting Transformer from language to vision, a hierarchical Transformer (Swin Transformer) was proposed, whose representation is computed using Shifted windows. The shifted windowing technique improves efficiency by limiting self-attention computation to non-overlapping local windows while allowing cross-window communication. Touvron et al. [309] presented the CaiT (Class-Attention in Image Transformers) architecture, which increased image transformers' convergence and accuracy at greater depths. Zhou [317] addresses the subject of vision transformers' attention collapsing as they progress deeper. DeepViT proposes a low-computing and memory-intensive re-attention approach [317].

Table 5.2: A literature review of the state-of-art Vision Transformers.

Author	Explanation
DETR [300]	A simple yet effective framework for high-level vision by viewing object detection as a direct set prediction problem. Deformable DETR allows us to exploit variants of end-to-end object detectors, thanks to its fast convergence and computational and memory efficiency.
iGPT [318]	Trains a sequence Transformer to auto-regressively predict pixels without incorporating the 2D input structure knowledge.
ViT [313]	A pure transformer applied directly to sequences of image patches can perform very well on image classification tasks. When pre-trained on large amounts of data and transferred to multiple mid-sized or small image recognition benchmarks (ImageNet, CIFAR-100, and VTAB), Vision Transformer (ViT) attains excellent results compared to state-of-the-art convolutional networks while requiring substantially fewer computational resources to train.
IPT [319]	Developed a pre-trained model for image processing using the transformer architecture, namely, Image Processing Transformer (IPT). The pre-trained model must be compatible with different image processing tasks, including super-resolution, denoising, and deraining. The entire network comprises multiple pairs of heads and tails corresponding to different tasks and a single shared body. The first transformer model for low-level vision by combining multi-tasks.
Swin [320]	Introduced a general-purpose backbone for computer vision. The shifted windowing scheme increases efficiency by limiting self-attention computation to non-overlapping local windows while also allowing for cross-window connection.
Deep ViT[317]	Developed an effective attention mechanism that considers information exchange among different attention heads. They re-generated the attention maps to increase their diversity at different layers with negligible computation and memory cost. This article notes that ViT struggles to attend at greater depths (past 12 layers) and suggests mixing the attention of each head post-softmax as a solution, dubbed Re-attention.
CaiT [309]	This article also notes the difficulty in training vision transformers at greater depths and proposes two solutions. First, it proposes to do per-channel multiplication of the output of the residual block. Second, it proposes that the patches attend to one another and only allow the CLS token to attend to the patches in the last few layers.
CCT [321]	CCT proposes compact transformers by using convolutions instead of patching and performing sequence pooling. This allows for CCT to have high accuracy and a low number of parameters.

5.2 Proposed Method

This section briefly explains an end-to-end trainable CNN architecture – depth transverse transformer network (DTTNet). A basic flow diagram of the proposed system is provided in Figure 5.1.

Fu et al. [117] deduced that higher depth estimation performance could be achieved by transforming the depth regression task into a classification task. Further, they divided the depth range using a predetermined number of bins. Binning can be defined as the process of continuously grouping data into specific ranges. Binning has certain advantages, such as reducing data variability and thus normalizing the data across the dataset. However, fixed bins reduce the finer details of the output, thus reducing the quality of the predicted depth map. Adabins proposed to overcome this limitation by computing the bins adaptively.

Furthermore, the final depth values are obtained from a linear combination of bin centers and predicted depth data. This method was used in the Adabins approach using a smaller version of the existing vision transformer. The Adabins transformer is named miniViT [302]. The proposed DTTNet will improve this approach by capturing and optimizing feature representation at the multi-scale using the FTNet decoder network. This network combines low-level layers with few semantic features but high resolution with high-level layers with abundant semantic features but low resolution.

Let I be an image, and F( ) be a neural network, then DTTNet aims to develop a function F(I), such that each pixel in an image is connected to a class label of the same dimension, where I is an input image of size (H, W). The following sections provide intricate details of the network.

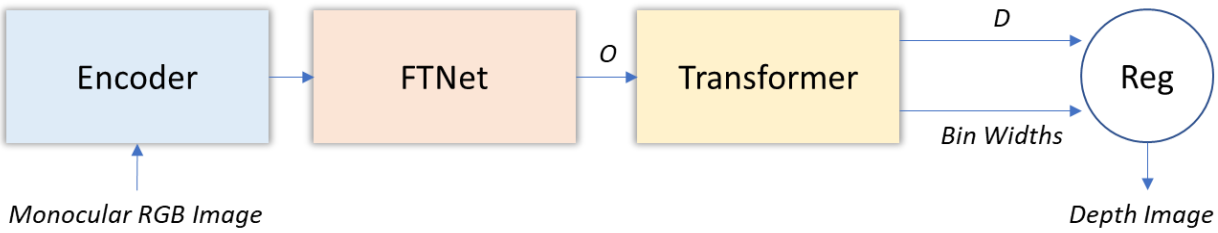


Figure 5.1: The proposed Depth Transverse Transformer Network (DTTNet) is shown. The depth estimation process is divided into three levels of learning. Level 1: Encoder, Level 2: FTNet Decoder, Level 3: Transformer, and Regression (Reg).

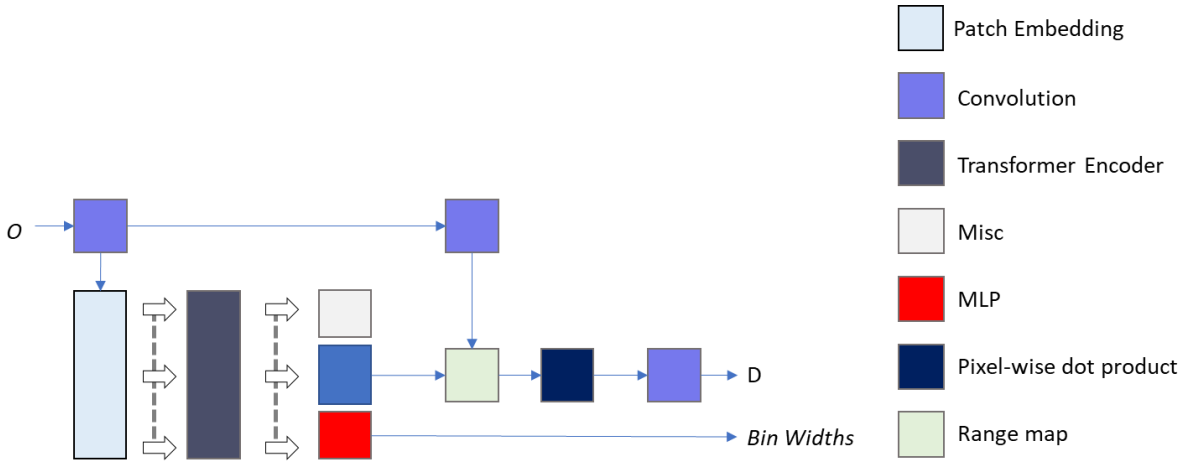


Figure 5.2: Shows the miniViT vision transformer (level 3) inspired by Bhat et al. [302]. The FTNet (decoder) O output is fed into the vision transformer. The outputs include bin widths *b* and intermediate regressed depth maps (*D*).

5.2.1 Level 1: Encoder Network

The encoder network employs existing SOTA backbones that follow the design rule of LeNet-5 [254]. For example, consider the case in which EfficientNet [187] is used as a backbone. The convolutions in these networks can be divided into four stages, and the output of each stage's last block from the encoder network can be represented as $\{E_i|i = 0, 1,2,3,4\}$. This bottom-up pathway extracts and establishes a feature pyramid by incorporating features from the last convolutional layer in each stage.

Table 5.3: The number of channels at each stage for different architectures of EfficientNet. The output of each stage's (Stages: 2, 3, 4, 6, and 9) last block is used in the DTTNet decoder.

Stage	B1	B2	B3	B4	B5
1	32	32	40	48	48
2	16	16	24	24	24
3	24	24	32	32	40
4	40	48	48	56	64
5	80	88	96	112	128
6	112	120	136	160	176
7	192	208	232	272	304
8	320	352	384	448	512
9	1280	1408	1536	1792	2048

5.2.2 Level 2: Feature Transverse Network Decoder

The FTNet decoder, described in detail in Chapter 4, is used as the Level 2 decoder. The output of the decoder is a feature set with a size $\frac{H}{2} \times \frac{W}{2} \times C$. This output is then passed to the Vision Transformer module to produce a $H/2 \times W/2 \times 1$ depth map which is then upsampled to match the target.

5.2.3 Level 3: Transformer

Similar to Adabins [302], the miniViT vision transformer is utilized. As shown in Figure 5.2, the transformer produces two outputs. Transformer output 1 is the intermediate depth map *O,* and output 2 is the bin widths.

It produces a) Range-Attention Maps R of size $\frac{H}{2} \times \frac{W}{2} \times C$, that contains useful information for pixel-level depth computation, and b) a vector b of bin-widths, which provides the various intervals of depth D in the input image. The range attention maps are generated using the dot

product mechanism between pixel-wise features and transformer output embeddings. This allows the network to integrate local and global information simultaneously. The bin widths are generated by passing the decoder features through a convolution layer of $k \times k$ kernel and k stride to produce an output with E filters, producing a feature map size of $H/k \times W/k \times E$. This is flattened to produce a vector which is known as patch embeddings. Positional embeddings are added to these and fed to the transformer. The output of the transformer is passed through a series of fully connected layers with LeakyRelu [322] and finally passed through $\Re$ to produce an N-dimensional vector b' The output is normalized to make sure the sum is 1 using equation 5.3.

$$b_i = \frac{b_i' + \mathcal{E}}{\sum_{j=1}^{N}(b_i' + \mathcal{E})} \tag{5.1}$$

where $\mathcal{E} = 10^{-3}$. This ensures that the bin widths are positive. These are then passed through the hybrid regression module [302].

5.2.4 Loss Function

DTTNet utilizes a combination of two losses, bin center density loss [302] and pixel-wise depth loss [305], with hyper-parameters μ and σ respectively as shown in equation 5.2.

$$\mathcal{L}\text{oss}_{total} = \mu\, \mathcal{L}\text{oss}_{pixelwise-depth} + \sigma\, \mathcal{L}\text{oss}_{bin-center\ density} \tag{5.2}$$

where μ and σ are continuous hyper-parameters and denote the weights for pixel and bins loss, respectively. For training, μ and σ are set to 1 and 0.1, respectively.

The Scale-Invariant (SI) loss, proposed by Eigen et al. [305], described in equation 5.3, is used to measure the relationship between points in the depth map.

$$\mathcal{L}\text{oss}_{pixelwise-depth} = \alpha \sqrt{\frac{1}{T}\sum_{i}(\log(pred_i) - \log(gt_i)^2) - \frac{\lambda}{T^2}\left(\sum_{i}(\log(pred_i) - \log(gt_i)^2)\right)^2} \tag{5.3}$$

where gt denotes ground truth depth map, *pred* denotes predicted depth map, and T denotes the valid ground truth values. Hyper-parameters λ and α are set to 0.85 and 10, respectively.

The bin-center density loss defined by Bhat et al. [302], shown in equation 5.4, keeps the bin centers close to the depth values of ground truth and vice versa.

$$\mathcal{L}\text{oss}_{bin-center\ density} = \left(\sum_{x \epsilon X} \min_{y \epsilon Y} ||x-y||_2^2 + \sum_{y \epsilon Y} \min_{x \epsilon X} ||x-y||_2^2\right) + \left(\sum_{x \epsilon Y} \min_{y \epsilon X} ||x-y||_2^2 + \sum_{y \epsilon X} \min_{x \epsilon Y} ||x-y||_2^2\right) \quad 5.4$$

Where Y is the predicted bin centers, and X is the ground truth.

5.3 Experimental Results

This section provides the quantitative and qualitative analysis of the proposed DTTNet. After outlining the experimental settings, chosen datasets, and training details, the performance comparisons with SOTA methods are provided to demonstrate the effectiveness and generalization ability of the proposed DTTNet.

5.3.1 Training Details

EfficientNet B5 is used as the encoder for all the experiments. The experiments were conducted on the PyTorch platform [259] in the Tufts High-Performance Clusters. DTTNet was trained using multiple GPUs (GPU=2). The AdamW optimizer [192] with weight decay 1e-2 was employed to train the network.

For training and testing purposes, NYU Depth v2 [303], KITTI [304], and SUN RGB-D [323] are utilized. NYU Depth v2 provides images and depth maps for various indoor settings with 640×480 resolution. The dataset is developed using a Microsoft Kinect sensor and contains

120K training samples and 654 testing samples. DTTNet is trained on a 50K subset. The depth maps range from 0 to 10m. All images are cropped according to the protocol defined in Eigen et al. [305]. A learning rate policy named 1cycle [324], which is a combination of curriculum learning [325] and simulated annealing [326], is used for training.

Table 5.4: Provides the hyper-parameters used for training the proposed DTTNet architecture.

Maximum learning rate	$\max_{lr} = 3.5 \times 10^{-4}$
Linear warm-up	$\max_{lr}/25$ to $\max_{lr}$
Epochs	25
Batch size	16

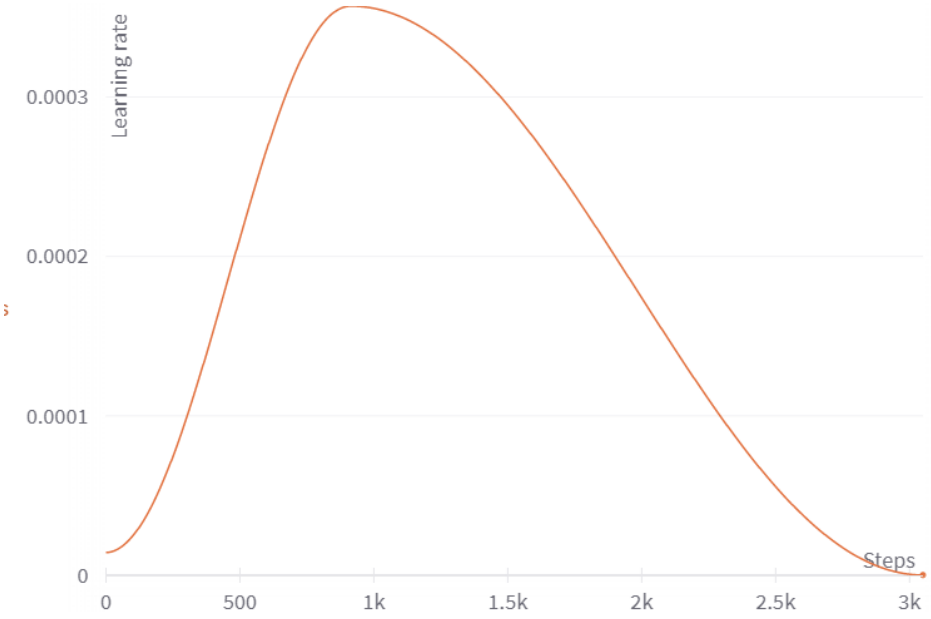


Figure 5.3: Visualizes the learning rate using the 1cycle policy [324]. The maximum learning rate ($\max_{lr}$) was set to 3.5×10^{-4}, linear warm-up from $\max_{lr}/25$ to $\max_{lr}$ for the first 30% of iterations followed by cosine annealing to $\max_{lr}/75$.

The models were trained on 2 V100 GPUs, and it takes around 12 hours to complete DTTNet training with the NYU Depth v2 dataset. Table 5.5. provides the equations for the metrics used for evaluation.

Table 5.5: Various evaluation metrics used for testing the proposed DTTNet.

Metric Name		Metric Equation
Average relative error (REL)	=	$\frac{1}{n}\sum_{p}^{n}\frac{\lvert y_p - \widehat{y_p}\rvert}{y}$
Average (log10) error	=	$\frac{1}{n}\sum_{n}^{p}\lvert\log_{10}(y_p) - \log_{10}(\widehat{y_p})\rvert$
RMSE log	=	$\sqrt{\frac{1}{n}\sum_{n}^{p}\lvert\lvert\log_{10}(y_p) - \log_{10}(\widehat{y_p})\rvert\rvert^2}$
Root mean squared error (RMS)	=	$\sqrt{\frac{1}{n}\sum_{n}^{p}(y_p - \widehat{y_p})^2}$
Squared Relative Difference (Sq. Rel)	=	$\frac{1}{n}\sum_{p}^{n}\frac{\left\lvert\lvert y_p - \widehat{y_p}\rvert\right\rvert^2}{y}$
Threshold accuracy % $of\ y_i\ s.t.$	=	$max\left(\frac{y_p}{\widehat{y_p}}, \frac{\widehat{y_p}}{y_p}\right) = \delta < thr; thr = 1.25, 1.25^2, 1.25^3$

where y_p is a pixel in depth image of image y, $\widehat{y_p}$ is a pixel in depth image of an image $\hat{y}$ and n is the total number of pixels for each depth image.

Test Time Augmentation (TTA) is performed during testing. TTA is the process of aggregating across transformed versions of a test input. The equation shows the TTA policy used for DTTNet.

$$Prediction_{Final} = \frac{1}{2}(F(X) + F(\overleftarrow{X}) \quad 5.5$$

Where X refers to input, and $\overleftarrow{X}$ refers to the mirrored input. F( ) is the trained DTTNet model.

5.3.2 Ablation Studies

1.1.1 Impact of varying the encoder topology

This section will evaluate the DTTNet decoder's performance against various encoder architectures on the NYU Depth v2 dataset. The DTTNet network's decoder components were fixed to ensure a fair comparison, and only the encoder was replaced. The findings of this ablation are summarized in Table 5.6. It can be seen that the EfficientNet-B5 architecture with DTTNet provided superior evaluation scores when compared to higher-order, higher parameter EfficientNet (v2) architectures such as EfficientNet-B6, EfficientNetB7, EfficientNetV2-M, and EfficientNetV2-L. It also performed better than lower-parameter architecture, EfficientNetV2-S. It indicates the importance of the limits of the capacity of the encoder to encapsulate the features of RGB images for depth estimation with respect to the NYU-Depth v2 dataset.

Table 5.6: Ablation study to understand the impact of utilizing different EfficientNet architectures in the proposed DTTNet with Filters=32 and residual units $\mathbb{U} = 4$.

Encoder	Encoder Parameters (M)*	$\uparrow \delta < 1.25$	$\uparrow \delta < 1.25^2$	$\uparrow \delta < 1.25^3$	$\downarrow$ Abs Rel	$\downarrow$ Log 10	$\downarrow$ RMSE
EfficientNetV2-S	22	0.771	0.956	0.993	0.161	0.067	0.515
EfficientNet-B5	30	0.907	0.987	0.997	0.102	0.044	0.367
EfficientNet-B6	43	0.894	0.982	0.996	0.11	0.046	0.387
EfficientNet-B7	66	0.813	0.976	0.996	0.137	0.059	0.481
EfficientNetV2-M	54	0.895	0.983	0.996	0.112	0.047	0.381
EfficientNetV2-L	120	0.87	0.979	0.995	0.122	0.052	0.412

Note: The reported encoder parameters are from the corresponding original paper [327].

1.1.2 Impact of changing the bin-center density

As the defined loss from equation 5.5 comprises two components, namely, pixel-wise depth loss and bin center density loss, with hyper-parameters μ and σ, respectively. It is necessary to adapt the losses to the same order of magnitude to obtain optimum results. A small bin center density weight may fail to extract optimum bins, while a large weight may dominate the binning process. For this experiment, hyper-parameters μ was set constant to 1, while σ was varied from 0.1 to 5. Table 5.7 provides the complete set of variations and their corresponding evaluation scores. DTTNET - EfficientNet B5 with 32 filter size and four residual blocks ($\mathbb{U}$) was utilized for this ablation study. The results show that bin-center density hyper-parameter weight, $\sigma = 0.1$ provided the best evaluation results.

Table 5.7: Impact of changing bin-center density hyper-parameter weight- σ. The variation of results for different weights indicates the sensitivity and importance of the hyper-parameter for training.

σ	$\uparrow \delta < 1.25$	$\uparrow \delta < 1.25^2$	$\uparrow \delta < 1.25^3$	$\downarrow$ Abs Rel	$\downarrow$ Log 10	$\downarrow$RMSE
0.1	**0.907**	**0.987**	**0.997**	**0.102**	**0.044**	**0.367**
0.5	0.828	0.977	0.996	0.133	0.058	0.455
1	0.874	0.985	0.996	0.128	0.052	0.405
1.5	0.898	0.986	**0.997**	0.109	0.046	0.38
2	0.898	0.985	0.996	0.105	0.045	0.377

***Bold** indicates the best performance

5.3.2.1 Benchmark Results

To show the effectiveness of the proposed DTTNet Convolution-Transformer architecture, it is compared with SOTA methods such as Hu et al. [153], Chen et al. [186], Yin et al. [187], BTS [181], DAV [188], LapDepth[180], Adabins [157]. For all these comparisons, DTTNet with EfficientNet backbone B5 was utilized. The hyperparameters used were filters=32, residual blocks=2, $\mu = 1$, and σ=0.1.

The performance was tested qualitatively and quantitatively on the NYU Depth v2 [303] and SUN RGB-D [323] datasets. NYU Depth v2 contains 654 testing samples. The depth maps range from 0 to 10m. All images are cropped according to the protocol defined in Eigen et al. [305].

SUN RGB-D [323] dataset includes 10K images of various scenes captured using four different sensors. This dataset is used for cross-evaluation purposes. The pretrained model from the NYU Depth v2 dataset is qualitatively evaluated on the 5050 test images from SUN RGB-D.

For better understanding, the evaluation metric results are provided in Table 5.8. The reported numbers for SOTA are from the corresponding original papers. These results demonstrate the superior performance of DTTNet when compared to the SOTA methods. DTTNet achieved higher thresholding accuracy when compared to Adabins [302] and was on par with respect to Average relative error (REL), Average (log10) error, and RMSE log. It should be noted that DTTNet uses only 35.5 M parameters when compared to the 78 M parameters used by Adabins [302].

Qualitative evaluation is essential in depth estimation assessment, and the depth maps of DTTNet and SOTA are assessed based on the human visual system. Human visual analysis is critical in identifying characteristics of algorithms that quantitative metrics may not identify correctly. For instance, in a less detailed or incorrectly labeled dataset, quantitative metrics will automatically penalize depth estimation algorithms for correctly estimating the scene's depth [262-266].

Figure 5.4 shows the qualitative analysis of DTTNet with EfficientNet backbone B5, filters=32, residual blocks=2, $\mu = 1$ and σ=0.1 on SUNRGB-D. Visual analysis of the images shows that DTTNet produced higher quality depth maps. Similar to the FTNet results in the previous chapter, the DTTNet decoder can extract higher resolution-high quality depth maps with a clear distinction between objects. These results show the generalization of the model and its learned weights.

Table 5.8: Comparison of performances on the NYU-Depth-v2 dataset.

Method	↓ Params	↑ $\delta < 1.25$	↑ $\delta < 1.25^2$	↑ $\delta < 1.25^3$	↓ Abs Rel	↓ Log 10	↓RMSE
Hu *et al.* [292]	157.0	0.866	0.975	0.993	0.115	0.050	0.530
Chen *et al.* [328]	210.3	0.878	0.977	0.994	0.111	0.048	0.514
Yin *et al.* [329]	114.2	0.875	0.976	0.994	0.108	0.048	0.416
BTS [330]	47.0	0.885	0.978	0.994	0.110	0.047	0.392
DAV [331]	25.1	0.882	0.980	0.996	0.108	-	0.412
LapDepth[332]	73.5	0.885	0.979	0.995	0.110	0.047	0.393
Adabins [302]	78.0	0.903	0.984	0.997	**0.103**	**0.044**	**0.364**
DTTNet	35.5	**0.907**	**0.987**	**0.997**	0.102	**0.044**	0.367

*Best results are in **bold**.

- results were not provided in the original paper.

The SUN RGB-D was qualitatively tested in this section. It contains over 10,000 images collected from various sensors, including – Intel RealSense 3D Camera for tablets, Asus Xtion LIVE PRO for laptops, and Microsoft Kinect versions 1 and 2 for desktops [323]. In Figure 5.4 row [a], it should be noted that the proposed method captured depth data that was superior to the ground truth image.

Another discovered issue is a significant lack of good quality depth datasets. Moreover, most depth datasets are indoor-based. Along with the deficit in the dataset, this also highlights a major issue in AI: Good AI will be penalized by current evaluation metrics even if they outperform the ground truth.

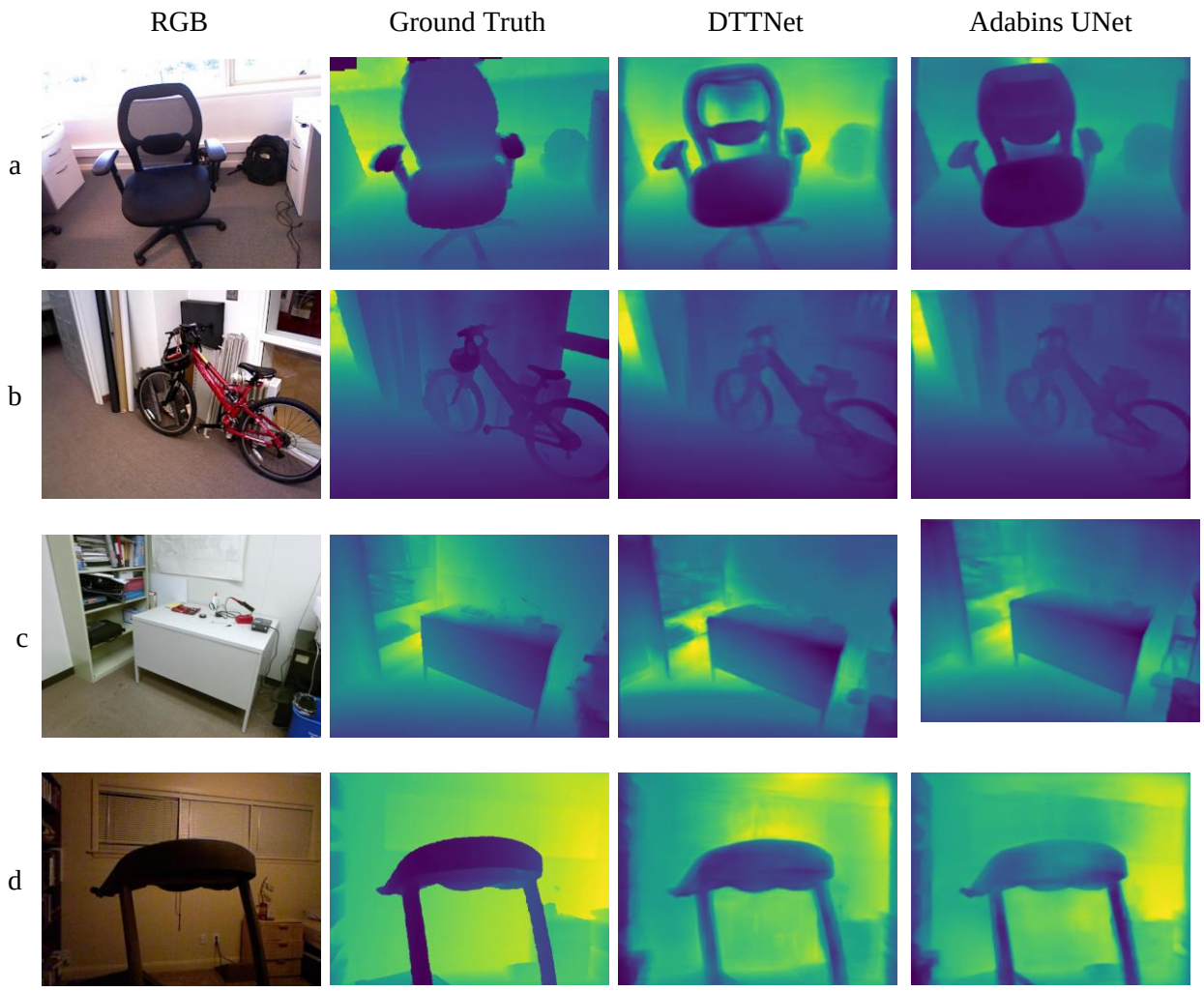


Figure 5.4: Qualitative comparison of SOTA methods with DTTNet for monocular depth estimation on the SUN RGB-D dataset [323]. Two things should be noted, (i) the ground truth (GT) data is not high quality. This results in good algorithms like DTTNet being penalized for getting better results than GT. Results clearly show the superior performance of DTTNet.

5.4 Conclusion

This chapter introduces a novel variation of the FTNet, named the Depth Transverse Transformer Network (DTTNet). The proposed end-to-end trainable approach leverages recent advances in neural networks, including convolutional neural networks and vision transformers. To produce high-quality output and model long-term dependencies, the proposed network encodes input data (1) locally using convolutional neural networks and (2) globally using a vision transformer. The

proposed DTTNet consists of an encoder network, a transverse decoder network, a transformer layer, and an adaptive binning mechanism. DTTNet captures and optimizes feature representation at the multi-scale resolution, thereby increasing the capability of the system to handle high-resolution images and producing high-quality output at a lower computational cost than SOTA techniques. Quantitative and qualitative analysis shows the robustness, superiority, and generalization quality of low-parameter DTTNet with respect to high-parameter state-of-the-art methods. Quantitatively, it performed better than the Adabins Unet architecture. It achieved this result with only 35.5 M parameters, while Adabins required 42.5 M more parameters (total: 78 M) to achieve a similar result. Finally, this chapter recognizes the lack of good quality indoor and outdoor depth datasets through literature review and experiments.

Understanding cause and effect would bring existing visual AI systems closer to human visual intelligence. However, deep learning architectures primarily focus on the correlation of data and ground truth. A future work that could solve the challenges of causality in Visual AI is to combine semantic information (FTNet), depth information (DTTNet), and attention (Gaze-FTNet).

5.5 Acknowledgement

This work is partially supported by the Tufts University School of Engineering, Center for Applied Brain & Cognitive Sciences (CABCS) under award numbers AR9008. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the CABCS. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Combat Capabilities Development Command, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

Chapter 6: No-reference Color-Depth Quality Measure: CDME

Abstract

The objective of image quality assessment is to represent the subjective perception of image quality quantitatively. Although there are several two-dimensional (2D or color) quality metrics, there is a visible void in objective depth quality assessment. This chapter proposes a novel no-reference-based depth image measure and further fuses this measure with an extended color quality metric. The Color-Depth image quality measure (CDME) demonstrates a very high correlation with human observations. CDME will have wide applications in evaluating 3D images in fields and industries such as virtual reality, autonomous driving, biometrics, and biomedical.

Index terms: 3D images, image quality, no reference, depth image, measure, 3D quality, metrics, stereo vision, structured light, structure from motion.

Publications

Rajeev, Srijith, Karen Panetta, and Sos S. Agaian. "No-reference color-depth quality measure: CDME." In Mobile Multimedia/Image Processing, Security, and Applications 2018, vol. 10668, p. 106680L. International Society for Optics and Photonics, 2018.

Accurate representation of objects, large and small, has always been at the forefront of scientific interest. 3D Scanning is the process of capturing reality data, that is, the exact 3D state of a space in time. Traditionally, reality data is captured using two-dimensional photographs or measuring tape and a pen and paper to make measurements and reconstruct a two-dimensional drawing of an object or scene. The introduction of automated 3D scanning limits or removes human error measurements and expedites the process of recording measurements. The ability to integrate 3D scan data with 3D software and virtual technologies can provide design and manufacturing education users with highly accurate data to construct complex organic shapes, which may not have been possible when only using advanced 3D surface modeling software in the past [333].

Rapid developments in the connectivity between three-dimensional (3D) software and hardware allow for quicker, easier applications of 3D scanning technologies and other virtual technologies. These have made the acquisition and reconstruction of complex surfaces possible in many disciplines [333]. Consequently, 3D images are widely used in applications such as building/cultural site restoration, artifact study, architecture, 3D printing, biometric identification, and automobile engineering.

3D scanners use one or more of several types of acquisition technology to capture scenes and automate measurements. Table 6.1 gives a brief description of the applications of 3D scanners. Stereo vision, laser light projection, structured light projection, structure from motion (SFM), and photometric stereo are some of the widely used 3D scanners.

During the sensing process, the scanner collects information about the shape and dimensions of the scanned object and, depending on the technology, can record, for example, information about the object's color. Although various 3D scanners utilize different capturing and reconstruction methods, they ultimately produce a form of depth images that are used to generate 3D point clouds.

However, the fundamental problem with depth image-based processing is that the depth image could contain blocky artifacts, discontinuities, and noise.

Table 6.1: Applications of 3D scanners [334].

Field	Description
Biometrics	3-D images of fingers and faces are used for verification and identification [335, 336].
Reverse engineering	3-D scanners are used to acquire the 3-D geometry of objects. This information is fed into 3-D printers, which can reconstruct the object.
Product design	To visualize and design products using simulation, thus reducing design a testing costs.
Quality control	Industries can use 3-D scanners to identify defects in products.
Architecture	3-D visualization of buildings for partial or complete reconstruction.
Multimedia	It can be used in video games for modeling game arenas and locations.
Artifacts	It can acquire 3-D images of artifacts that can be used for detailed reconstruction or study.
Medicine	Cancer detection by 3-D stacking CT/MRI scans. Fracture detection, determining the severity of wounds.

Estimating the quality of 3-D objects has appeared as an open and challenging issue in many processes such as mesh simplification, mesh compression, and 3-D shape matching. Quality metrics are necessary to address acceptable distortions and to which 3-D processing algorithms should resist. The effect on the 2-D projection of the 3D shape has to be carefully analyzed, including lighting and texturing procedures [337]. Image quality can be defined as a sense of fitness for image processing algorithms. For instance, 3D image quality metrics can identify blocky artifacts, discontinuities, and other distortions, present in-depth maps, and perform corrective measures based on the quality metrics. Smoothing and inpainting operations are performed to remove such anomalies. It also completely removes dis-occlusion areas where potential artifacts can arise from image warping. [338]

Image characteristics and attributes are crucial in 3D quality and authenticity; because digital images are the basis of the photogrammetric result [339]. Factors such as Field of View, Colorfulness, Contrast, Exposure, and Lighting play a dominant role in the image's efficiency level as photogrammetry principles. The two most important parameters related to 3D images are perceivable image depth and depth resolution because they are the main parameters when discriminating 3D images from plane images. The perceivable image depth and the depth resolution are defined as the depth range viewers perceive from an image of a scene and the range divided by the resolvable number of depth layers within the range. They play key roles in determining the depth to be covered and accuracy when a stereoscopic camera is used to determine object positions in space [340].

The research on objective image quality assessment metrics in image processing is highly developed. Existing algorithms can be classified according to the availability of a reference image: full reference (FR), no-reference (NR), and reduced-reference (RR). Full reference metrics require a ground truth image for comparison, whereas no-reference image qualities can be measured based on the target image alone. Some of the widely used 3D quality FR metrics are Hausdorff distance, Volume-based measure, Energy minimization, and Curvature based distance.

Although there exists no specific recommendation for qualitative analysis of 3D depth, several standards, which define the conditions for subjective experiments for other multimedia content (e.g., image and video), can be adapted [341]. For example, double-stimulus continuous quality-scale (DSCQS) and simultaneous double stimulus for continuous evaluation (SDSCE) can compare depth quality between two meshes. Furthermore, peak-signal-to-noise ratio (PSNR), visual signal-to-noise ratio (VSNR), structural similarity index (SSIM), and Mutual Information (MI) [341]. However, all the aforementioned metrics are Full-Reference based.

In that context, this article proposes a no-reference color-depth metric (CDME) for the quality assessment of RGB-D images. This metric has no constraint on the 3D images being compared and demonstrates a very high correlation with human judgment. The proposed method computes a novel color-depth measure by first computing the depth based on the pixel-wised contrast of the depth image and fusing them with the metrics of colorfulness, contrast, and edge-strength of the RGB image. The rest of this chapter is organized as follows: descriptions of related systems are presented in Section 6.1, and the proposed system, computer simulations, and performances are described in Section 6.2. Finally, conclusions and future works are drawn in Section 6.3.

6.1 Related Work

6.1.1 3D Quality Assessment

Tóth (2013) [342] compares the output of different 3D scanners against a ground truth based on the BEST-FIT function, where the first step involves a rapid, rough alignment and the next step a precision adjustment [342]. Although this technique provides excellent results, it is impractical to test the quality of 3D images when ground truth is unavailable.

Hausdorff distance: The two-sided distance between two meshes can be calculated in equation 6.1. Then, the mean distance between the meshes is calculated using equation 6.2.

$$E2(S_1, S_2) = \max\left(\max_{p \in S_1}\left(\min_{p' \in S^1} Euclidean\ (p, p')\right), \max_{p \in S_2}\left(\min_{p' \in S^2} Euclidean\ (p, p')\right)\right) \quad 6.1$$

$$E_m(S_1, S_2) = \frac{1}{|S|}\int_{S_2}\left(\min_{p' \in S^2} Euclidean\ (p, p')\, ds\right) \quad 6.2$$

Where p is a given point and, S1 and S2 are the surface meshes. This metric is well correlated to perception for quite similar meshes, guaranteeing a specific error bound between two meshes.

However, it tends to be less meaningful as the measure grows larger. It also does not consider linear transformations of the meshes to be compared. Furthermore, computing this distance is computationally expensive.

Volume-based measures: The error between two meshes M1 and M2 is defined as V (M1, M2), shown in 6.3, where V is a Lebesque formula. The main argument is that if M2 is a simplified version of M1, M2 is the best approximation of M1 if the volume between both meshes is minimized. It is also shown that this metric enables to restore original sharp edges when the triangle count is sufficient.

$$V(S_1, S_2) = \int_{R^3} I_{M_1 M_2}(\vec{q})\, d^3\vec{q} \qquad 6.3$$

Where $\vec{q}$ is a generic point in R3, $d^3\vec{q}$ is the Lebesque measure of R3, and $I_{M_1 M_2}(\vec{q})$ is the indicator function of the interior volume.

Energy minimization measures: These measures are common for performing edge collapse operations. Usually, the energy is a sum of squared distances between meshes M1 and M2, such as the quadratic error measure proposed by Garland et al. [343]. Let p and n be the plane and unit normal of a point. Then, the squared distance of any point v to this plane. This simplifies to a general quadric error measure, as shown in equation 6.4. For a given quadric Q, the level surface Q(v)=ε is the set of all points whose error with respect to Q is ε.

$$((\mathrm{v}-\mathrm{p}).\mathrm{n})^2 = \mathrm{v}^{\mathrm{T}}\mathrm{n}\mathrm{n}^{\mathrm{T}}\mathrm{v} - 2(\mathrm{n}\mathrm{n}^{\mathrm{T}}\mathrm{p})^{\mathrm{T}}\mathrm{v} + \mathrm{p}^{\mathrm{T}}\mathrm{n}\mathrm{n}^{\mathrm{T}}\mathrm{p}$$

$$\mathrm{Q}(\mathrm{v}) = (\mathrm{A},\mathrm{B},\mathrm{C}) = \mathrm{v}^{\mathrm{T}}\mathrm{A}\mathrm{v} - 2\mathrm{b}^{\mathrm{T}}\mathrm{v} + \mathrm{c} \qquad 6.4$$

$$\mathrm{Q_i} = (\mathrm{A_i},\mathrm{B_i},\mathrm{C_i}) = \mathrm{n_i}\mathrm{n_i^T}, -\mathrm{A_i}\mathrm{p_i}, \mathrm{p_i^T}\mathrm{A_i}\mathrm{p_i}$$

Geometric Laplacian Distance: It has been designed to capture the local smoothness of the mesh using a Geometric Laplacian. To capture the subtler visual properties the human eye appreciates, such as smoothness. This may be captured by a Laplacian operator, which considers both the topology and geometry. The value of this geometric Laplacian at vertex vi is given by equation 6.5. Furthermore, the average geometric distance between the meshes and the norm of the Laplacian distance is calculated using equation 6.6.

$$GLD(v_i) = v_i - \frac{\sum_{j \in n(i)} l_{ij}^{-1} v_j}{\sum_{j \in n(i)} l_{ij}^{-1}} \quad 6.5$$

$$\left|\left|M^1 - M^2\right|\right| = \frac{1}{2n}\left(\left|\left|v^1 - v^2\right|\right| + \left|\left|GLD(v^2) - GLD(v^2)\right|\right|\right) \quad 6.6$$

The advantage of this metric is to differentiate random noise addition on the vertices and poor reconstruction quality. As the human eye is very sensitive to local surface smoothness, the metric increases in case of local disturbances detected by the geometric Laplacian. In [344], the depth quality was analyzed by increasing the scale of the RGB-D database. For a quantitative evaluation, they measure structural similarity between the test and ground truth depth maps using the multi-scale structural similarity (MS-SSIM).

In summary, most state-of-the-art 3-D quality metrics are reference-based and are hence impractical to score a random, unique, or new 3D image captured by the scanner. Therefore, depth image-based quality metrics were introduced.

6.1.2 2D Quality Assessment

Measuring a color image's perceived quality is extremely difficult because human vision is highly nonlinear and not equally sensitive to details in each color component. The most widely recognized method of determining color image quality is the subjective evaluation Mean Opinion Score

(MOS). However, subjective evaluation is costly in time and resources; thus, it is difficult to be used in real-time processing systems. One would like to use such a measure to evaluate the quality of an image and automatically select optimal operating parameters in color image processing algorithms. [345]. Grayscale image quality measures such as EME [346], AME [347], logAME [345], and SDME [348] can be used and extended to assess color image quality. Fu proposed a general color image quality index (CIQI) based upon linear combinations of three variables, colorfulness, sharpness, and contrast [349]. Chen [350] presented two new color measures: CRME and CQE. Panetta et al. [351] proposed a no-reference, no-parameter, transform-domain color image quality measure (TDMEC). The definitions of the gray-scale NR measures are listed in Table 6.2.

Table 6.2: No-reference 2D quality metrics.

EME [346]	$EME_{m,n} = \frac{1}{mn}\sum_{x=1}^{m}\sum_{y=1}^{n}\left[20\ ln\left(\frac{I_{max;x,y}}{I_{min;x,y}}\right)\right]$
AME [347]	$AME_{m,n} = -\frac{1}{mn}\sum_{x=1}^{m}\sum_{y=1}^{n}\left[20\ ln\left(\frac{I_{max;x,y}-I_{min;x,y}}{I_{max;x,y}+I_{min;x,y}}\right)\right]$
logAME [345]	$logAME = \frac{1}{mn}\otimes\sum_{x=1}^{m}\sum_{y=1}^{n}\frac{1}{20}ln\left(\frac{I_{max;x,y}\ominus I_{min;x,y}}{I_{max;x,y}\oplus I_{min;x,y}}\right)$
SDME [348]	$SDME_{m,n} = -\frac{1}{mn}\sum_{x=1}^{m}\sum_{y=1}^{n}\left[20\ ln\left\|\left(\frac{I_{max;x,y}-2I_{center;x,y}+I_{min;x,y}}{I_{max;x,y}+2I_{center;x,y}+I_{min;x,y}}\right)\right\|\right]$

Where I is the intensity of the image pixel.

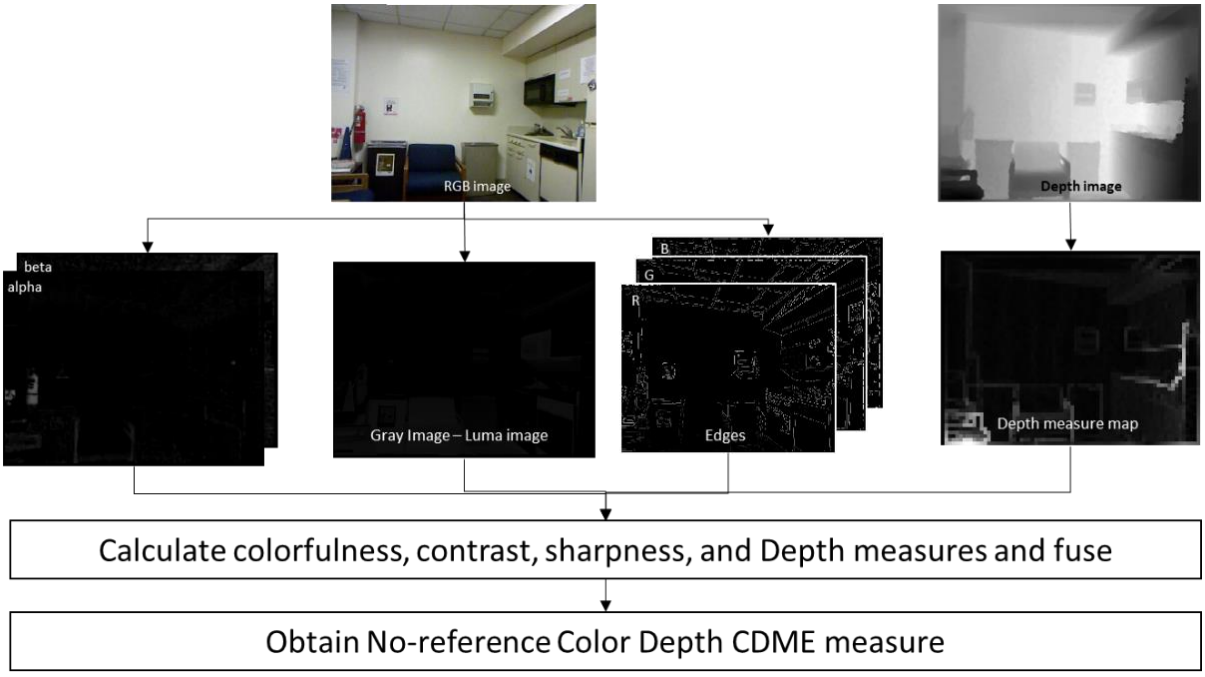


Figure 6.1: Block diagram of the proposed Color-Depth CDME approach.

6.2 Color-Depth Measure (CDME)

The proposed color-depth metric measures color, contrast, sharpness of the color, and depth of the depth images by fusing existing no-reference color measures and a novel no-reference depth measure. The following sections provide a wider description of this process. Figure 6.1 shows the operation flow of the proposed system.

6.2.1 Color measure

The existing CQE measure proposed by [350] with certain variations is used to assess the color image. Colorfulness, Contrast, and Sharpness are the three important attributes of a color image.

Colorfulness is the attribute of a visual perception according to which the perceived color of an area appears to be more or less chromatic. Colorfulness is the attribute of chrominance information humans perceive [5]. The CQE colorfulness metric is shown in equation 6.7, where $\alpha = R - G$

and $\beta = (R + G)/2$ -B are opponent red-green and yellow-blue spaces, and $\sigma_\alpha^2, \sigma_\beta^2, \mu_\alpha, \mu_\beta$ represent the variance and mean values along these two opponent color axes, respectively.

$$colorfulness = 0.02 * \log\left(\frac{\sigma_\alpha^2}{|\mu_\alpha|^{\frac{2}{10}}}\right) * log\left(\frac{\sigma_\beta^2}{|\mu_\beta|^{\frac{2}{10}}}\right) \quad 6.7$$

Contrast is the difference in luminance or color that makes the representation in an image distinguishable. The Michelson-Law measure of enhancement AME is used to evaluate image contrast. However, the difference between the Luma image and the grayscale image is used instead of a normal grayscale image. $Let\ k_1\ and\ k_2$ be the number of strides along the row and column, respectively. The measure of contrast is given in equation 6.8.

$$contrast = \frac{1}{k_1 k_2} \sum_{m-1}^{k_1} \sum_{n-1}^{k_1} log\left(\frac{I1_{max,m,n} + I1_{min,m,n}}{I_{1max,m,n} - I1_{min,m,n}}\right)^{\frac{-5}{10}} \quad 6.8$$

Sharpness is the attribute related to the preservation of fine details and edges. Sharpness is also defined by the boundaries between different tones or colors zones. The canny edge detector is first used on three channels to obtain three edge mask images. This image is fused with the original image channels. [350] states that sharpness can be determined by measuring the Weber contrast-based Measure of Enhancement (EME) with an overlapping window, as shown in equation 6.9. In this chapter, $s_1 = 0.29, s = 0.58, and\ s_3 = 0.11.$

$$Sharpness_{channel} = \frac{1}{k_1 k_2} \sum_{m-1}^{k_1} \sum_{n-1}^{k_1} log\left(\frac{E_{channel,max,m,n}}{E_{channel,min,m,n}}\right) \quad 6.9$$

$$Sharpness = 4 * (s_1 * Sharpness_R + s_2 * Sharpness_G + s_3 * Sharpness_B) \quad 6.10$$

6.2.2 Depth Measure

A novel depth measure is developed based on Weber's measure. Depth images are grayscale images with the pixel's intensity describing the depth or distance of the corresponding pixel in the RGB image. In the database in use, darker intensities indicate that the object is closer than objects with lighter intensities. A discontinuity can be detected by identifying the sharp discontinuities and the high-intensity maps.

Discontinuities at the edge of the windows will result in high-frequency components, which may lead to erroneous results. To prevent this effect, images are divided into non-overlapping blocks. Let D be the depth image, B_{wj} be a window of the depth image, and j represents the number of windows in the image (equation 6.10). B_{j1} and B_{j2} represent the upper and lower values of the sorted window (equation 6.12). DM_1 measures the contrast in Weber's measure form (equation 6.13). DM_2 is a two-dimensional Michelson contrast-based measure of visibility (equation 6.14). Finally, DM is the fused depth metric, as shown in equation 6.15, where d is a fusing constraint. It was set to 0.5 in this article.

$$B_{Wj} = D_j(window) \tag{6.11}$$

$$B_{j1} = \sum_{i=0}^{\frac{n}{2}-1} B_{Wj}(:) \qquad B_{j2} = \sum_{i=\frac{n}{2}+1}^{n} B_{Wj}(:) \tag{6.12}$$

$$DM_1 = \sum_{j=1}^{k_1 * k_2} \frac{B_{j1}}{B_{j2}} \tag{6.13}$$

$$DM_2 = 100 * \sum_{j=1}^{k_1 * k_2} \frac{I_{max} + I_{min}}{I_{max} - I_{min}} \tag{6.14}$$

$$DM = \frac{1}{2 * k_1 * k_2}(d * DM_1 + (d-1) * DM_2) \quad 6.15$$

DM is a no-reference measure of depth for depth images. The main concept is to identify the discontinuities and visibility components of the image and compute a varying form of Weber's and Michelson-based metrics for each window. Finally, colorfulness, contrast, sharpness, and DM are fused to provide a new RGB-D score, as shown in equation 6.16. The values of the constraints (c1, c2, and c3) for different conditions are provided in [350].

$$CDME = c_1 * colorfulness + c_2 * sharpness + c_3 * contrast + 0.25 * DM \quad 6.16$$

Figure 6.2: shows the examples of RGB(row1), Depth-inpainted (row2), and Depth-raw (row3) images.

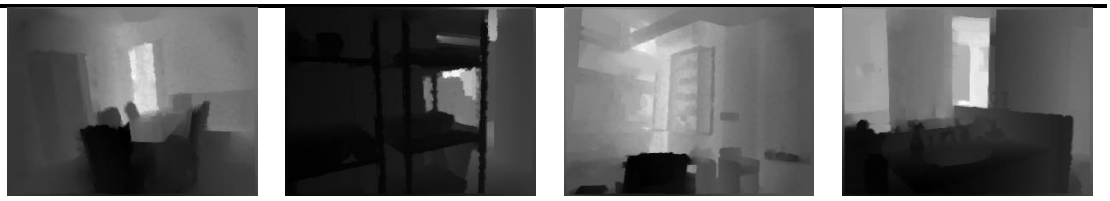
6.2.3 Experimental Results

The dataset consists of 1449 RGBD images from a wide range of commercial and residential buildings in three different US cities, comprising 464 indoor scenes across 26 scene classes [303].

Furthermore, the dataset contains raw depth images, which have a lot of discontinuities and inpainted alternatives. By visual observation, it can be presumed that inpainted images have higher quality due to the lack of high-frequency components and discontinuities. Examples of the dataset are shown in Figure 6.2. The error measures and the comparisons were conducted on all 1449 RGBD images, but 73 images were chosen randomly for visual demonstrations.

6.2.4 Evaluation of the depth images with existing NR image error measurements and DM

Table 6.3: Evaluation of inpainted depth images with DM and other NR image error measurements.



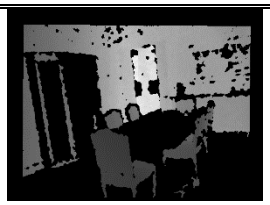
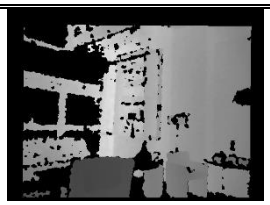
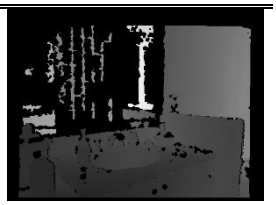
DM	1.328683	2.495381	1.285275	1.404211
EME	0.004184	0.011498	0.004202	0.005383
AME	0.060001	0.145134	0.057288	0.070277
logAME	0.022031	0.053413	0.021549	0.024973
SDME	0.017646	0.057162	0.018786	0.024249

*Higher scores indicate higher quality

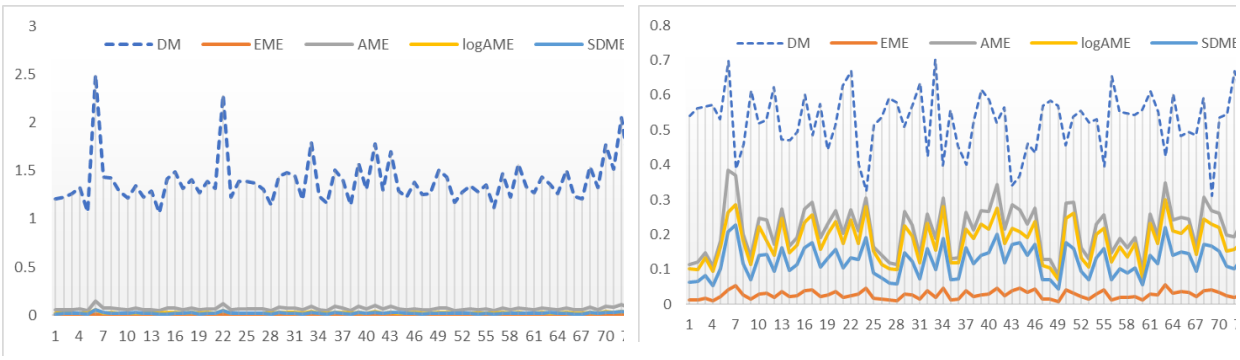
DM shows good performance in evaluating the depth image quality if the image is distorted with discontinuities and noise. Such high-frequency components are noise, and the DM measure identifies them and provides a lower value if they occur. Table 6.3 and Table 6.4 shows the result of DM, EME [346], AME [347], logAME [345], and SDME [348] for inpainted and raw depth images. From Table 6.3 and Table 6.4, and Figure 6.3, it can be inferred that all the NR measures other than DM provided the raw images a higher value indicating that the raw images have higher

quality. However, that is an incorrect evaluation. For instance, consider the first image in Table 6.3 and Table 6.4. The NR image quality metrics EME, AME, logAME, and SDME, provide the raw image a higher quality than the inpainted image. This is because of the relatively high contrast in the raw images.

Table 6.4: Evaluation of raw depth images with DM and other NR image error measurements.

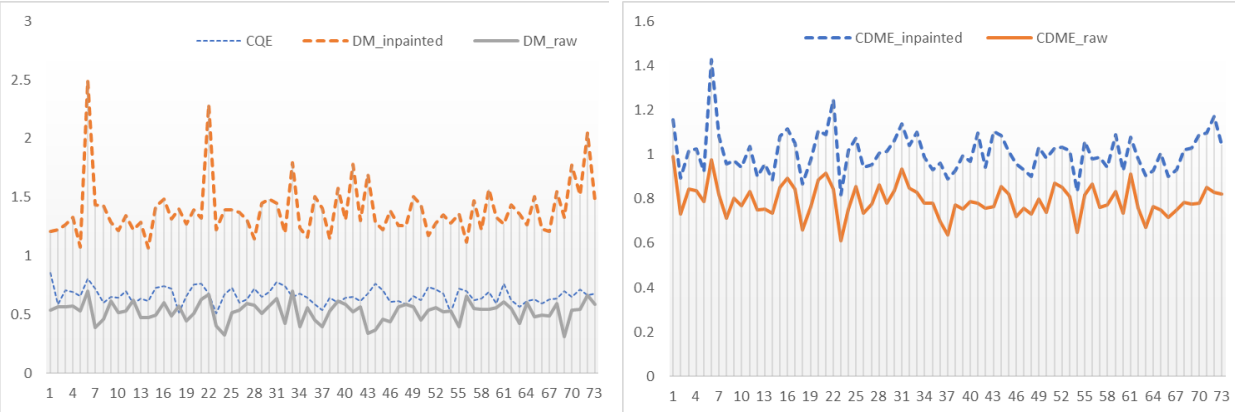
				
DM	0.31071	0.696659	0.471816	0.399396
EME	0.041516	0.041515	0.037363	0.03997
AME	0.268103	0.383835	0.273387	0.2635
logAME	0.231173	0.262731	0.2463	0.216635
SDME	0.168165	0.207645	0.161995	0.162228

*Higher scores indicate higher quality



(a) (b)

Figure 6.3: Comparisons of DM with other Image Error measurements for (a) inpainted depth images and (b) raw depth images.



(a) (b)

Figure 6.4: Comparisons of (a) CQE output for the RGB images and DM output for inpainted vs. raw depth images and (b) CDME for inpainted vs. raw depth images.

6.2.5 Evaluation of the CDME with inpainted images vs. raw images

Figure 6.4 (a) shows the graphical result of the color measure and the depth measure results for inpainted and raw images. The color and depth measures are fused to obtain the CDME scores. Figure 6.4 (b) shows the graphical result of the scores of CDME on inpainted (higher quality) images vs. the raw (lower quality images). The final fused scores clearly distinguish between the RBG-D raw images and the RGB-D inpainted images, which can be used to select optimal parameters for several processing algorithms.

6.3 Conclusion

In 3D acquisition and processing systems, NR quality metrics are necessary to address acceptable distortions. This article proposes a novel no-reference-based depth image measure and further fuses this measure with an extended color quality metric. The Color-Depth image quality measure CDME has no constraint on the 3D images being compared and demonstrates a very high correlation with human judgment. The proposed method computes a novel color-depth measure by first computing the depth based on the pixel-wised contrast of the depth image and fusing them

with the metrics of colorfulness, contrast, and edge-strength of the RGB image. CDME has a fast processing time, applicable for real-time color-depth processing systems.

Experimental results show that most NR image quality metrics cannot be applied to depth images as they do not detect blocky artifacts and dis-continuities. Further simulations showed that CDME clearly distinguishes between the high quality (RBG-D raw images) and low quality (RGB-D inpainted) images. Furthermore, the CDME could be used to provide optimal parameters for 3D post-processing algorithms.

Chapter 7: Non-Parametric 3D Modeling for Biometric Applications

Abstract

One of the most fundamental challenges in implementing 3D biometric authentication globally is that most existing biometric data was collected in two dimensions. This leads to a big challenge in biometric verification known as biometric interoperability. Most biometric systems assume that the data being compared were acquired using the same sensor and hence are limited in their capacity to match or compare biometric data acquired using different sensors [33-36]. With the advent of 3D biometrics, it will be crucial to develop 3D to 2D biometric interoperability for 3D biometric verification using the existing 2D biometric database. This chapter introduces a shape-independent 3D unrolling algorithm, pressure simulation, and mosaicking algorithm. This system can further transform and manipulate 3D images of real-world objects with geometric precision.

Index terms: biometric, fingerprint, palm-print, 2-D, 3-D, authentication, verification, identification, unrolling, mapping, modeling, security, simulation, synthetic database.

Publications

Panetta, Karen, Srijith Rajeev, KM Shreyas Kamath, and Sos S. Agaian. "Unrolling post-mortem 3D fingerprints using mosaicking pressure simulation technique." Ieee Access 7 (2019): 88174-88185.

Rajeev, Srijith, Shreyas Kamath KM, and Sos S. Agaian. "Method for modeling post-mortem biometric 3D fingerprints." In Mobile Multimedia/Image Processing, Security, and Applications 2016, vol. 9869, pp. 159-168. SPIE, 2016.

Rajeev, Srijith, Shreyas Kamath KM, Karen Panetta, and Sos S. Agaian. "Forensic print extraction using 3D technology and its processing." In Mobile Multimedia/Image Processing, Security, and Applications 2017, vol. 10221, p. 102210L. International Society for Optics and Photonics, 2017.

Biometrics refers to any human physiological or behavioral trait used for identification and verification [352, 353]. The need for biometrics can be found in federal, state, and local governments, military, and commercial applications. Several biometric modalities include fingerprint, palm print, footprint, face, iris, retina, voice, keystroke, ear, and hand geometry[354, 355]. Among them, fingerprints have the most prevalent form of easily retrievable evidence as they provide better security, higher efficiency (low error rate), user convenience, and feasibility (low-cost processing, expense) [356]. Numerous approaches for fingerprint verification exist, such as minutiae matching [357-359], basic pattern matching, and moiré fringe patterns [360, 361].

Fingerprint capturing technology has advanced from ink-based to sensor-based to three-dimensional (3D) fingerprint acquisition technology [359]. The ink-based acquisition is obtained by applying ink on the finger surface, pressing the finger face-first onto a card, or rolling the finger from one side of the nail to the other on a card, and then finally, scanning the card to generate a two-dimensional (2D) digital image [362]. In sensor-based acquisition, rolled or slap fingerprints are obtained by rolling/ pressing a finger over a sensor [363, 364]. A sensor-based system is one of the most prevalent commercially used systems, and it has a plethora of tools for processing and recognizing 2D fingerprint images. However, the performance of state-of-the-art fingerprint recognition systems is below the expected potential. This is because of complications, such as (i) non-linear plastic distortions, which arise due to non-uniform pressure applied by an individual onto the scanner, (ii) variability in minutiae markup by forensic examiners, (iii) accumulation of dirt, (iv) improper finger placement, and (v) sensor noise. These techniques require a trained fingerprint acquisition expert to assist the user in pressing/rolling the finger on the sensor [365-367].

Moreover, the accuracy of 2D fingerprinting recognition systems is profoundly reliant on the quality of the image [368, 369].

Special case: Forensic-biometric identification, particularly post-mortem fingerprints of the deceased, is crucial for criminal investigations. Post-mortem fingerprints are the deceased's fingerprints after the incident, whereas ante-mortem prints are the known prints stored in a database. However, matching post-mortem and ante-mortem fingerprints is quite tedious and cumbersome [370]. It is also one of the most complex problems in identifying unidentified human remains during singular deaths or mass disasters. Post-mortem fingerprint recovery is considered a scientific identification modality. It necessitates that all post-mortem technicians possess adequate training with a scientific background. This training is extensive, expensive, and requires considerable resources. Immaculate documentation and photography of the decedent's fingers must be performed before the procedure. The decedents' fingers must also undergo reconditioning processes to prevent the rapid onslaught of decomposition [371-375]. Traditional techniques such as ink and stock fingerprint card, adhesive lifter, and livescan are employed to acquire fingerprints with deformations. The sample is recorded by physically manipulating the recording medium against the post-mortem finger [376, 377].

Moreover, post-mortem fingerprints do not have a definite structure and are altered due to decomposition. Also, unlike traditional fingerprints, post-mortem fingerprints may be missing features and have deformities, leading to a higher error rate. State-of-the-art 2D fingerprint systems cannot model the deformities to provide viable fingerprint images.

Contactless or touchless biometric identification systems use sensors, such as ultrasound sensors [378] and single cameras to capture fingerprint images. Although these systems have overcome a few of the problems of touch-based systems, they present other complications in fingerprint

imaging. Ultrasound sensors are not widely used due to their substantial cost and size, and single cameras have a low field of view [379]. To overcome these problems, multi-view touchless sensing techniques are implemented [380, 381]. However, these systems pose the same problems as 2D fingerprint images; that is, scarred or post-mortem fingerprints may be deformed. These systems cannot model them to provide viable images for matching.

A growing realization in forensics and biometrics is that a 2D form cannot represent a human finger without loss. However, most establishments concentrate on recognizing individuals using 2D techniques. It led to the advent of 3D scanners as they provided a different scope for biometric verification [382-385]. They add depth, volume, and texture, among other 3D properties, to the biometric characteristics that can be measured, thus making them more robust and versatile [386, 387].

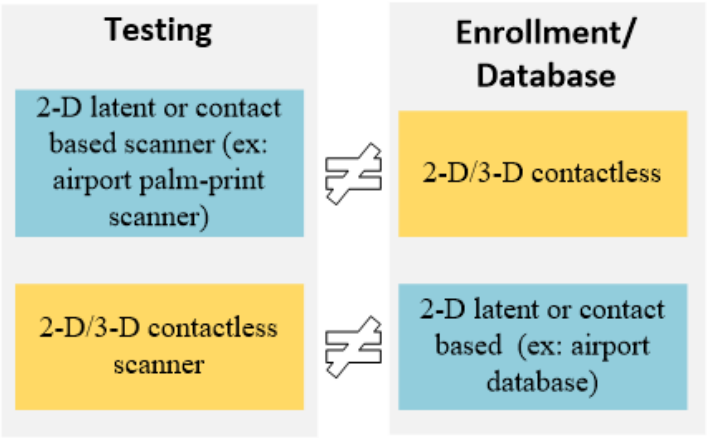


Figure 7.1: Problem of interoperability in biometric matching. Prints from one modality (example 2D) cannot be compared with prints from another modality (example 3D).

Chen [362] developed a system that generated a 3D image of a finger using a projector and camera (based on Structured Light) [388]. Similarly, 3D reconstruction can be performed using stereo or a sequence of images or videos. [389, 390]. 3D fingerprint scanners have unlocked various possibilities to strengthen biometric security [391]. However, 3D technology poses problems such

as (i) susceptibility to sensor noise, (ii) lack of modeling or mapping of post-mortem or deformed images in the state-of-the-art methods, and (iii) deficiency of affordable technology and resources that can make use of them [383].

Agencies worldwide have an extensive collection of touch-based and touchless fingerprint databases [392], but the technology required to map from 2D to 3D is not feasible [393]. Current law enforcement agencies use different acquisition methods, which require the interoperability of fingerprint templates. Additionally, agencies use fingerprint databases stored at the local, state, and federal levels. A nationwide transition of 2D to 3D biometric authentication will incur a significant expense in terms of time and finance.

Figure 7.2: An illustrative example of different pressure simulations on the same finger. Features will be displaced in different directions for different amounts of pressure.

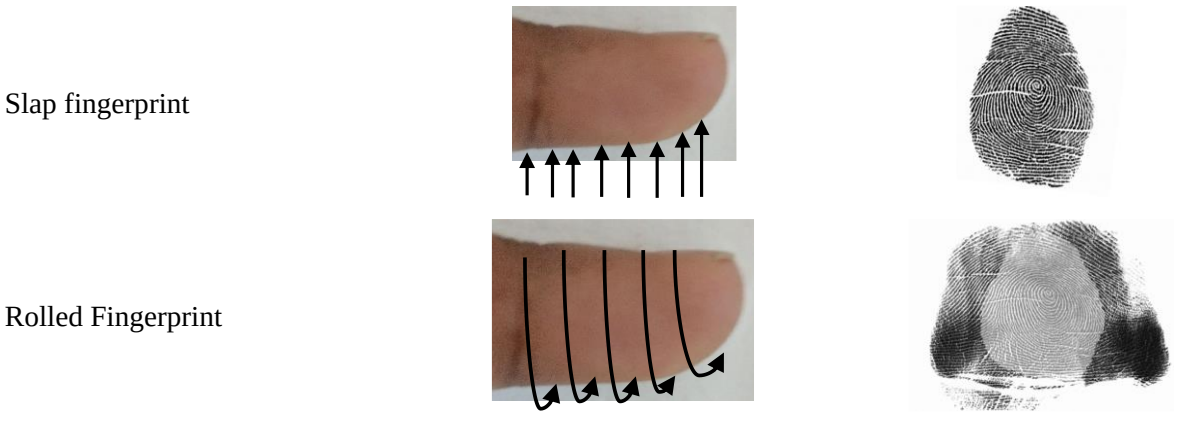


Figure 7.3: Shows the direction of pressure simulation and output for slap and rolled 2D fingerprint (Left to right or right to left) acquisition techniques.

A curved human finger is flattened on a 2D plane during acquisition, thus shifting the features in different orientations [394]. The quality of the fingerprint image is also dependent on individual artifacts such as skin conditions, deformations and pressure applied [395]. Different individuals exert different levels of pressure, which results in different variations of fingerprint [396]. This phenomenon is illustrated in Figure 7.2 and Figure 7.3. However, in a 3D system, due to its touchless properties, these variations in ridges and valleys are not present. Therefore, matching 2D and unrolled 3D fingerprints may lead to erroneous results [397]. Hence, there is an imminent need for a realistic 3D to 2D fingerprint mapping system to facilitate post-mortem biometric verification.

Some of the problems mentioned previously can be solved using 3D capture modeling. The process involves (i) capturing a 3D image of the deceased's finger after reconditioning and recovering, (ii) modeling the object onto a flat surface while maintaining the ridge mesh integrity and considering the severe deformations that may exist on the post mortem finger, (iii) resampling the unrolled image to improve structural integrity, (iv) rolling back the flat 3D image to the shape of a finger, (v) dividing the 3D image into 'n' cross-sections and then simulating pressure on each cross-section, and (vi) mosaicking each cross-section to provide an unrolled pressure simulated finger image. This process is repeated iteratively with different pressure levels to improve the probability of finding a correct match. Ante-mortem fingerprints can also be unrolled using the pressure simulation technique by performing steps (i), (v), and (vi). The afore-mentioned initial unrolling process is an extension of Rajeev's (2014) non-parametric unrolling approach [398].

Fingerprint mosaicking is widely used in multi-view touchless fingerprint systems to combine images captured from different cameras. It involves the combination of information obtained from multiple images to generate a single complete or panoramic image. Some of the main challenges

in mosaicking are: finding or generating the best seam line, visibility of the seam line after mosaicking, and color variations in the combined images [399, 400]. Rao [401] developed a new correlation technique called the alpha-trimmed correlation algorithm to mosaic finger images to tackle all these problems. The proposed Mosaic Pressure Simulation (MPS) algorithm utilizes the concepts of image mosaicking to stitch multi-views of the pressure simulated 3D finger.

This chapter describes (i) an extended unrolling framework that can efficiently unroll post-mortem fingerprints and (ii) a Mosaic Pressure Simulation (MPS) algorithm that can recreate the effects of a 2D fingerprint capturing procedure. The presented system can unroll and simulate rolling pressure on a 3D deformed post-mortem or ante-mortem finger images. The rest of this chapter is organized as follows: In Section 7.1, descriptions of related systems are presented. In 7.2, the unrolling framework is described. Section 7.3 describes the MPS system. Finally, conclusions are drawn in Section 7.4.

Table 7.1: Parametric Unrolling approaches.

Author	Description
Chen (2006) [383]	Unrolls the 3D finger by approximating it to a cylindrical surface. This approach assumes that the human finger is a collection of circles of equal radius stacked one above the other. Thus, every point of the finger is projected onto a cylinder.
Wang (2010) [401]	3D finger images are unrolled using a fit sphere approach. The distance between the sphere and the finger is then calculated.
Abramovich (2010) [402]	Unravels the finger using circular estimates of cross-sections of the 3D finger image. This approach is useful for normal slap fingerprint verification.
Wang (2010) [384]	Presented another approach that assumed that the human finger is a collection of circles of different radii. These radii are approximated to fit into the peaks and valleys of the human finger.

7.1 Related Work

This section discusses the existing systems and methods for unrolling 3D images. The 3D modeling process called unrolling represents a 3D image's surface as a 2D image. In other words,

Table 7.2: Non-Parametric Unrolling approaches.

Author	Description
Chen, (2006) [383]	This approach first upsamples the finger image to increase the density. Then each cross-section of the finger is unrolled. Theoretically, though this approach produces better results, it does not consider the variations caused by unrolling the fingerprint ridges and valleys.
Zhao, (2011) [407]	A post-processing method was presented to simulate slap level pressure on the unrolled finger image.
Fatehpuria, (2006) [393]	This approach extracts the surface of the 3D finger image using a weighted, non-linear, least-square algorithm before unrolling. The extracted surface is then unrolled using the springs algorithm.
Shafaei, (2009) [408]	It unrolls the 3D finger similarly to Fatehpuria (2006) [393] and then uses curvature analysis to superimpose the fingerprint texture on the unrolled surface.
Rajeev, (2016) [398]	This technique unrolls post-mortem fingerprints into equivalent 2D fingerprints. However, it did not concentrate on the effects of pressure.

it is the procedure of unfolding a 3D image onto a 2D plane. In this process, the letters "U" and "V" are used to denote the axes of the 2D image as "X," "Y," and "Z" denote the axes of the object in 3D space [403, 404]. This process has various applications in the gaming industry, medical imaging, industrial designs, surface recognition, and map projection [405]. Representation of the data would be preferable on a 2D plane to work with these applications [383]. Focusing on the conversion of a 3D fingerprint to a 2D flat image, pre-processing may be necessary to improve the quality of the image, as explained in [406]. Several unrolling algorithms have been proposed which can unwrap 3D images. Unrolling [407] or unraveling [393] algorithms can be divided into two classes, namely parametric (best fit) and non-parametric [383, 407].

7.1.1 Parametric Unrolling Technique

In this approach, the 3D finger is closely approximated to the shape of a known 3D entity such as a sphere, ring, or cylinder. A few parametric techniques are mentioned in Table 7.1. The unrolled outputs may have a sizeable stretching effect if the shapes are not identical. The main drawback of parametric approaches is that they assume the shape of a human finger to be a known and fixed entity. This is the case because a human finger may be bulky or thin and uneven or smooth, differing from individual to individual. These variations increase infinitely when deformed 3D fingerprints are considered.

7.1.2 Non-Parametric Unrolling Technique

Non-parametric unrolling algorithms utilize the angular relation and Euclidean distance between two successive points in the 3D mesh [383, 409]. These algorithms compute the corresponding pixels in the 2D equivalent fingerprint image from the points in the 3D finger image. Some of the implementations of this approach are shown in Table 7.2. Most of the literature assumes a non-deformed/regular 3D finger image, i.e., they are not designed for 3D images with deformities or images affected by pressure.

After an extensive literature search, the authors concluded that the state-of-the-art unrolling methods do not accurately simulate the rolling pressure exhibited during the physical fingerprint acquisition [410].

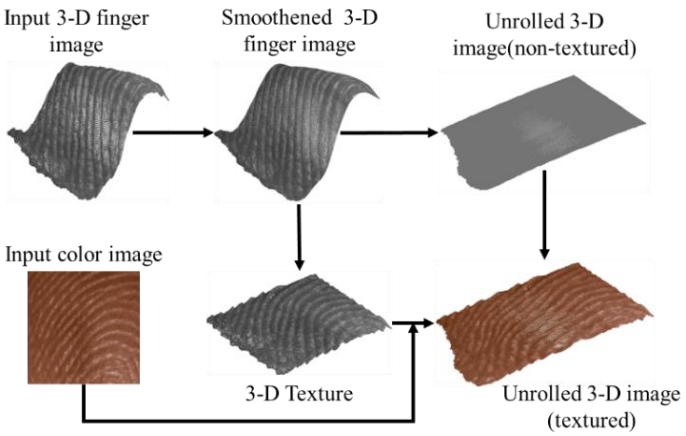


Figure 7.4: Shows the workflow involved in unrolling the deformed 3D finger. [Row 1] The input finger is de-texturized using a smoothing filter, followed by unrolling. [Row 2] Finally, the color and texture are mapped back to the unrolled 3D image. Note: A zoomed cross-section is shown for visual purposes.

7.2 Unrolling Framework

This section improves and extends Rajeev's [398] non-parametric unrolling approach. Both algorithms can unroll a 3D finger (normal or deformed) image into a flat and uniform 3D finger mesh. However, the extended algorithm can resample the mesh to improve mesh density, and it can be applied to both ordered and non-ordered point clouds. This process is developed to make it compatible with verifying a 3D finger with a 2D fingerprint database. Figure 7.4 shows the workflow involved in the unrolling algorithm. A comprehensive explanation of the unrolling algorithm is explained in the subsequent segments:

7.2.1 Anomaly Removal and Texture Extraction

A raw fingerprint image may have unnecessary background noise and undefined boundaries. A Gaussian filter (equation 7.1) removes the noise and smoothens the 3-D texture. The smoothed 3D image is used for unrolling purposes, and the difference between the original (3D) image and the Gaussian filter output (3D) provides the texture ($P(x,y,z)^{i,j}_{extracted}$) of the finger.

$$g(i,j) = \frac{1}{2\pi\sigma^2} . exp\left(-\frac{i^2 + j^2}{2\sigma^2}\right) \quad 7.1$$

where σ^2 is the variance of the Gaussian filter, i and j are the distance from the origin along the horizontal axis and vertical axis, respectively. The depth coordinates are convolved with this filter.

7.2.2 3D Image Unrolling

Unrolling is necessary to provide interoperability between 3D and 2D systems. Parametric unrolling methods depend on the knowledge of the shape of the system and are henceforth less robust. Therefore, an unrolling algorithm is presented that considers the existence of deformities and finds countermeasures to unroll the 3D finger images effectively. The following sections describe two 3D finger unrolling methods, i.e., without and with anomalies.

1) 3D images without anomalies

Few 3D reconstruction algorithms fill holes in the 3D image during mesh surface reconstruction. In such circumstances, non-parametric unrolling is performed by directly calculating the Euclidean distance between corresponding points. Equations 7.2 and 7.3 perform the regular non-parametric unrolling operation.

$$D_y(i,j) = \sqrt{\begin{matrix}\big(x(i,j) - x(i,j+1)\big)^2 + \\ \big(z(i,j) - z(i,j+1)\big)^2\end{matrix}} \quad 7.2$$

$$\bar{x}(i,j+1) = \bar{x}(i,j) \pm D(i,j) \quad 7.3$$

where x, y, and z are the surface, height, and depth coordinates, D be the Euclidean distance, and $\bar{x}$ is the new surface coordinates. The algorithm can be performed along the surface or the height, i.e., x and y are interchangeable.

2) 3D images with anomalies

Few biometric 3D scanners may avoid filling holes while performing surface reconstruction because they generate artificial data. Therefore, this chapter uses a compensation model developed by Rajeev [398] to perform non-parametric unrolling. The compensation model can be obtained by determining and saving the location of the last valid point in the row according to equations 7.4 and 7.5. These estimates are used in equations 7.6 and 7.7 to unroll the 3D image.

$$if(\, f(i,j) == 0 \;\&\&\; f(i,j-1)\; ! = 0)$$

$$\hat{f}(i,j) = \; f(i,j-1) \tag{7.4}$$

$$if\; f(i,j) == 0 \;\&\&\; f(i,j-1) == 0$$

$$\hat{f}(i,j) = \; \hat{f}(i,j-1) \tag{7.5}$$

$$D_y(i,j) = \sqrt{\begin{matrix}\big(\hat{x}(i,j) - x(i,j+1)\big)^2 + \\ \big(\hat{z}(i,j) - z(i,j+1)\big)^2\end{matrix}} \tag{7.6}$$

$$\bar{x}(i,j+1) = \dot{x}(i,j) \pm D(i,j) \tag{7.7}$$

where f = P(x,y,z) and $\hat{f} = P(\hat{x},\hat{y},\hat{z})$, P is a 3D point, x, y, and z are the surface, height, and depth, respectively, $\dot{x}$ possesses the last known real unrolled surface output in the current row. $\hat{x}$, $\hat{y}$, and $\hat{z}$ are the 3D coordinates of the points in the compensation model. The depth information is discarded after running the unrolling algorithm.

7.2.3 Texture Fusion

To create a realistic 3D finger image, the original color and texture should be incorporated into the unrolled output. The unrolled 3D image is a smooth and flat equivalent 2D image. The texture saved while pre-processing is applied to provide additional features for fingerprint recognition

systems. This step is vital to include the ridges and valleys of the original 3D finger image into the unrolled finger image.

If the input point cloud is an unorganized (or unordered) point cloud, it is first converted to an organized (ordered) point cloud. An organized point cloud is ordered as a 2D array of points, while an unorganized point cloud is a one-dimensional array. Texture fusion involves a direct one-to-one mapping using equation 7.8 [412] in this chapter.

$$For\ every\ i,j = 1\ to\ dimensions\ of\ pointcloud$$

$$P(x,y,z)_{textured}^{i,j} = P(x,y,z)_{unrolled}^{i,j} + P(x,y,z)_{extracted}^{i,j} \quad 7.8$$

7.2.4 3D Unrolled Image Resampling

The resampling process is similar to image interpolation, but 3D resampling occurs across the surface, height, and depth. The X, Y, and Z coordinates are upsampled by placing zeros between consecutive points. If no point exists adjacent to a point before up-sampling, then it is presumed that no points exist in that region. Thus, that region is excluded from interpolation after upsampling. In all other cases, interpolation is performed using a window of size (1x3). Linear interpolation is performed as shown in equation 7.9.

$$P(X,Y,Z) = (P(X,Y,Z)^{-} + P(X,Y,Z)^{+})/2 \quad 7.9$$

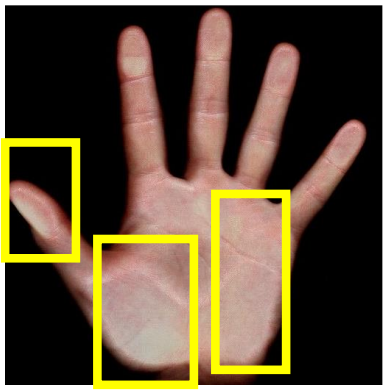
where $P(X,Y,Z)$ represents the current point coordinates, and + and – represent the next and the previous coordinates, respectively.

7.2.5 Extension to 3-D palm print mapping

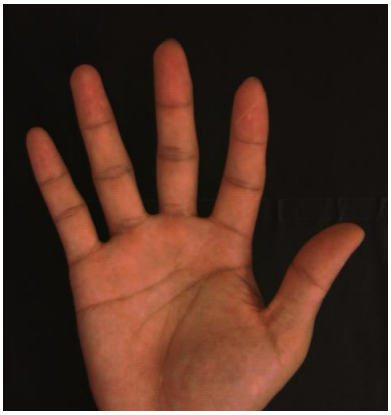
For over fifteen years, palm print recognition has been employed for identification/verification. Like fingerprints, palm-print biometrics are widely used because of their uniqueness,

distinctiveness, and permanence. Unlike fingerprints, palm prints contain additional features such as texture, indents, and marks, which can be used during a palm recognition system. The main identifiers used in the palm-print recognition system are hand geometrical features such as finger width, length and thickness, principal lines or flexion creases, secondary creases or wrinkles, and ridges [411]. These identifiers are captured using optical, capacitive, ultrasound, thermal, or 3-D scanners [312]. Currently, nearly 30 percent of the latent prints obtained from a forensic crime scene are palm prints. A major component of the FBI's Next Generation Identification (NGI) system is advancing an integrated national palmprint identification system [412].

CCD-based palm-print scanners capture high-quality images, but they require a controlled environment, and the equipment is large and expensive. Digital scanners provide portability and reduce the nature of a controlled environment, but problems still exist due to contact-based capture. Digital cameras provide portability, an easy user interface, and are non-contact-based scanners. While 2-D palmprint recognition has proved efficient in verification rate, it has some essential downsides. For example, a 2-D palm print image (i) cannot capture the palm depth information (a palm is not a pure plane); (ii) 2-D image-based palm print recognition systems are vulnerable to various kinds of attacks and can be easily copied and counterfeited; (iii) the illumination variations in the system may affect the 2-D palmprint image quality, and (iv) 2-D images can be easily affected by noise [398].

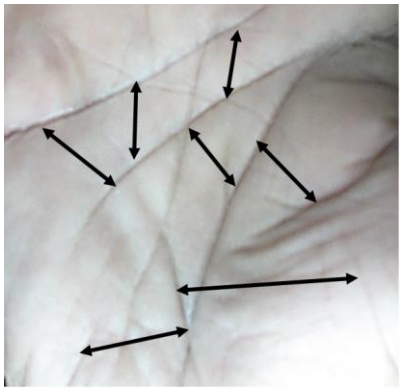


(a)



(b)

(c)



(d)

Figure 7.5: (a) contact-based palm image with the palm pressed effect [413], [b-c] 2-D and 3-D palm images with no touch-based distortions.[414] (d) shows some locations where linear distortions are likely to occur in latent or forensic cases.

However, since palm-print databases stored by most agencies worldwide are contact-based, it is of paramount importance first to transform the image into a contact-based template and then perform matching. Such transformations are nearly impossible to perform on 2-D images. There has been much development in the field of touchless biometric scanning. A panoramic image can be reconstructed by mosaicking/stitching numerous images from different views. These algorithms perform the stitching by extracting biometric or image features using point descriptors [379, 415].

From Figure 7.5(a), it can be comprehended that a large section of the palm is hard-pressed on the scanner surface during acquisition. This phenomenon is not observed in the 2-D/3-D images (can

be visualized in Figure 7.5 (b-c)) because of the contact-less property of 3-D scanners. This non-linearity may cause an increase in error rate matching. This could occur due to the change in relative distances between palm features during contact-based scanning (visualized in Figure 7.5 (d)). A direct mapping approach cannot be performed because most palmprint recognition databases contain 2-D palm-print images with the pressure exerted on the palm. The rest of this section discusses the proposed 3-D palm mapping technique.

a b c

Figure 7.6: ROI – (a) shows the input image, (b) shows the ROI 2-D image, and (c) shows the ROI 3-D distribution.

1. Estimating the Region of Interest (ROI)

The central part of the hand is required for palm-print authentication. This is extracted using a region of interest algorithm, influenced by the method proposed by Zhang et al. (2003) [416]. The ROI algorithm implemented involves:

Denoising: by smoothing the image using a low pass filter

Identification of the hand: by thresholding the image to obtain a binarized image

Locating centroid of the image: by computing the weighted centroid of the image

Obtain 2-D ROI: by obtaining setting a user-defined x, y offset, and finally

Obtain 3-D ROI: by mapping the region of interest onto the 3-D image sets.

Figure 7.6 shows the input and output image along with the depth distribution of the 3-D ROI palm output.

2. Texture Segmentation

Unrolling the 3-D image and the texture, i.e., the friction ridges, will model the features themselves. Therefore, it is vital to extract the biometric friction ridge information from the 3-D image. This paper uses a Gaussian filter to smoothen the depth information. The texture is extracted by subtracting the original 3-D image from the smoothened 3-D image. Figure 7.7 displays the input, smoothened, and extracted texture depths of the palm print. Although the texture data appears to be unrolled, it is important to note that the surface coordinates are still not mapped. Once the texture is extracted, the smoothened surface is forwarded to the unrolling stage.

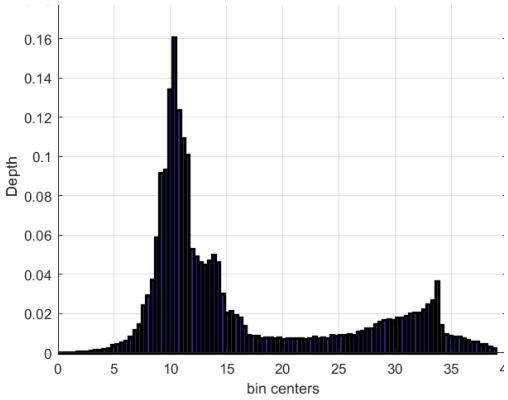
3. 3-D palm-print unrolling

A non-parametric approach of unrolling is implemented in this chapter to model the 3-D image. The non-parametric unrolling methods consider the Euclidean distance between two successive points in the 3-D mesh, mainly aiming at geodesic preservation, but do not consider the model's shape [383, 398]. This step provides interoperability with 2-D latent and 3-D palm-print images. It involves:

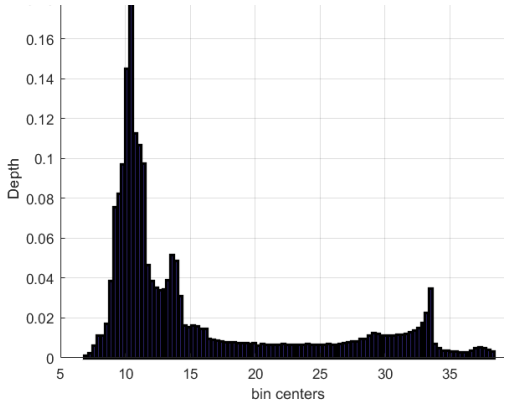
i. filtering the 3-D image for unintended or noisy points using a standard deviation filter described in equations 7.10 and 7.11.
ii. dividing the image into blocks and unrolling each block while accounting for the existence of holes or unknown data within the 3-D image using equations 7.12 and 7.13
iii. performing interpolation to manipulate the holes.
iv. combining the unrolled blocks to obtain a single unrolled 3-D palm image.

Each point in the 3-D point cloud ROI can be manipulated to model the 3-D image along the surface with the aforementioned procedure.

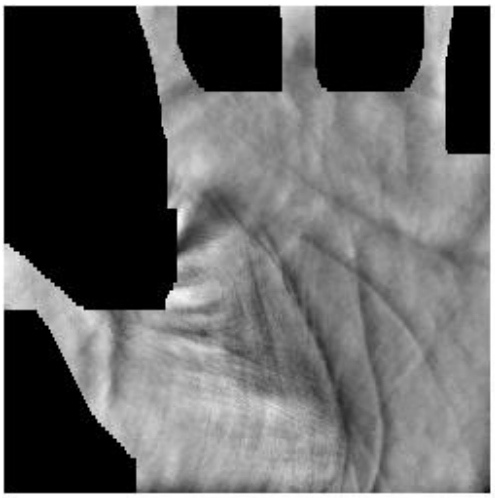
a



b



c



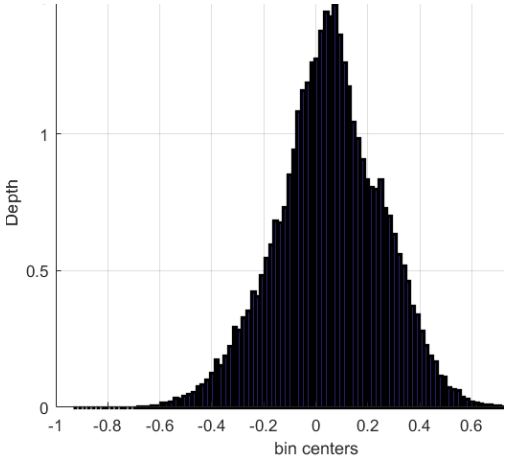


Figure 7.7: shows image-wise depth and the depth distribution/histogram of (a) Original 3-D image, (b) Smoothened 3-D image, and (c) extracted 3-D texture.

$$d_{i,j} = Euclidean\{(X,Z)_{i,j}, (X,Z)_{i,j+1} \forall\, Y, \forall\ X\ \neq 0\ \}$$ 7.10

$$X_{i,j+1} = concatenate\{X_{i,j}, d(i,j) \forall\, Y, \forall\ X \neq 0\ \}$$ 7.11

$$d1_{i,j} = Euclidean\{(X,Z)_{i,j-a}, (X,Z)_{i,j+1} \forall\, Y, \forall\ X = 0, X_{j-a}\ \neq 0\ \}$$ 7.12

$$X_{i,j+1} = concatenate\{X_{i,j-a}, d1(i,j) \forall\, Y, \forall\ X = 0\ , X_{j-a} \neq 0\ \}$$ 7.13

Where, X, Y, and Z are the 3-D coordinates, d and d1 are the Euclidean distance.

7.2.5.1 Texture and Color Fusion with the unrolled data

The final phase of the 3-D palm unrolling algorithm involves mapping the texture and color onto the unrolled palm data. The 2-D palm-print image is obtained by mapping each point in the unrolled data with corresponding RGB values. To acquire the 3-D textured data, it is necessary to map both the color and the texture segmented onto the unrolled data.

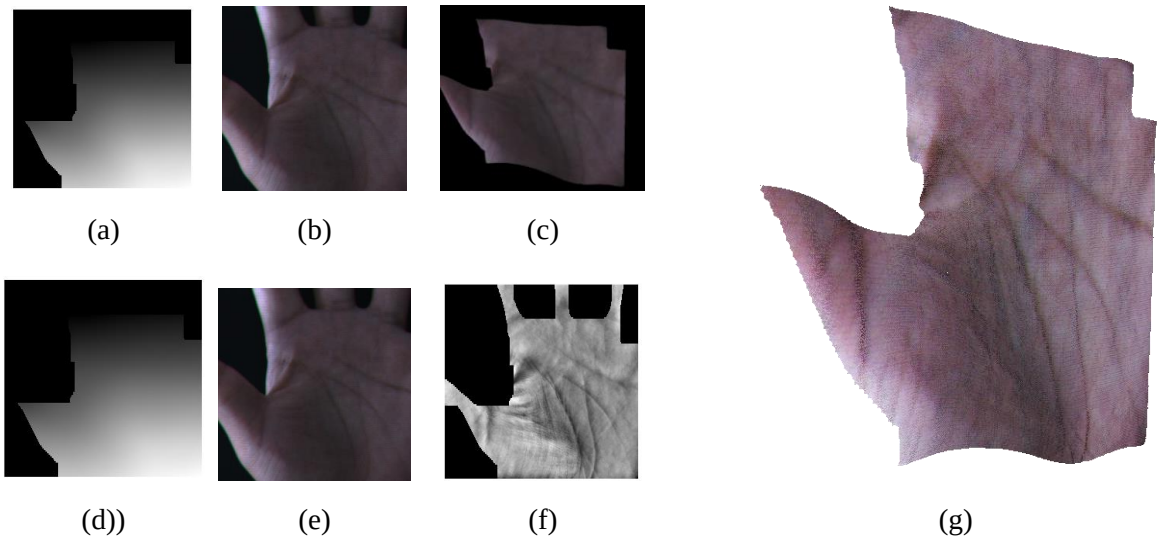


Figure 7.8: shows the 2-D color mapping and 3-D texture mapping. (a) and (d) are the unrolled surface, (b) and (e) 2-D input palm image, (c) output 2-D equivalent unrolled palm image; (f) is the extracted surface, and (g) is the 3-D unrolled palm output with color and texture.

7.2.6 Extension to 3-D Forensic mapping

Forensic science is defined as the body of scientific knowledge and technical methods used to investigate and construe pieces of evidence to provide valuable insight into criminal, civil, and administrative proceedings. It focuses on demonstrating the existence and investigation of an offense, the individualization of a perpetrator, and the description of a modus operandi. The process of forensic identification can be performed with (i) personal data, (ii) personal belongings such as keys, tags, laptops, and (iii) biometric traits[417, 418]. Common biometric features captured from a forensic scene are fingerprints, palm-prints, lip-prints, handprints, footprints, and boot prints. A sample obtained at the crime scene is called the crime scene sample, forensic print, trace, or unknown item. Biometric recognition assists forensic investigation in two distinct ways: 1) as an instrument to assist in the investigation and (ii) as evidence in a court of law [418].

Forensic prints can be classified as patent, latent, and three-dimensional plastic prints. Latent prints are invisible to the naked eye and are formed due to the sweat and oil on the skin's surface. Patent prints are visible to the naked eye and are formed on blood, grease, dye, or dirt. Three-dimensional plastic prints are formed when the individual presses wax, soap, fresh paint, or fresh cement [419]. Prints such as fingerprints, handprints, and lip marks are traditionally captured using physical means, whereas face, voice, and DNA are captured using digital means. The stability of certain biometric traits, such as fingerprint palm-print, is permanent (i.e., they do not change), while face and voice are not permanent. They change over time and emotion. Physical traces are preferred over digital traces due to their distinctiveness. However, new technologies can capture these physical traits by digital means. Traditionally, forensic prints are collected from a crime scene by experts proficient in forensic science to reveal or extract fingerprints from surfaces and objects using chemical or physical methods. The traces can then be photographed and marked up for

distinctive features by skilled examiners. The reliability and accuracy of the method depend on the ability of the officer or the electronic device to extract and analyze the data. Furthermore, the 2-D acquisition and processing system is laborious and cumbersome. This can lead to an increase in false-positive and true negative rates in print matching [418].

There are two main purposes for forensic identification of humans: suspect identification and victim identification. Evidence such as fingerprints, palm-prints, foot-prints, bite marks, and blood samples are collected at crime scenes to prove the guilt or innocence of the suspects. The goal of victim identification is to determine the identity of victims based on the characteristics of the human remains. Forensic-Biometric technology can also identify missing persons or John Doe from a mass disaster.

When forensic prints are revealed by forensic science experts using chemical methods, the prints should be captured using portable high-quality 3-D cameras. Images must be captured such that it includes minimum background information. Once captured, the image files are transferred onto a device with processing capabilities. The 3-D coordinates and the RGB image are extracted. The quality of the 3-D image is crucial for successful processing, and it is dependent on factors such as illumination, environment, camera, and 3-D reconstruction technique.

Consequently, a pre-processing segment is necessary to ensure that the unrolled 3-D image can be used for biometric recognition. Erroneous points can be removed by performing block-wise standard deviation filtering or alpha trim mean filtering [415]. The principle behind this approach is that erroneous features in a block have a larger deviation from the mean when compared to the object. These features are identified ad removed from the 3-D image. Figure 7.9 shows some of the input modalities used by the proposed system.

a b c

Figure 7.9: Sample 3-D images of biometrics used in this section. (a) 3-D fingerprint, (b) 3-D palm-print, and (c) 3-D lip-print.

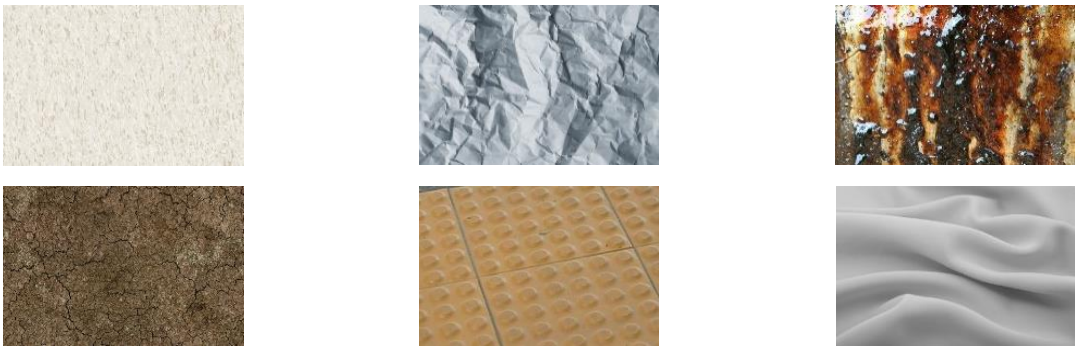


Figure 7.10: Different possible textures. From top left-right: smooth(marble), wrinkled, and greasy[420]; bottom left-right: cracked, bumpy, and silky [421].

Unlike generic 3-D biometric imaging, where the texture of the image provides the intricate friction ridge pattern, the texture on latent forensic prints cannot be relied upon. The texture of the surface may be rough or smooth, as shown in Figure 7.10. Therefore, the RGB values of each point in the input 3-D image are mapped onto the equivalent unrolled 2-D/3-D forensic print.

7.2.7 Computer Simulations

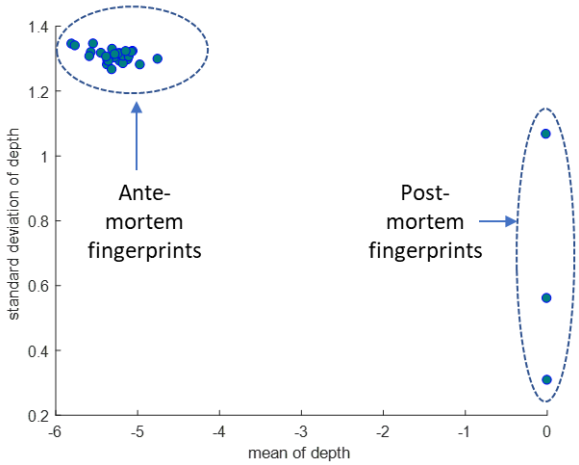


Figure 7.11: Shows the depth distribution of the 3D fingerprints in terms of mean and standard deviation. The 30 ante-mortem fingerprints have similar depth properties, while the post-mortem prints have highly varying properties.

1. 3D fingerprint database

There are no pre-existing post-mortem datasets. FlashScan3D [45]provided a post-mortem database, and it was constructed by simulating deformities on a few ante-mortem prints. The database included thirty ante-mortem 3D images and three severely deformed 3D post-mortem finger images. The mean and standard deviation of the depth images are shown in Figure 7.11. The finger images are in the MAT5 format [45, 422].

Figure 7.12 (a) shows a sub-section of the 3D finger images used in the presented work. Furthermore, the authors did not have access to the actual pressed or rolled fingerprint impressions. Additionally, this database did not have any id tags or numbers that could be used for recognition purposes.

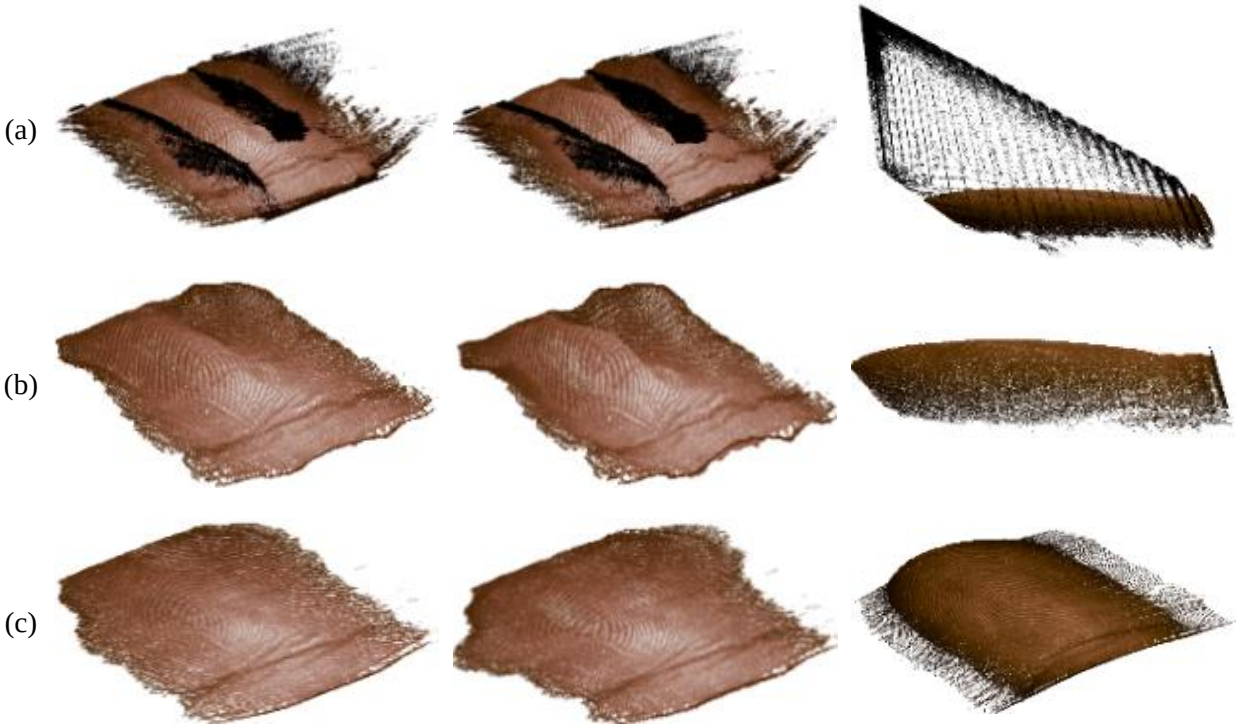


Figure 7.12: Examples of Biometric 3-D unrolling. (a) Shows the 3D input image captured using FLASHSCAN3D [45] (with reconstruction noise), (b) shows the 3D images after noise (anomaly) removal, and (c) shows the unrolled 3D images (output).

Table 7.3: Public domain palmprint database.

Database name	Database statistics	Type
(PolyU) Multispectral Palmprint Database [423]	Multispectral palmprint database Subjects: 250 subjects, Total images: 6000 images	2-D
Tsinghua 500PPI Palmprint Database (THUPALMLAB) [423]	High resolution database Subjects: 80 subjects, Total images: 1280 images	2-D
PolyU Palmprint database v1.0 [414, 424]	Subjects: 177, Total images: 3540	2-D* and 3-D
LPIDB v1.0 [425]	Total palms: 100, Total images: 380	Latent
IIT-D Touchless Palmprint database V1.0 [426]	Subjects: 235, Total images: 3290	2-D

*2-D contactless and its corresponding database

2. Palmprint-Databases

Table 7.3 describes most of the publicly available palm-print databases. Although there are a few public domain palm-print databases, they do not contain both 2-D contact-based (or latent) and 3-D palm images. Therefore, there is a dire need for a versatile palmprint database with 2-D latent or digital scanner-based images, 2-D contactless images, and 3-D images. The Hong Kong Polytechnic University - Contact-free 3-D/2-D Hand Images Database version 1.0 [414, 424] is used to conduct experiments for this section. The database contains 1770 frontal hand scans (range and the corresponding color images) acquired from 177 human subjects.

3. 3D Forensic Synthetic Database

It contains 2780 synthetic 3D forensic images, including 1440 fingerprints, 1280 palmprints, and 60 lip print samples. More details are provided in Chapter 8.

4. 3D Fingerprint Unrolling

The 3D images are filtered to remove anomalies (result shown in Figure 7.12 b)). Then, the filtered 3D images are smoothened, and the color and texture are captured. Next, the smoothened image is unrolled to a plane surface, and the texture and color are fused. All the images in the database were successfully unrolled and resampled into an equivalent 2D fingerprint based on the unrolling algorithm described in section III. The output of the unrolling phase is shown in Figure 7.12 (c). Visual inspection of the unrolled images demonstrates the success of the algorithm.

5. 3D palmprint unrolling

There are other state-of-the-art methods in this field to perform the quantitative and qualitative comparisons. Figure 7.13 shows the experimental results of the proposed 3-D palm unrolling method. The 3-D input images were arbitrarily chosen to display, as shown in Figure 7.13(a).

Figure 7.13(c) shows that the region of interest is first determined, and its texture is extracted. The 3-D surfaces are then further unrolled and fused with color and texture to obtain the results shown in Figure 7.13(d).

Table 7.4: Comparisons of lengths of principal lines before and after unrolling.

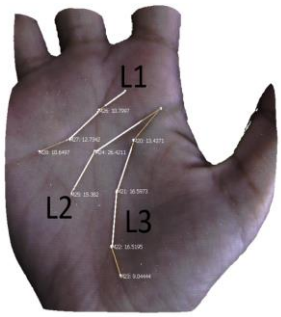
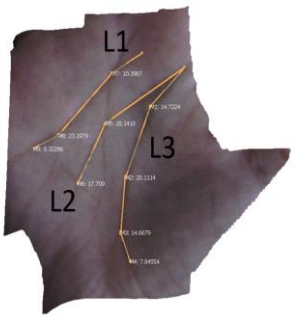
			
	Length of the biometric lines (2-D)	Length of the biometric lines (2-D Unrolled)	Difference ~
L1	55.59834	57.34724	1.7489
L2	41.8031	45.8508	4.0477
L3	34.1836	41.91746	7.73386

Figure 7.13 (b) shows that the 3-D palmprint inputs have different depth distributions. Initially, the ratio of maximum depth to a minimum depth of the displayed 3-D images were 39, 27, and 22.5. After unrolling, the ratio changed to 4.5, 3, and 4.6, respectively, which can also be verified in Figure 7.13(e). This proves the algorithm's robustness in unrolling 3-D images of different depths to their equivalent texture depth. Table 7.4 describes these changes more factually and visually. Minutiae or image feature-based matching algorithms depend on the point features [427]and the relative distance between its neighboring features. To accurately match 3-D palm prints with latent/contact based palm-prints, it is essential to model them prior to matching.

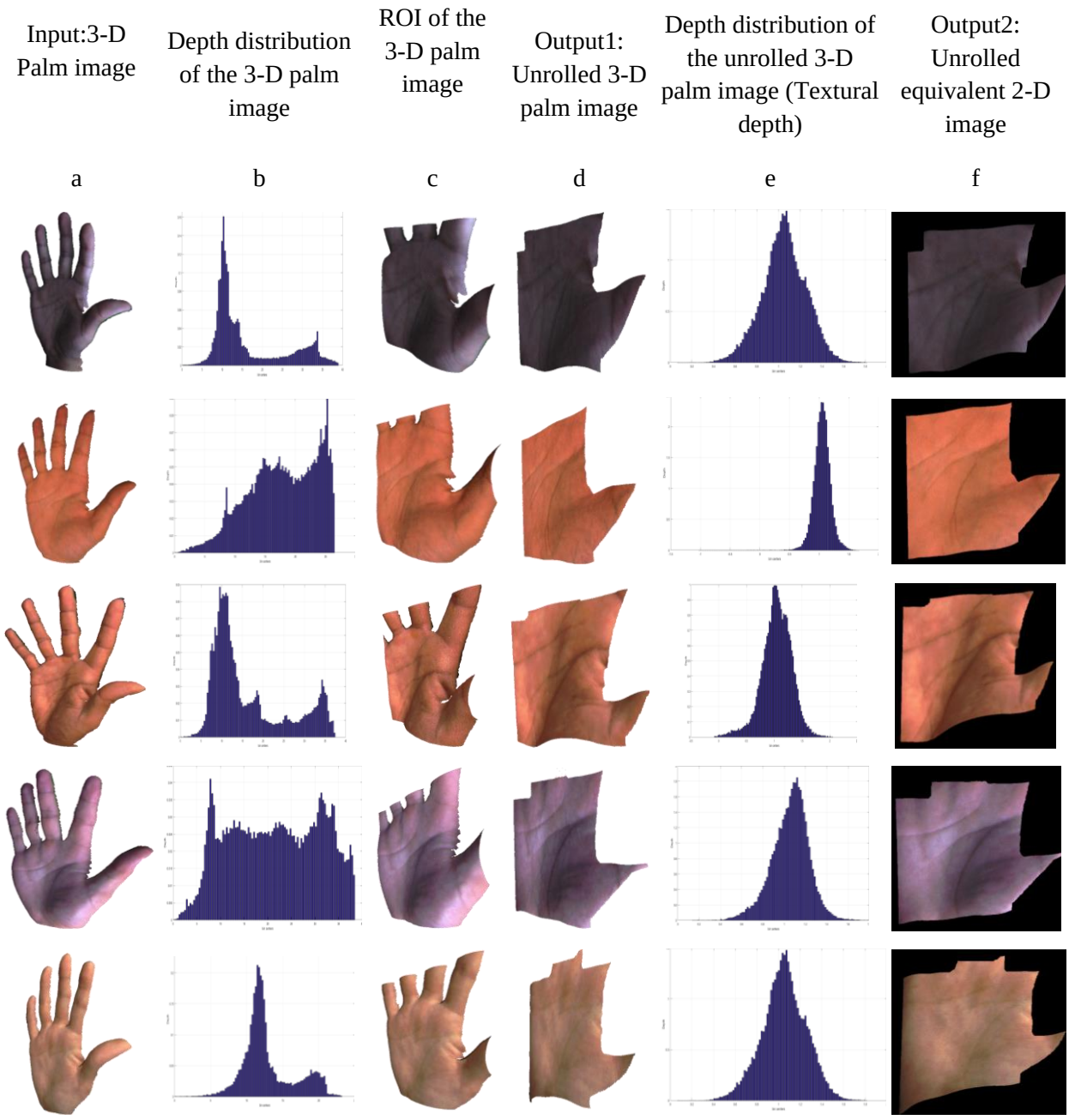


Figure 7.13: shows (a) the input 3-D palm image[414], (b) input 3-D palm depth distribution, (c) the 3-D region of interest of the palm, (d) the unrolled 3-D image, (e) the depth distribution of the unrolled 3-D palm and (f) the unrolled equivalent 2-D image. *Note: The images are not to scale.

6. 3D forensic unrolling

Figure 7.14, Figure 7.16, and Figure 7.15 show the results of the database synthesis in the first three rows of each figure, and the last row of each figure shows the results of the unrolling algorithm. The algorithm causes few distortions and stretching effects when the depth variations are relatively high. The distorted images are still viable for manual feature extraction. For verification and identification purposes, these unrolled images can be further enhanced, filtered, and binarized with an AME/EME [350, 428] based feedback system to yield biometric quality images. Feature-based matching algorithms that incorporate image features [415]with biometric features can be used to provide better recognition results.

Although the output images have clear color pixels, this may not always be the case. An actual 3-D forensic print may not have a higher resolution and quality. This, in turn, may or may not produce results of higher quality and precision. Another approach to boost the performance of the software is to combine the features extracted automatically with the markups of different examiners.

Figure 7.14: Shows the depth fused biometric image and the unrolled 3-D/2-D fingerprints.

Input: 3-D biometric with texture	
Output: Unrolled 3-D/2-D biometric print	

Figure 7.15: Shows the depth fused biometric image and the unrolled 3-D/2-D lip prints.

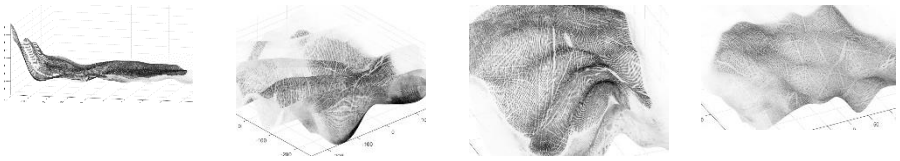
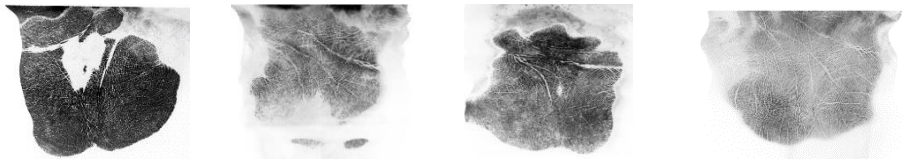
Input: 3-D biometric with texture	
Output: Unrolled 3-D/2-D biometric print	

Figure 7.16: Shows the depth fused biometric image and the unrolled 3-D/2-D palm prints.

7.3 Mosaic Pressure Simulation (MPS)

This process is performed to replicate the distortions that occur during the acquisition of a 2D fingerprint. Algorithm 7.1 describes the general method of simulating pressure on the 3D finger image:

Algorithm 7.1: Pressure Simulation Technique for simulating pressure on the 3D finger image.

1. Map the unrolled fingerprint to the shape of a finger
2. Provide the number of faces
3. Determine the geometry of the 3D fingerprint
4. Determine the location of the face endpoints
5. Calculate the surface length and depth of each region
6. Calculate the chord length from the data obtained
7. Calculate the pressure simulated depth

7.3.1 No-Reference Depth Calculation

This stage can be skipped if the original 3D finger image is not deformed. The amount or lack of deformity can be determined by measuring the standard deviation of the 3-D finger within the finger boundaries. Otherwise, the unrolled flat finger image is mapped to the shape of a finger by determining the geometry of the unrolled 3D image. The distance between the first and last point for a given slice provides the approximate diameter of the 3D finger. The depth is calculated using equations 7.14 to 7.18. Let X and Z be the surface and new depth coordinates, and D and h be the approximate width and height of the finger. Parameter t describes the ratio of depth at the top of the finger to the bottom of the finger.

$$\Delta X = |X(i,j) - X(i,1)| \quad 7.14$$

$$D_1 = D - D * t \quad 7.15$$

$$b = (D - D_1)/log\ (h) \quad 7.16$$

$$D_{11} = D_1 + b * log(h_i) \quad 7.17$$

$$Z(i,j) = \sqrt{D_{11}^2/2 - \Delta X^2} \quad 7.18$$

Figure 7.17: Shows the pressure simulation lines. Each line visualized the point of contact while pressure was applied. Pressure will be simulated along the normal of each line. Increasing the number of slices will produce a natural rolling effect but at the cost of a higher computational cost.

7.3.2 Pressure Simulation on Cross-Sections Of 3D Image

The rolling pressure can be simulated once a non-deformed 3D finger image is obtained. This process requires the knowledge of the geometrical parameters of the 3D image, such as (i) the width of the finger, i.e., the distance between the first and last point of the model (approximated,

in case of post-mortem fingerprint), (ii) the number of segments, (iii) total number of points that exist along that segment, (iv) the surface length (i.e., from one end to the other end) and the maximum depth of each segment is determined, and (v) the length of the chord for each segment is calculated, which will act as a reference scale. Figure 7.17 provides an illustrative example of the direction of the pressure applied to the finger.

The application of pressure is mimicked by changing the depth of the points in the selected face with the reference scale. A point in the face, which has a maximum depth $Z\left(\bar{\bar{\imath}}, a(k)\right)$Is chosen as the starting point.

The new pressure simulated depth coordinates are calculated using the distance between successive surface points and the reference scale as shown in equation 7.19.

where the count is the location of the current point with respect to the segment, c and s are the

$$\tilde{Z}(i,j) = Z\left(\bar{\bar{\imath}}, a(k)\right) + \sqrt{(c(k) - count * \bar{c}(k))^2 - (s(k) - count * \bar{s}(k))^2} \quad 7.19$$

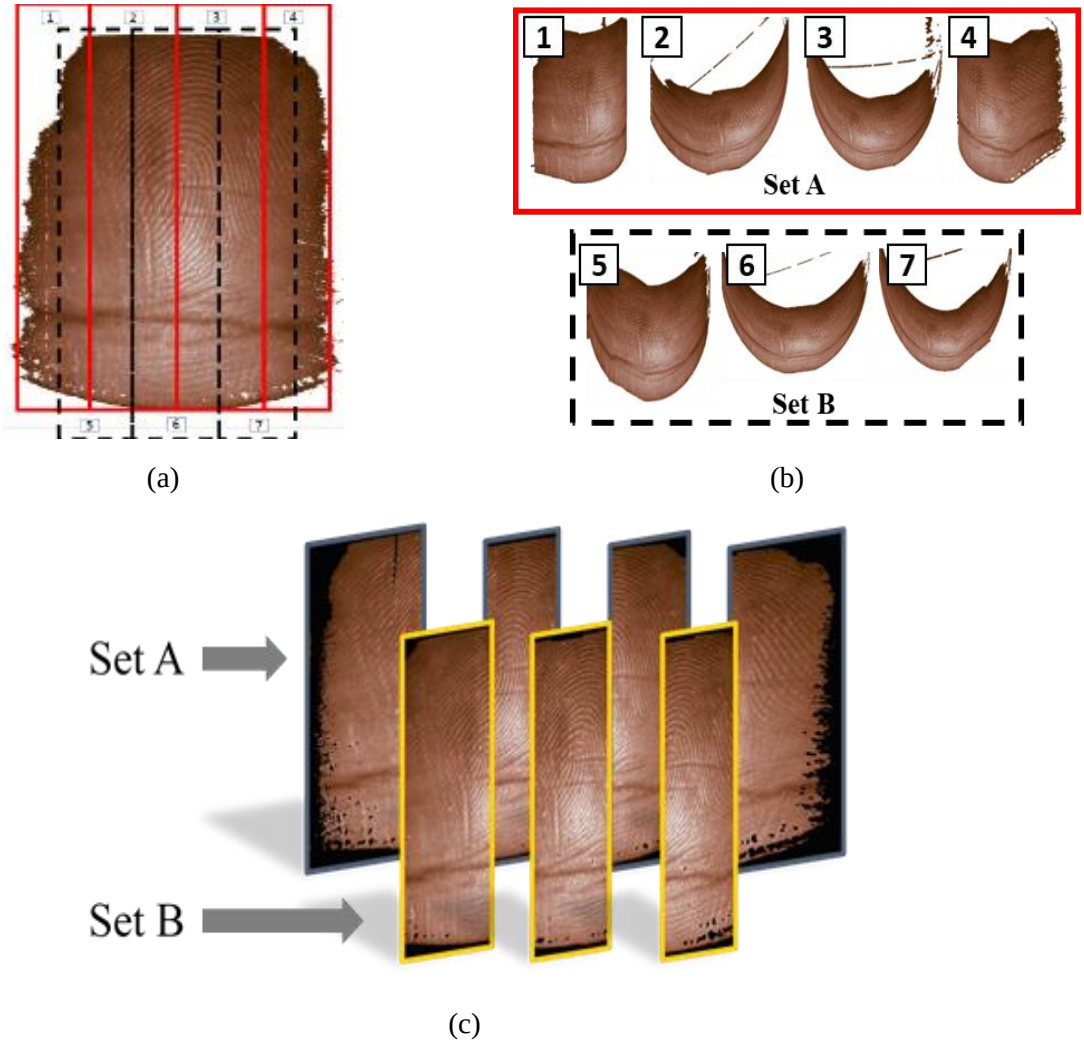
chord and surface information, respectively

This process is repeated for all the segments. Finally, a 2D image of each segment, including partial neighboring faces, is captured for final image stitching.

7.3.3 Mosaicking Or Image Stitching

The final stage of the algorithm includes a mosaicking process that combines all the pressure-simulated faces to obtain one unrolled image. It can be achieved by blending the input images into a single mosaic or integrating the input images' feature sets. The alpha-trimmed correlation approach developed by Rao [415] is used to perform multi-finger image mosaicking. Since major portions of these images will have the same texture, Scale-Invariant Feature Transform (SIFT)

(20) can be used for feature extraction to obtain correspondences between these images. Matching is performed by comparing these feature points.



(a) (b)

(c)

Figure 7.18: Pressure simulation and slice extraction. (a) Shows the areas where the pressure will be simulated. Areas in red will form Set A, and areas in dotted black will form Set B, (b) shows the pressure simulated of set A and set B, and (c) shows the simulated pressure segments of set A and set B. These images are mosaicked together using the alpha-trimmed correlation method. Note: This order is one of the many combinations used. These smaller sets have been chosen for visual and explanation purposes.

Generally, errors in the overlapping images lead to geometric misalignments and photometric differences. The most critical and final stage in producing a perfectly mosaicked image is image blending. This process will mosaic the pressure-simulated 3D image, obtaining an unrolled 2D

fingerprint, which will mimic the pressure distortions produced on a 2D fingerprint while using a 2D scanner. These two sets are mosaicked together to get one final unrolled image.

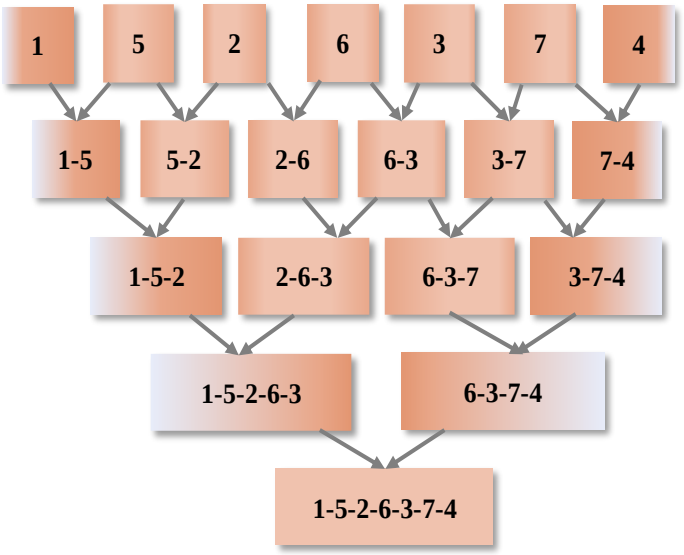


Figure 7.19: Shows one of the combinational methods involved in mosaicking the pressure simulated images. These images are mosaicked together using the alpha-trimmed correlation method [415].

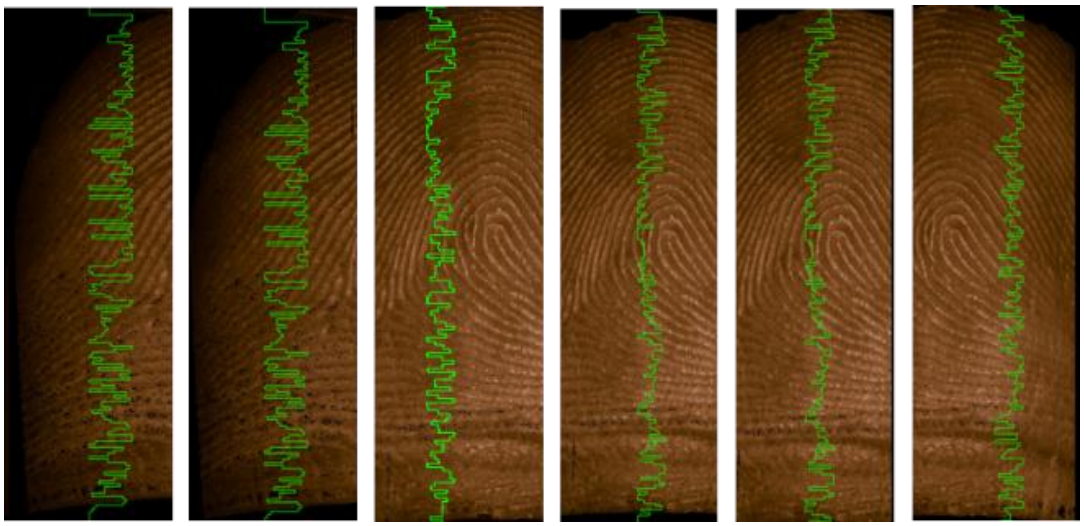


Figure 7.20: Different stages of the mosaicking algorithm for two sample images, (a) shows the region of interest and seam lines for different pressure images.

7.3.4 Pressure Simulation

For the number of faces, N=4, four sections of the 3D image (Figure 7.18 (a)) were individually pressed and named as images 1,2,3, and 4 and collectively called SET A (shown in Figure 7.18 (b)). Furthermore, three more images were created with a shared pressed region with at least two images from SET A. They were named as 5, 6, and 7 and are part of SET B (shown in Figure 7.18 (b)). Figure 7.18 (c) shows the cross-sections of the pressure simulated images that will be stitched together. A similar process was performed for N=5 and 6.

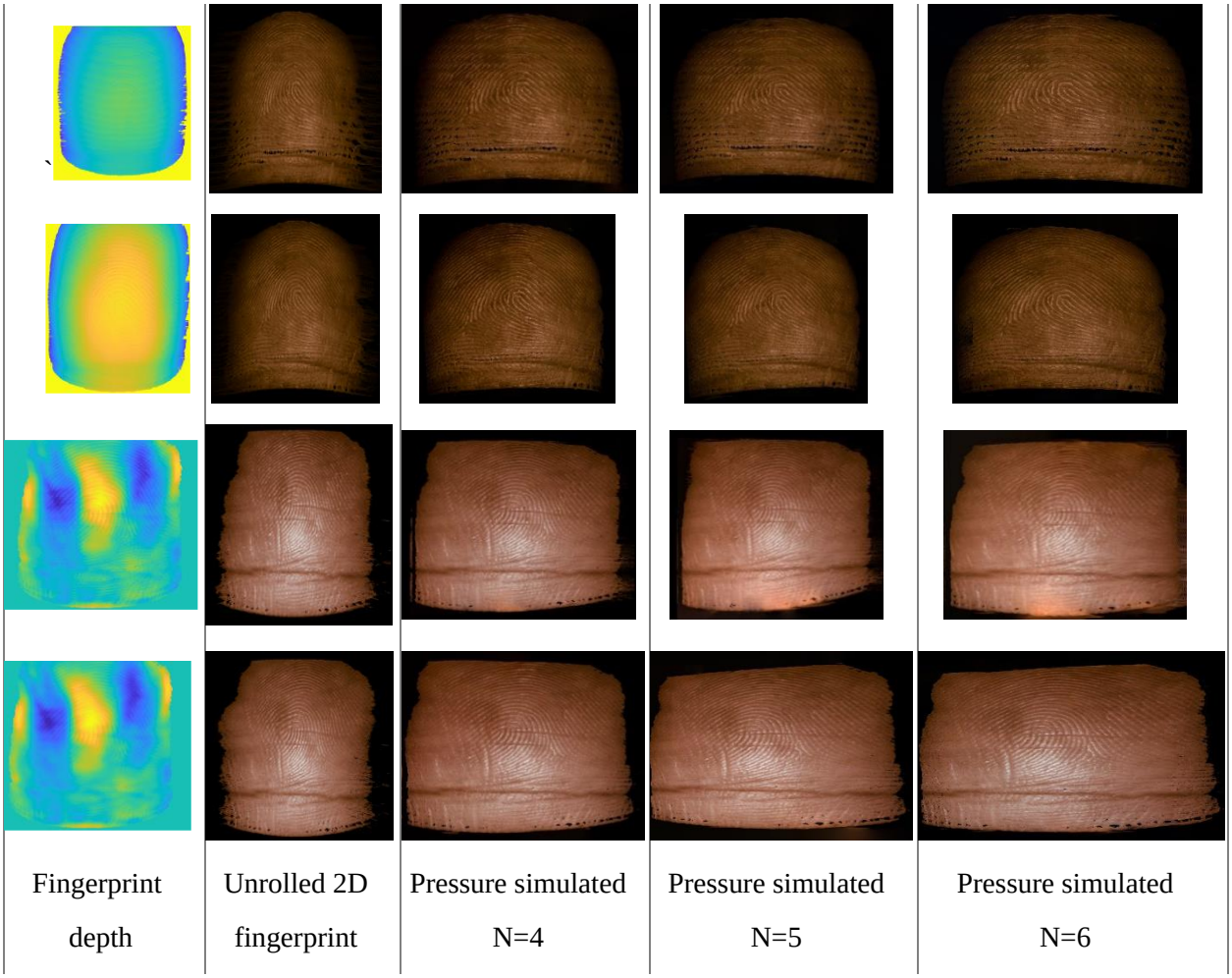


Figure 7.21: Shows the output image of the unrolling algorithm and pressure simulation with different parameters, where the first column shows the actual fingerprint depth, the second column shows the unrolled fingerprint image, and the remaining columns show the pressure simulated images with different values for the number of faces (N).

7.3.5 Mosaicking

Figure 7.19 shows the algorithm's order for mosaicking multi-finger 3D images. For instance, images 1 and 5 are stitched to obtain images 1-5. The final image, '1-5-2-6-3-7-4', is an unrolled fingerprint image that will have evenly incorporated all the pressure simulations. The mosaicking algorithm determines the region of interest between the images and then obtains the best-fit seam line through alpha-trimmed correlation (Figure 7.20 (a)). Finally, the images are mosaicked together to obtain a single image.

Figure 7.21 shows some of the results of unrolling and pressure simulation. As the number of faces increases, the amount of pressure applied increases along with the algorithm's complexity. Applying pressure and mosaicking can be repeated for different pressure parameters. The number of faces that can be pressed can also be changed depending on the user's requisites. With the exception of cases in which there is considerable injury or deformity of the fingerprint, the results of the MPS can be used for automated/manual biometric verification/authentication. MPS can be applied to ordered and unordered point clouds [429].

7.3.6 Evaluation

Numerous factors affect the performance of a fingerprint recognition system; however, the image quality of the fingerprint features is the most important part that can affect the overall performance of a biometric system. The MPS output images are subjected to various distortions and changes during capture, pre-processing, unrolling pressure simulation, and image mosaicking. Research has shown that such operations deter the visual and structural quality of the images [430]. Thus, this sub-section evaluates the visual and structural quality of the images using no-reference (NR) based measures.

Image enhancement measures based on Weber's and Michelson contrast law- Measure Of Enhancement or Measure Of Improvement [431], (EME) [346], Michelson Law EME (AME) [347], SDME [432, 433], and No-reference color measures [350] have been previously proposed. The definitions of these measures are listed in Table 7.5.

Table 7.5: No reference Quality Measure Definitions

$AME_{k1,k2} = -\frac{1}{k_1 k_2}\sum_{k=1}^{k_1}\sum_{l=1}^{k_2} 20\ln\left(\frac{I_{max;k,l} - I_{min;k,l}}{I_{max;k,l} + I_{min;k,l}}\right)$
$SDME_{k1,k2} = -\frac{1}{k_1 k_2}\sum_{k=1}^{k_1}\sum_{l=1}^{k_2} 20\ln\left(\frac{I_{max;k,l} - 2\,I_{center;k,l} + I_{min;k,l}}{I_{max;k,l} + 2\,I_{center;k,l} + I_{min;k,l}}\right)$
$EME_{k1,k2} = \frac{1}{k_1 k_2}\sum_{k=1}^{k_1}\sum_{l=1}^{k_2} 20\ln\frac{I_{max;k,l}}{I_{min;k,l}}$
NR color measure= c1 * colorfu lness + c2 * sharpness + c3* contrast

The definitions for colorfulness, sharpness, and contrast can be found in [350]. The magnitude of the score describes the quality of the image. NFIQ is the first publicly available fingerprint quality assessment tool (provided by NIST) [434]. NFIQ links image quality to operational recognition performance [434]. A lower NFIQ score indicates higher quality. The MPS output images were subjected to the afore-mentioned standard image error measurement techniques, and the results are tabulated in Table 7.6. Based on the scores, it can be inferred that the MPS output images are generally better than the original 2-D equivalent fingerprint (unrolled) in terms of quality. Furthermore, the average number of minutiae in the fingerprint images was calculated using the "MINDTCT" application [435] from the NIST Biometric Image Software (NBIS). The results are shown in Table 7.7. For pressure simulation, parameters N=4, 5, and 6, the average number of minutiae increased by 51.42 %, 17.14%, and 40.00%, respectively.

Table 7.6: No reference Quality Measure Results for the MPS Images. The unrolled pressure mosaicked images are evaluated using no-reference measures.

Measure	Unrolled N=0	Pressure simulated N=4	Pressure simulated N=5	Pressure simulated N=6
AME (mean)↑	16.6687	21.9314	23.2027	22.7161
SDME (mean)↑	29.3698	35.6478	36.9401	36.4427
EME (mean)↑	4.6339	2.4368	2.2015	2.2065
NR color measure (mean)↑	1.0176	0.6376	0.5962	0.6252
NFIQ (mean)↑ [436]	3	2.75	3.25	3.25

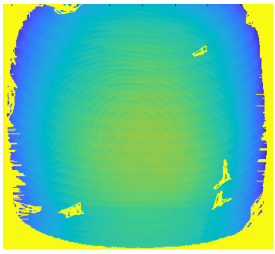
Note: Total number of images used =33

Table 7.7: Average Number of Minutiae Detected (for Minutiae quality > 0.1). The number of viable minutiae points detected by the state-of-the-art MINDTCT program increased with higher pressure simulation.

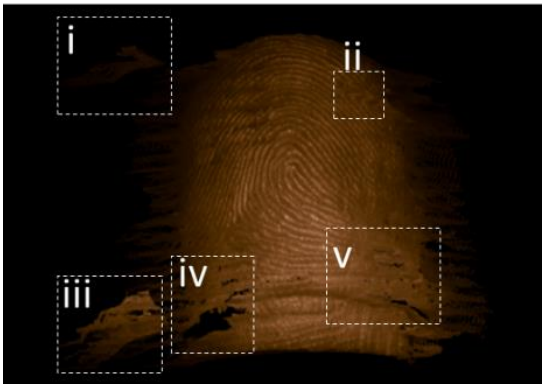
Total images=33	Unrolled N=0	Pressure simulated N=4	Pressure simulated N=5	Pressure simulated N=6
Average number of minutiae (MINDTCT) [436]	35	53	41	49

Although there are a few fingerprint unrolling algorithms, comparing them with the proposed technique seems unreasonable as the state-of-the-art techniques were not designed for post-mortem 3D unrolling. For instance, the algorithm developed by Chen [383] requires a dense representation of the fingerprint. However, when the algorithm (Re-implemented by the author) is applied to a post-mortem fingerprint with holes, it fails, as shown in Figure 7.22 (b). This is because a hole is usually represented by X = 0 in the world space. One of the limitations of the proposed method is the stretching effect which can be visualized in Figure 7.22 (c).

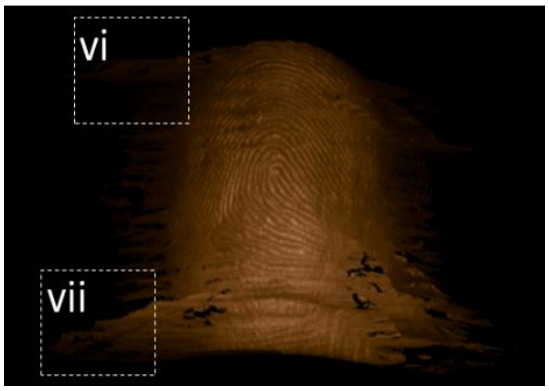
A small dataset is not sufficient to draw conclusive results. However, there is no pre-existing post-mortem dataset, and there are additional hurdles to overcome in terms of consent and access when trying to obtain data from deceased individuals. A total of thirty (normal or ante-mortem 3D) + three images (deformed or post-mortem 3D) were processed using the proposed method.



(a) Depth Map of the artificially generated input 3D post mortem fingerprint



(b) Generic non-parametric unrolling method (Re-implementation of [383])



(c) Proposed unrolling method

Figure 7.22: Qualitative comparison of unrolling algorithms; regions i and iii show disconnected pieces of the skin; regions ii and v show that the unrolled skin has covered the original holes; region iv shows that the holes have stretched; vi and vii show the stretching effect of the proposed unrolling method.

Note: The input image was artificially created and was not used in the evaluation stage described earlier.

7.4 Conclusion

This chapter further attempted to solve some of the most challenging problems in forensic-biometric science, namely, digital extraction of forensic print. It also attempted to solve one of the most challenging problems in 3D processing, namely, real-world 3D image modeling. To this end,

a suite of 3D processing algorithms, including a novel unrolling and Mosaic Pressure Simulation (MPS) algorithm, was proposed.

The proposed unrolling technique enables interoperable recognition of two-dimensional and three-dimensional biometric modalities. This is very important because most biometric data collected worldwide is obtained via contact-based sensors.

The unrolling algorithm extracted the region of interest and texture, unrolling the 3-D image non-parametrically and then applying the texture to obtain the 2-D/3-D equivalent unrolled palm-print images. The visual results show that the technique models the 3-D fingerprint and palm-print images to a textural depth. This technique mapped 3-D surfaces to a 2-D plane to bridge the problems of identification/verification between objects of different input forms.

This chapter also introduced a new 3-D multi-modal database created by synthesizing the depth of each biometric modal database. Extensive computer simulations were performed on multiple datasets. The results quantitatively and qualitatively show that the technique models the 3-D images back to the 2-D image with minimal distortions.

MPS comprises innovative 3D pressure simulation and 2D mosaicking algorithms, combined with other processing techniques such as 3D filtering, 3D non-parametric unrolling, and 3D resampling. MPS effectively solves the interoperability problem of authenticating 3D and 2D fingerprint images by producing realistic images. It can be used to solve the problem of rolling pressure distortions in fingerprints. The algorithm's robustness was tested by unrolling fingerprints with different depth constraints. Computer simulations demonstrated that MPS could be a useful tool to unroll post-mortem fingerprints for identification purposes. Consequently, MPS can be used to achieve reliable ante-mortem/post-mortem recognition. MPS can be used as an intermediary step

to find correspondences between 2D and 3D fingerprints until a time comes when 3D to 3D matching is entirely feasible.

7.5 Acknowledgement

The authors would also like to thank FlashScan3D, LLC for providing 3-D and 2-D input database.

Chapter 8: Comprehensive Multi-Modal, Multi-Dimensional Datasets for AI

Abstract

The preceding chapters discussed and developed several novel approaches to processing multidimensional, multimodal data. However, the primary research challenge at the moment is a lack of large, balanced, and versatile datasets suitable for building, training, and deploying neural networks. Three multi-modal and multi-dimensional datasets are developed to solve some of these challenges. Dataset 1 includes the Tufts Face Database (TDFACE), the most comprehensive publicly available face database globally. Dataset 2 includes Tufts I-Drive (TID), an open-source dataset collected in various driving scenarios. TID will be the most comprehensive multi-modal dataset set in driving conditions. Dataset 3 includes the first-of-its-kind synthetic 3D forensics database for biometric applications.

Index terms: Tufts Face Database; computerized face sketches; thermal; 3D; near infrared; face recognition, cross-modality, Tufts I-Drive, driving, IR, NIR, forensics, synthetic, eye-tracking.

Publications

Panetta, Karen, Qianwen Wan, Sos Agaian, Srijith Rajeev, Shreyas Kamath, Rahul Rajendran, Shishir Paramathma Rao et al. "A comprehensive database for benchmarking imaging systems." IEEE transactions on pattern analysis and machine intelligence 42, no. 3 (2018): 509-520.

Rajeev, Srijith, Shreyas Kamath KM, Karen Panetta, and Sos S. Agaian. "Forensic print extraction using 3D technology and its processing." In Mobile Multimedia/Image Processing, Security, and Applications 2017, vol. 10221, p. 102210L. International Society for Optics and Photonics, 2017.

Accurate cross-modality face matching is necessary due to the increasing variety of imaging sensors in different real-life scenarios, such as depth, optical, capacitive, and thermal: for example, night vision cameras can provide more information in low/no light military applications.

Table 8.1: Advantages and limitations of popular imaging sensors.

Characteristics	Thermal	Near-Infrared	3D	Optical
Visibility in low/no light	yes	yes	yes	no
Record image through translucent obstacles	yes	no	no	no
Provide color content information	no	no	yes	yes
Display surface temperatures of solid objects	yes	no	no	no
Distinguish objects at varying distances	no	yes	yes	yes
Image quality	low	medium	low	high
Presence of noise	medium	medium	high	high
Commercial cost	high	high	high	low

Such cross-modality matching algorithms need to be extensively tested and validated to ensure their robustness in recognizing faces across images from advanced sensors and traditional surveillance cameras [437]. Table 8.1 summarizes the advantages and limitations of different sensor technologies.

This chapter introduces three multi-modal, multi-dimensional datasets. Section 8.1 describes other publicly available face databases and introduces the Tufts Face Database. Section 8.2 describes other publicly available driving databases and introduces the Tufts Driving Dataset “Tufts I-Drive (TID).” Section 8.3 introduces a synthetically generated 3D Forensic Database. Finally, Section 8.4 summarizes the contributions of this work and discusses the future directions.

8.1 Dataset 1: TDFACE-Tufts Face Database

Tufts Face Database is the most comprehensive, large-scale (over 10,000 images, 74 females + 38 males, from more than 15 countries with an age range between 4 to 70 years old) face dataset that contains seven image modalities: visible, near-infrared, thermal, computerized sketch (CS), LYTRO, recorded video, and 3D images. This webpage/dataset contains the Tufts Face Database three-dimensional (3D) images. The other datasets are made available through separate links by the user.

Cross-modality face recognition is an emerging topic due to the widespread usage of different sensors in day-to-day life applications. The development of face recognition systems relies greatly on existing databases for evaluation and obtaining training examples for data-hungry machine learning algorithms. However, currently, no publicly available face database includes more than two modalities for the same subject. This work introduces the Tufts Face Database, that includes images acquired in various modalities: photograph images, thermal images, near-infrared images, a recorded video, a computerized facial sketch, and 3D images of each volunteer's face. An Institutional Research Board protocol was obtained, and images were collected from students, staff, faculty, and their family members at Tufts University.

Table 8.2: Summarized characteristics of the Tufts Face Database (TDFACE).

Frames	People	Nationalities	Gender	Age Range	Modalities
~10,000	118	15	Male 38, Female 74	4 to 70 years	RGB, NIR,IR, 3D, CS

This database will be available to researchers worldwide to benchmark facial recognition algorithms for sketch, thermal, NIR, 3D face recognition, and heterogamous face recognition. The modalities include different 2D face photographs, computerized facial sketches, thermal face

photos, near-infrared face images, a recorded video, and 3D face images. All six modalities are available for every one of 113 individuals depicted in the database.

8.1.1 Related Work-Face Database

Table 8.3 provides a brief overview of the most popular databases built for researchers to evaluate facial recognition algorithms and provide sufficient training data for machine learning algorithms.

Table 8.3: A summarized literature review of the existing optical (RGB) face database.

Database Name	People	Description
Labeled Faces in the Wild (LFW) [438]	5749	Each face has been labeled with the name of the person pictured.
10k US Adult Faces Database [439]	2,222	This database contains 10,168 natural face photographs and several measures for 2,222 of the faces, including memorability scores, computer vision, and psychology attributes, and landmark point annotations.
NIST Mugshot Identification Database [440]	1573	NIST Special Database 18 is being distributed for use in the development and testing of automated mugshot identification systems.
The Color FERET Database [441]	1199	The standard dataset was created to supply standard imagery to the algorithm developers and supply a sufficient number of images to allow testing of these algorithms.
The CMU Multi-PIE Face Database [442]	337	Subjects were imaged under 15 viewpoints and 19 illumination conditions while displaying a range of facial expressions.
The AR Face Database[443]	126	Images feature frontal view faces with different facial expressions, illumination conditions, and occlusions.
Cohn-Kanade AU Coded Facial Expression Database [444]	100	Subjects range in age from 18 to 30 years. Sixty-five percent were female, 15 percent were African-American, and three percent Asian or Latino.
PUT Face Database[445]	100	Dataset consists of almost 10000 hi-res images of 100 people.

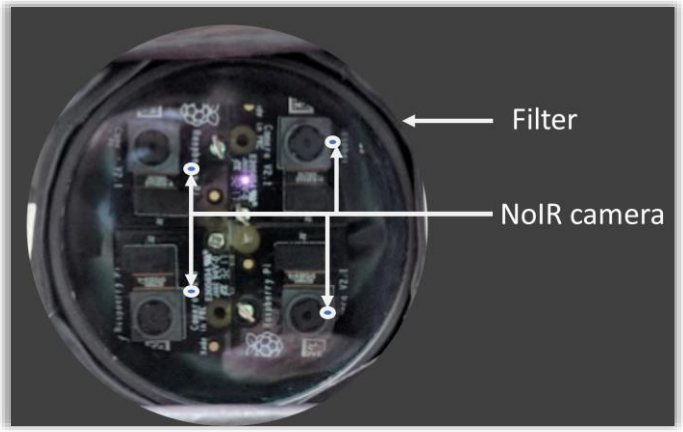
Table 8.4: A summarized literature review of existing non-optical modalities and multi-modal face databases, including optical (RGB), NIR, and 3D.

Modalities	People	Database	Description
NIR	15	The Hong Kong Polytechnic University NIR Face Database [446]	Face images with time variations samples from 15 subjects were collected at two different times with an interval of more than two months.
3D	120	3D_RMA database [447]	3D scans of 120 people were captured in 3 different sessions.
3D	118	Texas 3D Face Recognition Database (Texas 3DFRD) [448]	his database contains 1149 pairs of high resolution, pose normalized, preprocessed, and perfectly aligned color and range images of 118 adult human subjects acquired using a stereo camera.
RGB + NIR	725	CASIA NIR-VIS 2.0 database [449]	It contains 725 subjects imaged by visible (optical) and NIR cameras in four recording sessions.
RGB + NIR	100	Long Distance Heterogeneous Face Database (LDHF-DB) [450]	LDHF database contains both visible (VIS) and near-infrared (NIR) face images at distances of 60 m, 100 m, and 150 m outdoors and 1 m indoors.
RGB + Thermal IR	100	Natural Visible and Infrared Facial Expression (USTC-NVIE) [451]	It contains spontaneous and posed expressions of more than 100 subjects, recorded simultaneously by a visible and a thermal infrared camera.
RGB + 3D	295	The Extended M2VTS Database [452]	It contains synchronized video, speech, and face images.
RGB + 3D	143	The University of Milano Bicocca 3D face database [453]	Collection of multimodal (3D + 2D color images) facial acquisition (98 male, 45 female; a pair of male twins and a baby included).

Image Acquisition

Hardware development: While most hardware and software were purchased off-the-shelf, the researcher developed a quad-camera for RGB and NIR acquisition.

Table 8.5: Hardware specifications of the researcher developed VIS+NIR Quad cam.



1	NoIR camera- No Infrared Filter camera
2	Infrared Ultraviolet Cut Blocking Lens Filter
3	Infrared Filter (Passes infrared rays above 720nm)
4	Raspberry Pi Zero W (Four wireless connected)
5	Infrared Lighting: 850nm Infrared 96 LED light system

Each participant was seated in front of a blue background in close proximity to the camera. The cameras were mounted on tripods, and the height of each camera was adjusted manually to correspond to the image center. The distance to the participant was strictly controlled during the acquisition process. A stable lighting condition was maintained using diffused lights. The following represents the different modalities and sub-datasets of TDFACE.

D_CS: Computerized facial sketches were generated using FACES 4.0, one of the most widely used software packages by law enforcement agencies, the FBI, and the US Military.

TD_3D: The images were captured using a TDFACE researcher-developed quad-camera (an array of 4 cameras). Each individual was asked to look at a fixed viewpoint while the cameras were moved to 9 equidistant positions forming an approximate semi-circle around the individual. The 3D models were reconstructed using open-source structure-from-motion algorithms.

TDIRE(E stands for expression/emotion): The images were captured using a FLIR Vue Pro camera. Each participant was asked to pose with (1) a neutral expression, (2) a smile, (3) eyes closed, (4) an exaggerated shocked expression, and (5) sunglasses.

TDIRA (A stands for around): The images were captured using a FLIR Vue Pro camera. Each participant was asked to look at a fixed viewpoint while the cameras were moved to 9 equidistant positions forming an approximate semi-circle around the participant.

TDRGBE: The images were captured using a NIKON D3100 camera. Each participant was asked to pose with (1) a neutral expression, (2) a smile, (3) eyes closed, (4) an exaggerated shocked expression, and (5) sunglasses.

TDRGBA: The images were captured using a quad-camera (an array of 4 visible field cameras). Each participant was asked to look at a fixed viewpoint while the cameras were moved to 9 equidistant positions forming an approximate semi-circle around the participant.

TDNIRA: The images were captured using a quad-camera (an array of 4-night vision cameras). The lighting condition for NIR imaging was maintained by using an 850nm Infrared 96 LED light system. Each participant was asked to look at a fixed viewpoint while the cameras were moved to 9 equidistant positions forming an approximate semi-circle around the participant.

TDLYTA: The images were captured using a LYTRO ILLUM 40 Megaray Light Field camera. Each participant was asked to look at a fixed viewpoint while the cameras were moved to 9 equidistant positions forming an approximate semi-circle around the participant.

TD_VIDEO: The images were captured using one of the visible field quad cameras. Each participant was asked to look at a fixed viewpoint while the camera was moved around the participant forming an approximate semi-circle.

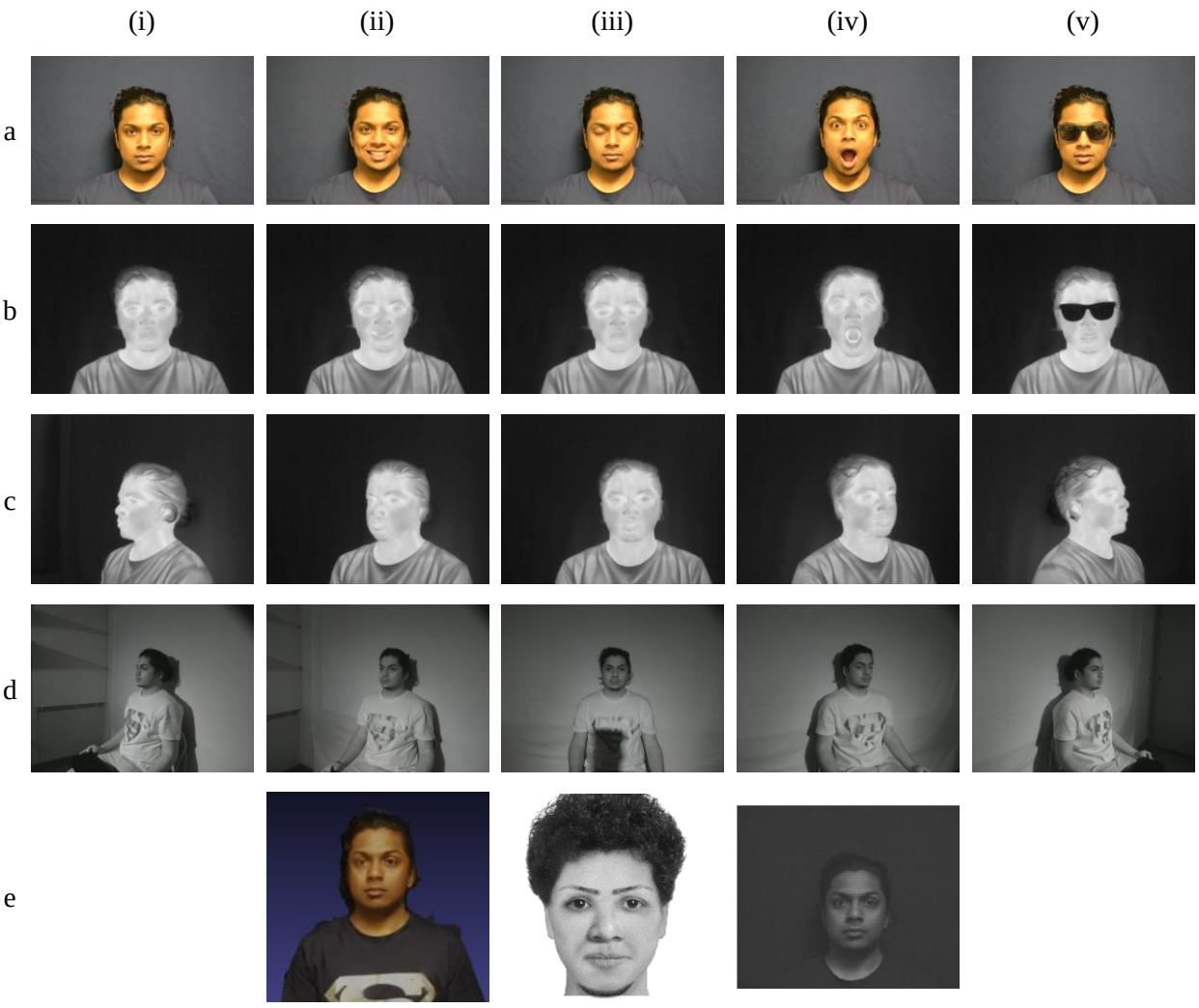


Figure 8.1: Sample set of multi-modal, multi-dimensional images from TDFACE. Row (a) Optical (2D-RGB), Row (b) Thermal (2D-IR), Row (c) Thermal (2D-IR) Around, Row (d) Near Infrared (2D NIR) Around, Panel [e,ii] Three-dimensional (3D-RGB), Panel [e,iii] Computerized sketch, Panel [e,iv] Lytro (3D-LightField RGB-Depth).

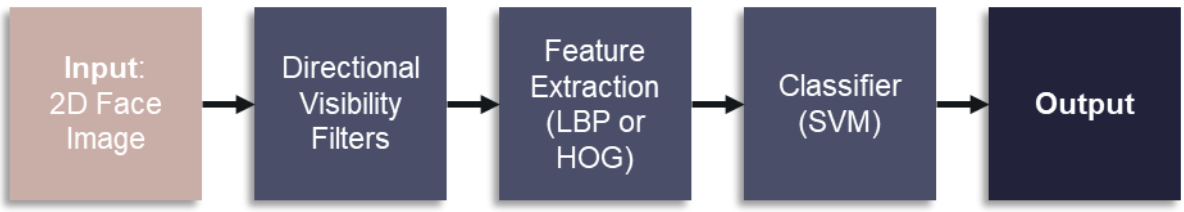


Figure 8.2: Block diagram of the proposed DV Filter based face recognition system.

8.1.2 NIR Face Recognition Evaluation

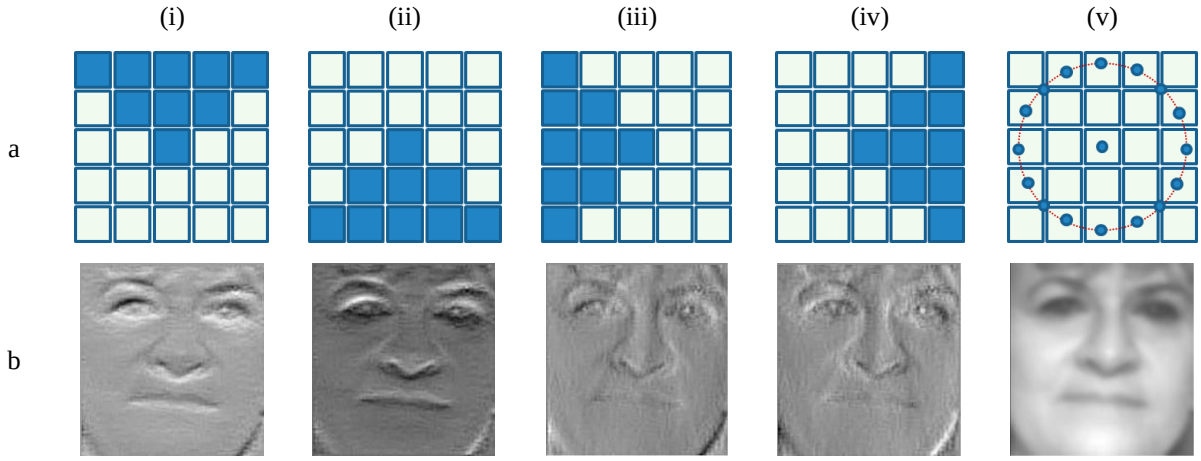


Figure 8.3: Panel [a,i-iv] shows the visualization of the q*uadrant DV filter*. Panel [a,v] shows the visualization of a 5x5 Circular DV filter. Row (b) shows the output of the DV filters.

This section proposes a novel illumination-invariant face recognition architecture. The proposed architecture consists of DV image generation, facial feature extraction, and SVM classification. Figure 8.2 shows a systematic representation of the proposed system. First, a sequence of images is passed through a Directional Visibility (DV) filter. The DV filter is a local direction filter that takes advantage of linear and circular direction directions in a k x k window. This can be visualized in Figure 8.3 (a). The values in the blue sections of the window are used as input to equations 8.1, 8.2, or 8.3. The output values replace the center pixel in the window. Figure 8.3 (b) shows the output of DV filters (stride =1, and kernel size=5) using equation 8.3. The resulting face images encode intrinsic information about the face using only a monotonic transform in the gray tone. The DV filtered images are then processed to extract Local Binary Patterns (LBP) or Histogram of Oriented Gradients (HOG) features. This step compensates for the monotonic transform, yielding an illumination invariant face representation. Finally, classification using SVM is performed.

$$I_{max}/I_{min} \quad 8.1$$

$$(I_{max} - I_{min})/(I_{max} + I_{min}) \quad 8.2$$

$$(I_{lc} - \alpha \times I_{mean})/(I_{lc} + \alpha \times I_{mean}) \quad 8.3$$

where I_{max} is the maximum pixel intensity, I_{min} is the minimum pixel intensity, I_{lc} is the local center pixel, I_{mean} is the mean of the pixels in consideration, and α can vary between 0 to 1

The TDFACE NIR database was randomly split into a 70:30 ratio for training and testing. The performance of the proposed method was computed using several combinations of DV filters with feature detectors, as shown in Table 8.6.

Table 8.6: Quantitative evaluation of DV filter based NIR image face recognition

Method	**Recognition Rate (%)**
Local Binary Pattern (LBP)	30.84
DVfilter (Quadrant) with LBP	35.32
DVfilter (Circular) with LBP	3.98
Histogram of Oriented Gradients (HOG)	92.53
DVfilter (Quadrant) with HOG	93.53
DVfilter (Circular) with HOG	**94.52**

Compared to other combinations, the proposed circular directional visibility filters in conjunction with the HOG detector perform well. This is the case because the circular directional filter tends to combine the majority of relevant face information in each kernel for recognition purposes while reducing high-frequency components. The proposed algorithm has several advantages over other state-of-the-art algorithms, including (1) the introduction of a novel directional visibility-based local feature extractor for generating complementary data types; (2) the elimination of the need for images captured under different lighting conditions to generate a normalizing coefficient; and (3) the elimination of the need for image alignment for recognition. This new method is unique

because it uses a user-designed directional filters technique to process face images before forwarding them to a traditional face recognition system.

8.1.3 TDFACE Contribution

The TDFACE was designed, collected, developed, processed, and published by members of the Vision and Sensing Systems Laboratory, Tufts University. The author's contributions include dataset ideation, sensor development, dataset collection, dataset processing, DV filter analysis, and article writing.

8.2 Dataset 2: Tufts Driving Dataset (Tufts I-Drive (TID) Dataset)

From photos to video, cameras are the most accurate way to create a visual representation of the world, especially when it comes to self-driving cars. Though they provide accurate visuals, cameras have their limitations. Rather than relying on one type of sensor data at specific moments, sensor fusion makes it possible to fuse various sensor suite information — such as shape, temperature, speed, and distance — to ensure reliability. Much like the human brain processes visual data taken in by the eyes, an autonomous vehicle must be able to make sense of this constant flow of information. Self-driving cars try to do this using a process called sensor fusion.

The study of gaze behavior has primarily been constrained to controlled lab environments. Consequently, little effort has been invested in developing algorithms to categorize gaze events while the head is free. This work aims to generate a dataset that captures complex ocular-motor strategies during natural tasks such as driving a vehicle. As an outcome, this research will strive to identify correlations between eye movements and decisions, thus answering the causality problem of Artificial Intelligence (AI). This research will build an eye-movement database of high-resolution sensor data collected across varying environmental and weather conditions. The

database will be named the Tufts I-Drive (TID) database. The data collection setup requires a wearable eye-tracker, thermal sensor, and dashcam mounted on a car driven in motorways. Data of interest include dashcam images and eye-movement data. The target weather conditions include sunny, rainy, and snowy at dusk, dawn, day, and night. The ultimate goal and expected impact of such a dataset, which shall be made public, is to foster advancements in machine perception for autonomous driving technologies.

The dataset, trained classifiers, and evaluation metrics will be made publicly available for download for research purposes, notably facilitating growth in the emerging area of head-free gaze event classification. The files will be stored on Tufts Technology-managed servers. Identifying information such as license plates or people (captured incidentally) will be anonymized.

Table 8.7: Summarized characteristics of the Tufts I-Drive Dataset.

Videos	Frames	Drivers	Weather conditions	Light conditions	Modalities*	Metadata	Roadways
92 (55,200 seconds)	0.5 million	8 6 men 2 women	Normal	Day	RGB	GPS	City
			Rain		NIR	Speed	Suburb
			Snow	Night	IR	Path	Highway
					ET		Countryside

**RGB: Optical camera, NIR: Near-Infrared CMOS sensor, IR: Thermal Infrared, ET: Eye-tracker*

The dataset consists of 444,000 frames. A total of 37 videos were collected, each 10 minutes long. Eight drivers, six men and two women of varying ages from 23 to 50, participated in the data collection. Videos were recorded in diverse weather conditions and light conditions. TID, when published, will be the most comprehensive publicly available multi-modal dataset set in driving conditions. Table 8.7 provides the characteristics of TID.

8.2.1 Related Work-Driving Database

This section summarizes key datasets for the development of self-driving cars. The primary focus is on datasets that have been composed of multi-modal sensor data. A comparison of modalities of the related work is provided in Table 8.8.

Table 8.8: Comparison of modalities of popular driving datasets. Descriptions of these databases are provided in the section.

Database	Ref	Optical RGB	3D	Thermal IR	Eye-tracking
ASTYX HIRES2019	[454]	✓	✓		
BDD100K	[455]	✓			
nuScenes	[456]	✓	✓		
Waymo Open Dataset	[457, 458]	✓	✓		
KITTI	[304, 459]	✓	✓		
Mapillary Vistas collection	[460]	✓			
ApolloScape	[461]	✓	✓		
Lyft Level 5 AV	[462]	✓	✓		
Cityscapes	[256, 463]	✓			
A2D2	[464]	✓	✓		
DR(EYE)VE dataset	[465]	✓			✓
TID		✓	✓	✓	✓

The KITTI [304] dataset and accompanying benchmarks [459] have had a significant impact. Four video cameras (two color, two grayscale), a 3D laser scanner, and a GPS/IMU inertial navigation system were mounted on the data gathering vehicle. KITTI offers 2D, 3D, and bird's eye view

object detection datasets, 2D object and multi-object tracking and segmentation, road/lane evaluation, and raw datasets.

CityScapes [256, 463] is a large dataset to improve the semantic understanding of urban street scenes in 50 German cities. It includes semantic, instance, and dense pixel annotations for 30 classes classified into eight categories. The dataset contains 5,000 images with fine annotations and another 20,000 images with coarse annotations. The data were obtained in 50 cities around Germany and were captured during periods of heavy urban traffic.

The Mapillary Vistas collection [460] provides semantic labeling for urban, rural, and off-road settings. 25,000 extensively annotated street-level images worldwide are included in the dataset. The dataset was captured using various devices, including mobile phones, tablets, and various cameras.

ApolloScape [461] is a big dataset comprised of over 140,000 video frames captured in various locales in China during a variety of weather conditions. In 2D, the recorded data is semantically annotated pixel by pixel, whereas, in 3D, 28 classes are semantically annotated point by point. Furthermore, the collection includes 2D annotations for lane markings.

The Berkeley Deep Drive dataset (BDD-100k) [455] provides 2D bounding boxes and pixel-level annotations for 10,000 images. BDD-100K comprises 100,000 photos with two-dimensional bounding boxes and street lanes, markings, and color identification of traffic lights.

The nuScenes dataset [456], developed by Motional, is one of the largest open-source datasets for autonomous driving. The dataset, which was collected in Boston and Singapore using a comprehensive sensor suite (32-beam LiDAR, six 360° cameras, and radars), contains over 1.44

million camera images that capture a variety of traffic situations and driving maneuvers, and unexpected behaviors. It was gathered during the day and night in clear weather.

The Lyft Level 5 AV Dataset [462] contains nuScene-format camera and LiDAR data. It places a premium on the detection and tracking of 3D bounding boxes. It contains over 55,000 human-labeled 3D annotated frames, a surface map, and an underlying high-resolution spatial semantic map captured by seven cameras and three LiDAR sensors.

Audi released the Audi Autonomous Driving Dataset (A2D2) [464] to assist startups, and academic researchers develop autonomous vehicles. The dataset contains over 41,000 images that have been labeled with 38 features. Along with labeled data, A2D2 provides unlabeled sensor data (390,000 frames) for sequences that contain multiple loops.

Waymo released the Waymo Open Dataset [457], which includes 12 million 3D bounding box annotations on LiDAR point clouds and 1.2 million 2D bounding box annotations on camera frames. It features 1000 20-second scenes equating to 200,000 frames at 10 frames per second per sensor. The data was captured in urban and suburban settings and under a variety of weather and lighting situations.

The current largest dataset with eye-tracking data is the DR(eye)VE [465] dataset. It contains 74 5-minute RGB + Eye-tracking videos, with over 500,000 annotated frames. Labels include gaze fixations, task-specific saliency maps, Geo-referenced locations, driving speed and course complete the released data set.

8.2.2 Multi-modal Sensors of TID

TID includes several multi-modal sensors, including a visible spectrum camera, thermal infrared camera, near-infrared camera, and an eye-tracker. The following describes the sensors and the type of data that were collected.

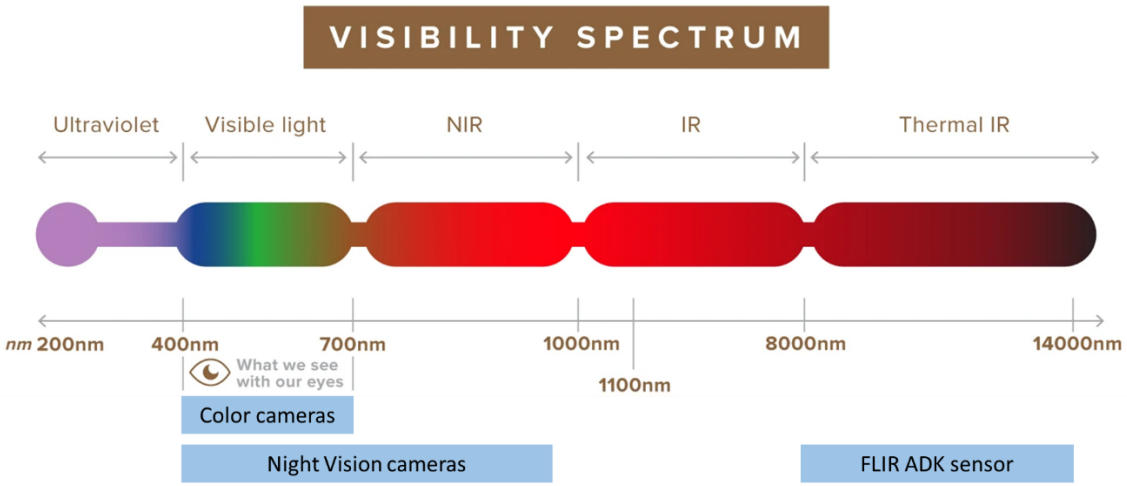


Figure 8.4: shows the spectral wavelengths at which different cameras operate [466].

a) Near-Infrared or Night Vision sensors, Vava Dashcam- Sony IMX307

These sensors are also popularly used as dashcams in vehicles and CCTV cameras.

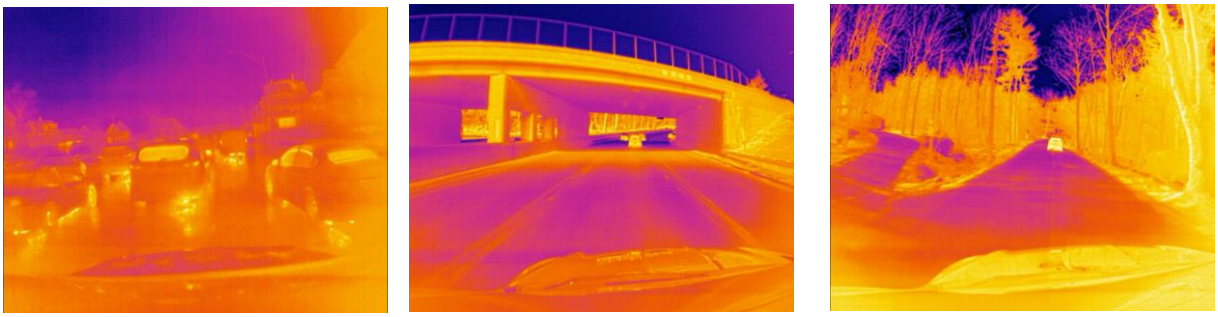
Normal visible camera image Night vision image

Figure 8.5: the left pane shows the view from a regular camera, and the right pane shows the view from a night vision or near-infrared camera [467].

Why are they important? It has excellent low illumination performance, and objects/scenes can be seen clearly under weak light or no light conditions. This camera provides a common view and acts as the base medium for data fusion. The Sony IMX307 sensor realizes high picture quality in the visible-light and near-infrared light regions.

What data do they collect? This camera captures images in the form of pixels in three channels. The dashcam also collects GPS data, accelerometer, and timestamps.

b) Thermal sensor (Thermal Infrared (IR) camera), FLIR Automotive Development Kit (ADK)[468]



(a) (b) (c)

Figure 8.6: Images show example outputs of a FLIR thermal camera. The image pixels are infrared energy [469]. These images are samples taken from the TID database show (a) city traffic conditions, (b) highway conditions, and country conditions.

The FLIR ADK is a cost-effective way to develop the next generation of automotive thermal vision for advanced driver assistance systems (ADAS) and autonomous vehicles (AV).

Why are they important? Thermal infrared cameras are the best sensor technology for pedestrian detection, reliably classifying people in cluttered environments and giving analytics the critical information required for automated decision-making.

Thermal sensors create images from heat, not light, to detect pedestrians and oncoming vehicles regardless of lighting conditions [470]. The FLIR thermal sensors can detect and classify

pedestrians, bicyclists, animals, and vehicles in challenging situations, including total darkness, fog, smoke, inclement weather, and glare, providing a supplemental dataset beyond radar and visible cameras. The detection range is four times farther than typical headlights. When combined with visible-light data and distance scanning data from radar or three-dimensional scanners, thermal data paired with machine learning creates a more comprehensive detection and classification system [470].

What data do they collect? A thermal camera is a non-contact device that detects infrared energy (heat) and converts it into a visual image. Heat sensed by an infrared camera can be measured precisely, allowing for many applications. A FLIR thermal camera can detect tiny differences in heat—as small as 0.01°C—and display them as shades of grey or with different color palettes. The potential uses for thermal cameras are nearly limitless. Originally developed for surveillance and military operations, thermal cameras are now widely used to build inspections (moisture, insulation, and roofing), firefighting, autonomous vehicles and automatic braking, skin temperature screening, industrial inspections, scientific research, and more [471]. The device also collects timestamps.

c) Eye-tracking glasses, 'Pupil Invisible' [472, 473]

Why are they important? This includes the 'Pupil Invisible' eye-tracking glasses developed by Pupil Labs [472]. Pupil Invisible provides robust gaze estimation in any environment, unlike traditional eye trackers. Moreover, it does not need calibration before every usage. The glasses provide an end-to-end gaze estimation pipeline. The user guide can be accessed at [306].

What data do they collect? It includes eye cameras that capture data at 200 Hz with 192x192 px Infrared (IR) illumination. The scene camera (from eye-tracker) captures world view data at 30 Hz

with 1088x1080 px resolution. It has a field of view (FOV) of 82°x82°. The eye-tracking glasses are controlled via a mobile device.

(a) (b)

Figure 8.7: Eye-tracking data sample. (a) shows an example output of a Pupil-lab eye-tracker. the circle in (a) shows the gaze position of the user. (b) the pupil invisible eye-tracker used in this study [473].

How do they work? Head-mounted systems feature near-eye cameras mounted on the test subject's head, recording the eye regions from close-up. Given a portable recording unit, this setup enables the test subject to move freely in space without affecting the cameras' view of the eyes. The front-facing scene camera correlates the obtained gaze data with environmental cues and stimuli. The methodology of Pupil Invisible has been extensively described in this research paper [473].

Table 8.9: Sensors on the Pupil Invisible.

Eye Cameras	200Hz @ 192x192px IR illumination
Scene Camera	Detachable scene camera. 30Hz @ 1088 x 1080px 82°x82° FOV. The square aspect ratio covers the full field of view
Inertial Measurement Unit (IMU)	Gyroscope and Accelerometer @200Hz

Pupil Invisible glasses were intentionally chosen for this project to minimize the risks for the wearer during data collection. Figure 8.7 (b) shows how the Pupil Invisible glasses appear in the real world.

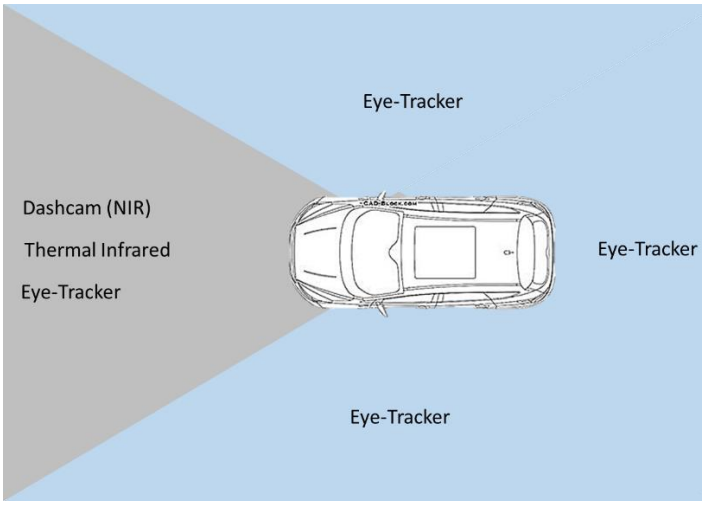


Figure 8.8: A top-view perspective of the range of the sensors with respect to the study vehicle.

8.2.3 TID collection and packaging protocols

1. Procedures Involved

Study design. The proposed data collection study was divided into the following 3 phases:

Phase 1: *Data collection*: The thermal camera was placed on top of the car. The dashcam was placed on the windshield (inside the car). The eye-tracking glasses are similar to reading glasses

or sunglasses with a clear lens. Data was collected at dusk, day, dawn, and night. Video/Image (visible and thermal when applicable) annotations include scenes such as 'coming to a halt,' 'changing lanes-Right/Left,' 'looking at the side-mirrors or rear-view mirrors,' 'braking the car at a red light,' 'yielding,' and 'stopping for pedestrians.' This database also has standard object detection annotations such as vehicles, signs, and pedestrians.

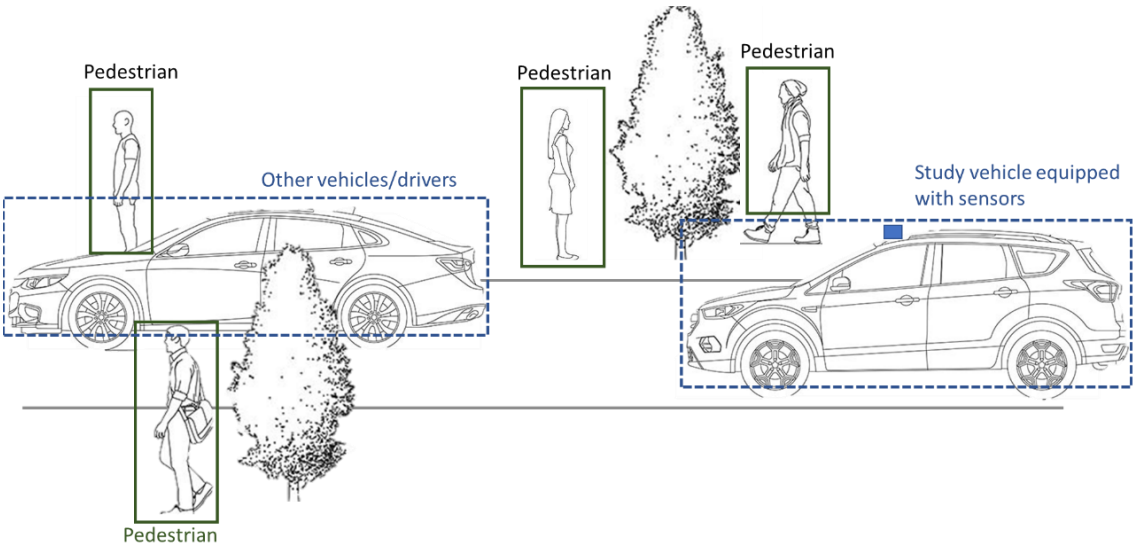


Figure 8.9: Shows the individuals who may be incidentally captured on video/camera in the TID dataset.

Phase 2: *Data anonymization:* The data was anonymized to remove any identifying elements.

Blurring protocol: On images, this can be achieved by hiding features that could lead to a person's identification, such as the person's face, tattoos, and scars. To do this, a 4 step blurring process was followed [474, 475]. Step 1 involves using a detector to identify all the elements that must be hidden. Typical face detectors include Haar cascades, HOG + Linear SVM [476], or deep learning-based face detectors [477]. Step 2 is to extract the region of interest (ROI). Step 3 is to blur the face. This step anonymizes the face. This is achieved using a Gaussian filter. Step 4 is to save this new data and delete the original data. The anonymization pipeline for face is complete. Similar

procedures are performed for other identifying elements in the data source (including license plates [478-480]).

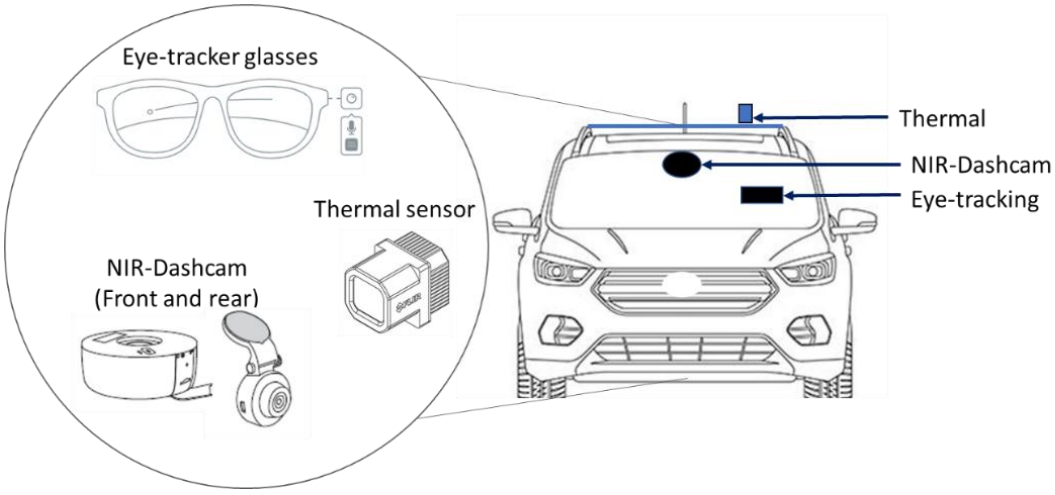


Figure 8.10: Show the sensors that are used in the TID dataset. They were placed on board the vehicle as illustrated above [481] [468] [472, 473] [468] [472, 473].

Eye-tracking outputs include iris images captured from the Infrared (IR) eye cameras. To ensure confidentiality and data privacy is maintained, the captured iris images are processed by applying filters that make the images sufficiently blurry and render them useless for iris authentication purposes, as suggested in [482].

Phase 3: *Data packaging*: The collected data streams have timestamps and are synchronized in time. The data of different modalities are thus fused using synchronization and homography. Videos and annotations include vehicles, vehicle types, humans, traffic signs, potholes, road work, and actions such as taking turns, stopping at the stop sign, yielding, and switching lanes. Real ground truth annotation and synthetic annotation are generated for the dataset, as shown in Figure 8.11. Human annotations were performed for action recognition of the driver based on eye movements.

Table 8.10: Type of data collected and their formats.

Sensor	Data Collected (Formats)			
Vava Dashcam	NIR image/video *(.jpg/.mp4/ .avi)*	GPS information *(.txt, .json, .csv)*	Accelerometer *(.txt, .json, .csv)*	Timestamp *(.txt, .json, .csv)*
FLIR ADK	Thermal LiDAR (IR) image/video *(.jpg/.mp4/.avi)*	Timestamp *(.txt, .json, .csv)*		
Pupil Invisible eye-tracker	Visible Spectrum image/video *(.jpg/.mp4/ .avi)*	Eye-movement data *(.txt, .json, .csv)*	Inertial movements of the wearer *(.txt, .json, .csv)*	Timestamp *(.txt, .json, .csv)*

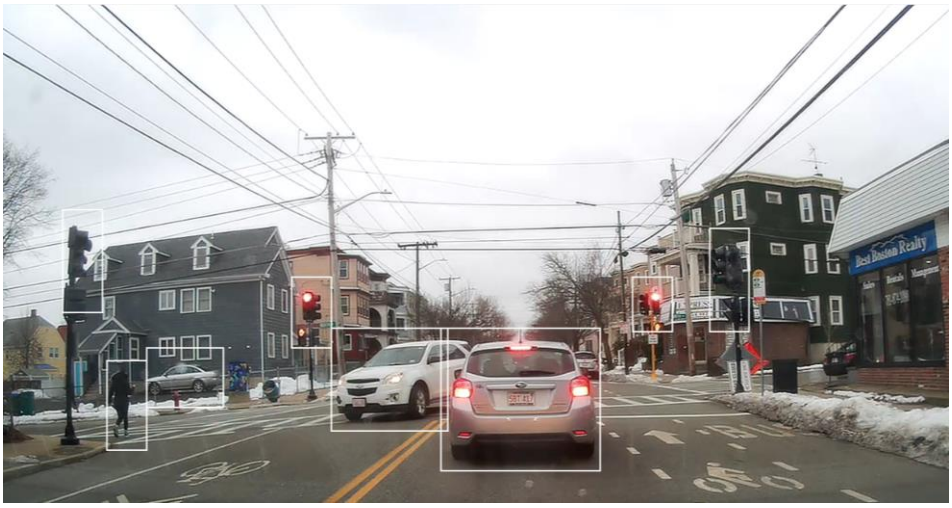
Table 8.11: Annotation types and annotation methods for each modality in the TID.

Modality		**Annotations Type**	**Classes**
Eye-tracker	RGB	Object Detection: *Synthetic*	30
		Semantic Segmentation: *Synthetic*	30
		Fixations ROI/OOI/AOI: *Synthetic*	30
	Eye-tracking data	Actions: *Human annotations*	27
IR-Thermal		Object Detection: *Synthetic*	20
		Semantic Segmentation: *Synthetic*	20
NIR-Dashcam		Object Detection: *Synthetic* Semantic Segmentation: *Synthetic*	30

Note: Respective GPS and Speed data are attached as metadata

Note: Synthetic annotations should be considered as coarse annotations. Future work involves creating fine annotations for the remaining modalities.

(a)



(b)

(c)

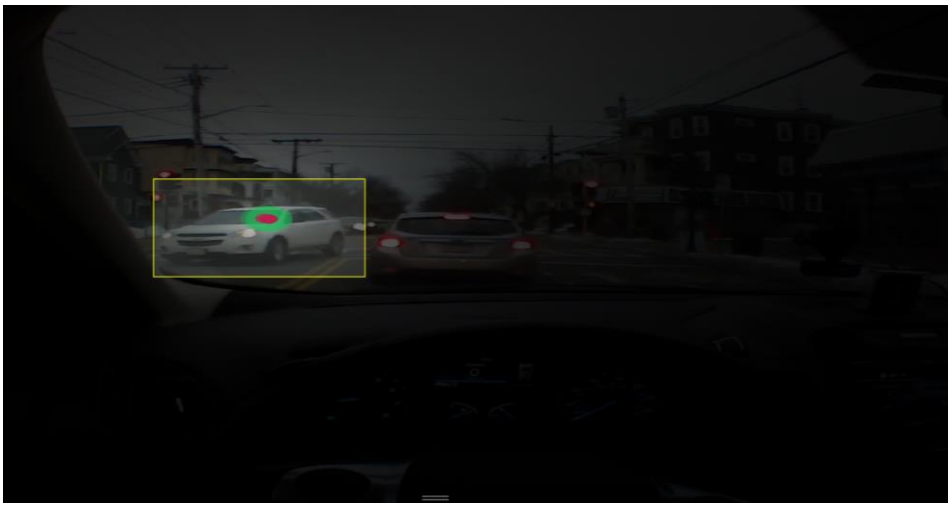


Figure 8.11: Sample frames of Synthetic bounding box annotations, where (a) is captured from the dashcam (NIR), (b) is captured from the thermal IR sensor, and (c) is captured from an eye-tracker. Note: (c) This frame has been post-processed to highlight the region of interest and the fixated object.

8.3 Synthetic Database- 3D Forensics

Biometric evidence plays a crucial role in criminal scene analysis. Forensic prints can be extracted from any solid surface, such as firearms, doorknobs, carpets, and mugs. Prints such as fingerprints, palm prints, footprints, and lip-prints can be classified into patent, latent, and three-dimensional plastic prints. Traditionally, law enforcement officers capture these forensic traits using an electronic device or extract them manually and save the data electronically using special scanners. The reliability and accuracy of the method depend on the ability of the officer or the electronic device to extract and analyze the data.

Furthermore, the 2-D acquisition and processing system is laborious and cumbersome. This can lead to an increase in false-positive and true negative rates in print matching. This section presents a method and system to simulate 3D forensics.

Figure 8.12: Different possible textures that can be used for 3D simulation. From top left-right: smooth(marble), wrinkled, and greasy[420]; bottom left-right: cracked, bumpy, and silky [421].

In order to benchmark the capability of the proposed forensic unrolling algorithm, it has to be tested on a versatile 3-D forensic database. However, due to the unavailability of such a database, the author has synthesized a 3-D forensic database by fusing the texture from Describable Textures Dataset [421] with the RGB information of the images in the FVC 2004 fingerprint database [483].

The Describable Textures Dataset (DTD) is a collection of textural images [421]. It has 47 categories, with 120 images for each category. A sample selection of images is shown in Figure 8.12. For this section, the textural image category was preferred. It was chosen because the textural pixel variations mimic depth changes on a 3-D surface. These images were used to synthesize the depth of the print's surface.

Table 8.12: Different scanners used in collecting the FVC2004 database [484].

FVC2004 Sub-datasets	**Technology**	**Image size**	**Resolution**
DB1	Optical Sensor (CrossMatch V300)	640x480	500 dpi
DB2	Optical Sensor (Digital Persona U.are.U 4000) 3	328x634	500 dpi
DB3	Thermal Sweeping Sensor (Atmel FingerChip)	300x480	512 dpi
DB4	Synthetic Generator (SFinGe v3.0)	288x384	Approximately 500 dpi

1) Database synthesis - Fingerprints

The FVC2004 database consists of 4 datasets acquired using different sensor types, and each of them contains 8 impressions of 10 fingers. Three different scanners and the SFinGE synthetic generator were used to collect fingerprints. Table 8.12 provides the sensors' details, image size, and resolution. Each database has 120 fingers and 12 impressions per finger (1440 impressions) [484].

2) Database synthesis – Palmprints

The Tsinghua 500PPI Palmprint Database (THUPALMLAB) has 1280 palmprint images captured from 80 subjects. Each image has an original resolution of 2040x2040. The images were acquired using the Hisign scanner [423]. The texture images from the describable texture database were synthesized as depth onto the THUPALMLAB database. For genuineness, random images from the wrinkled texture folder were chosen to provide depth to the palm prints.

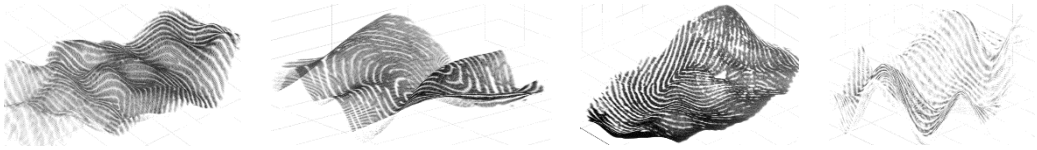
Original image	
Texture image	
Input: 3-D biometric with texture	

Figure 8.13: Shows the input image, texture image, and depth fused biometric image (fingerprint).

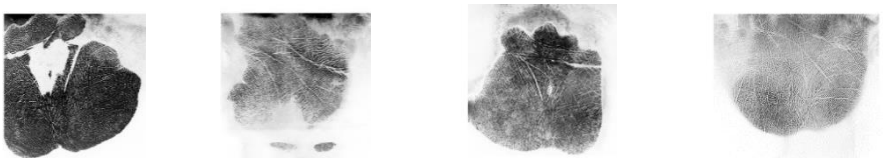
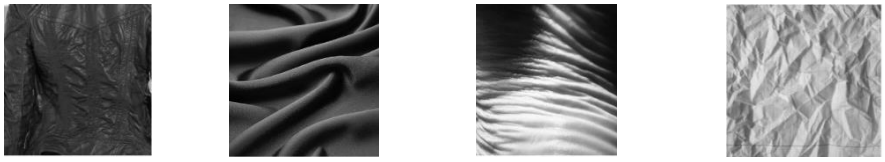
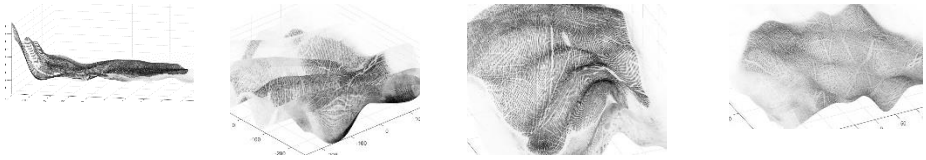
Original image	
Texture image	
Input: 3-D biometric with texture	

Figure 8.14: Shows the input image, texture image, and depth fused biometric image (palm-print).

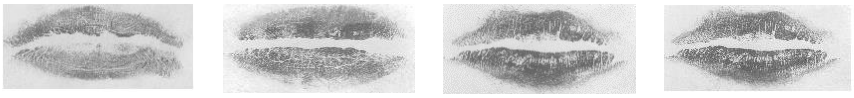
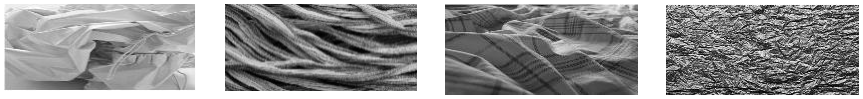
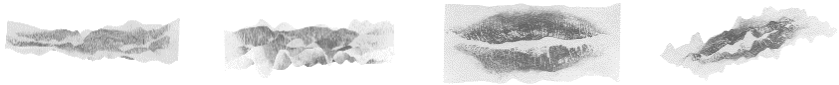
Original image	
Texture image	
Input: 3-D biometric with texture	

Figure 8.15: Shows the input image, texture image, and depth fused biometric image (lip-print).

3) Database synthesis – Lip prints

The lip-print database created by the University of Silesia at Katowice contains 60 lip print samples taken from 15 subjects and 4 samples from each subject. The scans are 16-bit grayscale images with a resolution of 300dpi [485]. The texture images from the describable texture database were synthesized as depth onto the Lip print database. For genuineness, random images from the wrinkled texture folder were chosen to provide depth to the lip prints.

Figure 8.13, Figure 8.14, and Figure 8.15 show the results of the database synthesis. The synthesis algorithm causes few distortions and stretching effects when the depth variations are relatively high.

8.4 Conclusion

This chapter introduced three new open-source, synchronous, multi-modal, and multi-dimensional datasets for research purposes. TDFACE contains 1) higher modalities than existing datasets; 2) Diversity in nationality, age, and gender; 3) multi-pose and expressions. TID contains 1) higher modalities than existing datasets; 2) Various environmental and lighting conditions; 3) Diversity in driver age and gender; 4) Eye-tracking actions annotation of drivers. 3D Forensic Database

contains 1) first-of-its-kind 3D-2D synthetically generated forensic texture database; 2) Real Ground-truth information; 3) Multi-modal biometric data.

The ultimate goal of creating these datasets is to provide researchers developing algorithms with real-world data so that they can train, validate, and benchmark their systems. These datasets aim to solve the scarcity of comprehensive multi-modal and multi-dimensional datasets in face recognition and autonomous driving. This chapter also introduces a cost-effective method for creating synthetic datasets for forensic science 3D recognition.

8.5 Acknowledgement

This work is partially supported by the US Department of Transportation, Federal Highway Administration (FHWA), under the grant contract: 693JJ320C000023. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the US Department of Transportation or the FHWA.

Chapter 9: Conclusion and Future Work

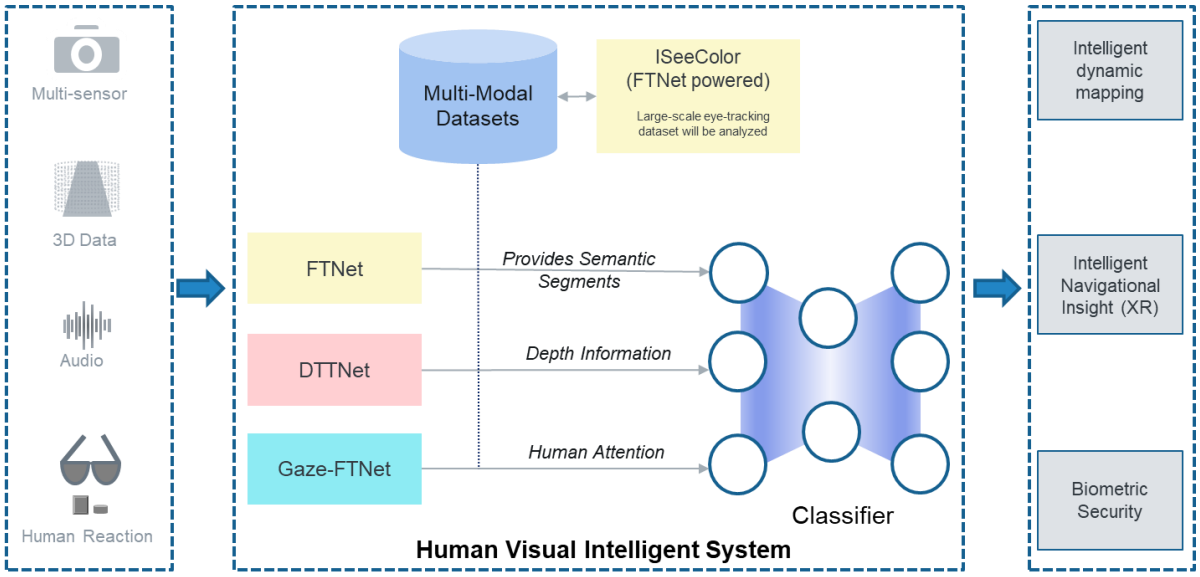


Figure 9.1: An illustration of the human visual intelligence-based multi-modal multi-dimensional pipeline using the algorithms and datasets developed in this dissertation.

This dissertation focuses on building the future of Visual Intelligence using a virtuous cycle of research in computer vision, human cognition, and artificial intelligence. Further, it aimed to develop AI algorithms with multi-modal, multi-dimensional features to overcome the limitations of visible spectrum systems. An illustration of the human visual intelligence-based system is shown in Figure 9.1. One of the future work involves fully realizing this pipeline.

The dissertation attempted to prove two key hypotheses: 1) The metaverse will embed human cognition in its Visual Intelligence systems. 2) It will become necessary to efficiently and in real-time transform multi-dimensional, multi-modal data into virtualized environments.

Chapters 2, 3, 4, 5, 6, and 8 studied the first hypothesis. Furthermore, chapters 2, 3, 4, 5, 6, 7, and 8 were studied to prove the second hypothesis. One of the key findings in this extensive research was the lack of good quality eye movement datasets and the limitation of current literature in analyzing controlled environments. Another challenge was the lack of tools to annotate eye-tracking datasets. A study conducted showed that an experience human annotator would take approximately 50 minutes to annotate 3-minute eye-tracking data. Chapter 2 introduced two novel methods for object of interest (OOI) annotation and recognition to solve this. A multi-level feature extraction and template matching technique were proposed in the first approach. While the number of known samples limits template matching, it is very powerful in annotating repetitive actions or objects. The second approach developed a deep learning-based eye-tracking analytics tool, 'ISeeColor.' It is the analytics tool to determine where participants looked, what they looked at (classification), and how long they looked (duration).

Chapter 3 attempted to develop multi-level dynamic mapping, localizing, and navigations systems. The proposed method used data fusion, path-planning AI, and AR to generate traversable terrestrial maps to provide navigational insight. This prototype was designed for firefighters utilizing LiDAR and environmental sensors to map the location in real-time irrespective of light, heat, and visibility.

One of the major limitations in current image processing AI is its dependency on optical (visible) imagery. For instance, visible cameras are susceptible to lighting conditions and become invalid in total darkness. Furthermore, their imaging quality decreases significantly in adverse environmental conditions, such as rain and smog [215]. Chapter 4 proposed to develop novel AI architectures for thermal image segmentation. It is superior to visible imaging because it operates in darkness and across illumination variations. It can penetrate smoke, aerosol, dust, and mist [217]. However, AI has not explored modalities such as thermal images significantly because of

the lack of defining features in thermal images. Their inter-class variance of objects in thermal images is extremely low, making accurate labeling near boundaries difficult, resulting in a large amount of semantic ambiguity and intensifying the challenge of semantic segmentation. This leads to the failure of most state-of-the-art methods, which traditionally focus on diminishing semantic ambiguity using the rich context information of visible images. To solve this, a novel unified end-to-end trainable network that captures discriminative thermal image features from multiple resolutions and combines them in a fully connected approach was proposed. Using a ResNext encoder as the backbone, FTNet successfully captured and optimized feature representation at the multi-scale resolution, thereby improving the capability of handling high-resolution images and producing quality output at a lower computational cost. FTNet outperformed the state-of-the-art methods on three different datasets showing its robustness. It was then successfully applied to human gaze applications to develop a very low-parameter FTNet architecture that could be used in edge devices.

In chapter 5, the recently developed architectures were applied to 3D processing. Specifically, depth map images were estimated using monocular RGB images. The proposed DTTNet was designed to capture and optimize multi-scale feature representation using the FTNet decoder network. DTTNet then combined ‘low-level layers with few semantic features but high resolution’ with ‘high-level layers with abundant semantic features but low resolution.’

In chapter 5, it was discovered that AI systems could not be evaluated by quantitative mechanisms alone due to the sub-par quality of ground-truth data. Chapter 6 proposed a novel no-reference color-depth measure (CDME) that conforms with human visual analysis to prevent a well-designed AI from getting penalized in depth estimation. CDME combines color measure metrics, including colorfulness, contrast, and sharpness, with a novel depth measure (DM) based on Weber’s

measure. DM is designed to identify the discontinuities and visibility components of the depth image.

In chapter 7, a shape-independent 3D unrolling algorithm and mosaicking algorithm were proposed. This system was tested on biometric fingerprint images with many deformities. The non-parametric approach was used to transform and manipulate 3D images of real-world objects with geometric precision. Furthermore, a novel pressure simulation with a multi-view mosaicking algorithm was introduced to simulate human contact pressure on 3D images. This system was significantly tested on several different biometric modalities, including fingerprint and palmprint. Furthermore, it was tested on the newly developed forensic 3D database successfully. By mapping 3-D surfaces to a 2-D plane and vice-versa, the proposed method aimed to bridge the problems of identification/verification between objects of different input forms.

Finally, one of the most common discovery in all the chapters has been the lack of comprehensive datasets for AI learning. Existing datasets are small, constrained, and lack quality in ground truth annotation. Chapter 8 introduced three new comprehensive, large, and versatile multi-modal and multi-dimensional datasets to solve these overarching challenges. This dataset introduced the largest multi-modal face database and driving database.

To conclude, this dissertation addressed key aspects of CV, AI, Robotics (Navigation), User-Interactivity, and Extended Reality; 5 of the 8 technology pillars of the metaverse. Several novel architectures, datasets, inventions, tools, and apps were successfully built for fire-fighting personnel, autonomous driving, biometric security, robotics, virtual reality simulations, and psychology through the course of this dissertation.

Future Work

Improving situational awareness is a very important field in defense and security. Situational awareness is the sense and comprehension of one's immediate surroundings. Maintaining situational awareness is crucial for performance and error prevention in safety-sensitive sectors. Prior work has examined applying mixed reality (XR) to improve situational awareness. However, in virtual reality, there is a lot of eye movement resulting in little to no convergence or fixation. Fixation mismatch can occur ahead of or behind the plane, and the eyes may also be vertically lagged. This is illustrated in Figure 9.2.

One solution would be to estimate gaze as a 3D coordinate (x, y, z) rather than the traditional 2D coordinate (x, y). This method could help develop a mapping between gaze and 3D objects in a 3D plane. Future work should include collecting an augmented dataset with gaze points and depth maps with high precision ground truth data. An illustration of the data collection method is shown in Figure 9.3. Next, develop new AI systems to predict the depth of gaze points using this dataset. Solving this problem will have several good implications in the metaverse. It would allow users to place and manipulate virtual objects with high precision. Understanding the gaze depth can also allow developers to place text content in optimal viewing points in virtual reality.

Causality in AI: Since bursting onto the scene around 2012, deep learning has demonstrated a particularly impressive ability to recognize patterns in data through correlation. However, deep learning is fundamentally blind to cause and effect. Furthermore, human involvement is pivotal in automobile, medical, and military; however current state-of-the-art systems are limited to visual imagery alone, and humans turn into passive observers. Deep learning (DL) algorithms are trained specifically using networks that ignore certain invariances (for instance, translation for CNN). Biological brains (Humans) appear to work differently. Rather than ignore invariances, humans

are hardwired to use patterns that convey semantics. DL networks are not trained to identify affordances that lead to pattern identification that leads to semantic interpretation. Too much of deep learning has focused on correlation without causation, and that often leaves deep learning systems at a loss when they are tested on conditions that are not quite the same as the ones they were trained on. While the dissertation developed human cognition research and learning tools, it did not apply that cognitive knowledge to applied research. A viable solution would be to develop a deep learning architecture that incorporates human sensory data into the ground-truth data.

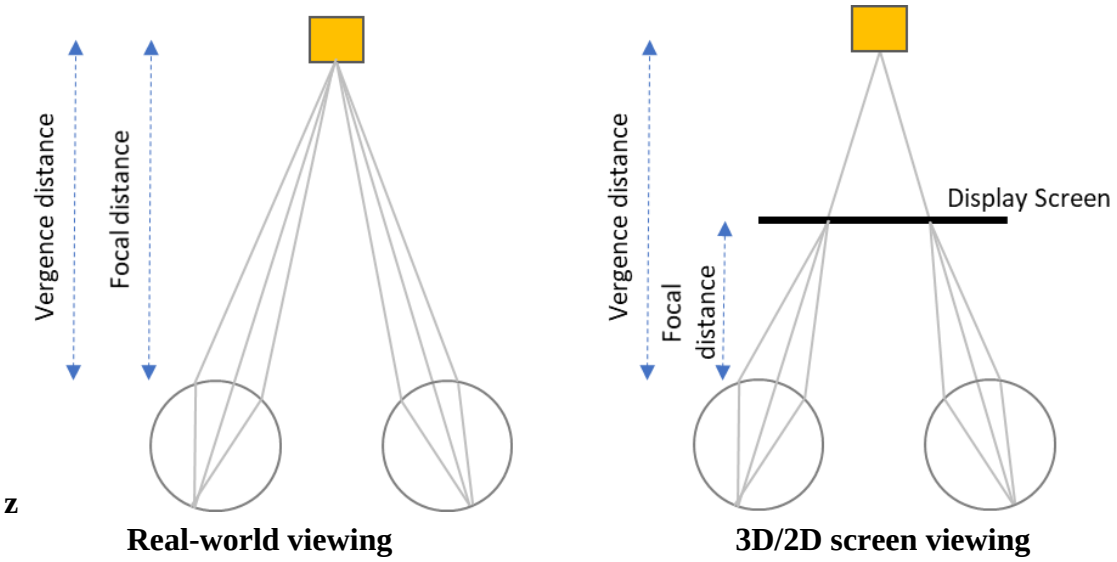


Figure 9.2: The vergence problem in virtual reality. This occurs both in 3D virtual reality and 2D screens. However, the effect of fixation mismatch is more common in virtual reality applications.

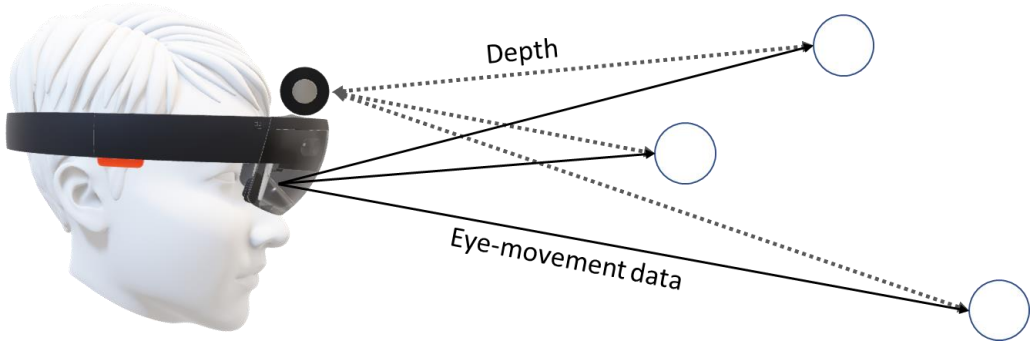


Figure 9.3: The user wearing an eye-tracker and depth-sensing camera views different objects at different distances and angles.

References

[1] L.-H. Lee *et al.*, "All one needs to know about metaverse: A complete survey on technological singularity, virtual ecosystem, and research agenda," *arXiv preprint arXiv:2110.05352,* 2021.

[2] H. Ning *et al.*, "A Survey on Metaverse: the State-of-the-art, Technologies, Applications, and Challenges," *arXiv preprint arXiv:2111.09673,* 2021.

[3] M. Sparkes, "What is a metaverse," ed: Elsevier, 2021.

[4] D. Pereira. (2021). *How AI will shape the Metaverse*. Available: https://towardsdatascience.com/how-ai-will-shape-the-metaverse-4ea7ae20c99

[5] J. D. N. Dionisio, W. G. B. III, and R. Gilbert, "3D virtual worlds and the metaverse: Current status and future possibilities," *ACM Computing Surveys (CSUR),* vol. 45, no. 3, pp. 1-38, 2013.

[6] H.-j. Jeon, H.-c. Youn, S.-m. Ko, and T.-h. Kim, "Blockchain and AI Meet in the Metaverse," *Advances in the Convergence of Blockchain and Artificial Intelligence,* p. 73, 2022.

[7] S. Mystakidis, "Metaverse," *Encyclopedia,* vol. 2, no. 1, pp. 486-497, 2022.

[8] S. Kraus, D. Kanbach, P. Krysta, M. Steinhoff, and N. Tomini, "Facebook and the creation of the metaverse: Radical business model innovation or incremental transformation?," *International Journal of Entrepreneurial Behavior & Research,* 2022.

[9] D. Forsyth and J. Ponce, *Computer vision: A modern approach*. Prentice hall, 2011.

[10] R. Szeliski, *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[11] K. G. Nalbant and Ş. UYANIK, "Computer Vision in the Metaverse," *Journal of Metaverse,* vol. 1, no. 1, pp. 9-12, 2021.

[12] F. K. Došilović, M. Brčić, and N. Hlupić, "Explainable artificial intelligence: A survey," in *2018 41st International convention on information and communication technology, electronics and microelectronics (MIPRO),* 2018, pp. 0210-0215: IEEE.

[13] K. Grace, J. Salvatier, A. Dafoe, B. Zhang, and O. Evans, "When will AI exceed human performance? Evidence from AI experts," *Journal of Artificial Intelligence Research,* vol. 62, pp. 729-754, 2018.

[14] C. S. U. Global. (2021). *Why is Artificial Intelligence (AI) So Important? | CSU Global*. Available: https://csuglobal.edu/blog/why-is-artificial-intelligence-so-important

[15] F. Lieder and T. L. Griffiths, "Strategy selection as rational metareasoning," *Psychological review,* vol. 124, no. 6, p. 762, 2017.

[16] T. Linzen, E. Dupoux, and Y. Goldberg, "Assessing the ability of LSTMs to learn syntax-sensitive dependencies," *Transactions of the Association for Computational Linguistics,* vol. 4, pp. 521-535, 2016.

[17] J. R Anderson, "Computer simulation of a language acquisition system: A first report," 1975.

[18] J. B. Tenenbaum, T. L. Griffiths, and C. Kemp, "Theory-based Bayesian models of inductive learning and reasoning," *Trends in cognitive sciences,* vol. 10, no. 7, pp. 309-318, 2006.

[19] D. Rumelhart, G. Hinton, and R. Williams, "Learning internal representation by error propagation. 318-362 in: Rumelhart, DE, McClelland, JL, PDP Research Group, Parallel. Distributes Processing," ed: MIT Press, Cambridge, 1986.
[20] J. L. Elman, "Finding structure in time," *Cognitive science,* vol. 14, no. 2, pp. 179-211, 1990.
[21] D. L. Yamins and J. J. DiCarlo, "Using goal-driven deep learning models to understand sensory cortex," *Nature neuroscience,* vol. 19, no. 3, pp. 356-365, 2016.
[22] J. E. Fan, D. L. Yamins, and N. B. Turk-Browne, "Common object representations for visual production and recognition," *Cognitive science,* vol. 42, no. 8, pp. 2670-2698, 2018.
[23] A. Banino *et al.*, "Vector-based navigation using grid-like representations in artificial agents," *Nature,* vol. 557, no. 7705, pp. 429-433, 2018.
[24] D. D. Bourgin, J. C. Peterson, D. Reichman, S. J. Russell, and T. L. Griffiths, "Cognitive model priors for predicting human decisions," in *International conference on machine learning,* 2019, pp. 5133-5141: PMLR.
[25] A. Chymis, "Should we fear Artificial Intelligence," *Artificial Intelligence and its Impact on Business,* pp. 39-53, 2020.
[26] M. Muro, R. Maxim, and J. Whiton, "Automation and artificial intelligence: How machines are affecting people and places," 2019.
[27] A. J. London, "Artificial intelligence and black-box medical decisions: accuracy versus explainability," *Hastings Center Report,* vol. 49, no. 1, pp. 15-21, 2019.
[28] L. K. Wenliang and A. R. Seitz, "Deep neural networks for modeling visual perceptual learning," *Journal of Neuroscience,* vol. 38, no. 27, pp. 6028-6044, 2018.
[29] J. C. Peterson, *Leveraging deep neural networks to study human cognition*. University of California, Berkeley, 2018.
[30] K. Panetta, K. S. Kamath, S. Rajeev, and S. S. Agaian, "FTNet: Feature Transverse Network for Thermal Image Semantic Segmentation," *IEEE Access,* vol. 9, pp. 145212-145227, 2021.
[31] N. C. Thompson, K. Greenewald, K. Lee, and G. F. Manso, "The computational limits of deep learning," *arXiv preprint arXiv:2007.05558,* 2020.
[32] (2017). *AFIS vs. IAFIS, How They Are Same Yet Different?* Available: https://www.bayometric.com/afis-vs-iafis/
[33] L. Lugini, E. Marasco, B. Cukic, and I. Gashi, "Interoperability in fingerprint recognition: A large-scale empirical study," in *2013 43rd Annual IEEE/IFIP Conference on Dependable Systems and Networks Workshop (DSN-W)*, 2013, pp. 1-6: IEEE.
[34] J. Priesnitz, C. Rathgeb, N. Buchmann, C. Busch, and M. Margraf, "An overview of touchless 2D fingerprint recognition," *EURASIP Journal on Image and Video Processing,* vol. 2021, no. 1, pp. 1-28, 2021.
[35] A. Ross and A. Jain, "Biometric sensor interoperability: A case study in fingerprints," in *International Workshop on Biometric Authentication*, 2004, pp. 134-145: Springer.
[36] S. Rajeev, S. K. KM, and S. S. Agaian, "Method for modeling post-mortem biometric 3D fingerprints," in *Mobile Multimedia/Image Processing, Security, and Applications 2016*, 2016, vol. 9869, pp. 159-168: SPIE.
[37] Y. Han and D. Hu, "Multispectral Fusion Approach for Traffic Target Detection in Bad Weather," *Algorithms,* vol. 13, no. 11, p. 271, 2020.
[38] Nvidia. (2021). *What is synthetic data?* Available: https://blogs.nvidia.com/blog/2021/06/08/what-is-synthetic-data/

[39] K. Panetta *et al.*, "A comprehensive database for benchmarking imaging systems," *IEEE transactions on pattern analysis and machine intelligence,* vol. 42, no. 3, pp. 509-520, 2018.
[40] Q. Wan, S. Rajeev, A. Kaszowska, K. Panetta, H. A. Taylor, and S. Agaian, "Fixation oriented object segmentation using mobile eye tracker," in *Mobile Multimedia/Image Processing, Security, and Applications 2018*, 2018, vol. 10668, p. 106680D: International Society for Optics and Photonics.
[41] K. Panetta *et al.*, "Iseecolor: method for advanced visual analytics of eye tracking data," *IEEE Access*, vol. 8, pp. 52278-52287, 2020.
[42] S. Rajeev, Q. Wan, K. Yau, K. Panetta, and S. Agaian, "Augmented reality-based vision-aid indoor navigation system in GPS denied environment," in *Mobile Multimedia/Image Processing, Security, and Applications 2019*, 2019, vol. 10993, p. 109930P: International Society for Optics and Photonics.
[43] S. Rajeev, K. Panetta, and S. S. Agaian, "No-reference color-depth quality measure: CDME," in *Mobile Multimedia/Image Processing, Security, and Applications 2018*, 2018, vol. 10668, p. 106680L: International Society for Optics and Photonics.
[44] K. Panetta, S. Rajeev, K. S. Kamath, and S. S. Agaian, "Unrolling post-mortem 3D fingerprints using mosaicking pressure simulation technique," *Ieee Access*, vol. 7, pp. 88174-88185, 2019.
[45] S. Rajeev, S. K. KM, K. Panetta, and S. S. Agaian, "Forensic print extraction using 3D technology and its processing," in *Mobile Multimedia/Image Processing, Security, and Applications 2017*, 2017, vol. 10221, p. 102210L: International Society for Optics and Photonics.
[46] S. Rajeev, Q. Wan, K. Yau, K. Panetta, and S. S. Agaian, "Augmented reality-based vision-aid indoor navigation system in GPS denied environment," in *Mobile Multimedia/Image Processing, Security, and Applications 2019*, 2019, vol. 10993, p. 109930P: International Society for Optics and Photonics.
[47] S. Rajeev, A. Samani, K. Panetta, and S. Agaian, "3D Navigational Insight using AR Technology," in *2019 IEEE International Symposium on Technologies for Homeland Security (HST)*, 2019, pp. 1-4: IEEE.
[48] S. Rajeev, K. K. Shreyas, K. Panetta, and S. Agaian, "3-D palmprint modeling for biometric verification," in *2017 IEEE International Symposium on Technologies for Homeland Security (HST)*, 2017, pp. 1-6: IEEE.
[49] S. K. KM, S. Rajeev, K. Panetta, and S. S. Agaian, "Comparative study of palm print authentication system using geometric features," in *Mobile Multimedia/Image Processing, Security, and Applications 2017*, 2017, vol. 10221, p. 102210M: International Society for Optics and Photonics.
[50] B. S. Yoo and J. H. Kim, "Evolutionary Fuzzy Integral-Based Gaze Control With Preference of Human Gaze," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 8, no. 3, pp. 186-200, 2016.
[51] A. T. Duchowski, "A breadth-first survey of eye-tracking applications," *Behavior Research Methods, Instruments, & Computers*, journal article vol. 34, no. 4, pp. 455-470, November 01 2002.
[52] S. Van der Walt *et al.*, "scikit-image: image processing in Python," *PeerJ*, vol. 2, p. e453, 2014.

[53] G. Andrienko, N. Andrienko, M. Burch, and D. Weiskopf, "Visual analytics methodology for eye movement studies," *IEEE Transactions on Visualization and Computer Graphics,* vol. 18, no. 12, pp. 2889-2898, 2012.
[54] H. Ashraf, M. H. Sodergren, N. Merali, G. Mylonas, H. Singh, and A. Darzi, "Eye-tracking technology in medical education: A systematic review," *Medical teacher,* vol. 40, no. 1, pp. 62-69, 2018.
[55] J. L. Orquin and K. Holmqvist, "Threats to the validity of eye-movement research in psychology," *Behavior research methods,* vol. 50, no. 4, pp. 1645-1656, 2018.
[56] M. Sodhi, B. Reimer, J. Cohen, E. Vastenburg, R. Kaars, and S. Kirschenbaum, "On-road driver eye movement tracking using head-mounted devices," in *Proceedings of the 2002 symposium on Eye tracking research & applications*, 2002, pp. 61-68: ACM.
[57] H. Song, J. Lee, T. J. Kim, K. H. Lee, B. Kim, and J. Seo, "GazeDx: Interactive Visual Analytics Framework for Comparative Gaze Analysis with Volumetric Medical Images," *IEEE transactions on visualization and computer graphics,* vol. 23, no. 1, pp. 311-320, 2017.
[58] M. Vidal, J. Turner, A. Bulling, and H. Gellersen, "Wearable eye tracking for mental health monitoring," *Computer Communications,* vol. 35, no. 11, pp. 1306-1311, 2012.
[59] R. Netzel, M. Burch, and D. Weiskopf, "Comparative eye tracking study on node-link visualizations of trajectories," *IEEE transactions on visualization and computer graphics,* vol. 20, no. 12, pp. 2221-2230, 2014.
[60] S.-H. Kim, Z. Dong, H. Xian, B. Upatising, and J. S. Yi, "Does an eye tracker tell the truth about visualizations?: findings while investigating visualizations for decision making," *IEEE Transactions on Visualization & Computer Graphics,* no. 12, pp. 2421-2430, 2012.
[61] A. Al-Rahayfeh and M. Faezipour, "Eye tracking and head movement detection: A state-of-art survey," *IEEE journal of translational engineering in health and medicine,* vol. 1, 2013.
[62] P. Isokoski, J. Kangas, and P. Majaranta, "Useful approaches to exploratory analysis of gaze data: enhanced heatmaps, cluster maps, and transition maps," in *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*, 2018, p. 68: ACM.
[63] J. Steil, M. X. Huang, and A. Bulling, "Fixation detection for head-mounted eye tracking based on visual similarity of gaze targets," in *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*, 2018, p. 23: ACM.
[64] H. Y. Tsang, M. Tory, and C. Swindells, "eSeeTrack—visualizing sequential fixation patterns," *IEEE Transactions on Visualization and Computer Graphics,* vol. 16, no. 6, pp. 953-962, 2010.
[65] K. Kurzhals, F. Heimerl, and D. Weiskopf, "ISeeCube: Visual analysis of gaze data for video," in *Proceedings of the Symposium on Eye Tracking Research and Applications*, 2014, pp. 43-50: ACM.
[66] R. Netzel, M. Burch, and D. Weiskopf, "Interactive scanpath-oriented annotation of fixations," in *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, 2016, pp. 183-187: ACM.
[67] K. Kurzhals, M. Hlawatsch, C. Seeger, and D. Weiskopf, "Visual analytics for mobile eye tracking," *IEEE transactions on visualization and computer graphics,* vol. 23, no. 1, pp. 301-310, 2017.
[68] P. Yin *et al.*, "A Novel Biologically-inspired Visual Cognition Model—Automatic Extraction of Semantics, Formation of Integrated Concepts and Re-selection Features for

Ambiguity," *IEEE Transactions on Cognitive and Developmental Systems,* vol. PP, no. 99, pp. 1-1, 2017.

[69] D. F. Pontillo, T. B. Kinsman, and J. B. Pelz, "SemantiCode: Using content similarity and database-driven matching to code wearable eyetracker gaze data," in *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*, 2010, pp. 267-270: ACM.

[70] T. Nakamura and T. Nagai, "Ensemble-of-Concept Models for Unsupervised Formation of Multiple Categories," *IEEE Transactions on Cognitive and Developmental Systems,* vol. PP, no. 99, pp. 1-1, 2017.

[71] A. Poole and L. J. Ball, "Eye tracking in HCI and usability research," in *Encyclopedia of human computer interaction*: IGI Global, 2006, pp. 211-219.

[72] T. Blascheck, K. Kurzhals, M. Raschke, M. Burch, D. Weiskopf, and T. Ertl, "State-of-the-art of visualization for eye tracking data," in *Proceedings of EuroVis*, 2014, vol. 2014.

[73] J. H. Goldberg and J. I. Helfman, "Visual scanpath representation," in *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*, 2010, pp. 203-210: ACM.

[74] D. S. Wooding, "Fixation maps: quantifying eye-movement traces," in *Proceedings of the 2002 symposium on Eye tracking research & applications*, 2002, pp. 31-36: ACM.

[75] K. Kurzhals and D. Weiskopf, "Space-time visual analytics of eye-tracking data for dynamic stimuli," *IEEE Transactions on Visualization and Computer Graphics,* vol. 19, no. 12, pp. 2129-2138, 2013.

[76] R. Netzel, M. Hlawatsch, M. Burch, S. Balakrishnan, H. Schmauder, and D. Weiskopf, "An evaluation of visual search support in maps," *IEEE transactions on visualization and computer graphics,* vol. 23, no. 1, pp. 421-430, 2017.

[77] J. H. Goldberg and X. P. Kotval, "Computer interface evaluation using eye movements: methods and constructs," *International Journal of Industrial Ergonomics,* vol. 24, no. 6, pp. 631-645, 1999.

[78] A. T. Duchowski, J. Driver, S. Jolaoso, W. Tan, B. N. Ramey, and A. Robbins, "Scanpath comparison revisited," in *Proceedings of the 2010 symposium on eye-tracking research & applications*, 2010, pp. 219-226: ACM.

[79] T. Opach and A. Nossum, "Evaluating the usability of cartographic animations with eye-movement analysis," in *25th International Cartographic Conference*, 2011, p. 11.

[80] A. Çöltekin, S. I. Fabrikant, and M. Lacayo, "Exploring the efficiency of users' visual analytics strategies based on sequence analysis of eye movement recordings," *International Journal of Geographical Information Science,* vol. 24, no. 10, pp. 1559-1575, 2010.

[81] S. Conversy, C. Hurter, and S. Chatty, "A descriptive model of visual scanning," in *Proceedings of the 3rd BELIV'10 Workshop: BEyond time and errors: novel evaLuation methods for Information Visualization*, 2010, pp. 35-42.

[82] M. Borys and M. Plechawska-Wójcik, "Eye-tracking metrics in perception and visual attention research," *EJMT,* vol. 3, pp. 11-23, 2017.

[83] K. Kurzhals, M. Hlawatsch, F. Heimerl, M. Burch, T. Ertl, and D. Weiskopf, "Gaze stripes: Image-based visualization of eye tracking data," *IEEE transactions on visualization and computer graphics,* vol. 22, no. 1, pp. 1005-1014, 2016.

[84] M. Burch and N. Timmermans, "Sankeye: A visualization technique for AOI transitions," in *ACM Symposium on Eye Tracking Research and Applications*, 2020, pp. 1-5.

[85] T. Blascheck, M. John, K. Kurzhals, S. Koch, and T. Ertl, "VA 2: a visual analytics approach for evaluating visual analytics applications," *IEEE transactions on visualization and computer graphics,* vol. 22, no. 1, pp. 61-70, 2016.

[86] P. K. Muthumanickam, K. Vrotsou, A. Nordman, J. Johansson, and M. Cooper, "Identification of Temporally Varying Areas of Interest in Long-Duration Eye-Tracking Data Sets," *IEEE transactions on visualization and computer graphics*, 2018.
[87] S. S. Alam and R. Jianu, "Analyzing eye-tracking information in visualization and data space: from where on the screen to what on the screen," *IEEE transactions on visualization and computer graphics*, vol. 23, no. 5, pp. 1492-1505, 2017.
[88] R. Jianu and S. S. Alam, "A Data Model and Task Space for Data of Interest (DOI) Eye-Tracking Analyses," *IEEE transactions on visualization and computer graphics*, vol. 24, no. 3, pp. 1232-1245, 2018.
[89] E. Kowler, "Eye movements: The past 25 years," *Vision research*, vol. 51, no. 13, pp. 1457-1483, 2011.
[90] L. R. Squire, N. Dronkers, and J. Baldo, *Encyclopedia of neuroscience*. Elsevier, 2009.
[91] J. L. Levine, *An eye-controlled computer*. IBM Research Division, TJ Watson Research Center, 1981.
[92] T. E. Hutchinson, K. P. White, W. N. Martin, K. C. Reichert, and L. A. Frey, "Human-computer interaction using eye-gaze input," *IEEE Transactions on systems, man, and cybernetics*, vol. 19, no. 6, pp. 1527-1534, 1989.
[93] J. B. Pelz, T. B. Kinsman, and K. M. Evans, "Analyzing complex gaze behavior in the natural world," in *Human Vision and Electronic Imaging XVI*, 2011, vol. 7865, p. 78650Z: International Society for Optics and Photonics.
[94] T. P. Keane, N. D. Cahill, J. A. Tarduno, R. A. Jacobs, and J. B. Pelz, "Computer vision enhances mobile eye-tracking to expose expert cognition in natural-scene visual-search tasks," in *Human Vision and Electronic Imaging XIX*, 2014, vol. 9014, p. 90140F: International Society for Optics and Photonics.
[95] K. M. Evans, R. A. Jacobs, J. A. Tarduno, and J. B. Pelz, "Collecting and analyzing eye tracking data in outdoor environments," *Journal of Eye Movement Research*, vol. 5, no. 2, p. 6, 2012.
[96] K. Chajka, M. Hayhoe, B. Sullivan, J. Pelz, N. Mennie, and J. Droll, "Predictive eye movements in squash," *Journal of Vision*, vol. 6, no. 6, pp. 481-481, 2006.
[97] K. Panetta, R. Rajendran, A. Ramesh, S. P. Rao, and S. Agaian, "Tufts Dental Database: A Multimodal Panoramic X-ray Dataset for Benchmarking Diagnostic Systems," *IEEE Journal of Biomedical and Health Informatics*, 2021.
[98] iMotionsGlobal. (2020). *10 Most Used Eye Tracking Metrics and Terms*. Available: https://imotions.com/blog/10-terms-metrics-eye-tracking/
[99] S. MaiSey, P. SMithen, A. Vilaro Soler, and T. J. Smith, "Recovering from destruction: the conservation, reintegration and perceptual analysis of a flood-damaged painting by John Martin," 2011.
[100] S. K. K. M., S. Rajeev, K. Panetta, and S. S. Agaian, "Comparative study of palm print authentication system using geometric features," in *SPIE Commercial + Scientific Sensing and Imaging*, 2017, vol. 10221, p. 9: SPIE.
[101] B. Zitova and J. Flusser, "Image registration methods: a survey," *Image and vision computing*, vol. 21, no. 11, pp. 977-1000, 2003.
[102] S. Juan, X. Qingsong, and Z. Jinghua, "A scene matching algorithm based on surf feature," in *2010 International Conference on Image Analysis and Signal Processing*, 2010, pp. 434-437: IEEE.

[103] K. G. Derpanis, "Overview of the RANSAC Algorithm," *Image Rochester NY,* vol. 4, no. 1, pp. 2-3, 2010.
[104] (2022). *Low-Pass Filtering (Blurring)*. Available: https://cdn.diffractionlimited.com/help/maximdl/Low-Pass_Filtering.htm
[105] J. Nayak. (2022). Available: https://universe.bits-pilani.ac.in/uploads/JNKDUBAI/ImageProcessing7-FrequencyFiltering.pdf
[106] P.-S. Liao, T.-S. Chen, and P.-C. Chung, "A fast algorithm for multilevel thresholding," *J. Inf. Sci. Eng.,* vol. 17, no. 5, pp. 713-727, 2001.
[107] S. C. Satapathy, N. Sri Madhava Raja, V. Rajinikanth, A. S. Ashour, and N. Dey, "Multi-level image thresholding using Otsu and chaotic bat algorithm," *Neural Computing and Applications,* vol. 29, no. 12, pp. 1285-1307, 2018.
[108] R. Achanta, A. Shaji, and K. Smith, "Aurelien Lucchi, Pascal Fua and Sabine Susstrunk, SLIC Superpixels," EPFL technical Report 1493002010.
[109] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International journal of computer vision,* vol. 59, no. 2, pp. 167-181, 2004.
[110] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on pattern analysis and machine intelligence,* vol. 22, no. 8, pp. 888-905, 2000.
[111] B. Fulkerson and S. Soatto, "Really quick shift: Image segmentation on a GPU," in *European Conference on Computer Vision*, 2010, pp. 350-358: Springer.
[112] K. Parvati, P. Rao, and M. Mariya Das, "Image segmentation using gray-scale morphology and marker-controlled watershed transformation," *Discrete Dynamics in Nature and Society,* vol. 2008, 2008.
[113] G. Intelligence. (2019). *Solutions for behavioral studies, psychology and multimodal research*. Available: https://gazeintelligence.com/smi-product-manual
[114] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," *arXiv preprint arXiv:1802.02611,* 2018.
[115] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," *arXiv preprint*, 2016.
[116] J. Dai *et al.*, "Deformable convolutional networks," *CoRR, abs/1703.06211,* vol. 1, no. 2, p. 3, 2017.
[117] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2002-2011.
[118] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587,* 2017.
[119] H. Zhang *et al.*, "Context encoding for semantic segmentation," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 7151-7160.
[120] C. Sutton and A. McCallum, "An introduction to conditional random fields," *Foundations and Trends® in Machine Learning,* vol. 4, no. 4, pp. 267-373, 2012.
[121] H. Zhang, J. Xue, and K. Dana, "Deep ten: Texture encoding network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 708-717.
[122] A. T. Duchowski, "Eye tracking methodology," *Theory and practice,* vol. 328, 2007.
[123] A. L. Gardony, S. B. Martis, H. A. Taylor, and T. T. Brunyé, "Interaction Strategies for Effective Augmented Reality Geo-Visualization: Insights from Spatial Cognition," *Human–Computer Interaction,* pp. 1-43, 2018.

[124] T. T. Brunyé, A. Gardony, C. R. Mahoney, and H. A. Taylor, "Going to town: Visualized perspectives and navigation through virtual environments," *Computers in Human Behavior*, vol. 28, no. 1, pp. 257-266, 2012.
[125] F. Zafari, A. Gkelias, and K. Leung, "A survey of indoor localization systems and technologies," *arXiv preprint arXiv:1709.01015*, 2017.
[126] D. Solanki and V. Khare, "Optical navigation system for robotics application," in *2012 Third International Conference on Intelligent Systems Modelling and Simulation*, 2012, pp. 268-272: IEEE.
[127] M. Blösch, S. Weiss, D. Scaramuzza, and R. Siegwart, "Vision based MAV navigation in unknown and unstructured environments," in *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 21-28: IEEE.
[128] Y.-Q. Jin and F. Xu, "Reconstruction of the building objects from multi-aspect high-resolution SAR images," in *2007 IEEE international geoscience and remote sensing symposium*, 2007, pp. 555-558: IEEE.
[129] F. Xu, Y.-Q. Jin, and A. Moreira, "A preliminary study on SAR advanced information retrieval and scene reconstruction," *IEEE Geoscience and remote sensing letters*, vol. 13, no. 10, pp. 1443-1447, 2016.
[130] Z. Guo, H. Liu, L. Pang, L. Fang, and W. Dou, "DBSCAN-based point cloud extraction for Tomographic synthetic aperture radar (TomoSAR) three-dimensional (3D) building reconstruction," *International Journal of Remote Sensing*, vol. 42, no. 6, pp. 2327-2349, 2021.
[131] X. Xu, S. Ding, and Z. Shi, "An improved density peaks clustering algorithm with fast finding cluster centers," *Knowledge-Based Systems*, vol. 158, pp. 65-74, 2018.
[132] Y. Chen, L. Zhou, N. Bouguila, C. Wang, Y. Chen, and J. Du, "BLOCK-DBSCAN: Fast clustering for large scale data," *Pattern Recognition*, vol. 109, p. 107624, 2021.
[133] M. Recla and M. Schmitt, "Deep-learning-based single-image height reconstruction from very-high-resolution SAR intensity data," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 183, pp. 496-509, 2022.
[134] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," *Advances in neural information processing systems*, vol. 27, 2014.
[135] M. Rizzi *et al.*, "Navigation-Aided Automotive SAR Imaging in Urban Environments," in *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, 2021, pp. 2979-2982: IEEE.
[136] team_templay. (2019). *3D Map Generator - GEO*. Available: https://graphicriver.net/item/3d-map-generator-geo/12451004
[137] (2019). *Data | UNAVCO*. Available: https://www.unavco.org/data/data.html
[138] (2019). *Synthetic Aperture Radar (SAR) Data | Data | UNAVCO*. Available: https://www.unavco.org/data/imaging/sar/sar.html
[139] F. Monterroso *et al.*, "Automatic generation of co-seismic displacement maps by using Sentinel-1 interferometric SAR data," *Procedia computer science*, vol. 138, pp. 332-337, 2018.
[140] (2022). *April 4, 2010 M7.2 Sierra El Mayor (Mexico) earthquake*. Available: https://sioviz.ucsd.edu/~fialko/baja.html
[141] (2022). *Synthetic Aperture Radar (SAR) Data | Data | UNAVCO*. Available: https://www.unavco.org/data/sar/sar.html

[142] (2022). *What is SAR?* Available: https://asf.alaska.edu/information/sar-information/what-is-sar/
[143] Wikipedia. (2019). *Navigation mesh --- {Wikipedia}, The Free Encyclopedia*. Available: https://en.wikipedia.org/wiki/Navigation_mesh
[144] U. Technologies, "Unity - Manual: Building a NavMesh," 2019.
[145] X. Cui and H. Shi, "An overview of pathfinding in navigation mesh," *IJCSNS,* vol. 12, no. 12, pp. 48-51, 2012.
[146] S. Brand, "Efficient obstacle avoidance using autonomously generated navigation meshes," MSc Thesis, Delft University of Technology, The Netherlands, 2009.
[147] S. Golodetz, "Automatic Navigation Mesh Generation in Configuration Space," *Overload,* vol. 117, pp. 22-27, 2013.
[148] X. Cui and H. Shi, "An overview of pathfinding in navigation mesh," *International Journal of Computer Science and Network Security,* vol. 12, no. 12, pp. 48-51, 2012.
[149] A. Javaid, "Understanding Dijkstra's algorithm," *Available at SSRN 2340905,* 2013.
[150] M. J. De Smith, M. F. Goodchild, and P. Longley, *Geospatial analysis: a comprehensive guide to principles, techniques and software tools*. Troubador publishing ltd, 2007.
[151] M. L. Hetland, *Python Algorithms: mastering basic algorithms in the Python Language*. Apress, 2014.
[152] (2019). *Sygic incorporates augmented reality into GPS navigation app : GPS World*. Available: https://www.gpsworld.com/sygic-incorporates-augmented-reality-into-gps-navigation-app/
[153] (2016). *RECENTLY POPULAR TECHNIQUES AND TECHNOLOGIES USED FOR INDOOR LOCATIONING SYSTEMS*. Available: https://medium.com/@Fizmon/recently-popular-techniques-and-technologies-used-for-indoor-locationing-systems-13afdb6a6adb
[154] A. Kingatua. (2020). *Bluetooth Indoor Positioning and Asset Tracking Solutions*. Available: https://medium.com/supplyframe-hardware/bluetooth-indoor-positioning-and-asset-tracking-solutions-8c78cae0a03
[155] R. Wong. (2017). *Google's VPS indoor navigation is a game-changer for the visually impaired*. Available: https://mashable.com/article/google-visual-positioning-service-tango-augmented-reality
[156] (2022). *Active RFID*. Available: http://comp7304indoorpositioning.weebly.com/active-rfid.html
[157] F. Lassabe, P. Canalda, P. Chatonnay, and F. Spies, "Indoor Wi-Fi positioning: techniques and systems," *annals of telecommunications-annales des télécommunications,* vol. 64, no. 9, pp. 651-664, 2009.
[158] A. M. Hossain and W.-S. Soh, "A survey of calibration-free indoor positioning systems," *Computer Communications,* vol. 66, pp. 1-13, 2015.
[159] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *IEEE infocom,* 2000, vol. 2, no. 2000, pp. 775-784: INSTITUTE OF ELECTRICAL ENGINEERS INC (IEEE).
[160] Wikipedia. *Indoor positioning system*. Available: https://en.wikipedia.org/wiki/Indoor_positioning_system
[161] G. Sterling and S. Hegenderfer. (2019). *Everything You Always Wanted To Know about Beacons*. Available: https://www.brighttalk.com/webcast/635/107617
[162] C. M. Roberts, "Radio frequency identification (RFID)," *Computers & security,* vol. 25, no. 1, pp. 18-26, 2006.

[163] G. Retscher, M. Zhu, and K. Zhang, "RFID positioning," in *Ubiquitous positioning and mobile location-based services in smart phones*: IGI global, 2012, pp. 69-95.
[164] (2019). *Indoor Navigation Using Beacons - Accuracy & More*. Available: https://www.infsoft.com/technology/sensors/bluetooth-low-energy-beacons
[165] (2019). *Provides a comprehensive overview of indoor location technologies, including GPS, WiFi, iBeacon and RFID - comparing the advantages and disadvantages of each*. Available: http://lighthouse.io/indoor-location-technologies-compared/rfid
[166] V. K. Vadlamani, M. Bhattarai, M. Ajith, and M. Martınez-Ramon, "A novel indoor positioning system for unprepared firefighting scenarios," *arXiv preprint arXiv:2008.01344*, 2020.
[167] D. A. de Lima and A. C. Victorino, "A hybrid controller for vision-based navigation of autonomous vehicles in urban environments," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 8, pp. 2310-2323, 2016.
[168] M. Faessler, F. Fontana, C. Forster, E. Mueggler, M. Pizzoli, and D. Scaramuzza, "Autonomous, vision-based flight and live dense 3D mapping with a quadrotor micro aerial vehicle," *Journal of Field Robotics*, vol. 33, no. 4, pp. 431-450, 2016.
[169] G. Gerstweiler, E. Vonach, and H. Kaufmann, "HyMoTrack: A mobile AR navigation system for complex indoor environments," *Sensors*, vol. 16, no. 1, p. 17, 2016.
[170] H.-C. Wang, R. K. Katzschmann, S. Teng, B. Araki, L. Giarré, and D. Rus, "Enabling independent navigation for visually impaired people through a wearable vision-based feedback system," in *2017 IEEE international conference on robotics and automation (ICRA)*, 2017, pp. 6533-6540: IEEE.
[171] J. S. Smith, J.-H. Hwang, F.-J. Chu, and P. A. Vela, "Learning to Navigate: Exploiting Deep Networks to Inform Sample-Based Planning During Vision-Based Navigation," *arXiv preprint arXiv:1801.05132*, 2018.
[172] E. M. Barhorst-Cates, K. M. Rand, and S. H. Creem-Regehr, "The Effects of restricted peripheral field-of-view on spatial learning while navigating," *PloS one*, vol. 11, no. 10, p. e0163785, 2016.
[173] D. Ball *et al.*, "Vision-based obstacle detection and navigation for an agricultural robot," *Journal of field robotics*, vol. 33, no. 8, pp. 1107-1130, 2016.
[174] X. Liang, T. Wang, L. Yang, and E. Xing, "Cirl: Controllable imitative reinforcement learning for vision-based self-driving," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 584-599.
[175] W. G. Aguilar, V. S. Salcedo, D. S. Sandoval, and B. Cobeña, "Developing of a video-based model for UAV autonomous navigation," in *Latin American Workshop on Computational Neuroscience*, 2017, pp. 94-105: Springer.
[176] S. S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: a survey," *Artificial Intelligence Review*, vol. 43, no. 1, pp. 1-54, 2015.
[177] J. Bai, D. Liu, G. Su, and Z. Fu, "A cloud and vision-based navigation system used for blind people," in *Proceedings of the 2017 International Conference on Artificial Intelligence, Automation and Control Technologies*, 2017, p. 22: ACM.
[178] H. Suenaga *et al.*, "Vision-based markerless registration using stereo vision and an augmented reality surgical navigation system: a pilot study," *BMC Medical Imaging*, journal article vol. 15, no. 1, p. 51, November 02 2015.

[179] C. Kanellakis and G. Nikolakopoulos, "Survey on Computer Vision for UAVs: Current Developments and Trends," *Journal of Intelligent & Robotic Systems,* journal article vol. 87, no. 1, pp. 141-168, July 01 2017.
[180] F. Bonin-Font, A. Ortiz, and G. Oliver, "Visual navigation for mobile robots: A survey," *Journal of intelligent and robotic systems,* vol. 53, no. 3, pp. 263-296, 2008.
[181] A. Mcfadyen and L. Mejias, "A survey of autonomous vision-based see and avoid for unmanned aircraft systems," *Progress in Aerospace Sciences,* vol. 80, pp. 1-17, 2016.
[182] S. U. Kamat and K. Rasane, "A Survey on Autonomous Navigation Techniques," in *2018 Second International Conference on Advances in Electronics, Computers and Communications (ICAECC)*, 2018, pp. 1-6: IEEE.
[183] E. Garcia-Fidalgo and A. Ortiz, "Vision-based topological mapping and localization methods: A survey," *Robotics and Autonomous Systems,* vol. 64, pp. 1-20, 2015.
[184] Blippar. (2019). *Augmented Reality (AR) & Computer Vision Company | Blippar*. Available: https://www.blippar.com/
[185] M. Neges, C. Koch, M. König, and M. Abramovici, "Combining visual natural markers and IMU for improved AR based indoor navigation," *Advanced Engineering Informatics,* vol. 31, pp. 18-31, 2017.
[186] (2019). *Vuforia | Augmented Reality for the Industrial Enterprise*. Available: https://www.vuforia.com/
[187] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning*, 2019, pp. 6105-6114: PMLR.
[188] W. G. Van Toll, A. F. Cook IV, and R. Geraerts, "A navigation mesh for dynamic environments," *Computer Animation and Virtual Worlds,* vol. 23, no. 6, pp. 535-546, 2012.
[189] (2019). *Mapbox*. Available: https://www.mapbox.com/
[190] (2019). *Welcome to the QGIS project!* Available: https://www.qgis.org/en/site/
[191] U. Technologies. (2019). *Unity - Manual: Inner Workings of the Navigation System*. Available: https://docs.unity3d.com/Manual/nav-InnerWorkings.html
[192] F. Duchoň *et al.*, "Path planning with modified a star algorithm for a mobile robot," *Procedia Engineering,* vol. 96, pp. 59-69, 2014.
[193] S. Park, S. Bokijonov, and Y. Choi, "Review of microsoft hololens applications over the past five years," *Applied Sciences,* vol. 11, no. 16, p. 7259, 2021.
[194] N. El-Sheimy and Y. Li, "Indoor navigation: State of the art and future trends," *Satellite Navigation,* vol. 2, no. 1, pp. 1-23, 2021.
[195] C. Cadena *et al.*, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on robotics,* vol. 32, no. 6, pp. 1309-1332, 2016.
[196] R.-C. Miclea, C. Dughir, F. Alexa, F. Sandru, and I. Silea, "Laser and LIDAR in a system for visibility distance estimation in fog conditions," *Sensors,* vol. 20, no. 21, p. 6322, 2020.
[197] J. W. Starr and B. Lattimer, "Evaluation of navigation sensors in fire smoke environments," *Fire Technology,* vol. 50, no. 6, pp. 1459-1481, 2014.
[198] T. Huang. (2022). *RPLIDAR-A3 Laser Range Scanner_ Robot Laser Range Scanner|SLAMTEC*. Available: https://www.slamtec.com/en/Lidar/A3
[199] (2022). *BerryGPS-IMU V4 - ozzmaker.com*. Available: https://ozzmaker.com/product/berrygps-imu/

[200] R. Pi. (2022). *Buy a Raspberry Pi Zero – Raspberry Pi*. Available: https://www.raspberrypi.com/products/raspberry-pi-zero/
[201] H. Ye, T. Gu, X. Tao, and J. Lu, "Scalable floor localization using barometer on smartphone," *Wireless Communications and Mobile Computing*, vol. 16, no. 16, pp. 2557-2571, 2016.
[202] M. Liu, H. Li, Y. Wang, F. Li, and X. Chen, "Double-windows-based motion recognition in multi-floor buildings assisted by a built-in barometer," *Sensors*, vol. 18, no. 4, p. 1061, 2018.
[203] G. Pipelidis, O. R. Moslehi Rad, D. Iwaszczuk, C. Prehofer, and U. Hugentobler, "Dynamic vertical mapping with crowdsourced smartphone sensor data," *Sensors*, vol. 18, no. 2, p. 480, 2018.
[204] G. Pipelidis, O. R. M. Rad, D. Iwaszczuk, C. Prehofer, and U. Hugentobler, "A novel approach for dynamic vertical indoor mapping through crowd-sourced smartphone sensor data," in *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2017, pp. 1-8: IEEE.
[205] A. Manivannan, W. C. B. Chin, A. Barrat, and R. Bouffanais, "On the challenges and potential of using barometric sensors to track human activity," *Sensors*, vol. 20, no. 23, p. 6786, 2020.
[206] I. Kim *et al.*, "Nanophotonics for light detection and ranging technology," *Nature nanotechnology*, vol. 16, no. 5, pp. 508-524, 2021.
[207] H. Kim *et al.*, "Vision-based real-time obstacle segmentation algorithm for autonomous surface vehicle," *IEEE Access*, vol. 7, pp. 179420-179428, 2019.
[208] B. Olimov, J. Kim, and A. Paul, "REF-Net: Robust, Efficient, and Fast Network for Semantic Segmentation Applications Using Devices With Limited Computational Resources," *IEEE Access*, vol. 9, pp. 15084-15098, 2021.
[209] W. Wang, Y. Fu, Z. Pan, X. Li, and Y. Zhuang, "Real-time driving scene semantic segmentation," *IEEE Access*, vol. 8, pp. 36776-36788, 2020.
[210] N. Salamati, D. Larlus, G. Csurka, and S. Süsstrunk, "Semantic image segmentation using visible and near-infrared channels," in *European Conference on Computer Vision*, 2012, pp. 461-471: Springer.
[211] P. Yin, R. Yuan, Y. Cheng, and Q. Wu, "Deep guidance network for biomedical image segmentation," *IEEE Access*, vol. 8, pp. 116106-116116, 2020.
[212] F. Meng, L. Guo, Q. Wu, and H. Li, "A new deep segmentation quality assessment network for refining bounding box based segmentation," *IEEE Access*, vol. 7, pp. 59514-59523, 2019.
[213] Y. Xu, S. Arai, F. Tokuda, and K. Kosuge, "A convolutional neural network for point cloud instance segmentation in cluttered scene trained by synthetic data without color," *IEEE Access*, vol. 8, pp. 70262-70269, 2020.
[214] I. Cabrilo, P. Bijlenga, and K. Schaller, "Augmented reality in the surgery of cerebral aneurysms: a technical report," *Operative Neurosurgery*, vol. 10, no. 2, pp. 252-261, 2014.
[215] C. Li, W. Xia, Y. Yan, B. Luo, and J. Tang, "Segmenting objects in day and night: Edge-conditioned cnn for thermal image semantic segmentation," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.
[216] T. Singha, D.-S. Pham, and A. Krishna, "FANet: Feature Aggregation Network for Semantic Segmentation," in *2020 Digital Image Computing: Techniques and Applications (DICTA)*, 2020, pp. 1-8: IEEE.

[217] K. J. Havens and E. J. Sharp, "Chapter 8 - Imager Selection," in *Thermal Imaging Techniques to Survey and Monitor Animals in the Wild*, K. J. Havens and E. J. Sharp, Eds. Boston: Academic Press, 2016, pp. 121-141.
[218] "Global Thermal Imaging Market for the Mobility Industry 2020-2025: Analysis of Applications, Products and Countries," Research and MarketsApril 2021 2021, Available: https://www.researchandmarkets.com/reports/5315057/.
[219] E. M. Zaihidee, K. H. Ghazali, and A. Abd Almisreb, "Comparison of human segmentation using thermal and color image in outdoor environment," in *2015 IEEE Conference on Systems, Process and Control (ICSPC)*, 2015, pp. 152-156: IEEE.
[220] S. Rajeev, S. K. KM, Q. Wan, K. Panetta, and S. S. Agaian, "Illumination invariant NIR face recognition using directional visibility," *Electronic Imaging,* vol. 2019, no. 11, pp. 273-1-273-7, 2019.
[221] S. K. Biswas and P. Milanfar, "Linear support tensor machine with LSK channels: Pedestrian detection in thermal infrared images," *IEEE transactions on image processing,* vol. 26, no. 9, pp. 4229-4242, 2017.
[222] J. Ge, Y. Luo, and G. Tei, "Real-time pedestrian detection and tracking at nighttime for driver-assistance systems," *IEEE Transactions on Intelligent Transportation Systems,* vol. 10, no. 2, pp. 283-298, 2009.
[223] C. Li, H. Cheng, S. Hu, X. Liu, J. Tang, and L. Lin, "Learning collaborative sparse representation for grayscale-thermal tracking," *IEEE Transactions on Image Processing,* vol. 25, no. 12, pp. 5743-5756, 2016.
[224] C. Li, C. Zhu, Y. Huang, J. Tang, and L. Wang, "Cross-modal ranking with soft consistency and noisy labels for robust rgb-t tracking," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 808-823.
[225] L. Kezebou, V. Oludare, K. Panetta, and S. Agaian, "TR-GAN: thermal to RGB face synthesis with generative adversarial network for cross-modal face recognition," in *Mobile Multimedia/Image Processing, Security, and Applications 2020*, 2020, vol. 11399, p. 113990P: International Society for Optics and Photonics.
[226] S. Kamath K. M., R. Rajendran, Q. Wan, K. Panetta, and S. Agaian, *TERNet: A deep learning approach for thermal face emotion recognition* (SPIE Defense + Commercial Sensing). SPIE, 2019.
[227] Q. Wan *et al.*, "Face description using anisotropic gradient: thermal infrared to visible face recognition," in *Mobile Multimedia/Image Processing, Security, and Applications 2018*, 2018, vol. 10668, p. 106680V: International Society for Optics and Photonics.
[228] Q. Ha, K. Watanabe, T. Karasawa, Y. Ushiku, and T. Harada, "MFNet: Towards real-time semantic segmentation for autonomous vehicles with multi-spectral scenes," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 5108-5115: IEEE.
[229] H. Xiong, W. Cai, and Q. Liu, "MCNet: Multi-level Correction Network for thermal image semantic segmentation of nighttime driving scene," *Infrared Physics & Technology,* vol. 113, p. 103628, 2021.
[230] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2881-2890.
[231] J. Wang *et al.*, "Deep high-resolution representation learning for visual recognition," *IEEE transactions on pattern analysis and machine intelligence,* 2020.

[232] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics,* vol. 9, no. 1, pp. 62-66, 1979.
[233] S. K. KM, R. Rajendran, K. Panetta, and S. Agaian, "A human visual based binarization technique for histological images," in *Mobile Multimedia/Image Processing, Security, and Applications 2017*, 2017, vol. 10221, p. 102210Q: International Society for Optics and Photonics.
[234] K. Piniarski and P. Pawłowski, "Segmentation of pedestrians in thermal imaging," in *2018 Baltic URSI Symposium (URSI)*, 2018, pp. 210-211: IEEE.
[235] R. Nock and F. Nielsen, "Statistical region merging," *IEEE Transactions on pattern analysis and machine intelligence,* vol. 26, no. 11, pp. 1452-1458, 2004.
[236] N. Dhanachandra, K. Manglem, and Y. J. Chanu, "Image segmentation using K-means clustering algorithm and subtractive clustering algorithm," *Procedia Computer Science,* vol. 54, pp. 764-771, 2015.
[237] L. Najman and M. Schmitt, "Watershed of a continuous function," *Signal Processing,* vol. 38, no. 1, pp. 99-112, 1994.
[238] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International journal of computer vision,* vol. 1, no. 4, pp. 321-331, 1988.
[239] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on pattern analysis and machine intelligence,* vol. 23, no. 11, pp. 1222-1239, 2001.
[240] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "Unet++: A nested u-net architecture for medical image segmentation," in *Deep learning in medical image analysis and multimodal learning for clinical decision support*: Springer, 2018, pp. 3-11.
[241] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431-3440.
[242] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, 2015, pp. 234-241: Springer.
[243] M. Lin, Q. Chen, and S. Yan, "Network in network," *arXiv preprint arXiv:1312.4400,* 2013.
[244] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence,* vol. 40, no. 4, pp. 834-848, 2017.
[245] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122,* 2015.
[246] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117-2125.
[247] S. Seferbekov, V. Iglovikov, A. Buslaev, and A. Shvets, "Feature pyramid network for multi-class land segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 272-275.
[248] H. Li, P. Xiong, J. An, and L. Wang, "Pyramid attention network for semantic segmentation," *arXiv preprint arXiv:1805.10180,* 2018.

[249] G. Ghiasi, T.-Y. Lin, and Q. V. Le, "Nas-fpn: Learning scalable feature pyramid architecture for object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7036-7045.
[250] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1395-1403.
[251] C. Li, D. Song, R. Tong, and M. Tang, "Illumination-aware faster R-CNN for robust multispectral pedestrian detection," *Pattern Recognition*, vol. 85, pp. 161-171, 2019.
[252] Y. Sun, W. Zuo, and M. Liu, "Rtfnet: Rgb-thermal fusion network for semantic segmentation of urban scenes," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2576-2583, 2019.
[253] S. S. Shivakumar, N. Rodrigues, A. Zhou, I. D. Miller, V. Kumar, and C. J. Taylor, "Pst900: Rgb-thermal calibration, dataset and segmentation network," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 9441-9447: IEEE.
[254] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, 1998.
[255] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770-778.
[256] M. Cordts *et al.*, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213-3223.
[257] Z. Yu, C. Feng, M.-Y. Liu, and S. Ramalingam, "Casenet: Deep category-aware semantic edge detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5964-5973.
[258] N. Ketkar, "Stochastic gradient descent," in *Deep learning with Python*: Springer, 2017, pp. 113-132.
[259] A. Paszke *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *arXiv preprint arXiv:1912.01703*, 2019.
[260] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818-2826.
[261] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1492-1500.
[262] S. Z. S. Zaini *et al.*, "Image Quality Assessment for Image Segmentation Algorithms: Qualitative and Quantitative Analyses," in *2019 9th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, 2019, pp. 66-71: IEEE.
[263] X. Zhang, P. Ye, and G. Xiao, "VIFB: a visible and infrared image fusion benchmark," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 104-105.
[264] G. Dong, Y. Yan, C. Shen, and H. Wang, "Real-time high-performance semantic image segmentation of urban street scenes," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3258-3274, 2020.
[265] D. Nie and D. Shen, "Adversarial Confidence Learning for Medical Image Segmentation and Synthesis," *International Journal of Computer Vision*, vol. 128, 2020.

[266] S. Qiu, Y. Zhao, J. Jiao, Y. Wei, and S. Wei, "Referring image segmentation by generative adversarial learning," *IEEE Transactions on Multimedia,* vol. 22, no. 5, pp. 1333-1344, 2019.
[267] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251-1258.
[268] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700-4708.
[269] A. G. Howard *et al.*, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861,* 2017.
[270] W. Wang, Q. Lai, H. Fu, J. Shen, H. Ling, and R. Yang, "Salient object detection in the deep learning era: An in-depth survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 2021.
[271] S. He, H. R. Tavakoli, A. Borji, Y. Mi, and N. Pugeault, "Understanding and visualizing deep visual saliency models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10206-10215.
[272] J. Xu, M. Jiang, S. Wang, M. S. Kankanhalli, and Q. Zhao, "Predicting human gaze beyond pixels," *Journal of vision,* vol. 14, no. 1, pp. 28-28, 2014.
[273] T. Judd, K. Ehinger, F. Durand, and A. Torralba, "Learning to predict where humans look," in *2009 IEEE 12th international conference on computer vision*, 2009, pp. 2106-2113: IEEE.
[274] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems,* vol. 25, pp. 1097-1105, 2012.
[275] X. Huang, C. Shen, X. Boix, and Q. Zhao, "Salicon: Reducing the semantic gap in saliency prediction by adapting deep neural networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 262-270.
[276] Z. Bylinskii, T. Judd, A. Oliva, A. Torralba, and F. Durand, "What do different evaluation metrics tell us about saliency models?," *IEEE transactions on pattern analysis and machine intelligence,* vol. 41, no. 3, pp. 740-757, 2018.
[277] N. Riche, M. Duvinage, M. Mancas, B. Gosselin, and T. Dutoit, "Saliency and human fixations: State-of-the-art and study of comparison metrics," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 1153-1160.
[278] I. Vasiljevic *et al.*, "Diode: A dense indoor and outdoor depth dataset," *arXiv preprint arXiv:1908.00463,* 2019.
[279] B. Xiao *et al.*, "Depth estimation of hard inclusions in soft tissue by autonomous robotic palpation using deep recurrent neural network," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 4, pp. 1791-1799, 2020.
[280] M. Kalia, N. Navab, and T. Salcudean, "A real-time interactive augmented reality depth estimation technique for surgical robotics," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 8291-8297: IEEE.
[281] L. Chen, Z. Yang, J. Ma, and Z. Luo, "Driving scene perception network: Real-time joint detection, depth estimation and semantic segmentation," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018, pp. 1283-1291: IEEE.

[282] Y. Huang and Y. Chen, "Survey of state-of-art autonomous driving technologies with deep learning," in *2020 IEEE 20th International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, 2020, pp. 221-228: IEEE.
[283] S. Niklaus, L. Mai, J. Yang, and F. Liu, "3d ken burns effect from a single image," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 6, pp. 1-15, 2019.
[284] Y.-M. Tsai, Y.-L. Chang, and L.-G. Chen, "Block-based vanishing line and vanishing point detection for 3D scene reconstruction," in *2006 international symposium on intelligent signal processing and communications*, 2006, pp. 586-589: IEEE.
[285] C. Tang, C. Hou, and Z. Song, "Depth recovery and refinement from a single image using defocus cues," *Journal of Modern Optics*, vol. 62, no. 6, pp. 441-448, 2015.
[286] L. He, J. Yang, B. Kong, and C. Wang, "An automatic measurement method for absolute depth of objects in two monocular images based on SIFT feature," *Applied Sciences*, vol. 7, no. 6, p. 517, 2017.
[287] A. El Bouziady, R. O. H. Thami, M. Ghogho, O. Bourja, and S. El Fkihi, "Vehicle speed estimation using extracted SURF features from stereo images," in *2018 International Conference on Intelligent Systems and Computer Vision (ISCV)*, 2018, pp. 1-6: IEEE.
[288] H. Xu, E. Chen, C. Liang, L. Qi, and L. Guan, "Spatio-temporal pyramid model based on depth maps for action recognition," in *2015 IEEE 17th International Workshop on Multimedia Signal Processing (MMSP)*, 2015, pp. 1-6: IEEE.
[289] Y. Ming, X. Meng, C. Fan, and H. Yu, "Deep Learning for Monocular Depth Estimation: A Review," *Neurocomputing*, 2021.
[290] R. Xiaogang, Y. Wenjing, H. Jing, G. Peiyuan, and G. Wei, "Monocular depth estimation based on deep learning: A survey," in *2020 Chinese Automation Congress (CAC)*, 2020, pp. 2436-2440: IEEE.
[291] C. Zhao, Q. Sun, C. Zhang, Y. Tang, and F. Qian, "Monocular depth estimation based on deep learning: An overview," *Science China Technological Sciences*, pp. 1-16, 2020.
[292] J. Hu, M. Ozay, Y. Zhang, and T. Okatani, "Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2019, pp. 1043-1051: IEEE.
[293] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *2016 Fourth international conference on 3D vision (3DV)*, 2016, pp. 239-248: IEEE.
[294] D. Xu, E. Ricci, W. Ouyang, X. Wang, and N. Sebe, "Multi-scale continuous crfs as sequential deep networks for monocular depth estimation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5354-5362.
[295] H. Sheng, S. Zhang, X. Cao, Y. Fang, and Z. Xiong, "Geometric occlusion analysis in depth estimation using integral guided filter for light-field image," *IEEE Transactions on Image Processing*, vol. 26, no. 12, pp. 5758-5771, 2017.
[296] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.
[297] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," California Univ San Diego La Jolla Inst for Cognitive Science1985.
[298] K. Han *et al.*, "A survey on vision transformer," *arXiv preprint arXiv:2012.12556*, 2020.
[299] A. Vaswani *et al.*, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[300] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," *arXiv preprint arXiv:2010.04159*, 2020.
[301] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*, 2020, pp. 213-229: Springer.
[302] S. F. Bhat, I. Alhashim, and P. Wonka, "Adabins: Depth estimation using adaptive bins," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4009-4018.
[303] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *European conference on computer vision*, 2012, pp. 746-760: Springer.
[304] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231-1237, 2013.
[305] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," *arXiv preprint arXiv:1406.2283*, 2014.
[306] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2650-2658.
[307] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12179-12188.
[308] V. Vanhoucke, "Learning visual representations at scale," *ICLR invited talk*, vol. 1, no. 2, 2014.
[309] H. Touvron, M. Cord, A. Sablayrolles, G. Synnaeve, and H. Jégou, "Going deeper with image transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 32-42.
[310] N. Parmar *et al.*, "Image transformer," in *International Conference on Machine Learning*, 2018, pp. 4055-4064: PMLR.
[311] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7794-7803.
[312] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," *arXiv preprint arXiv:2103.13413*, 2021.
[313] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.
[314] B. Graham *et al.*, "LeViT: a Vision Transformer in ConvNet's Clothing for Faster Inference," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12259-12269.
[315] A. Srinivas, T.-Y. Lin, N. Parmar, J. Shlens, P. Abbeel, and A. Vaswani, "Bottleneck transformers for visual recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 16519-16529.
[316] J. Yang *et al.*, "Focal self-attention for local-global interactions in vision transformers," *arXiv preprint arXiv:2107.00641*, 2021.
[317] D. Zhou *et al.*, "Deepvit: Towards deeper vision transformer," *arXiv preprint arXiv:2103.11886*, 2021.
[318] M. Chen *et al.*, "Generative pretraining from pixels," in *International Conference on Machine Learning*, 2020, pp. 1691-1703: PMLR.

[319] H. Chen *et al.*, "Pre-trained image processing transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12299-12310.
[320] Z. Liu *et al.*, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10012-10022.
[321] A. Hassani, S. Walton, N. Shah, A. Abuduweili, J. Li, and H. Shi, "Escaping the big data paradigm with compact transformers," *arXiv preprint arXiv:2104.05704*, 2021.
[322] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *arXiv preprint arXiv:1505.00853*, 2015.
[323] S. Song, S. P. Lichtenberg, and J. Xiao, "Sun rgb-d: A rgb-d scene understanding benchmark suite," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 567-576.
[324] L. N. Smith and N. Topin, "Super-convergence: Very fast training of residual networks using large learning rates," 2018.
[325] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 41-48.
[326] P. J. Van Laarhoven and E. H. Aarts, "Simulated annealing," in *Simulated annealing: Theory and applications*: Springer, 1987, pp. 7-15.
[327] M. Tan and Q. V. Le, "Efficientnetv2: Smaller models and faster training," *arXiv preprint arXiv:2104.00298*, 2021.
[328] X. Chen, X. Chen, and Z.-J. Zha, "Structure-aware residual pyramid network for monocular depth estimation," *arXiv preprint arXiv:1907.06023*, 2019.
[329] W. Yin, Y. Liu, C. Shen, and Y. Yan, "Enforcing geometric constraints of virtual normal for depth prediction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5684-5693.
[330] J. H. Lee, M.-K. Han, D. W. Ko, and I. H. Suh, "From big to small: Multi-scale local planar guidance for monocular depth estimation," *arXiv preprint arXiv:1907.10326*, 2019.
[331] L. Huynh, P. Nguyen-Ha, J. Matas, E. Rahtu, and J. Heikkilä, "Guiding monocular depth estimation using depth-attention volume," in *European Conference on Computer Vision*, 2020, pp. 581-597: Springer.
[332] M. Song, S. Lim, and W. Kim, "Monocular Depth Estimation Using Laplacian Pyramid-Based Depth Residuals," *IEEE Transactions on Circuits and Systems for Video Technology*, 2021.
[333] A. Kus, E. Unver, and A. Taylor, "A comparative study of 3D scanning in engineering, product and transport design and fashion design education," *Computer Applications in Engineering Education*, vol. 17, no. 3, pp. 263-271, 2009.
[334] A. Georgopoulos, C. Ioannidis, and A. Valanis, "Assessing the performance of a structured light scanner," *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 38, no. Part 5, pp. 251-255, 2010.
[335] S. Rajeev, S. K. K.M., and S. S. Agaian, "Method for modeling post-mortem biometric 3D fingerprints," in *SPIE Commercial + Scientific Sensing and Imaging*, 2016, vol. 9869, p. 10: SPIE.

[336] S. Rajeev, S. K. K. M., K. Panetta, and S. S. Agaian, "Forensic print extraction using 3D technology and its processing," in *SPIE Commercial + Scientific Sensing and Imaging*, 2017, vol. 10221, p. 7: SPIE.
[337] P. R. Alface, M. De Craene, and B. B. Macq, "Three-dimensional image quality measurement for the benchmarking of 3D watermarking schemes," in *Security, Steganography, and Watermarking of Multimedia Contents VII*, 2005, vol. 5681, pp. 230-241: International Society for Optics and Photonics.
[338] L. Zhang, W. J. Tam, and D. Wang, "Stereoscopic image generation based on depth images," in *Image Processing, 2004. ICIP'04. 2004 International Conference on*, 2004, vol. 5, pp. 2993-2996: IEEE.
[339] M. Vahdat, "A study of image quality, authenticity, and metadata characteristics of photogrammetric three-dimensional data in cultural heritage domain," 2010.
[340] J.-Y. Son, O. Chernyshov, C.-H. Lee, M.-C. Park, and S. Yano, "Depth resolution in three-dimensional images," *JOSA A*, vol. 30, no. 5, pp. 1030-1038, 2013.
[341] G. Lavoué, "A multiscale metric for 3D mesh visual quality assessment," in *Computer Graphics Forum*, 2011, vol. 30, no. 5, pp. 1427-1437: Wiley Online Library.
[342] T. Tóth and J. Živčák, "A Comparison of the Outputs of 3D Scanners," *Procedia Engineering*, vol. 69, pp. 393-401, 2014.
[343] P. S. Heckbert and M. Garland, "Optimal triangulation and quadric-based surface simplification," *Computational Geometry*, vol. 14, no. 1-3, pp. 49-65, 1999.
[344] S. Kim, Y. Kim, and K. Sohn, "A study on the effects of RGB-D database scale and quality on depth analogy performance," in *Three-Dimensional Imaging, Visualization, and Display 2016*, 2016, vol. 9867, p. 986707: International Society for Optics and Photonics.
[345] C. Gao, K. Panetta, and S. Agaian, "No reference color image quality measures," in *Cybernetics (CYBCONF), 2013 IEEE International Conference on*, 2013, pp. 243-248: IEEE.
[346] S. S. Agaian, K. Panetta, and A. M. Grigoryan, "A new measure of image enhancement," in *IASTED International Conference on Signal Processing & Communication*, 2000, pp. 19-22: Citeseer.
[347] S. S. Agaian, B. Silver, and K. A. Panetta, "Transform coefficient histogram-based image enhancement algorithms using contrast entropy," *IEEE transactions on image processing*, vol. 16, no. 3, pp. 741-758, 2007.
[348] K. Panetta, C. Gao, and S. Agaian, "Human-visual-system-inspired underwater image quality measures," *IEEE Journal of Oceanic Engineering*, vol. 41, no. 3, pp. 541-551, 2016.
[349] Y.-Y. Fu, *Color image quality measures and retrieval*. New Jersey Institute of Technology, 2006.
[350] K. Panetta, C. Gao, and S. Agaian, "No reference color image contrast and quality measures," *IEEE transactions on Consumer Electronics*, vol. 59, no. 3, pp. 643-651, 2013.
[351] K. Panetta, A. Samani, and S. Agaian, "A robust no-reference, no-parameter, transform domain image quality metric for evaluating the quality of color images," *IEEE Access*, vol. 6, pp. 10979-10985, 2018.
[352] A. Jain, R. Bolle, and S. Pankanti, *Biometrics: personal identification in networked society*. Springer Science & Business Media, 2006.
[353] A. Toosi, A. Bottino, S. Cumani, P. Negri, and P. L. Sottile, "Feature Fusion for Fingerprint Liveness Detection: a Comparative Study," *IEEE Access*, vol. 5, pp. 23695-23709, 2017.

[354] S. K. KM, S. Rajeev, K. Panetta, and S. S. Agaian, "Comparative study of palm print authentication system using geometric features," in *SPIE Commercial+ Scientific Sensing and Imaging*, 2017, pp. 102210M-102210M-9: International Society for Optics and Photonics.
[355] K. K. Shreyas, S. Rajeev, K. Panetta, and S. S. Agaian, "Fingerprint authentication using geometric features," in *Technologies for Homeland Security (HST), 2017 IEEE International Symposium on*, 2017, pp. 1-7: IEEE.
[356] A. K. Jain, A. Ross, and S. Prabhakar, "An introduction to biometric recognition," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 14, no. 1, pp. 4-20, 2004.
[357] S. Liu and M. Silverman, "A practical guide to biometric security technology," *IT Professional*, vol. 3, no. 1, pp. 27-32, 2001.
[358] E. H. Holder, L. O. Robinson, and J. H. Laub, *The fingerprint sourcebook*. US Department. of Justice, Office of Justice Programs, National Institute of Justice, 2011.
[359] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, *Handbook of fingerprint recognition*. Springer Science & Business Media, 2009.
[360] A. Sankaran, M. Vatsa, and R. Singh, "Latent fingerprint matching: A survey," *IEEE Access*, vol. 2, no. 982-1004, p. 1, 2014.
[361] A. Sankaran, M. Vatsa, and R. Singh, "Multisensor optical and latent fingerprint database," *IEEE Access*, vol. 3, pp. 653-665, 2015.
[362] G. Parziale and Y. Chen, "Advanced technologies for touchless fingerprint recognition," in *Handbook of Remote Biometrics*: Springer, 2009, pp. 83-109.
[363] V. I. Ivanov and J. S. Baras, "Authentication of Swipe Fingerprint Scanners," *IEEE Transactions on Information Forensics and Security*, 2017.
[364] J. Feng, S. Yoon, and A. K. Jain, "Latent fingerprint matching: Fusion of rolled and plain fingerprints," in *Advances in Biometrics*: Springer, 2009, pp. 695-704.
[365] Z. Chen and C. Kuo, "A topology-based matching algorithm for fingerprint authentication," in *Security Technology, 1991. Proceedings. 25th Annual 1991 IEEE International Carnahan Conference on*, 1991, pp. 84-87: IEEE.
[366] D. Maltoni, "A tutorial on fingerprint recognition," in *Advanced Studies in Biometrics*: Springer, 2005, pp. 43-68.
[367] R. D. Labati, A. Genovese, V. Piuri, and F. Scotti, "Toward unconstrained fingerprint recognition: A fully touchless 3-D system based on two views on the move," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 2, pp. 202-219, 2016.
[368] S. Bakhtiari, S. S. Agaian, and M. Jamshidi, "Local fingerprint image reconstruction based on gabor filtering," in *SPIE Defense, Security, and Sensing*, 2012, pp. 840602-840602-11: International Society for Optics and Photonics.
[369] Y. Chen, S. C. Dass, and A. K. Jain, "Fingerprint quality indices for predicting authentication performance," in *AVBPA*, 2005, vol. 3546, pp. 160-170: Springer.
[370] T. Kahana, A. Grande, D. Tancredi, J. Penalver, and J. Hiss, "Fingerprinting the deceased: traditional and new techniques," *Journal of forensic sciences*, vol. 46, no. 4, pp. 908-912, 2001.
[371] D. Keating and J. Miller, "A technique for developing and photographing ridge impressions on decomposed water-soaked fingers," *Journal of forensic sciences*, vol. 38, no. 1, pp. 197-202, 1993.

[372] H. C. Lee, R. Ramotowski, and R. Gaensslen, *Advances in fingerprint technology*. CRC press, 2001.
[373] L. Richardson and H. Kade, "Readable fingerprints from mummified or putrefied specimens," *Journal of forensic sciences,* vol. 17, no. 2, p. 325, 1972.
[374] F. T. Zugibe and J. T. Costello, "A new method for softening mummified fingers," *Journal of forensic sciences,* vol. 31, no. 2, pp. 726-731, 1986.
[375] W. D. Haglund, "A technique to enhance fingerprinting of mummified fingers," *Journal of forensic sciences,* vol. 33, no. 5, pp. 1244-1248, 1988.
[376] D. Porta, M. Maldarella, M. Grandi, and C. Cattaneo, "A new method of reproduction of fingerprints from corpses in a bad state of preservation using latex," *Journal of forensic sciences,* vol. 52, no. 6, pp. 1319-1321, 2007.
[377] M. Mulawka and L. Miller, *Postmortem Fingerprinting and Unidentified Human Remains*. Routledge, 2014.
[378] W. Bicz *et al.*, "Fingerprint structure imaging based on an ultrasound camera," *Instrumentation science & technology,* vol. 27, no. 4, pp. 295-303, 1999.
[379] S. S. Agaian, M. M. A. Mulawka, R. Rajendran, S. P. Rao, S. K. KM, and S. Rajeev, "A comparative study of image feature detection and matching algorithms for touchless fingerprint systems," *Electronic Imaging,* vol. 2016, no. 15, pp. 1-9, 2016.
[380] H. Choi, K. Choi, and J. Kim, "Mosaicing touchless and mirror-reflected fingerprint images," *Information Forensics and Security, IEEE Transactions on,* vol. 5, no. 1, pp. 52-61, 2010.
[381] A. Kumar, C. Kwong, and L. Yang, "Contactless 3D fingerprint reconstruction using photometric stereo," Technical Report No. COMP-K2012.
[382] A. Kumar and C. Kwong, "Towards contactless, low-cost and accurate 3D fingerprint identification," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 37, no. 3, pp. 681-696, 2015.
[383] Y. Chen, G. Parziale, E. Diaz-Santana, and A. K. Jain, "3D touchless fingerprints: compatibility with legacy rolled images," in *Biometric Consortium Conference, 2006 Biometrics Symposium: Special Session on Research at the*, 2006, pp. 1-6: IEEE.
[384] Y. Wang, L. G. Hassebrook, and D. L. Lau, "Data acquisition and processing of 3-D fingerprints," *Information Forensics and Security, IEEE Transactions on,* vol. 5, no. 4, pp. 750-760, 2010.
[385] L. Sweeney, V. Weedn, and R. Gross, "HandShot: A Fast 3-D Imaging System for Capturing Fingerprints, Palm Prints and Hand Geometry," 2004.
[386] G. Parziale, E. Diaz-Santana, and R. Hauke, "The surround imagertm: A multi-camera touchless device to acquire 3d rolled-equivalent fingerprints," in *Advances in Biometrics*: Springer, 2005, pp. 244-250.
[387] S. Malassiotis, N. Aifanti, and M. G. Strintzis, "Personal authentication using 3-D finger geometry," *IEEE Transactions on Information Forensics and Security,* vol. 1, no. 1, pp. 12-21, 2006.
[388] F. Chen, "3D fingerprint and palm print data model and capture devices using multi structured lights and cameras," ed: Google Patents, 2009.
[389] C. H. Esteban and F. Schmitt, "Multi-stereo 3d object reconstruction," in *3D Data Processing Visualization and Transmission, 2002. Proceedings. First International Symposium on*, 2002, pp. 159-166: IEEE.

[390] M. Pollefeys *et al.*, "Detailed real-time urban 3d reconstruction from video," *International Journal of Computer Vision,* vol. 78, no. 2-3, pp. 143-167, 2008.
[391] D. L. Woodard, T. C. Faltemier, P. Yan, P. J. Flynn, and K. W. Bowyer, "A comparison of 3D biometric modalities," in *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW'06. Conference on*, 2006, pp. 57-57: IEEE.
[392] J. N. Bradley, C. M. Brislawn, and T. Hopper, "FBI wavelet/scalar quantization standard for gray-scale fingerprint image compression," in *Optical Engineering and Photonics in Aerospace Sensing*, 1993, pp. 293-304: International Society for Optics and Photonics.
[393] A. Fatehpuria, D. L. Lau, and L. G. Hassebrook, "Acquiring a 2D rolled equivalent fingerprint image from a non-contact 3D finger scan," in *Defense and Security Symposium*, 2006, pp. 62020C-62020C-8: International Society for Optics and Photonics.
[394] F. Liu and D. Zhang, "3D fingerprint reconstruction system using feature correspondences and prior estimated finger model," *Pattern Recognition,* vol. 47, no. 1, pp. 178-193, 2014.
[395] S. J. Xie, J. Yang, S. Yoon, D. Park, and J. Shin, "Fingerprint quality analysis and estimation for fingerprint matching," *State of the art in Biometrics. Intech, Vienna. ISBN,* pp. 978-953, 2011.
[396] C. S. Mlambo and Y. Moolla, "Complexity and distortion analysis on methods for unrolling 3D to 2D fingerprints," in *Signal-Image Technology & Internet-Based Systems (SITIS), 2015 11th International Conference on*, 2015, pp. 103-109: IEEE.
[397] X. Feng, T. Liu, D. Yang, and Y. Wang, "Saliency inspired full-reference quality metrics for packet-loss-impaired video," *IEEE Transactions on Broadcasting,* vol. 57, no. 1, pp. 81-88, 2010.
[398] S. Rajeev, S. K. KM, and S. S. Agaian, "Method for modeling post-mortem biometric 3D fingerprints," in *SPIE Commercial+ Scientific Sensing and Imaging*, 2016, pp. 98690S-98690S-10: International Society for Optics and Photonics.
[399] S. P. Rao, K. Panetta, and S. S. Agaian, "A novel method for rotation invariant palm print image stitching," in *SPIE Commercial+ Scientific Sensing and Imaging*, 2017, pp. 102210N-102210N-11: International Society for Optics and Photonics.
[400] R. Rajendran, S. P. Rao, K. Panetta, and S. S. Agaian, "Adaptive Alpha-Trimmed Correlation based underwater image stitching," in *Technologies for Homeland Security (HST), 2017 IEEE International Symposium on*, 2017, pp. 1-7: IEEE.
[401] Y. Wang, D. L. Lau, and L. G. Hassebrook, "Fit-sphere unwrapping and performance analysis of 3D fingerprints," *Applied Optics,* vol. 49, no. 4, pp. 592-600, 2010.
[402] G. Abramovich *et al.*, "Mobile, contactless, single-shot, fingerprint capture system," in *SPIE Defense, Security, and Sensing*, 2010, pp. 766708-766708-12: International Society for Optics and Photonics.
[403] R. Hess, *The essential Blender: guide to 3D creation with the open source suite Blender*. No Starch Press, 2007.
[404] T. Igarashi and D. Cosgrove, "Adaptive unwrapping for interactive texture painting," in *Proceedings of the 2001 symposium on Interactive 3D graphics*, 2001, pp. 209-216: ACM.
[405] K. C. Finney, *3D game programming all in one*. Cengage Learning, 2013.
[406] H. Guan, B. Stanton, A. Dienstfrey, and M. Theofanos, *A Measurement Metric for Forensic Latent Fingerprint Preprocessing*. 2014.
[407] Q. Zhao, A. Jain, and G. Abramovich, "3D to 2D fingerprints: Unrolling and distortion correction," in *Biometrics (IJCB), 2011 International Joint Conference on*, 2011, pp. 1-8: IEEE.

[408] S. Shafaei, T. Inanc, and L. G. Hassebrook, "A new approach to unwrap a 3-d fingerprint to a 2-d rolled equivalent fingerprint," in *Biometrics: Theory, Applications, and Systems, 2009. BTAS'09. IEEE 3rd International Conference on*, 2009, pp. 1-5: IEEE.
[409] S. Rajeev, S. K. KM, K. Panetta, and S. S. Agaian, "3-D palmprint modeling for biometric verification," in *IEEE International Symposium on Technologies for Homeland Security*, Waltham, MA, USA, 2017, pp. 1-6: IEEE, 2017.
[410] R. D. Labati, A. Genovese, V. Piuri, and F. Scotti, "Touchless fingerprint biometrics: a survey on 2D and 3D technologies," *Journal of Internet Technology*, vol. 15, no. 3, pp. 325-332, 2014.
[411] A. W.-K. Kong, D. Zhang, and G. Lu, "A study of identical twins' palmprints for personal verification," *Pattern Recognition*, vol. 39, no. 11, pp. 2149-2156, 2006.
[412] (2008). *The FBI's Next Generation Identification (NGI)*. Available: http://fingerprint.nist.gov/standard/presentations/archives/NGI_Overview_Feb_2005.pdf
[413] (2016). *hand scan* Available: http://creativity103.com/collections/Interference/slides/handscanner.html
[414] V. Kanhangad, A. Kumar, and D. Zhang, "A unified framework for contactless hand verification," *IEEE transactions on information forensics and security*, vol. 6, no. 3, pp. 1014-1027, 2011.
[415] S. P. Rao, R. Rajendran, S. S. Agaian, and M. M. A. Mulawka, "Alpha trimmed correlation for touchless finger image mosaicing," in *SPIE Commercial+ Scientific Sensing and Imaging*, 2016, pp. 98690U-98690U-12: International Society for Optics and Photonics.
[416] D. Zhang, W.-K. Kong, J. You, and M. Wong, "Online palmprint identification," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 25, no. 9, pp. 1041-1050, 2003.
[417] A. Pfitzmann, "Biometrie–wie einsetzen und wie keinesfalls?," *Informatik-Spektrum*, vol. 29, no. 5, pp. 353-356, 2006.
[418] A. K. Jain and A. Ross, "Bridging the gap: from biometrics to forensics," *Phil. Trans. R. Soc. B*, vol. 370, no. 1674, p. 20140254, 2015.
[419] "Fingerprints - Crime Museum," 2017.
[420] (2017). *greasy texture on boat, free photos, #1168708* Available: http://www.freeimages.com/photo/greasy-texture-on-boat-1168708
[421] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi, "Describing textures in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3606-3613, 2014.
[422] D. L. Hassebrook. *MAT5 DATA FORMAT*. Available: http://www.engr.uky.edu/~lgh/soft/softmat5format.htm
[423] (2017). *Biometric and Forensic Research Database Catalog*. Available: https://tsapps.nist.gov/BDbC/Search?page=10
[424] (2016). *The Hong Kong Polytechnic University Contact-free 3D/2D Hand Images Database version 1.0*. Available: http://www4.comp.polyu.edu.hk/~csajaykr/myhome/database_request/3dhand/Hand3D.htm
[425] A. Morales, M. A. Medina-Pérez, M. A. Ferrer, M. García-Borroto, and L. A. Robles, "LPIDB v1. 0-Latent palmprint identification database," in *Biometrics (IJCB), 2014 IEEE International Joint Conference on*, 2014, pp. 1-6: IEEE.

[426] (2017). *IIT Delhi Touchless Palmprint Database*. Available: http://www4.comp.polyu.edu.hk/~csajaykr/IITD/Database_Palm.htm
[427] S. K. K. M, S. Rajeev, K. Panetta, and S. Agaian, "Fingerprint Authentication Using Geometric Features," in *Technologies for Homeland Security (HST), 2017 IEEE Symposium on*

2017: IEEE.
[428] S. Agaian, P. Rad, R. Rajendran, and K. Panetta, "A Novel Technique to Enhance Low Resolution CT and Magnetic Resonance Images in Cloud," in *Smart Cloud (SmartCloud), IEEE International Conference on*, 2016, pp. 73-78: IEEE.
[429] D. Popescu, F. Popister, S. Popescu, C. Neamtu, and M. Gurzau, "Direct toolpath generation based on graph theory for milling roughing," *Procedia CIRP*, vol. 25, pp. 75-80, 2014.
[430] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *Image Processing, IEEE Transactions on*, vol. 13, no. 4, pp. 600-612, 2004.
[431] S. S. Agaian, K. Panetta, and A. M. Grigoryan, "Transform-based image enhancement algorithms with performance measure," *IEEE Transactions on Image Processing*, vol. 10, no. 3, pp. 367-382, 2001.
[432] Y. Zhou, K. Panetta, and S. Agaian, "Human visual system based mammogram enhancement and analysis," in *2010 2nd International Conference on Image Processing Theory, Tools and Applications*, 2010, pp. 229-234: IEEE.
[433] K. Panetta, Y. Zhou, S. Agaian, and H. Jia, "Nonlinear unsharp masking for mammogram enhancement," *IEEE Transactions on Information Technology in Biomedicine*, vol. 15, no. 6, pp. 918-928, 2011.
[434] (2011). *Development of NFIQ 2.0 | NIST*. Available: https://www.nist.gov/services-resources/software/development-nfiq-20
[435] (2010). *NIST Biometric Image Software (NBIS) | NIST*. Available: https://www.nist.gov/services-resources/software/nist-biometric-image-software-nbis
[436] E. Vig, M. Dorr, and D. Cox, "Large-scale optimization of hierarchical features for saliency prediction in natural images," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 2798-2805.
[437] S. Ouyang, T. Hospedales, Y.-Z. Song, X. Li, C. C. Loy, and X. Wang, "A survey on heterogeneous face recognition: Sketch, infra-red, 3d and low-resolution," *Image and Vision Computing*, vol. 56, pp. 28-48, 2016.
[438] E. Learned-Miller, G. B. Huang, A. RoyChowdhury, H. Li, and G. Hua, "Labeled faces in the wild: A survey," in *Advances in face detection and facial image analysis*: Springer, 2016, pp. 189-248.
[439] W. A. Bainbridge, P. Isola, and A. Oliva, "The intrinsic memorability of face photographs," *Journal of Experimental Psychology: General*, vol. 142, no. 4, p. 1323, 2013.
[440] R. Gross, "Face databases," in *Handbook of face recognition*: Springer, 2005, pp. 301-327.
[441] P. J. Phillips, H. Wechsler, J. Huang, and P. J. Rauss, "The FERET database and evaluation procedure for face-recognition algorithms," *Image and vision computing*, vol. 16, no. 5, pp. 295-306, 1998.
[442] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker, "Multi-pie," *Image and Vision Computing*, vol. 28, no. 5, pp. 807-813, 2010.
[443] A. M. Martinez, "The AR face database," *CVC Technical Report24*, 1998.

[444] J. Cohn, "Cohn-Kanade AU-coded facial expression database," *Pittsburgh University*, 1999.
[445] A. Kasinski, A. Florek, and A. Schmidt, "The PUT face database," *Image Processing and Communications*, vol. 13, no. 3-4, pp. 59-64, 2008.
[446] B. Zhang, L. Zhang, D. Zhang, and L. Shen, "Directional binary code with application to PolyU near-infrared face database," *Pattern Recognition Letters*, vol. 31, no. 14, pp. 2337-2344, 2010.
[447] http://www.sic.rma.ac.be/~beumier/DB/3d_rma.html. [Online].
[448] S. Gupta, K. R. Castleman, M. K. Markey, and A. C. Bovik, "Texas 3D face recognition database," in *IEEE southwest symposium on image analysis and interpretation*, 2010, pp. 97-100.
[449] S. Z. Li, D. Yi, Z. Lei, and S. Liao, "The casia nir-vis 2.0 face database," in *Computer vision and pattern recognition workshops (CVPRW), 2013 IEEE Conference on*, 2013, pp. 348-353: IEEE.
[450] H. Maeng, S. Liao, D. Kang, S.-W. Lee, and A. K. Jain, "Nighttime face recognition at long distance: Cross-distance and cross-spectral matching," in *Asian Conference on Computer Vision*, 2012, pp. 708-721: Springer.
[451] S. Wang *et al.*, "A natural visible and infrared facial expression database for expression recognition and emotion inference," *IEEE Transactions on Multimedia*, vol. 12, no. 7, pp. 682-691, 2010.
[452] K. Messer, J. Matas, J. Kittler, J. Luettin, and G. Maitre, "XM2VTSDB: The extended M2VTS database," in *Second international conference on audio and video-based biometric person authentication*, 1999, vol. 964, pp. 965-966.
[453] A. Colombo, C. Cusano, and R. Schettini, "UMB-DB: A database of partially occluded 3D faces," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, 2011, pp. 2113-2119: IEEE.
[454] M. Meyer and G. Kuschk, "Automotive radar dataset for deep learning based 3d object detection," in *2019 16th European Radar Conference (EuRAD)*, 2019, pp. 129-132: IEEE.
[455] F. Yu *et al.*, "Bdd100k: A diverse driving dataset for heterogeneous multitask learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2636-2645.
[456] H. Caesar *et al.*, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11621-11631.
[457] P. Sun *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2446-2454.
[458] C.-É. N. Laflamme, F. Pomerleau, and P. Giguère, "Driving datasets literature review," *arXiv preprint arXiv:1910.11968*, 2019.
[459] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "The KITTI vision benchmark suite," *URL http://www. cvlibs. net/datasets/kitti*, vol. 2, 2015.
[460] G. Neuhold, T. Ollmann, S. Rota Bulo, and P. Kontschieder, "The mapillary vistas dataset for semantic understanding of street scenes," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 4990-4999.

[461] X. Huang *et al.*, "The apolloscape dataset for autonomous driving," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 954-960.
[462] J. Houston *et al.*, "One thousand and one hours: Self-driving motion prediction dataset," *arXiv preprint arXiv:2006.14480*, 2020.
[463] M. Cordts *et al.*, "The cityscapes dataset," in *CVPR Workshop on the Future of Datasets in Vision*, 2015, vol. 2.
[464] J. Geyer *et al.*, "A2d2: Audi autonomous driving dataset," *arXiv preprint arXiv:2004.06320*, 2020.
[465] A. Palazzi, D. Abati, F. Solera, and R. Cucchiara, "Predicting the Driver's Focus of Attention: the DR (eye) VE Project," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 7, pp. 1720-1733, 2018.
[466] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical automated data augmentation with a reduced search space," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 702-703.
[467] (2020). *Build The BEST Security Camera NVR: Free Locally Processed AI Computer Vision with Blue Iris*. Available: https://www.youtube.com/watch?v=fwoonl5JKgo
[468] FLIR. *FLIR ADK-Thermal Vision Automotive Development Kit*. Available: https://www.flir.com/products/adk/
[469] W. Lee, N. Park, and W. Woo, "Depth-assisted real-time 3D object detection for augmented reality," in *ICAT*, 2011, vol. 11, no. 2, pp. 126-132.
[470] FLIR. *FREE FLIR Thermal Dataset for Algorithm Training*. Available: https://www.flir.com/oem/adas/adas-dataset-form/
[471] FLIR. (2020). *How do thermal cameras work?* Available: flir.com/discover/rd-science/how-do-thermal-cameras-work/
[472] P. Labs. *Pupil Invisible*. Available: https://pupil-labs.com/products/invisible/
[473] M. Tonsen, C. K. Baumann, and K. Dierkes, "A High-Level Description and Performance Evaluation of Pupil Invisible," *arXiv preprint arXiv:2009.00508*, 2020.
[474] A. Rosebrock. (2020). *Blur and anonymize faces with OpenCV and Python*. Available: https://www.pyimagesearch.com/2020/04/06/blur-and-anonymize-faces-with-opencv-and-python/
[475] L. Vallantin. (2019). *Anonymize facial data on video: blur people's faces using OpenCV*. Available: https://medium.com/swlh/anonymize-facial-data-on-video-blur-peoples-faces-using-opencv-14ce5ed1f1aa
[476] Q. Zhu, M.-C. Yeh, K.-T. Cheng, and S. Avidan, "Fast human detection using a cascade of histograms of oriented gradients," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, 2006, vol. 2, pp. 1491-1498: IEEE.
[477] M. Arsenovic, S. Sladojevic, A. Anderla, and D. Stefanovic, "FaceTime—Deep learning based face recognition attendance system," in *2017 IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY)*, 2017, pp. 000053-000058: IEEE.
[478] L. Y. Chan, A. Zimmer, J. L. da Silva, and T. Brandmeier, "European Union Dataset and Annotation Tool for Real Time Automatic License Plate Detection and Blurring," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1-6: IEEE.
[479] L. Xue-bin SUN Xuan-chao, "Method of blurring vehicle license plate enhancement and location based on texture and color analysis," *Microcomputer Information*, vol. 9, 2009.

[480] P. P. Rao and R. K. Muthu, "A new de-blurring technique for license plate images with robust length estimation," in *2017 International Conference on Intelligent Computing and Control (I2C2)*, 2017, pp. 1-6: IEEE.
[481] (2020). *CAD-BLOCK.COM-CAD Library of free DWG Blocks and premium AutoCAD drawings*. Available: https://cad-block.com/
[482] B. John, S. Koppal, and E. Jain, "EyeVEIL: degrading iris authentication in eye tracking headsets," in *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications*, 2019, pp. 1-5.
[483] D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman, and A. K. Jain, "FVC2004: Third fingerprint verification competition," in *Biometric Authentication*: Springer, 2004, pp. 1-7.
[484] D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman, and A. K. Jain, "FVC2004: Third fingerprint verification competition," in *International conference on biometric authentication*, 2004, pp. 1-7: Springer.
[485] (2017). *Home - Biometrics Research at University of Silesia*. Available: http://biometrics.us.edu.pl/

Appendix

This dissertation developed foundational architectures that solve a complex problem of human visual intelligence-based systems using multi-modal and multi-dimensional data. However, the availability of labeled multi-modal and multi-dimensional datasets proved to be the main limitation of this dissertation.

Datasets and testbeds are critical for researchers and AI developers to conduct training and testing on real-world operational data. Despite the advancement and widespread usage of machine learning, much work has to be done regarding bias, diversity, and inclusion within datasets. These machine learning examples are often supervised, requiring humans to manually label data. However, even unsupervised machine learning algorithms, which process massive amounts of data without human intervention, have limitations.

The author notes that most current AI systems specialize in analyzing RGB data. However, AI is capable of discerning image data in other regions of the spectrum, as shown in chapter 4. Furthermore, AI models are trained using data collected under ideal conditions. Thus, while the algorithms may perform well in bright daylight when objects and scenes are visible to the sensor, they may perform poorly when the scene's visibility is compromised by insufficient lighting or adverse weather conditions such as rain, snow, haze, or fog.

In chapters 2 and 4, the author noted a significant lack of labeled eye-tracking datasets. In chapter 2, the authors could attempt to develop a novel gaze map-based semantic segmentation algorithm due to the lack of any publicly available semantic segmentation datasets that included fixation data. Chapter 4 recognized the lack of real eye-tracking datasets with free world movements. Moreover, most eye-tracking datasets are collected in restricted and controlled environments.

In chapters 5 and 6, a significant lack of good quality depth datasets was identified. Moreover, most depth datasets that are collected are indoor-based. In chapter 7, the author identified a lack of comprehensive and good-quality 3D biometric databases.

While new datasets, in the order of millions of images, were developed in this dissertation, they could solve only some of the challenges described above. Acquiring high-quality datasets for machine learning is a challenge that can be overcome by efficient data integration, governance, and exploration until all data needs are met consistently.