ABSTRACT

Title of Thesis:	RECORDING TO REDESIGN: REDESIGNING A SITE USING CAMERA VISION TO OPTIMIZE PEDESTRIAN FLOW
	Benjamin Phoenix Morrison, Master of Landscape Architecture, 2025
Thesis Directed By:	Dr. Christopher D. Ellis, Professor in the Department of Plant Sciences and Landscape Architecture

Examining how people interact with public space is an important dimension of design. As technology has advanced, we have used it to capture how people move and interact in public spaces. Harnessing technology allows landscape architects to leverage the power of computers to empower their designs and increase the effectiveness of designed space. This thesis explores two uses of such technologies: using computer vision to explore how students move through a public space and calibrating a microscopic pedestrian flow model based on collected pedestrian traffic data. These technologies are used to show how such a site might be redesigned to make a space more effective. The site that has been selected to do this is the westernmost plaza of McKeldin Mall at the University of Maryland. The site is adjacent to the McKeldin Library and home to the Testudo sculpture.

RECORDING TO REDESIGN: REDESIGNING A SITE USING CAMERA VISION TO OPTIMIZE PEDESTRIAN FLOW

by

Benjamin Phoenix Morrison

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Master of Landscape Architecture
2025

Advisory Committee:
Professor Christopher D. Ellis, Chair
Professor David M. Barbarash
Professor David N. Myers

Dedication

This thesis is dedicated to Stephen Morrison and Carol Fishback, my loving parents whose belief in my academic aptitude and ability to succeed at times were the primary pillars of support on which this thesis stands. Much of what follows stands as a testament to their commitment to supporting their son.

Acknowledgments

Before getting into the bulk of this thesis, I would like to thank all of those who have been critical in developing this project. Thank you to everyone who supported me on this journey. And it *was* a ***journey***. Anyone who has completed a master's thesis or similar project knows that for every word written, graphic created, sources cited, and line coded, there were many times more words deleted, graphics abandoned, sources unused, and lines of code discarded. If it were easy, everyone would do it. Thankfully, I am blessed with many people in my life who have been more than generous with their time and support. I want to acknowledge them before continuing with my work here.

To Prof. Christopher Ellis, thank you for your support throughout this thesis process. Your willingness to enable this unconventional technology-based thesis really has meant the world to me. You have been a steady hand throughout this thesis process even as it took me months to complete what I thought would be done in a matter of weeks.

To Prof. David Barbarash, without your work, I would be 10 steps behind where I am currently. You lead the way for the use of this technology in our field and your guidance, feedback, and support are immensely appreciated.

To Prof. David Myers, without you I would not be in the Master of Landscape Architecture program at the University of Maryland. You have been a constant presence and pillar of support throughout my journey at the University of Maryland. It was and is a privilege to have you on my thesis committee.

To my parents, Carol Fishback and Steve Morrison, thank you for "blocking" for me and providing me the time and space to get this thesis done.

To my partner, Emma Heck, thank you for being such an understanding partner and for tolerating me going off the grid at times and zoning in on my work. Your affection and steadfast dedication are greatly appreciated.

To my cohort, the past three years have been such a blessing in my life. It has been so wonderful to know each and every one of you. Even though we have respectively siloed into our areas of research, I hope that we will remain in contact in the years to come.

And finally, to the many unnamed others who have supported me on my journey, thank you.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

2-D: Two-Dimensional

3-D: Three-Dimensional

AI: Artificial Intelligence

ABM: Agent-Based Model

AR: Augmented Reality

CAD: Computer-Aided Design

CNN: Convolutional Neural Network

GCP: Ground Control Point

GIS: Geographic Information Systems

GPS: Global Positioning System

IBM: Individual Based Model

IMU: Inertial Measurement Unit

IPM: Inverse Perspective Mapping

LiDAR: Light Detection and Ranging

MOT: Multi-Object Tracking

ML: Machine Learning

MVS: Multi-View Stereo

Re-ID: Re-Identification

RFID: Radio-Frequency Identification

SfM: Structure from Motion

SLAM: Simultaneous Localization and Mapping

ToF: Time-of-Flight

VR: Virtual Reality

YOLO: You Only Look Once

Part 1: An Overview

In addition to the typical division by chapter, for ease of understanding and navigation, this Master of Landscape Architecture thesis is divided into several parts. Each section contains one or more chapters and parts are based on similarity in content within chapters. The structure of the body of this thesis follows:

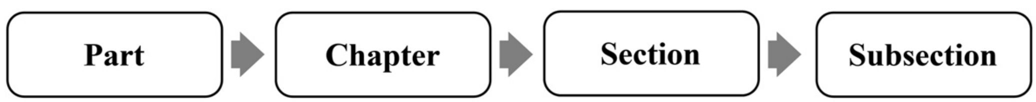


Figure 1: Structure of Thesis, illustrated

The following are summaries of each of the parts:

Part 1: An Overview—covers an overview of the subjects dealt with in this thesis, from the thesis's beginnings to a chronology of the study of urban space.

Part 2: Technologies in Focus—covers the basics of the technologies used in this thesis or relevant to the field of study.

Part 3: On the Ground—details the practicalities of using this technology, from analyzing site characteristics to overcoming difficulties in its implementation.

Part 4: Analysis and Results—deals with the results from the implementation of this thesis and discusses areas of further research.

As with most theses, this paper is broken down into a number of chapters. The following are summaries of each of the chapters:

Part 1: An Overview

Chapter 1: Introduction—introduces the thesis and some of the central questions posed in it.

Chapter 2: The Study of Small Urban Spaces—goes over the history of the study of pedestrians in urban environments and how that study has evolved to the present day.

Chapter 3: Cameras, Computers, and Design—gives a brief introduction to how computers can be used by landscape architects and design professionals at large.

Part 2: Technologies in Focus

Chapter 4: An Introduction to Computer Vision—gives a cursory overview of computer vision and discusses different detection and classification algorithms.

Chapter 5: Digitizing Reality—describes the different possible processes for taking photo and video recordings (two-dimensional representations of reality) and translating them into simulated two-dimensional space.

Chapter 6: How People Move—details the science of pedestrian flow and dynamics and goes over the different approaches used to describe and predict how people move through space.

Part 3: On the Ground

Chapter 7: McKeldin Mall—goes over the physical details of the site selected as well as a site history and analysis.

Chapter 8: Technical Issues—details the process this thesis took to implement the technologies and the challenges involved in doing so.

Chapter 9: Redesigning the Site—gives a brief use case of how the technology might be used to redesign outdoor space.

Part 4: Analysis and Results

Chapter 10: Discussion—details the results of this thesis as well as areas of future research.

Chapter 11: Conclusion—covers high-level analysis and takeaways from this thesis.

The body of this thesis follows. Part 1 of the thesis pertains to all overview materials required to understand this thesis from how this thesis's conceptualization came about to some of the challenges that this thesis had to overcome.

Chapter 1: Introduction

This chapter is written in first person, unlike most of the body of this thesis, as it deals with the author's personal experiences. This thesis only returns to the first person for the conclusion, Chapter 11: Conclusion.

This Master of Landscape Architecture thesis focuses on studying people's movement through and within small urban spaces. The question of how people move through space is one that I believe is central to the field of landscape architecture. Traditionally, when factoring in how users will potentially interact with a design, landscape architects have relied on their own experience -- a north star, if you will. Conversely, to see how landscapes are used, landscape architects have relied on expensive post-occupancy evaluations. In this thesis, I will explore newer tools to do what landscape architects have always done. I essentially use computers to automate the observation of people across a small urban area and then predict how people will move across that area using that data. The exact methods that will be used will be discussed later in this paper.

Central to my thesis is the following research question: How much of an individual's movement in small urban spaces can we predict knowing only where they entered a site, and can we use that information to design better small urban spaces? Subquestions include the following:

- What environmental factors induce movement in people, and how does that induced movement present itself?
- What information about people can we gather from their movement alone?
- What digital technologies could be harnessed to gather data about how people move, and what technologies could be used to predict movement based on that data?

With these questions in mind, we can examine why they might be important for landscape architects to study; one might ask, what is the purpose of this study? Put simply, the purpose of this Master of Landscape Architecture thesis is to study the very basics of how people interact with a small urban space on a granular scale and to see if those results can yield predictions that inform design structures and systems. To reiterate, for landscape architects, predicting how a site that you are designing will be used is more of an art than it is a science. It relies heavily on a landscape architect's previous experience in the field. Thus, design is primarily a matter of experience or prior learning. Moreover, and somewhat in keeping with this thought, finding out how a site is actually being used is typically the domain of traditional post-occupancy evaluation, which can often be costly and time-consuming. This research paper aims to develop predictive modeling of site visitor behavior and to use it to streamline design development and facilitate rapid post-occupancy evaluation using automated site analysis. This thesis's aim is to test out a technology in a relatively simple environment. In essence, it is a pilot study.

Good spatial design, we know, takes into account how people move through space. Designing a landscape, as stated before, often involves drawing on experience, specifically people's experience interacting with the world around them. With that in mind, major efforts have been made to understand how people interact with the landscape around them, specifically how people behave in small urban spaces. Over time, while the goals of these studies have remained the same, even the methods used to evaluate have evolved with the progress of time.

Some of the seminal work that has taken place in the study of small urban areas has been done by William Whyte. Whyte's recordings of people using small urban spaces and his conclusions based on his observations are the foundation of the study of such spaces (Whyte, 1980). Additional researchers have made serious contributions to this field of study. Jan Gehl and Birgitte Svarre have pushed the needle forward in their book, How to Study Public Life. There, they discuss several methods of recording small urban spaces (Gehl & Svarre, 2013). As of recently, new computer vision techniques are beginning to be deployed in the study of such spaces. Global teams, from China to the United States, have begun work on using these technologies (Hou et al., 2020). This thesis largely follows and attempts to build on the research of Prof. David Barbarash of Purdue University.

While research into this area has made significant headway, there is still much to move forward on. In a 2020 paper, Prof. Barbaresh listed his ten research goals, ranging from "Develop a robust multi-camera reidentification system based around video camera data (Barbarash & Lu, 2020)" to developing an artificial intelligence-based computer actor to simulate how a person might interact with a given landscape. To date, much of the research in this area revolves around the third step. Thus, there is a significant opportunity for increased research and development in this domain.

With all of this in mind, one might ask: why is this important to the field of landscape architecture? Answer: The goal of research in this domain will make the lives of landscape architects easier and it will make the work of many landscape architects better. Imagine what landscape architects could do if they knew, with a high degree of certainty, how people would move through a space that they are creating. Imagine the mistakes that would be prevented and the creativity of low-cost or cost-free design that automation and predictive models could enable. Stated another way, this research can help landscape architects better understand how sites are being used currently and drive down the cost of different "what if" scenarios. Additionally, if costs of post-occupancy evaluations are brought down significantly via automated site analysis, there could and should be a significant increase in the amount of data available to landscape architects to see if any given intervention is effective.

Ultimately, due to the more modest scale of this thesis, I will have to make a couple of major assumptions about how people move across small urban areas. First, I will assume that all individuals' characteristics are drawn from a singular distribution. Demographic information will not be collected and approximate indicators will not be used. For example, age, race, sex, etc. will not affect how likely someone is to be a fast walker in the predictive model that will be created (and discussed later). Artificial Intelligence (AI) person recognition systems have difficulty predicting demographic information, such as age and race, even in ideal situations. Since the study will involve the use of outdoor cameras, most of the time the system will be operating under non-ideal circumstances making that effort especially difficult. Ideal circumstances for person recognition systems typically are when the person is close to the camera in good lighting with minimal background clutter. Second, the study assumes that all site features of a given type are equivalent; for example, all paved areas are equally likely to be walked on or all water features are equally attractive. This assumption is made in order to keep the predictive model as simple as possible.

Due to the assumptions made, the study will not take into account how demographics affect someone's interaction with a small urban space. Additionally, it will not cover how effective any individual site feature is at performing its function.

Chapter 2: The Study of Small Urban Spaces

Landscape architects and other design professionals like urban planners and architects have been interested in the study of small urban spaces as long as their professions have existed. In the last half-century, it has become an area of increasingly rigorous research, and the study has evolved with technology and time. Progress in this field and advancements are critical to our future as cities become increasingly compact, complex, and densely populated and the effective use of land becomes all the more critical.

In recent history, the study of pedestrian behavior has evolved into a more rigorous and demanding discipline as it has been driven by evolving technology, allowing a deeper understanding of pedestrian dynamics. Researchers in the field are able to implement sophisticated tools such as image detection, algorithms, GPS tracking, LiDAR scanning, cell tower data collection, etc., and use them to capture real-time movement data and analyze intricate interactions in near real-time. This data-driven technological approach allows a more granular study and possibly greater understanding of how pedestrians navigate space and respond to various stimuli in addition to how they interact with their surroundings. Data collected in this manner may be immensely useful for interdisciplinary teams, including architects, data scientists, landscape architects, mathematicians, urban planners, psychologists, and more... Design professionals, architects, landscape architects, and urban planners may be critical to our future, and using data to create safer, more vibrant, more efficient, and more enjoyable urban areas that cater to the needs of their users will likely only increase.

The world is becoming more urban, and we need to design our urban spaces more efficiently and effectively. Cities are becoming the home to the majority of the global population (Elmqvist et al., 2018; *World Urbanization Prospects*, 2019). The World Bank predicts that by

2050, 7 in 10 people will like in cities (*Urban Development: Overview*, n.d.).The way that they are structured directly impacts the quality of life of a huge portion of our species. Critical to this endeavor is the creation of high-performing urban spaces that foster community, promote public life, and are implemented sustainably. These spaces are essential services that encourage economic activity and contribute to the resilience of urban environments.

In order to design and create these spaces, an effective understanding of how people move through them is critical. Modeling pedestrian behavior in the pursuit of effective urban space is equally important. It may greatly enhance the effectiveness of the designs that make it into our urban landscape. The advanced data collection techniques listed previously, in addition to other sources like social media sentiment analysis and landscape performance metrics, may be used to improve design. Landscape Designers may be able to make these places increasingly safe and vibrant. By mixing design and big data, designers may be able to create tailor-made urban environments that have a greater likelihood of being effective.

2.1 William Whyte's Seminal Work

Whyte (1980) has been instrumental in shaping the understanding of how public spaces can contribute to social interactions in an urban environment. Whyte's primary motivation for conducting his research was to explore the relationship between designed space and observed social behavior. The focus of *The Social Life of Urban Spaces* was on small urban spaces such as parks, plazas, and street corners. He did this all at a time when urban development often prioritized vehicular traffic over the pedestrian experience. In studying how people exist in these spaces, Whyte was able to understand the subtlety of human interaction and the importance of good design to encourage social engagement. He realized that vibrant public spaces could nurture a sense of community and promote well-being for urban inhabitants.

Figure 2: William Whyte recording and performing his observational study, picture by the Project for Public Space (https://www.pps.org/article/wwhyte)

Whyte employed methods that were available to him at the time to conduct his observational study. He combined observation from a high vantage point, as well as on-the-ground observations, to get a good understanding of how space was used or occupied. His approach used quantitative and qualitative analysis, counting the numbers of people, observing their interactions, and noting how they utilize the space. This observational methodology proved valuable for his pursuit of defining the significance of well-designed small urban areas. Due to his meticulous fieldwork, Whyte was able to gather large amounts of compelling data about how different elements such as arrangement, seating, landscape, and availability of amenities directly influenced the social behaviors of users of small urban areas.

A key conclusion for Whyte was that the presence of seating was a huge factor in creating social interaction. He found that well-placed chairs and benches could increase the number of people congregating in a small urban space, which, in turn, often created spontaneous conversations and community interactions. Additionally, he noticed the importance of visual connections within space, suggesting the importance of clear sight lines and openness to foster a sense of safety and inclusivity. Finally, he also observed that certain features, like water elements and food stalls, attracted people and were able to create increased vibrancy in these areas. Many of his observations pointed towards a need to design spaces that prioritized the human experience.

Whyte's work laid a foundation for future research in urban studies and the study of small urban spaces. He illustrated how careful planning and designing of these places could lead to numerous gains for both the individual and society at large. His use of empirical observation and data collection were pioneering for the study of small urban spaces. It paved the way for subsequent studies that have used ever-increasing technological tools to collect and process data regarding how people engage with these sites, fostering an area of research that can be used to increase social engagement, inclusivity, and a sense of belonging for inhabitants.

2.2 Modern Studies

In recent decades, the study of pedestrian movement and pedestrian dynamics in small urban areas has evolved by leaps and bounds. Building upon the foundational insights presented by William Whyte's seminal work, *The Social Life of Small Urban Spaces*, where Whyte was able to use analytical tools to describe the importance of seating, sunlight, and other landscape features on social interaction and community engagement, today's researchers have been able to continue to explore these dynamics, using advanced technology in order to create even deeper insights into how people engage with space.

Modern studies are able to harness a variety of technological tools in order to analyze pedestrian dynamics. One such tool is the use of video analytics to monitor foot traffic patterns in real-time (Barbarash et al., 2022). By using cameras and employing machine learning algorithms, researchers are able to track movement, gather demographic data, and analyze social interactions. This data allows design professionals like architects, landscape architects, and urban planners to enhance the pedestrian experience and improve the usability of small urban spaces.

Video and image analytics are not the only ways of collecting pedestrian data and data on pedestrian behavior. Some studies include both Bluetooth and Wi-Fi tracking in order to capture movement patterns anonymously, revealing how different areas of space are utilized over time (Suraweera et al., 2021). These methods can provide insights into peak usage times and behavior. Other studies use LiDAR to preserve privacy and handle some of the challenges inherent to translating photography into 3-D space on a computer (Ohno et al., 2023).

Data from various sources can be integrated into geographic information systems, allowing architects, landscape architects, and urban planners to visualize how people move through space at larger scales, identifying patterns and trends that could guide future development.

Additionally, another emerging area of research involves the use of both virtual reality and augmented reality tools to simulate pedestrian interactions (Deb et al., 2017). This technology enables planners to create models of potential urban designs and analyze how they might perform in the real world without the need to perform physical alterations to the space. In particular, virtual reality allows users to be immersed in these redesigned spaces, allowing researchers to gather valuable feedback on what features could encourage social interaction and what features could discourage the use of that space.

Ultimately, all these technological advances build upon the rigorous principles established by Whyte. They allow for more nuance in the understanding of pedestrian behavior in urban areas. They allow data to fine-tune our understanding of how people move through space and potentially refine our actual understanding of human nature. As cities continue to grow and expand, the weight carried by these modern technology-driven observational studies will likely carry more and more weight. However, despite the immense challenge, this represents a promising path toward creating more and more livable, engaging, and dynamic urban centers. The ultimate goal remains the same: to analyze and create public spaces that serve the needs of the community, foster social engagement, and create lively, dynamic, vital social hubs for our increasingly urbanized world.

2.2.1 Urban Plazas: Revisited

Figure 3: SWA's Plaza Life Revisited, picture by SWA Group (https://www.swagroup.com/idea/plaza-life-revisited/)

SWA group's *Field Guide to Urban Plaza: A Study in New York* acts is a follow-up to William White's seminal research (Schlickman & Domlesky, 2019). It provides an in-depth exploration of how a number of urban plazas in New York function as vital social and community spaces for the urban fabric. By examining a number of well-known plazas, the guide is able to capture the dynamics of social interaction and explore a variety of factors that contribute to the vibrancy of public areas. SWA group's study emphasizes the role of design in fostering community engagement and, similarly to Whyte, highlights elements such as seating and landscaping. Additionally, it focuses on accessibility as a function of encouraging people to gather and interact.

Through the use of observational research, the guide is able to identify certain characteristics that make plazas more or less successful than others. Key findings in this domain are that plazas with ample seating and visual connectivity often attract higher foot traffic and promote more spontaneous social interactions than plazas that lack those qualities. Additionally, the presence of amenities, such as cafés or public arts, can further enhance these spaces, encouraging their use by diverse groups of people. SWA group noted the importance of thoughtful design in creating small urban areas as focal points for urban community life.

Additionally, the guide emphasizes the importance of continual observation and adoption of public urban design in order to meet the evolving and ever-changing needs of the community. Integrating real-time feedback and data-driven approaches, urban designers can use their understanding of pedestrian behaviors to create more flexible spaces that respond to the dynamic needs of the city. Ultimately, this report serves as a valuable follow-up to Whyte's earlier work.

2.2.2 Prof. David Barbarash's Work

Prof. David Barbarash is a professor who takes a holistic approach to the questions this thesis seeks to answer. His work with his research team at Purdue University is looking at the complete picture and has begun designing a pipeline and a lens through which one can begin to solve some of these questions (Barbarash, 2021). Working since 2018, this team is comprised of undergraduate researchers in landscape architecture, computer science, electrical and computer engineering, and applied statistics. Their focus is on evaluating human behaviors in built environments. Currently, they focus on controllable spaces on campuses that allow them to address privacy concerns while developing a database of human behavior influenced by spatial, cultural, and environmental factors. Ultimately, the database will aid in creating AI-driven actors to simulate interactions within environments that have yet to be built. The goal is to allow designers to explore how their different perspective designs might affect social behaviors. Barbarash's team has been developing a multi-camera computer vision system built for post-occupation evaluation to collect this data.

Barbarash & Lu (2020) lays out a 10-step process that needs to be accomplished for a complete system to be realized:

1. Develop a robust multi-camera reidentification system based around video camera data
2. Record attributes of people and objects within a camera's vision while maintaining privacy of people in frame
3. Map the location of people within a defined area of the built environment in three dimensions with corresponding temporal data included
4. Observe video of people within a given space and record behaviors of occupants linked with their AI provided ID codes. This requires expert opinion from environmental and social psychologists, sociologists, etc.
5. Develop and distribute a network of sensor packages that measure temperature, humidity, air quality, light intensity, wind flow, and other environmental factors
6. Compare human behaviors and activity with designed intent and environmental conditions of a constructed space, developing a library of designerly terms and programs that influence human use of space (attractors, repulsors, thresholds, gathering areas, movement areas, calm spaces, energetic spaces, etc.)

7. Develop an AI actor model within an existing video game engine (Unreal, CryEngine, Unity, etc.)
8. Recreate a real space within the game engine and run the simulation, comparing AI behaviors with those of the real site
9. Make a structural change in the digital model and analyze how it changes use of a space
10. Recreate the same change in the real environment and compare the digital and real in a test of accuracy and validity

The multi-camera computer vision system built for post-occupation evaluation involves a time synced video collection system, using multiple cameras to identify human figures using bounding boxes, rectangular regions on an image where people are located. The system creates a series of cropped images assigned to each individual. Each individual is additionally assigned a unique ID. The coordinates of identified individuals are then converted into real-world location data.

This methodology is unique in that it uses a very modular multi-camera setup. This enhances accuracy and coverage. Furthermore, it records metadata that can aid in post-occupancy evaluations. These attributes include articles of clothing, modes of transportation, and objects carried. By analyzing patterns of movement, the system can distinguish between different behavior groups such as solitary or group movement.

The data generated from this system is output in vector format which is compatible with open street maps. Ultimately, the team aims for an open-source release but first they need to ensure stability in the release.

Chapter 3: Cameras, Computers, and Design

We are in the age of computers. There is no way that we can get around them. As designers, our challenge is not to question whether or not computers are inherently good for our profession but, instead, how computers can be used to advance our goals and agendas. These powerful machines are, in fact, able to aid in our collective pursuit of good design and meaningful spaces. We just have to know enough about them in order to use them effectively. This chapter discusses two applications of technology and how they can be used in our field. These technologies are camera vision and pedestrian flow models.

Before giving an overview of these technologies, this paper will discuss how they and technology in general relate to landscape architecture.

3.1 Computers in Design

In recent decades, the introduction and integration of computers and software technologies have significantly transformed a variety of design fields. Particularly with regard to landscape architecture, some of the most transformative have been Computer-Aided Design (CAD) and Geographic Information Systems (GIS).

Prior to the mid-20th century, landscape architects were primarily drafting manually. However, after the introduction of desktop computers in the 1970s, this began to change in a major way.

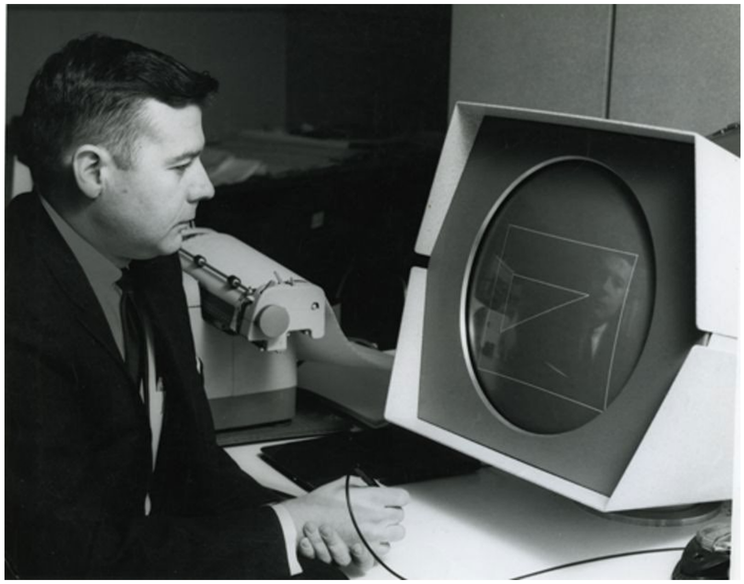


Figure 4: User interacting with an early Computer Aided Design (CAD) program running on a PDP-1 computer, 1962 ca., picture by the Computer History Museum (https://www.computerhistory.org/pdp-1/f190e5632731c66de256ff6b15c8b864/)

The first significant advance came with the development of Computer-Aided Design (CAD) software in the 1980s. CAD allowed landscape architects to draft more precisely and in a more standard format. This technology improved both the efficiency and accuracy of plans created by landscape architects, allowing more complex designs.

In the 1990s, Geographic Information Systems (GIS) technologies began to cement themselves as a powerful tool in landscape architecture. GIS enabled landscape architects to analyze and visualize spatial data, allowing for new, unique ways of landscape analysis and design. This capacity remains crucial for large-scale site analysis as well as planning. GIS allows landscape architects to combine and layer different sources of data, such as topography, soil types, vegetation, impervious cover, etc., to analyze and design in unique ways.

Additionally, the introduction of 3-D modeling software and rendering tools into the field of landscape architecture has allowed for more immersive presentations and design. This technology allows landscape architects to communicate better with people outside of the field, notably clients and stakeholders. Additionally, these visualizations often also help in the landscape architect's own iterative design process.

In recent years, newer technologies have begun infiltrating into the field. These technologies include drone surveying, LiDAR point clouds, and virtual reality drones that can provide unique aerial views of sites. LiDAR point clouds allow sites to be recorded in an incredibly detailed fashion. They allow for the immersive experience of simulated designs letting designers and clients alike experience the landscape before construction even begins.

Over the years, the impact of computers and software technologies on landscape architecture has been profound. The tools listed above have enhanced design, capacities, and increased efficiency in the field. Furthermore, they have allowed for new and innovative approaches to arise. As technology continues to evolve and percolate into landscape architecture, the field will likely be pushed and stretched into new and unexpected ways.

3.2 Designing with Simulation

Landscape architects can use simulation in their designs in a number of unique and interesting ways. Increasingly, computer simulation is employed in the design process. These tools can allow designers to create virtual models of landscapes and explore spatial relationships, user interactions, and environmental conditions in detail (Kinateder et al., 2018).

Using computer simulation design has the unique ability to allow designers to analyze the impact of design choices before implementation and construction of their designs, potentially saving clients significant amounts of money and potentially lives (Angulo et al., 2023; Schwebel et al., 2008). Simulation tools can be employed for a wide variety of conditions, such as pedestrian and vehicle movement, weather conditions/flooding, air quality, temperature, etc. These simulations can be enhanced by the addition of real-world data.

Using simulations allows landscape architects to explore the long-term implications of their design, seeing how a design might be used as well as how it might be affected by different climate scenarios: heat distribution, air, quality, water, runoff, etc. This allows landscape architects to potentially vastly increase the effectiveness of their designs. Additionally, computer simulations can facilitate stakeholder engagement by providing a visual means for landscape architects to present their designs(Ehab et al., 2023; Lusco et al., 2018). This can be integrated into a participatory approach, increasing community buy-in.

Computer simulation can serve as an invaluable tool for landscape architects, allowing for comprehensive analysis, increased stakeholder engagement, and design resilience. By harnessing these powerful tools, landscape architects can push the bounds of their creativity, creating projects that are more functional, inclusive, and environmentally responsible. As the field continues to progress, the integration of such technologies will likely play an ever more critical role in its practice.

3.3 Challenges to Overcome

Gathering pedestrian data presents a unique set of challenges that directly impact the usefulness and accuracy of pedestrian dynamics models. Human behavior in public space is an incredibly variable phenomenon. Pedestrian movement is influenced by a cornucopia of factors, including social and environmental conditions as well as unique individual, spontaneous, and decision-making processes. To capture this variability, models often require sophisticated data collection methods to account for these complexities. Unfortunately, traditional observational techniques can fail to capture transient behaviors such as group formation or spontaneous changes in directions as well (Mamidipalli et al., 2015, p. 230).

However, as the data is collected, there is a chance of introducing biases that may ultimately limit the applicability of the results (Brandao, 2019). For example, using fixed cameras and positions can provide lots of visual data but uneven information density across a site (Fan & Loo, 2021, p. 17). Equally, tracking mobile and personal devices can raise concerns regarding privacy and consent(Michael et al., 2006).

Even after data is collected, new challenges can arise. Most models rely on simplifications that do not always accurately represent complexity in the real world (Adrian et al., 2019, p. 6). This may lead to inaccuracies in predicting individual and crowd behavior. Additionally, the calibration and validation of these models require comprehensive large data sets that must accurately reflect diverse conditions under which pedestrians exist. As a result of this, models need to be continually refined. This interplay between data collection and modeling is critical for ongoing research in the field of pedestrian dynamics.

3.3.1 Detection and Identification Problem

One of the problems faced when studying pedestrian dynamics is the process of gathering pedestrian movement data. It presents a significant challenge and, if done improperly, can hinder the accuracy and applicability of any given model (Moreira et al., 2010).

One of the primary difficulties in data collection lies in the fact that human movement is fundamentally, to some level, unpredictable. It is ultimately influenced by a myriad of factors and requires robust data collection methodologies.

The scale and granularity of data collected are highly dependent on the data source. In order to track pedestrian movement in high resolution, complicated technologies such as video surveillance, must be implemented.

Further, while some technologies assume that all data relates to pedestrians, this is not true for every data collection technique. Both imagery and LiDAR data require post-processing to extract pedestrian information. In order to extract pedestrian information from these data sources, advanced algorithms, like those used in computer vision, must be implemented. The quality of data extracted can be greatly affected by a number of factors, notably variability in the person's appearance, lighting conditions, background, complexity, camera, angle, motion, blurring, bodily position, etc (Enzweiler & Gavrila, 2009).

3.3.2 2-D to 3-D Placement Problem

One of the main sources of pedestrian data is imagery. Using imagery to obtain pedestrian data presents a significant challenge. Translating positions on an image into 3-D real-world coordinates is far from straightforward.

Images only capture the projected silhouette of 3-D objects. This means that crucial data about special relationships and distances is inherently compromised; “since one dimension is lost in the projection from the three-dimensional world onto the two-dimensional imaging surface” (Ma et al., 2004, p. 2). Intuitively, we can understand that two objects that appear close together in an image may, in reality, be very far apart. Lacking depth complicates the task of reconstructing 3-D positions from 2-D data immensely.

Several additional phenomena take place that complicate the process of 3-D placement. Perspective means that cameras view object that are closer as larger than objects that are further away. Additionally, they may be significantly warped depending on their depth. Further, occlusion can happen; it is the process of one object partially or completely blocking another. Objects can be displaced as a result of parallax, which is the fact that the apparent positions of an object can change depending on line-of-sight.

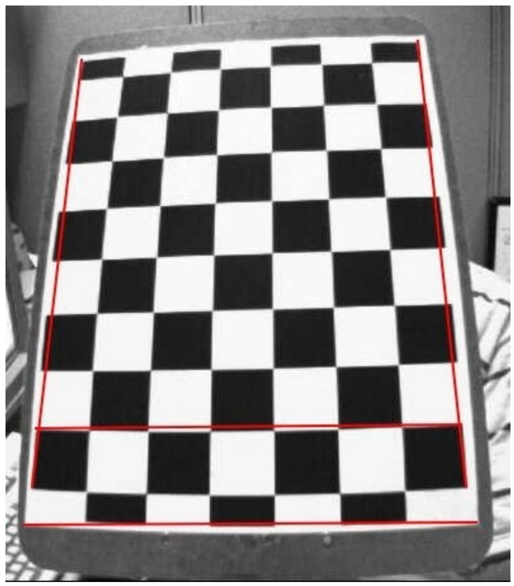


Figure 5: Example of Radial Distortion, by OpenCV (https://docs.opencv.org/4.x/dc/dbb/tutorial_py_calibration.html)

Additionally, most cameras involve some level of distortion. This can change the perceived shape of an object immensely, correcting this distortion, which is a critical challenge in the placement of detections in a 3-D coordinate space.

Another challenge is estimating the position of the camera in real-world space. The camera's pose can dramatically affect where the calculated 3-D points are.

3.3.3 Modeling and Simulation Problem

Modeling pedestrian behavior and motion is an incredibly challenging task due to the complex and dynamic nature of how humans exist. No model can perfectly predict the inherent variability that individuals have in their behavior. Pedestrians exhibit a wide range of reactions to their environment based on a cornucopia of personal factors such as ability, age, cultural background, and emotional and psychological state. This variability makes it effectively impossible to create a perfect one-size-fits-all model.

When modeling pedestrian movement, the unpredictability of environmental factors also presents a large hurdle. External conditions, such as weather, time of day, and environment, dramatically impact how people move through space. For example, poor weather conditions can lead to slower or altered movement (Dong et al., 2022; Fossum & Ryeng, 2021). Conversely, damaged walkways might influence a pedestrian's route selection. These conditions can result in unexpected behaviors that are hard for a model to anticipate.

The movement of pedestrians oftentimes presents non-linearly. People avoid obstacles or change direction dramatically in response to social conditions. This can dramatically complicate the modeling process.

It is also very close to impossible to perfectly quantify the mental state of a given pedestrian as well as their exact social relation to other pedestrians around them. As one might expect, many models that deal with psychology or the social relations of pedestrians simplify these aspects of the human condition dramatically. Researchers have amended models, like the Social Force Model, to account for social contagion in panic situations (Ren et al., 2023).

3.3.4 Subproblems

In addition to these problems, there are a set of some problems, perhaps lesser in scope, but each challenging in its own right. Each subproblem demands nuanced exploration and careful thought.

Tracking Subproblem

Tracking pedestrian movement between detections presents complex challenges. These challenges are highly dependent on whether or not the detections have been moved into real-world space or are still in the form of their raw data.

When tracking people in between frames of a video, a number of different problems arise. As people move through space, the position of their bodies changes. Additionally, lighting fluctuates and shadows move. Furthermore, no video frame image is perfect and motion blurring can alter detected features.

When tracking pedestrian movement using data placed in 3-D space, the ultimately unpredictable variability in pedestrian movement patterns can become a challenge. One needs to balance the usefulness of eliminating misdetections based on special proximity with the chance of missing a reidentification based on erratic movement.

For a variety of reasons, data sets can be incomplete. For example, in imagery data, pedestrian detections can be affected by occlusion where either they are fully concealed or partially concealed from view. This can result in a missing detection at a given time or place or a misplaced detection.

Ultimately, all data sources have a certain amount of noise and level of inaccuracy. This can result in either false positives or false negatives (Zheng et al., 2016, p. 9). Addressing this challenge requires sophisticated algorithms that can often integrate multiple data sources, and even then, motion can be missed, undetected, or misunderstood.

Insufficient Data Subproblem

When handling data regarding pedestrians, data collected often does not fully match the needs of a given model. For example, a given model might require velocity data, but the data that has been collected only contains position data. Conversely, a model might use a timestep different from the data collected. In order to have useful pedestrian data, some transformation must be made.

If more or different data is needed, one can use synthetic or approximate data. Synthetic data is data that does not have a real-world origin and often is algorithmically generated. It can be useful to test out methodologies when ground truths cannot be assessed. Conversely, approximate data might be used if data does not contain needed variables. Approximate data can be called many things: fit, inferred, predicted, etc. “Approximate” is being used as a catch-all term.

In order to use a model, the creation of approximate data is often a necessity. This is an inherently limited methodology, as approximate data relies on assumptions that may not be present in a given environment. This can lead to overall system biases and result in major discrepancies between modeled outcomes and ground truths. This can be a result of the assumptions made to create the data. Approximate data can miss critical nuances, leading to unreliable predictions and misinformed decision-making.

Possibly, the largest problem with approximate data is bias. If the underlying methodology for the creation of approximate data is inaccurate, the biases that it can create propagate across the entire system.

Additionally, differences in granularity between synthetic data and real data can lead to misleading insights. Approximate data is often inherently less granular than direct observations. A potential outcome of using models that are over-reliant on approximate data can lead to overconfidence in estimations, which could potentially have major real-world consequences.

Parameter Estimation Subproblem

Every pedestrian dynamics model is reliant on a series of finely tuned parameters. These parameters govern how the model predicts pedestrian behavior. Calibrating for these parameters requires either empirical study or enough data to calibrate a model for a given site. The difference between these two approaches is detailed in Section 6.7.

3.4 Underlying Assumptions

There are a number of common underlying assumptions in this domain of research. On the computer vision side, images are assumed to have valuable information that can be derived. This assumption would not be true for randomly generated static. Computer vision models assume that observations in a new image will be similar enough to observations in previous images used in training. For example, a person in a picture that a model perceived will look similar enough to people in pictures that it has already seen for it to identify that pattern as a person. Additionally, computer vision models often assume uniform lighting (Gevers et al., 2012, p. 11; Ye et al., 2025). How light reflects off of an object is not assumed to vary by the viewing angle. Further, these models assume that the underlying data sets that they are trained on are accurate representations of real-world observations and that there are not any major biases in their underlying data sets. This assumption has been challenged consistently, particularly when it comes to social biases of datasets (gender, race, etc.) (Brandao, 2019; Schwemmer et al., 2020; Torralba & Efros, 2011).

Conversely, pedestrian dynamics models are rooted in some major assumptions about human behavior. Fundamentally, they all assume that pedestrians behave rationally and somewhat predictably in response to their environment. For example, a number of pedestrian dynamics models are built off the premise that individuals will attempt to minimize costs associated with time, effort, and personal space in order to get to a certain desired destination (Helbing, 1998, p. 2; Tong & Bode, n.d., p. 4).

Computer vision models rely on assumptions about the coherence and density of visual information and their training data is representative of future data input into them. Pedestrian dynamics models rest upon the assumption that people move in rational ways that can be, to some degree, predicted (Helbing, 1998, p. 2). Ultimately, both of these models attempt to simplify complex problems.

3.5 Movement Towards a Unified Methodology

There have been various efforts to create unified multi-scale systems for pedestrian detection and behavior modeling. These efforts involve integrating complex processes at many different levels accounting for individual choice and movement patterns as well as large-scale crowd dynamics. Researchers in this domain employ models at various macroscopic, microscopic, and mesoscopic models to capture different aspects of pedestrian behavior (Cristiani et al., 2014). Combining the predictive capacity of the models above, new hybrid models are developed allowing information and predictions to flow between different scales.

Additionally, these researchers use complex data-driven approaches to obtain pedestrian movement patterns from various data sources such as imagery (Enzweiler & Gavrila, 2009), GPS (Korpilo et al., 2017), Wi-Fi (Suraweera et al., 2021) and Bluetooth (Gu & Ren, 2015), LiDAR (Ohno et al., 2023), etc. Sometimes, nontraditional data can be collected, such as social media analytics (Chen et al., 2018) or real-time weather data (Dong et al., 2022).

These methodologies are all combined to create the most robust system of pedestrian behavior modeling possible.

Part 2: Technologies in Focus

In the following part, each of the technologies involved in the three main challenges that this thesis focuses on will be broken down and described. While this thesis focuses specifically on one subsection of these broader technological groupings, it is important for one to understand what is being done. However, the methodology used here is just one grouping of such technologies. There is a myriad of potential implementations.

Part 2 covers chapters 4 through 6, taking you through the process of recording, recording people, getting detections from those recordings, placing those detections in 3-D space, and fitting a predictive or explanatory model to the observed data. This part serves more as a scoping review of these technologies and less as a specific methodology for this thesis. If so desired, the chapters in this part can be skipped and used as a reference for later work specific to the implementation presented in this thesis.

Chapter 4: An Introduction to Computer Vision

The following chapter will cover the basics of computer vision. Computer vision is a field that combines several disciplines, from computer science to data science to optical engineering. The overarching goal of computer vision is to replicate tasks that the human visual system can perform digitally. The tasks that computer vision systems handle range from recognizing individual objects to understanding images/scenes as a whole. To perform this task, computer vision systems typically implement machine learning algorithms, particularly convolutional neural networks using deep learning, to process complex visual data and recognize patterns in it (Voulodimos et al., 2018; X. Zhao et al., 2024).

In the context of identifying people in a designed landscape, computer vision holds a lot of promise. Object detection and classification systems are designed to analyze and locate human figures. There are a number of large data sets to train models on where humans are located in an image. The object detection model that is used later (YOLOv3) was trained using the COCO (Common Objects in Context) training set on a Darknet-53 backbone.

Additional systems exist to analyze visual data further with respect to human subjects. These systems include facial recognition systems and bodily pose detection. Facial recognition systems can be used to identify individuals, and bodily pose is a good predictor of how people move. That said, graphic fidelity is required to be implemented accurately. Either of these systems makes it prohibitive for them to be implemented in the context of a landscape. Such systems need clear lines of sight (Saleh et al., 2021), high-quality and high-resolution images (Hao et al., 2023), and consistent and good lighting to function properly (Ghari et al., 2024). Additionally, they can be thrown off by slight changes to the human appearance, whether that be makeup in terms of facial recognition (Dantcheva et al., 2012) or loose clothing in terms of bodily pose (Wuhrer et al., 2014).

While more granular techniques like the ones listed above still require time to mature in the landscape context, computer vision systems that detect and classify objects in images can be useful tools for landscape architects. These systems are particularly useful for locating and tracking pedestrians and modeling their behavior.

Using image data paired with computer vision holds a lot of promise. However, it is not without its problems. If implemented at a large scale, this technology could have serious privacy implications. That being said, there are solutions to these concerns that have been proposed (Ni et al., 2024).

4.1 Types of Computer Vision Models

Computer vision models are not a one-size-fits-all approach. Different models have different scopes, characteristics, and goals. These range from looking at an image as a whole and deciding whether or not certain objects are present in it to deciding if that is true on a pixel-by-pixel basis.

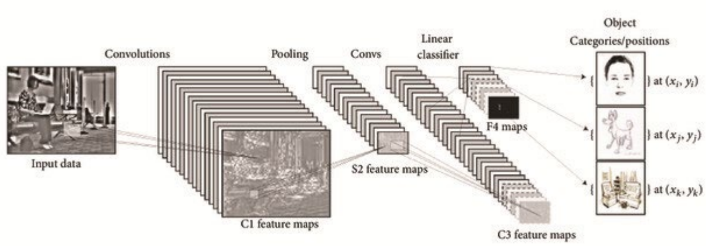


Figure 6: Example architecture of a CNN for a computer vision task (object detection), from (Voulodimos et al., 2018) (Figure 1)

Typically, these models are built using convolutional neural networks (CNNs), which are "…kind(s) of feedforward neural network that (are) able to extract features from data with convolution structures" (Li et al., 2022, p. 7000). These machine-learning algorithms are used to extract features critical to the task of computer vision from imagery.

4.1.1 Classification vs. Detection

One major delineation in computer vision models is between image classification models and object detection models.

Classification

Model Name	Year(s) Developed	Notes
AlexNet	2012	Introduced deep CNNs to image classification, winning the ImageNet competition.
VGGNet	2014	Uses deep stacks of small (3x3) convolutional layers for high accuracy but is computationally expensive.
GoogLeNet/Inception	2014	Introduced the Inception module to improve computational efficiency.
ResNet	2015	Uses residual connections to enable very deep networks
DenseNet	2017	Enhances information flow by connecting each layer to every other layer.

Table 1: List of Popular Classification Models

Classification models are some of the simplest computer vision models. The task of classification is primarily to answer the question: How confident are you that this image contains X object where X is an element from a predetermined list of possible elements? Simple image classification models just flag whether or not there is an instance of a given element in an input image. A number of classic and seminal computer vision models are of the classification variety.

Detection

Model Name	Year(s) Developed	Notes
R-CNN	2014	Region-based CNN for object detection.
Fast R-CNN	2015	Improved speed and accuracy over R-CNN.
Faster R-CNN	2016	Improved speed and accuracy over Fast R-CNN.
YOLO	2016-Present	You Only Look Once models process images in a single pass, optimizing speed and accuracy.
SSD	2016	Single Shot MultiBox Detector balances speed and precision for real-time detection.

Table 2: List of Popular Object Detection Models

The task of detection is one step more advanced than classification. Object detection models not only decide whether or not certain elements are contained within an image but also determine where that object is located. These models answer the following question: where on this image does X object exists, if at all? If you determine X exists, how confident are you that it does exist, and where does X exist as an element from a predetermined list of possible elements.

The most popular and widespread computer vision models are object detection models. These models output in the form of a bounding box, meaning a top, bottom, left, and right bound extend for a region where the detection exists in a given input image along with a confidence level in that detection.

4.1.2 One-stage vs. Two-stage

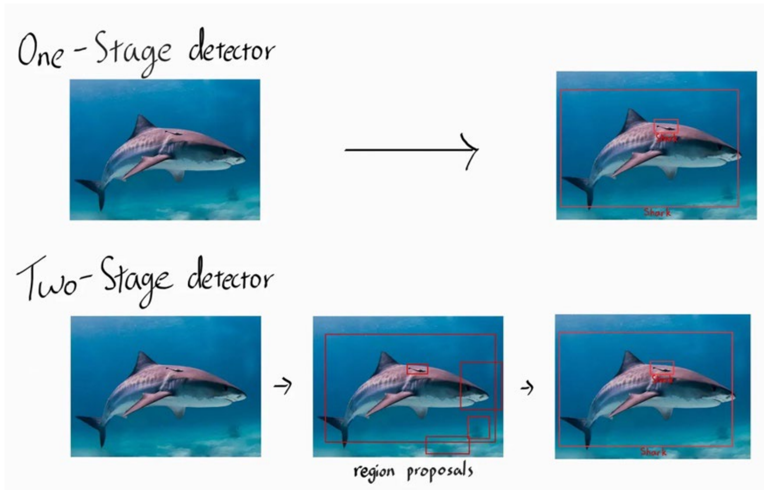


Figure 7: Stages of One-Stage vs. Two-Stage object detector computer vision models, illustrated, image by SharkYun on Medium (https://sharkyun.medium.com/computer-vision-object-detection-one-stage-vs-two-stage-b05dbff88195)

The next big difference between models relates to their internal architecture. This key difference really determines whether a computer vision model is designed to operate quickly or accurately (Padilla et al., 2020).

One-stage

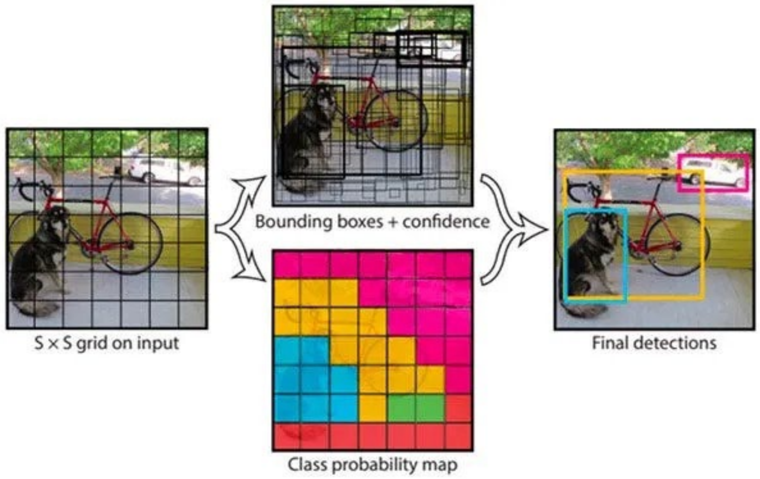


Figure 8: Steps in YOLO one-stage object detection model, illustrated (https://sharkyun.medium.com/computer-vision-object-detection-one-stage-vs-two-stage-b05dbff88195)

Sometimes called single-pass models, one-stage models divide a given input image into a grid and simultaneously generate detections.

One-stage models are known for their speed. They are ideal for scenarios when high speed and low latency are critical. A major use case for these types of models is in autonomous vehicles, where milliseconds make a difference in potentially life-or-death situations. While one-stage models were designed with speed in mind, recent advancements in one-stage technology have seen these models narrow the gap in terms of accuracy with their two-stage peers.

Two-stage

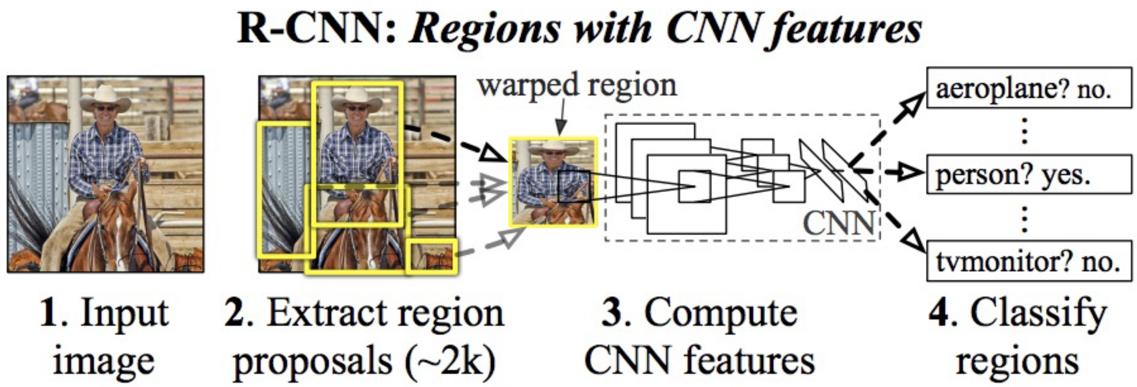


Figure 9: Steps in R-CNN two-stage object detection model, illustrated
(Girshick et al., 2014, p. 1)

Two-stage models work similarly to one-stage models. However, as may be expected, they add an additional step. Before generating detections, these models generate region proposals, areas of an image where an object is suspected of existing. They then individually check these regions for detections.

Figure 10: The process of using segmented region proposals to generate candidate object locations, illustrated, image by GeeksForGeeks (https://www.geeksforgeeks.org/selective-search-for-object-detection-r-cnn/?ref=blog.roboflow.com)

Two-stage models are useful when increased accuracy is required. This is especially true when an image has visually complicated information. A critical use case for these types of models is medical image analysis, where high detection accuracy is essential.

4.1.3 Segmentation Models

Model Name	Year(s) Developed	Notes
FCN	2015	Fully convolutional network for pixel-wise segmentation.
U-Net	2015	Used primarily for biomedical image segmentation.
DeepLab	2017-2022	Introduced Atrous convolutions and Conditional Random Fields for segmentation.
Mask R-CNN	2017	Extends Faster R-CNN by adding a segmentation branch.

Table 3: List of Popular Segmentation Models

Segmentation models are a special type of classification model. The theory behind them is similar, save one major exception: They execute the classification task for each pixel individually on an image.

These types of models are most useful when the boundary of a detection in an image needs to be predicted in a highly specific, granular form.

4.1.4 Vision Transformers

Model Name	Year(s) Developed	Notes
Vision Transformer (ViT)	2020	Applies transformer architecture to images, performing well on large datasets.
Swin Transformer	2021	Uses a hierarchical approach for better efficiency and accuracy.

Table 4: List of Popular Vision Transformers

A visual transformer is a model that uses a transformer-like architecture, meaning it aggregates patches of an image and processes them simultaneously. When matched with large amounts of parallel processing power, these models can work with incredible amounts of data.

These models are best used when massive amounts of data need to be processed quickly and one has the resources to do so via parallel processing. Another use case for these models happens to be pedestrian gender prediction which they have been shown to be effective at (Abbas et al., 2023).

4.2 Detecting Movement

When tracking movement, the relationship between detections becomes incredibly critical. Ultimately, the question of movement boils down to: Is this person in this image the same person as that person in a previous image? The process of detecting movement is a process of detection and re-identification.

4.2.1 Re-Identification (Re-ID)

Re-identification (Re-ID) is the process by which detections are categorized as coming from a mutual source in the real world. Re-ID systems are typically performed visually in an image-based way. However, location-based systems also exist, tracking how detections move spatially rather than via physical information.

Image-based

Image-based Re-ID systems visually match new detections to previous detections. They compare the proximity between two different matches, and if it passes a predetermined threshold, the detection is identified as coming from the same source. The way that detections can be compared is by performing a feature extraction, where things like clothing or hair color and texture, body shape, or gait are encoded into a higher-dimensional space. Once feature extraction is performed, a process of ranking and re-ranking is used to compare scores (Sun et al., 2024).

Location-based

Location-based Re-ID systems operate on the principle that location data is a useful indicator of whether an object is the same. Location-based systems have been combined with image-based systems to dramatically increase the accuracy of the Re-ID process (Spurlock & Souvenir, 2015; Wang et al., 2018). This process works on an intuitive basis, where detections should not be identified as the same when that would imply that our object detected is moving in a way that is not physically possible.

4.3 Other Sources of Pedestrian Data

It is important to note that data about pedestrian movement is not only limited to data sourced from imagery taken at or near the ground plane. Pedestrian data can be sourced from various technologies and methodologies that are not in the domain of ground-based image and video surveillance. Imagery can come from different sources and that imagery in turn needs to be processed differently. Data also doesn't have to be from imagery at all. Data can be sourced from signals, such as GPS, Wi-Fi, or Bluetooth. Useful data might not always be directly containing pedestrian movement. Public sentiment, demographic factors, and environmental conditions also play a role in pedestrian behavior but are not always accounted for in modeling endeavors.

4.3.1 Other Imagery Sources

Collecting data about pedestrian movement through imagery, as previously stated, extends beyond just ground-level sources. In addition to standard photography and video on the same plane as pedestrians, aerial perspectives from drones or satellite imagery can enhance data collection efforts (Yeom & Cho, 2019). Drones can capture high-resolution images from a variety of altitudes, allowing for clear lines or angles of detection that might not exist on the ground plane. Conversely, satellite data covers large areas, allowing for larger-scale insights.

Drone

Drone imagery can be a powerful tool for collecting pedestrian data, offering unique advantages over ground-based imagery methods. Drones can cover large areas in short amounts of time. Additionally, providing data from a bird's-eye view can reveal movement patterns that might not be observed or observable from the ground plane. Drone imagery, similar to ground-based sources, can be accessed in real time. Another advantage of using drone imagery is that drones nearly always come with an inertial measurement unit. Measurements of this type can be used to find the camera's pose at the time of the creation of an image. This information can be immensely valuable, depending on the methodology by which one places image-based detections in 3-D space.

Drone imagery offers many of the advantages that ground-based imagery offers. The disadvantage is that ground-based sensors can be extremely stable, allowing simpler 3-D placement techniques to be more robust. Additionally, processing drone data can be more complicated and lead to lower detection accuracy (Liao et al., 2024). This is a result of objects changing in appearance due to the shifting perspectives of a drone at different elevations. The same computer vision model may have a tough time identifying a person from eye level and from above their head. That being said, there are models that exist to address this exact issue (Yatim et al., 2017).

Satellite imagery

Satellite data can also be a useful source for pedestrian data. Satellite imagery provides a broad top-down view of a landscape. Large historical records of satellite data are publicly available. This data can be used for large-scale modeling of pedestrian movements by modeling at the macroscopic scale.

Two downsides to using satellite data are that images from satellites tend to be spread out temporally, and the resolution of satellite data is often much lower than terrestrial sources. Identifying an individual person via a satellite image is a challenging endeavor, and in all likelihood, that person will be nowhere near where they were previously upon the next image capture. As a result, data from this source is much more useful at an extremely large scale.

Additionally, most users do not have immediate access to satellite data and analyses performed on satellite data often are delayed due to the amount of data, data handling challenges, and sometimes proprietary concerns. Data collected via this source has mostly been used for larger-scale, longer-term endeavors such as urban planning or infrastructure development (Eikvil et al., 2009; Golej et al., 2024, p. 1; Ibrahim et al., 2021).

4.3.2 Audio Sensors

Audio sensors have shown promise as a source of pedestrian data in recent years (Han et al., 2024). While audio sensors have been used successfully to monitor vehicle traffic, using audio sensors to track pedestrian movement is a nascent technology (Han et al., 2024). Using audio and seismic sensor to collect data about pedestrian movement may hold a lot of promise as a cheaper alternative to other detection methods, but still lacks the same accuracy that other methods, like imagery, have (Choudhary et al., 2022).

4.3.3 LiDAR

Ground-based LiDAR (light detection and ranging) can be a useful source of pedestrian data. Unlike imagery, it has three-dimensionality built into it. Ground-based LiDAR systems eliminate a lot of the guesswork that is required in imagery-based solutions. Due to depth being a dimension of data collected via LiDAR, pedestrian insights can be placed locationally and often immediately. The only downside to these systems is that they are not widespread, and such systems do not always have a corresponding imagery component, meaning point clouds generated by them lack visual information. This could mean that common camera vision techniques would not work with this data.

4.3.4 Device-Based Sources

Data can also be sourced directly from devices that are on pedestrians themselves (from cellphones, tablets, laptops, and other smart devices). This provides a set of unique advantages and disadvantages. The placement of these devices is variable between different individuals, and not every pedestrian has one of these devices, while some pedestrians might carry several.

Global Positioning System (GPS)

GPS data has the unique advantage of being already located geospatially. It can come from cell phones or other wearable devices such as smartwatches or fitness trackers (Korpilo et al., 2017). When anonymized, this data can be incredibly useful (McArdle et al., 2014). However, before accessing this type of data for pedestrian behavior analysis, important questions and privacy concerns must be addressed. Handling this data must be transparent, extreme caution must be used, and a significant degree of good judgment and oversight is required.

Radio Frequency Identification (RFID)

RFID devices are a good source of locational data due to these devices use radio frequency (L. Zhao et al., 2025). They do not require a line of sight. Their radio waves pass directly through barriers such as windows, walls, and doors. Detections from these devices do not suffer from the same challenge that comes in the form of occlusion that imagery-based detections suffer from. Additionally, detections can be placed in 3-D space easily if there are more than four scanners spread out.

Wi-Fi/Bluetooth

As smart devices become increasingly ubiquitous, Wi-Fi and Bluetooth data become an increasingly valuable source of pedestrian data. Most phones automatically communicate with nearby Wi-Fi networks and Bluetooth devices. These systems can be used to very accurately predict the number of people there are in a receiver's radius (Suraweera et al., 2021). These devices have unique digital footprints that can be used to re-identify detections across multiple networks (Traunmueller et al., 2018). Network traffic from these devices can also be used as an additional source of pedestrian information.

While Wi-Fi and Bluetooth have some unique advantages for collecting pedestrian data, they come with some significant drawbacks when used as observational techniques. It is almost impossible to use this data to get spatial information. Rarely are multiple overlapping networks available to gather position information and even if there were, all sorts of barriers can affect the signal strength of both Wi-Fi and Bluetooth signals (Gu & Ren, 2015; Waqas et al., 2024). Additionally, data from these technologies come with serious privacy concerns that are similar to those associated with GPS data (Ziegeldorf et al., 2014).

4.3.5 Alternative Data Sources

Direct observational data is not the only useful source of data regarding pedestrian interaction. Other sources of data can also indicate how pedestrians interact with the environment around them.

These indirect data sources are not typically introduced into pedestrian modeling. However, as the field of pedestrian behavior simulation advances, they may become increasingly useful. As such, they have been included as potential data sources for this thesis.

Social Media

Social media can be a useful tool for getting a sense of public sentiment. Much of social media content generation can be tied to a location even if it is just finding out the country that a post or data is coming from. Most data from these sources is locationally granular. For example, reviews on Google Maps are directly tied to location data. This data can be useful in quantifying and qualifying site usage (Chen et al., 2018; Fisher et al., 2018).

Surveys

Surveys, especially those that collect data about demographic information, can be useful for understanding how pedestrians might behave on a site. These surveys give useful information about who a site's user might typically be. This is an important factor in how individuals move through space and how they might interact with their environment. Experimentally, this has also been shown to be true and should be strongly considered for integration into pedestrian modeling (Papadimitriou et al., 2016).

Weather

Surprisingly, weather data is not typically an input into pedestrian dynamics models. Landscape architects intuitively know that how people move is very much dependent on weather conditions (de Montigny et al., 2012). People do not typically travel, visit, or linger when conditions are inhospitable or uncomfortable, no matter what the space around them is or who is present in it (Arens & Bosselmann, 1989).

Chapter 5: Digitizing Reality

3-D reconstruction from imagery is a mathematically rigorous domain. Much of this chapter was written consulting Ma et al. (2004).

After answering the first challenge listed in this thesis, detecting a pedestrian in an image, a second challenge arises: how to take a detection located in 2-D space and move it into 3-D space.

5.1 Methods of translating 2-D to 3-D space

There are several methods for taking points in a 2-D space (in this case, an image) and mapping them to a 3-D space (a point in the real world). The processes to perform this task can be divided into three categories, depending on the type of data available. There are methods using data only from a single camera, methods using data from multiple cameras, and methods supplemented by depth-based measurements.

5.1.1 Single Camera Methods

If only data from a single camera is available, single-camera (also called monocular vision) methods must be used.

Inverse Perspective Mapping

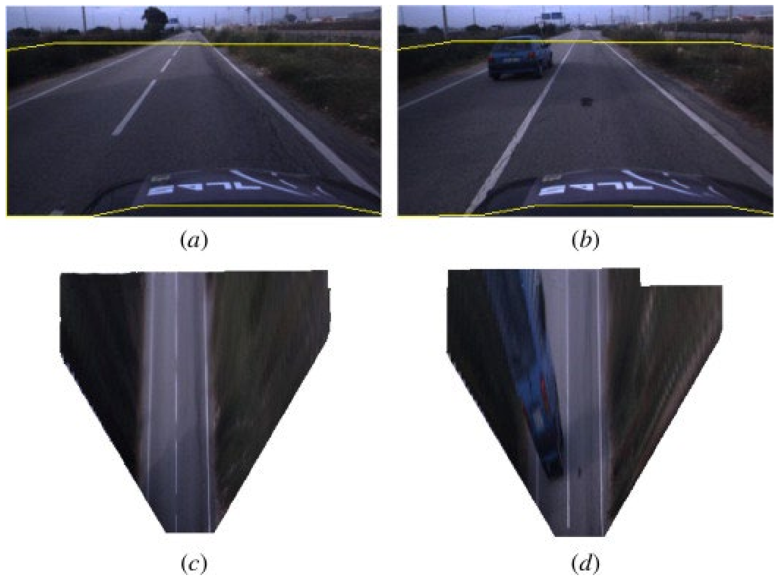


Figure 11: Inverse Perspective Mapping Illustrated (Oliveira et al., 2015)

Inverse perspective mapping is the process of converting 2-D points to 3-D points by assuming that the points lie on a flat plane. The basic mathematics behind this process is homography, the mathematics of planes in perspective. This is the simplest way of translating a 2-D point into a 3-D one (Oliveira et al., 2015).

Known Object Size

Another method for placing a point in 3-D space is to know the size of the object. This information, along with the camera's pose and focal length, can be used to place an object in 3-D space.

For the purpose of human detection, this process is not the most useful as people are not of uniform size. That said, such a method could be useful in validating detections by calculating the perceived heights of pedestrians.

Structure from Motion (SfM)

This method estimates depth in an image by identifying the same 3-D points across multiple images taken in different positions. The biggest challenge with this method is feature tracking across frames, which can be a computationally intensive process (Huang et al., 2024).

Neural Network-Based Depth Estimation

The final method listed here to get 3-D information from a 2-D image is to use a deep learning model-based approach (Kumar et al., 2018). These methods are based on data sets of similar-looking images with known depth values. This method works best when conditions are similar to the training data set. This method has been proven useful for both autonomous vehicles and augmented reality applications and is already starting to be used for pedestrian detection and tracking (Mancusi et al., 2022).

5.1.2 Multi-Camera Methods

If two or more cameras are used and their positions relative to each other are known, multi-camera (also called stereo vision) methods can be implemented. These methods require images to be taken at the same time. This method can be more precise than single camera systems.

Stereo Triangulation

This method uses epipolar geometry to compute the depth from the difference in position? of the same point across two images.

Multi-View Stereo Triangulation

This method works similarly to the stereo triangulation method but can factor in additional cameras.

Structured Light

This method uses a known pattern of light (typically a laser grid) to get useful locational information. The displacement and warping of the pattern are used to calculate depth.

5.1.3 Hybrid Methods

Another methodology for translating 2-D points into 3-D space is to use some form of depth mapping. A depth map is an image where each pixel represents the distance between the image plane and the surface patch corresponding to that pixel (Haralick & Shapiro, 1994, p. 70). In essence, how far away the pixel is. Using depth mapping and camera pose, points are placed in 3-D space.

Image-to-LiDAR

There are methods that use both image and LiDAR data sources (Schlosser et al., 2016). This method maps 2-D points onto the closest LiDAR depth measurement.

Time-of-Flight (ToF) Imaging

Depth location can also originate from Time-of-Flight (ToF) Cameras, which measure the time it takes light to bounce back from objects to compute depth. That said, this methodology works best for close-range applications.

5.2. Geospatial Coordinate Mapping

Once a method, or some combination of methods has been chosen to map 2-D points in 3-D space, those points can be transformed into a geospatial reference system.

Camera Pose Method

If the camera's position (latitude, longitude, altitude) and orientation (yaw, pitch, roll) are known, depth measurements can be used to project the point into 3-D space.

Ground Control Points (GCPs) Method

This method involves identifying known geospatially referenced points in an image and using homography or affine transformations to map the image to real-world coordinates. GCPs can also be used for the inverse perspective mapping (IPM) method.

5.3 Camera Pose Estimation Methods

Camera calibration and position/pose estimation are essential steps in many of the above methods. In essence, these methods require knowledge of a camera's position and orientation before they can map 2-D points to 3-D space. Below are listed a few methods for calculating a camera's position, orientation, or both.

Image-Based

Image-based techniques compare different images to estimate the camera pose when each image was taken relative to the other cameras. There are different methods to do this, from solving the perspective-n-pose problem to machine learning approaches (Kendall et al., 2016; Lu, 2018). The central approach to this method is feature tracking: detection and mapping. Using these detected features, the geometric relationships between how they change can be used to calculate the relative rotation and translation of camera pose between different images. Camera positions and orientations calculated this way are in an arbitrary local reference frame.

LiDAR

Aligning a camera's image data with 3-D point cloud data generated by a LiDAR sensor can extract the camera's pose with respect to that 3-D point cloud. Camera positions and orientations calculated this way are in an arbitrary local reference frame.

Global Positioning System (GPS)

Using GPS data to calibrate camera position can greatly increase the ease of projecting 3-D points in a geospatial reference system. Unfortunately, GPS data does not help calculate a camera's orientation.

Inertial Measurement Unit (IMU)

An IMU contains both angular velocity data and linear acceleration data. As such, data from it can be used to track changes in a camera's orientation and position. This data, when matched to a reference pose or fit to a series of them, can be used to calculate both position and orientation for camera frames.

Chapter 6: How People Move

The previous two chapters dealt with methods of collecting and transforming pedestrian data into something useful for the study of pedestrian movement. This chapter will discuss how models, specifically computational models, describe and predict pedestrian movement behavior.

This field of modeling combines elements from various disciplines, from physics to psychology to urban design, to model pedestrian behavior. Models in this domain can be helpful in multiple applications, from pedestrian trajectory analysis to evacuation bottleneck mitigation. The information derived from these models can guide the design of walkways, street crossings, and passageways (Fernández-Arango et al., 2024; López Baeza et al., 2021; *Tool for Predicting Pedestrian Flow Expands Its Reach*, 2021). In the context of landscape architecture, modeling can inform site layout and predict site usage and wear over time. Pedestrian models can and should play a pivotal role in designing space in the 21st Century.

6.1 Overview of Pedestrian Models

Computational modeling of pedestrian behavior is an area with a diverse variety of applications, from urban planning to transportation engineering to architecture to emergency management (Papadimitriou et al., 2009). Pedestrian models have found their usefulness in these domains. While computational modeling of pedestrian dynamics has been a field of study for almost 50 years, getting its start in the late 70s, this domain is in a revolutionary period as dramatic increases in computational power and new and innovative methods of analyzing large quantities of data have become accessible to more and more researchers.

Pedestrian models can be broadly categorized into three distinct scales: macroscopic, microscopic, and mesoscopic, each of which offers a distinct perspective on pedestrian behavior.

Macroscopic models describe pedestrian behavior for groups of individuals. These models typically aggregate pedestrians into a type of fluid and describe the dynamics of that fluid (Cristiani et al., 2014, p. 13). Microscopic models simulate the movement of individual pedestrians, treating each person as a different particle (Cristiani et al., 2014, p. 12). Mesoscopic models sit somewhere between the macro and micro approaches. "Mesoscopic" means the scale between microscopic and macroscopic. That said it may be helpful to consider these models as statistical models. These models treat pedestrians as probability distributions and they hold the potential of helping to explain how these probability distributions overlap. In sum, when discussing pedestrian modeling, one can think of the different scales as follows: macroscopic is the *fluid* scale; microscopic is the *particle* scale; and mesoscopic is the *statistical* scale.

When discussing pedestrian models at various scales, it is important to understand a few key differences between types of models. Before going into detail about models at individual scales, this paper will discuss these categorical differences.

6.1.1 Continuous vs. Discrete

The first key difference between different types of pedestrian models is to find out if a given model is either continuous or discrete. Continuous models describe pedestrians using continuous mathematical functions. All possible points in a given dimension are possible outcomes for the model. Discrete models group outcomes in a given dimension into a finite number of increments.

Models can be continuous or discrete in both the spatial and time dimensions. For example, a given model might be continuous-temporally but discrete-spatially. Such a model could predict pedestrian behavior at any time but might categorize space into a number of features. In this hypothetical model, a user could predict how many people are on a footpath or in a plaza at any time but are unable to find exactly where pedestrians are in the plaza or on the footpath. Conversely, a model could be continuous-spatially but discrete-temporally. This hypothetical model might be able to predict the locations of pedestrians but is limited to a specific time interval throughout the day.

Continuous

For a model to be continuous in a given dimension, the model must allow for all outputs in that dimension. Continuous models typically yield more precise results but are harder to create and calibrate effectively. Such models are hard to use in real time as they often require significant computational power. This is especially true when many pedestrians or groups of pedestrians (crowds) are observed and simulated. The computational power required for many of these models increases at a significantly faster rate than discrete models. Models at the macroscopic scale are almost always of this variety.

Discrete

Discrete pedestrian models are often represented as either grid-based systems or have rule-based algorithms. In particular, models can be discreet in that they output positions in a grid format or they use a number of predefined locations. For a model to be temporally discrete, it progresses at fixed time intervals, updating pedestrian positions and states in steps rather than continuously. Models can be discrete spatially in that they output positions in a grid format, or they use a number of predefined locations. For a model to be temporally discrete, it progresses at fixed time intervals, updating pedestrian positions and states at these times rather than continuously.

6.1.2 Differential vs. Non-Differential Models

Another key difference between the types of pedestrian models is whether or not the model is differentiable. Differential models describe pedestrian dynamics using systems of differential equations, whereas non-differential pedestrian models rely on rules-based approaches. Discrete models are always non-differential.

Differential

Differential pedestrian models describe pedestrian motion using systems of differential equations. Typically, these systems are rooted in physics. Most differential microscopic models treat crowd movement as analogous to fluid dynamics, using equations derived from those in that subfield of physics as their basis. Conversely, various differential microscopic models predict pedestrian movement using Newtonian-like equations (Chraibi et al., 2011).

Non-Differential

Non-differential pedestrian models rely on discrete rules-based approaches. Non-differential pedestrian models can contain differential components. For example, in an agent-based model, the path a pedestrian takes to get to a destination may be differential, but the decision-making process used to arrive at choosing that path is non-differential.

6.1.3 First-Order vs. Second-Order Models

Differential models can be either first- or second-order. First-order models account for at least pedestrian location and velocity, while second-order models additionally account for pedestrian acceleration. Each type of model has benefits and drawbacks.

First-Order

First-order pedestrian models (sometimes called velocity-based models) only account for position and velocity. These models assume that changes in pedestrian velocity occur instantaneously, which is not an assumption that is necessarily that far off. In many conditions, pedestrians are able to reach their maximum speed quite quickly, and models that account for acceleration can add needless complexity. These models are computationally efficient and good for large-scale simulations, particularly those where average speed is a good predictor of how people move at any given time. The use case for these models typically is at the urban planning scale, and they are useful for transportation studies.

Second-Order

Second-order pedestrian models (sometimes called acceleration-based models or force models) extend the first-order approach, incorporating acceleration as a governing factor for pedestrian behavior. These models can more accurately reflect real-world pedestrian behavior, accounting for inertia. On the microscopic scale, these models can predict the microscopic changes in pedestrian trajectory and between different pedestrian velocities. On the macroscopic scale, the additional acceleration factor can be used as a proxy for danger. This is particularly useful in evacuation dynamics for crowds. The acceleration term in these models can be used to predict areas where over-compression can occur.

6.1.4 Rational or Not

Another key difference in pedestrian models is whether or not they assume behavior to be rational. Rational, in this sense, means having some form of decision-making built explicitly into the model. This decision-making can pertain to crowds on a macroscopic scale or individuals on a microscopic scale.

Models that account for rational pedestrian behavior assume that individuals make deliberate goal-oriented decisions based on some internal criteria that can range from minimizing travel time to avoiding congestion to maximizing personal comfort. These models require accurate data on pedestrian preferences and their decision-making processes, which can be difficult to obtain. Additionally, these models can struggle to capture spontaneous or irrational behaviors. Rational models also struggle with subconscious biases. The most significant barrier to the use of rational models, however, is their sheer computational complexity. For pedestrian rationality to be a useful predictor of pedestrian behavior, these models can require significantly more inputs and are more cumbersome to use. That said, as computational power has dramatically increased in recent decades, rational pedestrian models are likely to prove more and more useful.

On the other hand, models that do not account for rationality on the part of pedestrians are not driven by decision-making but rather by local heuristics. These models rely on a predefined set of rules. While they do not consider pedestrians to be rational actors, that does not mean that they do not account for emergent rational phenomena (Bailo et al., 2018, p. 2; Taherifar et al., 2019). Large macroscopic models often result in emergent behaviors, such as self-organization and lane formation. Without the need for the computationally expensive process of decision-making, these models do struggle with strategic choices such as pedestrian behavior in an unfamiliar environment or pedestrian rerouting as conditions change. Models with non-rational pedestrians are most useful at large scales over long timesteps where pedestrian decision-making is limited.

Contrary to what might be expected, the majority of models in pedestrian dynamics are not considered to have rational pedestrians. There are a few reasons for this approach. Models for rational pedestrians are typically significantly more complex than models that do not have rational pedestrian behavior, and these models do not necessarily predict more accurate results. Many models exist to project pedestrian behavior in very specific contexts. In these contexts, assuming rationality is not a prerequisite for effective modeling because decision-making processes are already assumed to have taken place. For example, models for evacuating pedestrians do not become appreciably more robust when they account for pedestrians deciding whether or not the danger is worthy of evacuation. These models are only intended to be used when that statement is true. Accounting for that decision-making process within the model would make it significantly more complex and prone to error. Additionally, these models account for rationality as an emergent behavior within the model itself (Helbing & Molnár, 1995, p. 4282; Moussaïd et al., 2011, p. 1). Models on the mesoscopic scale account for pedestrian decision-making as a random distribution, where decision-making is represented as a set of probabilities. This probabilistic approach speaks to rationality without the need for it to be built into a given model.

6.2 Macroscopic Models

Macroscopic pedestrian models attempt to model the overall movement of groups of pedestrians (crowds). The study of these models can also fall under the category of crowd dynamics rather than pedestrian dynamics. Macroscopic models analyze how large numbers of people navigate through areas like streets, parks, transit stations, or stadiums.

Key variables in macroscopic models typically include flux (directional flow), velocity, and density.

6.2.1 Fundamental Diagram

The fundamental diagram in pedestrian modeling is a graphical representation that illustrates the relationship between pedestrian flow and density.

6.2.2 Select Models

Model Name	Year Proposed	Differentiable	Order	Defining Feature
Magnetic Force Model	1979	Differential	2nd	Uses Coulomb's law to describe pedestrian behavior
Social Force Model	1995	Differential	2nd	Pedestrians are either attracted or repelled by each other depending on their familiarity with each other.
Centrifugal Force Model	2005	Differential	2nd	This is a version of the Social Force model that considers bodily position and size. Additionally, it modifies the force unknown pedestrians have on eath other.
Maury and Venel's Model	2007	Differential	1st	Assumes that pedestrians never collide and works back from there
Cellular Automata Models	1999	Nondifferential	∅	Space is split up into cells, and pedestrians can move into unoccupied cells according to a grid of preferences.
Discrete Choice Models	2000	Nondifferential	∅	Generalizes the Cellular Automata Models to be spatially continuous.
Hoogendoorn and Bovy's Microscopic Model	2003	Nondifferential	∅	Pedestrians forecast the movement of other pedestrians in this model for rational pedestrians.

Table 5: List of Macroscopic Models

6.3 Microscopic Models

Microscopic models focus on individual behaviors and interactions of pedestrians within their immediate environment. These models typically treat pedestrians as a type of particle moving through a particle system. This approach relates to how specific people navigate their local environment, including their movements and decisions.

Microscopic models emphasize personal space preferences, walking speeds, changes in direction, and social interactions. Microscopic models are useful in understanding and predicting individual choices and can be valuable in optimizing the design of space. By considering individual behaviors, these models typically explain behaviors and produce predictions that are more granular in nature than other scales of models.

6.3.1 Select Models

Model Name	Year Proposed	Differentiable	Order	Defining Feature
Magnetic Force Model	1979	Differential	2nd	Uses Coulomb's law to describe pedestrian behavior
Social Force Model	1995	Differential	2nd	Pedestrians are either attracted or repelled by each other depending on their familiarity with each other.
Centrifugal Force Model	2005	Differential	2nd	This is a version of the Social Force model that considers bodily position and size. Additionally, it modifies the force unknown pedestrians have on eath other.
Maury and Venel's Model	2007	Differential	1st	Assumes that pedestrians never collide and works back from there
Cellular Automata Models	1999	Nondifferential	$\emptyset$	Space is split up into cells, and pedestrians can move into unoccupied cells according to a grid of preferences.
Discrete Choice Models	2000	Nondifferential	$\emptyset$	Generalizes the Cellular Automata Models to be spatially continuous.
Hoogendoorn and Bovy's Microscopic Model	2003	Nondifferential	$\emptyset$	Pedestrians forecast the movement of other pedestrians in this model for rational pedestrians.

Table 6: List of Microscopic Models

6.4 Mesoscopic Models

Mesoscopic pedestrian models (sometimes called kinetic models) represent a middle ground between microscopic and macroscopic models. These models can both capture individual pedestrian behavior while also considering dynamic interactions between groups of pedestrians. Unlike microscopic models, which simulate the movement of individual pedestrians, or macroscopic models, which aggregate pedestrians into larger crowds, mesoscopic models describe pedestrian behavior based on statistical equations that aggregate behavioral tendencies.

It is worth noting that sometimes microscopic models are called mesoscopic when authors intend to indicate that a model has applications in conditions in which macroscopic models are typically used. These models are not to be confused with mesoscopic models as statistical representations of pedestrian dynamics. Conversely, mesoscopic models are sometimes called microscopic models with stochastic elements.

6.4.1 Select Models

Model Name	Year Proposed	Defining Feature
Dogbé's Model	2012	Uses a mean-field limit approach to pedestrian dynamics.
Bellomo and Bellouquid's Model	2011	Uses the classical kinetic theory of light to model pedestrian movement.

Table 7: List of Mesoscopic Models

6.5 Other Pedestrian Models

It is worth noting that the above three scales are not the only ones at which pedestrian behavior has been modeled. Some models exist on scales beyond those that have been described above. For example, some models consider the "nano" scale where only a single pedestrian is considered, and instead, these models focus on the walking dynamics of that pedestrian. Others might produce outputs that can exist across multiple scales.

6.5.1 Select Models

Model Name	Year Proposed	Defining Feature
Arechavaleta et al.'s Model	2008	Models the behavior of a single pedestrian and how they move. It is a "nanoscopic" model.
Actor-Based Models (ABMs)	Various	Treat pedestrians or crowds as individual units with some form of decision-making process inherent to them, i.e., they have a "brain".

Table 8: List of Pedestrian Models at Other Scales

6.6 Movement Towards Generalized Models

In recent decades, there have been efforts to model pedestrian behavior on many scales (Cristiani et al., 2014). These efforts have worked to combine models that have applicability across all scales. Many of those working on this type of endeavor focus on using Actor-Based Models (ABMs).

6.7 Model Calibration

Another consideration when using pedestrian flow models is whether or not a model is calibrated via laboratory settings or in the field. These two approaches offer distinct advantages and disadvantages.

Experimentally Derived Constants

One method of calibrating a model is using experimentally derived constants. This method is the most straightforward way of model fitting. Many pedestrian models also include constants when they are first published (Helbing & Molnár, 1995; Yu et al., 2005).

Model Fitting Based on Collected Data

Another method of model calibration is fitting the model constants to collected data on site. This method of model calibration can be used to discover unexpected results in a given site (Ko et al., 2013). Fitting parameters for these models can be a computationally intensive endeavor as the process often involves non-linearities. That said, there are robust sets of tools to help perform this type of calibration (Madsen et al., 2004).

Part 3: On the Ground

Part 3 of this thesis contains Chapters 7 through 9. These Chapters pertain to the specific implementation of the technologies discussed in Part 2. Chapter 7 will go into some detail about a site that has been chosen as the test site for this technology, namely, the plaza east of McKeldin Library facing McKeldin Mall. This site is home to the statue of Testudo, University of Maryland's mascot. The sculpture is almost ritualistic in its importance to the student body. Chapter 8 will go through the specific technology used in this thesis and talk about the reasoning behind using each method of methodology. This chapter will cover technical details as well as technical hurdles that came up during the process of working on this thesis. Chapter 9 will take us back to the site and show a potential use case of how using the technologies in this thesis can aid in the process of site redesign.

Chapter 7: McKeldin Mall

Central to the University of Maryland campus is McKeldin Mall. It serves as a pivotal hub for campus life at the university. The mall is asymmetrical and rectangular. The lawn of the mall is quite large, extending from the west at McKeldin Library to the east at the main admissions building to the north and south, the mall is lined with tall willow oak trees. At its center lies the ODK fountain dedicated to the university Omicron Delta Kappa honor society. Particularly in warmer months, the mall is often abuzz with activity. Students sit on the grass studying, lounging on benches, and participating in various outdoor activities.

To the west of the mall stands McKeldin Library. Designed in the University of Maryland's typical neoclassical architecture, the library is a focal point for the university. Inside the library, students can often be found studying on its seven floors. The library houses an extensive collection of books, digital resources, and study spaces, making it integral to student life.

At the intersection between the library and the mall, there is the home of the Testudo sculpture and steps leading to the mall this site will be used as the test bed for this thesis. This site exists both as a space for transience and as a destination. Central to the University of Maryland's campus, students pass through this site periodically throughout the day in order to get to their classes. Other students use the stairs and surrounding seating as places to rest and mingle with their peers. It is a transitional space that exists between different distinct environments, indoor spaces and outdoor lawns and open spaces. In a way, it exists as a front porch for the university.

Testudo Sculpture

Figure 12: Photo of Testudo Sculpture, in front of McKeldin Library

The Testudo sculpture, situated near the entrance of McKeldin Library, is an iconic bronze statue of a diamondback terrapin, the University of Maryland mascot. Originally cast in 1933 by sculptor Aristide Cianfarani, the sculpture embodies the resilience and determination of the student body. It is positioned on a large granite pedestal with the University of Maryland *M* on it.

The sculpture is not only important for its history but also for its place in campus tradition. Students frequently rub Testudo's nose for good luck on exams, a practice that has polished the terrapin's snout noticeably compared to the rest of its body. The sculpture reflects the intersection of institutional identity and communal spirit. Its significance extends beyond aesthetics, embodying the university's history and campus superstitions. Over the many years, it has been the subject of many pranks as well as various more well-intentioned initiatives.

7.1 History

The University of Maryland, established in 1856, started off history as an agricultural college (*University System of Maryland Timeline*, n.d.) Over the years, as the university grew, so did its need for communal spaces that could be used to foster social interaction and academic engagement. Among the communal spaces that have been created is McKeldin Mall, the virtual heart of the University of Maryland's campus. The origins of the mall date back to the early 1900s when it was intended as a place for expansion on the campus landscape. However, in the 1960s, it was transformed into the open and iconic space that it is today. The mall was named after Theodore Roosevelt McKeldin, a former governor of Maryland, who was known to be a symbol of the progressive spirit of his time (*Traditions & History*, n.d.).

McKeldin Mall has been developed and redeveloped throughout the years. In 1997, the mall underwent a significant redevelopment that enhanced both its functionality and aesthetic quality. During this time, new landscaping was added, along with additional seating areas and pathways. The redesign of the mall reflected the university's commitment to various green initiatives.

Over the years, the mall has hosted a variety of activities, from student-led to university-sponsored. It has hosted concerts, festivals, and various spontaneous activities, becoming a beloved location for students, faculty members, and visitors alike.

7.2 Analysis

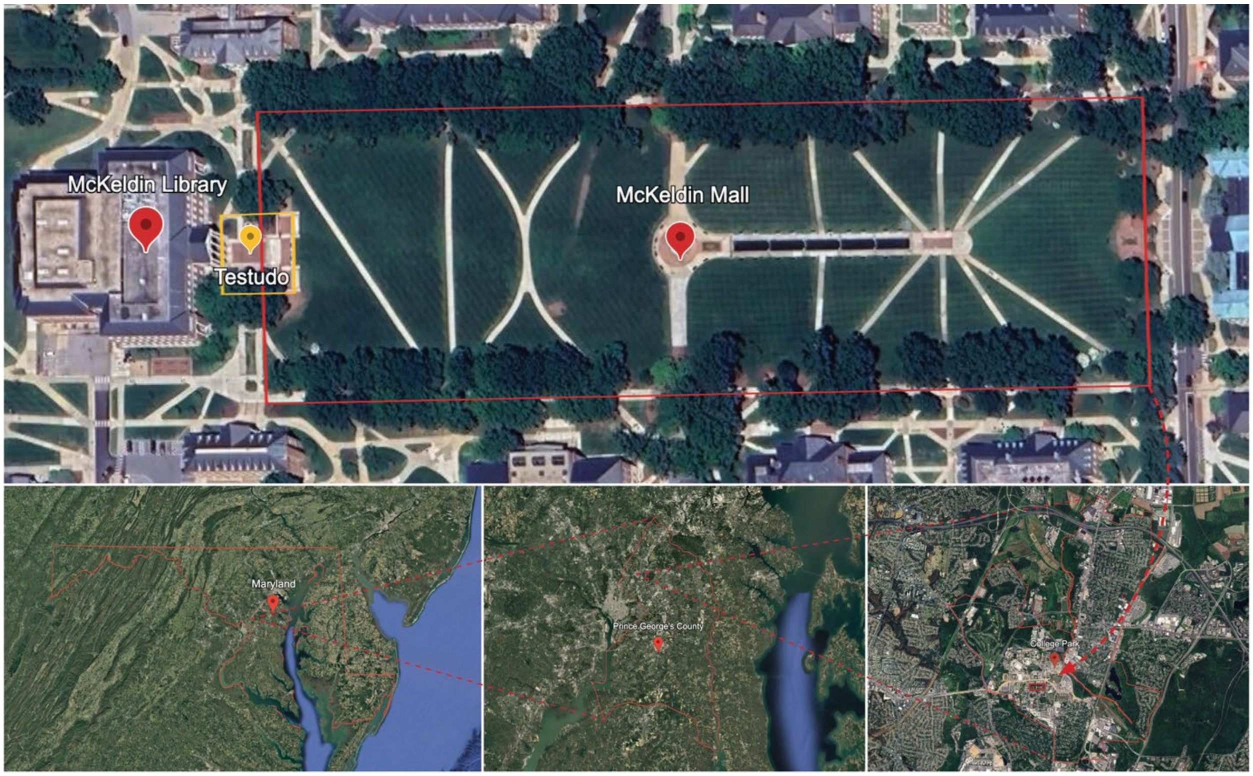


Figure 13: Site Context, pictures from Google Earth

The site exists at the virtual heart of the University of Maryland, so a number of activities take place there. Students transit to go to the library, lounge on the steps leading to McKeldin Mall, man booths for student groups or causes, pay respects to the Testudo sculpture, and generally use this site as an area of transit area as it is centrally located on campus.

Figure 14: Site Features

The site has a number of features including McKeldin Library on one side, at its center, the sculpture of Testudo, and to its west, the stairs leading to the lawn. On the north and south, there are two large planters, and on their side facing away from the site, they each have a trash and recycling bin. Surrounding the site are various planting areas.

7.2.1 Traces of Activity

Figure 15: Traces of Activity

Not only was the site observed, but traces of activity were also recorded. This was done using a general structure given by Gehl & Svarre (2013). These traces indicate the use of a site when it is not being observed. Several areas of interest were noticed, including areas of garbage collection near the entrances to the library, areas of writing around and on Testudo's base, items placed in offerings near the sculpture of Testudo on top of its base, areas of wear on the pavement and brick, and areas where the planting beds appear to be walked on.

Other methods and standards for behavioral observation exist such as those listed in Owens et al. (2023).

7.3 Recording Process

Figure 16: Recording Locations

The site was visited five times for the express purpose of recording. Recordings were taken from a balcony that overlooks the site. These recordings last between 1.5 minutes to 8.5 minutes.

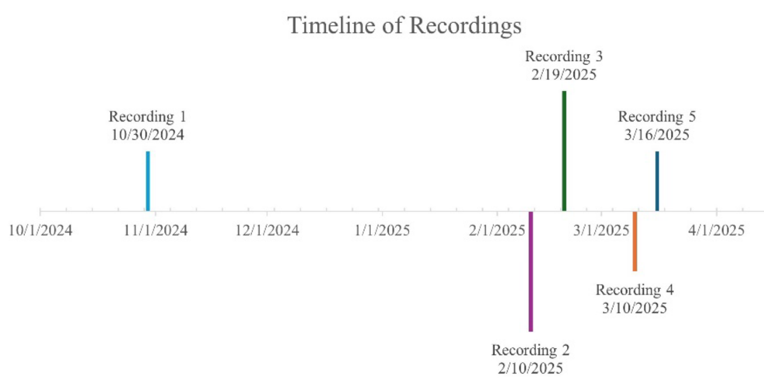
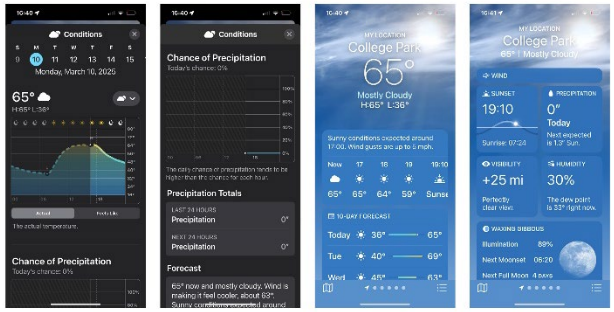


Figure 17: Timeline of Recordings

7.3.1 Weather Data



Tempurature (Hourly) | Chance of Precipitation (Hourly) | Tempurature (Weekly) | Other Weather Conditions

Figure 18: Example Weather App Screenshots

Weather on the days of the recordings (except for Recording 1) was cataloged by screenshots taken on an iPhone 13 of the Weather app.

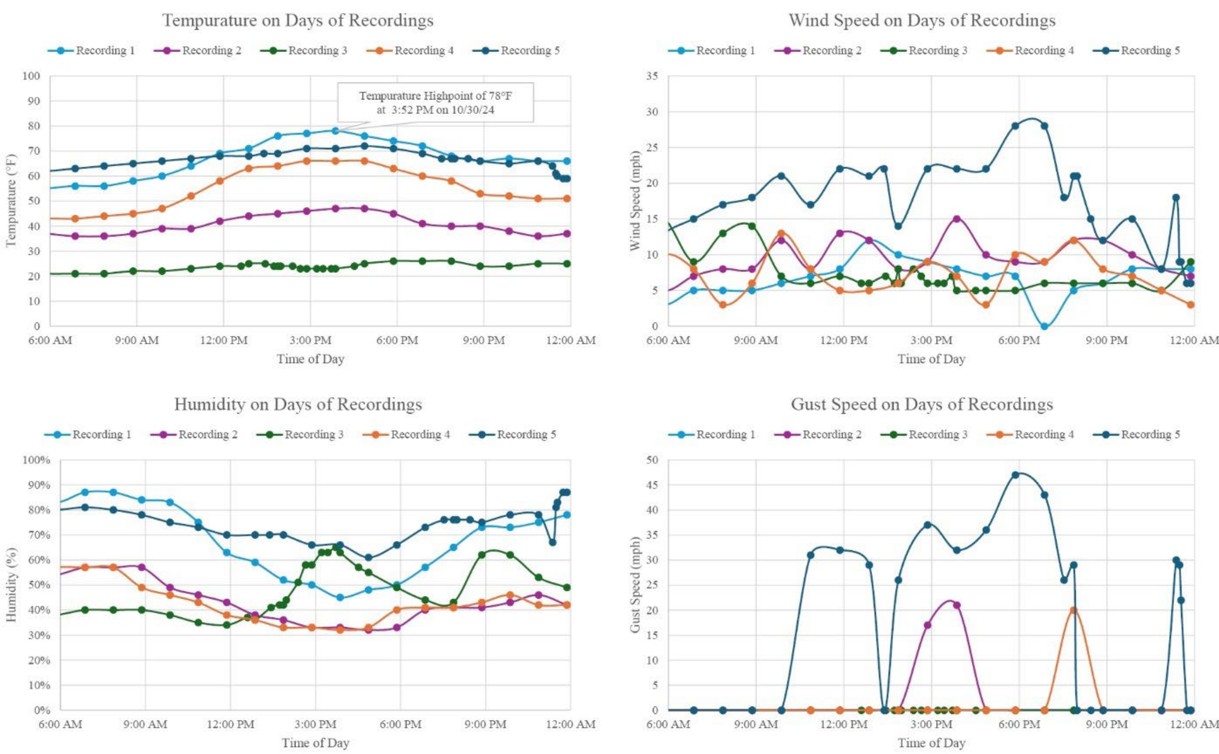


Figure 19: Weather on Recording Days, data from Weather Underground (https://www.wunderground.com/)

Weather was also recorded using data from Weather Underground. This data shows that Recording 3 was taken on the coldest day of the recordings, and Recording 1 was taken on the warmest.

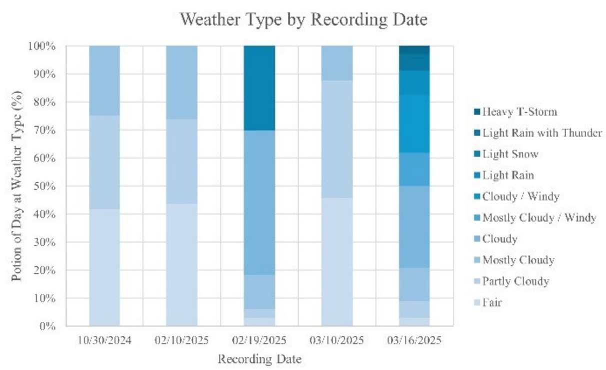


Figure 20: Weather by Type on Days of Recordings, data from Weather Underground (https://www.wunderground.com/)

7.3.2 Descriptions of Recordings

The following are descriptions of recordings taken for this thesis.

Recording 1 on 10/30/2024

Time: 4:07 PM
Length: 1:31
Temperature: 78°F
Recording Location: Recording Location 1

In this recording, students are sitting, individually and in groups, on the edge of the stairs leading to McKeldin Mall and on benches on the stairs. Additionally, two students are staffing a table to the north of Testudo. At the start of the video, a student is looking at material on the table. Approximately 30 seconds into the video, that student begins to leave. Just before the one-minute mark, a new student approaches the table. Students cross the site at various points, moving north to south and south to north. Students are also seen entering and exiting the library. At the clip's end, a student can be seen on a micro-mobility device traveling northward.

Recording 2 on 2/10/2025

Time:12:38 PM
Length: 2:02
Temperature: 44°F
Recording Location: Recording Location 1

Again, there appear to be students tabling on the site's north end. Additionally, a group is seen together near the entrance of the library. This group moves away towards the north around the 30-second mark. Several groups of two travel across the site.

Recording 3 on 2/19/2025

Time: 2:56 PM
Length: 2:02
Temperature: 23°F
Recording Location: Recording Location 1

This recording is the least densely populated. There are very few moments where students linger. Pedestrians travel through the site both in northern and southern directions and to and from the library.

Recording 4 on 3/10/2025

Time: 4:38 PM
Length: 2:02
Temperature: 66°F
Recording Location: Recording Location 1

Recording 5 on 3/16/2025

Time: 2:23 PM
Length: 8:32
Temperature: 70°F
Recording Location: Recording Location 2

Not much happens in this recording until the four-minute mark. When off in the distance, pedestrians can be seen approaching the site. Near the six-minute mark, they sit down on the northern planter. Those pedestrians and several other students travel southward. At the 6:45 minute mark, a group of pedestrians approach from the south and look at the Testudo sculpture before traveling onward.

Chapter 8: Technical Issues

This chapter covers the implementation of the technology in this thesis. In particular, it discusses how the model developed for this thesis performed on each of the challenges associated with imagery-based pedestrian movement data: detection, 3-D placement, and modeling.

8.1 Model Overview

The structure of the methodology developed for use in this thesis was based on one out of necessity. Due to time constraints (and quite frankly, the author is learning on the go), the simplest method was implemented for every step of this process.

The methods used in each of the steps from detection to placement are the following:

Detection

For detection, this thesis used YOLOv3 (Redmon & Farhadi, 2018). Specifically, the thesis used code from Swedish programmer Erik Linder-Norén's PyTorch YOLOv3 (https://github.com/eriklindernoren/PyTorch-YOLOv3?tab=readme-ov-file). This camera vision model is neither the fastest nor the most accurate model on the market. However, this version of the model was quite innovative at the time of its release and this model has been written quite a bit about it. Additionally, since it is so frequently used, it is very well troubleshooted so someone without very much experience can get it up and running.

This thesis uses the model directly as it has been published in PyTorch YOLO with one major exception: in the detection.py file, the function to export a set of images with detections on them has been amended to also export a CSV file that has information about bounding boxes, detection confidences, detection types, and source images. (See Appendix A: Computer Code under the Edits to YOLO code subsection under detection.py.)

Re-ID

Atypical for camera vision-based pedestrian tracking systems, this model doesn't use an image-based re-ID system. Again, this was a decision based on necessity. The simplest method was instead used which was an exclusively location-based re-ID system.

Several methods to do this were developed in the thesis, all of which were based on the assumption that if a person was detected twice in a row, they were close enough to the previous detection. The method that was used was based on the average person's walking speed and given a 50% margin of error. For two detections to be considered potentially from the same source, they both had to be a frame apart, from pedestrians, and no more than the time between frames times pedestrian times one plus the margin of error. This works out to the relatively simple formula:

$$|\vec{x}_i - \vec{x}_j| \leq v_p \Delta t \cdot (1 + \varepsilon)$$

Here, $\vec{x}_i$ and $\vec{x}_j$ are position coordinates in the form $\begin{bmatrix} x \\ y \end{bmatrix}$, and represent two detections in two concurrent frames, v_p is the average person's walking speed, Δt is the time between frames, and ε is a margin of error.

Additionally, this method matches pedestrian detections based on the Jonker-Volgenant algorithm, a more efficient version of the Hungarian algorithm. This method is used to ensure that the pedestrians are given unique matches. Effectively, this method is used to make sure that pedestrian detections are not endlessly lost.

Placement

Pedestrian detection 3-D placement is done in the simplest possible format. This thesis uses an inverse perspective mapping (IPM) based approach where points on the initial frame for each of the videos are selected. These points are then matched with point coordinates on the site. Then, the bottom center of a bounding box is assumed to be aware that the individual is touching the ground and directly underneath their center mass. Using a homographic transformation, the coordinates in the image's reference frame are projected onto a flat plane in the 3-D reference frame.

Geolocation

The geolocation methodology used in this thesis is, again, very simple. When performing the homographic transformation, points in the destination reference frame are given in a geospatial coordinate system. These coordinates are located in a very straightforward manner; they are acquired using visual inspection in ArcGIS Pro.

Modeling

The model used in this thesis is the magnetic force model. It is the oldest well-known computational model for microscopic pedestrian flow. A couple of considerations were taken into account when this model was selected. First, a microscopic model for pedestrian behavior was desired as this thesis desired to study pedestrian behavior on a smaller scale than crowds. Second, as this is a landscape architecture thesis, a model that accounts for parts of the environment around a model pedestrian that are not other pedestrians was required. Newer (though still dated) models for pedestrian behavior were rejected as they primarily based pedestrian behavior on other pedestrians. Additionally, models like the centrifugal force model must account for bodily orientation. Acquiring data about bodily pose was outside the scope of this thesis.

The magnetic force model is characterized by the decision to treat destinations, pedestrians, and environmental features as charged particles where pedestrians are attracted to their destinations and either attracted or repelled by environmental features. Pedestrians in a magnetic force model accelerate at a rate equal to the following equation (equation given for a given pedestrian i):

$$\vec{a}_{i_M} = k_m \sum_{l=0}^{n} \frac{k_i k_l (\vec{x}_i - \vec{x}_l)}{|\vec{x}_i - \vec{x}_l|^3}$$

Where:
k_m is a constant for the model
k_i is a constant for the pedestrian
k_l is a constant for the destination/feature
$\vec{x}_i$ is the position of the pedestrian
$\vec{x}_l$ is the position of the destination/feature
for all other destinations/features l

Additionally, pedestrians accelerate in a way that is relative to their distance from each other and dependent on their angular velocity. This acceleration vector is added to avoid collisions with each other. The equation for this acceleration for a given pedestrian i is the following:

$$\vec{a}_{i_I} = \vec{v}_i \sum_{j=0}^{o} \cos(\alpha_{ij}) \tan(\beta_{ij})$$

Where:
$\vec{v}_i$ is the velocity of pedestrian i

α_{ij} is the angle between the relative velocity of pedestrian j to pedestrian i and $\vec{v}_i$

β_{ij} is the angle between the relative velocity of pedestrian j to pedestrian i and a line from pedestrian i to pedestrian j

for all other pedestrians j

Thus, the total equation for the acceleration of a pedestrian is:

$$\vec{a}_{i_M} = k_m \sum_{l=0}^{n} \frac{k_i k_{;}(\vec{x}_i - \vec{x}_l)}{|\vec{x}_i - \vec{x}_l|^3} + \vec{v}_i \sum_{j=0}^{m} \cos(\alpha_{ij}) \tan(\beta_{ij})$$

Approximate Data

The model that was selected is a force-based model. All four spaced models act on pedestrian movement via acceleration. An issue with this was that only position data was collected. This model also takes in velocity data and produces accelerations. Since the model was to be calibrated on-site, approximate velocities and accelerations needed to be calculated. The method for doing this was by taking the derivative of a cubic spline fit to the pedestrian's detections. A result of this is that only pedestrians who are detected three times have enough information to be useful to the model. Additionally, this data is not directly measured, making the entire modeling process dependent on location data.

Calibration

When fitting the model, the least squares approach was used because it was considered to be the simplest. Other methods, specifically ones that do not consider outlying data, could have been used.

8.1.1 Assumptions

There are several major assumptions built directly into the architecture of the model, in addition to the assumptions made by the components of the model. They are as follows:

Bounding Boxes

Bounding boxes are a good approximation of the three-dimensional position of a pedestrian. They consistently capture the bounds of a single pedestrian in space.

Camera Stillness

The camera position and orientation are still or steady enough that detections do not drift based on camera movement.

Flat Plane

The site is close enough to a flat plane that modeling pedestrian movement and approximating it as such will not be a major source of error.

Fitting Acceleration

Approximating acceleration and velocity data based solely on location data is a suitable solution for both fitting the model and testing the model.

8.2 Data Pipeline

Video data is recorded on the site. That video data is converted into an image series. Those images are then fed into the modified version of YOLOv3. That modified version of YOLO returns bounding boxes and confidence intervals for those detections. Simultaneously, the model takes in ground control points and matches them to points in a selected frame from the recording. The model then generates a homography matrix, transforming the image coordinate space to the spatial coordinate space. Data from the detections is then projected into the 3-D coordinate space using the homography matrix. Specifically, the bottom center of the bounding box is projected into the space. Then, a re-ID method is implemented, giving unique IDs to each of the identified people. Velocity and acceleration data are then approximated using cubic splines. That data is then fed into the magnetic force model along with two other data sets, one of the potential destinations for pedestrians and one of the features that might affect pedestrian movement. The model then uses a method of non-linear least squares to approximate constants in the model equation. Those constants can be read into the model along with pedestrian data to return predictions.

8.3 File Structure

The file structure of this thesis is built around the problems and challenges that were listed at the beginning of this thesis. The general file structure of this thesis is as follows: the root folder is split up into three directories: `graphics`, `pedestrian_flow`, and `run`. `graphics` handles helper functions for imagery. `pedestrian_flow` handles the majority of the data processing done in this thesis. `run` is a folder for work on the GUI. `graphics` has a single subdirectory called `data`, which has input and output photo and video data. `pedestrian_flow` has the following subdirectories `csv_methods`, `data`, `interpolate`, `model`, `place`, `solve`, `track`, `utils`.

`csv_methods`—contains various single-use scripts to clean the CSV files output by this model.

`data`—contains data generated by this model.

`interpolate`—is a directory for methods of approximate and synthetic data creation. Currently, it contains a single method to approximate velocity and acceleration data based on cubic splines. It should possibly be renamed `generate`.

`model`—is a directory holding model equations. It contains a single subdirectory, `magnetic_force_model`.

`place`—involved methods of translating 2-D (image) space into 3-D (real-world) space.

`solve`—used to contain methods of fitting the model. It is no longer in use but should be recreated in a directory related to fitting models, potentially called `fit`.

`track`—contains methods re-ID methods.

`utils`—contains helper functions that are used in other files.

The `run` directory contains a couple of possible GUI structures.

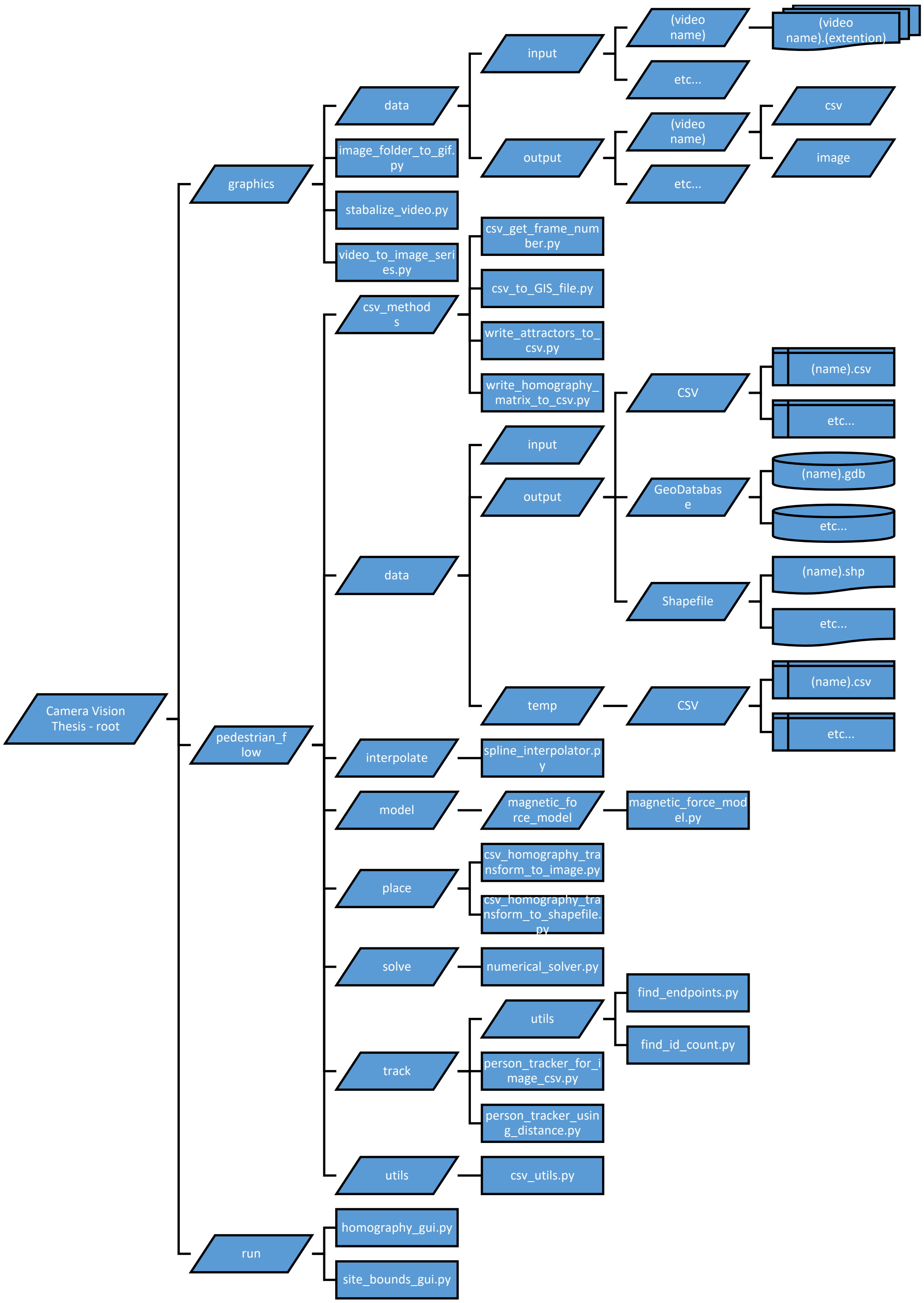


Figure 21: Thesis File Structure, visualized

Chapter 9: Redesigning the Site

Originally, this chapter was intended to be a collection of design recommendations based on a series of simulations. The Magnetic Force Model did not predict behavior well on the site and such an approach cannot be undertaken without a model that accurately describes the behavior of site users. As such, this chapter has been kept in a limited form where design recommendations are not based so much on what the model accurately predicts but also on what it fails to predict.

This chapter goes over how the site might be redesigned using the technology used in this thesis technology.

9.1 Observations from the Model

This section is dedicated to using the model to redesign the site (or, as ended up happening with this thesis, giving design recommendations for the site).

Destinations dominate movement. This may seem like a given. However, the destinations were significantly more influential no matter how the model was trained.

A major source of error was simply people lingering. People enjoy standing on the steps and finding locations to rest that are not necessarily designed as such. People linger in doorways and on stairs.

It is hard for the model to decide what to do with booths. They both exist as attractors and repellers of pedestrians. Because the model that was chosen is not a model for rational pedestrians, there was no process to simulate decision-making in pedestrians, deciding whether or not they want to interact with a booth. To some pedestrians, booths are attractors, drawing them in, and to others, they are repellers, pushing movement away from them.

Testudo also experienced some different movement behaviors around it. However, it was not of the variety that the model can easily describe. Proximity to the sculpture did not affect the pedestrian trajectory. Rather, again, pedestrians seemed to perform some decision-making process where either they lingered or moved past it without considering it.

9.2 Design Recommendations

While the model did not work out to give accurate predictions for pedestrian behavior on the site (due to the model predicting inaccurate accelerations), some of the observations above, considering the model's shortcomings, might be used for site redesigns going forward. In short, the following recommendations are based on a north star of design experience. While the model was not able to capture the pedestrian dynamics on the site, patterns of movement that exist on it were unmistakable to the educated eye.

For a thesis focused on quantifiable pedestrian movement, the central design recommendation given is quite ironic. It is to give people more space to linger and engage in behaviors that are not as easy to quantify as simple movement. This site is used as a transit hub, but it is not only used for that. It is a place for community engagement, socialization, contemplation, and interaction with public art/community tradition.

While the thesis set out to "optimize pedestrian flow," the type of intervention that this site needs is beyond the scope of what has been worked up to this point. Instead of an updated site plan like originally intended, this thesis includes a simplified vision with some basic recommendations.

Part 4: Analysis and Results

Part 4 will contain everything related to the analysis and results of this thesis. In Chapter 10, major results will be walked through and discussed in detail. Further, this chapter will handle potential areas of research that were not able to be addressed in detail in this thesis. In Chapter 11, general conclusions will be drawn, and the potential for this area of research will be discussed.

Chapter 10: Discussion

This chapter discusses the results of the thesis. This discussion is addressed in the same way that the thesis has been written. Results will be discussed, from detections to placement of people across the site to the development and implementation of the pedestrian dynamics model.

10.1 Getting Detections

Getting detections with the YOLO computer vision model had various degrees of success. Unsurprisingly, detections that were closer to the camera than those that were farther away had higher levels of confidence. What was surprising, though, was how much of a dominating factor distance was. This is possibly because the area that a pixel covers increases with respect to distance squared. At approximately 60 feet from the camera, detection success drops off dramatically. This might not be the same for a more computationally intensive object detection model. However, it stands to reason that at a certain distance, there is insufficient visual information to confidently determine if a person is or not. Higher resolution cameras might increase this range. It seems that occlusion and category of detection also have significant roles to play in determining detection confidence. However, detection type has played little to no role in detection confidence, and occlusion has played a much smaller role than originally anticipated.

Figure 22: Detection Success Graphed

Figure 22 shows detection success by location and recording. The differences in success by recording were primarily due to where people were located. Recordings 2 through 4 had a higher percentage of people on the mall than Recordings 1 and 5.

Figure 23: Detection Success Graphed 2

10.1.1 Pedestrian Tracking

After the implementation of the third iteration of the pedestrian tracking algorithm, tracking pedestrians between different detections got significantly better. Contrary to expectations, micro-mobility users were relatively easy to detect and track. This is possibly due to their routine and consistent movement across a site. Additionally, able-bodied users of these spaces tend to have a few truths about their movement characteristics. They avoid barriers as well as people on their path, and they travel towards a distinct destination and choose the shortest and/or fastest path to that destination. As a result, their trajectories avoid proximity to sources of occlusion and are relatively straight.

An unexpected difficulty in tracking movement relates to the detection of groups of people at a distance. Because of the change in granularity of visual data being inversely proportional to the distance from the camera square, grounding boxes can shift substantially, and any unaccounted-for movement of the camera dramatically affects the projected position. These factors combine, making it seem that site users at rest are, in fact, moving chaotically and rapidly in small areas. Additionally, as these detections are at a distance, they can fall below the minimum confidence threshold on certain frames. This results in those site users being repeatedly re-identified in short, uneven bursts. This caused an imbalance in model weighting for simulating pedestrian movement, resulting in extreme predicted accelerations and movement speeds along with inaccuracies in direction vector along a possible movement path.

Additionally, an assumption of this thesis was that a bounding box that was labeled as containing a person contained a single person. At times, the YOLOv3 model would return groups of people as one person.

10.2 3-D Placement

> ***Note:*** *the model created in this thesis generates latitudes and longitudes for detections. However, it still uses a 3-D placement methodology. It just assumes that the site is at a uniform elevation.*

The process of 3-D placement on the site was relatively successful. However, there exist two sources of notable bias. First, an assumption required for the use of homography to place people in 3-D space is that the site is a flat plane. The majority of the site matches that description, with notable exceptions. Beyond the paved area at the library's exit, there is a large set of stairs that represent a substantial grade change. Because this grade change exists almost in line with the camera position, pedestrians who are either on the stairs or at the bottom of the stairs are positioned several meters closer to the camera and have a higher degree of variability in terms of detection placement.

An additional problem that arose in the thesis's methodology for 3-D placement came from the assumption that the bottom center of a pedestrian's bounding box was a good approximation for the directly underneath center of mass. Even when YOLOv3 accurately placed the bounding box on a given pedestrian, the bounding box-derived pedestrian position was closer to the camera than the position of the pedestrian determined observationally.

10.3 Calibrated Model

Working with and calibrating the model was probably the most challenging and least successful aspect of this thesis, partly because the model is very old and in a relatively niche field (pedestrian dynamics). In order to implement this model in Python, this thesis referenced both an English translation of the original publication of the model and a dissertation regarding microscopic pedestrian flow. (Okazaki, 1979; Teknomo, 2016).

Analysis of the model has been split up into a couple of categories: fitting the model and comparing predictions to observations.

10.3.1 Training the Model

The fitting process was performed many times. However, since Recording 1 was the shortest recording and used the most throughout the thesis process, it was used as a benchmark. It was trained using a shortened version of the dataset, which cut out just the first 30 seconds of recording. It was trained so that all pedestrians were attracted to all destinations (called the "general magnetic force model" here). It was trained so that pedestrians were only attracted to their destination, which was assigned manually (called the “specific magnetic force model” here). Attractors and destinations can be seen in Figure 24 below.

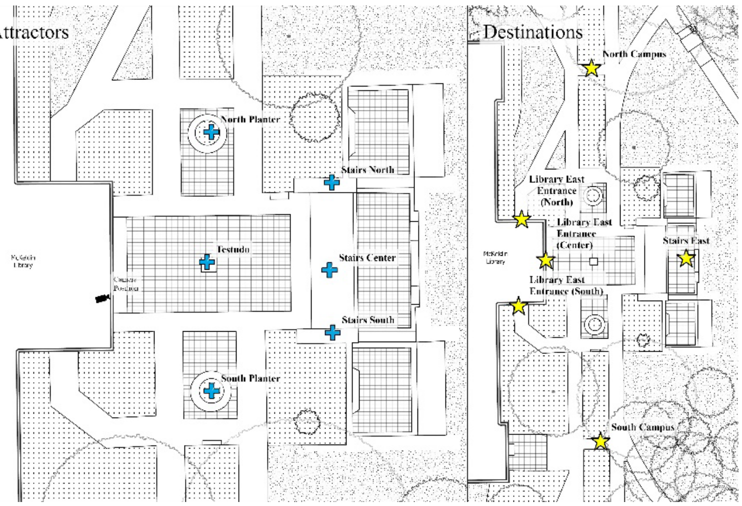


Figure 24: Model Attractors and Destinations

Unfortunately, the fitting process worked significantly better for the limited data set. In the end, the fitting process worked better for accounting for random variability than it did for fine-tuning the explanatory power of the model. The specific model fits the data better than the general model, and the model fits the smaller dataset better than either model fits the larger dataset. Additionally, fitting the model took approximately 30 minutes when all the 30-second data was used, and approximately 4 1/2 to 6 hours were needed to fit the complete data sets.

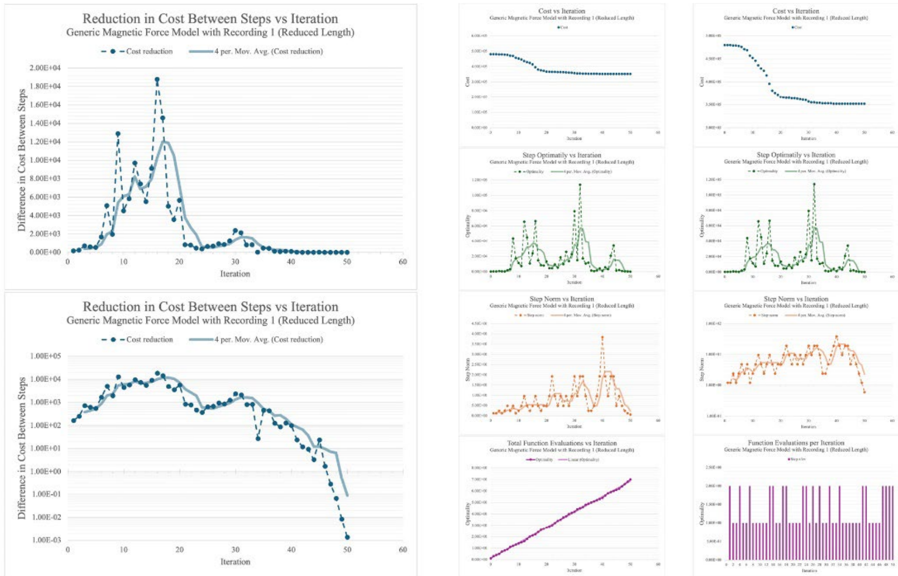


Figure 25: Fitting Generic Magnetic Force Model with Reduced Data

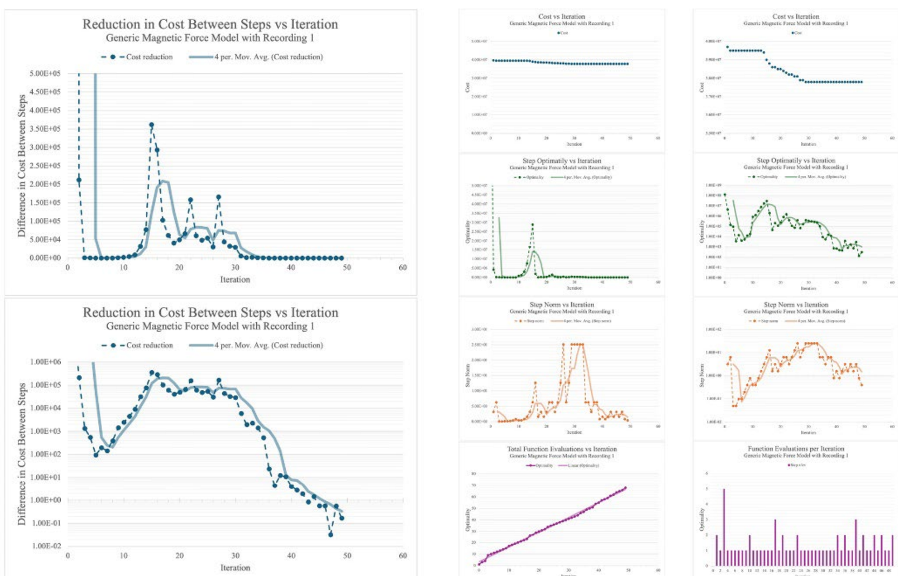


Figure 26: Fitting Generic Magnetic Force Model

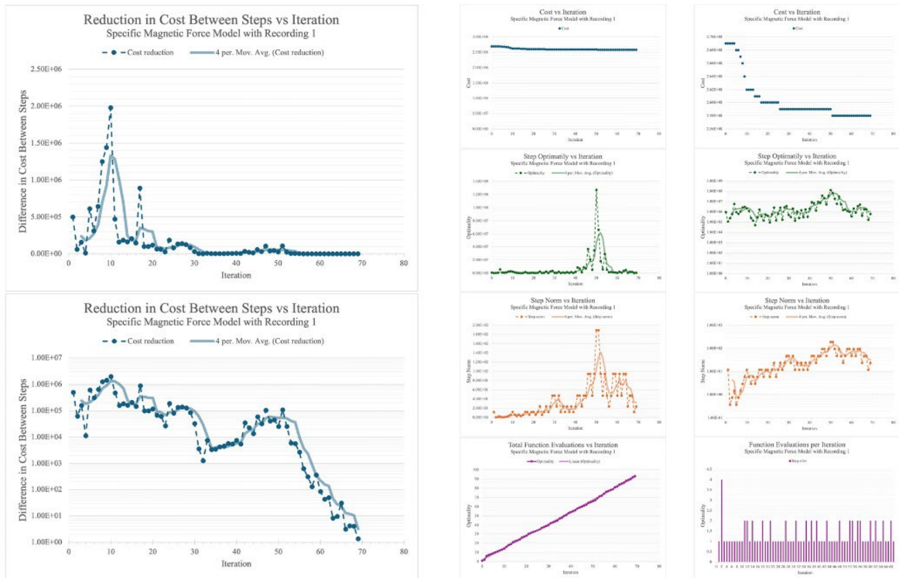


Figure 27: Fitting Specific Magnetic Force Model

When the model was fit, not only did it show that pedestrians interacted very little with the model most often, but it also showed that some pedestrians behaved in a way that was opposite to how the model predicted them to behave. The model weights are visualized in Figure 27 below.

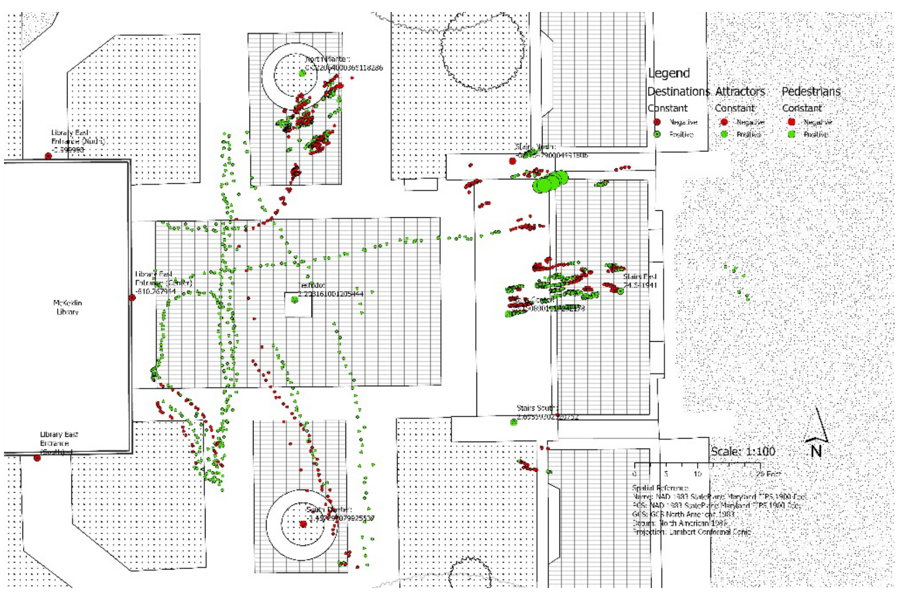


Figure 28: Model Values, visualized

10.3.2 Comparing Predictions to Observations

Predicted values for accelerations deviated significantly from the values that were fit using cubic splines. This is partially due to the model not predicting pedestrian movement very well on the site and partially because, at times, the model predicted vast acceleration because of the interaction acceleration portion of the model. The tangent function approaches ∞ at $\frac{\pi}{2}$ (a 90° angle) and ∞ at $\frac{\pi}{2}$ (270° angle). Whenever that happens, the acceleration from this element is massive and can profoundly throw off predictions.

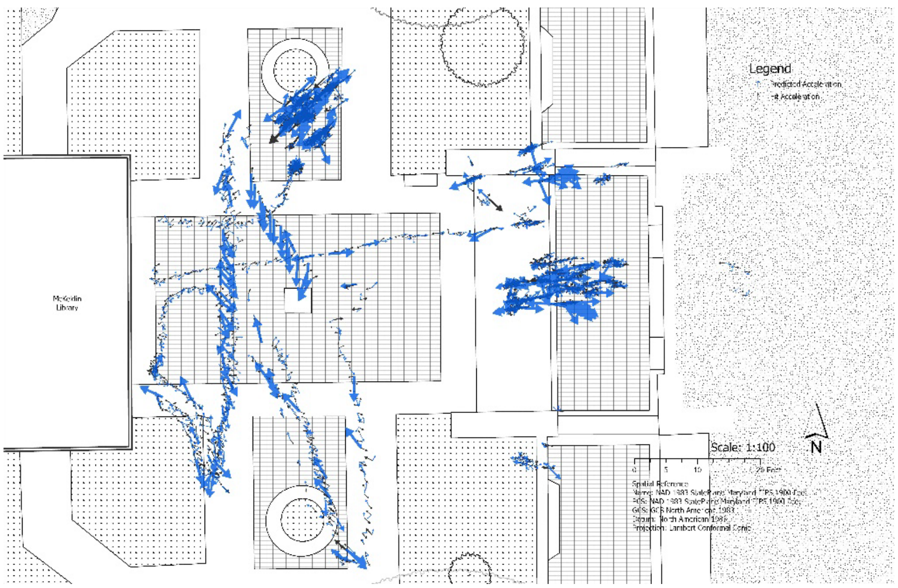


Figure 29: Model Predictions, visualized

Both the generalized and specific model turned did not predict movement well. The general model's mean absolute percentage error is 9,913% and the specific model's is 95.31%. While the specific model predicted acceleration better, neither model predicted it well.

10.4 Future Exploration

This thesis only scratches the surface of the work possible in this domain. Below are listed a number of future areas of exploration that need to be developed. These are ordered roughly from most achievable to most aspirational. Additionally, these areas of exploration are limited to the specific type of research that has been done in this thesis. These topics are not representative of future areas for the field to pursue as a whole.

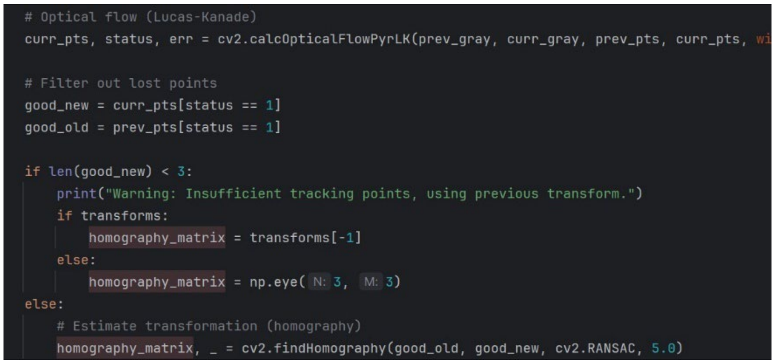
Simplified Configuration

Due to this thesis's limited timeline, each step has been brought up to a minimum working state, the result of which is that parts of the model developed earlier in the thesis process are less informed by a holistic approach to model development. Some of the libraries originally used were replaced with more elegant solutions, and some of the scripts originally developed became obsolete as the structure of data flow changed as this project progressed. The resulting model requires significant user input.

The first step in any future exploration using this model would have to be a standardization effort where the model is run using a singular configuration file and a standard set of libraries on which it is reasonable to build dependencies. Part of this standardization process must also include a consistent structure for output libraries, such that the data that the model yields is clearly interpretable. A smaller but less important part of this process is standardizing all the code comments and variable naming. This step is critical for future development of this project.

Feature Tracking for Homography

If this model continues to rely on homography for placement of detections in 3-D space, a methodology for accounting for slight movement in camera pose must be developed. There are a number of movement-tracking algorithms that could be implemented to account for the movement of source points in the source frame. A promising approach uses optical flow to track motion in a video. This approach may also bear fruit in terms of pedestrian movement tracking.



```
# Optical flow (Lucas-Kanade)
curr_pts, status, err = cv2.calcOpticalFlowPyrLK(prev_gray, curr_gray, prev_pts, curr_pts, wi

# Filter out lost points
good_new = curr_pts[status == 1]
good_old = prev_pts[status == 1]

if len(good_new) < 3:
    print("Warning: Insufficient tracking points, using previous transform.")
    if transforms:
        homography_matrix = transforms[-1]
    else:
        homography_matrix = np.eye( N: 3,  M: 3)
else:
    # Estimate transformation (homography)
    homography_matrix, _ = cv2.findHomography(good_old, good_new, cv2.RANSAC, 5.0)
```

Figure 30: Screenshot from Stabilization Script, WIP

Graphic User Interface

In order for the technology developed in the thesis to have any usability outside of people with an aptitude for computer programming, a graphical user interface must be developed. Such a graphical user interface would have to walk users through all steps of the process, notably helping users find solutions for the three problems and three subproblems listed at the start of the thesis.

Figure 31: Draft of GCP Selection GUI

More 3-D Placement Methods

This thesis used the most simple way of placing detections in a 3-D space. This methodology is, unfortunately, rife with errors and systemic biases. As a result, more robust methods of detection placement in 3-D space should be a high priority when it comes to increasing the technical capacity of the model developed here. Both multi-camera and augmented reality-based approaches should be considered and implemented.

More Computer Vision Models

Additional computer vision models are an essential addition to the framework built in this thesis. The detection model used, YOLOv3, served its role. However, more robust detection frameworks should be considered. Specifically, segmentation models may yield higher accuracy for pedestrian 3-D placement, and vision transformers hold a lot of promise for predicting pedestrian attributes like gender. These approaches might be paired with machine learning models for depth perception, allowing for significantly more accurate data collection.

Robust Re-Identification System

As discussed earlier in this thesis, the most common way of performing re-ID with camera-based detection systems is using some form of imagery reidentification. This thesis does not implement such a system. Implementing such a system would be a good step in future work. Additionally, such a system could be integrated with the existing spatiotemporal system. Additional unexplored data sources could also be incorporated, potentially increasing pedestrian detection confidence.

More Pedestrian Models

This thesis used a singular simple model for pedestrian behavior. The next steps in research should include the introduction of more pedestrian models, not just ones at the microscopic scale. These models should have the capacity to work in tandem and interact with each other. Particularly, the introduction of actor-based models is critical to making the system more robust.

More Methods of Optimization

This system used a numerical approximation of least squares. If work was to continue, additional methods of optimizing pedestrian models should also be integrated.

More Data Sources

The framework built in this thesis exclusively considers position-related data. Such a framework seems limited in the overall study of pedestrian behavior. Additional data sources, like the ones listed in Chapter 4, should be integrated into this approach.

Chapter 11: Conclusion

This chapter returns to the first person, unlike most of the body of this thesis, as it deals with the author's thoughts.

Looking at this thesis as a whole, this project has come to a resting point of mixed success. Clearly, it is possible to complete all the steps involved, from pedestrian data collection to using a model derived from that data. That being said, there is a significant amount of research required and advancement in technology needed before systems like the one used in this thesis are to be consistently implemented in the field.

11.1 What Next

In my opinion, there will be a revolution in pedestrian behavior analysis. Models of pedestrian dynamics, finely calibrated to specific site needs, are coming. It is just a matter of time. It may take years for this research to bear fruit. However, when it does, it will revolutionize how we design space and, honestly, think of it altogether.

There is a theory in psychology that, I believe, is pertinent to how we should think of this technology.

> *The Sapir–Whorf hypothesis, also known as the linguistic relativity hypothesis, refers to the proposal that the particular language one speaks influences the way one thinks about reality (Lucy, 2001).*

I believe a similar theory exists with regard to technology. For anyone with even a cursory knowledge base in developmental psychology, the idea that technology is fundamentally changing our brains is almost self-evident. There is no shortage of studies showing new ways that technology is changing human cognitive and social development (Benvenuti et al., 2023; Hoehe & Thibaut, 2020; Osiurak et al., 2018; Vedechkina & Borgonovi, 2021).

I believe that there is a parallel to the linguistic relativity hypothesis with regard to technology, a technological relativity hypothesis if you will. It is my belief that technology, once ubiquitous, becomes the language that we use to communicate and, arguably, design. It is our choice as designers to define that language or be defined by it.

Appendices

The following are the appendices for Recording to Redesign: Designing a Site Using Camera Vision to Optimize Pedestrian Flow.

Appendix A: Computer Code

Python Script: graphics/video_to_image_series.py

```
import cv2
import os
from pandas import DataFrame
import re

def extract_frames(video_path, output_dir, frame_interval=0.6):
    """
    Extracts frames from a video at specified intervals and saves them as JPG
images.

    Args:
        video_path (str): Path to the input video file.
        output_dir (str): Path to the folder where frames will be saved.
        frame_interval (float): Time interval between frames in seconds.
    """

    ### Get video and perform checks.

    # Check if the video exists/can be opened.
    video_capture = cv2.VideoCapture(video_path)
    if not video_capture.isOpened():
        print(f"Error: Could not open video file: {video_path}")
        return

    # Check if video has a valid fps.
    fps = video_capture.get(cv2.CAP_PROP_FPS)
    if fps <= 0:
        print("Error: Invalid FPS value. Cannot extract frames.")
        video_capture.release()
        return

    # Get the video filename from the video path using a regex.
    video_file_name = re.search(r"./([^/]+)\.", video_path).group(1)
    video_folder_name = re.search(r"./([^/]+)/[^/]+\.", video_path).group(1)


    ### Set up directories for outputs.

    # Create an output folders for the output photos.
    output_folder = os.path.join(output_dir, video_folder_name)
    csv_output_folder = os.path.join(output_folder, "csv")
    image_output_folder = os.path.join(output_folder, "images")
    # Create output directory.
    os.makedirs(output_folder, exist_ok=True)
    os.makedirs(csv_output_folder, exist_ok=True)
    os.makedirs(image_output_folder, exist_ok=True)


    ### Output images and csv.
```

```
    # Set up lists.
    frame_number_list = []
    seconds_from_vid_start_list = []
    image_file_name_list = []

    # Print loop starting.
    print(f"Saving frames of {video_file_name} as images in
{output_folder}:")
    # Start while loop for outputting video frames as images.
    frame_count = 0 # iterator for current frame in this while loop
    while True:
        # See if the frame count is out of bounds and read in frame.
        success, frame = video_capture.read()
        # If it is out of bounds, finish loop.
        if not success:
            print(f"Frame saving complete.\n")
            break
        # If the frame is at a multiple of our frame interval time, output
the frame as an image.
        if frame_count % int(frame_interval * fps) == 0:
            frame_filename = f"{video_file_name}_frame_{int(frame_count /
int(frame_interval * fps)):04d}_at_{(1000 *
frame_count/fps):06.0f}_milliseconds.jpg"
            frame_path = os.path.join(image_output_folder, frame_filename)
            cv2.imwrite(frame_path, frame)
            print(f"\t - Saved frame number {frame_count} in
{frame_filename}.")
            frame_number_list.append(frame_count)
            seconds_from_vid_start_list.append(frame_count/fps)
            image_file_name_list.append(frame_filename)
        frame_count += 1 # increase iterator
    video_capture.release()

    # Create dataframe and output as csv.
    d = {'Frame Number': frame_number_list, 'Seconds Since Start':
seconds_from_vid_start_list, 'Image File Name': image_file_name_list}
    df = DataFrame(data=d)
    csv_output_filename = os.path.join(csv_output_folder,
f"{video_file_name}.csv")
    print(f"Saving dataframe in {csv_output_filename}.")
    df.to_csv(csv_output_filename, index=False)
    print(f"\t - Dateframe saved.")


if __name__ == "__main__":
    video_file = "data/input/IMG_7626/IMG_7626.mp4"  # Replace with your
input video file path
    output_directory = "graphics/output" # Replace with your output file path
    frame_interval = 0.6 # Replace with your desired frame interval
    extract_frames(video_file, output_directory, frame_interval)
```

Python Script: pedestrian_flow/csv_methods/csv_get_frame_number.py

```
#!/usr/bin/env python3

import numpy as np
```

```
import csv
import re

def run():

    with open('../data/output/Shapefile/detections/detections.csv', 'r') as
file:
        reader = csv.reader(file)
        header = next(reader)  # Read the header row
        detections = list(reader)

    with open('../data/temp/CSV/detections_with_frame.csv', 'w', newline='')
as file:
        writer = csv.writer(file)
        writer.writerow(header + list(["Frame Number"]))
        for row in detections:
            temp_str = row[7]
            temp_str = temp_str.removesuffix(')')
            temp_str = temp_str.removeprefix('ezgif-frame-')
            if temp_str[0] == 'a':
                set_number = 0
            elif temp_str[0] == 'b':
                set_number = 1
            elif temp_str[0] == 'c':
                set_number = 2
            elif temp_str[0] == 'd':
                set_number = 3
            else:
                set_number = 4
            temp_str = re.sub(". .", "", temp_str)
            temp_str = str((150 * set_number) + int(temp_str))
            writer.writerow(row + list([temp_str]))


if __name__ == '__main__' :
    run()
```

Python Script: pedestrian_flow/csv_methods/csv_to_GIS_file.py

```
import geopandas as gpd
import pandas as pd
import os

def run(csv_filename, output_filename, output_file_type):

    # Create output directory.
    os.makedirs("../data/temp/CSV", exist_ok=True)
    os.makedirs("../../output", exist_ok=True)

    # Create an output folders for the output photos.
    csv_file_path = os.path.join("../data/temp/CSV", csv_filename)
    output_file_path = os.path.join("../../output", output_filename)

    if output_file_type == 'gdb':
        get_geodatabase_from_csv(csv_file_path, output_file_path)
    elif output_file_type == 'shp':
```

```
        get_shapefile_from_csv(csv_file_path, output_file_path)
    else:
        print('Invalid output file type.')
        return


def get_geodatabase_from_csv(csv_filename, output_file_path):
    # Read the CSV file into a pandas DataFrame
    df = pd.read_csv(csv_filename)

    # Assuming your CSV has 'latitude' and 'longitude' columns, create a
GeoDataFrame from the DataFrame.
    gdf = gpd.GeoDataFrame(df, geometry=gpd.points_from_xy(df.longitude,
df.latitude))

    # Write the GeoDataFrame to a GeoDatabase
    output_file_path += ".gdb"
    a = 1
    gdf.to_file(output_file_path, driver="OpenFileGDB")

    return gdf


def get_shapefile_from_csv(csv_filename, output_file_path):
    # Read the CSV file into a pandas DataFrame
    df = pd.read_csv(csv_filename)

    # Assuming your CSV has 'latitude' and 'longitude' columns, create a
GeoDataFrame from the DataFrame.
    gdf = gpd.GeoDataFrame(df, geometry=gpd.points_from_xy(df.longitude,
df.latitude))

    # Write the GeoDataFrame to a Shapefile
    output_file_path += ".shp"
    gdf.to_file(output_file_path)

    return gdf


if __name__ == '__main__' :

    csv_file =
'../data/temp/CSV/detections_for_shapefile_with_people_labelled.csv'
    output_file = 'detections_3'
    output_type = 'gdb'
    run(csv_file,output_file, output_type)
```

Python Script: pedestrian_flow/csv_methods/write_attractors_to_csv.py

```
import pandas as pd

# Create a sample DataFrame
data = {
    'longitude': [-8565429.44, -8565429.97, -8565429.49, -8565416.66, -
8565409.67, -8565416.32, -8565416.39, -8565440.10, -8565446.01, -8565445.32,
-8565428.26, -8565430.16],
```

```
    'latitude': [4719648.88, 4719662.92, 4719677.08, 4719662.01, 4719663.44, 
4719655.23, 4719671.58, 4719663.05, 4719653.02, 4719671.91, 4719623.53, 
4719704.79],
    'name': ["South Planter", "Testudo", "North Planter", "Stairs Center", 
"Stairs East", "Stairs South", "Stairs North", "Library East Entrance 
(Center)", "Library East Entrance (South)", "Library East Entrance (North)", 
"South Campus", "North Campus"]
}

df = pd.DataFrame(data)

# Export DataFrame to CSV
df.to_csv('attractors.csv', index=False)  # Set index=False to exclude the 
index column

print("CSV file has been saved successfully.")
```

Python Script: pedestrian_flow/csv_methods/write_homography_matrix_to_csv.py

```
#!/usr/bin/env python3

import cv2
import numpy as np
import csv

def run():

    # Points in the source image (at least 4)
    pts_src = np.array([[97, 653], [676, 214], [1268, 262], [1479, 824], 
[898, 472]])

    # Points in the destination image (at least 4)
    pts_dst = 
np.array([[8565435.91,4719670.33],[8565418.75,4719670.43],[8565418.64,4719655
.71],[8565435.61,4719655.37],[8565430.53,4719661.79]])
    #pts_dst = np.array([[599, 373], [1048, 369], [1009, 761], [606, 765], 
[739, 596]])

    # Calculate homography
    h, status = cv2.findHomography(pts_src, pts_dst)

    # Save the array to a CSV file
    np.savetxt('../data/input/Example/homography_matrix_coord.csv', h, 
delimiter=',')
    #np.savetxt('homography_matrix.csv', h, delimiter=',')


if __name__ == '__main__' :
    run()
```

Python Script: pedestrian_flow/interpolate/spline_interpolator.py

```
import pandas as pd
import numpy as np
from scipy.interpolate import CubicSpline
```

```
def fit_splines(input_csv, output_csv, num_points=100):
    # Read the CSV file
    df = pd.read_csv(input_csv)

    # Initialize variables
    t_column = 'Frame Number'
    x_column = 'longitude'
    y_column = 'latitude'
    spline = 'Person Number'

    # Ensure required columns exist
    required_columns = {t_column, x_column, y_column, spline}
    if not required_columns.issubset(df.columns):
        raise ValueError(f"Missing required columns: {required_columns -
set(df.columns)}")

    # Drop rows with null spline numbers
    df = df.dropna(subset=[spline])

    # Prepare output data storage
    results = []
    equations = []

    # Convert
    df[spline] = df[spline].astype(int)
    df[t_column] = df[t_column].astype(int)

    # Process each unique spline number
    for spline_num in df[spline].unique():
        subset = df[df[spline] == spline_num]
        subset = subset.sort_values(t_column)

        # Check if there are at least three points
        if subset.shape[0] < 3:
            continue

        # Set up values
        t_vals = subset[t_column].values
        x_vals = subset[x_column].values
        y_vals = subset[y_column].values

        # Ensure time values are strictly increasing
        if np.any(np.diff(t_vals) <= 0):
            continue

        # Fit cubic splines for x_column and y_column separately
        cs_x = CubicSpline(t_vals, x_vals)
        cs_y = CubicSpline(t_vals, y_vals)

        # Store spline equations
        equations.append((spline_num, cs_x.c, cs_y.c))
        print(f"\t - Spline fit for Person {spline_num}")

        # Compute velocity and acceleration
        vx = cs_x.derivative(1)(t_vals)
```

```
        vy = cs_y.derivative(1)(t_vals)
        ax = cs_x.derivative(2)(t_vals)
        ay = cs_y.derivative(2)(t_vals)

        # Store results
        for t, x, y, vx, vy, ax, ay in zip(t_vals, x_vals, y_vals, vx, vy,
ax, ay):
            results.append([spline_num, t, x, y, vx, vy, ax, ay])

    # Convert to DataFrame and save as CSV
    output_df = pd.DataFrame(results, columns=[spline, t_column, x_column,
y_column, 'vx', 'vy', 'ax', 'ay'])
    # output_df = output_df[output_df[spline] < 20]
    output_df.to_csv(output_csv, index=False)
    print(f"Interpolated spline results saved to {output_csv}")

    # Save equations to a separate file
    equations_df = pd.DataFrame(equations, columns=['spline number',
'x_coefficients', 'y_coefficients'])
    equations_df.to_csv('spline_equations.csv', index=False)
    print("Spline equations saved to spline_equations.csv")
    return

# Example usage:
if __name__ == "__main__":

fit_splines("data/csv/detections_for_shapefile_with_people_labelled.csv","out
put.csv")
# fit_splines('input.csv', 'output.csv')
```

Python Script: pedestrian_flow/place/csv_homography_transform_to_image.py

```
#!/usr/bin/env python3

import numpy as np
import csv

def run():

    h = np.loadtxt('../data/input/Example/homography_matrix.csv',
delimiter=',')

    with open('../data/temp/CSV/detections_with_frame.csv', 'r') as file:
        reader = csv.reader(file)
        header = next(reader)  # Read the header row
        detections = list(reader)

    with open('../data/temp/CSV/detections_for_image.csv', 'w', newline='')
as file:
        writer = csv.writer(file)
        writer.writerow(list(["Feet Position X (x_avg')", "Feet Position Y
(y2')"]) + header)
        for row in detections:
            x_avg = float(row[4])
            y2 = float(row[3])
```

```
            source_point = np.array([[x_avg],[y2],[1]])
            destination_point = h @ source_point
            scale_factor = destination_point[2,0]
            x_avg_prime = destination_point[0,0]/scale_factor
            y2_prime = destination_point[1,0]/scale_factor
            writer.writerow(list([x_avg_prime,y2_prime]) + row)

    # Test Code
    #im_src = cv2.imread('thesis_source_image_example.jpg')
    #im_dst = cv2.imread('thesis_destination_image_example.jpg')
    #im_out = cv2.warpPerspective(im_src, h,
(im_dst.shape[1],im_dst.shape[0]))
    #cv2.imshow("Warped Source Image", im_out)
    #cv2.waitKey(0)

if __name__ == '__main__' :
    run()
```

Python Script: pedestrian_flow/place/csv_homography_transform_to_shapefile.py

```
#!/usr/bin/env python3

import numpy as np
import csv
from pedestrian_flow.utils.csv_utils import read_csv_to_list

def run():

    h = np.loadtxt('../data/input/Example/homography_matrix_coord.csv',
delimiter=',')

    header, detections = read_csv_to_list('detections_with_frame.csv')

    with open('../data/temp/CSV/detections_for_shapefile.csv', 'w',
newline='') as file:
        writer = csv.writer(file)
        writer.writerow(list(['longitude', 'latitude']) + header)
        for row in detections:
            x_avg = float(row[4])
            y2 = float(row[3])
            source_point = np.array([[x_avg],[y2],[1]])
            destination_point = h @ source_point
            scale_factor = destination_point[2,0]
            x_avg_prime = destination_point[0,0]/scale_factor
            y2_prime = destination_point[1,0]/scale_factor
            writer.writerow(list([-x_avg_prime,y2_prime]) + row)


    # Test Code
    #im_src = cv2.imread('thesis_source_image_example.jpg')
    #im_dst = cv2.imread('thesis_destination_image_example.jpg')
    #im_out = cv2.warpPerspective(im_src, h,
(im_dst.shape[1],im_dst.shape[0]))
    #cv2.imshow("Warped Source Image", im_out)
    #cv2.waitKey(0)
```

```
if __name__ == '__main__' :
    run()
```

Python Script: pedestrian_flow/model/solve_magnetic_force_model.py

```
import numpy as np
import pandas as pd
from scipy.optimize import least_squares

# Load the datasets
pedestrian_data = pd.read_csv("output_small.csv")  # Contains p_x, p_y, v_x,
v_y, a_x, a_y, n, t
attractor_data = pd.read_csv("attractors.csv")  # Contains m_x, m_y, m_name

# Extract unique pedestrian IDs and attractor names
pedestrian_ids = pedestrian_data["Person Number"].unique()
attractor_names = attractor_data["name"].unique()

# Count the number of observations per pedestrian
pedestrian_counts = pedestrian_data["Person Number"].value_counts().to_dict()

# Initialize unknown constants
c_pedestrian = {ped_id: 1.0 for ped_id in pedestrian_ids}  # One for each
pedestrian
c_attractor = {att: 1.0 for att in attractor_names}  # One for each attractor
c_model = 1.0  # Single model constant


# Function to compute acceleration from attractors
def compute_attractor_acceleration(px, py, ped_id):
    ax_attr, ay_attr = 0.0, 0.0
    for _, attractor in attractor_data.iterrows():
        mx, my, m_name = attractor["longitude"], attractor["latitude"],
attractor["name"]

        # Compute distance and unit vector
        dx, dy = mx - px, my - py
        d = np.sqrt(dx ** 2 + dy ** 2) + 1e-6  # Small epsilon to prevent
division by zero
        ux, uy = dx / d, dy / d

        # Compute acceleration contribution
        ax_attr += c_model * c_pedestrian[ped_id] * c_attractor[m_name] * ux
/ (d ** 2)
        ay_attr += c_model * c_pedestrian[ped_id] * c_attractor[m_name] * uy
/ (d ** 2)

    return ax_attr, ay_attr


# Function to compute interaction acceleration from other pedestrians
def compute_interaction_acceleration(px, py, vx, vy, time, ped_id):
    ax_int, ay_int = 0.0, 0.0
    current_pedestrians = pedestrian_data[pedestrian_data["Frame Number"] ==
```

```
time]

    for _, other_ped in current_pedestrians.iterrows():
        if other_ped["Person Number"] == ped_id:
            continue  # Skip self

        # Relative velocity
        dvx, dvy = other_ped["vx"] - vx, other_ped["vy"] - vy
        dv_mag = np.sqrt(dvx ** 2 + dvy ** 2) + 1e-6  # Small epsilon

        # Direction between pedestrians
        dx, dy = other_ped["longitude"] - px, other_ped["latitude"] - py
        d = np.sqrt(dx ** 2 + dy ** 2) + 1e-6

        # Compute angles
        alpha = np.arccos((vx * dvx + vy * dvy) / (np.linalg.norm([vx, vy]) *
dv_mag + 1e-6))
        beta = np.arctan2(dy, dx) - np.arctan2(dvy, dvx)

        # Compute acceleration contribution
        ax_int += vx * np.cos(alpha) * np.tan(beta)
        ay_int += vy * np.cos(alpha) * np.tan(beta)

    return ax_int, ay_int


# Function to minimize in least squares
def residuals(params):
    global c_model, c_pedestrian, c_attractor

    # Unpack parameters
    c_model = params[0]
    c_pedestrian = {ped_id: params[i + 1] for i, ped_id in
enumerate(pedestrian_ids)}
    c_attractor = {att: params[len(pedestrian_ids) + i + 1] for i, att in
enumerate(attractor_names)}

    res = []
    for _, row in pedestrian_data.iterrows():
        px, py, vx, vy, ax_obs, ay_obs, ped_id, time = row["longitude"],
row["latitude"], row["vx"], row["vy"], row["ax"], row[
            "ay"], row["Person Number"], row["Frame Number"]

        # Compute predicted acceleration
        ax_pred, ay_pred = compute_attractor_acceleration(px, py, ped_id)
        ax_interact, ay_interact = compute_interaction_acceleration(px, py,
vx, vy, time, ped_id)

        ax_total = ax_pred + ax_interact
        ay_total = ay_pred + ay_interact

        # Residuals (difference between observed and predicted)
        res.append(ax_obs - ax_total)
        res.append(ay_obs - ay_total)

    return res
```

```
def residuals_weighted(params):
    global c_model, c_pedestrian, c_attractor

    # Unpack parameters
    c_model = params[0]
    c_pedestrian = {ped_id: params[i + 1] for i, ped_id in
enumerate(pedestrian_ids)}
    c_attractor = {att: params[len(pedestrian_ids) + i + 1] for i, att in
enumerate(attractor_names)}

    res = []
    for _, row in pedestrian_data.iterrows():
        px, py, vx, vy, ax_obs, ay_obs, ped_id, time = row["longitude"],
row["latitude"], row["vx"], row["vy"], row[
            "ax"], row["ay"], row["Person Number"], row["Frame Number"]

        # Compute predicted acceleration
        ax_pred, ay_pred = compute_attractor_acceleration(px, py, ped_id)
        ax_interact, ay_interact = compute_interaction_acceleration(px, py,
vx, vy, time, ped_id)

        ax_total = ax_pred + ax_interact
        ay_total = ay_pred + ay_interact

        # Weight based on the number of observations for this pedestrian
        weight = np.sqrt(pedestrian_counts[ped_id])  # Square root scaling to
avoid dominance

        # Weighted residuals (difference between observed and predicted)
        res.append(weight * (ax_obs - ax_total))
        res.append(weight * (ay_obs - ay_total))

    return res


# Function to compute total predicted acceleration
def compute_predicted_acceleration(px, py, vx, vy, time, ped_id):
    ax_attr, ay_attr = compute_attractor_acceleration(px, py, ped_id)
    ax_int, ay_int = compute_interaction_acceleration(px, py, vx, vy, time,
ped_id)

    return ax_attr + ax_int, ay_attr + ay_int

### Get parameters

# Initial guess for parameters
initial_params = [1.0] + [1.0] * len(pedestrian_ids) + [-1.0] *
len(attractor_names)

# Solve least squares problem
solution = least_squares(residuals_weighted, initial_params, verbose = 2)
# solution = least_squares(residuals_weighted, initial_params, loss =
'soft_l1', verbose = 2)

# Extract final values
c_model = solution.x[0]
```

```
c_pedestrian = {ped_id: solution.x[i + 1] for i, ped_id in
enumerate(pedestrian_ids)}
c_attractor = {att: solution.x[len(pedestrian_ids) + i + 1] for i, att in
enumerate(attractor_names)}

# Print results
print("Optimized Constants:")
print("c_model:", c_model)
print("c_pedestrian:", c_pedestrian)
print("c_attractor:", c_attractor)
```

Python Script: pedestrian_flow/track/person_tracker_for_image_csv.py

```
#!/usr/bin/env python3

from csv import writer as csv_writer
from math import sqrt
import numpy as np

from pedestrian_flow.utils import csv_utils


def person_tracker_1(detections, frame_index, x_index, y_index, threshold =
10):
    frame_number_list = [int(row[frame_index]) for row in detections]
    person_number_list = np.empty(len(frame_number_list), dtype=int)
    person_number_temp = 1
    for frame_number in range(min(frame_number_list), max(frame_number_list)
+ 1):
        if frame_number == min(frame_number_list):
            for index in [i for i, x in enumerate(frame_number_list) if x ==
frame_number]:
                person_number_list[index] = person_number_temp
                person_number_temp += 1
        else:
            for index in [i for i, x in enumerate(frame_number_list) if x ==
frame_number]:
                previous_frame_indexes = np.array([i for i, x in
enumerate(frame_number_list) if x == frame_number - 1],
                                                  dtype=int)
                previous_detections = detections[previous_frame_indexes]
                distance = lambda i: sqrt(
                    (float(detections[index][x_index]) -
float(previous_detections[i][x_index])) ** 2 + (
                            float(detections[index][y_index]) -
float(previous_detections[i][y_index])) ** 2)
                detection_under_min_distance = np.array(
                    [x for x in range(len(previous_detections)) if
distance(x) < threshold])
                if len(detection_under_min_distance) == 0:
                    person_number_list[index] = person_number_temp
                    person_number_temp += 1
                elif len(detection_under_min_distance) == 1:
                    min_index = min(range(len(previous_detections)),
key=distance)
                    person_number_list[index] =
```

```
person_number_list[previous_frame_indexes[min_index]]
                else:
                    print(f"\t+ {detection_under_min_distance} detections
below the threshold on frame number: {frame_number}")
                    print(f"\t\t- Possibly related to person numbers:")
                    for i in detection_under_min_distance:
                        print(f"\t\t\t  Person number
{person_number_list[previous_frame_indexes[i]]}")
                    min_index = min(range(len(previous_detections)),
key=distance)
                    person_number_list[index] =
person_number_list[previous_frame_indexes[min_index]]
    return person_number_list

def get_detection_coordinates_for_frame(detections, frame_number_list,
frame_number, x_index, y_index):
    frame_indexes = np.array([i for i, x in enumerate(frame_number_list) if x
== frame_number], dtype=int)
    frame_detections = detections[frame_indexes]
    detection_coordinates = np.empty((len(frame_indexes), 2))
    for index in [i for i, x in enumerate(frame_detections)]:
        row = frame_detections[index]
        detection_coordinates[index, 0] = float(row[x_index])
        detection_coordinates[index, 1] = float(row[y_index])
    print (f"Detection coordinates are: {detection_coordinates}")
    return detection_coordinates

def pair_points_under_threshold(previous_detection_coordinates,
current_detections_coordinates, threshold):
    paired_indexes = np.array([])
    distance = lambda p1, p2: sqrt((p2[0] - p1[0]) ** 2 + (p2[1] - p1[1]) **
2)
    distances = np.empty((len(previous_detection_coordinates),
len(current_detections_coordinates)), dtype=float)
    for i in range(len(previous_detection_coordinates)):
        for j in range(len(current_detections_coordinates)):
            distances[i, j] = distance(previous_detection_coordinates[i],
current_detections_coordinates[j])
    print(f"\t - Distances between detections: {distances}")
    if distances.min() < threshold:
        paired_indexes = np.argwhere(distances == distances.min())
        distances_update = np.delete(distances, (paired_indexes[0, 0]),
axis=0)
        while np.min(distances_update) < threshold:
            new_pair = np.argwhere(distances == np.min(distances_update))
            paired_indexes = np.append(paired_indexes, new_pair, axis=0)
            distances_update = np.delete(distances_update,
                                         (np.argwhere(distances_update ==
np.min(distances_update))[0, 0]), axis=0)
            if distances_update.size == 0:
                break
    return current_detections_coordinates, paired_indexes

def person_tracker_2(detections, frame_index, x_index, y_index, threshold =
10):
    frame_number_list = [int(row[frame_index]) for row in detections]
    person_number_list = np.zeros_like(frame_number_list, dtype=int)
```

```
    person_number_temp = 1

    for index in [i for i, x in enumerate(frame_number_list) if x ==
min(frame_number_list)]:
        person_number_list[index] = person_number_temp
        person_number_temp += 1

    previous_detection_coordinates =
get_detection_coordinates_for_frame(detections, frame_number_list,
min(frame_number_list), x_index, y_index)

    for frame_number in range(min(frame_number_list) + 1,
max(frame_number_list) + 1):
        temp_detections_coordinates, paired_indexes = (
            pair_points_under_threshold(previous_detection_coordinates,

get_detection_coordinates_for_frame(detections, frame_number_list,
frame_number, x_index, y_index),
                                        threshold))
        previous_detection_coordinates = temp_detections_coordinates

        previous_frame_indexes = np.array([i for i, x in
enumerate(frame_number_list) if x == frame_number-1], dtype=int)
        current_frame_indexes = np.array([i for i, x in
enumerate(frame_number_list) if x == frame_number], dtype=int)

        if paired_indexes.size != 0:
            for index in range(0, current_frame_indexes.size):
                if index in paired_indexes[:,1]:
                    temp_pair = paired_indexes[paired_indexes[:,1] == index]
                    person_number_list[current_frame_indexes[index]] =
person_number_list[previous_frame_indexes[temp_pair[0,0]]]
                else:
                    person_number_list[current_frame_indexes[index]] =
person_number_temp
                    person_number_temp += 1
        else:
            for index in range(0, current_frame_indexes.size):
                person_number_list[current_frame_indexes[index]] =
person_number_temp
                person_number_temp += 1

    return person_number_list


def run():
    header, detections =
csv_utils.read_csv_to_list('detections_for_image.csv')
    detections = np.array(detections)

    frame_index = header.index("Frame Number")
    x_index = header.index("Feet Position X (x_avg')")
    y_index = header.index("Feet Position Y (y2')")

    min_distance = 40

    person_number_list = person_tracker_2(detections, frame_index, x_index,
```

```
y_index, min_distance)
    #person_number_list = person_number_list.tolist()
    #person_number_list = [str(x) for x in person_number_list]

    with open('../data/temp/CSV/detections_with_people_labelled.csv', 'w',
newline='') as file:
        writer = csv_writer(file)
        writer.writerow(header + ["Person Number"])
        for index in range(len(detections)):
            a = detections[index]
            b = np.array([person_number_list[index]])
            c = np.concatenate((a, b), axis=0)
            writer.writerow(c)
    #print(person_number_list)

    header, detections =
csv_utils.read_csv_to_list('detections_for_shapefile.csv')
    detections = np.array(detections)

    with
open('../data/temp/CSV/detections_for_shapefile_with_people_labelled.csv',
'w', newline='') as file:
        writer = csv_writer(file)
        writer.writerow(header + ["Person Number"])
        for index in range(len(detections)):
            a = detections[index]
            b = np.array([person_number_list[index]])
            c = np.concatenate((a, b), axis=0)
            writer.writerow(c)


if __name__ == '__main__':
    run()
```

Python Script: pedestrian_flow/utils/csv_utils.py

```
#!/usr/bin/env python3

# csv imports
from csv import reader as csv_reader
from csv import writer as csv_writer
# pandas imports
from pandas import read_csv as pd_reader


def read_csv_to_list(csv_string):
    """
    Reads a csv file and returns a header and body list.

    Args:
        csv_string (str): Path to the csv files.
    Returns:
        header (list): List of headers.
        body (list): List of body lists.
    """
```

```python
    with open(csv_string, 'r') as file:
        reader = csv_reader(file)
        header = next(reader)  # Read the header row
        detections = list(reader)
    return header, detections


def write_csv_from_list(csv_string, header, detections):
    """
    Writes a csv file given a header and body list.

    Args:
        csv_string (str): Path to the csv file.
        header (list): List of headers.
        body (list): List of body lists.
    """
    with open(csv_string, 'w', newline='') as file:
        writer = csv_writer(file)
        writer.writerow(header)
        for row in detections:
            writer.writerow(row)


def read_csv_to_dataframe(csv_string):
    """
    Reads a csv file and returns a dataframe.

    Args:
        csv_string (str): Path to the csv files.
    Returns:
        dataframe: Pandas dataframe.
    """
    dataframe = pd_reader(csv_string)
    return dataframe


def write_csv_from_dataframe(csv_string, dataframe):
    """
    Writes a csv file given a dataframe.

    Args:
        csv_string (str): Path to the csv file.
        dataframe (dataframe): Pandas dataframe.
    """
    dataframe.to_csv(csv_string, index=False)
    return dataframe
```

Glossary

- **Artificial Intelligence:** "Artificial intelligence - is a field of knowledge, which studies the structure and functioning of intelligent systems based on multidimensional receptor-effector neural-like growing networks" (Yashchenko, 2014).
- **Automated Site Analysis:** Site analysis methods centering automated algorithms, processes, and workflows (Barbarash & Lu, 2020).
- **Behavioral Mapping:** "Activities, people, places for staying and much more … drawn as symbols on a plan of an area being studied to mark the number and type of activities and where they take place" (Gehl & Svarre, 2013, p. 23).
- **Bounding Box:** "A **bounding box**, or **bbox**, is simply a rectangle drawn on an image to highlight the presence of an **object of interest** at that spatial location" (Singh, 2024).
- **Computer Vision (*also referred to as* Camera Vision):** "Computer Vision is the science and technology of machines that see" (Crowley et al., 2011, p. V).
- **Crowd:** "A large group of pedestrians who have gathered together. Depending on the perspective, more specific definitions exist" (Chraibi et al., 2018, p. 1).
- **LiDAR:** "Light detection and ranging (LiDAR), also known as laser detection and ranging (LaDAR) or optical radar, is an active remote sensing technique which uses electromagnetic energy in the optical range to detect an object (target), determine the distance between the target and the instrument (range), and deduce physical properties of the object based on the interaction of the radiation with the target through phenomena such as scattering, absorption, reflection, and fluorescence" (Diaz et al., 2017, p. 930).
- **Homography:** "The planar homography is a non-singular linear relationship between points on planes… Images of points on a plane in one view are related to corresponding

image points in another view by a planar homography using a homogeneous representation. This is a projective relation since it only depends on the intersection of planes with lines. The homography transfers points from one view to the other as if they were images of points on the plane" (Agarwal et al., 2008, p. 1).

- **Machine Learning:** "Machine learning is the field of study concerned with giving computers the ability to learn without being explicitly programmed" (Nagy, 2018, p. 2).
- **Macroscopic (Model):** "…based on the assumption that the number of agents is large enough to be described by locally averaged quantities, typically density ρ and velocity v, regarded as dependent variables of time and space" (Cristiani et al., 2014, p. 80).
- **Macroscopic (Scale):** "(The focus of the macroscopic scale is on pedestrians') average distribution, which is described by means of a density in space, usually denoted by ρ, evolving in time" (Cristiani et al., 2014, p. 12).
- **Mesoscopic (Model):** "Mesoscopic (or *kinetic*) models are based on a *statistical* representation of the crowd in the space of the *microscopic* states of pedestrians by means of the so-called *(one-particle) distribution function* f" (Cristiani et al., 2014, p. 89).
- **Mesoscopic (Scale):** "The mesoscopic (or *kinetic*) scale is more properly a scale of *representation* rather than of description. Indeed, it is based on the concept of *statistical distribution* of the states of the microscopic particles, viz. pedestrians…" (Cristiani et al., 2014, p. 13).
- **Microscopic (Model):** "… based on the assumption that every single person can be tracked individually, and her trajectory can be forecast. Microscopic models can be differential, if they are based on ordinary *differential* equations, or *nondifferential*, otherwise" (Cristiani et al., 2014, p. 73).

- **Microscopic (Scale):** "The microscopic scale is the one at which the minimal entities composing a system, henceforth called particles for brevity, are visible" (Cristiani et al., 2014, p. 12).
- **Model (*also referred to as a* Behavior Model):** "A model is always a simplification of reality that focuses on certain aspects. The range of validity or applicability is an essential part of any model. Thus, a model is always an approximation that has limited applicability. Conflicting models can coexist, e.g. the wave and particle models of light" (Adrian et al., 2019, p. 6)
- **Partial Differential Equation:** "A *partial differential equation* (PDE) is an equation involving an unknown function of two or more variables and certain of its partial derivatives" (Evans, 2022, p. 1).
- **Pedestrian:** "A pedestrian is a person travelling on foot. In this article, other characterizations are used, depending on the context, e.g., agent or particle" (Cristiani et al., 2014, p. 1).
- **Pedestrian Dynamics:** The overall field of study of pedestrian behavior.
- **Pedestrian Flow:** "How many people are moving (Gehl & Svarre, 2013, p. 13)."

Bibliography

Abbas, F., Yasmin, M., Fayyaz, M., & Asim, U. (2023). ViT-PGC: Vision transformer for pedestrian gender classification on small-size dataset. *Pattern Analysis and Applications*, *26*(4), 1805–1819. https://doi.org/10.1007/s10044-023-01196-2

Adrian, J., Bode, N., Amos, M., Baratchi, M., Beermann, M., Boltes, M., Corbetta, A., Dezecache, G., Drury, J., Fu, Z., Geraerts, R., Gwynne, S., Hofinger, G., Hunt, A., Kanters, T., Kneidl, A., Konya, K., Köster, G., Küpper, M., … Wijermans, N. (2019). A Glossary for Research on Human Crowd Dynamics. *Collective Dynamics*, *4*, 1–13. https://doi.org/10.17815/CD.2019.19

Agarwal, A., Jawahar, C., & Narayanan, P. (2008). *A Survey of Planar Homography Estimation Techniques*.

Angulo, A. V., Robartes, E., Guo, X., Chen, T. D., Heydarian, A., & Smith, B. L. (2023). Demonstration of virtual reality simulation as a tool for understanding and evaluating pedestrian safety and perception at midblock crossings. *Transportation Research Interdisciplinary Perspectives*, *20*, 100844. https://doi.org/10.1016/j.trip.2023.100844

Arens, E., & Bosselmann, P. (1989). Wind, sun and temperature—Predicting the thermal comforf of people in outdoor spaces. *Building and Environment*, *24*(4), 315–320. https://doi.org/10.1016/0360-1323(89)90025-5

Bailo, R., Carrillo, J. A., & Degond, P. (2018). Pedestrian Models Based on Rational Behaviour. In L. Gibelli & N. Bellomo (Eds.), *Crowd Dynamics, Volume 1: Theory, Models, and Safety Problems* (pp. 259–292). Springer International Publishing. https://doi.org/10.1007/978-3-030-05129-7_9

Barbarash, D. (2021). Progress of an Artificial Intelligence System for Continuous Site Inventory and Assessment. *CELA 2021: 100+1*. https://par.nsf.gov/biblio/10393347-progress-artificial-intelligence-system-continuous-site-inventory-assessment

Barbarash, D., & Lu, Y.-H. (2020). Artificial intelligence systems for automated site analytics and design performance evaluation. *Landsc. Res. Rec*, *9*, 192–203.

Barbarash, D., Rasheed, M., Gupta, A., & Wang, T. (2022). Automated recording of human movement using an artificial intelligence identification and mapping system. *Journal of Digital Landscape Architecture*, 59–70.

Benvenuti, M., Wright, M., Naslund, J., & Miers, A. C. (2023). How technology use is changing adolescents' behaviors and their social, physical, and cognitive development. *Current Psychology*, *42*(19), 16466–16469. https://doi.org/10.1007/s12144-023-04254-4

Brandao, M. (2019). *Age and gender bias in pedestrian detection algorithms* (No. arXiv:1906.10490). arXiv. https://doi.org/10.48550/arXiv.1906.10490

Chen, Y., Liu, X., Gao, W., Wang, R. Y., Li, Y., & Tu, W. (2018). Emerging social media data on measuring urban park use. *Urban Forestry & Urban Greening*, *31*, 130–141. https://doi.org/10.1016/j.ufug.2018.02.005

Choudhary, P., Kumari, P., Goel, N., & Saini, M. (2022). An Audio-Seismic Fusion Framework for Human Activity Recognition in an Outdoor Environment. *IEEE Sensors Journal*, *22*(23), 22817–22827. https://doi.org/10.1109/JSEN.2022.3208271

Chraibi, M., Kemloh, U., Schadschneider, A., & Seyfried, A. (2011). Force-based models of pedestrian dynamics. *Networks and Heterogeneous Media*, *6*(3), 425–442. https://doi.org/10.3934/nhm.2011.6.425

Chraibi, M., Tordeux, A., Schadschneider, A., & Seyfried, A. (2018). Modelling of Pedestrian and Evacuation Dynamics. In *Encyclopedia of Complexity and Systems Science* (pp. 1–22). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-27737-5_705-1

Cristiani, E., Piccoli, B., & Tosin, A. (2014). *Multiscale Modeling of Pedestrian Dynamics* (A. Quarteroni, Ed.; Vol. 12). Springer.

Crowley, J. L., Draper, B. A., & Thonnat, M. (Eds.). (2011). *Computer Vision Systems: 8th International Conference, ICVS 2011, Sophia Antipolis, France, September 20-22, 2011. Proceedings* (Vol. 6962). Springer. https://doi.org/10.1007/978-3-642-23968-7

Dantcheva, A., Chen, C., & Ross, A. (2012). Can facial cosmetics affect the matching accuracy of face recognition systems? *2012 IEEE Fifth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, 391–398. https://doi.org/10.1109/BTAS.2012.6374605

de Montigny, L., Ling, R., & Zacharias, J. (2012). The Effects of Weather on Walking Rates in Nine Cities. *Environment and Behavior*, *44*(6), 821–840. https://doi.org/10.1177/0013916511409033

Deb, S., Carruth, D. W., Sween, R., Strawderman, L., & Garrison, T. M. (2017). Efficacy of virtual reality in pedestrian safety research. *Applied Ergonomics*, *65*, 449–460. https://doi.org/10.1016/j.apergo.2017.03.007

Diaz, J. C. F., Carter, W. E., Shrestha, R. L., & Glennie, C. L. (2017). LiDAR Remote Sensing. In *Handbook of Satellite Applications* (pp. 929–980). Springer, Cham. https://doi.org/10.1007/978-3-319-23386-4_44

Dong, F., Li, X., Xie, Q., Ye, R., & Cao, S. (2022). The influence of weather and temperature on pedestrian walking characteristics on the zigzag bridge. *International Journal of Biometeorology*, *66*(12), 2541–2552. https://doi.org/10.1007/s00484-022-02379-y

Ehab, A., Burnett, G., & Heath, T. (2023). Enhancing Public Engagement in Architectural Design: A Comparative Analysis of Advanced Virtual Reality Approaches in Building Information Modeling and Gamification Techniques. *Buildings*, *13*(5), Article 5. https://doi.org/10.3390/buildings13051262

Eikvil, L., Aurdal, L., & Koren, H. (2009). Classification-based vehicle detection in high-resolution satellite images. *ISPRS Journal of Photogrammetry and Remote Sensing*, *64*(1), 65–72. https://doi.org/10.1016/j.isprsjprs.2008.09.005

Elmqvist, T., Bai, X., Frantzeskaki, N., Griffith, C., Maddox, D., McPhearson, T., Parnell, S., Romero-Lankao, P., Simon, D., & Watkins, M. (2018). *Urban Planet: Knowledge towards Sustainable Cities*. Cambridge University Press.

Enzweiler, M., & Gavrila, D. M. (2009). Monocular Pedestrian Detection: Survey and Experiments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *31*(12), 2179–2195. https://doi.org/10.1109/TPAMI.2008.260

Evans, L. C. (2022). *Partial Differential Equations*. American Mathematical Society.

Fan, Z., & Loo, B. P. Y. (2021). Street life and pedestrian activities in smart cities: Opportunities and challenges for computational urban science. *Computational Urban Science*, *1*(1), 26. https://doi.org/10.1007/s43762-021-00024-9

Fernández-Arango, D., Varela-García, F.-A., & Esmorís, A. M. (2024). Methodology for Identifying Optimal Pedestrian Paths in an Urban Environment: A Case Study of a School Environment in A Coruña, Spain. *Smart Cities*, *7*(3), Article 3. https://doi.org/10.3390/smartcities7030060

Fisher, D. M., Wood, S. A., White, E. M., Blahna, D. J., Lange, S., Weinberg, A., Tomco, M., & Lia, E. (2018). Recreational use in dispersed public lands measured using social media data and on-site counts. *Journal of Environmental Management*, *222*, 465–474. https://doi.org/10.1016/j.jenvman.2018.05.045

Fossum, M., & Ryeng, E. O. (2021). The walking speed of pedestrians on various pavement surface conditions during winter. *Transportation Research Part D: Transport and Environment*, *97*, 102934. https://doi.org/10.1016/j.trd.2021.102934

Gehl, J., & Svarre, B. (2013). *How to Study Public Life* (Vol. 1–1 online resource (xiii, 179 pages)). Island Press. http://site.ebrary.com/id/10815761

Gevers, T., Gijsenij, A., Van de Weijer, J., & Geusebroek, J.-M. (2012). *Color in computer vision: Fundamentals and applications*. John Wiley & Sons.

Ghari, B., Tourani, A., Shahbahrami, A., & Gaydadjiev, G. (2024). Pedestrian detection in low-light conditions: A comprehensive survey. *Image and Vision Computing*, *148*, 105106. https://doi.org/10.1016/j.imavis.2024.105106

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). *Rich feature hierarchies for accurate object detection and semantic segmentation* (No. arXiv:1311.2524). arXiv. https://doi.org/10.48550/arXiv.1311.2524

Golej, P., Kukuliač, P., Horák, J., Orlíková, L., & Partila, P. (2024). People Detection Using Artificial Intelligence with Panchromatic Satellite Images. *Applied Sciences*, *14*(18), Article 18. https://doi.org/10.3390/app14188555

Gu, Y., & Ren, F. (2015). Energy-Efficient Indoor Localization of Smart Hand-Held Devices Using Bluetooth. *IEEE Access*, *3*, 1450–1461. https://doi.org/10.1109/ACCESS.2015.2441694

Han, C., Seshadri, P., Ding, Y., Posner, N., Koo, B. W., Agrawal, A., Lerch, A., & Guhathakurta, S. (2024). Understanding pedestrian movement using urban sensing technologies: The promise of audio-based sensors. *Urban Informatics*, *3*(1), 22. https://doi.org/10.1007/s44212-024-00053-9

Hao, Y., Pei, H., Lyu, Y., Yuan, Z., Rizzo, J.-R., Wang, Y., & Fang, Y. (2023). Understanding the Impact of Image Quality and Distance of Objects to Object Detection Performance. *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 11436–11442. https://doi.org/10.1109/IROS55552.2023.10342139

Haralick, R. M., & Shapiro, L. G. (1994). Glossary of Computer Vision Terms. In *Digital Image Processing Methods*. CRC Press.

Helbing, D. (1998). *Models for Pedestrian Behavior* (No. arXiv:cond-mat/9805089). arXiv. https://doi.org/10.48550/arXiv.cond-mat/9805089

Helbing, D., & Molnár, P. (1995). Social force model for pedestrian dynamics. *Physical Review E*, *51*(5), 4282–4286. https://doi.org/10.1103/PhysRevE.51.4282

Hoehe, M. R., & Thibaut, F. (2020). Going digital: How technology use may influence human brains and behavior. *Dialogues in Clinical Neuroscience*, *22*(2), 93–97. https://doi.org/10.31887/DCNS.2020.22.2/mhoehe

Hou, J., Chen, L., Zhang, E., Jia, H., & Long, Y. (2020). Quantifying the usage of small public spaces using deep convolutional neural network. *PLOS ONE*, *15*(10), e0239390. https://doi.org/10.1371/journal.pone.0239390

Huang, Q., Guo, X., Wang, Y., Sun, H., & Yang, L. (2024). A survey of feature matching methods. *IET Image Processing*, *18*(6), 1385–1410. https://doi.org/10.1049/ipr2.13032

Ibrahim, M. R., Haworth, J., & Cheng, T. (2021). URBAN-i: From urban scenes to mapping slums, transport modes, and pedestrians in cities using deep learning and computer vision. *Environment and Planning B*, *48*(1), 76–93. https://doi.org/10.1177/2399808319846517

Kendall, A., Grimes, M., & Cipolla, R. (2016). *PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization* (No. arXiv:1505.07427). arXiv. https://doi.org/10.48550/arXiv.1505.07427

Kinateder, M., Wirth, T. D., & Warren, W. H. (2018). Crowd Dynamics in Virtual Reality. In L. Gibelli & N. Bellomo (Eds.), *Crowd Dynamics, Volume 1: Theory, Models, and Safety Problems* (pp. 15–36). Springer International Publishing. https://doi.org/10.1007/978-3-030-05129-7_2

Ko, M., Kim, T., & Sohn, K. (2013). Calibrating a social-force-based pedestrian walking model based on maximum likelihood estimation. *Transportation*, *40*(1), 91–107. https://doi.org/10.1007/s11116-012-9411-z

Korpilo, S., Virtanen, T., & Lehvävirta, S. (2017). Smartphone GPS tracking—Inexpensive and efficient data collection on recreational movement. *Landscape and Urban Planning*, *157*, 608–617. https://doi.org/10.1016/j.landurbplan.2016.08.005

Kumar, A. C. S., Bhandarkar, S. M., & Prasad, M. (2018). DepthNet: A Recurrent Neural Network Architecture for Monocular Depth Prediction. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 396–3968. https://doi.org/10.1109/CVPRW.2018.00066

Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2022). A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. *IEEE Transactions on Neural Networks and Learning Systems*, *33*(12), 6999–7019. https://doi.org/10.1109/TNNLS.2021.3084827

Liao, G., Jin, C.-J., & Yao, X. (2024). A lightweight algorithm for pedestrian detection in overhead images. *Multimedia Systems*, *30*(3), 158. https://doi.org/10.1007/s00530-024-01365-1

López Baeza, J., Carpio-Pinedo, J., Sievert, J., Landwehr, A., Preuner, P., Borgmann, K., Avakumović, M., Weissbach, A., Bruns-Berentelg, J., & Noennig, J. R. (2021). Modeling Pedestrian Flows: Agent-Based Simulations of Pedestrian Activity for Land Use Distributions in Urban Developments. *Sustainability*, *13*(16), Article 16. https://doi.org/10.3390/su13169268

Lu, X. X. (2018). A Review of Solutions for Perspective-n-Point Problem in Camera Pose Estimation. *Journal of Physics: Conference Series*, *1087*(5), 052009. https://doi.org/10.1088/1742-6596/1087/5/052009

Lucy, J. A. (2001). Sapir–Whorf Hypothesis. In N. J. Smelser & P. B. Baltes (Eds.), *International Encyclopedia of the Social & Behavioral Sciences* (pp. 13486–13490). Pergamon. https://doi.org/10.1016/B0-08-043076-7/03042-4

Lusco, T., Binkley, D., & Ice, R. (2018). Modeling for Stakeholder Engagement. *2018 IEEE International Conference on Software Architecture Companion (ICSA-C)*, 116–119. https://doi.org/10.1109/ICSA-C.2018.00039

Ma, Y., Soatto, S., Košecká, J., & Sastry, S. S. (2004). *An Invitation to 3-D Vision* (Vol. 26). Springer. https://doi.org/10.1007/978-0-387-21779-6

Madsen, K., Nielsen, H., & Tingleff, O. (2004). *Methods for Non-Linear Least Squares Problems (2nd ed.)*. 60.

Mamidipalli, S., Sisiopiku, V. P., Schroeder, B., & Elefteriadou, L. (2015). A review of analysis techniques and data collection methods for modeling pedestrian crossing behaviors. *Journal of Multidisciplinary Engineering Science and Technology*, *2*(2).

Mancusi, G., Fabbri, M., Egidi, S., Verasani, M., Scarabelli, P., Calderara, S., & Cucchiara, R. (2022). First Steps Towards 3D Pedestrian Detection and Tracking from Single Image. In S. Sclaroff, C. Distante, M. Leo, G. M. Farinella, & F. Tombari (Eds.), *Image Analysis and Processing – ICIAP 2022* (pp. 335–346). Springer International Publishing. https://doi.org/10.1007/978-3-031-06430-2_28

McArdle, G., Demšar ,Urška, van der Spek ,Stefan, & and McLoone, S. (2014). Classifying pedestrian movement behaviour from GPS trajectories using visualization and clustering. *Annals of GIS*, *20*(2), 85–98. https://doi.org/10.1080/19475683.2014.904560

Michael, K., McNamee, A., & Michael, M. (2006). The Emerging Ethics of Humancentric GPS Tracking and Monitoring. *2006 International Conference on Mobile Business*, 34–34. https://doi.org/10.1109/ICMB.2006.43

Moreira, A., Santos, M., Wachowicz, M., & Orellana, D. (2010). The Impact of Data Quality in the Context of Pedestrian Movement Analysis. In *Lecture Notes in Geoinformation and Cartography* (pp. 61–78). https://doi.org/10.1007/978-3-642-12326-9_4

Moussaïd, M., Helbing, D., & Theraulaz, G. (2011). How simple rules determine pedestrian behavior and crowd disasters. *Proceedings of the National Academy of Sciences*, *108*(17), 6884–6888. https://doi.org/10.1073/pnas.1016507108

Nagy, Z. (2018). Artificial intelligence and machine learning fundamentals: Develop real-world applications powered by the latest AI advances. In *Artificial intelligence and machine learning fundamentals: Develop real-world applications powered by the latest AI advances* (1st edition). Packt.

Ni, R., Lu, Y., Yang, B., Yang, C., & Liu, X. (2024). A federated pedestrian trajectory prediction model with data privacy protection. *Complex & Intelligent Systems*, *10*(2), 1787–1799. https://doi.org/10.1007/s40747-023-01239-5

Ohno, M., Ukyo, R., Amano, T., Rizk, H., & Yamaguchi, H. (2023). *Privacy-preserving Pedestrian Tracking using Distributed 3D LiDARs* (No. arXiv:2303.09915). arXiv. https://doi.org/10.48550/arXiv.2303.09915

Okazaki, S. (1979). A Study of Pedestrian Movement in Architectural Space. *Journal of Architecture, Planning, and Environmental Engineering*, *No.283*.

Oliveira, M., Santos, V., & Sappa, A. D. (2015). Multimodal inverse perspective mapping. *Information Fusion*, *24*, 108–121. https://doi.org/10.1016/j.inffus.2014.09.003

Osiurak, F., Navarro, J., & Reynaud, E. (2018). How Our Cognition Shapes and Is Shaped by Technology: A Common Framework for Understanding Human Tool-Use Interactions in the Past, Present, and Future. *Frontiers in Psychology*, *9*. https://doi.org/10.3389/fpsyg.2018.00293

Owens, P., Koo, J., & Huang, Y. (2023). *Outdoor Environments for People: Considering Human Factors in Landscape Design*. https://doi.org/10.4324/9781315100036

Padilla, R., Netto, S. L., & da Silva, E. A. B. (2020). A Survey on Performance Metrics for Object-Detection Algorithms. *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, 237–242. https://doi.org/10.1109/IWSSIP48289.2020.9145130

Papadimitriou, E., Lassarre, S., & Yannis, G. (2016). Introducing human factors in pedestrian crossing behaviour models. *Transportation Research Part F: Traffic Psychology and Behaviour*, *36*, 69–82. https://doi.org/10.1016/j.trf.2015.11.003

Papadimitriou, E., Yannis, G., & Golias, J. (2009). A Critical Assessment of Pedestrian Behaviour Models. *Transportation Research Part F: Traffic Psychology and Behaviour*, *12*(3), 242–255. https://doi.org/10.1016/j.trf.2008.12.004

Redmon, J., & Farhadi, A. (2018). *YOLOv3: An Incremental Improvement* (No. arXiv:1804.02767). arXiv. https://doi.org/10.48550/arXiv.1804.02767

Ren, J., Mao, Z., Gong, M., & Zuo, S. (2023). Modified social force model considering emotional contagion for crowd evacuation simulation. *International Journal of Disaster Risk Reduction*, *96*, 103902. https://doi.org/10.1016/j.ijdrr.2023.103902

Saleh, K., Szénási, S., & Vámossy, Z. (2021). Occlusion Handling in Generic Object Detection: A Review. *2021 IEEE 19th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, 000477–000484. https://doi.org/10.1109/SAMI50585.2021.9378657

Schlickman, E., & Domlesky, A. (2019). *Field Guide to Life in Urban Plazas: A study in New York City*. SWA. https://live-swa-2019. pantheonsite. io/wpcontent/uploads/2019/08/Field ….

Schlosser, J., Chow, C. K., & Kira, Z. (2016). Fusing LIDAR and images for pedestrian detection using convolutional neural networks. *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2198–2205. https://doi.org/10.1109/ICRA.2016.7487370

Schwebel, D. C., Gaines, J., & Severson, J. (2008). Validation of virtual reality as a tool to understand and prevent child pedestrian injury. *Accident Analysis & Prevention*, *40*(4), 1394–1400. https://doi.org/10.1016/j.aap.2008.03.005

Schwemmer, C., Knight, C., Bello-Pardo, E. D., Oklobdzija, S., Schoonvelde, M., & Lockhart, J. W. (2020). Diagnosing Gender Bias in Image Recognition Systems. *Socius*, *6*, 2378023120967171. https://doi.org/10.1177/2378023120967171

Singh, R. (2024, January 17). A Quick Reference for Bounding Boxes in Object Detection. *Medium*. https://medium.com/@rajdeepsingh/a-quick-reference-for-bounding-boxes-in-object-detection-f02119ddb76b

Spurlock, S., & Souvenir, R. (2015). Pedestrian Verification for Multi-Camera Detection. In D. Cremers, I. Reid, H. Saito, & M.-H. Yang (Eds.), *Computer Vision – ACCV 2014* (pp. 322–334). Springer International Publishing. https://doi.org/10.1007/978-3-319-16865-4_21

Sun, Z., Wang, X., Zhang, Y., Song, Y., Zhao, J., Xu, J., Yan, W., & Lv, C. (2024). A comprehensive review of pedestrian re-identification based on deep learning. *Complex & Intelligent Systems*, *10*(2), 1733–1768. https://doi.org/10.1007/s40747-023-01229-7

Suraweera, N., Winter, A., Sorensen, J., Li, S., Johnson, M., Collings, I. B., Hanly, S. V., Ni, W., & Hedley, M. (2021). Passive Through-Wall Counting of People Walking Using WiFi Beamforming Reports. *IEEE Systems Journal*, *15*(4), 5476–5482. IEEE Systems Journal. https://doi.org/10.1109/JSYST.2020.3019496

Taherifar, N., Hamedmoghadam ,Homayoun, Sree ,Sushmitha, & and Saberi, M. (2019). A macroscopic approach for calibration and validation of a modified social force model for bidirectional pedestrian streams. *Transportmetrica A: Transport Science*, *15*(2), 1637–1661. https://doi.org/10.1080/23249935.2019.1636156

Teknomo, K. (2016). *Microscopic Pedestrian Flow Characteristics: Development of an Image Processing Data Collection and Simulation Model* (No. arXiv:1610.00029). arXiv. https://doi.org/10.48550/arXiv.1610.00029

Tong, Y., & Bode, N. W. F. (n.d.). The principles of pedestrian route choice. *Journal of the Royal Society Interface*, *19*(189), 20220061. https://doi.org/10.1098/rsif.2022.0061

Tool for predicting pedestrian flow expands its reach. (2021, October 8). MIT News | Massachusetts Institute of Technology. https://news.mit.edu/2021/predicting-pedestrian-flow-1008

Torralba, A., & Efros, A. A. (2011). Unbiased look at dataset bias. *CVPR 2011*, 1521–1528. https://doi.org/10.1109/CVPR.2011.5995347

Traditions & History. (n.d.). University of Maryland. Retrieved March 23, 2025, from https://umd.edu/about/history

Traunmueller, M. W., Johnson, N., Malik, A., & Kontokosta, C. E. (2018). Digital footprints: Using WiFi probe and locational data to analyze human mobility trajectories in cities. *Computers, Environment and Urban Systems*, *72*, 4–12. https://doi.org/10.1016/j.compenvurbsys.2018.07.006

University System of Maryland Timeline. (n.d.). Retrieved March 23, 2025, from https://www.usmd.edu/usm/overview/timeline/

Urban Development: Overview. (n.d.). [Text/HTML]. World Bank. Retrieved April 8, 2025, from https://www.worldbank.org/en/topic/urbandevelopment/overview

Vedechkina, M., & Borgonovi, F. (2021). A Review of Evidence on the Role of Digital Technology in Shaping Attention and Cognitive Control in Children. *Frontiers in Psychology*, *12*, 611155. https://doi.org/10.3389/fpsyg.2021.611155

Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). Deep Learning for Computer Vision: A Brief Review. *Computational Intelligence and Neuroscience*, *2018*(1), 7068349. https://doi.org/10.1155/2018/7068349

Wang, G., Lai, J., Huang, P., & Xie, X. (2018). *Spatial-Temporal Person Re-identification* (No. arXiv:1812.03282). arXiv. https://doi.org/10.48550/arXiv.1812.03282

Waqas, M., Abbas, Q., Qureshi, A., Amin, F., Díez, I. de la T., Rios, C. U., & Gongora, H. F. (2024). Advanced Line-of-Sight (LOS) model for communicating devices in modern indoor environment. *PLOS ONE*, *19*(7), e0305039. https://doi.org/10.1371/journal.pone.0305039

Whyte, W. H. (1980). *The social life of small urban spaces*. Conservation Foundation; WorldCat.

World Urbanization Prospects: The 2018 Revision. (2019). United Nations, Department of Economic and Social Affairs, Population Division. https://population.un.org/wup/assets/WUP2018-Report.pdf

Wuhrer, S., Pishchulin, L., Brunton, A., Shu, C., & Lang, J. (2014). Estimation of human body shape and posture under clothing. *Computer Vision and Image Understanding*, *127*, 31–42. https://doi.org/10.1016/j.cviu.2014.06.012

Yashchenko, V. (2014). Artificial intelligence theory (Basic concepts). *2014 Science and Information Conference*, 473–480. https://doi.org/10.1109/SAI.2014.6918230

Yatim, H. S. M., Talib, A. Z., & Haron, F. (2017). An Automated Image-Based Approach for Tracking Pedestrian Movements from Top-View Video. *Advances in Visual Informatics*, 279–289. https://doi.org/10.1007/978-3-319-70010-6_26

Ye, S., Huang, W., Liu, W., Chen, L., Wang, X., & Zhong, X. (2025). YES: You should Examine Suspect cues for low-light object detection. *Computer Vision and Image Understanding*, *251*, 104271. https://doi.org/10.1016/j.cviu.2024.104271

Yeom, S., & Cho, I.-J. (2019). Detection and Tracking of Moving Pedestrians with a Small Unmanned Aerial Vehicle. *Applied Sciences*, *9*(16), Article 16. https://doi.org/10.3390/app9163359

Yu, W. J., Chen, R., Dong, L. Y., & Dai, S. Q. (2005). Centrifugal force model for pedestrian dynamics. *Physical Review E*, *72*(2), 026112. https://doi.org/10.1103/PhysRevE.72.026112

Zhao, L., Cheong, P., Zhang, W., & Choi, W.-W. (2025). Pedestrian Identification System Based on RFID Signalling and Deep Learning. *IEEE Sensors Journal*, 1–1. IEEE Sensors Journal. https://doi.org/10.1109/JSEN.2025.3544298

Zhao, X., Wang, L., Zhang, Y., Han, X., Deveci, M., & Parmar, M. (2024). A review of convolutional neural networks in computer vision. *Artificial Intelligence Review*, *57*(4), 1–43. https://doi.org/10.1007/s10462-024-10721-6

Zheng, L., Yang, Y., & Hauptmann, A. G. (2016). *Person Re-identification: Past, Present and Future* (No. arXiv:1610.02984). arXiv. https://doi.org/10.48550/arXiv.1610.02984

Ziegeldorf, J. H., Morchon, O. G., & Wehrle, K. (2014). Privacy in the Internet of Things: Threats and challenges. *Security and Communication Networks*, *7*(12), 2728–2742. https://doi.org/10.1002/sec.795