Data-Driven Analysis and Insight of Human Motion

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Nicholas Edward Sohre

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Stephen J. Guy, Adviser

January 2021

Acknowledgements

There are many to whom I owe thanks for their support in the creation of this body of work, which represents thousands of hours of study, data collection, experimentation, and synthesis over the course of several years. First and foremost is Hannah, the angel without whom I would never have made it this far. Thank you for being a never-ending source of encouragement and support, and for doing far more than your fair share of caring for our children through the many long deadline crunches. This accomplishment is as much yours as it is mine.

To my advisor, Stephen J. Guy, go many thanks for all the long whiteboard expeditions, the video and voice call discussions far too late into the night, and the outstanding mentorship above and beyond what anyone would reasonably expect from a faculty member with so many other responsibilities. Thank you for your patience through the (at many times philosophical) debates, availability to my incessant pestering, and friendship over the last half decade.

To my fellow lab mates present and past, I thank you all for your sharpening feedback and productive discussion surrounding my work. Ioannis, thanks for all your collaboration, advice, and helpful discussion. Bilal, thanks for the perspective and advice on how to be a successful grad student. Bobby, thanks for all the math. Fenix, thanks for being a great algorithms TA and project partner. Zach, thanks for diving with me into technical subjects we both wanted to understand on a deeper level. And thank you all for making grad school a great experience.

There are also several faculty who, even if briefly, had a profound impact on my graduate school career. To Professor Gini, thank you for your encouragement and guidance early on, for taking me on as a grad student as my initial advisor, and

for encouraging me to explore the landscape of research in computer science. To Dr Lyford-Pike and Professor Helwig, thank you for your collaborations and teamwork in completing large user studies and corresponding analysis. To Professor Knights, thanks for getting me hooked on R. To Professor Yarosh, thank you for teaching me the incredibly important lesson not to "compare my insides to someone else's outsides." To Professor Elison, I owe many thanks for the interdisciplinary collaborations and for supporting me through research assistantships.

I am forever grateful to my parents Mike and Eunice for their impact on this endeavor; thank you for always believing in me and encouraging me to work hard, strive for excellence, and persevere. And to my parents-in-law Tom and Heather, for their constant love and support over the years (a special thanks to Heather for proofreading this entire manuscript!).

Finally, I would also like to thank the numerous participants for their time and efforts in helping me to complete user studies. Your contributions to the various datasets used in this work and others are crucial for advancing the science that makes the world a better place.

The work in this dissertation has been supported in part by the National Science Foundation through grants #CHS-1526693, #CNS-1544887, and #IIS-1748541.

> is expected) in a dissertation as long as citations and DOI pointers to the Versions of Record in the ACM Digital Library are included."

The following serves as a declaration of the Versions of Record for ACM published works included in this dissertation:

- Chapter 6 contains a paper published in the Motion in Games (MIG) 2020 conference proceedings, and is available at `https://doi.org/10.1145/3424636.3426911`

IEEE's policy on reuse of published materials in a dissertation are as follows:

> "You may reuse your published article in your thesis or dissertation without requesting permission, provided that you fulfill the following requirements depending on which aspects of the article you wish to reuse... [for] Text excerpts: Provide the full citation of the original published article followed by the IEEE copyright line: ©20XX IEEE. If you are reusing a substantial portion of your article and you are not the senior author, obtain the senior author's approval before reusing the text."

As well as the requirement to include the following message:

> "In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of The University of Minnesota's products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to `http://www.ieee.org/publications_standards/publications/rights/rights_link.html` to learn how to obtain a License from RightsLink. If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation."

The following serves as the full citations of the IEEE published works included in this dissertation:

- Chapter 4 contains substantial portions of a paper published in the IEEE 2017 VHCIE Workshop: Sohre, N., Mackin, C., Interrante, V., & Guy, S. J. (2017, March). Evaluating collision avoidance effects on discomfort in virtual environments. *In 2017 IEEE Virtual Humans and Crowds for Immersive Environments* (VHCIE) (pp. 1-5). IEEE.

Dedication

This dissertation is dedicated to Eli and Ada, without whom I would probably have graduated much, much sooner.

And to my darling Hannah, without whom I would not have graduated at all.

Abstract

Motion is a central element of the human experience. Artificial Intelligence (AI) and robotics technologies continue to transform society, but work is needed to enable solutions that engage with our motion-driven reality. Critical to an understanding human motion is the ability to model and accurately simulate virtual humans. To that end, my thesis provides data-driven analysis and insight for human motion. I identify two key aspects of realistic human motion simulations: being both *natural* in appearance while covering the rich *variety* of motions exhibited by humans. I describe how motion data can be leveraged to both simulate realistic motion, as well as validate simulation realism through a combination of data-driven analysis and user study approaches.

Computational methods for human motion are largely studied in the context of computer graphics and virtual character animation. Drawing from and expanding on work in this field, my work applies data-driven methods for simulating humans in several settings: that of facial motion, local crowd simulation, and global navigation. The methods and analysis in this dissertation present contributions to the fields of AI, robotics, and computer graphics in supporting my thesis that data-driven methods can be used to create and validate realistic simulations of human motion.

In the first part of my thesis, I study the simulation of realistic human smiles by conducting a large user study to connect observer reactions to computer animated faces. The result is a rich dataset providing value beyond that of this thesis to interdisciplinary research. I use the data to train a generative model with a new machine learning heuristic (PVL) that I develop, which tunes the trade-offs in creating a variety of happy smiles. I validate the realism of the PVL results with a follow up

user study.

The second part of my thesis studies the simulation of realistic human navigation. I perform a data-driven evaluation of the impact of collision avoidance on user experiences in virtual reality (VR), validating its importance for enabling the feeling of presence. I leverage motion data of shoppers to drive new insights for human navigation decisions, discovering an entropy law governing item retrieval patterns. Finally, I present a deep-learning technique (SPNets) for simulating realistic human navigation behaviors in indoor settings trained on optimal paths. The resulting agents exhibit several human-like behaviors, such as intelligent backtracking, narrowing down goal locations, and environment familiarity. I validate the realism of SPNet simulations using paths from a user study on the same navigation tasks.

Contents

List of Tables **xiii**

List of Figures **xiv**

1 Introduction **1**

1.1 Motion Data 3

1.2 Leveraging Motion Data for Human Motion Simulation and Analysis 4

1.2.1 Creating Realistic Motion 5

1.2.2 Validating Simulation Realism 9

1.3 Thesis Statement 9

1.4 Main Results 10

1.4.1 Building a Dataset to Capture the Essence of a Smile 10

1.4.2 Data-Driven Simulation of Realistic Smiles 11

1.4.3 Validating Collision Avoidance as a Realism Technique 12

1.4.4 Data-Driven Insights for Multi-Task Human Navigation Decisions 14

1.4.5 Realistic Navigation Behavior with Uncertain Goals in Building-like Environments 16

1.5 Impact & Contributions 17

1.6 Dissertation Organization 19

2 Building a Dataset to Capture the Essence of a Smile **20**

2.1 Facial Space 21
2.2 User Study Design 22
2.2.1 Using A 3D Model to Study Smiles 23
2.3 Study Results 25
2.4 Interdisciplinary Collaborations 27

3 Data-Driven Simulation of Realistic Smiles 29
3.1 Introduction 30
3.2 Background 32
3.2.1 Facial Animation 32
3.2.2 Machine Learning for Facial Analysis 33
3.2.3 PCG as Machine Learning 33
3.2.4 Diverse, High-Quality Content 34
3.3 Problem Definition 34
3.3.1 Measuring Quality & Diversity 35
3.3.2 Feature Space (F) 36
3.3.3 Classifier (C) 37
3.3.4 Semantic Classes (S) 38
3.4 Approach & Implementation 39
3.5 Maximizing Variety, Maintaining Precision 40
3.5.1 Precision Variety Learning 41
3.5.2 Proof of Monotonicity 44
3.6 Results & Analysis 46
3.6.1 Behavior of m. 46
3.6.2 Comparing Classifiers 47
3.6.3 Analysis of Faces. 48
3.7 Validation Study. 51

3.7.1 Study Design 51
3.7.2 Participant Information 52
3.7.3 Study Results 52
3.8 Conclusion 56

4 Validating Collision Avoidance as a Realism Technique 59
4.1 Introduction 60
4.2 Background 62
4.2.1 Personal Space 62
4.2.2 Interaction with Virtual Crowds 63
4.3 Experiment Design 64
4.3.1 Participant Information 66
4.4 Results 68
4.5 Conclusion 72

5 Data-Driven Insights for Multi-Task Human Navigation Decisions 75
5.1 Introduction 76
5.2 Dataset 78
5.3 Data Analysis 80
5.3.1 Decisions 80
5.3.2 Sub-Tasks 81
5.4 An Entropy Law for Inversion Likelihood 83
5.4.1 Measuring Difficulty 83
5.4.2 Independent Perceptual Error Model 85
5.4.3 Simulation Method 86
5.5 Conclusion 91
5.6 Proofs 91

6 Realistic Navigation Behavior with Uncertain Goals in Building-like Environments 96

6.1 Introduction . . . 97
6.2 Related Work . . . 98
6.2.1 Local & Global Navigation . . . 98
6.2.2 Cognitive & Behavioral Studies of Human Navigation . . . 99
6.2.3 Deep Learning . . . 100
6.3 Local-Global Planning . . . 100
6.3.1 Problem Formulation . . . 101
6.3.2 Execution Strategy . . . 102
6.4 Learned Navigation . . . 106
6.4.1 Network Structure . . . 106
6.4.2 Navigation Prediction . . . 107
6.4.3 Data Generation & Network Training . . . 109
6.5 Results & Analysis . . . 113
6.5.1 Network Analysis . . . 113
6.5.2 Behavioral Analysis . . . 115
6.6 Validation . . . 120
6.6.1 User Study . . . 120
6.6.2 Comparison to Local Heuristics . . . 126
6.6.3 Generalization to Untrained Maps . . . 129
6.7 Limitations & Future Work . . . 131
6.7.1 Map Renderings . . . 131

7 Closing Remarks 134

7.1 Summary of Contributions . . . 135
7.2 Impact of Contributions . . . 138

7.3 Limitations 139
7.4 Future Work 141
7.4.1 Next Steps 141
7.4.2 Larger Challenges 143
7.5 Conclusion 145

References **145**

List of Tables

4.1 **Follow-Up Survey Results.** Medians, inner-quartile ranges, and paired Wilcoxon signed-rank test results for follow-up survey measures. In all tests except *Intimidated*, the alternative hypothesis was that the Avoidance value would be greater than No Avoidance. For the *Intimidated* measure, the alternative hypothesis was flipped. 68

6.1 **Datasets:** the collection of maps analyzed in this work. The table indicates the size of the maps, and the number of instances of training data that would be generated from that map. The three largest maps and one smaller map were excluded from training and used to test generalization. 111

6.2 **User study and participant summary.** The task set breakdown indicates the number of available tasks in maps on which the network was trained or not trained. Task frequency refers to the average number of times a given task was performed by a user. The game experience reflects participant responses to the question "About how often do you play First-Person or other Action-based 3D video games?" 120

List of Figures

1.1 *left:* Natural motion is critical for smooth integration of AI into everyday settings (figure adapted from (Fan et al., 2018)). *right:* Natural motion is needed for virtual characters to enhance realism in digital media such as video games and movies (figure adapted from (Karamouzas et al., 2017)). 3
1.2 Variety is an integral component to reproducing realistic human smiles. 7
1.3 The top-down view of human paths (grey lines) in an indoor environment (black lines are walls). People took a variety of different routes in this environment when asked to navigate from a start to a goal (the participants had never seen the environment before). 7
1.4 A depiction of the uncanny valley, showing the relationship between increasing human-likeness of an entity and the elicited emotional responses (adopted from (Mori et al., 2012)) 8
1.5 A variety of happy smiles generated by my method for two different 3D models. 12
1.6 Experimental setup and example participant paths for both collision avoidance (*top right*) and non-collision avoidance (*bottom right*). . . . 13
1.7 The chance of choosing the farther of two items as a function of difficulty (entropy) score for the shopper data (black with grey confidence region), simulated data (blue dashed), and random choice. 15

1.8 Example paths generated with an SPNet agent are shown overlaid on user paths and the optimal path (dashed) for two navigation tasks (start is indicated by a circle, the goal is a star). 17

2.1 Computation of facial space features. Control points are shown as dots, and the vertical bisecting line (black) shows the position of the sagittal plane. Angle is computed as the angle between the diagonal (green) and horizontal (red) arrows. Extent is the length of the horizontal arrow, and dental show is the extent of the vertical arrows (blue). . . 22

2.2 A screenshot of the application used to conduct the user study. Subjects responded to stimuli by rating each in terms of smile effectiveness and emotional intent. 23

2.3 The distribution of responses along gender and age (adapted from (Helwig et al., 2017)). 26

2.4 **User Study Data. (a)** Each face's smile quality with its standard error. The data is binned into effective happiness bins. **(b)** The distribution of faces by happiness bins. 27

2.5 (Adopted from (Helwig et al., 2017)): A heat-map plotting the interaction between the facial space parameters. The three vertical bars behind each face denote the predicted score for the three response variables: effective, genuine, and pleasant (respectively). Greener colors correspond to better smiles, and redder colors correspond to worse smiles. 28

3.1 A variety of happy, smiling mouth shapes generated by our method, rendered in a high-quality real-time engine. 30

3.2 **Training Data (a)** A visual summary of the semantic classes. **(b)** Sample counts by class. 36

3.3 A graphical overview of my approach. 38
3.4 **Sample Precision.** Conceptual regions when adding a positive sample into the training set are depicted and labeled. I define sample precision as the ratio of area **B** to area **A**. The precision of the existing classifier is the ratio $|\mathbf{C} \cup \mathbf{E}|/|\mathbf{C} \cup \mathbf{E} \cup \mathbf{D} \cup \mathbf{F}|$, and the precision of the resulting classifier is $|\mathbf{B} \cup \mathbf{C} \cup \mathbf{E}|/|\mathbf{B} \cup \mathbf{C} \cup \mathbf{E} \cup \mathbf{A} \cup \mathbf{C} \cup \mathbf{F}|$ 42
3.5 Precision-Variety Trade-off curve over m for synthetic circular boundary data. 47
3.6 **Specificity Curve Comparison.** Specificity curves over m for each semantic class using different supervised learning methods. 49
3.7 **PVL Curve Comparison.** PVL curves over m for a two-class split of the face data using different supervised learning methods. 49
3.8 Example set of generated faces with Full (top row) Partial (middle row) and None (bottom row) targeted happiness levels. 50
3.9 **Validation Study Results.** *left*: The results from the first part of the study, which show that expressions generated using smaller m are perceived as more similar than those generated with larger m. *right*: The results from the second part of the study, which show that smiles generated targeting the *Full* smile class appear happier than those generated targeting the *Partial* smile class. 53
3.10 A screenshot of the first part of the validation user study. 54
3.11 A screenshot of the second part of the validation user study, featuring a different virtual character. 55

4.1 **Experimental Setup.** Participants are placed in the above virtual environment and asked to walk along the U-shaped track as a stream of virtual agents walk by. Two conditions are used: one where the agents avoid the users and one were they do not. 61

4.2 **Experimental Conditions.** A comparison of the two experimental conditions. Simulated agents either (a) avoided the participant or (b) did not react to their presence. The inset shows first person views. The user is rendered as a white cylinder inside the crowd flow. 64

4.3 **Lab Setup.** A participant being tracked as she moves through the physical lab environment reacts to virtual agents in a simulated crowd. 66

4.4 **Example Trajectories.** A comparison of the trajectories from two trials of the same user. In the case with no collision avoidance, the user hesitates, backtracks, and ultimately follows a less smooth path. 67

4.5 **Self-reported Experiences.** Participants evaluated both experimental conditions across several perceptual metrics. Stars indicate level of statistical significance: * for $p < 0.1$, ** for $p < 0.05$, *** for $p < 0.01$. 67

4.6 **Simulator Sickness Questionnaire Results.** Mean SSQ scores are shown for the questionnaires taken before, between, and after both trials. The vertical bars indicate one standard error above and below the mean. In general, participants experienced very low levels of simulator sickness across the trials. 69

4.7 **Behavioral Analysis.** Behavioral metrics were computed for the portion of the path that intersected the stream of agents. When collision avoidance was used (a) participants took shorter, more direct paths and (b) less acceleration was experienced. Both metrics indicate less hesitation in walking. 71

4.8 **Follow-Up Survey** 74

5.1 An example item sequence from the dataset, embedded in the abstracted store layout (black obstacles) and product shelf embedding (x's). 79

5.2 An example decision $\mathbf{p} = (p_1, p_2, p_3)$. Distances $d(p_i)$ (the shortest walking path) to each remaining item (cylinders) are depicted by dashed lines. 80

5.3 *left*: The likelihood of choosing the locally optimal (closest) item decreases as a function of distance to the closest item for shopper paths (grey region is the 98% confidence interval), and eventually converges to that of a random choice. *right*: The likelihood of not choosing the closer of two items in a sub-task as a function of the difference in distances (to the shopper) between them. The random choice is shown for reference (always 0.5 for two items). 82

5.4 The chance of choosing the farther of two items in a sub-task as a function of difficulty (entropy) score for the shopper data (black w/ grey 98% confidence interval) and random choice (red). 84

5.5 *left*: the difficulty plot from Figure 5.4 with overlaid simulation inversion rate for the independent distance estimation model. *right*: observed inversion rates from simulated paths overlaid on the trend from Figure 5.3 (right). 87

5.6 *left*: observed optimal choice rates from simulated paths overlaid on the same data from Figure 5.3 (left). *right*: The log frequency of observed inversion sizes (i.e. the extraneous distance travelled to the chosen item over the closest) for simulated, shopper, and random item decisions. Inset is the density for inversion sizes cropped to show the knee of the curve. 87

5.7 Simulated paths are shown (dashed arrows) alongside the shopper paths (solid arrows) through the store for several different baskets. . . 90

6.1 **Feature Representation**. An example training feature is rendered in the context of an example environment. The goal distribution is shown as 1 and 2 standard deviation rings (σ= 2m). 102

6.2 **SPNet Network Structure**: Isovist feature rays from the path history are transformed by the scene representation network and accumulated into a scene representation. The planning network predicts two potential actions it thinks are likely to lead to efficient paths given the representation, along with a relative confidence between the two choices. 106

6.3 **Prediction under Ambiguity.** The agent (circle with inset arrow) navigates to its goal (circle inset in large uncertainty region) from 2 locations in a courtyard-style map. Lines eminating from the agent terminate at predictions (larger circle → more confident prediction). Unambiguous scenarios produce greater confidence in the chosen direction. 110

6.4 **Training Loss**: Total training loss (outset) and just the predictions loss L_{pred} (inset) throughout training. 112

6.5 **Training Profile Comparisons.** A + or − in a condition label denotes presence or absence of an SPNet network feature respectively. "S" and "H" represent split predictions and path history respectively. 114

6.6 **SPNet Runtime**: Average total runtime for each step of planning in SPNet across history and map size. 50 random tasks were executed for a small (House) and large (Business Park) map to generate timing samples. Runtime cost is primarily split between executing the prediction network, scene network, and computing isovist features. . . 115

6.7 **Human-Like Behavior:** SPNet produces efficient, human-like paths both for maps on which it was trained (left), and maps not seen in training (right). 116

6.8 **Effect of Goal Uncertainty:** The right path has $\sigma = 2.5$m and the left has no goal uncertainty. The high uncertainty case produces exploratory behavior as the agent searches for the goal, while the agent on the right heads directly for the certain goal location. 117

6.9 **Effect of Behavioral Parameters.** Goal uncertainty and history size can be tuned to affect agent behavior. Large uncertainty leads to exploratory behavior, and increasing the history size leads to more optimal paths. 117

6.10 **Effect of Path History.** Two simulated paths for the same task are shown. The right path was generated by an agent with no path history (n=1), and in the left path the agent incorporates the past three visited map nodes in planning. 118

6.11 *Left:* Effect of history size on path optimally averaged across all paths with goal size σ=2. *Right:* The representation network allows for the integration of past observations that influences future decisions, resulting in more human-like path lengths on untrained maps. 119

6.12 **User Study Interface**: Snapshot from the user-study tasks. Participants were asked to navigate to a goal whose relative position was indicated by a green dot in the mini-map at the lower right. 123

6.13 **User Study Participation**: The study was implemented as a web-hosted 3D game experience, allowing users to participate in the study on their own computers via a web browser in full-screen mode. 123

6.14 **User Path Comparison**: Paths generated by SPNet agents cover the same general routes as those taken by users. 124

6.15 **Path Length Comparison**: Path lengths are shown for the optimal path, those generated by SPNet agents (σ = 6m, path history = 3, stochastic node selection enabled), and those taken by participants in the second part of the user study (for the same tasks) on untrained maps. 125

6.16 Human users take longer paths than the graph-optimal sequence. For trained maps, SPNet path lengths are close to graph-optimal, but on new maps SPNet paths are closer to the paths taken by humans. . . . 127

6.17 **Generalization under Goal Uncertainty.** Simulated paths are shown for a mid-size test map (Office). Dots and stars indicate the start and goal locations respectively. SPNet agents are able to find efficient paths even underneath mid-sized goal uncertainty (here σ=2m, n=3). 129

6.18 **Generalization to Large Maps.** *Left:* Selected SPNet paths on the Conference map (n= 3, σ = 0m). *Right:* Selected SPNet paths on the Business Park map (n= 3, σ = 0m). SPNet agents are generally able to find efficient paths, especially when given the true goal location. . 130

6.19 Training Maps. 132

6.20 Testing Maps. 133

Chapter 1

Introduction

Physical motion is a central part of human life. We are constantly perceiving, analyzing, planning, and executing motion in various forms as we interact with the complex, stochastic world around us. More than just a means for transporting ourselves and other things, we employ motion to encode meaning and communicate to others. Along with every motion task comes the added challenge of planning and performing subject to the dynamics of our uncertain environment. In order to understand, and cooperate with one another, humans are adept at perceiving and analyzing motion, particularly that of other humans. This enables a wide range of interactions that support the relationships and collaborations that form modern society.

Artificial Intelligence (AI) and robotics technologies have and continue to transform industry and enable new and exciting applications. However, there is need for new solutions to broaden the impact potential in the settings where the physical embodiment of humans is the central focus. Currently many existing robotics solutions are aimed at domains where humans are absent or not the primary focus, such as automating manufacturing, remote security and surveillance, or autonomous transportation. AI in industry has been most successfully applied to digital platforms: suggesting what to buy, which content to consume, or predicting which ads we are most likely to click. Future technologies that seek to successfully engage with our

motion-centered reality must incorporate an understanding of how we move. This involves a change in focus to accurately portraying, anticipating, and appropriately responding to human motion. Building good models of human motion is vital to achieving this goal.

This dissertation provides data-driven insights for advancing the understanding of human motion. These insights take the form of simulation methods and evaluative analysis informed by real world data to move toward a more social form of intelligence for AI and robotics applications. The chapters herein explore human motion in several different contexts, including facial animation, crowds in virtual reality (VR), multi-task navigation decisions, and long-term navigation behaviors with uncertain goals. The results provide benefits to many applications. For example, a better understanding of how facial movements portray different emotions has value for virtual character animation and facial reconstructive surgery. Experiential evaluation of crowd simulation techniques can help make more believable characters for games and VR, and can help inform the field of motion planning to generate more natural robot movements in populated environments (Figure 1.1). Realistic simulations of human navigation behaviors provides can help guide simulation-based analysis for urban and retail planning.

Accurately capturing and modeling moving humans is a challenging and multi-faceted problem. Faithfully simulating human motion requires insights both for the representation for a generative model (what are the salient features that drive realistic motion) as well as the simulation technique (how they evolve over time). Furthermore, realistic motion must capture the diverse range of movements seen both across individuals and even within individuals on the same motion task. My thesis demonstrates that a key step to overcoming these challenges is leveraging *Motion Data* to both build and validate models of human motion.

(a) (b)

Figure 1.1: *left:* Natural motion is critical for smooth integration of AI into everyday settings (figure adapted from (Fan et al., 2018)). *right:* Natural motion is needed for virtual characters to enhance realism in digital media such as video games and movies (figure adapted from (Karamouzas et al., 2017)).

1.1 Motion Data

In recent years, substantial technological advances have been achieved in the fields of AI and computer vision, specifically in our ability to capture and learn from large amounts of data. The past two decades have seen substantial advancement of motion tracking techniques, including those specifically for capturing human motion at different scales (Brunetti et al., 2018; Li et al., 2013; OptiTrack, 2019). With these advancements come greater practicality for collecting *Motion Data*. Specifically, I will define motion data for the scope of this dissertation to be time series samples describing the spatial configuration (typically position and/or orientation) of a set of physical features of interest. These kinds of data have been identified as holding great promise for new and valuable insights about human behavior (Hui et al., 2009a), and are becoming increasingly available (Karamouzas et al., 2019).

This movement has helped support advancements in the understanding and simulation of human motion in a variety of contexts. For example, motion data has led to advancements in character animation that focus on local body movements and interactions with other characters and virtual objects (Starke et al., 2019, 2020), in-

cluding environments with realistic physical constraints (Peng et al., 2018). Others have utilized motion data to enhance realism in virtual crowds (Karamouzas et al., 2014; Sieben et al., 2017) and facial expressions (Helwig et al., 2017; Li et al., 2013). My work stands alongside these efforts, serving as both an expansion of some as a validation component, and an extension into new areas such as global navigation.

While there are many possible types of motion data, this thesis looks at two main kinds. The first is *face data*, which involves following key feature points of the face as they move to form various expressions. Face data is introduced in Chapter 2 and utilized for a facial simulation technique in Chapter 3. The second is *path data*, which tracks the pose of an entity as it navigates throughout a physical environment such as a building. I use path data in several contexts, including to evaluate crowd avoidance in Chapter 4, analyze and simulate navigation decisions in Chapter 5 and Chapter 6.

1.2 Leveraging Motion Data for Human Motion Simulation and Analysis

The accurate simulation of physical processes has been the subject of research far before the advent of modern computing. Discovering and understanding the laws of physics in domains such as aerodynamics or thermodynamics has made physically accurate systems of motion the subject of study in a variety of fields. With equations derived from the related research, many processes can be well defined and accurately simulated, leading to various kinds of technologies (such as those found in meteorology, civil engineering, or aerospace). In contrast, human motion is not well captured by existing physically-based simulation techniques, and presents its own set of unique simulation challenges. The simulation techniques I develop seek to capture the less frequently considered elements of human behavior, and utilize motion data as a source for learning.

This thesis will consider the use of agent-based models for simulation. In agent-based simulations, the **agent** is an autonomous entity that chooses actions to take in an environment based on available information to achieve some goal. An agent typically keeps track of some important information about itself and the environment (such as its position, orientation, velocity and goal) known as a **state**. An **agent-based simulation** is the step-by-step process in which an agent's actions are determined (using some observations about the environment and the agent's behavioral model) and enacted repeatedly until some termination condition is satisfied. The resulting series of agent states together represent simulated motion data, also called its **path** through the environment.

Developing any physical system that involves motion and motion planning by an agent presents multi-faceted challenges. These challenges include answering detailed, concrete questions such as where in the world the agent is, how sure the agent is where it thinks it is, and how the agent's actions will affect where it goes, as well as broader questions of planning and optimization strategy (given we know where we are, and an established set of dynamics, how to achieve the desired result). This dissertation will focus on the latter, specifically agents that are able to exhibit human-like behaviors in their motion. This requires producing *natural* motions that well match what is seen in humans, while maintaining as much as possible the rich *variety* of the resulting behavior to maximize **realism**. Following is an expansion of each of these ideas of naturalness and variety in motion, and how they combine to produce realistic simulations.

1.2.1 Creating Realistic Motion

Natural Simulations When a simulation appears natural, it suspends our disbelief and can go virtually unnoticed when contextualized with the appropriate surroundings. For example, when you walk amidst a crowd of other humans, your attention

need not be focused on the navigational behaviors of those around you. Likewise, if you were to walk through a crowd of a mix of humans and robots using very naturally simulated motion, you would not need to treat the robots differently in your planning. It is this inconspicuous resemblance to real-world observed human behaviors that I call naturalness. The methods I propose in this work seek to move toward simulation approaches that produce natural behaviors.

For example, robotics applications that involve a robot moving through populated areas (such as seen in Figure 1.1a) benefit greatly from being able to accurately model human navigation behaviors. Accurately simulating potential ways in which the flow of pedestrians might evolve over time can help inform motion planning systems to minimize the likelihood of collisions and help maintain a smooth overall path. Additionally, understanding the navigation decisions humans might make can help a robot avoid actions that would be perceived as unexpected or abrupt, causing undue congestion and inefficient use of space.

Variety in Simulation Equally important in human simulation is recovering the stunning variety found not only from person to person, but even in the same individual carrying out multiple instances of the same motion task. Each of us has unique facial movements that form the various facial expressions we can perform(Figure 1.2), and one's walking path through an environment (whether densely or sparsely populated) may take slightly different routes even given the same initial and goal conditions (Figure 1.3). Research has shown that variety in appearance of digital characters is an important factor in creating experiences that exhibit realism and immersion (McDonnell et al., 2008; O'Sullivan, 2009), specifically the human face (Sinha et al., 2006), which I consider in Chapter 3. To achieve this high level of realism, my work considers data-driven approaches seeking to account for and recover the variety found in the real-world behaviors they are simulating.

Figure 1.2: Variety is an integral component to reproducing realistic human smiles. (*images sourced from various public domains*).

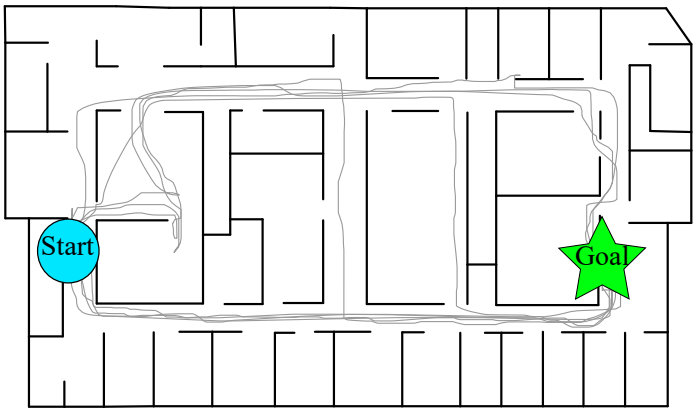


Figure 1.3: The top-down view of human paths (grey lines) in an indoor environment (black lines are walls). People took a variety of different routes in this environment when asked to navigate from a start to a goal (the participants had never seen the environment before).

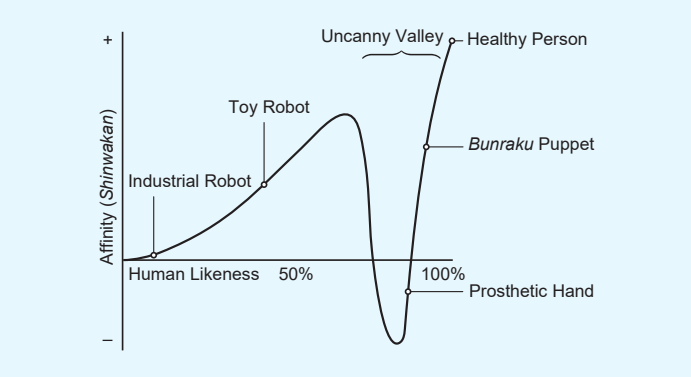


Figure 1.4: A depiction of the uncanny valley, showing the relationship between increasing human-likeness of an entity and the elicited emotional responses (adopted from (Mori et al., 2012))

.

Producing realistic simulations presents significant challenges. A popular ontology by which naturalness is often studied when pertaining to human-likeness is that of the *Uncanny Valley*. The main idea proposed in this theory is that as artificial representations of humans (be it images, videos or robots) become more realistic, they seem more natural or familiar, up until a certain point at which they (counter-intuitively) suddenly appear unsettling before crossing the line into indistinguishable from that of humans (see Figure 1.4). While some have questioned the merits of the familiarity-human-likeness curve (Brenton et al., 2005; Hanson et al., 2005), there is some consensus that the more lifelike something appears, the more we expect it to be beholden to realistic constraints (Brenton et al., 2005; Geller, 2008).

It is clear that the more methods strive for realism in human depictions, the more sensitive audiences are to subtle abnormalities. My goal to capture variety while simultaneously producing realistic simulation behaviors is impacted by this sensitivity, presenting a unique and precarious task. However, given data that sufficiently captures all these elements, my dissertation seeks to demonstrate that it is possible to learn and validate models via data-driven means that effectively recover and reflect

these important aspects of human motion.

1.2.2 Validating Simulation Realism

When using motion data to learn simulation models of human motion, the models themselves will be fit using an optimization process that uses a computational method to minimize an objective function. However, minimizing an objective function does not guarantee that the simulated motion will appear natural or capture a sufficient variety. To establish confidence that the resulting models indeed produce realistic simulations, a method for validating the realism of a simulation is needed.

In the context of this thesis, I refer to *Data-Driven Validation* as the process of using data to measure the extent to which simulated paths exhibit the important aspects of human motion such as naturalness and variety. I do this by conducting *User Studies*, allowing humans to be the judges and standard by which realism is measured. A User Study for validation consists of data collected from a number of willing participants designed to test the results of a simulation method. This data can take the form of subjective responses to the realism of a simulation, or can be motion data generated by participants for objective comparisons to simulated paths of the same motion tasks. Examples of both kinds of user data for validation are proposed and employed on the simulation methods seen in this dissertation.

1.3 Thesis Statement

To address the goals discussed above, my work demonstrates evidence for the following thesis:

Data-driven techniques can be used to capture, model, and faithfully simulate hu-

*man motion in a variety of contexts. The resulting set of methods are able to characterize and exhibit **realistic** motion behaviors, appearing **natural** in form while recovering the rich **variety** of motion seen in humans. In doing so, human motion data can be employed to drive **simulations** as well as to **validate** the effectiveness of simulations.*

1.4 Main Results

I begin the dissertation by presenting my approach for identifying and addressing the fundamental conflict between creating natural motion and a variety of motions, applied to simulating realistic human smiles. The next two chapters demonstrate leveraging human paths for the purposes of evaluating the realism supported by other data-driven models, and to generate new insights on human motion for use in simulations. The final chapter focuses on simulating a variety of compelling paths, and is evaluated using human paths. Following is a synopsis of each.

1.4.1 Building a Dataset to Capture the Essence of a Smile

To support studies of human smiles, I begin by describing my effort to design and conduct a large scale user study in Chapter 2. My collaborators and I collect thousands of casual observer responses to a systematic sweep of facial expressions on a 3D computer animated model. The study took the form of a mobile app running on a tablet, with the screen split between a video of a random expression from the pool and a panel of response options. For each video, participants could simultaneously view and consider the options, proceeding to respond to subsequent videos if they so wished.

The results of the study provide over 10,000 responses each containing 4 measures of the perceived emotional content of the expressions. For my purposes of simulating

realistic smiles, I applied some post processing to aggregate responses into robust measures of smile intensity for each facial expression, and categorized the smiles into a set of smile intensity classes. However, the original dataset itself also exists as a valuable resource interdisciplinary research in other fields. One example is in quantitative psychology for exploring the dynamic properties of successful smiles (Helwig et al., 2017). Additionally, both the data and the notion of capturing casual observer perceptions of facial expressions to broaden scientific understanding of facial expressions has informed and inspired work in medical fields such as facial reconstructive surgery (Lyford-Pike et al., 2018).

1.4.2 Data-Driven Simulation of Realistic Smiles

Happiness is among one of the most basic and important emotions conveyed by the human face. As smiles vary widely both within and across individuals, creating compelling virtual characters must also exhibit these kinds of diversity. In this work I propose a method for creating natural, varied human smiles. I formalize the problem within the context of a generative method for creating virtual character smiles in an interactive setting such as a video game. This involves identifying quantitative measures of quality (naturalness), diversity, and for the generative model I choose a classifier. The classifier is trained on a dataset which is able to categorize points in facial space as to their semantic class (low, medium or high smile intensity). Smiles can then be generated by rejection sampling from the classifier until an appropriate facial space point is found.

I then introduce the notion of the precision-variety trade-off. This is the fundamental conflict between a "safe" classifier that is more restrictive (high precision, but the positive decision boundary covers a low area of facial space) and one that more freely classifies candidates as belonging to a certain semantic category (higher chance of false positives, but larger coverage of the space of possible expressions, leading to

Figure 1.5: A variety of happy smiles generated by my method for two different 3D models.

more variety). I propose a heuristic for ordering the data such that as more data is included, it tunes this trade-off to allow for the maximum amount of variety given a precision threshold. This heuristic and corresponding algorithm I call *Precision-Variety Learning* (PVL). I explore the theoretical properties of datasets that are likely to produce a precision-variety trade-off curve, and show that when certain assumptions hold, my heuristic is guaranteed to produce a monotonic curve useful for tuning the trade-off.

I demonstrate the efficacy of the method by applying it to the problem of simulating a variety of mouth shape animations corresponding to realistic human smiles with targeted smile intensity (Figure 1.5). I validate the realism of the results with a two-part follow-up user study, while also showing that the facial feature space can be successfully applied to other facial models.

For a detailed description and analysis of this work, see Chapter 3. The original publication can be found in the Association for the Advancement of Artificial Intelligence 2018 conference proceedings (Sohre et al., 2018).

1.4.3 Validating Collision Avoidance as a Realism Technique

Dynamic, moving characters are increasingly a part of interactive virtual experiences enabled by immersive display technologies such as head-mounted displays (HMDs).

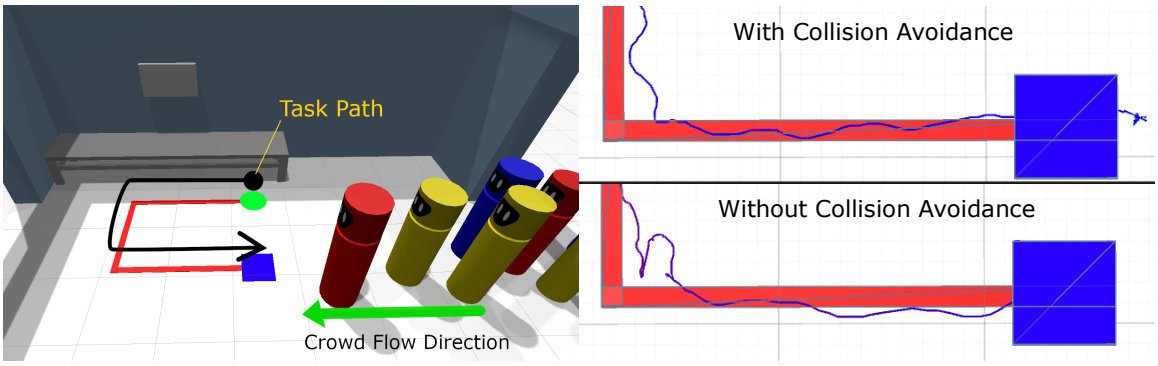


Figure 1.6: Experimental setup and example participant paths for both collision avoidance (*top right*) and non-collision avoidance (*bottom right*).

In this context, it is important to consider the impact their behavior has on user experiences. To evaluate the impact of collision avoidance in virtual environments, I design and conduct a user study to capture paths and experiences of participants interacting with a virtual crowd in an immersive 3D environment.

In the study, the participants are tasked to walk through a crowd of virtual agents with an opposing flow (Figure 1.6 *left*). Two possible collision avoidance conditions are implemented for the virtual crowd: one in which the agents exhibited anticipatory collision avoidance behaviors with both each other and the participant, and one in which the agents exhibited collision avoidance with each other, but not the participant. Each participant performs the task twice, where in one trial, collision avoidance was enabled, and disabled in the other (the condition order was randomized to combat ordering effects). During the task, the participants' 3D positions and orientations are tracked using an external motion capture system. Additionally, participants fill out a simulator sickness questionnaire both between the two trials and after completion of the experiment.

I then perform and present an analysis of the collected data. The results from the follow-up questionnaires show a strong impact of the collision avoidance on a participant's feeling of presence and realism of the crowd. As an objective measure, I

perform a path analysis on the tracked paths by computing the total acceleration of each. A comparison of the two conditions shows statistically higher total acceleration felt by participants when there was no collision avoidance, consistent with the jarring discomfort felt by the participants as they attempted to navigate through a crowd where the agents ignored and seemingly passed through them (Figure 1.6 *right*).

This work was originally published in the 2017 IEEE Virtual Humans and Crowds for Immersive Environments Workshop (Sohre et al., 2017). Chapter 4 contains a presentation in greater depth.

1.4.4 Data-Driven Insights for Multi-Task Human Navigation Decisions

Understanding human flow through indoor buildings is important for various layout design tasks such as evacuation planning, product placement, and security. Advancements in technologies such as computer vision and motion-tracking have enabled the collection of large amounts of high quality, long-term motion data. In this work I take a data-driven approach to analyzing the navigation decisions of shoppers in a grocery store.

The data includes point-of-sale transactions representing sets of items purchased together (*baskets*), along with a 2D embedding of item locations within the store's floor layout, which includes obstacle walls. Each basket is a list of time-ordered items corresponding to the order in which they were retrieved.

For the purpose of analysis, I model a shopping trip as a series of decisions where a shopper must choose the next item to pick up. The data show that most of the time, a shopper goes to the item having the *shortest path walking distance* from the their current location. I define an *inversion* to be any case when a shopper makes a "mistake" and travels to something farther than the closest item. My analysis of all

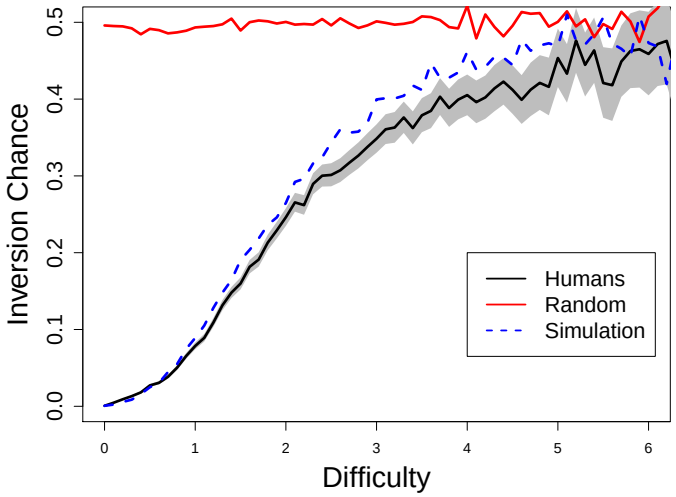


Figure 1.7: The chance of choosing the farther of two items as a function of difficulty (entropy) score for the shopper data (black with grey confidence region), simulated data (blue dashed), and random choice.

the pair-wise item comparisons in the data for each navigation decision shows that the *entropy* of the ordering task for an item pair (that is, the information required to determine their order) governs the likelihood that they are inverted in the shopper's navigation decisions. I adopt this as a measure of difficulty, which monotonically increases the likelihood of an item pair being inverted in the data (Figure 1.7).

Drawing on this insight, I propose a one-parameter simulation model to generate plausible item orderings given a hypothetical basket. I provide a theoretical guarantee that this model will have the same relationship between the entropy of pair-wise ordering tasks and the likelihood of inversion in simulation. Simulating on the baskets in the data, I show the resulting item orderings match the data with high accuracy along several key trends. The stochastic nature of the model naturally supports the ability to generate a variety of plausible shopping routes for the same basket.

Chapter 5 contains a presentation of this work in full. A pre-print publication of this work is available on ArXiv at article reference arXiv:2102.00057 (Sohre et al., 2021).

1.4.5 Realistic Navigation Behavior with Uncertain Goals in Building-like Environments

The interactive simulation of human motion is important in many scenarios, with applications ranging from video games to building design and smart city planning all benefiting from high-quality human movement and behavior. In Chapter 6, I present a deep-learning method for producing realistic human-like routing behaviors through indoor environments.

In this work, I take a deep neural network approach to global navigation. To overcome the challenge of collecting the large amounts of human paths it would take to train on motion data directly, I propose that in general, humans attempt to take efficient paths, but make mistakes due to the local nature of the information available to them. I then model this conflict in my formulation of the navigation problem by limiting the information available to the network to the same constraints that humans face in unfamiliar settings. Then, the network can be trained on generated, globally optimal routes.

To do this, I formulate the global navigation task as a series of discrete navigation decisions between waypoints in an environment. At each step, the network takes in local isovist features as well as a partial path history and goal region, and outputs multiple predictions of the next step in the optimal path. The neural network architecture I propose (Figure 1.8) utilizes a custom loss function to integrate the inputs in a way that produces intelligent predictions based on an internal map representation built up over its path so far.

I show that the trained model reproduces several human-like routing behaviors, such as narrowing down goal locations and intelligent back-tracking. Additionally, the network is capable of identifying multiple promising directions in ambiguous scenarios, which can be used to generate a variety of human-like routes. I provide an analysis

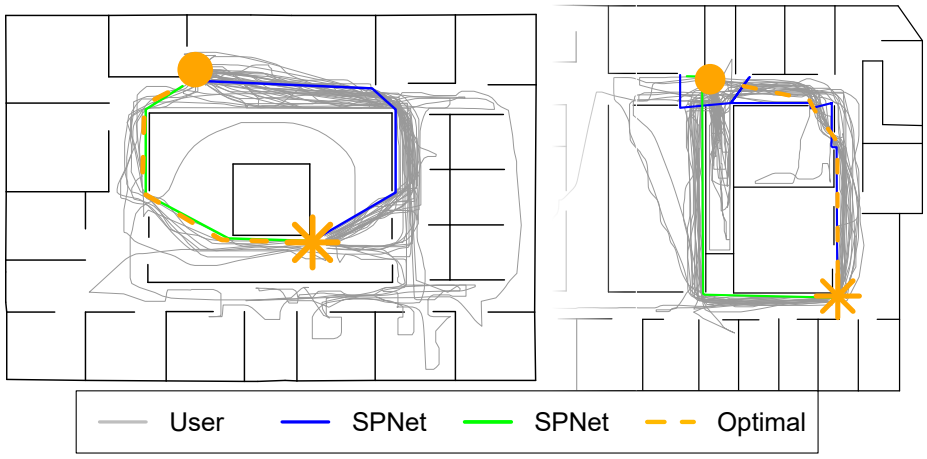


Figure 1.8: Example paths generated with an SPNet agent are shown overlaid on user paths and the optimal path (dashed) for two navigation tasks (start is indicated by a circle, the goal is a star).

comparing the generated paths to human paths from a user study I conduct on the same tasks via a 3D game-like interface. The various paths generated on many tasks well matched the distribution of high level routes taken by users (Figure 1.8), as well as on average being closer to user paths in length than the optimal route or by using heuristics proposed in cognitive science literature.

This work was originally published in the 2020 ACM Conference on Motion, Interaction and Games (Sohre and Guy, 2020). For more details, see Chapter 6.

1.5 Impact & Contributions

My dissertation focuses on an important area of research, not only as it pertains to the broad goal of social AI, but in its widespread impact in various existing domains. For example, better understanding the dynamics of human smiles offers significant benefit to the fields of facial reconstructive surgery and physical therapy, while simultaneously enhancing facial animation techniques for computer graphics applications. Characterizing the impact of collision avoidance on user experience advances the state

of the art for applications in VR and robotics. Learning to understand, anticipate, and reproduce the decision patterns humans exhibit in long-term navigational planning, particularly under local information constraints, enables more human-like agents in interactive digital media such as video games, can improve the expected efficiency of mission planning for robots in unknown buildings, and enhances building layout design, safety and security.

To that end, this dissertation includes the following contributions to the state of the art in computational methods for the data-driven analysis and insight of human motion:

- A large dataset containing responses from hundreds of participants on their assessments of computer animated smiles
- A framework for the data-driven procedural generation of natural, varied human smiles
- A machine learning heuristic for governing the trade-off between precision and variety (PVL)
- A validation of collision avoidance for producing realistic crowds in immersive virtual reality settings
- A data-driven analysis of shopper paths that reveals an entropy law describing local errors in navigation decisions
- A novel neural network architecture for incorporating goal uncertainty into global navigation tasks under local information constraints
- A data-driven algorithm ($SPNets$) for simulating realistic global routes through indoor environments using path-optimal training data, validated against human paths

1.6 Dissertation Organization

The overall structure of this dissertation is as follows. Chapter 3 describes my work capturing a large dataset to support various analyses, including enabling a richer variety of motion in simulated human smiles while preserving naturalness. Chapter 4 covers using real-world human paths to validate the importance of reactive collision avoidance in supporting natural and comfortable user experiences in VR. Chapter 5 and Chapter 6 cover work that studies the underlying factors leading to natural and varied global navigation behaviors and planning decisions.

Chapter 2

Building a Dataset to Capture the Essence of a Smile

Happiness is among one of the most basic and important emotions conveyed by the human face. As technology has become more sophisticated and graphics capabilities increase, the desire for more realistic characters in both appearance and motion also grows. However, the interest in understanding the motion that characterizes different emotions, especially happiness, extends beyond the domain of computer graphics. Researchers in psychology as well as medicinal fields such as reconstructive plastic surgery are also very interested in exploring the spatio-temporal dynamics of successful smiles.

In an endeavor to support the multifaceted, interdisciplinary research to deepen the state of the art in the understanding of human smiles, a group of colleagues from different fields and I designed, implemented and conducted a large scale user study that took place at the 2015 Minnesota State Fair. Following is a detailed description of the study design, resulting data, a corresponding feature embedding, and examples of its broad applicability in other fields.

2.1 Facial Space

My collaborators and I propose a low dimensional configuration space to parameterize expressions based on key facial feature points as a manifold for learning the connection between motion and perception of smiles. Through consideration of related literature in facial medicine and psychology, these feature points are chosen to focus on the region of the mouth. The mouth has been shown to be a principal source of variation in facial expressions (Köhn, 2006), as well as the primary source of information for detecting happiness (Nusseck et al., 2008). One of my colleagues, a board-certified facial reconstructive surgeon, further identifies semantically meaningful areas of the mouth, which we combine to create the manifold. This includes a combination of three facial features: *lip corner angle, lip corner extension,* and *dental show*. We use these features to compose our mouth configuration space, referred to here as *Facial Space*.

Formally, Facial Space is made up of three interactions between mouth feature points: the intersections of the upper and lower lips and the sagittal plane, and the left mouth corner (spatial symmetry across the sagittal plane helps support low dimensionality). The values for each feature are computed as relationships between these control points (see Figure 2.1). Each measure is normalized relative to the inter-pupilary distance of the face to standardize across different face sizes, which is known to be proportional to other features of the face, such as resting mouth distance (Stephan, 2003).

The naturally low dimensional nature of this feature space provides several benefits. One is that it is easy to use and understand. Another is that low dimensional feature spaces are conducive to computational learning for data-driven models. Being derived from key facial points, each dimension is semantically meaningful, which extends well to other fields of study. Additionally, facial space is a standardized space,

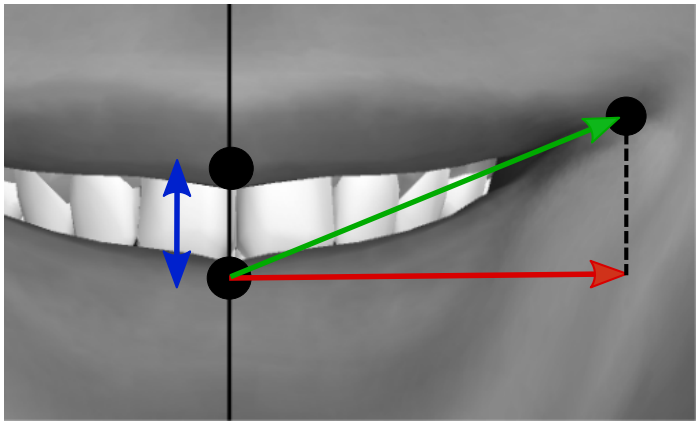


Figure 2.1: Computation of facial space features. Control points are shown as dots, and the vertical bisecting line (black) shows the position of the sagittal plane. Angle is computed as the angle between the diagonal (green) and horizontal (red) arrows. Extent is the length of the horizontal arrow, and dental show is the extent of the vertical arrows (blue).

and generalizes to any setting for studying faces where these facial features can be tracked.

2.2 User Study Design

To capture the relationship between movements in facial space and perceived emotional intent, my collaborators and I take a crowd-sourced data-driven approach. Collecting many reactions to different motions across many individuals enables many types of analysis, such as statistical models that describe these relationships. To do this, I designed and implemented a mobile app that allows users to easily and smoothly participate by tapping response options on tablet devices. The study involved participants responding to a sequence of video stimuli composed of digital facial animations representing array of anatomically plausible faces rendered on a high quality 3D model. For each stimulus, the user was able to view and replay the video as long as they desired, and could submit as many or as few responses as they

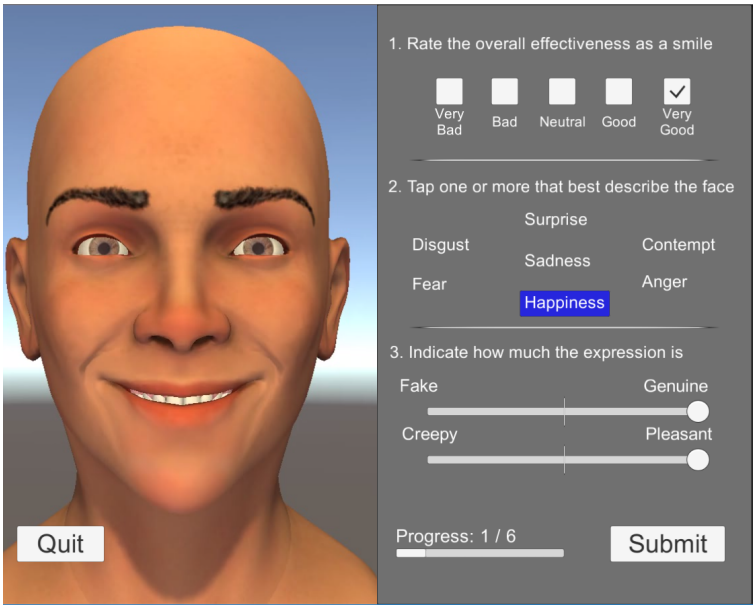


Figure 2.2: A screenshot of the application used to conduct the user study. Subjects responded to stimuli by rating each in terms of smile effectiveness and emotional intent.

liked (up to the total number of available stimuli). The order of stimuli presented was randomized to ensure even coverage of responses on the different expressions. For each stimulus, participants were asked to evaluate each in terms of emotional intent, as well as assign a quality score for how well the face portrayed a smile. The stimuli contained mostly smile-like faces, but also had some negatively angled facial expressions, which served as controls. Figure 2.2 shows an exmaple screenshot of the app, containing the 3D model, a sample expression, and the response UI.

2.2.1 Using A 3D Model to Study Smiles

Using computer animated faces for both communicating and studying human perception of emotion has been utilized in previous works (Griesser et al., 2007). In addition, the literature shows that user studies of human perception of virtual characters is useful for studying their emotional expressiveness (Liu et al., 2016).

Several methods have been proposed for the digital representation and manipulation of the human face. Common to all methods in computer graphics is the use of a 3D spatial mesh to represent the face, that is then deformed according to some model of facial movement. Such models include those based on physical interactions of skin, subcutaneous tissue, muscle, and skeletal structure (Waters, 1987; Cong et al., 2015; Lee et al., 1995; Sifakis et al., 2005, 2006). These approaches can achieve very realistic behavior, at the cost of a high level of complexity in the model. Parametric descriptors based on these models have also been developed to categorize muscle movements as facial actions (Ekman and Friesen, 1977; Essa and Pentland, 1997).

Another model involves mixing amounts of predefined deformations of a base mesh (that is, a set of alternate 3D meshes with a 1-1 mapping between corresponding vertices in each). Known as animation *blendshapes*, these are widely used in both industry and research. A large body of work involves producing customized blendshapes (or other representations of facial movement) based on the facial performance of an actor. Tracking the movements of an actor's face, called performance capture, is then translated into deformations of a 3D facial mesh in a process called retargeting. Many approaches to this task have been proposed (Zhang et al., 2016; Bouaziz et al., 2013; Li et al., 2013; Xu et al., 2014). Researchers have proposed various method to accomplish this in a way that accurately captures the original performance (Li et al., 2013; Costigan et al., 2014; Zhang et al., 2016).

For our study, my collaborators and I chose to use a computer animated model to create the videos used in the study over footage of real human facial expressions or facial performance capture techniques as it provides several benefits for our research goals. For example, Using an animated model allows a greater level of control over the resulting expressions than would be achievable by human actors. An actor may be unable to contort their mouth to a precisely defined set of coordinates for key feature points, or completely hold one part of their face completely still while moving

others. Additionally, a 3D model allows the expressions to move outside the range of motion an actor may posses, allowing the data to include a broader coverage of interactions between facial movement and perceptual affect. This provides more support for machine learning applications that benefit from good coverage in both positive and negative regions of facial space.

The facial model used in our study was hand-crafted by one of my co-authors (a 3D artist) in collaboration with medical professionals. The result is a highly detailed 3D facial model (seen in Figure 2.2), along with a set of blendshapes to support a wide range of plausible facial motions. To create the expressions used in the study, parameters corresponding to blendshape weights were systematically swept to produce a rich sampling to draw from in analysis.

2.3 Study Results

Over 900 subjects participated in the survey, providing over 10,000 data samples. A summary of participant demographics is depicted in Figure 2.3. A single sample contains, for a given facial expression, the smile quality score the participant assigned, any emotions the participant associated with the expression, and two Likert scale measures on the axes of authenticity and pleasantness. My work focuses on utilizing the smile quality scores. I aggregated the scores for each facial expression, producing the average quality score for each face. The result is a dataset composed of 63 facial expressions annotated with perceived smile quality. Although the user study stimuli were rendered on a single facial model, the design and use of facial space is intended (and subsequently verified in Section Chapter 3) to generalize the findings to other facial models.

The results of this processing are summarized in Figure 2.4a. The range of quality scores is well covered by the systematic sweep of facial space points, with the majority

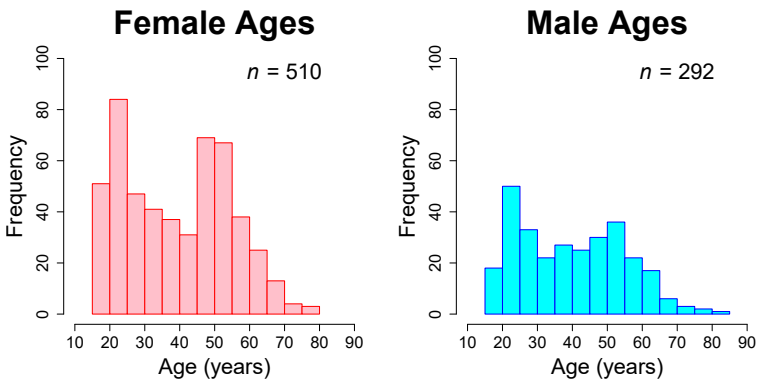
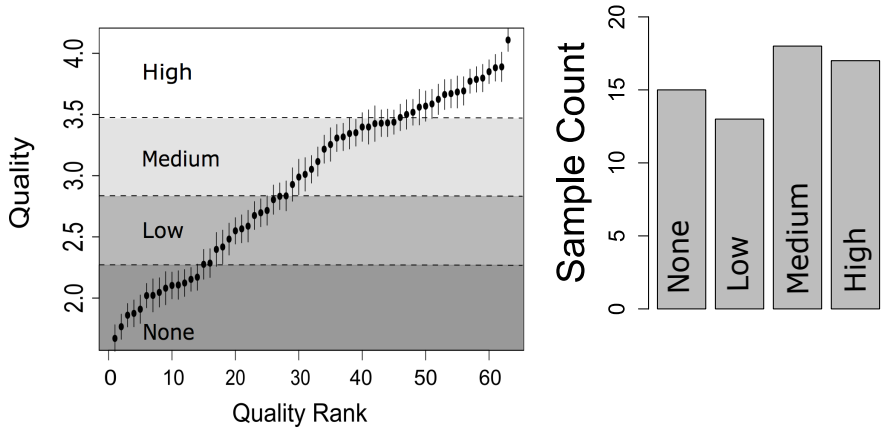


Figure 2.3: The distribution of responses along gender and age (adapted from (Helwig et al., 2017)).

of the standard errors being reasonably small. Additionally, to support my work with classifiers in Section Chapter 3, I created smile intensity bins. The quality thresholds for membership are shown as the shaded regions in Figure 2.4a, and the counts of faces in each is shown in Figure 2.4b.



(a) Smile Quality Scores (b) Happiness Bins

Figure 2.4: **User Study Data. (a)** Each face's smile quality with its standard error. The data is binned into effective happiness bins. **(b)** The distribution of faces by happiness bins.

2.4 Interdisciplinary Collaborations

A prime example of the extension of this work's value to other fields is that of (Helwig et al., 2017), on which I was a co-author. In that article, my psychologist collaborators explored the perceived emotional intent along the varying axes of facial space to build an understanding of what motions make a successful smile. Among other results, they found that well accepted smiles struck a harmonious balance between mouth angle, smile extent, and dental show coupled with dynamic symmetry. Figure 2.5 demonstrates the interactions between these elements.

Additionally, the efficacy of the approach taken for capturing casual observer perception of facial expressions I helped to prove out in this study inspired subsequent works with my collaborators in facial reconstructive surgery. In (Lyford-Pike et al., 2018), I helped conduct a similar user study where we collected responses from over 500 participants to help build a connection between clinical scores of facial function

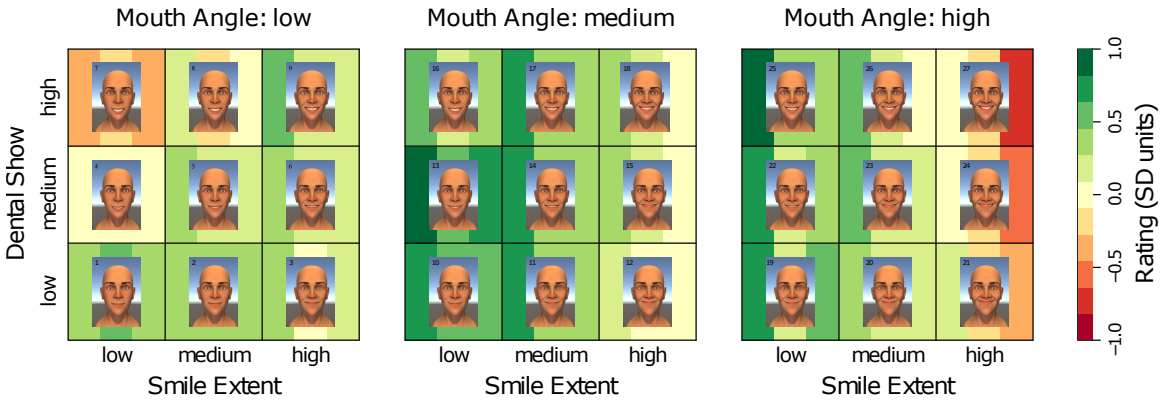


Figure 2.5: (Adopted from (Helwig et al., 2017)): A heat-map plotting the interaction between the facial space parameters. The three vertical bars behind each face denote the predicted score for the three response variables: effective, genuine, and pleasant (respectively). Greener colors correspond to better smiles, and redder colors correspond to worse smiles.

and casual observer perceptions of disfigurement. The results help reconstructive surgeons and clinicians better assess intervention success, helping victims of facial paralysis achieve a higher quality of life.

Chapter 3

Data-Driven Simulation of Realistic Smiles

In this chapter, I demonstrate how data can be used to create natural, varied motion by leveraging the results of the user study presented in Chapter 2 to produce a model of what makes human mouth movements convey happiness. I then use this data to investigate the problem of procedurally generating a diverse variety of facial animations that express a given semantic quality (e.g. very happy). To accomplish this, I introduce a new learning heuristic for generative models called Precision Variety Learning (PVL) which actively identifies and exploits the fundamental trade-off between the precision of the model, and the variety of possible outputs. I both identify conditions where important theoretical properties can be guaranteed, and show good empirical performance in a variety of conditions. Lastly, I apply the PVL heuristic to the motivating problem of generating smile animations. The results are validated by a follow-up user study showing the generated smiles exhibit a high level of realism (naturalness and variety) for different semantic targets (e.g. very happy, slightly happy).

A version of this work appears in the Association for the Advancement of Artificial Intelligence 2018 conference proceedings (Sohre et al., 2018).

Figure 3.1: A variety of happy, smiling mouth shapes generated by our method, rendered in a high-quality real-time engine.

3.1 Introduction

Virtual humans are increasingly a part of our games and other digital media. They appear in movies as animated actors, video games as interactive non-player characters, personal avatars in games, virtual reality and social media, and are even used to control human-like robots. A critical challenge in the field is to generate animations which accurately reflect the state of the animated characters, without looking repetitive or unnatural. A key component of creating compelling interactions with digital characters is the animation of the human face. Humans use and expect faces to produce a variety of cues for nonverbal communication such as intonation and the expression of emotions. Understanding the full variety of movements that control and effect these cues is important both to fields that study real humans (e.g., medicine and psychology) as well as those which seek to create realistic virtual characters (e.g., games and movies).

My goal in this work is to create algorithms that can automatically generate a variety of realistic animations for virtual characters, a problem which is closely related to a field of AI known as Procedural Content Generation (PCG). PCG is especially relevant in the realm of games and interactive digital entertainment where is it important to present the user with engaging, dynamic experiences that respond to the user's actions in real time.

For procedurally animated virtual characters to meet their goal of emotionally engaging the users, there are two important qualities the procedural animations must maintain. First, it is important that their expressions are as high quality and natural in appearance as possible. If the generated motion is halting, confusing, or otherwise unrealistic in its execution, the users will be distracted from the intended emotional content of the expression. Second, the procedural generation system must be able to create a variety of motions that is reflective of the full diversity real people have in showing the same basic expression. In fact, the importance of variety in character animations has been established through multiple users studies (McDonnell et al., 2008; O'Sullivan, 2009) and has been highlighted as an important challenge in PCG (Preuss et al., 2014).

Unfortunately, these dual goals of generating high quality content and generating a diverse variety of content are often in direct conflict. Algorithms that focus too much on the quality of their content often do so by sacrificing the variety of their output. In this paper, I examine this trade-off in the context of procedural systems for creating mouth movements for virtual characters to form smiles of different intensity (e.g., slight, full, none), and propose new methods to produce a broad diversity of smiles that accurately display the target intensity level. My work presents three main contributions:

- *Formalization and analysis of quality-variety trade-off*: I formally define the notions of quality and variety for a certain class of content generation models (constraint-based optimization formulations), and explore the theoretical basis of the inherent trade-offs between the two.

- *Precision Variety Learning heuristic (PVL)*: I introduce a framework for a constraint-based optimization formulations of PCG which allows a user to tune the level of precision needed for a specific application, and automatically maxi-

mize its variety of procedurally generated content for a given level of precision.

- *Variety-Enhanced, Data-driven Facial Animation System*: I apply the PVL generation approach to a non-parametric classifier trained on the dataset introduced in Chapter 2 in order to create a system capable of producing a large variety of smiles at a given level of smile intensity. I evaluate the quality and diversity of the resulting smiles through user studies.

While the results presented here focus on PCG smiles (e.g., see Figure 3.1), the approach is generic and can be directly applied both to other facial expressions (e.g., sad, angry) and to other forms of procedurally generated content.

3.2 Background

The animation of digital human-like faces has a rich history in the literature, from performance capture to modeling, to human perception of facial actions, and creating facial expressions for digital characters. Likewise, the study of PCG is a quickly growing field, covering everything from game maps and mechanics to textures and audio (Hendrikx et al., 2013). Below, I briefly highlight some closely related works.

3.2.1 Facial Animation

There is a rich literature surrounding the task of facial animation, an overview of which can be found in Vinayagamoorthy et al. (2006). The most common technique is the use of a 3D spatial mesh that is then manipulated according to some model of facial movement. As with the models I employ here, many models of natural facial deformations are based on interpolative blendshapes (Zhang et al., 2016; Bouaziz et al., 2013; Li et al., 2013; Xu et al., 2014). Blendshape-based models involve linearly interpolating the mesh between a set of exemplar configurations.

In many cases, the approach to animating these models utilize the capture of a facial performance by a human actor. Researchers have proposed various methods to accomplish this, from adaptive dimensionality reduction (Li et al., 2013), to neural networks (Costigan et al., 2014) to local patch alignment (Zhang et al., 2016), and generating blendshape segmentation schemes (Joshi et al., 2005).

Generative methods for digital character facial expressions have also recently been explored. Some generate facial expressions from dialogue audio and text transcripts (Marsella et al., 2013). Physically-based models of the face can also be used to synthesize facial animation, such as speech (Sifakis et al., 2006).

Researchers have employed user studies to evaluate the effectiveness of digital character animation (Kokkinara and McDonnell, 2015; McDonnell, 2012; Liu et al., 2016), as well as to study the impact of variety (McDonnell et al., 2008; O'Sullivan, 2009).

3.2.2 Machine Learning for Facial Analysis

Supervised learning is the most closely related area of machine learning to this work, surveyed in Kotsiantis et al. (2007). Others have developed specialized algorithms to recognize faces and facial actions (Pantic and Rothkrantz, 2000; Franco and Treves, 2001; Bartlett et al., 2005), as well as recognizing emotions (Michel and El Kaliouby, 2003).

3.2.3 PCG as Machine Learning

There are many PCG techniques, and some synopses of the field are given in Smith (2014); Hendrikx et al. (2013). Recent works have considered how to create engaging (Togelius et al., 2013), diverse (Liapis et al., 2015), and interactive (Yannakakis and Togelius, 2011; Smith, 2014) content. Machine learning techniques can be ap-

plied to PCG problems in different ways, as content evaluators or to generate content directly (Summerville et al., 2017; Togelius et al., 2011).

3.2.4 Diverse, High-Quality Content

Quality-Diversity algorithms have recently been identified as an important type of algorithm, with search-based approaches like evolutionary algorithms (Pugh et al., 2015) and Human-in-the-loop methods that combine user input with search to efficiently traverse search spaces (Mouret and Clune, 2015) showing promise in this area. To the authors' knowledge, this work is the first to propose a machine-learning-based approach for this class of algorithms.

3.3 Problem Definition

As a motivating context for my problem formulation, consider the task of creating a 3D role-playing style game (RPG) where the player is immersed in an open world, free to explore and interact with many non-player characters (NPCs). To keep the NPCs engaging, their behaviors should be both appropriate to context (e.g., convey the right emotion), and appear natural and lifelike (i.e., not mechanically repetitious or robotic). To do this, a plausible method must be able to produce facial movements that exhibit the desired semantic meaning, while capturing the diversity of motion seen in real human faces, both within and across individuals. With these two goals as the primary focus, I establish a formal definition of the problem.

I represent facial animations as parameterized into a feature space $\mathcal{F}$, so that $x \in \mathcal{F}$ represents a complete facial motion, and define $\mathcal{S}$ to be the set of semantic labels. Let us also define a function $D : X \subseteq \mathcal{F} \mapsto \mathcal{R}$ that operates on a set of faces to measure its diversity, and a function $Q_s : X \subseteq \mathcal{F} \mapsto \mathcal{R}$ as the quality of a set. Finally, let $C_s : \mathcal{F} \mapsto \{0, 1\}$ be a binary function that identifies whether or not a

given animation exhibits a target semantic label s.

Then, given some target $s \in \mathcal{S}$, the task is to find the set of faces exhibiting the desired semantic that maximizes the diversity and quality functions:

$$\underset{C_s}{\operatorname{argmax}} \big[Q_s(X), D(X) : \forall x \in X \big(C_s(x) = 1\big)\big]. \tag{3.1}$$

This equation represents a multi-objective optimization problem. To develop a solution for the domain of facial animations, we need to develop quantitative definitions of Q and D, identify an appropriate feature space for $\mathcal{F}$, and learn C. The remainder of this section describes my approach to each, followed by a proposed method for actually generating animations.

3.3.1 Measuring Quality & Diversity

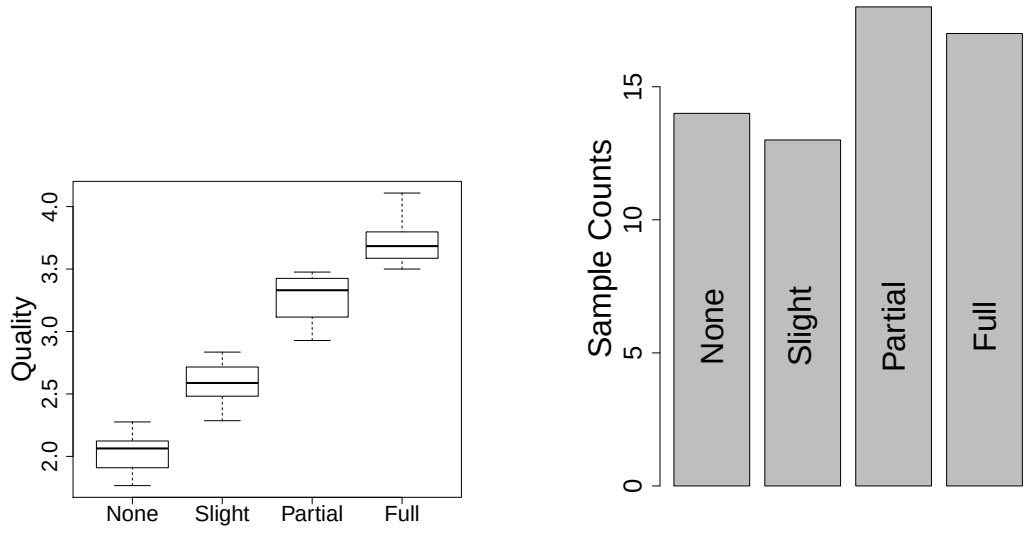
From here on I will assume X to be a finite set that is representative of C's continuous positive decision region in feature space. Then I define the Quality $Q(X)$ as the percentage of $x \in X$ that are true members of the target class:

$$Q_s(X) = \frac{|\{x \in X : [C_s^*(x) = 1]\}|}{|\{X\}|}. \tag{3.2}$$

Where C_s^* is the true semantic label function. In the context of Equation 3.1, this is equivalent to the precision of the classifier D, which is how we will measure Q.

To measure the diversity of a set X, I take its cardinality. This approach is consistent with existing measures of diversity for finite sets of candidate samples proposed in PCG (Preuss et al., 2014). Formally,

$$D(X) = |\{X\}|. \tag{3.3}$$



(a) Smile Quality Scores (b) Happiness Bins

Figure 3.2: **Training Data (a)** A visual summary of the semantic classes. **(b)** Sample counts by class.

An important property for D is that adding members to X can never decrease the diversity measure overall (other reasonable diversity metrics, such as the variance of the set, do not satisfy this property).

3.3.2 Feature Space (F)

In Chapter 2, I proposed a generalizable, low-dimensional feature space to be used to represent smile animations. I refer to this feature space as *facial space*, and adopt it for $\mathcal{F}$. This feature space is composed of three key distances between feature points surrounding the mouth as identified by medical professionals: angle, extent, and dental show. Angle is computed as the angle between the bottom lip and mouth corner, extent is the width of the smile, and dental show is the separation between the upper and lower lips. The focus of this feature space on the mouth shape is consistent with previous research and has established the effectiveness of mouth shape

for identification of facial expressiveness (Gosselin and Schyns, 2001) and specifically happiness (Nusseck et al., 2008).

3.3.3 Classifier (C)

By definition, the task for C is one of classification. To do this, I construct binary classifiers from annotated data via supervised learning that maps samples to a membership prediction given a target class in $\mathcal{S}$. This formulation allows for any binary classifier, though different classifiers will have different theoretical properties and performance. Here, I consider several well established classifiers:

- *Nearest Neighbor models (KNNs)*: I employ a variant of KNN known as Restricted Neighborhood Search. The prediction for a sample is positive if a sufficient number of nearby neighbors (called witnesses) within some distance r are positive. The prediction for a query sample $q \in \mathcal{F}$ is positive if and only if

$$\frac{\sum_{x \in \mathcal{W}_q} \pi(x)}{|\mathcal{W}_q|} \geq t \wedge |\mathcal{W}_q| \geq k, \tag{3.4}$$

  where $\mathcal{W}_q$ is the set of witnesses for q, t is the minimum proportion of witnesses that must be positive, k is the minimum number of witnesses to make a prediction, and $\pi(x)$ takes the value 1 if x is a positive training sample and 0 otherwise. For my classifier, I choose $k = 6$ based off the density of my training data, $r = 0.4$ based on the distribution of inter-point distances, and $t = 0.3$ via tuning.

- *Support Vector Machines (SVMs)*: these classifiers use quadratic programming to find a linear separator between positive and negative samples that maximizes the margin between them. A key property of SVMs is their use of kernels, which transform training data into higher dimensional spaces (where linear separators

are more likely to be found) before measuring distances via an inner product. In this way, learning can take place in a high dimensional space while computation stays in a low dimensional space. In this paper I use the *kernlab* SVM package (Karatzoglou et al., 2004) for the R programming language, using the "vanilla" kernel.

- *Random Forests (RFs)*: these classifiers take many random subsets of the training data and build decision trees on each. For prediction, a majority vote is taken of the random decision trees on the query sample, combating the tendency of decision trees to over-fit. To build Random Forests I employ the *randomForest* package for the R programming language (Liaw and Wiener, 2002) with 1000 trees.

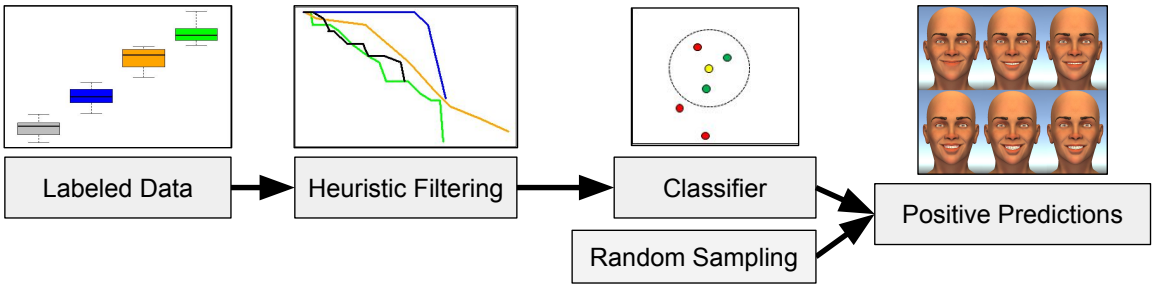


Figure 3.3: A graphical overview of my approach.

3.3.4 Semantic Classes (S)

I choose $\mathcal{S}$ to be a set of discrete classes derived from training data that is then used to learn C. Discrete classes are motivated in part by the scenario of RPG-style video games; here, characters typically need to display one of a small set of emotions depending on the players behavior. Additionally, classification is a natural formulation for this problem (as opposed to regression) in that it allows a single semantic class to contain a variety of feature space points.

3.4 Approach & Implementation

An overview of how I utilize $\mathcal{S}$, C, and $\mathcal{F}$ to generate facial animations can be seen in Figure 3.3. Once training data has been labeled for a target class, I apply my learning heuristic as a pre-processing step, which I discuss in detail in the next section. A binary classifier is then trained on the labeled data to predict target class membership. I then use rejection sampling to generate new animations, uniformly sampling $\mathcal{F}$ and passing them through C keeping only those that yield positive predictions. To render these new samples as a human facial expression, they must be transferred onto a digital facial model. For this I use a 3D mesh with interpolative blendshapes as defined by an artist. I employ an iterative local optimization technique to solve for blendshape weights given a facial space target. Some examples (transferred onto the 3D model by this method and rendered by professional software) are shown in Figure 3.1 The transfer method can be applied to any facial animation system that is locally controllable in the feature space.

Dataset. In order to learn a model of happiness C and derive a set of semantic classes for $\mathcal{S}$, I turn to the dataset of annotated facial movements from Chapter 2. This dataset consists of results from a large-scale user study at a state-wide fair. Participants were shown expressions from a sweep of anatomically plausible mouth movements on a tablet device, and asked to assign a quality score for how well the face portrayed a smile. Over 900 subjects participated in the survey, providing over 10,000 responses in total. The stimuli contained mostly smile-like faces, but also had some negatively angled mouths, which served as controls. I aggregated the responses to produce a dataset composed of 63 facial expressions annotated with their mean perceived smile quality.

Smile Intensities. I derive $\mathcal{S}$ by defining ranges of quality scores from the dataset as four discrete classes of smiles: *None, Slight, Partial, and Full.* A summary view of the resulting classes are shown in Figure 3.2. An ANOVA test shows high statistical significance with 4 classes, with $[F(3, 60) = 863.5, p < 0.001]$. A post-hoc analysis also confirms statistical significance between all pairs of classes. Figure 3.2b shows similar class sizes.

Experimental Methodology. I compute a (noisy) estimate of precision on the real-world face data via a hold-one-out cross validation loop, computing the mean precision over the folds using held out samples as test data. I also compute a variety estimate within each fold, taking the mean over the folds. Variety estimates are computed on a set of 1000 uniformly sampled points in facial space within the bounding volume of the training data. These samples are passed to the classifier, and the variety is reported as the proportion that are predicted positive. I then validate the results with a follow-up user study.

3.5 Maximizing Variety, Maintaining Precision

Our approach makes use of a binary classifier to identify faces that match the targeted semantic class. While traditional binary classifiers seek to maximize predictive accuracy, maximizing this alone fails to highlight the important trade-off between the precision of the classifier and the diversity of faces that will be generated. Because my model only generates positively classified faces, false positives will be discarded, unseen by any user. As a result, for many animation contexts maximizing precision is the most important goal; the quality of the faces generated by my method is unaffected by false negatives.

However, I would like to support the generation of a large variety of positively

classified faces (e.g., many faces that look happy in different ways). In every classification task there is a fundamental trade-off between precision and variety; maximizing one comes at the cost of the other. Consider the positive decision region of the feature space on which my definition of variety depends. As this region grows larger, the classifier has an increased risk of producing false positives due to encroaching on regions that contain true negatives. Below, I explore this trade-off within the context of my facial generation system, and then present a learning heuristic method that exposes this trade-off to allow us to maximize variety in positively classified faces while retaining as much precision as possible.

3.5.1 Precision Variety Learning

The key insight which enables my approach is that high precision can be ensured by carefully selecting which positive samples are allowed in to the training set. For example, choosing to only include positive training samples that are far away from negative samples can increase the precision of the model at the cost of false negatives, which is a favorable trade given our goals. However, including too few positive training samples results in very little variety, which is an equally important objective. Varying the positive samples allowed into the training set exposes this trade-off for tuning between precision and variety.

To that end, I introduce a parameter m that controls what samples are used in the training set for a binary classifier (e.g., KNN). The training set is constructed by a heuristic ordering of the positive training set by *sample precision*. To define sample precision, I examine at the subregion of the positive decision region that is added by a sample given an existing classifier. Sample precision is taken to be the proportion of this new region that overlaps the true positive region of the feature space. Figure 3.4 illustrates the regions involved and how they are used. Importantly, sample precision considers only the *additional* positive decision area supported by the new training

Algorithm 3.1: PVL Prediction

Input : $sample, trainData, pClass, m$
Output: $prediction$
$pos \leftarrow$ getPositiveSamples$(trainData, pClass)$;
$neg \leftarrow$ getNegativeSamples$(trainData, pClass)$;
$pos \leftarrow$ sortByDistanceToNearest(pos, neg);
$pos \leftarrow$ getfirstNSamples$(positive, m)$;
$trainData \leftarrow$ union$(positive, negative)$;
$prediction \leftarrow$ getPrediction$(trainData, sample)$;
return$(prediction)$;

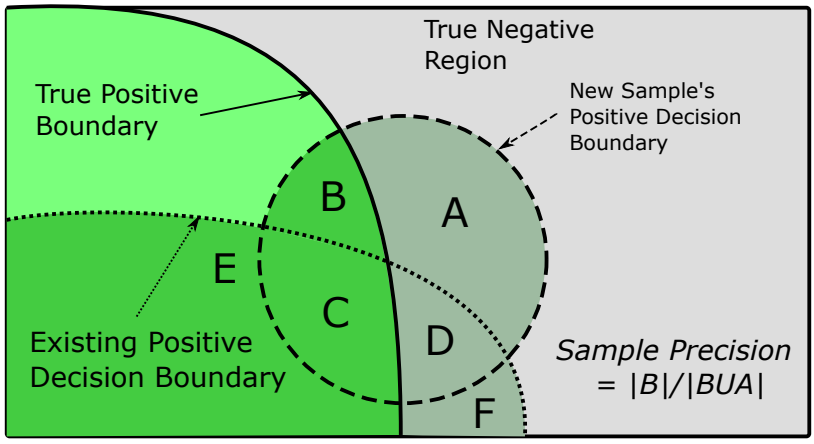


Figure 3.4: **Sample Precision.** Conceptual regions when adding a positive sample into the training set are depicted and labeled. I define sample precision as the ratio of area **B** to area **A**. The precision of the existing classifier is the ratio $|\mathbf{C} \cup \mathbf{E}|/|\mathbf{C} \cup \mathbf{E} \cup \mathbf{D} \cup \mathbf{F}|$, and the precision of the resulting classifier is $|\mathbf{B} \cup \mathbf{C} \cup \mathbf{E}|/|\mathbf{B} \cup \mathbf{C} \cup \mathbf{E} \cup \mathbf{A} \cup \mathbf{C} \cup \mathbf{F}|$

sample. When samples are arranged such that sample precision is decreasing, I define this as the *precision-optimal* order. The first m positive training samples (i.e., with the m highest sample precisions), together with all of negative training samples are provided as input the the binary classifier. For all m, all negative training samples are included as they do not increase the risk of generating a false positive. I present the resulting approach as *Precision-Variety Learning* (PVL), which is detailed in Algorithm 3.1.

Unfortunately, a positive training sample's sample precision cannot be computed

directly as it depends on the ordering of the points added to the classifier before it. We therefore propose an order-independent estimation of sample precision as the distance of a given positive training sample to its nearest negative neighbors. Intuitively, this heuristic captures the fact that false positives (which reduce precision) are likely to lie near negative training samples. This assumption is explored further in the following section.

The key feature of m is the way in which it captures and exposes the trade-off between precision and variety. This is my solution to the multi-objective optimization problem posed in Equation 3.1. Like the pareto-fronts used in many solutions to multi-objective problems, m exposes a precision-variety front that can be exploited to gain as much variety as possible for a desired level of precision. While I do not claim that m generates a pareto-optimal front, I can identify conditions that guarantee the monotonicity of the front, which m is designed to produce. A critical property of pareto fronts, monotonicity insures that any loss of one objective does not allow the loss of the other (e.g., giving up precision will either maintain or increase variety). This also enables a directed search for optimizing m given a desired precision or variety.

In the case of a neighbor-based classifier such as KNN with a precision-optimal ordering of positive training samples, the resulting trade-off front is provably monotonic in m under some supporting assumptions. By monotonicity, I mean for increasing m, variety does not decrease and precision does not increase, and vice versa for decreasing m. To prove this, it is sufficient to show that as m increases, there must be non-increasing precision and non-decreasing variety. The formal arguments for each are as follows.

3.5.2 Proof of Monotonicity

When used with a neighbor-based classifier (such as KNN), there are several key theoretical properties which are maintained by using the PVL approach, which I demonstrate below. The first is that, under certain conditions of the underlying data, the precision of the classifier decreases monotonically as m increases. I also show, regardless of the quality of data, both that specificity (the rate of true negatives) decreases monotonically and variety increases monotonically. Taken together, this means as m increases the predictions will have more inaccuracies (both in terms of admitting false positives and rejecting true negatives), but will increase variety; this serves as the theoretical bases for my claim that PVL is navigating a trade-off between the quality of procedurally generated content and its variety.

Definition 1: Quality of Approximation. I define the *quality of approximation* of my heuristic for a given dataset as the degree to which the distance-based ordering maintains a precision-optimal ordering. The quality of approximation will be high when two conditions hold: 1) the data has a clear positive decision boundary (i.e., samples are more homogeneous the further they are from the boundary), and 2) the boundary has limited curvature. Because my heuristic ordering first adds points that are far away from negative samples, the existence of a clear decision boundary ensures initial points will contribute new positive classification area with higher precision than later points which are closer to the boundary. Assuming limited curvature allows us to safely approximate the distance to the decision boundary as the distance to the single nearest negative sample.

Theorem 1: Decreasing Precision as m increases. Let P_m be the precision of the classifier for arbitrary m and P_{m+1} be the precision of the classifier after including the $(m+1)$th positive training sample. Further let P_s be the sample precision of the

$(m+1)$th sample. Given their respective false positive (FP) and true positive (TP) counts we can compute the precision of the new classifier with $m+1$ samples as:

$$P_{m+1} = \frac{TP_m + TP_s}{TP_m + TP_s + FP_m + FP_s}. \tag{3.5}$$

We therefore need to show that $P_m \geq P_{m+1}$, that is:

$$\frac{TP_m}{TP_m + FP_m} \geq \frac{TP_m + TP_s}{TP_m + TP_s + FP_m + FP_s}, \tag{3.6}$$

which (by cross multiplication) is equivalent to the condition

$$TP_m * FP_s \geq TP_s * FP_m. \tag{3.7}$$

When the quality-of-approximation (*Definition 1*) hold perfectly, we have $P_m \geq P_s$, which implies

$$\frac{TP_m}{TP_m + FP_m} \geq \frac{TP_s}{TP_s + FP_s}$$

$$\iff TP_m * FP_s \geq TP_s * FPm, \tag{3.8}$$

satisfying the requirement of Equation 3.7.

Theorem 2: Decreasing Specificity as m increases. As with precision, maximizing specificity (true negative rate) is important for a classifier that is to be used in the generation of procedural content. I note that specificity and precision can be jointly optimized via the elimination of false positives. Formally, specificity is defined as:

$$TN/(TN + FP), \tag{3.9}$$

where TN represents the true negatives and FP the false positives of a classifier. To show we have decreasing specificity over m, it suffices to observe that increasing m only adds positive training samples to the classifier. As a result, the negative decision region of a neighbor-based classifier cannot increase, and the positive decision region cannot decrease. Thus, false positives are increasing and true negatives decreasing, constraining specificity to decrease. Notably, this property is independent of the order in which the positive samples are added.

Theorem 3: Increasing Variety as m ***increases.*** The supporting argument for increasing variety over m is already established in *Theorem 2*; since adding positive samples constrains the positive decision region to increase, by definition the variety of the classifier will also increase. This property is also independent of the positive samples' order of inclusion.

3.6 Results & Analysis

3.6.1 Behavior of *m*.

To observe the impact of m on classification, I estimate precision and variety over different values of m on a synthetic dataset with a circular ground truth decision boundary. This allows precision to be computed with arbitrary accuracy by sufficiently sampling the feature space and testing them on the classifier. Similarly, variety can be estimated by sampling in the feature space and measuring the positive classification rate. Figure 3.5 shows the results using the KNN classifier: for small m, the precision of the model remains high, but the resulting classifier produces little variety when sampled. Conversely, for large m, a larger variety of points can be generated, at the cost of precision. As the data conditions identified in Section 3.5 hold well for this example dataset, m produces the expected monotonic curve. Thus,

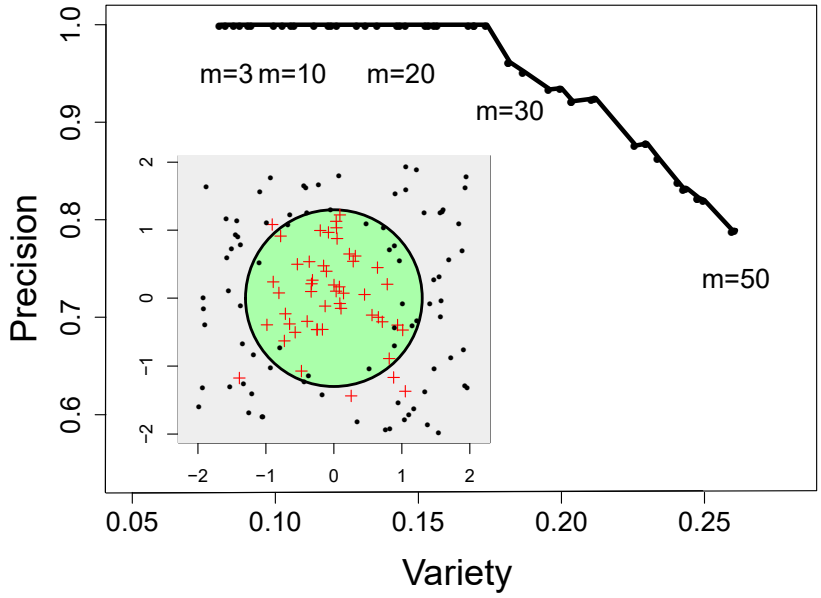


Figure 3.5: Precision-Variety Trade-off curve over m for synthetic circular boundary data.

m allows us to tune the precision/variety trade-off in the learning process.

The curve produced by varying m resembles the ROC curves used to indicate the performance of binary classifiers. Just as ROC curves report the interplay between two conflicting goals of interest (true positive rate and false positive rate), the PVL curves report the performance of a binary classifier in terms of two other conflicting goals relevant to the task at hand.

3.6.2 Comparing Classifiers

As the PVL heuristic supports multiple classification techniques, I compare several algorithms in terms of their precision and specificity over m. Specificity-variety curves for the different classifiers on the real-world data with four classes are shown in Figure 3.6. As in Figure 3.5, each curve exhibits increasing m from left to right, with the exception of the Partial and Slight classes for SVM. The KNN classifier curves exhibit the monotonicity guaranteed for specificity over increasing m.

We also compute curves for Precision, as depicted in Figure 3.7. The limited number of positive samples in the face data cause the uncertainty in estimating precision to be prohibitively large for four classes. To accommodate for this, I construct two classes from the data, and suppress values of m that produce less than 10 positive predictions. In the case of KNN, my method shows a strong monotonic trend, demonstrating its effectiveness on real-world data where my data condition assumptions (see *Definition 1*) do not hold perfectly.

Notably, the SVM and RF algorithms differ from KNN in both the specificity and precision variety curves; the general behavior is similar, but can be erratic for some classes (such as Slight and Partial). While my theoretical guarantees concerning monotonicity do not extend to RFs and SVMs, in practice the curves tend towards monotonicity; SVMs preserve monotonicity when conditions are favorable (and suffer more erratic behavior when conditions are poor), and RFs robustly exhibit a general if not local monotonic trend. Theoretical similarities between RFs and neighbor-based methods have been noted (Lin and Jeon, 2006), which likely contribute to this phenomenon.

3.6.3 Analysis of Faces.

My method is capable of producing a variety of mouth shapes with a targeted smile intensity. Taking advantage of its theoretical properties, I use the KNN based set of classifiers to train C and render the resulting facial animations. Figure 3.1 demonstrates some examples where the *Full* smile class was targeted, with $m = 8$. This m value provides a large gain in variety without a large loss of precision, resulting in faces that differ in appearance, but are all happy. Figure 3.8 shows some examples of training the PVL model to produce faces from other semantic categories. The middle and bottom rows show faces generated from classes Slight and None respectively.

I note that all of the semantic classes exhibited a large amount of variety, though

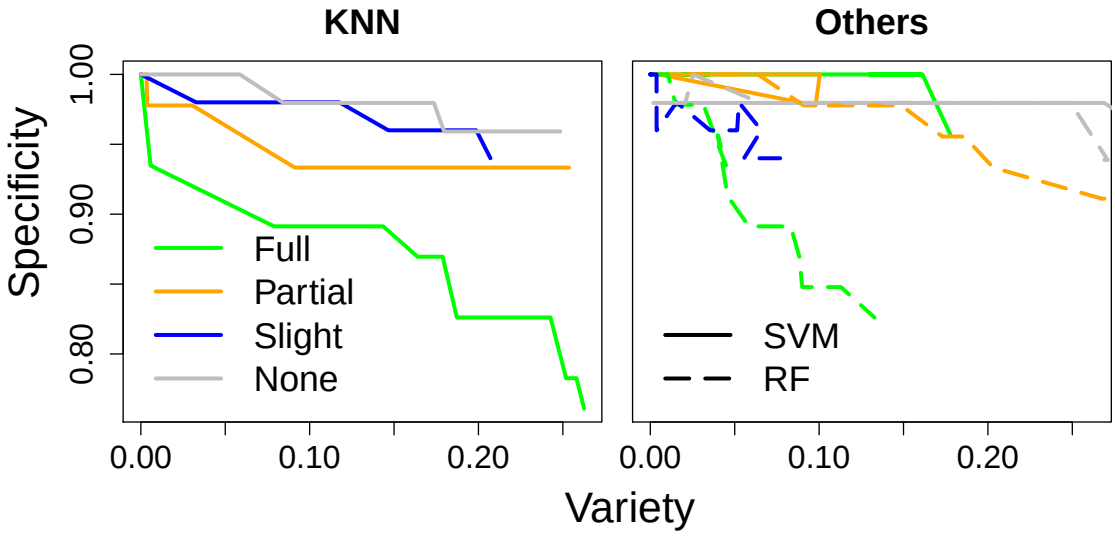


Figure 3.6: **Specificity Curve Comparison.** Specificity curves over m for each semantic class using different supervised learning methods.

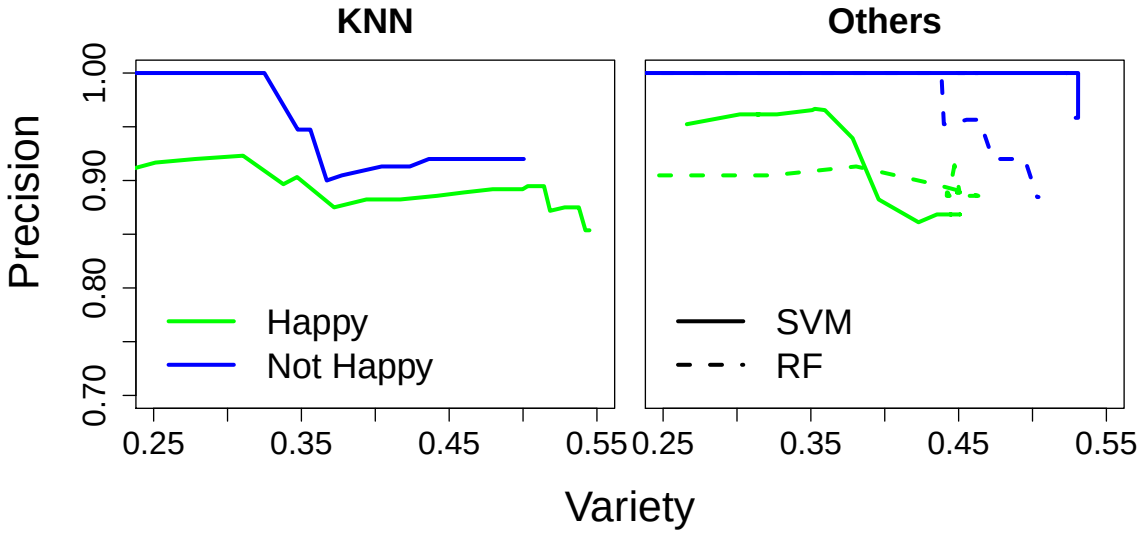


Figure 3.7: **PVL Curve Comparison.** PVL curves over m for a two-class split of the face data using different supervised learning methods.

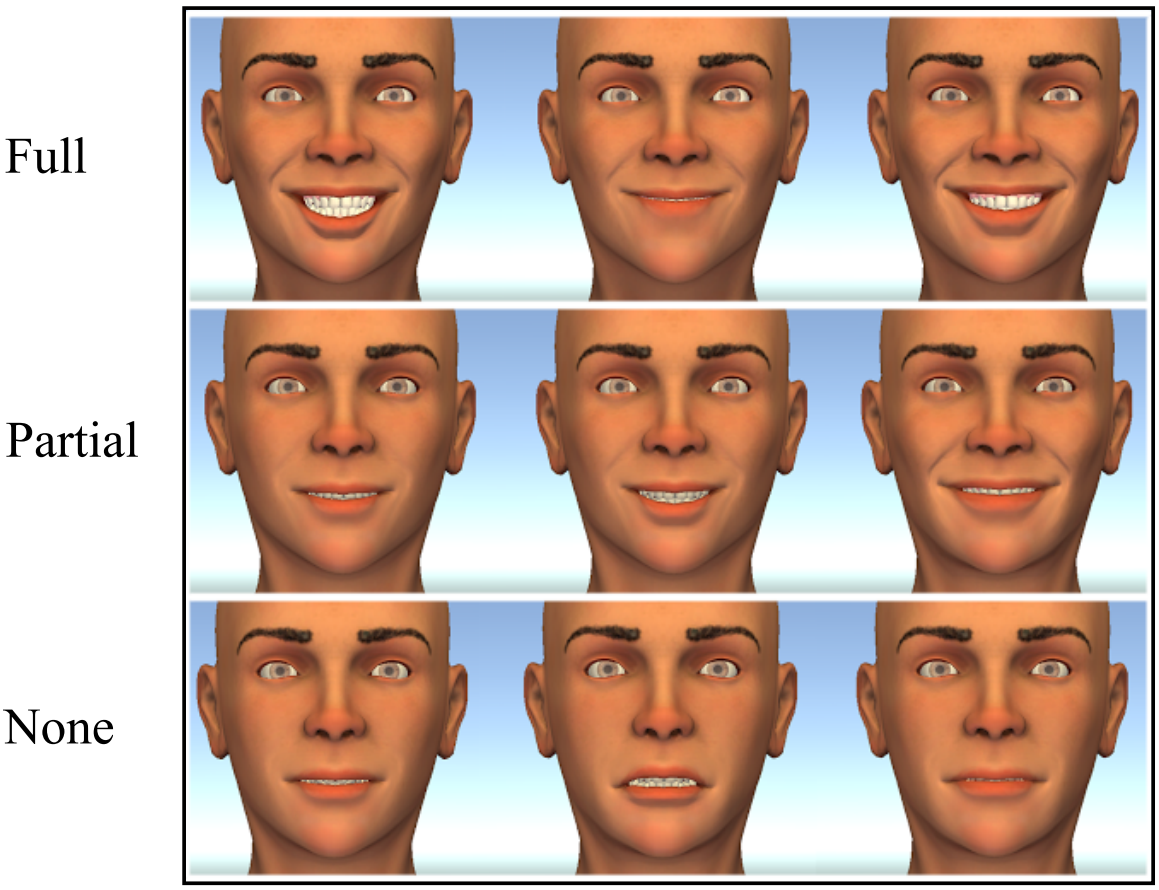


Figure 3.8: Example set of generated faces with Full (top row) Partial (middle row) and None (bottom row) targeted happiness levels.

it varies with the range of m (which is bound by the sample counts in Figure 3.2). While there is generally more variety achievable for a given level of precision in the *None* class (as a smile is just one of many kinds of facial expressions), there is still significant variety present for *Full* smiles, including those that have no dental show (as in the top center of Figure 3.8). This highlights the fact that no single feature was responsible for the semantic meaning of the expression.

3.7 Validation Study.

To validate my method's effectiveness, I conducted a two-part user study to examine the capability of m to support variety, as well as the ability to target different semantic classes.

3.7.1 Study Design

The first section of the study analyzed the ability of the PVL learning approach to tune variety. Here, side-by-side videos showing expressions of a pair of faces were shown to participants via a web browser interface (see Figure 3.10 for an example screenshot). The videos could be individually replayed as desired. Below the videos, participants were asked to respond using a web form the extent to which the expressions looked similar, on a discrete scale of 1 (not at all similar) to 5 (very similar). For each pair of expressions, both were independently, randomly drawn (without replacement both within the pair and across pairs) from a pre-defined set of expressions generated using the same m. Participants were asked to evaluate 10 such pairs of faces, 5 pairs from a set with $m = 3$ and 5 pairs from a set using $m = 8$, but could choose to terminate the study at any time. Both expression sets were generated targeting the *Full* smile class.

The second section of the study aimed at validating the predictive accuracy of

PVL. Similar to the first section of the study, participants were shown side-by-side videos in a web browser of two facial expressions with a response form underneath (Figure 3.11). To test the ability of Facial Space and PVL to extend to other 3D models, my collaborators and I created and used in this section a second 3D facial model. The response prompts were a two-alternative forced choice format where participants were asked to indicate which of two smiles appeared happier. In each pair of expressions, one came from a set of smiles generated from a classifier targeting the *Partial* smile class and one targeting the *Full* class (both independently, randomly drawn without replacement). Both sets of smiles were generated with $m = 8$, to support a variety of different smiles. As in the first section, participants were asked to evaluate 10 pairs, but could exit at any time.

3.7.2 Participant Information

The study saw a total of 17 participants (12 female, 5 male, aged 28.1 $\pm$ 5.5 years). All study participants were proficient in written and verbal English as required to complete the study. Participants were recruited by word-of-mouth, consisting primarily of students from computer science labs. Most participants completed the full study, yielding 170 responses for the first section of the study and 141 responses for the second.

3.7.3 Study Results

The overall result from the first section of the study is shown in the left of Figure 3.9. The hypothesis was that pairs from sets with lower m would be perceived as more similar (have less variety) than those from the set generated with higher m. A Wilcoxon signed rank test confirms ($\mathbf{Z} = -5.07$, $\mathbf{p} < 0.0001$) that comparisons between two expressions from $m = 3$ were perceived as more similar ($\text{Mdn} = 4, \text{IQR} = 2$) than

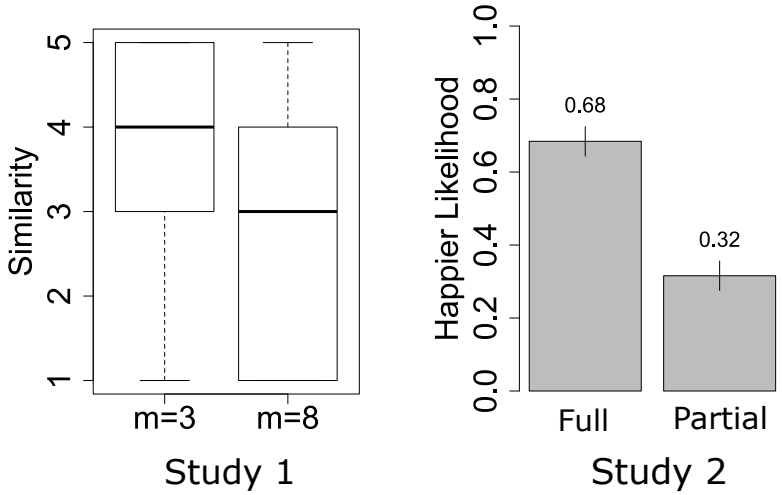


Figure 3.9: **Validation Study Results.** *left*: The results from the first part of the study, which show that expressions generated using smaller m are perceived as more similar than those generated with larger m. *right*: The results from the second part of the study, which show that smiles generated targeting the *Full* smile class appear happier than those generated targeting the *Partial* smile class.

those generated with m = 8 (Mdn = $3, \text{IQR} = 3$). This affirms the hypothesis and serves as evidence to validate the PVL learning approach in its ability to tune variety.

The hypothesis for the second section of the study was that expressions generated targeting the *Full* smile class would be perceived as happier than those generated targeting the *Partial* smile class. The results from this part of the study are shown in the right of Figure 3.9. A two-sided binomial test confirms that smiles predicted as *Full* were most likely to be seen as happier ($\mathbf{p} < 0.001$, $P(\text{success}) = 0.68$, $RR = 1.26$), validating PVL's ability to generate faces with different smile intensities. This result is particularly strong evidence as PVL's task here (producing *Full* smiles distinguishable from *Partial* smiles) is more difficult as the two classes being compared require more subtle differences to be captured as opposed to comparing *Full* vs *None*.

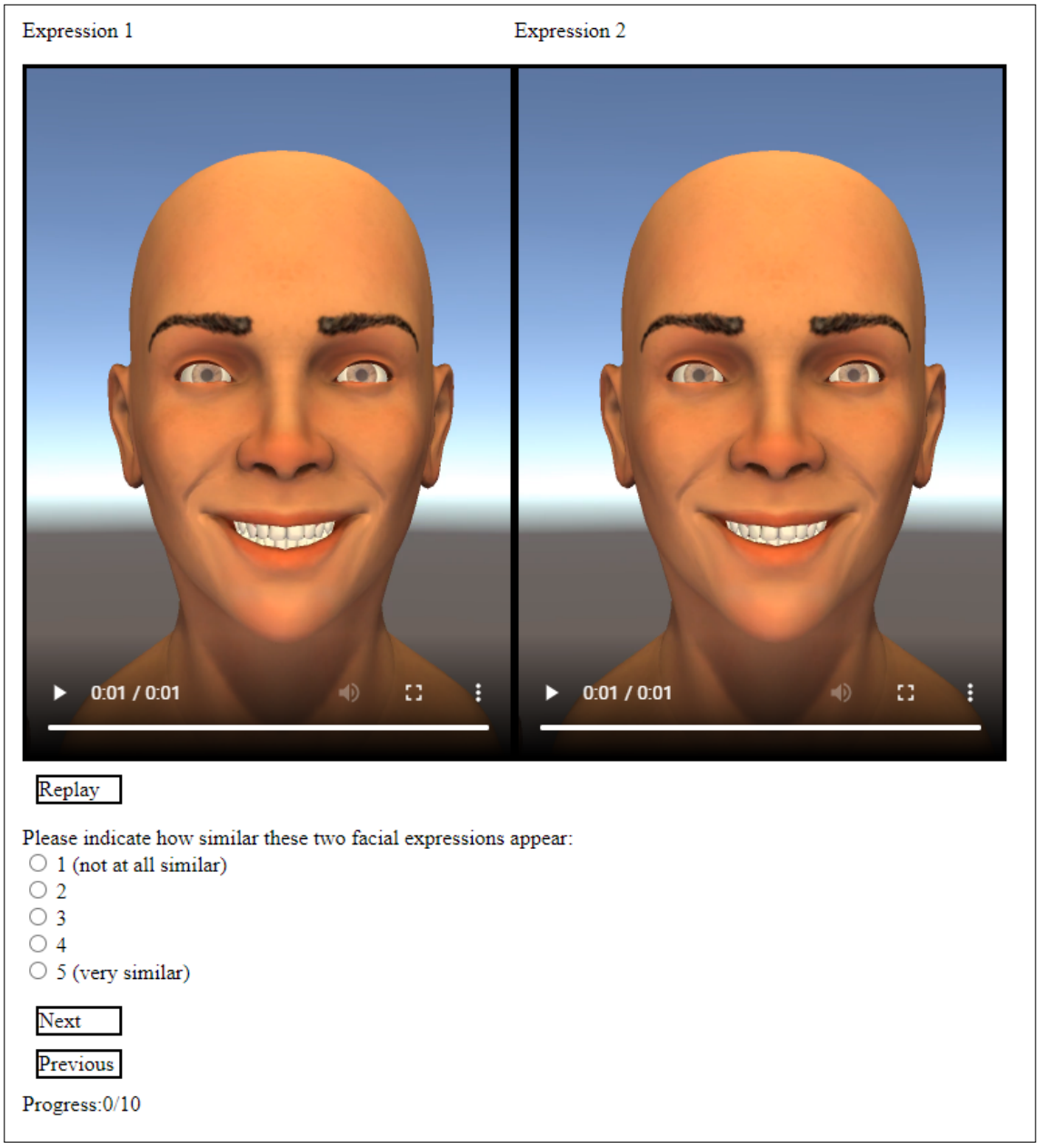


Figure 3.10: A screenshot of the first part of the validation user study.

Figure 3.11: A screenshot of the second part of the validation user study, featuring a different virtual character.

3.8 Conclusion

In this work, I have proposed and implemented a system for the generation of a variety of smiles for use in digital characters. I formulated the problem as a multi-objective optimization task, seeking both high quality and diverse animations. My approach to generate new animations utilized a dataset of annotated facial expressions as training samples for a binary classifier to predict whether or not a new facial expression would be perceived as having a targeted semantic class. To solve the multi-objective problem, I introduced *Precision-Variety Learning*, which allows a balance between precision and variety of a classifier to be directed by manipulating the training data set, providing theoretical guarantees under certain conditions. The classifier was then used to generate a variety of faces with targeted smile intensities novel to the existing data.

Technical Limitations. Some limitations of my method motivate further study. The dataset I used is limited in terms of its coverage of plausible facial positions. Data covering a larger range could enable the study of a more diverse set of emotions. Another limitation is the fact that the blendshapes used for animation have a limited extent, thereby necessitating constrained optimization. This could be relaxed by allowing blendshapes to be extrapolated past their original bounds. I also note that increasing the coverage and density of the annotated faces could allow for more granular categories or regression classifiers to be trained. This could support more fine-tuned control over the target emotions or the generation of mixtures of emotions. While empirically the PVL approach performs very well, the theoretical properties are dependent on some data assumptions that may not hold in real-world settings. Further analysis may identify guarantees that hold when these assumptions are relaxed.

Toward Equity & Diversity in AI. With the rapid growth an incorporation of AI and the associated algorithms comes the increasing impact it has on everyday life to an increasing number of people. To ensure a world where everyone is valued, included and represented, it critical that AI researchers and implementers explicitly seek out and address bias wherever it may arise. In particular, visual appearance and ethnic, gender, and racial diversity is an important part of variety when discussing humans. This important fact is what lead my colleagues and I to include multiple races in our follow up study, shown in Figure 3.11 (the dataset we used in these experiments was collected using the model in Figure 6.2). Our initial results in this area show that the low-dimensional feature space used in our examples transferred well between the two racially distinct examples. However, this issue requires much deeper exploration, and a proper treatment would involve a larger initial study for data collection on a wider variety of faces.

Future Work. Avenues for future work include extensions to my work both in the areas of facial animation and the uses and properties of PVL. One such avenue is the generation of other emotions and mixtures of emotions by incorporating additional datasets and facial features. Building data-driven models that capture how perceived emotional intent relates to facial movement has implications beyond making compelling digital characters. Since computational techniques allow us to permute facial positions in a way that human actors cannot, another exciting area of future work is to investigate faces with large asymmetry or other issues which may arise from facial trauma or nervous system damage. This can allow this work to inform areas of medicine such as facial reconstructive surgery, emotional recognition therapy, and psychologists looking to quantitatively study how intervention can help patients express emotional intent.

While this work utilized data-driven methods for the generation of realistic hu-

man smiles, it does not use explicitly human path data, such as capturing mouth movements from real smiling humans (though I did collaborate with colleagues on a subsequent capture and analysis of just this kind of data detailed in Dong (2019)). The next chapter of this thesis focuses on my work that leverages actual human path data.

Chapter 4

Validating Collision Avoidance as a Realism Technique

Expanding on the previous chapter's focus on generating realistic local human motion, in this chapter I consider the capture and use of real human paths in a validation effort for the importance of collision avoidance for virtual crowds. Through the use of both the captured paths and qualitative follow up surveys, I conduct a user study to observe, for the first time, how the presence or lack of anticipatory collision avoidance techniques impacts user experiences in an immersive virtual reality environment. Following is a detailed report of the user study design as well as the data captured and an analysis that shows how crucial these methods are in maintaining a sense of realism and presence when interacting with crowds in VR. This work is featured in a paper in the 2017 IEEE Workshop on Virtual Humans and Crowds for Immersive Environments (VHCIE). (Sohre et al., 2017).

Dynamic, moving characters are increasingly a part of interactive virtual experiences enabled by immersive display technologies such as head-mounted displays (HMDs). In this new context, it is important to consider the impact their behavior has on user experiences. Here, I explore the role collision avoidance between virtual agents and the VR user plays on overall comfort and perceptual experience in an

immersive virtual environment. Several users participated in an experiment where they were asked to walk through a dense stream of virtual agents who may or may not be using collision avoidance techniques to avoid them. When collision avoidance was used, participants took more direct paths with less jittering or backtracking, and found the resulting simulated motion to be less intimidating, more realistic, and more comfortable.

4.1 Introduction

Advancements in virtual reality via technology (VR), particularly head-mounted displays (HMDs) have led to increased capability and availability of virtual experiences. While the concept of virtual reality is far from new, the field has seen recent and rapid growth in industry and research. Experiencing an immersive environment in VR increases the importance of certain perceptual elements as compared with other virtual experiences such as that provided by PCs. Here, I study the effect of collision avoidance for virtual characters by observing how the presence or absence of this behavior changes a user's experience when interacting with a virtual crowd.

Collision avoidance is one of the primary ways virtual character behavior supports the presence of the user experiencing a virtual environment. There are a variety of approaches for achieving collision avoidance, enabling virtual characters to maintain a minimum distance between both users and other virtual characters in the environment. More recent methods for collision avoidance incorporate more complex strategies that exhibit anticipatory behavior and more human-like trajectories, as well as robust handling of dense scenarios (Karamouzas et al., 2014). In PC-based experiences, collision avoidance is important to make virtual characters act realistically, but in VR collision avoidance takes on a new importance. Characters that don't avoid collisions may cause the user to lose their sense of presence, feel various

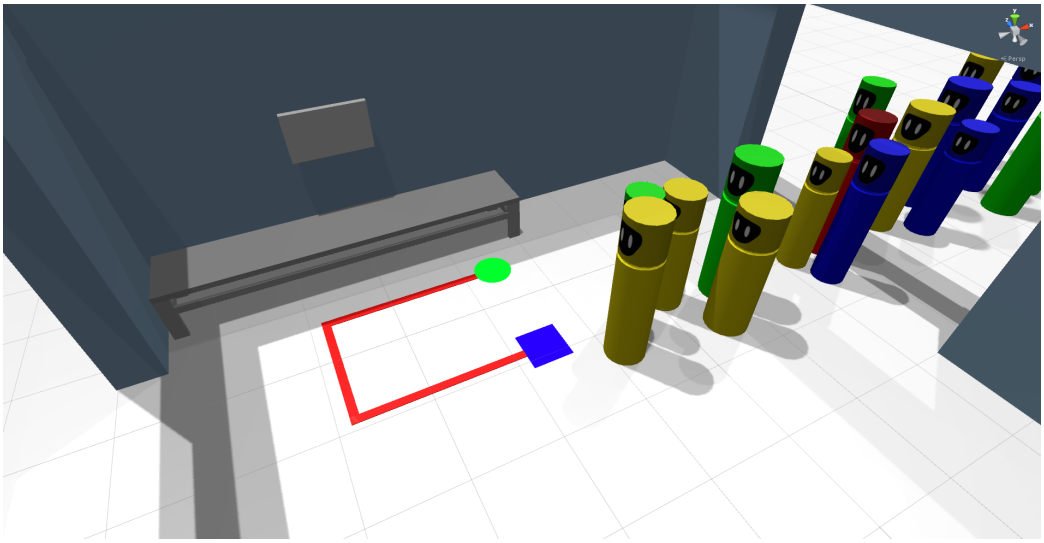


Figure 4.1: **Experimental Setup.** Participants are placed in the above virtual environment and asked to walk along the U-shaped track as a stream of virtual agents walk by. Two conditions are used: one where the agents avoid the users and one were they do not.

forms of discomfort, and intimidate or otherwise negatively impact the experience of the user.

In this paper, I investigate the connection between collision avoidance and the quality of experience for the user such as overall comfort and sense of presence in the virtual environment. To do this, I conduct user studies in which participants interact with a crowd of virtual agents, both with and without collision avoidance behavior (Figure 4.1). The effect of the presence or absence of collision avoidance can be seen both through the subjective experience and the physical actions of the subjects in the virtual environment.

4.2 Background

4.2.1 Personal Space

The study of personal space dates back to at least the 1950s with Edward T. Hall's notion of proxemics (Hall et al., 1959), which identifies the region around each person that they identify as uncomfortable for others to enter. More recently, researchers have turned to VR as a tool to study how humans perceive their personal space. For example, Bailenson et al. used immersive VR (HMDs) to study how much interpersonal distance was maintained between participants and virtual humans. They found a positive correlation between magnitude of emotional reaction and magnitude of avoidance behavior in participants interacting with avatars (Bailenson et al., 2003). In later work, Bailenson and colleagues found that the graphical realism of the simulation had little impact on the minimum interpersonal distance users maintained in VR, but they did find a greater hesitancy to closely approach agents that exhibited more realistic head motion behavior (Bailenson et al., 2005).

Measurements. Measuring personal space presents its own challenges. Researchers have analyzed both behavioral measures (how people act) and self-reported measures (how people say they feel) to gauge peoples' social presence and their response to violations of their personal space in immersive virtual environments. This is important because some variables that affect interpersonal avoidance behaviors may not be captured by self-reported measures (Bailenson et al., 2004). Moreover, Pütten et al. found that peoples' subjective assessments of their interaction with a virtual agent were significantly influenced by their own personality traits (Astrid et al., 2010). Beyond influencing peoples' motion trajectories, violations of personal space can affect their physiological responses; for example Llobera et al. found higher skin conductance readings associated with closer approach distances and greater numbers

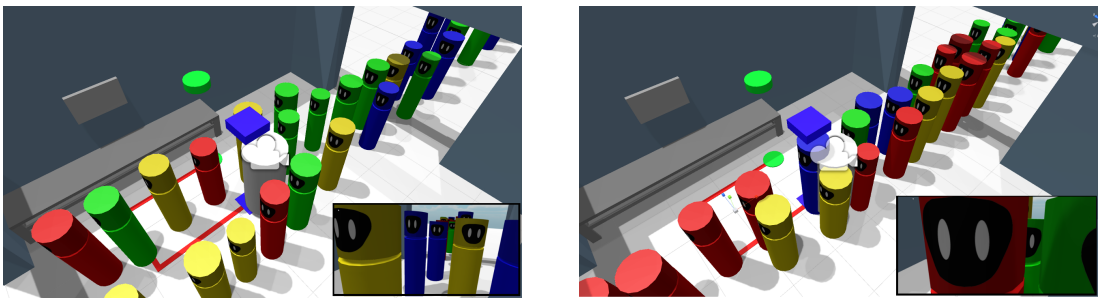
of approaching characters during interactions both with virtual humans and with human-sized cylinders (Llobera et al., 2010).

4.2.2 Interaction with Virtual Crowds

Narang et al. developed a simulation method that robustly generates plausible behaviors for large numbers of virtual humans, including full-body motion and eye gaze as well as motion trajectories. They found a significant impact of the higher fidelity animations on users' ratings of social presence (Narang et al., 2016). Pelechano et al. used navigation tasks in virtual environments to evaluate the sensation of being part of a crowd (Pelechano et al., 2008).

Recently, researchers have explored interactive crowd simulations in immersive environments. Kyriakou et al. found that facilitating collision avoidance increased perceived realism of virtual characters (Kyriakou et al., 2016). Sanz et al. showed that humans use different locomotion behaviors when navigating around human vs non-human virtual obstacles. Bruneau et al. explored user interactions when navigating with groups of virtual humans (Bruneau et al., 2015). While this previous work has considered CAVE-like and semi-immersive environments, my work focuses on users in an HMD-based virtual environment.

Realistic Crowd Simulation. Much of the recent work in crowd simulation has focused on improving the realism in the motion of virtual agents (Kapadia et al., 2015). Other recent work has explored the role anticipation plays in person-to-person interactions (Karamouzas et al., 2014), the role non-linearity has in simulating interactions (Wolinski et al., 2016), vision-based collision avoidance (Ondřej et al., 2010), and physically-based pushing behaviors (Kim et al., 2015). A recent survey providing a wide coverage of the field was recently published by Pelechano et al. (2016)



(a) With Collision Avoidance (b) No Collision Avoidance

Figure 4.2: **Experimental Conditions.** A comparison of the two experimental conditions. Simulated agents either (a) avoided the participant or (b) did not react to their presence. The inset shows first person views. The user is rendered as a white cylinder inside the crowd flow.

4.3 Experiment Design

The goal of my experiment was to induce users to interact with virtual crowds with and without collision avoidance. During the experiments, participants wore an HMD which showed a virtual environment of the same basic shape as the physical lab they were in, with the addition of moving simulated agents. Participant movements were tracked and the virtual environment was updated accordingly.

The experiment consisted of two tasks in which the subject walked along a specified path in the virtual environment (Figure 4.1). After positioning the subject on a starting location in the real world, the HMD was fitted and the virtual environment turned on. The subject would then appear in a virtual room similar to the real one in which they stood. The path was indicated in the virtual environment as a U-shaped line leading from their current position (indicated by a red circle both on the floor and overhead) to the final position (indicated by a blue square). The first leg of the path traveled in an open area, and the second took the subject head-on through a crowd of virtual agents. In this way, traversing the path involved both walking in and outside of a virtual crowd. Both tasks consisted of walking the same path, across which a trial

condition was varied. In one condition (4.2a), the virtual agents performed collision avoidance between themselves and the subject using the Power-Law model proposed in Karamouzas et al. (2014). In the other condition (4.2b), the virtual agents would perform collision avoidance amongst themselves, but not the subject, passing through them as if they were not present. The order of the trial conditions was random for each subject, and counter-balanced across the subjects.

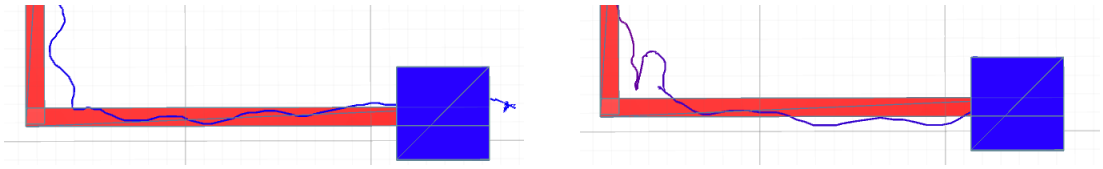
During both tasks, each subject's 3D position and orientation were captured at a sample rate of 10 Hertz. For analysis, the trajectories were cropped to an observation region containing the second leg of the path, where interaction with the virtual crowd occurred. Before, between, and after the trials, participants completed the simulation sickness questionnaire (SSQ) proposed in Kennedy et al. (1993). On completing the study, subjects were asked to complete an additional follow-up survey assessing their overall perception of various aspects of their experience. The survey included items related to the experienced realism of virtual character movement, overall comfort during the simulation, and other subjective measures related to their perception of the virtual characters such as intimidation and reactiveness. Each item was rated on a 1 to 7 discrete scale (See Figure 4.8 for follow-up survey question details).

Physical Set-up. All experiments were conducted in a 3.7 x 2.6m indoor lab area. Position tracking was performed using a 6 camera OptiTrack^{TM} tracking system. The consumer release Oculus VR^{TM} HMD was used for the immersive virtual display. This setup is pictured in Figure 4.3. The Unity Game Engine is used to render the environment. In order to reduce latency induced by fast head rotations, the internal gyroscope on the Oculus was used for head orientation tracking.

Figure 4.3: **Lab Setup.** A participant being tracked as she moves through the physical lab environment reacts to virtual agents in a simulated crowd.

4.3.1 Participant Information

Participants were recruited from computer science labs and courses by word-of-mouth and email invitations. A total of 9 subjects participated in the experiment (3 female, 6 male aged 23.3 $\pm$ 4.9 years). All but one participant had extensive experience with PC or console based video games. All participants had normal or corrected-to-normal vision, and the study personnel ensured each participant could see clearly in the HMD environment. To complete the study, simulator sickness questionnaires, and follow-up surveys, all participants were required to be able to communicate in verbal and written English. For 5 subjects, the virtual crowd used collision avoidance in the first trial, and did not use collision avoidance in the first trial condition for 4 subjects.



(a) With Collision Avoidance (b) No Collision Avoidance

Figure 4.4: **Example Trajectories.** A comparison of the trajectories from two trials of the same user. In the case with no collision avoidance, the user hesitates, backtracks, and ultimately follows a less smooth path.

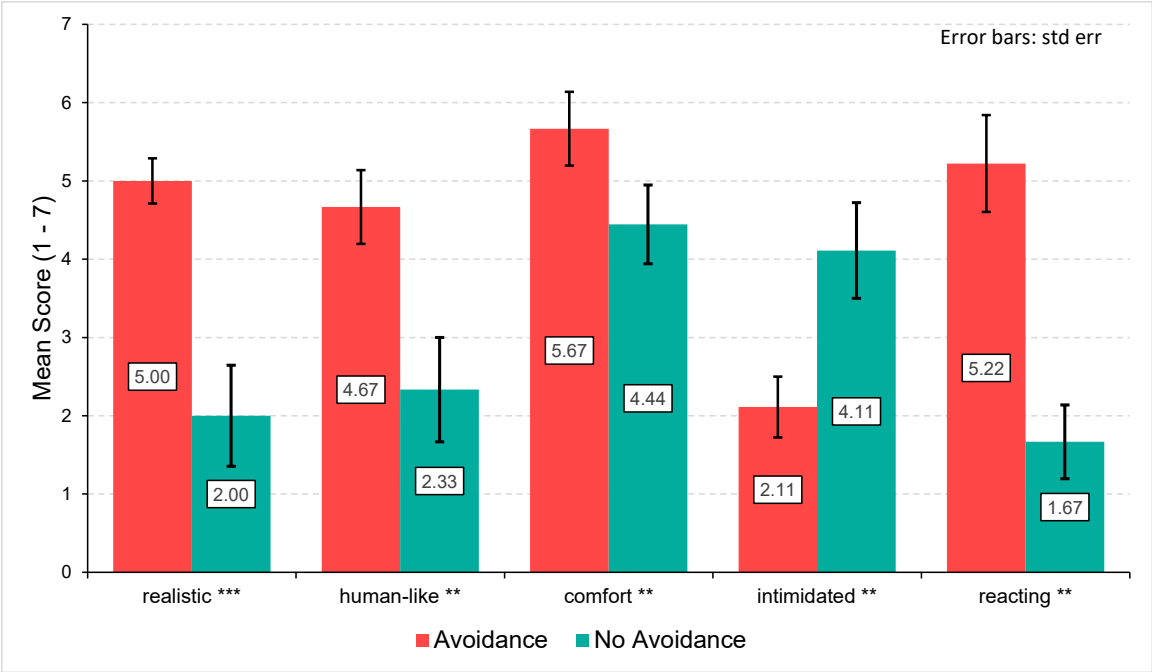


Figure 4.5: **Self-reported Experiences.** Participants evaluated both experimental conditions across several perceptual metrics. Stars indicate level of statistical significance: * for $p < 0.1$, ** for $p < 0.05$, *** for $p < 0.01$.

		Realistic	Human-like	Comfort	Intimidated	Reacting
Avoidance	Mdn	5	4	6	2	6
	IQR	2	1	2	2	1
No Avoidance	Mdn	1	2	4	4	1
	IQR	1	1	1	3	0
p-value		0.006	0.010	0.036	0.012	0.013
Z statistic		-2.769	-2.573	-2.095	-2.507	-2.477

Table 4.1: **Follow-Up Survey Results.** Medians, inner-quartile ranges, and paired Wilcoxon signed-rank test results for follow-up survey measures. In all tests except *Intimidated*, the alternative hypothesis was that the Avoidance value would be greater than No Avoidance. For the *Intimidated* measure, the alternative hypothesis was flipped.

4.4 Results

Both objective and subjective measures showed that the absence or presence of collision avoidance behavior had a significant impact on the subjects' experiences. The responses given in the follow-up survey show strong evidence that participants felt the simulation was affected by the virtual characters' avoidance (or lack thereof). The survey results are depicted in Figure 4.5, with descriptive statistics and statistical test results in Table 4.1. While overall comfort level was higher when avoidance was used, stronger effects emerge when factors related to the motion of the virtual characters are considered. Subjects reported significantly higher perceived reactiveness, lower experienced intimidation, and increased human-likeness of the virtual characters when they exhibited collision avoidance behaviors. Additionally, a very significant increase in perceived realism of character movement was associated with the collision avoidance as well.

Performing the tasks for the experiment had little observable effect on reported physical discomfort levels. The Simluator Sickness Questionnaire asked participants questions to measure the current extent of nausea, ocular-motor, and disorientation discomfort symptoms. The results are shown in Figure 4.6. For the questionnaires

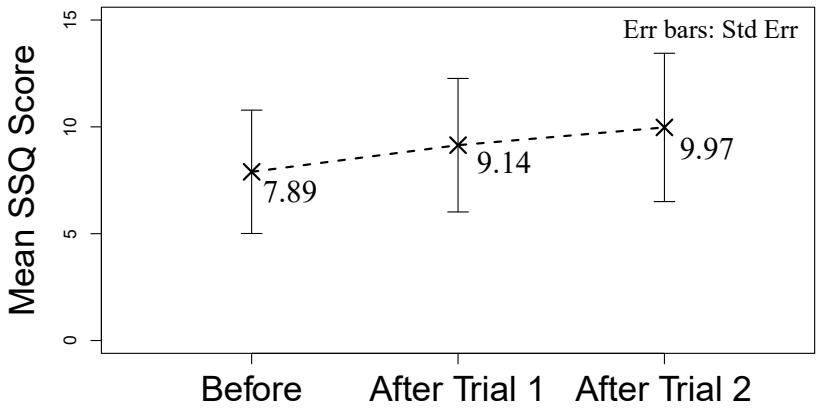


Figure 4.6: **Simulator Sickness Questionnaire Results.** Mean SSQ scores are shown for the questionnaires taken before, between, and after both trials. The vertical bars indicate one standard error above and below the mean. In general, participants experienced very low levels of simulator sickness across the trials.

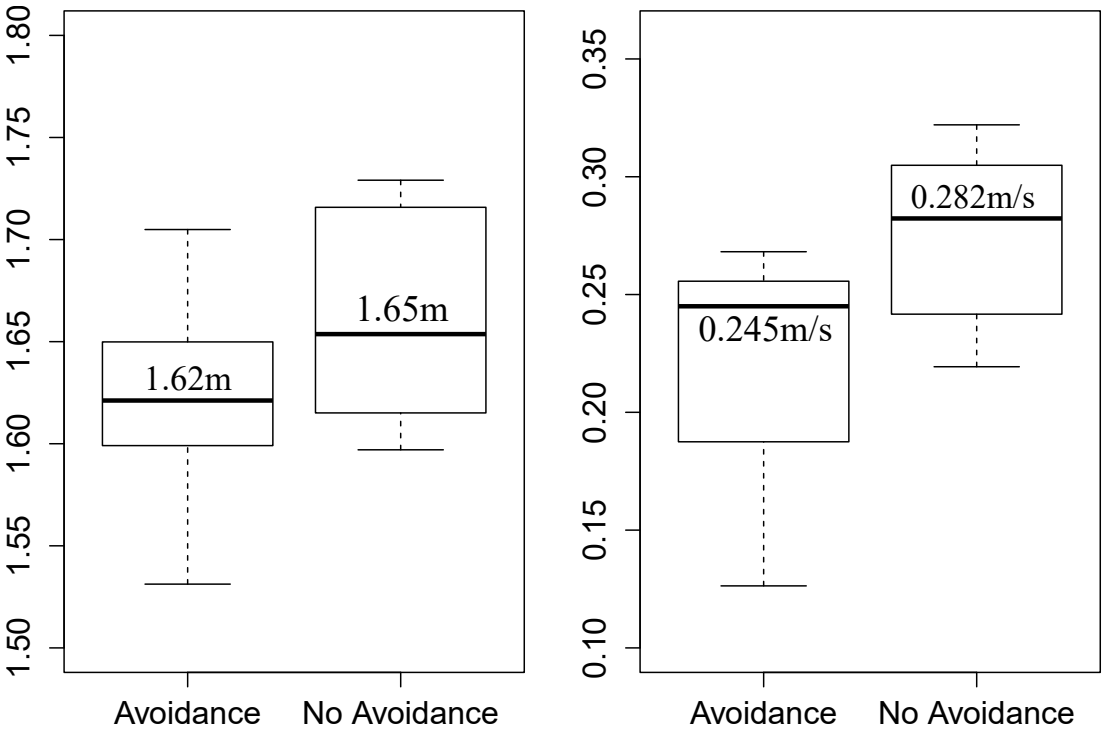
taken before the virtual experience and after the first task, very little change in either discomfort measure was seen. While a small increase in average score was observed from the first to the final SSQ, a one-way ANOVA shows this change is not statistically significant ($F(2, 27) = 0.10, p = 0.89$), suggesting participants did not feel a significant change in their level of discomfort at any point in the study. Possible reasons for lack of symptom levels regularly associated with VR experiences include the short nature of my experiment (the average time spent in VR per trial was 62 seconds), and the participants' backgrounds in other forms of interactive environments and virtual experiences.

The trajectory data captured from the participants motion allows us to perform an objective analysis of behavior displayed in each condition. As participants interacted with the virtual crowd over the different trial conditions, changes in their trajectories could be seen. As a measure of how the discomfort impacted their experience, the path lengths for each trial was computed as the sum of the spatial distances between each sample. These distances were computed in 2D using the 3D coordinate projections onto the ground plane (Figure 4.4). As shown in 4.7a, the trials with collision avoid-

ance saw slightly more efficient (smaller) path lengths ($\text{Mdn} = 1.62\text{m}, \text{IQR} = 0.05\text{m}$) than without avoidance ($\text{Mdn} = 1.65\text{m}, \text{IQR} = 0.10\text{m}$). A paired Wilcoxon signed rank test supports the stability of the effect ($Z = -2.88, p < 0.01$).

While path length only considers the spatial component of the trajectory, there is also important temporal information to consider, such as how often the participant stops or even backtracks. To account for this, I measure the total acceleration taken by each participant over the course of their interaction with the crowd (as measured by the sum of the magnitude of the acceleration at each time step). The results are shown in 4.7b. As expected, when the crowd did not use collision avoidance, participants experienced more acceleration, indicative of less smooth motion with more stopping, backtracking, and veering off-path. As with path lengths, a Wilcoxon signed rank test yields a stable difference between the collision avoidance and non-collision avoidance conditions for the total accelerations ($Z = -2.76, p < 0.05$). However, the effect size is more substantial, with collision avoidance trials showing a 13% smaller total acceleration ($\text{Mdn} = 0.245\text{m/s}, \text{IQR} = 0.068\text{m/s}$) than without ($\text{Mdn} = 0.282\text{m/s}, \text{IQR} = 0.063\text{m/s}$). The larger effect size suggests that a significant aspect of the adverse reactions lies in the temporal component of the motion (e.g., participants experience a jarring reaction due to discomfort with the lack of collision avoidance).

Figure 4.4 shows a single participant whose path illustrates the statistical trends described above. The path traveled in the trial with collision avoidance (Figure 4.4) follows a smooth cadence and has little deviation from the path markers. Without collision avoidance (Figure 4.4b), the path shows a more erratic and irregular shape, straying from the path indicator. This could indicate the subject either trying to avoid the virtual characters (who no longer avoid them), or perhaps having difficulty focusing on the task due to the discomforting (lack of) interaction.



(a) Path Length (m)

(b) Total Acceleration (m/s)

Figure 4.7: **Behavioral Analysis.** Behavioral metrics were computed for the portion of the path that intersected the stream of agents. When collision avoidance was used (a) participants took shorter, more direct paths and (b) less acceleration was experienced. Both metrics indicate less hesitation in walking.

4.5 Conclusion

This experiment provides insight on the impact of collision avoidance behavior for virtual characters on user experiences in an immersive virtual environment. My findings suggest that the presence or absence of collision avoidance has a significant impact on user perception of discomfort, perceived realism and intimidation from virtual characters, as well as the physical actions taken by participants during the study. With high statistical significance, users experienced higher levels of perceived realism, presence, and lower levels of discomfort and intimidation with collision avoidance than without.

Limitations. The limited space of the physical environment constrained the movement based tasks, and resulted in short durations of the VR experiences. The statistical significance for some effects may have been limited by the number of participants in the study, which could be followed up by studies with a larger group of users with a wider range of previous VR and gaming experience. During the trials, many participants appeared to look down while walking in order to better follow the path markings. This has the potential to limit the extent to which the users visually experience the crowd interactions (or lack thereof), which may reduce the effect of the different conditions.

Future Work. Opportunities for future work include experimenting with different modes of user interaction besides navigation, such as giving people virtual hands to interact with the crowd or allowing verbal communication. Additionally, it may be valuable to directly compare the strength of the discomfort felt from lack of collision avoidance in VR to that felt with first-person non-immersive displays (e.g., PC or console games). Lastly, a natural extension of my work is to consider various types of collision avoidance or collision response between the virtual agents and the user.

Indeed, realistic collision avoidance behavior is a crucial component of virtual re-

ality experiences when interacting with humanoid agents. While this work considers validation using captured trajectories of a single human interacting with a simulated crowd, some of my other collaborations use the trajectories of real human crowds themselves to validate the naturalness of various simulation techniques (Karamouzas et al., 2019). The idea that collision avoidance methods are in need of careful consideration for realistic motion is relevant beyond the context of virtual agents. In other works, my colleagues and I consider collision avoidance in the motion planning of real-world robot dynamics (Davis et al., 2018) that can anticipate multiple potential human-like trajectories resulting in more realistic paths, as well as explore human-in-the-loop approaches to robot motion planning (Hutton et al., 2019).

NO_____ **FOLLOW-UP QUESTIONNAIRE**

On the scales below, please circle the number that corresponds best to your overall impression of:

How *realistic* did you find the *motion* of the characters in trial **1**:

very unrealistic — somewhat realistic — very realistic
1 2 3 4 5 6 7

How *realistic* did you find the *motion* of the characters in trial **2**:

very unrealistic — somewhat realistic — very realistic
1 2 3 4 5 6 7

How *human-like* did you find the motion of the characters in trial **1**:

not human-like — somewhat human-like — very human-like
1 2 3 4 5 6 7

How *human-like* did you find the motion of the characters in trial **2**:

not human-like — somewhat human-like — very human-like
1 2 3 4 5 6 7

How often did you feel the need to *close your eyes* during trial **1**:

not at all — a few times — the whole time
1 2 3 4 5 6 7

How often did you feel the need to *close your eyes* during trial **2**:

not at all — a few times — the whole time
1 2 3 4 5 6 7

How *comfortable* did you feel during trial **1**:

very uncomfortable — mild discomfort — very comfortable
1 2 3 4 5 6 7

How *comfortable* did you feel during trial **2**:

very uncomfortable — mild discomfort — very comfortable
1 2 3 4 5 6 7

How *intimidated* by the characters did you feel during trial **1**:

not at all — somewhat intimidated — very intimidated
1 2 3 4 5 6 7

How *intimidated* by the characters did you feel during trial **2**:

not at all — somewhat intimidated — very intimidated
1 2 3 4 5 6 7

The extent to which you felt as if you were *moving* when standing still in trial **1**:

not at all — slight sensation of movement — strong sensation of movement
1 2 3 4 5 6 7

The extent to which you felt as if you were *moving* when standing still in trial **2**:

not at all — slight sensation of movement — strong sensation of movement
1 2 3 4 5 6 7

The extent to which you felt the characters were *reacting to your presence* in trial **1**:

not at all — some reaction — very reactive
1 2 3 4 5 6 7

The extent to which you felt the characters were *reacting to your presence* in trial **2**:

not at all — some reaction — very reactive
1 2 3 4 5 6 7

Figure 4.8: **Follow-Up Survey**

Chapter 5

Data-Driven Insights for Multi-Task Human Navigation Decisions

Humans often face the longer-term planning task of navigating through environments with static and dynamic obstacles. Whether it be terrain, walls in a building, or other people, we regularly route through a wide variety of environments to reach goals that may be far away or out of sight. A natural but challenging obstacle that arises from these tasks is the presence of local minima combined with the absence of global information; sometimes we need to make global plans in an environment for which we do not have a mental or physical map. However, while a map can help us plan optimal routes through large environments, humans are not helpless in such cases – we are capable of forming reasonable conclusions about promising directions of travel without one. Of course, adversarial maze-like environments can be designed to fool us into choosing dead-ends and large local minima, but in the general, we utilize our world knowledge to guide a global path using local decisions based on local information.

In this chapter, I continue to analyze human motion data to provide insights on human motion, this time in the context of how people make decisions in multi-task navigation settings (such as retrieving items from a shopping list). The analysis re-

veals an entropy law that governs the high level trends in the decision making process, and can be reproduced with high accuracy in terms of the high level behavioral trends by a single-parameter simulation model.

A pre-print publication of this work is available on ArXiv at article reference arXiv:2102.00057 (Sohre et al., 2021).

5.1 Introduction

Understanding human flow through indoor buildings is important for various layout design tasks such as evacuation planning, product placement, and security. Advancements in technologies such as computer vision and motion-tracking have enabled the large scale collection of long-term motion data, enabling new analyses aimed at incorporating these high level decisions into human flow models. Here, I take a data-driven approach to analyzing the navigation decisions of shoppers in a grocery store.

Path data, or varying spatial configurations of individuals as a function of time, provides valuable insight into human navigational behavior in a variety contexts (Hui et al., 2009a). Various works have utilized path data both as a means to understand and simulate human behavior, using different assumptions and conceptualizations to achieve specific research goals. Some of these works involve learning to predict human behavior from path data, from inferring high level flow models for retail floor optimization (Ying et al., 2019), to frameworks based on random walks (Gutiérrez-Roig et al., 2016), to crowd simulations (Karamouzas et al., 2014). Other works focus on analysis of behaviors, such as comparing human route selection to the theoretical optimum (Hui et al., 2009b), or discovering behavioral patterns (Larson et al., 2005; Karamouzas et al., 2018). Often these analyses are coupled with various mobility models for predicting human movement, designed to operate on different scales, from global migratory patterns (Riascos and Mateos, 2012) and city level traffic (Ca-

margo et al., 2019; Piovani et al., 2018) to more local path planning (Lima et al., 2016; Bailenson et al., 2000) and collision avoidance (Karamouzas et al., 2014; Van Den Berg et al., 2011; Helbing and Molnar, 1995). Less explored is the mid-level scale of multi-task planning, such as how fair-goers might visit multiple attractions, or how customers determine which item to pick up next from a shopping list.

In this work, I use path data to better understand the process by which human shoppers make decisions about navigation when shopping. I focus on characterizing the task of selecting a next item to retrieve, and perform multiple analyses that provide insights into high level features governing shopper decisions. While numerous external factors affect this process, such as building layout and other human factors (specific in-store attractions, unplanned purchases (Massara et al., 2014), or tendency to follow the perimeter (Farley and Ring, 1966)), I adopt a general formulation of a shopping trip as a series of decisions given a predefined list of items, and utilize a large dataset of shopper trajectories to gain insights about how the navigation process may be described and modeled from the path data.

The rest of this chapter is structured as follows. In Section 5.2, I describe the dataset as well as preprocessing steps and give some formal definitions that support the analysis. In Section 5.3, I model shopping trips as a series of discrete navigation decisions that produces an item sequence, identify several high level trends, and propose a measure of decision difficulty that governs the sub-optimalities in the data when decomposed into pair-wise decision tasks. In Section 5.4.2, I incorporate my findings into a general decision model for each step of a shopping trip. In Section 5.4.3, I propose a stochastic decision model that is theoretically guaranteed to produce the same trends seen in the data, but in practice matches the trends with high accuracy.

5.2 Dataset

Here I use an anonymized dataset consisting of item sequences from shoppers in a retail store (*paths*), corresponding to individual transactions from point-of-sale records of sets of items purchased together (*baskets*). Each sequence reflects the order in which the items were retrieved. The items are embedded in a 2D representation of the store layout corresponding to product shelf locations. Additionally I use the set of 2D wall obstacles representing the sales floor layout to compute features such as walking distance between items. Figure 5.1 shows a contextualized example of a single shopping trip as it would appear in the data. The data contains over 13,000 such basket sequences spanning a period of two weeks.

The item sequences (and corresponding decision points) for each shopping trip represent paths over the fully connected item graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where each vertex $v \in \mathcal{V}$ represents the spatial embedding of an item shelf in the store (I assign the shelf's position to be its corner), and the edges $e_{ij} \in \mathcal{E}$ represent the straight line connections between co-visible vertices v_i and v_j. I augment this graph with an additional vertex v_{start} for the entrance to the store to serve as the shopper location for the first decision point in a trip. I use this graph to compute the *shortest path* walking distance between any two items for analysis.

Given a list of items left to collect $i \in \{1, 2, ...n\}$ I decompose a decision point into the set of available (shortest walking path) travel distances it represents:

$$\mathbf{p} = \{d_1, d_2 ... d_n\} \tag{5.1}$$

where each distance d_i is the length of the shortest path over $\mathcal{G}$ from the item's shelf location to the shopper location (either v_{start} or the vertex of the most recently chosen item's location). I extract for analysis only those $\mathbf{p}$ having more than one item, as having only a single item remaining does not present a choice to the shopper.

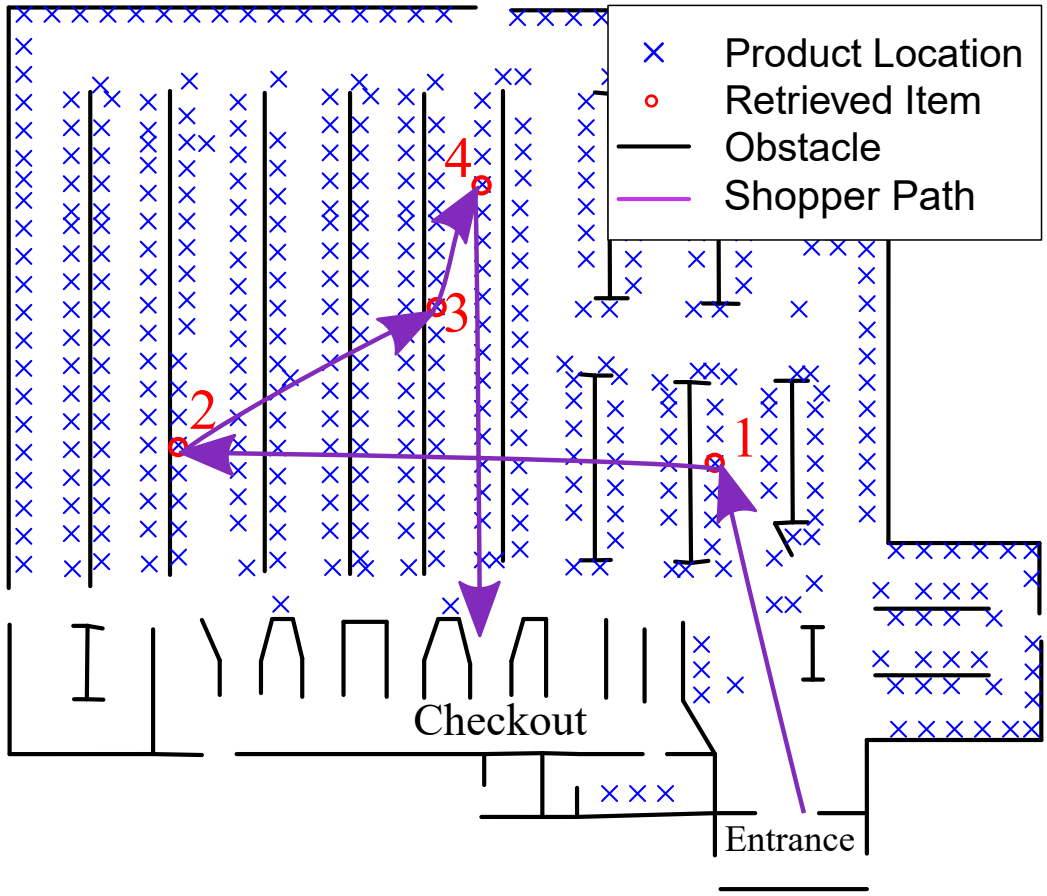


Figure 5.1: An example item sequence from the dataset, embedded in the abstracted store layout (black obstacles) and product shelf embedding (x's).

Figure 5.2 shows an example decision scenario consisting of a choice between three remaining items to be collected. Comparing the observed shopper item sequences to a locally optimal one enables the extraction of where the relative sub-optimalities occur in human paths. These sub-optimalities enable a predictive model of shopper navigation decisions from the distances in a decision point.

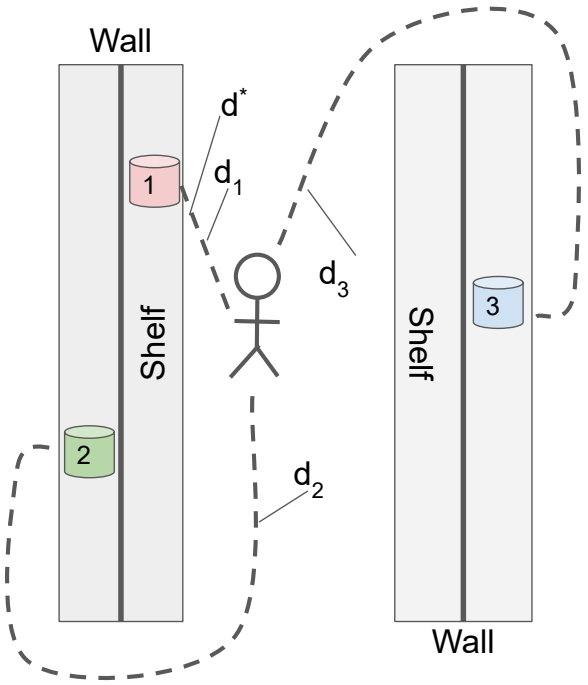


Figure 5.2: An example decision $\mathbf{p} = (p_1, p_2, p_3)$. Distances $d(p_i)$ (the shortest walking path) to each remaining item (cylinders) are depicted by dashed lines.

5.3 Data Analysis

5.3.1 Decisions

The navigation decisions in the data show that shoppers are generally efficient. Despite not necessarily having full knowledge of all item locations or store layout, about 79% of the time, the next item a shopper picks up is the closest item to their current location v_j, having distance $d^* = \min_i(d_i \in \mathbf{p})$. While always picking the next closest item is not a globally optimal strategy (which involves solving a traveling salesperson problem, studied by (Hui et al., 2009b) in the context of shopping), I refer to it as "locally optimal" in the sense that it is the best path a person could take without knowing other future items they have not yet gathered. This suggests the conceptualization of a shopping trip where each decision involves forming an estimate of which

item left to collect is the closest, and navigating there. For the sake of analysis, I denote the next selected item for retrieval as $\hat{i}$, which is the index into a given $\mathbf{p}$ that reflects which item was chosen. Then $d_{\hat{i}}$ represents the distance to the chosen item. Similarly I define i^* such that $d_{i^*} = d^*$ to be the index of the closest item for the decision point.

Due to the local nature of available information (and potentially limited familiarity shoppers may have with the store layout), cases where the closest item is not chosen ($\hat{i} \neq i^*$) naturally arise. The presence of these suboptimal local choices are consistent with other studies of human route selection and cognitive tasks where suboptimalities are found to be a natural part of these processes (Lima et al., 2016; Hui et al., 2009b). A suboptimal choice occurs whenever the chosen item's distance was larger than the optimal distance for that decision point. I call these choices *inversions*, where the preference order of the chosen item with respect to the optimal one has been flipped.

An analysis of the inversions in the set of decision points reveal a strong trend between d^* and the likelihood of making a locally optimal choice (i.e., $\hat{d} = d^*$). Figure 5.3 (left) shows this trend, where the likelihood of choosing the optimal item at a decision point decreases as a function of increasing d^*. As this distance (and necessarily the distance of all other remaining items) increases, it becomes more difficult to consistently choose the closest item, converging toward the same likelihood as a random selection.

5.3.2 Sub-Tasks

In addition to the set of per-decision inversions, I perform a decomposition of the shopping decisions to produce a larger set of inversions for analysis. To do this, I extract all pair-wise comparisons $(\hat{d}, d_i)$ from the data, each of which represents a possible inversion of the chosen item $\hat{d}_i$ with some alternative d_i. This yields a dataset of over $883,000$ such item pairs, which I refer to as *sub-tasks*, in which a

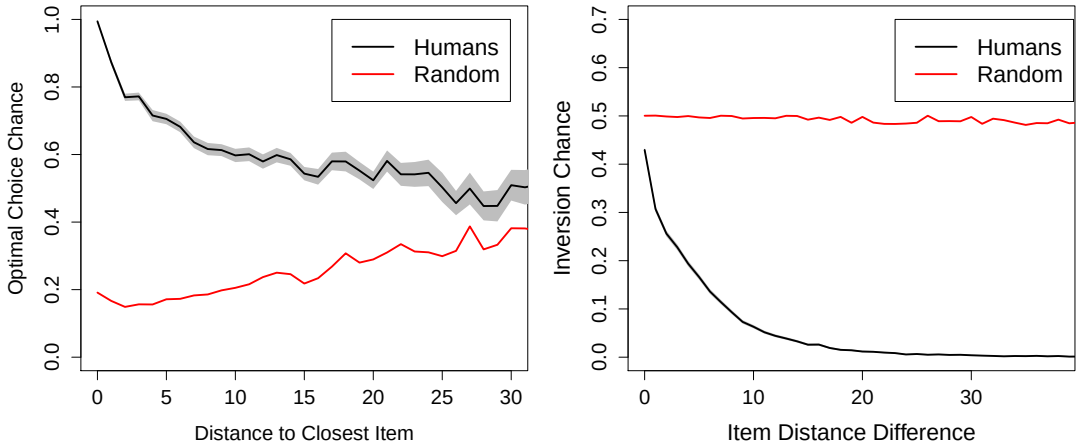


Figure 5.3: *left*: The likelihood of choosing the locally optimal (closest) item decreases as a function of distance to the closest item for shopper paths (grey region is the 98% confidence interval), and eventually converges to that of a random choice. *right*: The likelihood of not choosing the closer of two items in a sub-task as a function of the difference in distances (to the shopper) between them. The random choice is shown for reference (always 0.5 for two items).

shopper estimates the closer of the two. This yields a dataset of over $883,000$ such item pairs, which I refer to as *sub-tasks*, in which a shopper estimates the closer of the two.

The extracted pairs represent a subset of all the possible sub-tasks (and potential pair-wise inversions) that exist in the full decisions. Sub-task samples having $\hat{d} > d_i$ constitute pair-wise inversions, whose inversion amount can be measured as how much farther the shopper traveled then they would have by choosing item i. A sub-task can be alternatively decomposed as the pair (F, C), where $F = \max(\hat{d}, d_i)$ is the larger of the two, and $C = \min(\hat{d}, d_i)$ is the smaller. This enables an inversion analysis based on both the closer item distance C as well as the relative item distance $F - C$ (always a positive value).

The right side of Figure 5.3 shows an analysis of the sub-tasks. Here, the chance of a pair-wise inversion falls off as the items grow farther apart in their relative distances to the shopper. This suggests that as the difference of relative distances

grows, it becomes easier to distinguish which is closer.

5.4 An Entropy Law for Inversion Likelihood

5.4.1 Measuring Difficulty

As is evident from the non-uniformity of both trends examined in Figure 5.3, some decisions and sub-tasks are more likely to see inversions than others. Given the assumption that shoppers desire efficient paths, these trends serve as evidence that some scenarios present more difficult estimation tasks. Here I adopt a description of difficulty that follows from information theory, which is the *entropy* of the sub-task. In the context of pair-wise decisions, the entropy is the minimum number of bits required to represent the item distances such that they can be reliably ordered. Formally, given the distance F of the farther of the two items in a sub-task and the distance C of the closer item, the entropy H can be computed as

$$H = \log_2 \left(\frac{F}{(F - C)} \right). \tag{5.2}$$

We adopt the entropy of a sub-task as a measure of difficulty, and define the difficulty D of a sub-task as:

$$D = \log_2 \left(\frac{F}{(F - C) + \epsilon} \right). \tag{5.3}$$

where $\epsilon = 0.01$ places an upper bound on difficulty at $F = C$. I call $W = F - C$ the *tolerance* of the task, as it is the maximum relative error in distance estimation that preserves their rank order. This difficulty measure is consistent with (and inspired by) those proposed in other psychophysical studies of human cognition (Moyer and

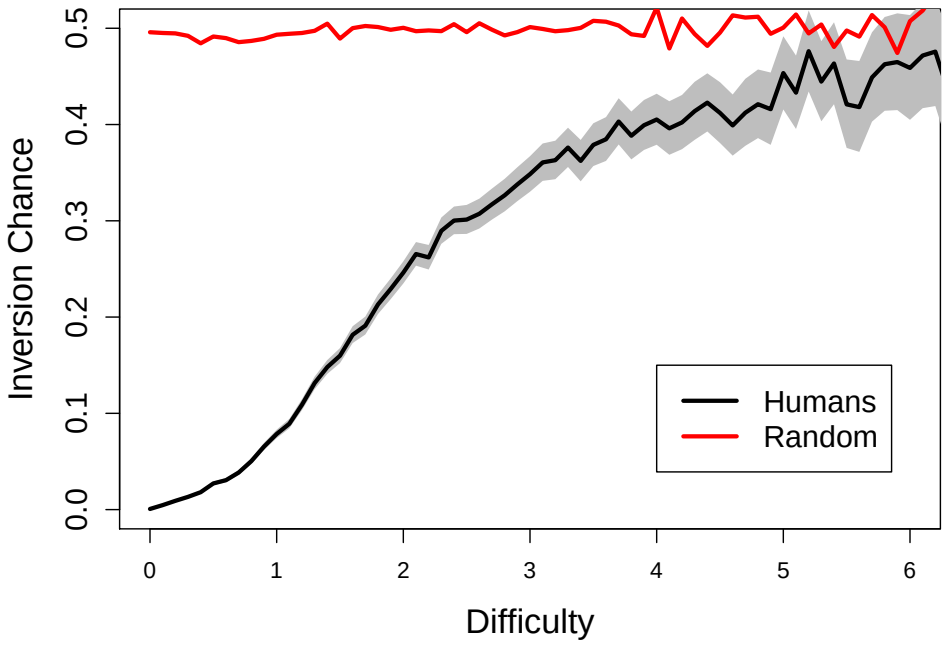


Figure 5.4: The chance of choosing the farther of two items in a sub-task as a function of difficulty (entropy) score for the shopper data (black w/ grey 98% confidence interval) and random choice (red).

Landauer, 1967; Fitts and Peterson, 1964).

When inversion rate is graphed as a function of difficulty (Figure 5.4), there emerges a clear monotonic relationship between difficulty and inversion chance; $P(\hat{d} = F)$ (i.e., choosing a farther than necessary item to go to next) increases as difficulty gets larger (with some noise affecting the trend at higher difficulties that occur less frequently in the data). Additionally, the inversion rates naturally converge with random selections at a maximum of 50%. The ability for shoppers to distinguish between closer and farther items saturates at around $D = 5$, indicating that the information carrying capacity of the item selection process of shoppers was around 5 bits.

This cognitive difficulty not only drives inversion rates, but also captures both empirical trends in Figure 5.3. The closer together in distance from the shopper ($F - C$ is small), the larger D becomes, consistent with Figure 5.3 (right) where there is

greater confusion between item pairs having small distance differences. Additionally, the farther away the closer item is (large F), the greater D becomes, matching the left side decision level trend showing lower chance of choosing optimally.

5.4.2 Independent Perceptual Error Model

To extend the scope of the analysis in Section 5.4.1 to multi-item decisions, I note that the empirical trends for misordering items are consistent with a selection process that involves independent assessments of perceived item distances. I introduce $\tilde{d}_i$ as a noisy, estimated distance to an item that incorporates uncertainty into a shopper's decision. The selection process can then be modeled as forming estimates for each item, then choosing the item $i = \text{argmin}_i\{\tilde{d}_i \in \mathbf{p}\}$ that is estimated to be closest. In light of the analysis from Section 5.3, I note that the uncertainty of an item's estimated distance should grow with the true distance, and the ability to discriminate between them should diminish with close relative distances to the shopper. To meet these criteria, I design a generative model of noisy item distance estimation and model the noisy estimated distance, $\tilde{d}_i$, as follows:

$$\tilde{d}_i = d_i + \epsilon_i,\ \epsilon_i \sim \mathcal{N}(0, \alpha d_i) \tag{5.4}$$

where the standard deviation αd_i is a linear function of the item's true distance, and each item's noise w_i is sampled independently. For a sub-task, the chance of inversion can be computed directly from the Gaussian noise model:

$$\begin{aligned} P(\hat{d} > d_i) &= P(\hat{d} = F \mid F, C) \\ &= \frac{1}{2}\left[1 + \text{erf}\left(\frac{F - C}{\alpha\sqrt{2(C^2 + F^2)}}\right)\right] \end{aligned} \tag{5.5}$$

where F and C represent the farther and closer distances of the sub-task $(\hat{d}, d_i)$ respectively.

Section 5.6 provides a derivation of Equation 5.5. This analytical expression for the inversion chance both supports theoretical guarantees for the model and is efficient to compute, making it practical to directly fit α to the inversion chances in the data via numerical optimization techniques. Here I fit α to the trend shown in Figure 5.3 (*right*), as it has good data support over the entire domain (i.e., very tight confidence bounds across the x axis). Using BFGS gradient descent optimization, with a mean-squared error loss, yields $\alpha = 0.30$ as a minimizing value. The optimization was performed using the *optim* function of the *stats* package for statistical computing language R (R Core Team, 2020).

This noisy distance estimation model has key theoretical properties that guarantee inversion likelihoods are congruent with the trends seen in Figure 5.3 and Figure 5.4, independent of the choice of α. First, the distribution for choosing between N items converges to uniform as the relative distances become closer in magnitude (this can be seem by taking the limit as F approaches C in from Equation 5.5). Second, the chance of pair-wise inversions monotonically approaches the asymptote of 0.5 both as relative item-agent distances decrease and as distance to the closest item increases. Finally, my model provably recovers the monotonic relationship between difficulty and chance of inversion (see Section 5.6 for a proof).

5.4.3 Simulation Method

The generative decision model from Section 5.4.2 is easily incorporated into a simulation for predicting a shopping trip given a basket of items to be collected, a spatial embedding of the items, and obstacles representing a store layout. I propose a stochastic agent-based simulation that does this with an execution strategy as follows. The agent begins at v_{start} and has available a list of all (d_i, v_i) tuples corresponding to

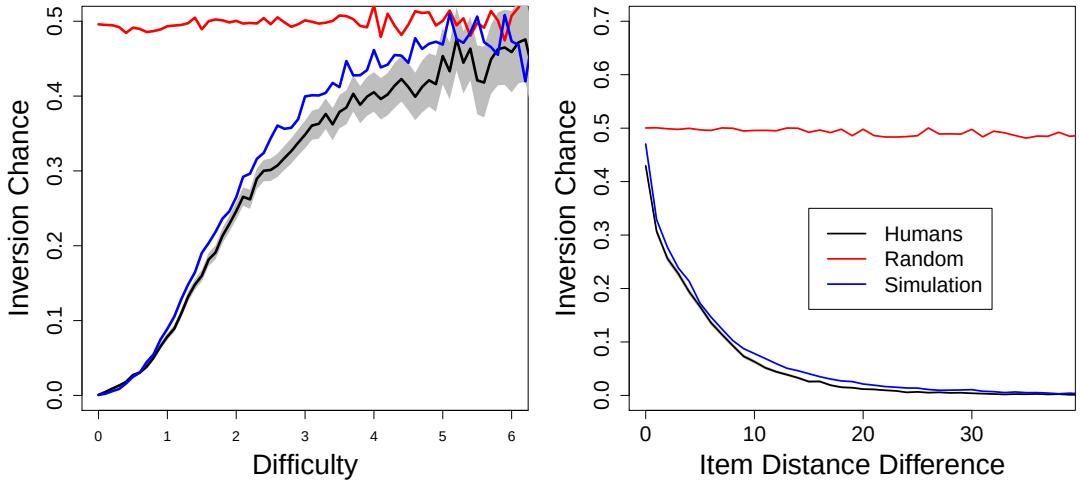


Figure 5.5: *left*: the difficulty plot from Figure 5.4 with overlaid simulation inversion rate for the independent distance estimation model. *right*: observed inversion rates from simulated paths overlaid on the trend from Figure 5.3 (right).

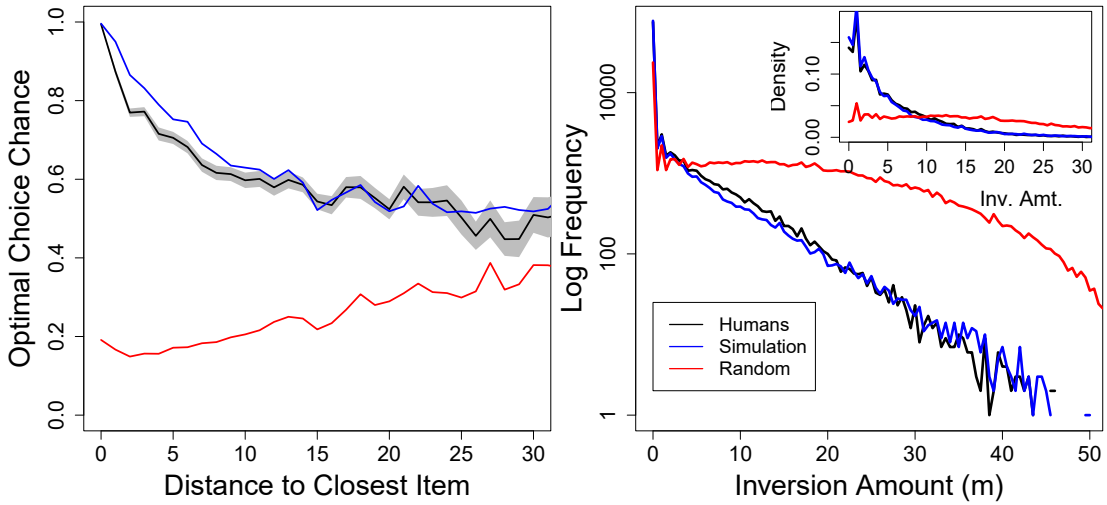


Figure 5.6: *left*: observed optimal choice rates from simulated paths overlaid on the same data from Figure 5.3 (left). *right*: The log frequency of observed inversion sizes (i.e. the extraneous distance travelled to the chosen item over the closest) for simulated, shopper, and random item decisions. Inset is the density for inversion sizes cropped to show the knee of the curve.

where each item in the list can be picked up in the store. Noise is independently sampled according to Equation 5.4 to produce a $\tilde{d}_i$ for each candidate (v_i), and the agent selects the one having the smallest estimated distance as the next navigation target. The agent then navigates to the chosen item location using a shortest path planner over a visibility graph of the item locations. This strategy is repeated until all items have been collected (see Algorithm 5.1 for details). Since the distance estimates are stochastic, the simulation method can be used to generate a variety of plausible shopping routes for a single basket.

Algorithm 5.1: Shopping Trip Simulation

```
Input: itemsToRetrieve;
Output: itemOrder;
itemOrder = [];
alpha = 0.315;
while While itemsToRetrieve.length < 1 do
    distances = [];
    for i in 0 : itemsToRetrieve.length-1 do
        trueDistance = getGeodesicDistance(itemsToRetrieve[i]);
        noisyDistance = trueDistance + sampleNormal(0, alpha *
          trueDistance);
        distances.push(noisyDistance);
    end
    itemsByEstimatedDist = sort(itemsToRetrieve, by = distances);
    itemOrder.push(itemsByEstimatedDist[0]);
    itemsToRetrieve.remove(itemsByEstimatedDist[0]);
end
itemOrder.push(itemsToRetrieve[0]);
return itemOrder;
```

To evaluate the simulation model, I run simulations of the same baskets seen in the data. Figure 5.5 shows the sub-task analyses applied to the simulated data overlaid on the shopper data. On the left, the simulated item sequences closely match the human data. Similarly, on the right, the fit α produces a very tight correspondence between the simulation and data trends as a function of the item distance difference.

To validate the simulation's accuracy at the decision level, I compare both the likelihood of inversions in the simulations to the shopper data, as well as the inversion magnitudes. Figure 5.6 (left) shows the simulation results for the decision-level data trend from Section 5.3, where the likelihood of choosing the optimal item (the complement of inversion chance) matches well the shopper data compared to a random choice. On the right is a comparison of the log frequencies of *inversion amounts* for the decisions, defined as $\hat{d} - d^*$. An inversion amount describes how much farther the chosen item was to the shopper than the closest item for a decision. Here, the simulation method continues to show good alignment with the shopper data, matching the actual inversion amounts with high accuracy.

The α parameter can be fit to data or used to tune behavior (for example, $\alpha \to 0$ will approach purely locally optimal decisions, which could be used to emulate shopper familiarity with the store layout). Additionally, the simulation technique is agnostic of store layout, and may be directly extended (or re-fit given new data) on any new store layout while retaining all the same properties that well describe the observed human behaviors.

Figure 5.7 compares simulated paths to shopper paths for the several baskets. Much of the time, the simulation makes the same decisions as the shoppers, matching several inversions and many locally optimal ones.

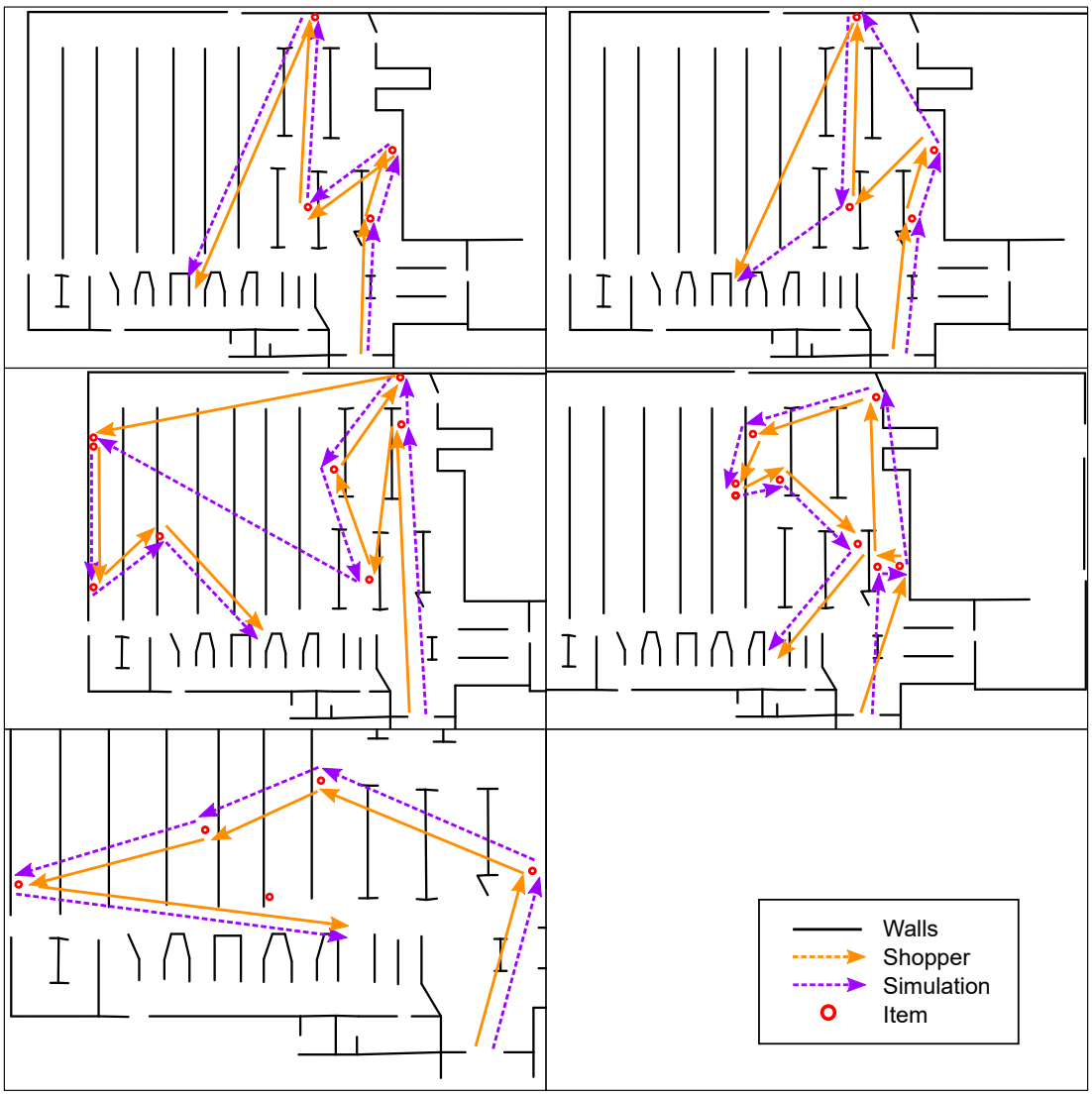


Figure 5.7: Simulated paths are shown (dashed arrows) alongside the shopper paths (solid arrows) through the store for several different baskets.

5.5 Conclusion

In this work, I present a new insight about human shopping behavior from a novel analysis of shopper path data. When viewed as a sequence of decisions among remaining items, the data shows that shoppers very typically choose the next closest item to retrieve (validating a modeling method suggested but not explored in (Hui et al., 2009a)). The chance of inversions in the data (that is, going to an item that was not the closest) follows monotonically the entropy of pair-wise item sub-tasks (that is, the difficulty of discriminating the closer of two items). I observe that an independent error estimation model with a linear relationship between an item's true distance and uncertainty well captures these trends. Based on these findings I propose an agent-based method for simulating the order of item retrieval given a basket and a store layout. The simulated data recovers the relationship between the chance of mistakes and sub-task entropy, and in practice well matches the shopper data.

5.6 Proofs

Here, I provide proofs for the following Lemmas and Theorems that support the analysis and results in this work:

Lemma: Simulated Inversion rate of two items Given two items having true geodesic distances b and c from the agent, their estimated distances $\hat{b}$ and $\hat{c}$ under the simulation model are

$$\begin{aligned} \hat{b} &= b + w_b, \epsilon_b \sim \mathcal{N}(0, \alpha b) \\ \hat{c} &= c + w_c, \epsilon_c \sim \mathcal{N}(0, \alpha c) \end{aligned} \tag{5.6}$$

Which produce Gaussian random variables of the estimated distances $B \sim \mathcal{N}(b, \alpha b)$ and $C \sim \mathcal{N}(c, \alpha c)$. Suppose $b < c$ (that is, item b is closer to the agent than item c). Then, the probability of an inversion is the probability that the estimated distances swap in magnitude: $C < B \rightarrow C - B < 0$. Let $Y = C - B$ be a new Gaussian random variable, then

$$\begin{aligned} Y &\sim \mathcal{N}(c - b, \sqrt{(\alpha c)^2 + (\alpha b)^2}) \\ &= \mathcal{N}(c - b, \alpha\sqrt{c^2 + b^2}) \end{aligned} \tag{5.7}$$

and the likelihood of inversion is $P(Y < 0)$. For a given b, c, and α, this quantity can be computed analytically from the CDF of Y evaluated at 0:

$$P(Y < 0) = \frac{1}{2}\left[1 + \text{erf}\left(\frac{0 - (c - b)}{\alpha\sqrt{2(c^2 + b^2)}}\right)\right] \tag{5.8}$$

Theorem: Simulated inversion chance increases with distance to the closer item First, we can note that $1 + \text{erf}(x)$ is a positive, increasing function of x. Then, given Equation 5.8, it suffices to show that for any b (the distance to the closer item), the input to *erf* is increasing:

$$\forall b \left[\frac{\delta}{\delta b} \frac{b - c}{\alpha\sqrt{2(c^2 + b^2)}} \geq 0\right], \; 0 < b \leq c \tag{5.9}$$

Proof: evaluating the partial derivative in Equation 5.9 with respect to b, we have

$$
\begin{aligned}
&\frac{\delta}{\delta b}\frac{b-c}{\alpha\sqrt{2(c^2+b^2)}} \\
&= \frac{1}{\alpha\sqrt{2}}\frac{\delta}{\delta b}\frac{b-c}{\sqrt{c^2+b^2}} \\
&= \frac{1}{\alpha\sqrt{2}}\frac{\sqrt{b^2+c^2}-\frac{(b-c)b}{\sqrt{b^2+c^2}}}{(b^2+c^2)} && \text{(quotient rule)} \\
&= \frac{1}{\alpha\sqrt{2}}\frac{c(c+b)}{(b^2+c^2)^{\frac{3}{2}}} && \text{(simplify)} \qquad (5.10)
\end{aligned}
$$

Since Equation 5.10 is positive whenever b, c and α are positive, the derivative is positive with respect to b and constrains Equation 5.8 to be increasing with increasing b.

Theorem: Simulated inversion chance decreases with increasing distance between items Another important property for maintaining the relationship between difficulty and inversion chance is that the inversion chance must decrease with increasing $W = c - b$.

Proof: We wish to show that the derivative with respect to W is always negative:

$$\forall W = c - b, \left[\frac{\delta}{\delta W}\frac{b-c}{\alpha\sqrt{2(c^2+b^2)}} \leq 0\right],\ 0 < b \leq c \qquad (5.11)$$

Noting that $(b^2 + c^2) = W^2 + 2cb = W^2 + 2b(b + W)$, we can substitute into Equation 5.11 and get

$$\frac{\delta}{\delta W}\frac{-W}{\alpha\sqrt{2(W^2+2b(b+W))}} \qquad (5.12)$$

While we cannot write the derivative in terms of only W, we can treat b as a positive constant and take the partial with respect to W. If the result is negative for

any value of b, then the derivative with respect to W is negative regardless of b and the property is satisfied:

$$\begin{aligned}
&\frac{\delta}{\delta W}\left[-\frac{W}{\alpha\sqrt{2(W^2+2b(b+W))}}\right] \\
&= \frac{1}{\alpha\sqrt{2}}\frac{\delta}{\delta W}\left[-\frac{W}{\sqrt{W^2+2b(b+W)}}\right] \\
&= \frac{1}{\alpha\sqrt{2}}\left[-\frac{\sqrt{W^2+2b(b+W)}-\frac{W(W+b)}{\sqrt{W^2+2b(b+W)}}}{W^2+2b(b+W)}\right] && \text{(quotient rule)} \\
&= \frac{1}{\alpha\sqrt{2}}\left[\frac{W(W+b)}{(W^2+2b(b+W))^{\frac{3}{2}}}-\frac{W^2+2b(b+W)}{(W^2+2b(b+W))^{\frac{3}{2}}}\right] && \text{(simplify)} \\
&= \frac{1}{\alpha\sqrt{2}}\frac{-b(2b+W)}{(W^2+2b(b+W))^{\frac{3}{2}}} && \text{(simplify)}
\end{aligned} \tag{5.13}$$

Since W and b are both positive non-zero quantities, the resulting derivative is always negative as desired.

Theorem: Simulation inversion chance increases monotonically with difficulty To show this property, it is sufficient to show that difficulty also increases monotonically with decreasing W and increasing b, since both these two values fully specify both the inversion rate (given an α) and the difficulty.

Proof: First, we note that W and b are sufficient to fully describe difficulty:

$$\textit{difficulty} = \log_2\left(\frac{A}{W}\right) = \log_2\left(\frac{b+W}{W}\right) \tag{5.14}$$

Then we can construct the partial derivatives with respect to both W and b for difficulty and see that they are always positive and negative respectively:

$$\frac{\delta}{\delta W}\frac{A}{W} = \frac{\delta}{\delta W}\frac{b+W}{W} = \frac{-b}{W^2} \tag{5.15}$$

$$\frac{\delta}{\delta b}\frac{A}{W} = \frac{\delta}{\delta b}\frac{b+W}{W} = \frac{b+W}{W^2} \tag{5.16}$$

Thus, both inversion rate and difficulty monotonically increase with increasing b and decrease with increasing W. Since W and b are both sufficient to fully specify both inversion rate and difficulty, we cannot make one smaller or larger (by adjusting W or b) without having the same effect on the other. Therefore, both these quantities will have a monotonic relationship with each other.

Chapter 6

Realistic Navigation Behavior with Uncertain Goals in Building-like Environments

In the same vein as Chapter 5, in this chapter I study simulating human-like navigation behaviors under local information constraints, this time incorporating uncertain goal locations. However, whereas in Chapter 5 I derived insight on human motion from motion data directly, here I introduce a data-driven simulation technique to produce human-like behavior, but do not use actual human paths to train. Instead, I propose a global navigation task formulation in a way that matches semantically the task humans face in the real world. Applying a deep learning approach that trains on automatically generated optimal routes produces several human-like global navigation behaviors. I then use human paths on the same navigation tasks (in a virtual environment) as a validation technique.

A version of this work appears in the 2020 ACM Conference on Motion, Interaction and Games (Sohre and Guy, 2020).

6.1 Introduction

The interactive simulation of human motion is important in many scenarios, with applications ranging from video games to building design and smart city planning all benefiting from high-quality human movement and behavior. Recent years have seen many exciting advances centered around developing more human-like approaches in areas such as collision avoidance (Dutra et al., 2017) and character animation (Lee et al., 2019). However, there has not been similar progress in developing human-like approaches to the mid-level task of determining how an agent plans long-term paths through an environment. Here, I look to close that gap by proposing a human-like approach to global navigation planning.

When navigating in new or unfamiliar environments, humans must develop long-term navigational plans to reach their goals, while only seeing local information (e.g., doors, walls, and hallways), and are often uncertain about the precise goal locations. Cognitive Science has revealed several key components of human navigation behaviors, such as looking for long goal directed avenues (Bailenson et al., 2000; Lima et al., 2016) and the use of *fuzzy mental maps* to aid in decision making (Epstein et al., 2017; Kaplan et al., 2017). Here, I propose the Scene-Planning Network (SPNet) framework. SPNet emulates the human-like approach to global navigation through a custom neural network structure that first builds up an additive representation of the places an agent has explored so far, and then leverages that representation, together with uncertain goal locations, to develop a (stochastic) plan of what next action is likely to make progress toward the goal. My approach captures several important behaviors that are not possible with either full global planning or simple local heuristics. SPNet agents will integrate the history of what they have seen to improve their navigation, make different decisions in response to varying levels of certainty with the task at hand, and will stochastically follow a variety paths in situations where

multiple different paths to reach the goal are reasonable. Importantly, the training approach used in SPNet reaches human-like performance on these tasks without requiring any human training data, allowing the approach to be easily adopted in a variety of applications.

6.2 Related Work

Our method draws from research in Computer Graphics, Deep Learning, and Cognitive Science. Here I examine some closely related work from these fields.

6.2.1 Local & Global Navigation

Previous work in human-like navigation has considered both local and global contexts. Related work in local navigation has focused on realistic behaviors (McDonnell et al., 2008; Kapadia et al., 2015; Karamouzas et al., 2017), often with data-driven (Charalambous and Chrysanthou, 2014; Wang et al., 2016; Karamouzas et al., 2014) or geometric (van den Berg et al., 2008) reactive approaches. Others have looked to produce realistic local behaviors by modeling problems similarly to what humans experience (Ondřej et al., 2010; Lee et al., 2019). Here, I take a similar approach, broadening the scope to considering compelling human-like behavior in the context of global navigation.

Graph search has been used for global navigation in interactive games, VR, and animation to allow virtual characters to plan over a graph-like representation of their environment, producing graph-optimal global routes to their goals, (Botea et al., 2013; van Toll et al., 2016). Other works have focused on acceleration techniques (Lee and Lawrence, 2013), or exploiting heuristics and optimized structures to accelerate the search (Kallmann and Kapadia, 2014). Planners such as Rapidly-exploring Random Trees and its variants (LaValle, 1998) use a sampling based approach to build plans

in continuous environments under control constraints, while others apply computer vision to directly predict next actions (Rabiee and Biswas, 2019), perform waypoint navigation (Bansal et al., 2020), or vision-based navigation (Gupta et al., 2017).

Very related to my work is the area of agent-centered search. Works in this field provide approaches for real-time heuristic search under local information and time constraints. Approaches such as D* Lite (Koenig and Likhachev, 2002) and LRTA* (Korf, 1990) incrementally learn heuristic information in an online fashion, and extensions focus on improving convergence properties (Hernández and Meseguer, 2005), learning speed ((Koenig and Sun, 2009; Sturtevant and Bulitko, 2011)), and memory efficiency (Bulitko and Björnsson, 2009). LRTS provides a learning framework with a tunable relationship between performance and optimality (Bulitko, 2004). Here, my work will focus on creating human-like behavior (as opposed to guarantees on heuristic admissibility) while allowing the goal to be fuzzy (the agents only have a distribution describing where the goal may be).

6.2.2 Cognitive & Behavioral Studies of Human Navigation

In cognitive science, researchers have found that humans do not tend to take shortest path routes (Duckham and Kulik, 2003), and have proposed heuristics to capture human exploration behavior, such as traveling as far as possible towards a goal (Bailenson et al., 2000), or maintaining a small relative angle in heading with respect to the goal (Lima et al., 2016). Warren et al. (Warren et al., 2001) performed user studies to learn a model as a linear function of goal distance and angle. Beyond local heuristics, research has shown that humans integrate memory to form fuzzy mental maps that influence navigation (Epstein et al., 2017; Kaplan et al., 2017).

6.2.3 Deep Learning

Deep Learning (DL) techniques have been broadly applied to train large neural networks in many fields (Sze et al., 2017). Approaches for navigation can either be supervised with known training data (as in (Pfeiffer et al., 2017)), or the network can be trained to optimize path cost using reinforcement learning (as in (Tamar et al., 2016) and (Long et al., 2018)). Deep reinforcement learning (DRL) has recently shown promise for navigation as a fine-tuning final step, as was done in (Pfeiffer et al., 2018) and (Luo et al., 2020). Very recent work has sought to improve the practicality of DRL navigation by augmenting it with classical, analytical planners((Chaplot et al., 2020)). Neural networks have also proven useful in transforming raw input into rich and meaningful representations (Cai et al., 2019; Doersch, 2016). Additionally, works such as (Peters and O'Sullivan, 2002) have used learning techniques to model ways people understand sensory information. Eslami et al. used an additive representation network to encode virtual scene descriptors that can be used to query views from novel perspectives (Eslami et al., 2018). A similar cognitive structure has recently be used for efficient robot navigation on a grid (Gupta et al., 2017). I adapt a similar architecture to build scene representations in a continuous environment.

6.3 Local-Global Planning

Like humans, SPNet agents are assumed to have only local, limited information about their environment and imprecise knowledge of their long-term goals. The agent's task, then, is to integrate its local observations, together with a learned understanding of typical building layouts, to plan efficient paths toward likely goal locations. I assume the agent is updated in real-time following a sense-plan-act loop where it computes its direction of travel frequently based on its history of recent observations.

This section details the underlying problem formulation and path execution strat-

egy of the SPNet framework. In the next section I will introduce a novel network structure which makes the actual predictions of the next optimal step for the agent.

6.3.1 Problem Formulation

Given a single agent tasked with navigating to given goal whose location is not precisely known, I represent each aspect of the problem as follows:

Agent Representation The agent is represented by a circle with a radius r large enough to encompass the collision model associated with the agent's animation.

Environment Representation The agent's environment is defined by a series of line-segments which represent impassible obstacles, and the agent is assumed to be able to move otherwise freely and continuously in $\mathbb{R}^2$. An example map can be seen in Figure 6.1.

Goal Representation An agent is not given the exact location of its goal, but rather only a general fuzzy notion of its location. I represent this as 2D Gaussian distribution centered at (μ_x,μ_y), with a standard deviation of σ in each dimension.

Sensing and memory An agent can only directly observe the portion of line segment obstacle walls which are visible from its current position. In practice, I represent this as a series of i rays centered on the agent, whose length is the distance to the wall. Together these rays form a visibility *isovist* from the point of view of the agent. An example isovist can be seen in Figure 6.1. At any moment, the agent is assumed to have access not only to its current visibility isovist, but also up to the last n isovists it has previously seen. In this way, agents have memory of the past.

For all results, I use $i = 60$ rays to capture the local visibility isovist. The agent's

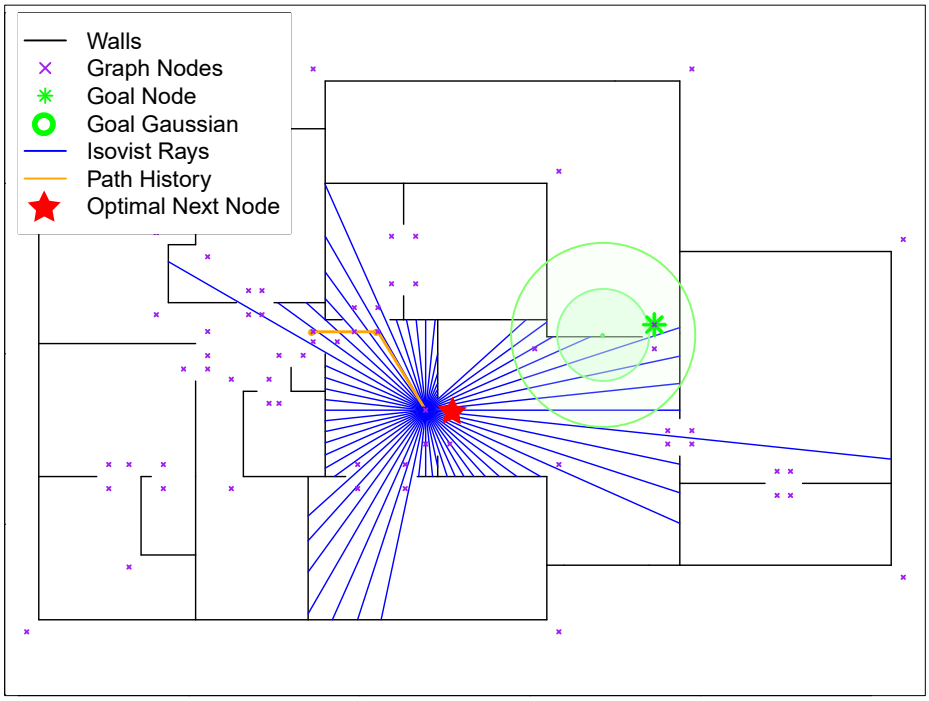


Figure 6.1: **Feature Representation**. An example training feature is rendered in the context of an example environment. The goal distribution is shown as 1 and 2 standard deviation rings (σ= 2m).

memory, which I will refer to as the *path history* of length n, is capped at 3 in training, but is allowed to grow to larger values in execution.

6.3.2 Execution Strategy

The optimal path between any two points in a 2D map can always be found by connecting straight lines between corners or ends of walls (Lozano-Pérez and Wesley, 1979). I refer to any such (potentially optimal) point as a navigation node, and statically compute all such navigation nodes for any map (purple x's in Figure 6.1). This discretizes the continuous navigation problem without sacrificing any potential path quality by choosing between reachable navigation nodes. I move navigation nodes to be r away from the closest wall to account for the agent's radius.

My fundamental execution strategy is as follows. Given a set of navigation nodes, a start node, and a goal distribution, the agents utilize a neural-network to select the next navigation node to travel to. At each step, given the current and recent local visibility isovists, the network predicts the next optimal node position. The agent then moves to whichever of the neighboring navigation nodes (e.g., currently visible nodes) is closest to the network's prediction. To guarantee the agent eventually reaches the goal (and avoids infinite travel loops), I introduce a *maxVisitCount* parameter, which specifies the maximum number of times the agent can visit the same node before the node is removed from consideration.

Algorithms 6.1 and 6.2 detail the SPNet execution strategy. Because SPNet executes very quickly, it is possible to plan fully to the goal and smooth away minor inefficiencies in the resulting path. Here, I generally avoid such smoothing techniques as they can suppress desired exploration behaviors. One exception is that if a network predicts visiting a node it has already visited (or a node for which the agent has already seen all of the node's neighbors), I allow the agent to move directly toward the next predicted node as soon as it comes into view. In practice, the particular character animation system used may provide some additional path smoothing as the character animates between nodes.

Algorithm 6.1: SPNet

Input: map (Walls and Nodes), start (Node), goal (Node), goalRegionMean (Position), goalRegionSigma (float)
Output: path (Node List)

```
#global maxVisitCount = 3;
/* visit count for each node                                    */
#global visits = [0] * map.numNodes;
/* path is full route (will include back-tracking sequences)    */
path = [];
/* history is path with cycles removed                          */
history = [];
current = start;
while current != goal do
    next = SPNetStep(map,current,goalRegionMean,
     goalRegionSigma,history,path);
    path.push(next);
    current = next;
end
return path;
```

Algorithm 6.2: SPNet Step

```
Input: map (Walls and Nodes), current (Node), goal(Node),
 goalRegionMean (Position), goalRegionSigma (float), history (Node List),
 path (Node List)
Output: next (Node) visits[current]++;
if isVisible(goal) then
    next = goal;
else
    /* gets all visible nodes from current node visited less than
       maxVisitCount times                                          */
    candidates = map.getVisibleNodes(current, maxVisitCount);
    if candidates.size == 0 then
        /* retrace our steps                                       */
        next = history.pop();
    else
        position = getNetworkPrediction(current, goalRegionMean,
         goalRegionSigma, candidates, path);
        next = map.getClosestNode(position);
        history.push(current);
    end
end
return next;
```

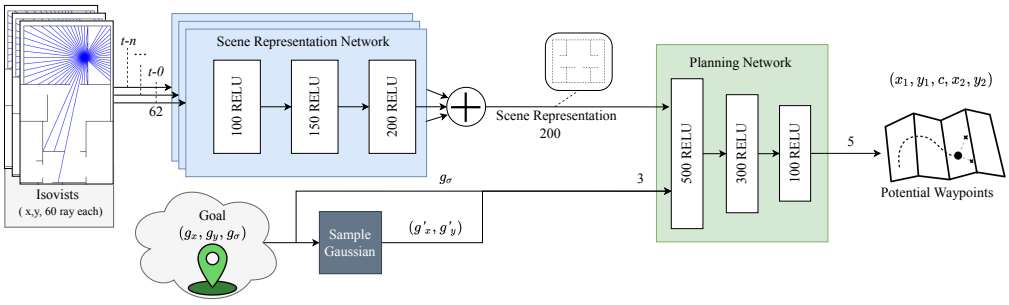


Figure 6.2: **SPNet Network Structure**: Isovist feature rays from the path history are transformed by the scene representation network and accumulated into a scene representation. The planning network predicts two potential actions it thinks are likely to lead to efficient paths given the representation, along with a relative confidence between the two choices.

6.4 Learned Navigation

SPNet's novel network-based formulation encapsulates both the process of the agent building a mental map of the environment and planning the next best location towards which to move. In this section I both describe the network architecture and discuss the training approach.

6.4.1 Network Structure

The structure of the prediction network (shown in Figure 6.2) draws on the idea of mental maps. Rather than taking in a visibility isovist (or series of isovists) and directly predicting the next location, I divide the tasks into separate responsibilities: a Scene Representation Network whose job is to build a representation of the map seen so far, and a Planning Network, which takes this representation along with the (fuzzy) goal location and predicts the next step to take. This separation of responsibilities also leads to greater flexibility in how the network is used in execution.

I implement the Scene Representation Network using a three layer neural network (500-300-100 nodes respectively), with a ReLU activation function between each layer.

The input of this network is the normalized 60 isovist ray lengths, together with the x and y offset of where the isovist was seen relative to the agent's current position. The output of this network is a 200 valued feature vector which represents the environment. The Scene Representation Network is run once for each isovist in the agent's *path history*, defined as the most recent n isovists (and their offsets) that the agent has seen. The resulting encoding from each isovist are then summed to produce the final 200-dimensional scene representation vector. Critically, this additive representation does not require the agent to use a fixed path history size in execution or training. Summing the Scene Representation Network output over more or fewer previously observed isovists will change the agents behavior in a natural fashion. This behavior tuning is discussed further in Section 6.5.2.

The Planning Network combines the 200-dimensional scene representation together with the goal region to determine the likely next actions. The goal input is represented as three parameters defining a Gaussian which describes a distribution over possible goal locations. These values are concatenated with the encoding, and the resulting 203 dimensional vector is then passed through another three layer network (100-150-200 nodes respectively), with a ReLU activation function between each layer. The output of this network encodes the next step for the agent to take.

6.4.2 Navigation Prediction

Rather than producing a single 2D coordinate for the next navigation step, the network outputs a stochastic prediction split over two alternatives: $p_0 = (x_0, y_0)$ and $p_1 = (x_1, y_1)$ represent the probable offsets (relative to the agent) of optimal next navigation nodes, and c ranges continuously from 0 to 1 represent the network's choice between p_0 and p_1 (0 for choice 0, 1 for choice 1).

I train this output using a three-part loss function:

$$\mathrm{L}(p_0, p_1, c, y) = \qquad k_m \mathrm{L}_{\min} + k_a \mathrm{L}_{\text{avg}} + k_p \mathrm{L}_{\text{pred}} \tag{6.1}$$

with p_0, p_1, and c as defined above, y as the true optimal prediction for this scenario, and k_m, k_a, and k_p as tuning parameters that relatively weight the three components of the loss.

Each of the three components of the loss function serves a different purpose. The first term:

$$\mathrm{L}_{\min}(p_0, p_1, y) = \min(s_0 \|p_0 - y\|, s_1 \|p_1 - y\|) \tag{6.2}$$

measures how close the better of the two predictions is to the optimal next node. The s_i terms, computed as $s_i = 5 * \text{sigmoid}(||p_i - y|| - ||p_a - y||)$ where p_a is the agent position, add penalties to discourage overly short predictions. The $min()$ ensures this component of the loss only provides a reward for improving the better of the two predictions. This allows the network to split the prediction, as there is no penalty for the wrong half of the split. Combined over training data representing ambiguous scenarios, the two predictions are driven towards different, but equally promising next steps.

Simply optimizing the better of two predictions during training is insufficient when one of the two predictions is much further away from the correct node than the other. The second loss term, L_{avg}, addresses this by providing a small penalty for the average of the two predictions:

$$\mathrm{L}_{\text{avg}}(p_0, p_1, y) = (\|p_0 - y\| + \|p_1 - y\|)/2 \tag{6.3}$$

This term must have a small weight to ensure the primary loss comes from L_{min} to

maintain split predictions.

The network also must learn to indicate which of the two predictions is the correct using an output value c. This is accomplished via the third loss term, $\mathrm{L}_{\mathrm{pred}}$:

$$\mathrm{L}_{\mathrm{pred}}(p_0, p_1, c, y) = |(c - \mathrm{sigmoid}(\|p_1 - y\| - \|p_0 - y\|)|, \tag{6.4}$$

where the sigmoid function, $1/(1 + \mathrm{e}^{-x})$, maps the relative difference between the errors of the two options to the continuous domain $[0, 1]$. This loss will be driven to zero when c matches the (sigmoid of) the relative error. When c is very near 0 or 1, it means the network has identified this scenario as unambiguous, having a clear choice as to which node is the better prediction. ($c = 0$ implies the agent should move to p_0, and p_1 when c=1.) When c is not very close to 0 or 1, the network has identified this scenario as ambiguous, and neither choice is clearly better than the other. In this case, the network can choose the prediction with which c is more aligned, or randomly select an option, leading to a spread of reasonable paths (which I will refer to as *stochastic node selection*). Unless otherwise stated, for all results I used a k_m of 0.999, a k_a of 0.001, and a k_p of 10.

Figure 6.3 shows the effect of the network's split prediction. At the top left of the map, the network does not have enough information to know which side of the courtyard is most optimal, and bifurcates the prediction between the two hallways. When the agent is in a less ambiguous part of the building, the network is more confident in the next step.

6.4.3 Data Generation & Network Training

To facilitate training and evaluation, I created a dataset of environments based on real floor plans from a variety of buildings of different shapes and sizes (see Figure 6.19 and Figure 6.20). Maps used in training the network include three smaller buildings,

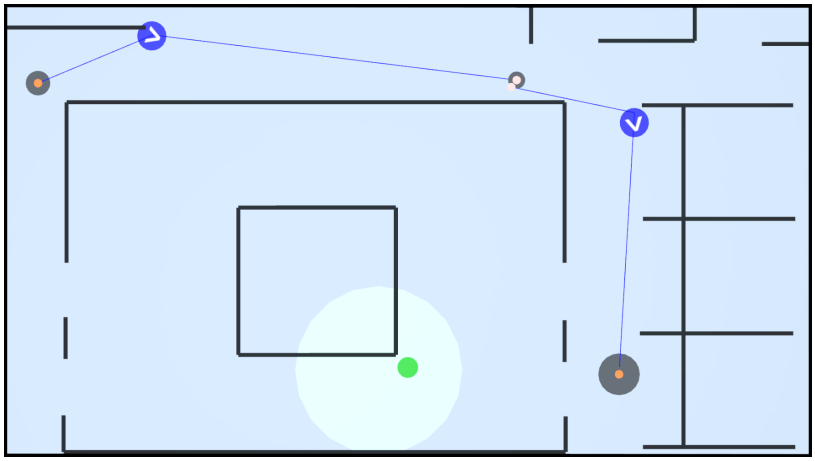


Figure 6.3: **Prediction under Ambiguity.** The agent (circle with inset arrow) navigates to its goal (circle inset in large uncertainty region) from 2 locations in a courtyard-style map. Lines eminating from the agent terminate at predictions (larger circle → more confident prediction). Unambiguous scenarios produce greater confidence in the chosen direction.

two larger buildings, and a small test map made by hand. This dataset consists of a wide range of different environments, with very different types of layouts, and very different sizes (see Table 6.1). Additionally, I withheld a set of maps from training to be used in analysis of how the method performs in untrained scenarios. All maps are rendered in Figure 6.20 and Figure 6.19.

While I employ a supervised learning approach, it would be impractical to gather enough human path data to demonstrate good behavior on a meaningful percentage of the potential tasks. For example, the 6 maps I use in the training dataset contain 6.9 million training examples of different start goal pairs permuted with possible previous observations. In order to create this data, I instead use a graph-optimal search to determine the optimal path from start to goal, and use the resulting step-wise decisions in training. Although SPNet is trained on optimal paths, my goal is not to produce purely optimal behavior. Rather, attempting to be globally optimal while restricting the network to only local, uncertain information results in paths

Table 6.1: **Datasets:** the collection of maps analyzed in this work. The table indicates the size of the maps, and the number of instances of training data that would be generated from that map. The three largest maps and one smaller map were excluded from training and used to test generalization.

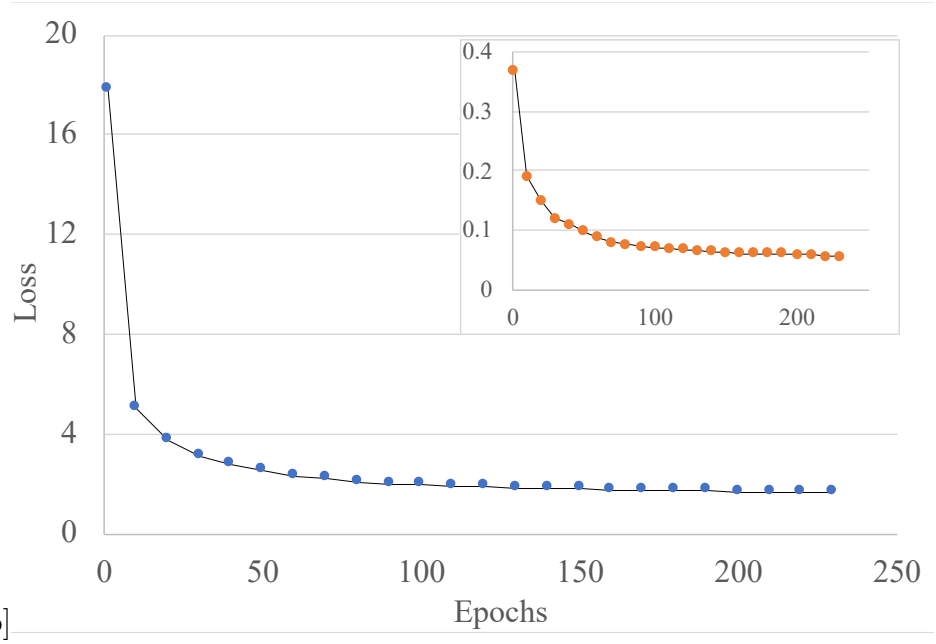
	Walls	**Nodes**	**Size** (m)	**Data (#)**	**Use**
Simple	29	28	12x8	16 K	Train
Apartment	65	65	14x9	157 K	Train
House	72	84	29x19	168 K	Train
Courtyard	75	97	17x12	867 K	Train
Medical	101	109	23x27	1.1 M	Train
Office	84	95	28x17	781 K	Test
University Labs	129	154	44x26	4.66 M	Test
Business Park	161	197	33x27	4.60 M	Test
Conference	276	332	61x33	15.44 M	Test

with human-like sub-optimal behaviors, even on maps used in training.

In training, the network is never given the actual goal position, but a distribution with a mean sampled from a Gaussian centered at the true goal location with uncertainty σ. In order to integrate this uncertain representation into a gradient-based optimization architecture, I employ a custom network layer which samples the fuzzy goal region to generate a specific training instance (the dark shaded box in Figure 6.2). Importantly, the σ used to perturb the goal location is also an *input to the navigational network*, enabling the input σ to be used as a run-time tuning parameter that controls how much the agent is willing to explore.

The Scene Representation Network and the Planning Network are trained together as one system. The training data is generated by an exhaustive sampling of every reachable start-goal pair of navigation nodes in training maps. Given a start-goal pair, I permute possible path histories to generate training rows. The network weights are trained using Keras (Chollet et al., 2015) using stochastic minibatch gradient decent with an ADAM optimizer.

The network was trained on 10 cores of an 2.3 GHz Intel(R) Xeon(R) CPU.



[b]

Figure 6.4: **Training Loss**: Total training loss (outset) and just the predictions loss L_{pred} (inset) throughout training.

Training was run for 230 epochs that iterated through all the training data, with each epoch taking about 3 minutes, and the entire run taking 12 hours. As can be seen in Figure 6.4, most of the improvements in loss came at the start of training, but the longer run ensured the network had sufficient time to converge.

Training was performed across 5 maps and included all possible paths with a history of length three for a total of 2,315,665 training examples, each of which was sampled with three levels of goal uncertainty (sampled from 0, 1, and 2m Gaussian distributions respectively), leading to about 7 million samples trained each epoch. Epochs were trained in batches of size 100,000 using the Keras framework with the ADAM optimizer and a dynamic learning rate that started at 3e-4 and ended at 1e-5.

To improve numerical stability, isovist ray lengths are normalized to lie on the interval $[0, 3]$ by dividing by $10 = 1/3$ of the maximum length of 30m. Unless otherwise stated below, the network was trained with a goal uncertainty $\sigma \in \{0, 1, 2\}$

meters, and a maximum path history of size 3. I note that while 2m may seem like a small distance, in practice sampling from a 2D Gaussian with i.i.d dimensions at $\sigma = 2m$ often yields points significantly farther than 2m from the mean (in fact, the 1 std dev circle having diameter 4m will probabalistically contain less than half of sampled points). As discussed in Section 6.5.2, the network was able to extrapolate good behavioral performance beyond the uncertainties and history sizes on which it was trained.

In order to help reduce the tendency of the network to overfit to the maps in the training pool, noise was added during training. First, a small amount of zero-meaned Gaussian noise (std. dev. of .01m) was added to each of the input isovist rays, and then a larger amount of noise (std. dev. of 0.1) was added to the encoded, goal-independent local representation. Previous work in the neural network literature has shown improvements by adding this type of noise during training, both in terms of the ability of networks to generalize to situations not seen in training, and to create smoother, more meaningful internal representations (see for example (Noh et al., 2017) or (Eslami et al., 2018)).

6.5 Results & Analysis

6.5.1 Network Analysis

As discussed above, allowing the network to split its prediction led to more human-like behaviors. Additionally, these split predictions improve the training performance of the network. Figure 6.5 shows a comparison of the SPNet network trained on a single map with 6m of goal uncertainty with and without split predictions. With split predictions enabled, there is over a 60% reduction in the final loss L (from 3.68 down to 2.26). Similarly, the additive representation enabling path history also led

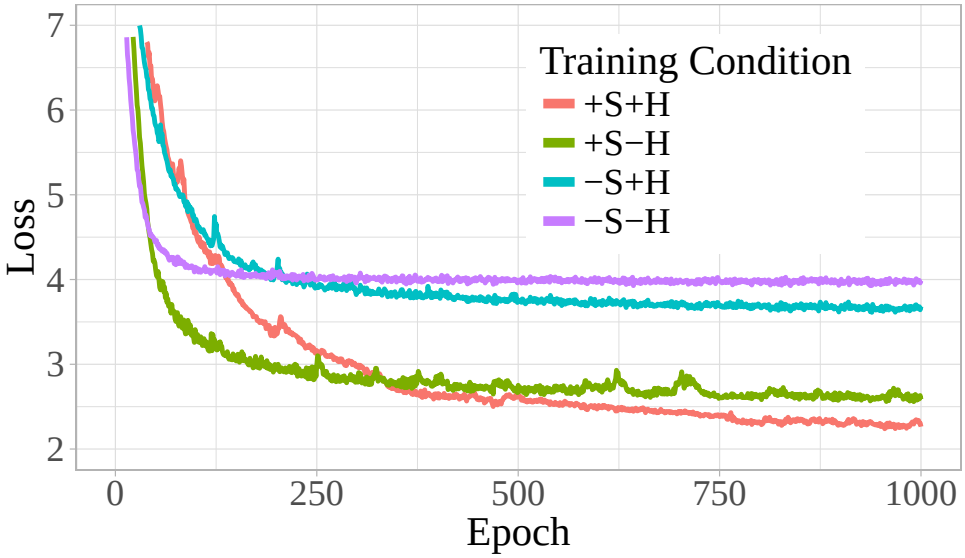


Figure 6.5: **Training Profile Comparisons.** A + or − in a condition label denotes presence or absence of an SPNet network feature respectively. "S" and "H" represent split predictions and path history respectively.

to improved loss in training. Figure 6.5 shows that incorporating path history led to an additional 15% reduction in the training loss.

Runtime Breakdown

The runtime of SPNet planning is dominated by three major components. Figure 6.6 shows the runtime breakdown of each for maps of various sizes. All experiments were run on a desktop PC with an Intel i7 4770K CPU and 16GB of memory. While computing isovists takes longer on larger maps as more visibility checks are needed, the planning network runtimes are fairly constant. Longer path histories add incremental cost as the scene representation network is run once per recalled isovist. Even for very large maps, SPNet runs in real-time, taking less than 2ms to output a next step. If faster performance is required, isovists can be precomputed for each navigation node.

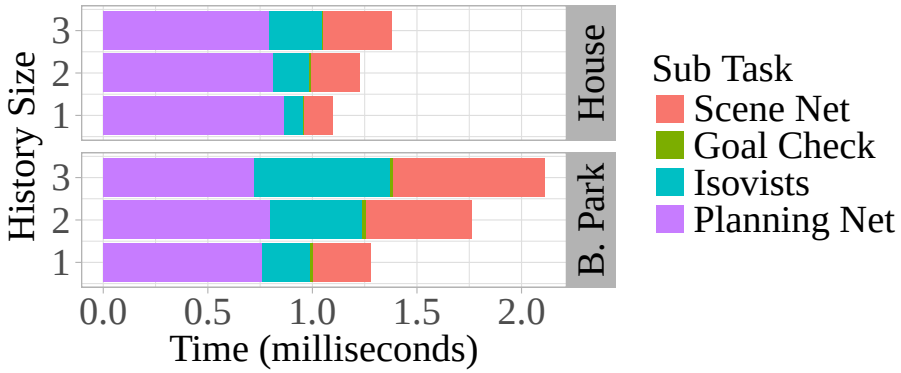


Figure 6.6: **SPNet Runtime**: Average total runtime for each step of planning in SPNet across history and map size. 50 random tasks were executed for a small (House) and large (Business Park) map to generate timing samples. Runtime cost is primarily split between executing the prediction network, scene network, and computing isovist features.

6.5.2 Behavioral Analysis

Our problem formulation and custom network structure allows SPNet agents to display several human-like navigation behaviors that are not possible either with optimal planning techniques or simple local heuristics. SPNet agents respond only to their local conditions, explore in search of vague goals, integrate their observations over time, and intelligently backtrack when they get stuck in local minima. The framework runs in real-time, with a full planning step taking 1-2ms (see Appendix Section 6.5.1 for runtime details).

As the SPNet predictions are a distribution over two likely possible actions, stochastically selecting one can create a natural diversity of paths. Because the choice of which node to go to next follows the distribution predicted by the network, the agent paths tend to vary more in ambiguous situations. Figure 6.7 shows some example path bifurcations, overlaid with user paths from a user study I detail in Section 6.6.1.

Figure 6.8 shows a SPNet agent navigating past a fork juncture on an apartment-

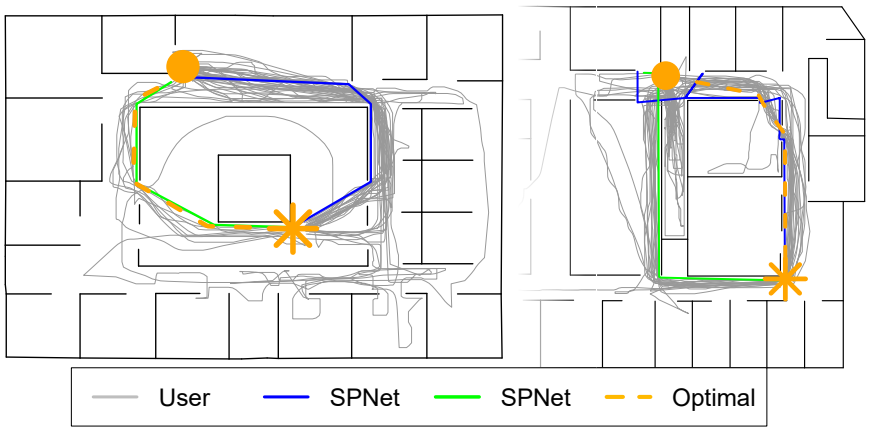


Figure 6.7: **Human-Like Behavior:** SPNet produces efficient, human-like paths both for maps on which it was trained (left), and maps not seen in training (right).

style map. Here, when the goal uncertainty is small (left), the agent navigates straight to the goal, taking a near-optimal path. While this path might be expected from a person who is very familiar with both this building and the exact goal location, its not very reflective of the ambiguity inherent in the navigation task. Increasing the goal uncertainty (right) naturally leads to an exploration behavior.

SPNet agents have two main tunable parameters which can control the behaviors: the size of the goal region σ (creating less certainty in the agents decisions), and the maximum path history n (creating more intelligent backtracking behaviors). These two parameters can also interact with each other. For example, for low σ, paths are more efficient regardless of the size of the path history. When the goal region grows larger, path history becomes more important, especially as the agent needs to reason about which previously explored paths are unlikely to lead to the goal.

Figure 6.9 explores the interaction of goal uncertainty and path history by comparing SPNet paths to optimal averaged over 200 random tasks spread over the 5 training maps. A larger history size increases path optimality across goal uncertainties, even for history sizes and uncertainties beyond those used in training. A 2-way ANOVA reveals that goal uncertainty has a more significant effect on the agent's path

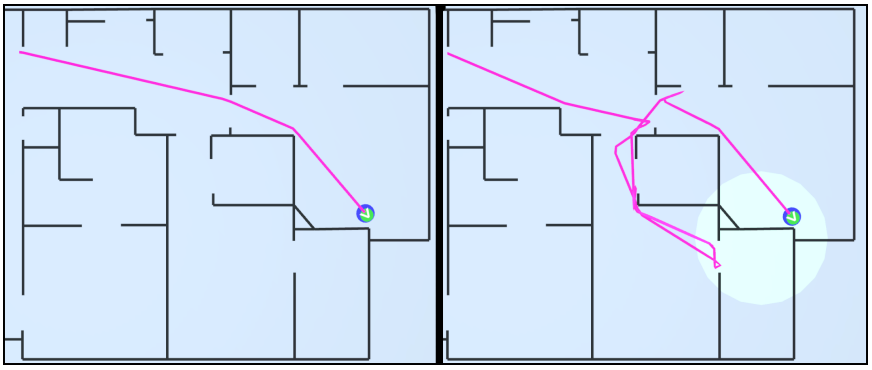


Figure 6.8: **Effect of Goal Uncertainty:** The right path has $\sigma = 2.5$m and the left has no goal uncertainty. The high uncertainty case produces exploratory behavior as the agent searches for the goal, while the agent on the right heads directly for the certain goal location.

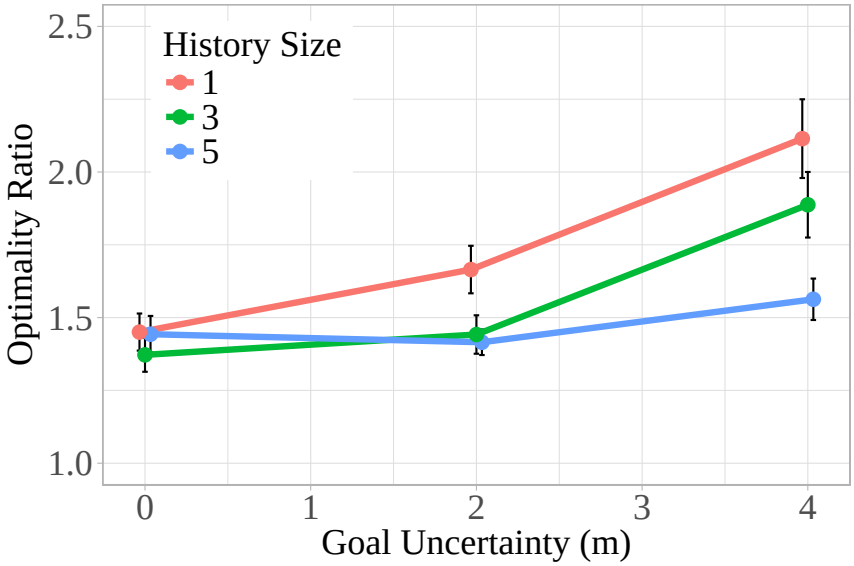


Figure 6.9: **Effect of Behavioral Parameters.** Goal uncertainty and history size can be tuned to affect agent behavior. Large uncertainty leads to exploratory behavior, and increasing the history size leads to more optimal paths.

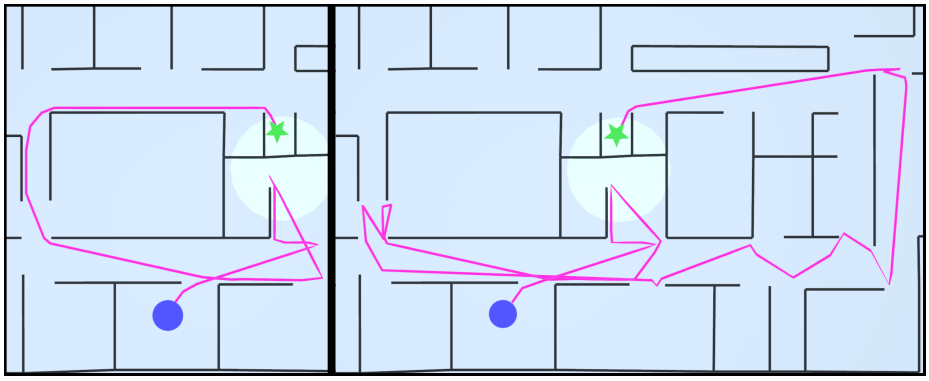


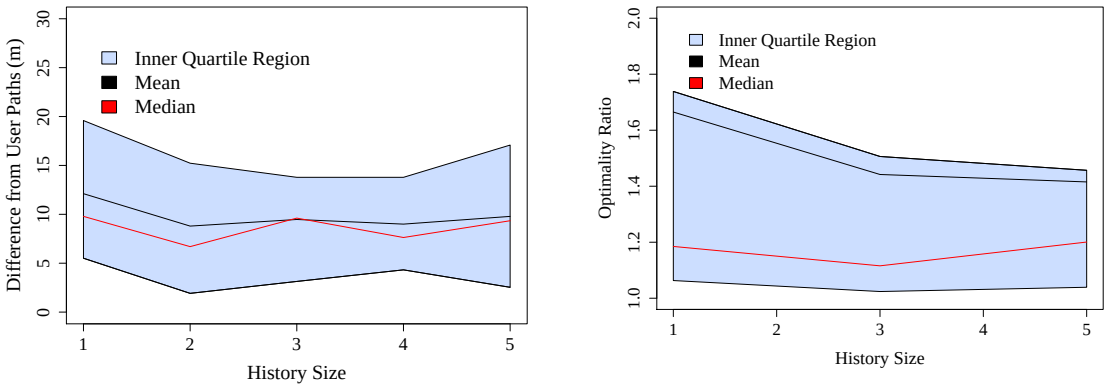
Figure 6.10: **Effect of Path History.** Two simulated paths for the same task are shown. The right path was generated by an agent with no path history (n=1), and in the left path the agent incorporates the past three visited map nodes in planning.

than history (history $F = 2.4$, $p < 0.1$, uncertainty $F = 7, p < 0.05$). These results confirm the ability of the scene representation and planning networks to generalize outside the range of parameter values used in training.

Effect of History

Figure 6.10 shows an example of the key navigational behaviors SPNet agents exhibit. On the left is an agent with a max path history of 3 isovists and a relatively large goal distribution (σ = 2m of uncertainty, shown as the shaded circle at 1 std dev radius). After initially exploring the room overlapping the bottom half of the goal distribution, the agent rules out the area as a potential goal location and remembers this as it travels up the left side of the environment. In contrast, an agent who is using only its single, current isovist (no path history) is unable to remember where it has been. Figure 6.10 right shows such an agent, who must return to places it had been before, eventually finding a longer route to the goal.

The effective size of the history can be increased simply by summing additional encodings together before sending them to the planning network. As shown in Figure 6.9 and 6.11a, this additive encoding scheme allows the SPNet to make meaningful



(a) (b)

Figure 6.11: *Left:* Effect of history size on path optimally averaged across all paths with goal size σ=2. *Right:* The representation network allows for the integration of past observations that influences future decisions, resulting in more human-like path lengths on untrained maps.

use of additional history even outside the sizes it was trained on. 6.11b shows the effect of planning over longer histories on the optimality of the resulting paths. Plans that account for more history lead to more optimal paths, even when using longer histories than were seen in training.

Some over-fitting effects can be observed in trained maps where SPNet produces path lengths closer to optimal than human paths, while matching human path lengths better on untrained maps. This suggests that techniques to help combat over-fitting such as larger training data sets and early-stopping may further improve SPNet's path quality on trained maps. Alternatively, this over-fitting can be exploited as a feature when agents should exhibit familiarity with a particular environment.

		Study 1	Study 2
Participants		48	27
Task Set Size	Training Maps	8	25
	Testing Maps	2	15
Tasks Performed	(per participant)	16.2 ± 9.5	13.4 ± 7.0
Task Frequency		36.9 ± 11.3	9.1 ± 2.4
Gaming Experience	<1 hr/month	13	7
	1 - 4 hrs/month	11	5
	5 - 20	9	9
	20 +	15	6

Table 6.2: **User study and participant summary.** The task set breakdown indicates the number of available tasks in maps on which the network was trained or not trained. Task frequency refers to the average number of times a given task was performed by a user. The game experience reflects participant responses to the question "About how often do you play First-Person or other Action-based 3D video games?"

6.6 Validation

6.6.1 User Study

In order to better understand the routes that would actually be taken by humans, I conducted a user study in which participants performed navigation tasks in the same environments considered in this work via a 3D game-like interface. During the navigation tasks, the participant trajectories were captured, yielding a dataset of human paths useful for validation comparisons with SPNet agent paths.

Experiment Design

To partake in the study, participants used first-person WSAD controls to navigate from a start position to a goal position (which together compose a navigation task) through a 3D rendering of the environment. The study software was developed using the Unity Game Engine ©, and built to be run in a web browser. This allowed users to participate using their own computers by simply following a hyperlink. Instructions

advised using a mouse and keyboard, along with recommended browser and computer specifications.

To render the user's perspective as they moved about the environment, the 2D line segments of the map are transformed into an opaque wall with a random, neutral-toned color. As the maps come from real-world layouts, the units (in meters) are preserved in the rendering. A mini-map in the bottom right corner of the screen displayed a live update of a top-down view of the user's position, orientation, and relative goal location. Importantly, walls are not rendered in the mini-map in order to maintain local information constraints. This both supports a similar informational context for users and what would be available to an SPNet agent, and more importantly helps elicit the navigational behaviors that I am interested in capturing (namely, those that humans exhibit when navigating under local information constraints). An example screenshot from the study application is shown in Figure 6.12.

Each participant took one of the two versions of the study. In the first study variant, all participants performed the same eleven navigation tasks in a random order, followed by randomly generated tasks that were not used in analysis. In the second version of the study, participants were given a randomly ordered set of 40 predefined tasks across all maps (5 from each of the 8 maps excluding the "Simple" map, amounting to 5 trained and 3 untrained).

When the user reached the goal for a given navigation task, they were presented with a success screen and could choose to continue to the next task, which would place them at a new start position (of a potentially new environment) with a new goal. Participants were given a target number of navigation tasks to complete (11 for the first variant and 15 for the second), but were allow to do as many or few as they chose. Each user was given a sample "warm-up" task to acclimate to the controls where the goal was visible from the start position. As the user moved through each task, their path was tracked at a frequency of 10 samples per (virtual) meter displacement.

Participant Information

The first study variant had 48 participants, with 27 participants in the second variant. Participants were recruited primarily by word-of-mouth by computer science lab students. While separate recruitment was conducted for each study variant, participants from the first study were not barred from partaking in the second. All participants were asked to indicate their amount of experience with first-person action-based video games on a discrete scale of < 1 hour per month to 20+ hours per month (see Table 6.2). The participant pool represented a range of gaming experience, with good coverage across all experience levels.

Study Results

Table 6.2 shows a breakdown of some participation statistics. In total, 879 paths were collected for the first study variant, ensuring a robust sampling on each task to support comparisons within a single task. The second study variant yielded 394 paths. As compared to the first study variant, each condition has less coverage from different users, but the study overall includes a larger number of tasks. In the first study, no participant saw more than 5 tasks in the same map, and in the second, the most tasks for a single user on the same map was 3. This supports interpreting the paths as being taken in an unfamiliar environment.

Before performing analyses, both user and simulated paths are put through a low-pass running median filter as implemented in the *runmed* function of the statistical programming language R (R Core Team, 2020) with k=11 to remove any small jittering in the paths.

The thorough coverage of each task in the first variant of the study allows a qualitative comparison of the routes frequently taken by humans to be compared to those generated by SPNet agents. Figure 6.7 shows the results from two tasks in the first

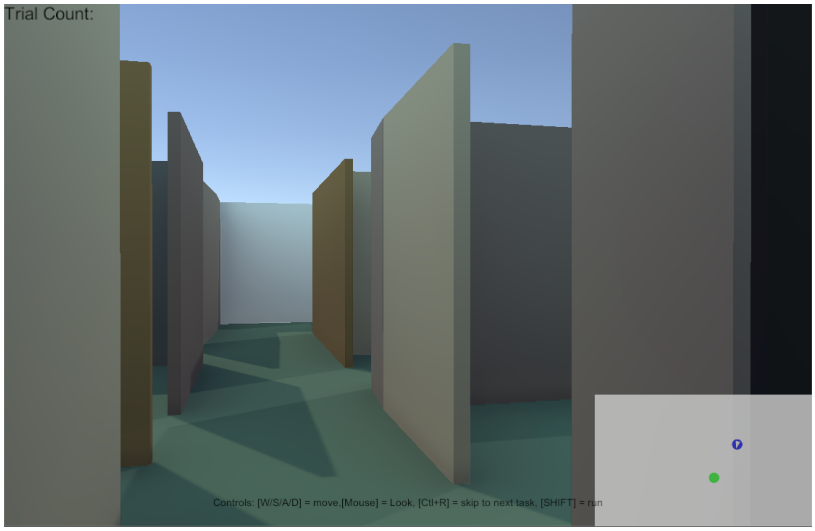


Figure 6.12: **User Study Interface**: Snapshot from the user-study tasks. Participants were asked to navigate to a goal whose relative position was indicated by a green dot in the mini-map at the lower right.

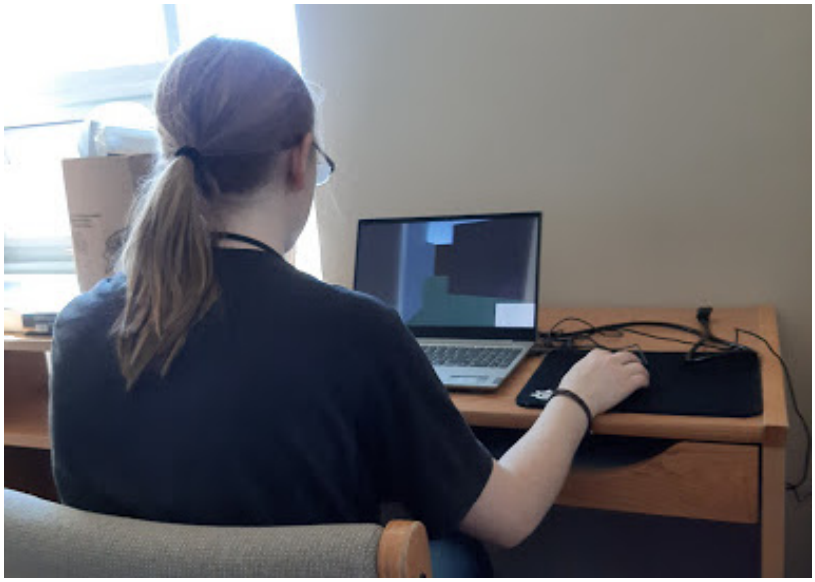


Figure 6.13: **User Study Participation**: The study was implemented as a web-hosted 3D game experience, allowing users to participate in the study on their own computers via a web browser in full-screen mode.

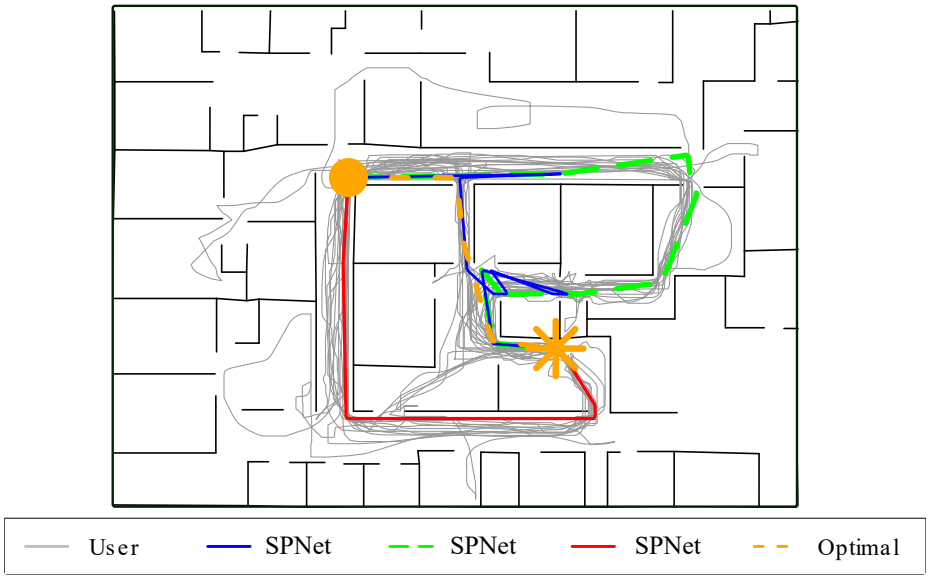


Figure 6.14: **User Path Comparison**: Paths generated by SPNet agents cover the same general routes as those taken by users.

study, where we see both SPNet agents and participants take a variety efficient paths (that frequently differ from the optimal path commonly used in crowd simulation). While the map on the left side was a map used in training, and the map on the right was not, in both cases the generated paths are well aligned with a frequented participant route. The different routes are made possible by using the stochastic node selection approach described in Section 6.4.2 where the agent randomly selects one of the two predicted nodes weighted by the c value output by the navigation network. Simiarly, Figure 6.14 further supports the ability of SPNet agents to support a variety of natural routes, matching well the different general routes taken by participants.

The data from a larger number of tasks in the second variant of the study supports a quantitative comparison between SPNet agent paths, user paths, and optimal paths. As users in my study had never navigated through the maps before and only saw each map a few times throughout the course of the study, it is more appropriate to compare tasks on maps for which the SPNet was not trained. Figure 6.15 shows a

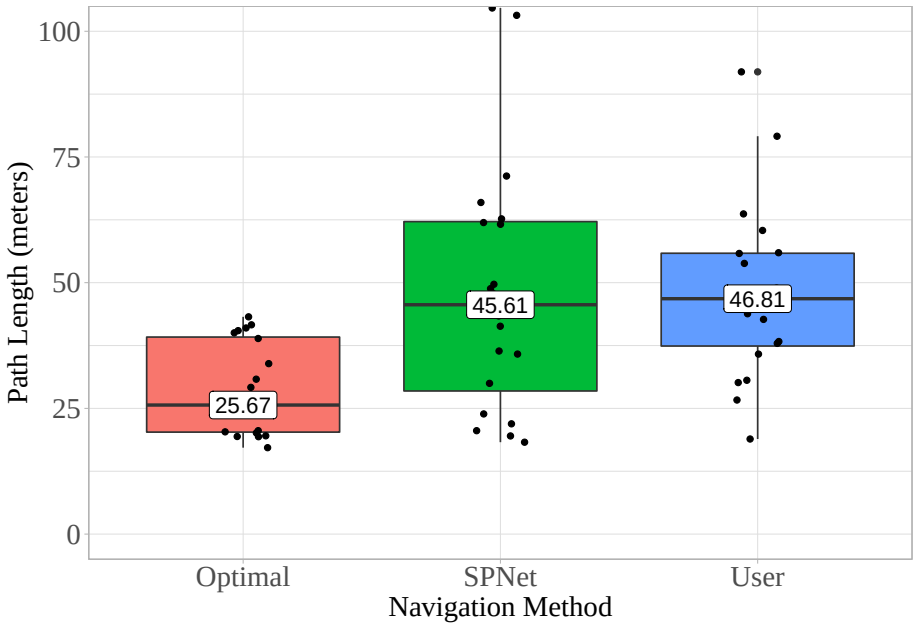


Figure 6.15: **Path Length Comparison**: Path lengths are shown for the optimal path, those generated by SPNet agents (σ = 6m, path history = 3, stochastic node selection enabled), and those taken by participants in the second part of the user study (for the same tasks) on untrained maps.

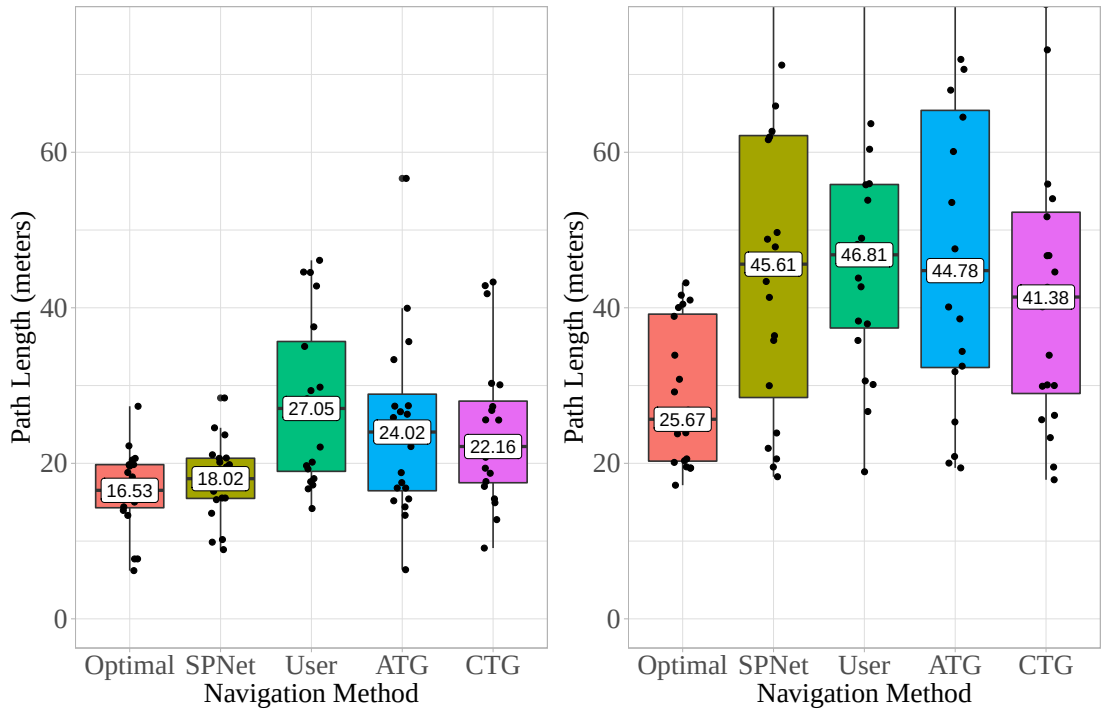
comparison from the second study variant between the path lengths taken for tasks on untrained maps by an optimal planning technique, the study participants, and a SPNet agent with goal uncertainty $\sigma = 2$m, path history size = 3, and stochastic node selection enabled. As compared to optimal routing, SPNet agents were on average 6.2m closer to the human path length for each task, and closer to the user path length than the optimal path 75% of the time. For path lengths in general, SPNet path lengths ($\mathrm{M} = 48.43\mathrm{m}, \mathrm{SD} = 25.22\mathrm{m}$) better matched the distribution of user paths ($\mathrm{M} = 47.81\mathrm{m}, \mathrm{SD} = 17.43\mathrm{m}$) than the optimal paths ($\mathrm{M} = 28.73\mathrm{m}, \mathrm{SD} = 9.17\mathrm{m}$). An ANOVA analysis confirms a statistically significant effect ($F(2, 57) = 7.3, p < 0.01$). A Tukey post-hoc analysis confirms a stable difference between optimal and user paths ($p < 0.01$) and between optimal and SPNet ($p < 0.01$), but not between user paths lengths and those generated by SPNet ($p > 0.9$).

6.6.2 Comparison to Local Heuristics

SPNets bridge the gap between global planning techniques and local navigation heuristics. Human route selection studies have revealed high-quality local route selection heuristics that strongly influence human paths. Two of these techniques from the cognitive science domain that can be implemented in the broader navigation framework of this paper (with a relaxation of the problem formulation) are as follows:

- Traveling as far as possible towards a goal (Bailenson et al., 2000), which I will refer to as Closest-to-Goal (CTG)
- Maintaining a small relative angle in heading with respect to the goal (Lima et al., 2016), which I will refer to as Angle-to-Goal (ATG)

Notably, both of these methods require the true goal location to properly compute the heuristic. This requires a relaxation of the problem formulation, namely, that the true goal location is known. While the mean of the goal region could be used as it



(a) Trained Maps

(b) Untrained Maps

Figure 6.16: Human users take longer paths than the graph-optimal sequence. For trained maps, SPNet path lengths are close to graph-optimal, but on new maps SPNet paths are closer to the paths taken by humans.

maximizes the likelihood of the true goal location, this would cause the methods to behave very sub-optimally. In this case, the heuristics would continually bias towards the same errant locations in cases where the true goal is sufficiently different than the mean of the goal region, backtracking to the same locations repeatedly until node visit counts are exhausted. Instead, to enable a comparison, I allow the heuristics to "cheat" and use the true goal position.

Figure 6.16 shows the results of replacing the network predictions with CTG and ATG-based heuristics for round 2 of the user study. The left side shows results for tasks on trained maps, and the right shows tasks on untrained maps (same tasks as shown in Figure 6.15). When compared to the tasks in the user study data, both CTG and ATG had path lengths which were more similar to human paths than the optimal route selection for tasks on both trained and untrained maps. This result is consistent with the existing findings in psychology of the general importance and applicability of these local navigation heuristics to humans. For trained maps, SPNet agents are much more familiar with the environment, and naturally take more efficient paths (the same σ and path history are used in all comparisons).

While the CTG and ATG heuristics perform quite well compared to user paths, SPNets provides several benefits worth noting. SPNet paths match human path lengths as well as (or better than) these heuristics on untrained maps, while enabling more natural behaviors in their global routes; in practice, ATG tends to oscillate and CTG can get stuck in obstacle local minima. SPNets supports a variety of natural paths with stochastic node selection, while heuristics must yield the same result every time. For maps on which it is trained, σ and path history size can tune agents in real time to exhibit greater or lesser familiarity with the environment. Additionally, SPNets supports uncertain goals and memory integration, while both ATG and CTG must be given the true goal location to be computed.

As expected, users had much longer path lengths than SPNets on trained maps.

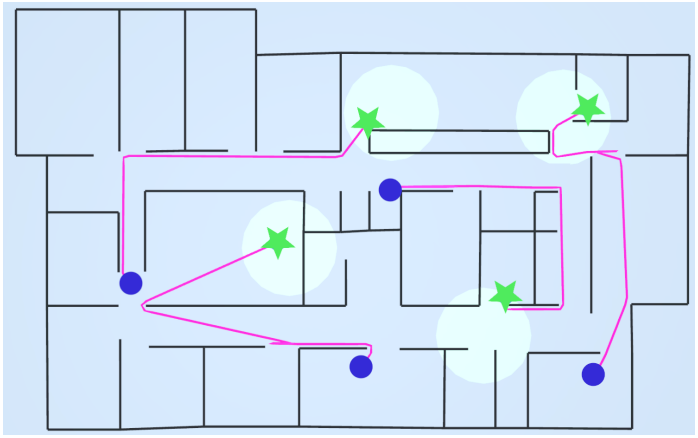


Figure 6.17: **Generalization under Goal Uncertainty.** Simulated paths are shown for a mid-size test map (Office). Dots and stars indicate the start and goal locations respectively. SPNet agents are able to find efficient paths even underneath mid-sized goal uncertainty (here σ=2m, n=3).

It is plausible that after a long time exploring the same map, humans would build up a familiarity with that map and the paths would look closer to optimal. For maps on which it was trained, low σ close to 0m will bring SPNet paths close to optimal as well, allowing animators to set a dynamic level of familiarity displayed by the agent.

6.6.3 Generalization to Untrained Maps

Figure 6.17 and Figure 6.18 show SPNet agents navigating in a variety of paths on untrained maps. Empirically, the agents regularly find efficient and natural paths that reach their goal, despite having never seen the map in training. In practice, goal uncertainty has a more natural effect on behavior for smaller to mid-sized untrained maps. In large, complex maps such as those in Figure 6.18, even a small history size compared to the total path length can produce efficient paths for tasks with relatively shallow local obstacle minima (the paths depicted were produced with history size = 3). For very large and complex environments, larger history values (in both training and execution) are important for producing higher efficiency paths.

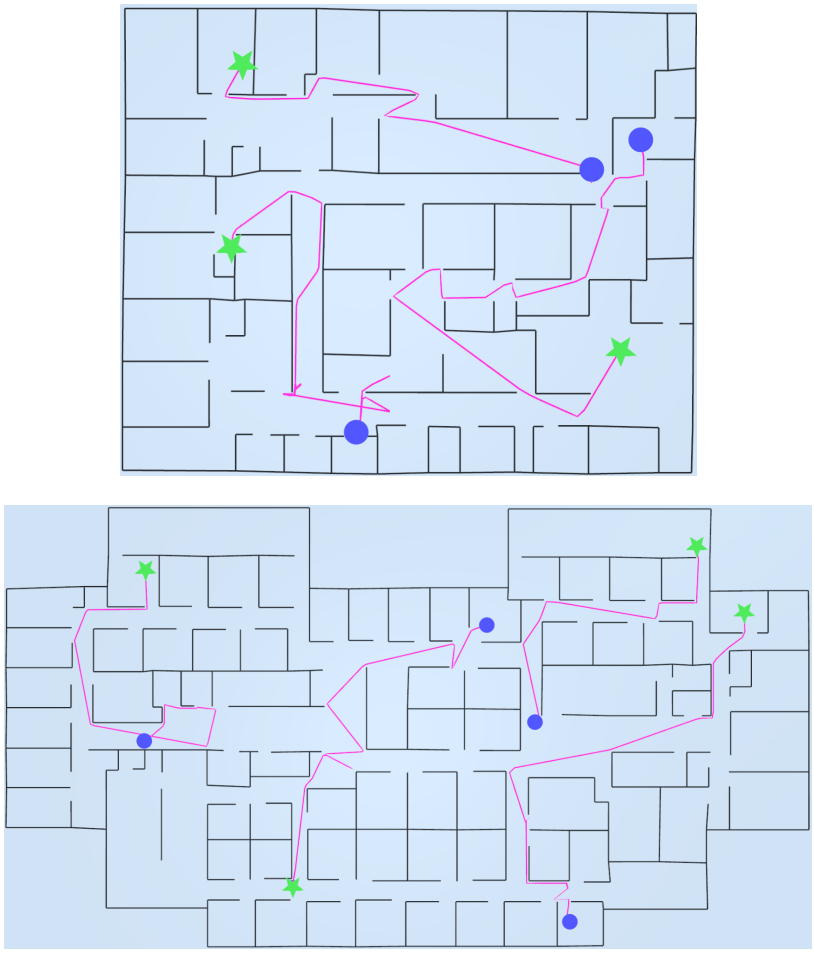


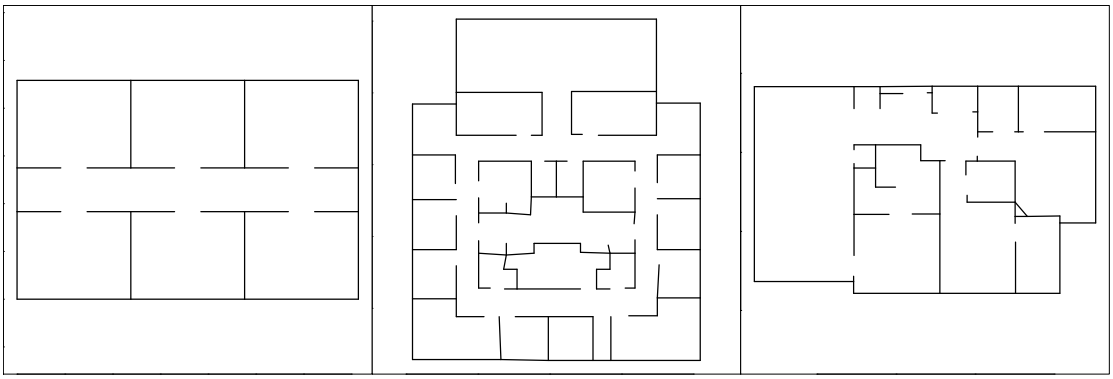
Figure 6.18: **Generalization to Large Maps.** *Left:* Selected SPNet paths on the Conference map (n= 3, σ = 0m). *Right:* Selected SPNet paths on the Business Park map (n= 3, σ = 0m). SPNet agents are generally able to find efficient paths, especially when given the true goal location.

6.7 Limitations & Future Work

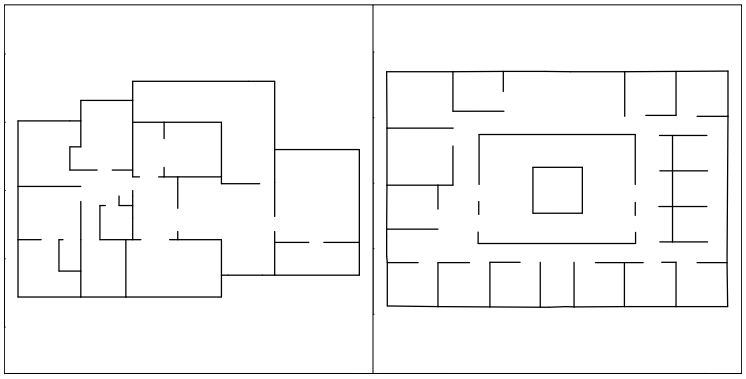
While SPNet covers an important aspect of a human-like global navigation, it focuses on single agent navigation. Planning in environments with multiple agents typically require specialized search techniques (Atzmon et al., 2020) or hierarchical models (Musse and Thalmann, 2001). Expanding the network input to include the relative position and velocity of other agents could combine local and global planning into an integrated network. While SPNet is already real-time for a single agent, faster performance may be needed to support large crowd simulations with thousands of agents in a shared scene. Here, it may be possible to accelerate isovist construction using spatial data structures, and to speed up network computation by using GPUs.

6.7.1 Map Renderings

Below are renderings of maps used in training (Figure 6.19) and testing (Figure 6.20). All environments were based on real floor plans, abstracting away irrelevant obstacle detail.

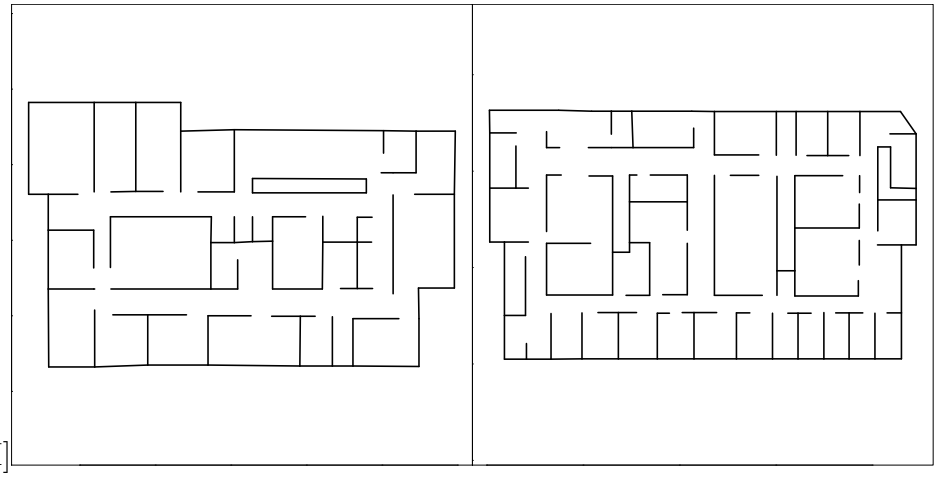
[H] 

(a) Simple (b) Medical (c) Apartment

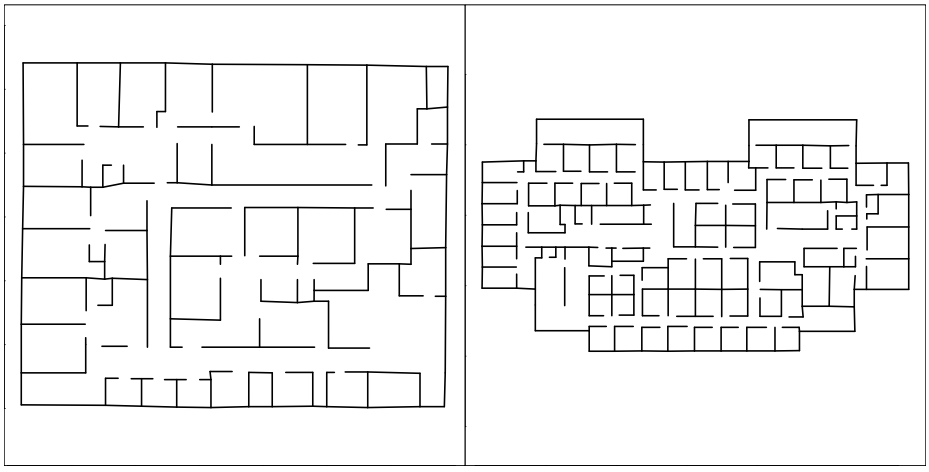


(d) House (e) Courtyard

Figure 6.19: Training Maps.

[H] 

(a) Office (b) University Labs



(c) Business Park (d) Conference

Figure 6.20: Testing Maps.

Chapter 7

Closing Remarks

As evidenced by the examples included in this thesis, a large part of moving towards AI that enriches everyday life is understanding how humans move. Building realistic models of human motion requires both capturing the aspects that make the motion appear natural, as well as the rich variety of motion seen within and among individuals. No matter where one looks, this duality presents itself in the real world time and again. My approach to understanding humans by studying movement in this way has value pervading many domains and scales of behavior, from smiles to collision avoidance to long-term navigation decisions. The increasing practicality with which human motion data can be captured opens doors to new insights and analysis, and will continue to drive research in this area. Even so, the challenges of producing ever increasing realism will continue to require innovation as AI solutions seek to enter a widening set of application domains.

Throughout this dissertation, I have presented techniques as evidence to support my thesis that data-driven methods are useful for creating realism in and evaluating the realism of human motion simulations. Identifying naturalness and variety as two major components of realism, I approached the endeavor by leveraging a combination of datasets containing human paths along with user studies to assist with both the simulation of motion and evaluating the realism in the results. In each presented

work, I have given demonstrations of the efficacy of proposed methods by applying them to concrete cases in real domains. Following is a summary of the key results of each.

7.1 Summary of Contributions

At the opening of this dissertation I provided a concise list of contributions contained in my work that support my thesis statement. Here I expand briefly on each contribution to connect it with the corresponding material presented throughout the previous chapters.

A Dataset of Casual Observer Reactions to Digital Smiles

In Chapter 2, I designed, implemented and conducted a large user study of nearly 1,000 casual observers to collect over 10,000 reactions to a set of digital smile animations. These animations represented a systematic sweep of a low dimensional feature space I introduce, called Facial Space. The semantically meaningful basis supported by Facial Space is useful for studying the interactions between movements of the mouth and perceptions along several subjective axes of emotional intent, pleasantness, and authenticity. This data represents a rich sampling of the interaction between movements of the mouth and the associated measures, and is useful for a variety of interdisciplinary fields. The raw data is made publicly available in digital form in (Helwig et al., 2017).

A Procedural Generation Framework for Human Smiles

In Chapter 3, I demonstrated how to leverage the dataset presented in Chapter 2 to build a generative model of realistic human smiles for virtual characters. This framework constitutes a novel method focused on enabling a variety of mouth movements

while simultaneously achieving a targeted emotional intent. I validated this claim with a follow-up user study that showed with strong statistical significance that my method accomplished both its goals of variety and semantic targeting.

A Precision-Variety Learning Heuristic

In Chapter 3, I also addressed the fundamental trade-off between the precision of a classifier and its variety by proposing a heuristic ordering of the data that exposes the trade-off through a tuning parameter to allow the maximum variety to be achieved given a precision threshold. I provided a theoretical motivation for underlying data conditions that guarantee when this parameter will have the desired effect, and showed both empirically in the smile dataset and a toy dataset that the assumptions hold sufficiently well. This heuristic represents a novel contribution to machine learning in the context of generative classifiers with targeted semantic classes, which directly supports my thesis goal of simulating realistic motion.

A Validation of Collision Avoidance for Realism in VR

In Chapter 4, I performed an evaluation of the impact of data-driven collision avoidance on user experiences in immersive 3D virtual environments (VR). This evaluation was two-pronged, using both an objective measure of tracked user paths in the environments and a subjective simulator sickness questionnaire to measure user experiences. The analysis I provide gives clear indication that data-driven collision avoidance methods are critical to supporting realism in immersive virtual settings. These results represent both a contribution as a first user study to evaluate data-driven collision avoidance in HMD-based VR and a demonstration of data-driven validation as outlined in my thesis statement.

A Data-Driven Analysis of Shopper Navigation Decisions

In Chapter 5, I propose an analysis framework for studying multi-task global navigation decisions, and apply it to a large dataset of paths collected from shoppers in a grocery store. The analysis reveals a strong relationship between the entropy (or difficulty) of sub-tasks and the likelihood of making sub-optimal choices relative to the locally greedy choice. These findings contribute to the state of the art in understanding human decision processes as they apply to global navigation. I proposed a simulation model that follows from the entropy relationship for generating a variety of plausible (natural) routes given a set of items to retrieve in a store, and show the results match the human dataset with high accuracy, providing support in an additional domain for my thesis claim that data-driven methods can produce realistic motion.

A Deep Learning approach to Global Navigation with Uncertain Goals

In Chapter 6 I propose a unique approach to generating realistic global navigation simulations for building-like environments with uncertain goals. There I demonstrated that even without collecting large samples of human trajectories for learning, realistic simulations could still be made possible by reformulating the learning task to solve a human-like problem. I did this by training a custom neural network architecture I designed (SPNets) on automatically generated optimal paths, but subjecting the information available to similar constraints that humans face in the real world. Additionally the neural network was designed to include human-like faculties such as memory, the ability to represent informational ambiguity, and an internal separation of scene representation and planning.

A Simulation Algorithm for Realistic Global Paths

Placed in the broader context of a simulation framework, I showed the SPNet network is capable of producing a variety of human-like routes in a number of buildings. The paths exhibit several human-like navigation behaviors, which contributes to the state of the art in human-like global path simulation. These include intelligent backtracking, narrowing down goal locations, and familiarity (increased optimality) for maps on which the network was trained. Both the naturalness and the variety of the results are validated with a small user study collecting human paths on the same tasks via a virtual interface.

7.2 Impact of Contributions

While the works included here are intended to be directly useful in the applications in which they were contextualized, the insights gained extend beyond the specific domains to the broader field of AI. For example, Precision-Variety Learning is not only a novel machine learning heuristic with it's own merits, but the work also shows that user studies conducted on systematic simulations enables the capture of intangible elements of how humans perceive naturalness in simulated motion. This was validated by using the data to train a generative model, and confirming the realism of the resulting simulations with a follow-up user study. Similarly, the collision avoidance evaluation work complements this idea by establishing that human paths can be used to evaluate the naturalness of existing data-driven methods. The shopper path analysis produces valuable insight for the study of human gathering behavior, and is consistent with other studies of human cognition that relate informational complexity of problems to performance on a given task. Finally, the indoor environment navigation work shows that even when collecting large quantities of human paths impractical, a data-driven method can still produce a variety of natural routes for

a virtual agent (closer to human paths than optimal or using simple heuristics) by formulating the problem using similar constraints to what humans face in real life.

7.3 Limitations

While I have discussed the limitations inherent to the techniques presented in the corresponding chapters, I provide a summary here.

A limitation of the machine learning heuristic I presented for tuning the precision-variety trade-off is its reliance on underlying data conditions. While these conditions are desirable for any classification task, they may not always be strongly present, and are less likely to be so with increasing dimensionality and sparsity of sampling within the feature space. In generating facial expressions with this method, while our dataset supported the challenging task of producing a variety of happy facial expressions at different intensities, the feature space we used might need to be extended beyond mouth shapes to support other emotions. Similarly, the feature space for the classifier considered only spatial measures, whereas evidence points to temporal aspects having importance for perceiving emotional intent. For example, in (Helwig et al., 2017) my co-authors and I found that subtle temporal asymmetries had a significant impact on the perceived pleasantness of smile motions. Additionally, classification methods that focus on optimizing for computational speed and memory requirements would be needed for practical implementations in real-time settings such as video games that might require expressions to be generated on-the-fly. Finally, a dataset using expressions animated on a more diverse set of facial models would better capture the diversity of human expressions.

While the results in my work on evaluating collision avoidance in immersive virtual environments showed compelling support for its importance, the size and scale of the user study was limited to under 10 participants in a relatively small working area.

An open question is how the presence or absence of collision avoidance may impact other more typical crowd interaction scenarios, such as walking along with a crowd or allowing them to merge in a more organic fashion (our track required participants to enter the flow at a 90° angle). Additionally, it is not clear how instructional cues (such as the positioning of the path markers in VR) may have affected the interactions.

The analysis performed on the shopper motion data was based on the simplifying assumption that shoppers were making decisions based on full knowledge of the list of items they would eventually purchase. Of course, this is not always the case, as shoppers sometimes make unplanned purchases (Massara et al., 2014) or consider but ultimately reject items they had planned to buy (though the dataset I used does not have sufficient information to study such phenomena). While the local greedy formulation I adopt for modeling a shopping trip (where the shopper attempts to identify the next closest item and goes there) has intuitive merit, the extent to which this is valid for human shoppers is an open question. Additionally, my analysis of task entropy considered only pair-wise item decisions. While it seems to extend well to the N-item case in practice, additional work would be needed to prove this in theory.

My work on human-like global navigation behaviors focuses on a single agent in a building-like environment. The problem of planning for multiple agents in the same space can present new challenges requiring specialized approaches (Atzmon et al., 2020; Musse and Thalmann, 2001). An example of this is when agents have to coordinate to share limited spatial resources such as narrow passageways (Hildreth and Guy, 2019). Integration with other features or an extension of the SPNets network would be needed to leverage the existing features in such settings. While SPNets are already real-time for a single agent, faster performance may be needed for scenarios with many agents or to extend the work to roadmaps with many nodes, such as probabalistic roadmaps (Kavraki et al., 1996).

7.4 Future Work

While this dissertation represents a substantial step towards the goals identified in Chapter 1, there are many opportunities for future resesarch in this area. A more human-centered, social form of AI will require many new insights, innovations, and technical contributions to become a fully recognized priority. Here I highlight several avenues related to my existing work and establish a vision for the future of the field that inspires further exploration.

7.4.1 Next Steps

Despite facial animation being the focus of a large body of work in computer graphics and other fields, more effort is needed to fully capture and connect the subtle motions of virtual faces to perceived emotion and realism. A combination of data-driven methods and user studies like the one I performed in Chapter 2 can help move the field towards a more explicit coupling of these elements in both modeling and animation. For example, extending the work in Chapter 3 to identify key facial features corresponding to other emotions and similar data-driven analysis of emotional intent with the motion of those features can lead to increasing support for automatically generated realistic facial expressions over greater set of emotions. Additionally, the complex interplay between the spatial and temporal dynamics of facial movements is likely to have a substantial impact on realism. While my collaborators and I saw evidence of this in (Helwig et al., 2017), that aspect deserves much deeper inspection. Finally, another promising area for discovery includes using a data-driven approach to characterize the variety of motion seen in real human facial expressions. With the increasing availability of computer vision techniques for tracking facial features without the need for extensive setup and calibration, such a dataset may be pratcical to build and analyze.

As immersive virtual environments become more commonplace and available to an ever widening audience, it is important to understand how to support a user's sense of presence, embodiment, and agency in this setting. My work in Chapter 4 supports the importance of collision avoidance in this respect and suggests avenues of future work both for virtual crowds and other embodied agents in VR. Larger user studies that consider additional aspects to support presence when interacting with virtual agents (such as personal space, gaze, or others), as well as motion-related behaviors at different scales (quick or subtle movements, or long-term motion planning), can add to the state of the art in creating pleasant and realistic experiences through virtual agent motion.

The works I present in Chapter 5 and Chapter 6 focus on high level routes that consider picking a general next direction or waypoint to travel to, and do not explicitly consider the motion when traveling between them. While many works consider the local problem of reaching nearby or directional goals (Yang et al., 2020), additional work is needed to couple these two motion scales in a way that explicitly maintains the realism of both. Hierarchical methods that enable a form of multi-scale planning is a promising direction for this purpose.

Specifically in the context of simulating human browsing behavior (as I studied in the context of shopping in Chapter 5), our data contained many slow-spots that were not aligned with item purchases. This suggests further analysis could help discover different elements that drive this part of the observed behavior. Other types of analysis, such as Fourier analysis or methods for studying self-similarity, may help identify classes of behavior for providing structured variety in browsing simulations when coupled with unsupervised learning methods.

In both settings, my work on global navigation considers indoor environments. Additional research is needed to verify whether these methods extend well in other settings such as outdoors or in non-human designed spaces. In the SPNets work,

the information available to the agent is a sampling of ray lengths that terminate at obstacles, but does not provide much in the way of semantic information. Other recent deep learning methods have been used in the context of global navigation in robotics and virtual settings, using visual information such as images from on-board cameras or renderings of imaged environments (Pfeiffer et al., 2017; Wu et al., 2018; Pfeiffer et al., 2018; Gupta et al., 2017). This allows for a richer set of inputs and more semantic forms of planning (such as looking for a certain type of door or flooring to identify promising directions given a semantic goal such as a kitchen or bathroom). An investigation as to how the inclusion of such information can support increased realism from a human motion simulation perspective is an interesting future direction.

7.4.2 Larger Challenges

While these avenues relate closely to the work presented in this dissertation, they exist as steps toward addressing the larger scale challenges of bringing an understanding of human motion to bear on AI and robotics solutions.

The ability to reason about the internal state of humans without the need to explicitly solicit such information is a critical element for solutions that must gauge and appropriately respond to the impact they have on the human experience in real-time. For example, it would be impractical for a robot navigating through a crowd to collect questionnaires from people nearby to measure whether its motion is disruptive or causing unease, and making adjustments when necessary as it moves along. Rather, gleaning this information based on external measures such as trajectories enables a more real-time contentiousness. The work in Chapter 4 serves as promising initial evidence that this is possible, as it shows captured motion data can be used to draw insight about someone's internal state. Future work can both expand upon the particular technique I propose and broaden the tool set for drawing internal insights from motion.

I have demonstrated in this dissertation multiple examples of using motion data to learn models of human motion. With any data-driven process comes the limitations inherent in the data that is used, human motion data being no exception. One of the main resulting challenges is that data-driven models in general do not distinguish between inputs that are similar to those seen in training (and are likely to produce high quality output) and those that are especially novel. The result is a tendency to extrapolate from the training data, in some cases indefinitely beyond the reasonable region the data supports, resulting in poor performance or biasing. This is particularly true for parametric models such as neural networks when asked to predict or classify on data very different from that seen in training. While one solution may be to simply increase the breadth of the dataset (and potentially the complexity of the model), practical constraints in application domains can be prohibitive of such an approach. Future research that looks to address this challenge may include those that move forward the state of the art in techniques such as outlier detection to identify when a particular query is not likely to be supported by a model, or new kinds of models that focus on supporting graceful degradation of quality.

One final challenge that I will identify here for the future of using motion data to build realistic models of human motion is grappling with the myriad of different types of motion on different scales in which humans engage. As is already apparent from the types of motion considered in my work, from quick and subtle movements of the mouth to larger scale navigation decisions, there are many levels at which opportunities exist for furthering our understanding of how and why humans move. For application domains such as creating realistic virtual characters, a multi-faceted understanding of human motion on multiple levels is central to increasing the quality of the resulting simulations. Future work that looks to move towards more holistic modeling of human motion may include "bottom-up" approaches that produce these multi-scale phenomenon emergently, or those that employ an ensemble of methods

that specialize for each type of motion relevant for the domain.

7.5 Conclusion

The body of work contained in this dissertation serves as a cohesive exploration into the effectiveness of data-driven approaches for both the simulation and validation of natural, varied human behaviors, as well as the importance of incorporating these aspects in several application domains. As a result, this dissertation represents concrete progress towards understanding and incorporating the human motion element into the fields of AI for graphics, robotics, and others such as medicine and psychology. It is my sincere hope that the frameworks, datasets, and simulation techniques presented here will continue to contribute towards a basis for a more social form of intelligence.

References

Astrid, M., Krämer, N. C., and Gratch, J. (2010). How our personality shapes our interactions with virtual characters-implications for research and development. In *International Conference on Intelligent Virtual Agents*, pages 208–221. Springer.

Atzmon, D., Stern, R., Felner, A., Sturtevant, N. R., and Koenig, S. (2020). Probabilistic robust multi-agent path finding. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 30, pages 29–37.

Bailenson, J. N., Aharoni, E., Beall, A. C., Guadagno, R. E., Dimov, A., and Blascovich, J. (2004). Comparing behavioral and self-report measures of embodied agents' social presence in immersive virtual environments. In *Proceedings of the 7th Annual International Workshop on PRESENCE*.

Bailenson, J. N., Blascovich, J., Beall, A. C., and Loomis, J. M. (2003). Interpersonal distance in immersive virtual environments. *Personality and Social Psychology Bulletin*, 29(7):819–833.

Bailenson, J. N., Shum, M. S., and Uttal, D. H. (2000). The initial segment strategy: A heuristic for route selection. *Memory & Cognition*, 28(2):306–318.

Bailenson, J. N., Swinth, K., Hoyt, C., Persky, S., Dimov, A., and Blascovich, J. (2005). The independent and interactive effects of embodied-agent appearance and behavior on self-report, cognitive, and behavioral markers of copresence in immersive virtual environments. *Presence: Teleoperators and Virtual Environments*, 14(4):379–393.

Bansal, S., Tolani, V., Gupta, S., Malik, J., and Tomlin, C. (2020). Combining optimal control and learning for visual navigation in novel environments. In *Conference on Robot Learning*, pages 420–429.

Bartlett, M. S., Littlewort, G., Frank, M., Lainscsek, C., Fasel, I., and Movellan, J. (2005). Recognizing facial expression: machine learning and application to spontaneous behavior. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 568–573. IEEE.

Botea, A., Bouzy, B., Buro, M., Bauckhage, C., and Nau, D. (2013). Pathfinding in games. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Bouaziz, S., Wang, Y., and Pauly, M. (2013). Online modeling for realtime facial animation. *ACM Transactions on Graphics (TOG)*, 32(4):40.

Brenton, H., Gillies, M., Ballin, D., and Chatting, D. (2005). The uncanny valley: does it exist. In *Proceedings of conference of human computer interaction, workshop on human animated character interaction*. Citeseer.

Bruneau, J., Olivier, A.-H., and Pettre, J. (2015). Going through, going around: A study on individual avoidance of groups. *IEEE transactions on visualization and computer graphics*, 21(4):520–528.

Brunetti, A., Buongiorno, D., Trotta, G. F., and Bevilacqua, V. (2018). Computer vision and deep learning techniques for pedestrian detection and tracking: A survey. *Neurocomputing*, 300:17–33.

Bulitko, V. (2004). Learning for adaptive real-time search. *arXiv preprint cs/0407016*.

Bulitko, V. and Björnsson, Y. (2009). knn lrta*: Simple subgoaling for real-time search. In *AIIDE*.

Cai, L., Gao, H., and Ji, S. (2019). Multi-stage variational auto-encoders for coarse-to-fine image generation. In *Proceedings of the 2019 SIAM International Conference on Data Mining*, pages 630–638. SIAM.

Camargo, C. Q., Bright, J., and Hale, S. A. (2019). Diagnosing the performance of human mobility models at small spatial scales using volunteered geographical information. *Royal Society open science*, 6(11):191034.

Chaplot, D. S., Gandhi, D., Gupta, S., Gupta, A., and Salakhutdinov, R. (2020). Learning to explore using active neural slam. *arXiv preprint arXiv:2004.05155*.

Charalambous, P. and Chrysanthou, Y. (2014). The pag crowd: A graph based approach for efficient data-driven crowd simulation. In *Computer Graphics Forum*, volume 33, pages 95–108. Wiley Online Library.

Chollet, F. et al. (2015). Keras.

Cong, M., Bao, M., Bhat, K. S., Fedkiw, R., et al. (2015). Fully automatic generation of anatomical face simulation models. In *Proceedings of the 14th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 175–183. ACM.

Costigan, T., Prasad, M., and McDonnell, R. (2014). Facial retargeting using neural networks. In *Proceedings of the Seventh International Conference on Motion in Games*, MIG '14, pages 31–38, New York, NY, USA. ACM.

Davis, B., Sohre, N., and Guy, S. J. (2018). Multiworld motion planning. *IEEE Robotics and Automation Letters*, 3(4):3968–3974.

Doersch, C. (2016). Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*.

Dong, Y. (2019). Assessing Dynamic Qualities of Emotional Expressions in Faces Using a Neural Network. Master's thesis, University of Minnesota, Minneapolis, MN USA.

Duckham, M. and Kulik, L. (2003). "simplest" paths: automated route selection for navigation. In *International Conference on Spatial Information Theory*, pages 169–185. Springer.

Dutra, T. B., Marques, R., Cavalcante-Neto, J. B., Vidal, C. A., and Pettré, J. (2017). Gradient-based steering for vision-based crowd simulation algorithms. In *Computer graphics forum*, volume 36, pages 337–348. Wiley Online Library.

Ekman, P. and Friesen, W. V. (1977). Facial action coding system.

Epstein, R. A., Patai, E. Z., Julian, J. B., and Spiers, H. J. (2017). The cognitive map in humans: spatial navigation and beyond. *Nature neuroscience*, 20(11):1504.

Eslami, S. A., Rezende, D. J., Besse, F., Viola, F., Morcos, A. S., Garnelo, M., Ruderman, A., Rusu, A. A., Danihelka, I., Gregor, K., et al. (2018). Neural scene representation and rendering. *Science*, 360(6394):1204–1210.

Essa, I. A. and Pentland, A. P. (1997). Coding, analysis, interpretation, and recognition of facial expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):757–763.

Fan, T., Cheng, X., Pan, J., Manocha, D., and Yang, R. (2018). Crowdmove: Autonomous mapless navigation in crowded scenarios. *arXiv preprint arXiv:1807.07870*.

Farley, J. U. and Ring, L. W. (1966). A stochastic model of supermarket traffic flow. *Operations Research*, 14(4):555–567.

Fitts, P. M. and Peterson, J. R. (1964). Information capacity of discrete motor responses. *Journal of experimental psychology*, 67(2):103.

Franco, L. and Treves, A. (2001). A neural network facial expression recognition system using unsupervised local processing. In *Image and Signal Processing and Analysis, 2001. ISPA 2001. Proceedings of the 2nd International Symposium on*, pages 628–632. IEEE.

Geller, T. (2008). Overcoming the uncanny valley. *IEEE computer graphics and applications*, 28(4):11–17.

Gosselin, F. and Schyns, P. G. (2001). Bubbles: a technique to reveal the use of information in recognition tasks. *Vision research*, 41(17):2261–2271.

Griesser, R. T., Cunningham, D. W., Wallraven, C., and Bülthoff, H. H. (2007). Psychophysical investigation of facial expressions using computer animated faces. In *Proceedings of the 4th symposium on Applied perception in graphics and visualization*, pages 11–18. ACM.

Gupta, S., Davidson, J., Levine, S., Sukthankar, R., and Malik, J. (2017). Cognitive mapping and planning for visual navigation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2616–2625.

Gutiérrez-Roig, M., Sagarra, O., Oltra, A., Palmer, J. R., Bartumeus, F., Diaz-Guilera, A., and Perelló, J. (2016). Active and reactive behaviour in human mobility: the influence of attraction points on pedestrians. *Royal Society open science*, 3(7):160177.

Hall, E. T. et al. (1959). *The silent language*, volume 3. Doubleday New York.

Hanson, D., Olney, A., Prilliman, S., Mathews, E., Zielke, M., Hammons, D., Fernandez, R., and Stephanou, H. (2005). Upending the uncanny valley. In *AAAI*, volume 5, pages 1728–1729.

Helbing, D. and Molnar, P. (1995). Social force model for pedestrian dynamics. *Physical review E*, 51(5):4282.

Helwig, N. E., Sohre, N. E., Ruprecht, M. R., Guy, S. J., and Lyford-Pike, S. (2017). Dynamic properties of successful smiles. *PloS one*, 12(6):e0179708.

Hendrikx, M., Meijer, S., Van Der Velden, J., and Iosup, A. (2013). Procedural content generation for games: A survey. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 9(1):1.

Hernández, C. and Meseguer, P. (2005). Lrta*(k). In *Proceedings of the 19th international joint conference on Artificial intelligence*, pages 1238–1243.

Hildreth, D. and Guy, S. J. (2019). Coordinating multi-agent navigation by learning communication. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 2(2):1–17.

Hui, S. K., Fader, P. S., and Bradlow, E. T. (2009a). Path data in marketing: An integrative framework and prospectus for model building. *Marketing Science*, 28(2):320–335.

Hui, S. K., Fader, P. S., and Bradlow, E. T. (2009b). Research note—the traveling salesman goes shopping: The systematic deviations of grocery paths from tsp optimality. *Marketing science*, 28(3):566–572.

Hutton, C., Sohre, N., Davis, B., Guy, S., and Rosenberg, E. S. (2019). An augmented reality motion planning interface for robotics. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 1313–1314. IEEE.

Joshi, P., Tien, W. C., Desbrun, M., and Pighin, F. (2005). Learning controls for blend shape based realistic facial animation. In *ACM SIGGRAPH 2005 Courses*, page 8. ACM.

Kallmann, M. and Kapadia, M. (2014). Navigation meshes and real-time dynamic planning for virtual worlds. In *ACM SIGGRAPH 2014 Courses*, pages 1–81.

Kapadia, M., Pelechano, N., Allbeck, J., and Badler, N. (2015). Virtual crowds: Steps toward behavioral realism. *Synthesis lectures on visual computing: computer graphics, animation, computational photography, and imaging*, 7(4):1–270.

Kaplan, R., Schuck, N. W., and Doeller, C. F. (2017). The role of mental maps in decision-making. *Trends in Neurosciences*, 40(5):256–259.

Karamouzas, I., Skinner, B., and Guy, S. J. (2014). Universal power law governing pedestrian interactions. *Physical review letters*, 113(23):238701.

Karamouzas, I., Sohre, N., Hu, R., and Guy, S. J. (2018). Crowd space: a predictive crowd analysis technique. *ACM Transactions on Graphics (TOG)*, 37(6):1–14.

Karamouzas, I., Sohre, N., Hu, R., and Guy, S. J. (2019). Crowd space: a predictive crowd analysis technique. *ACM Transactions on Graphics (TOG)*, 37(6):186.

Karamouzas, I., Sohre, N., Narain, R., and Guy, S. J. (2017). Implicit crowds: Optimization integrator for robust crowd simulation. *ACM Transactions on Graphics (TOG)*, 36(4):1–13.

Karatzoglou, A., Smola, A., Hornik, K., and Zeileis, A. (2004). kernlab – an S4 package for kernel methods in R. *Journal of Statistical Software*, 11(9):1–20.

Kavraki, L. E., Svestka, P., Latombe, J.-C., and Overmars, M. H. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE transactions on Robotics and Automation*, 12(4):566–580.

Kennedy, R. S., Lane, N. E., Berbaum, K. S., and Lilienthal, M. G. (1993). Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness. *The international journal of aviation psychology*, 3(3):203–220.

Kim, S., Guy, S. J., Hillesland, K., Zafar, B., Gutub, A. A.-A., and Manocha, D. (2015). Velocity-based modeling of physical interactions in dense crowds. *The Visual Computer*, 31(5):541–555.

Koenig, S. and Likhachev, M. (2002). D^* lite. *Aaai/iaai*, 15.

Koenig, S. and Sun, X. (2009). Comparing real-time and incremental heuristic search for real-time situated agents. *Autonomous Agents and Multi-Agent Systems*, 18(3):313–341.

Köhn, H.-F. (2006). Combinatorial individual differences scaling within the city-block metric. *Computational statistics & data analysis*, 51(2):931–946.

Kokkinara, E. and McDonnell, R. (2015). Animation realism affects perceived character appeal of a self-virtual face. In *Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games*, MIG '15, pages 221–226, New York, NY, USA. ACM.

Korf, R. E. (1990). Real-time heuristic search. *Artificial intelligence*, 42(2-3):189–211.

Kotsiantis, S. B., Zaharakis, I., and Pintelas, P. (2007). Supervised machine learning: A review of classification techniques.

Kyriakou, M., Pan, X., and Chrysanthou, Y. (2016). Interaction with virtual crowd in immersive and semi-immersive virtual reality systems. *Computer Animation and Virtual Worlds*, pages n/a–n/a. CAVW-16-0002.R1.

Larson, J. S., Bradlow, E. T., and Fader, P. S. (2005). An exploratory look at supermarket shopping paths. *International Journal of research in Marketing*, 22(4):395–414.

LaValle, S. M. (1998). Rapidly-exploring random trees: A new tool for path planning.

Lee, S., Park, M., Lee, K., and Lee, J. (2019). Scalable muscle-actuated human simulation and control. *ACM Transactions on Graphics (TOG)*, 38(4):1–13.

Lee, W. and Lawrence, R. (2013). Fast grid-based path finding for video games. In *Canadian Conference on Artificial Intelligence*, pages 100–111. Springer.

Lee, Y., Terzopoulos, D., and Waters, K. (1995). Realistic modeling for facial animation. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 55–62. ACM.

Li, H., Yu, J., Ye, Y., and Bregler, C. (2013). Realtime facial animation with on-the-fly correctives. *ACM Trans. Graph.*, 32(4):42–1.

Liapis, A., Yannakakis, G. N., and Togelius, J. (2015). Constrained novelty search: A study on game content generation. *Evolutionary computation*, 23(1):101–129.

Liaw, A. and Wiener, M. (2002). Classification and regression by randomforest. *R News*, 2(3):18–22.

Lima, A., Stanojevic, R., Papagiannaki, D., Rodriguez, P., and González, M. C. (2016). Understanding individual routing behaviour. *Journal of The Royal Society Interface*, 13(116):20160021.

Lin, Y. and Jeon, Y. (2006). Random forests and adaptive nearest neighbors. *Journal of the American Statistical Association*, 101(474):578–590.

Liu, K., Tolins, J., Tree, J. E. F., Neff, M., and Walker, M. A. (2016). Two techniques for assessing virtual agent personality. *IEEE Transactions on Affective Computing*, 7(1):94–105.

Llobera, J., Spanlang, B., Ruffini, G., and Slater, M. (2010). Proxemics with multiple dynamic characters in an immersive virtual environment. *ACM Transactions on Applied Perception (TAP)*, 8(1):3.

Long, P., Fanl, T., Liao, X., Liu, W., Zhang, H., and Pan, J. (2018). Towards optimally decentralized multi-robot collision avoidance via deep reinforcement learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6252–6259. IEEE.

Lozano-Pérez, T. and Wesley, M. A. (1979). An algorithm for planning collision-free paths among polyhedral obstacles. *Communications of the ACM*, 22(10):560–570.

Luo, Y.-S., Soeseno, J. H., Chen, T. P.-C., and Chen, W.-C. (2020). Carl: Controllable agent with reinforcement learning for quadruped locomotion. *ACM Trans. Graph.*, 39(4).

Lyford-Pike, S., Helwig, N. E., Sohre, N. E., Guy, S. J., and Hadlock, T. A. (2018). Predicting perceived disfigurement from facial function in patients with unilateral paralysis. *Plastic and reconstructive surgery*, 142(5):722e–728e.

Marsella, S., Xu, Y., Lhommet, M., Feng, A., Scherer, S., and Shapiro, A. (2013). Virtual character performance from speech. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 25–35. ACM.

Massara, F., Melara, R. D., and Liu, S. S. (2014). Impulse versus opportunistic purchasing during a grocery shopping experience. *Marketing letters*, 25(4):361–372.

McDonnell, R. (2012). *Appealing Virtual Humans*, pages 102–111. Springer Berlin Heidelberg, Berlin, Heidelberg.

McDonnell, R., Larkin, M., Dobbyn, S., Collins, S., and O'Sullivan, C. (2008). Clone attack! perception of crowd variety. In *ACM SIGGRAPH 2008 papers*, volume 27, pages 1–8.

Michel, P. and El Kaliouby, R. (2003). Real time facial expression recognition in video using support vector machines. In *Proceedings of the 5th international conference on Multimodal interfaces*, pages 258–264. ACM.

Mori, M., MacDorman, K. F., and Kageki, N. (2012). The uncanny valley [from the field]. *IEEE Robotics & Automation Magazine*, 19(2):98–100.

Mouret, J.-B. and Clune, J. (2015). Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909*.

Moyer, R. S. and Landauer, T. K. (1967). Time required for judgements of numerical inequality. *Nature*, 215(5109):1519–1520.

Musse, S. R. and Thalmann, D. (2001). Hierarchical model for real time simulation of virtual human crowds. *IEEE Transactions on Visualization and Computer Graphics*, 7(2):152–164.

Narang, S., Best, A., Randhavane, T., Shapiro, A., and Manocha, D. (2016). Pedvr: simulating gaze-based interactions between a real user and virtual crowds. In *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology*, pages 91–100. ACM.

Noh, H., You, T., Mun, J., and Han, B. (2017). Regularizing deep neural networks by noise: Its interpretation and optimization. In *Advances in Neural Information Processing Systems*, pages 5109–5118.

Nusseck, M., Cunningham, D. W., Wallraven, C., and Bülthoff, H. H. (2008). The contribution of different facial regions to the recognition of conversational expressions. *Journal of vision*, 8(8):1–1.

Ondřej, J., Pettré, J., Olivier, A.-H., and Donikian, S. (2010). A synthetic-vision based steering approach for crowd simulation. *ACM Transactions on Graphics (TOG)*, 29(4):1–9.

OptiTrack (2019). Optitrack-motion capture systems.

O'Sullivan, C. (2009). Variety is the spice of (virtual) life. In *International Workshop on Motion in Games*, pages 84–93. Springer.

Pantic, M. and Rothkrantz, L. J. (2000). An expert system for recognition of facial actions and their intensity. In *AAAI/IAAI*, pages 1026–1033.

Pelechano, N., Allbeck, J. M., Kapadia, M., and Badler, N. I. (2016). Simulating heterogeneous crowd with interactive behaviors.

Pelechano, N., Stocker, C., Allbeck, J., and Badler, N. (2008). Being a part of the crowd: towards validating vr crowds using presence. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 1*, pages 136–142. International Foundation for Autonomous Agents and Multiagent Systems.

Peng, X. B., Abbeel, P., Levine, S., and van de Panne, M. (2018). Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions on Graphics (TOG)*, 37(4):1–14.

Peters, C. and O'Sullivan, C. (2002). Synthetic vision and memory for autonomous virtual humans. *Computer Graphics Forum*, 21(4):743–752.

Pfeiffer, M., Schaeuble, M., Nieto, J., Siegwart, R., and Cadena, C. (2017). From perception to decision: A data-driven approach to end-to-end motion planning for autonomous ground robots. In *2017 ieee international conference on robotics and automation (icra)*, pages 1527–1533. IEEE.

Pfeiffer, M., Shukla, S., Turchetta, M., Cadena, C., Krause, A., Siegwart, R., and Nieto, J. (2018). Reinforced imitation: Sample efficient deep reinforcement learning for mapless navigation by leveraging prior demonstrations. *IEEE Robotics and Automation Letters*, 3(4):4423–4430.

Piovani, D., Arcaute, E., Uchoa, G., Wilson, A., and Batty, M. (2018). Measuring accessibility using gravity and radiation models. *Royal Society open science*, 5(9):171668.

Preuss, M., Liapis, A., and Togelius, J. (2014). Searching for good and diverse game levels. In *Computational Intelligence and Games (CIG), 2014 IEEE Conference on*, pages 1–8. IEEE.

Pugh, J. K., Soros, L. B., Szerlip, P. A., and Stanley, K. O. (2015). Confronting the challenge of quality diversity. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, pages 967–974. ACM.

R Core Team (2020). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Rabiee, S. and Biswas, J. (2019). Ivoa: Introspective vision for obstacle avoidance. In *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*. IEEE.

Riascos, A. P. and Mateos, J. L. (2012). Long-range navigation on complex networks using lévy random walks. *Physical Review E*, 86(5):056110.

Sieben, A., Schumann, J., and Seyfried, A. (2017). Collective phenomena in crowds—where pedestrian dynamics need social psychology. *PLoS one*, 12(6):e0177328.

Sifakis, E., Neverov, I., and Fedkiw, R. (2005). Automatic determination of facial muscle activations from sparse motion capture marker data. In *Acm transactions on graphics (tog)*, volume 24, pages 417–425. ACM.

Sifakis, E., Selle, A., Robinson-Mosher, A., and Fedkiw, R. (2006). Simulating speech with a physics-based facial muscle model. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 261–270. Eurographics Association.

Sinha, P., Balas, B., Ostrovsky, Y., and Russell, R. (2006). Face recognition by humans: Nineteen results all computer vision researchers should know about. *Proceedings of the IEEE*, 94(11):1948–1962.

Smith, G. (2014). The future of procedural content generation in games. In *Proceedings of the Experimental AI in Games Workshop*.

Sohre, N., Adeagbo, M., Helwig, N., Lyford-Pike, S., and Guy, S. J. (2018). Pvl: A framework for navigating the precision-variety trade-off in automated animation of smiles. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Sohre, N. and Guy, S. J. (2020). Spnets: Human-like navigation behaviors with uncertain goals. In *Motion, Interaction and Games*, pages 1–11.

Sohre, N., Mackin, C., Interrante, V., and Guy, S. J. (2017). Evaluating collision avoidance effects on discomfort in virtual environments. In *2017 IEEE Virtual Humans and Crowds for Immersive Environments (VHCIE)*, pages 1–5. IEEE.

Sohre, N., Wallis, A. O. G., and Guy, S. J. (2021). An information-theoretic law governing human multi-task navigation decisions.

Starke, S., Zhang, H., Komura, T., and Saito, J. (2019). Neural state machine for character-scene interactions. *ACM Transactions on Graphics (TOG)*, 38(6):1–14.

Starke, S., Zhao, Y., Komura, T., and Zaman, K. (2020). Local motion phases for learning multi-contact character movements. *ACM Transactions on Graphics (TOG)*, 39(4):54–1.

Stephan, C. N. (2003). Facial approximation: An evaluation of mouth-width determination. *American journal of physical anthropology*, 121(1):48–57.

Sturtevant, N. R. and Bulitko, V. (2011). Learning where you are going and from whence you came: h-and g-cost learning in real-time heuristic search. In *IJCAI*, pages 365–370. Citeseer.

Summerville, A., Snodgrass, S., Guzdial, M., Holmgård, C., Hoover, A. K., Isaksen, A., Nealen, A., and Togelius, J. (2017). Procedural content generation via machine learning (pcgml). *arXiv preprint arXiv:1702.00539*.

Sze, V., Chen, Y.-H., Yang, T.-J., and Emer, J. S. (2017). Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE*, 105(12):2295–2329.

Tamar, A., Wu, Y., Thomas, G., Levine, S., and Abbeel, P. (2016). Value iteration networks. In *Advances in Neural Information Processing Systems*, pages 2154–2162.

Togelius, J., Champandard, A. J., Lanzi, P. L., Mateas, M., Paiva, A., Preuss, M., and Stanley, K. O. (2013). Procedural content generation: Goals, challenges and actionable steps. In *Dagstuhl Follow-Ups*, volume 6. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Togelius, J., Yannakakis, G. N., Stanley, K. O., and Browne, C. (2011). Search-based procedural content generation: A taxonomy and survey. *IEEE Transactions on Computational Intelligence and AI in Games*, 3(3):172–186.

Van Den Berg, J., Guy, S. J., Lin, M., and Manocha, D. (2011). Reciprocal n-body collision avoidance. In *Robotics research*, pages 3–19. Springer.

van den Berg, J., Lin, M., and Manocha, D. (2008). Reciprocal velocity obstacles for real-time multi-agent navigation. In *2008 IEEE International Conference on Robotics and Automation*, pages 1928–1935. IEEE.

van Toll, W., Triesscheijn, R., Kallmann, M., Oliva, R., Pelechano, N., Pettré, J., and Geraerts, R. (2016). A comparative study of navigation meshes. In *Proceedings of the 9th International Conference on Motion in Games*, pages 91–100.

Vinayagamoorthy, V., Gillies, M., Steed, A., Tanguy, E., Pan, X., Loscos, C., and Slater, M. (2006). Building expression into virtual characters. *Eurographics*.

Wang, H., Ondřej, J., and O'Sullivan, C. (2016). Path patterns: Analyzing and comparing real and simulated crowds. In *Proceedings of the 20th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, pages 49–57.

Warren, W., Fajen, B., and Belcher, D. (2001). Behavioral dynamics of steering, obstacle avoidance, and route selection. *Journal of Vision*, 1(3):184–184.

Waters, K. (1987). A muscle model for animation three-dimensional facial expression. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '87, pages 17–24, New York, NY, USA. ACM.

Wolinski, D., Lin, M. C., and Pettré, J. (2016). Warpdriver: context-aware probabilistic motion prediction for crowd simulation. *ACM Transactions on Graphics (TOG)*, 35(6):164.

Wu, Y., Wu, Y., Gkioxari, G., and Tian, Y. (2018). Building generalizable agents with a realistic and rich 3d environment. *arXiv preprint arXiv:1801.02209*.

Xu, F., Chai, J., Liu, Y., and Tong, X. (2014). Controllable high-fidelity facial performance transfer. *ACM Transactions on Graphics (TOG)*, 33(4):42.

Yang, S., Li, T., Gong, X., Peng, B., and Hu, J. (2020). A review on crowd simulation and modeling. *Graphical Models*, 111:101081.

Yannakakis, G. N. and Togelius, J. (2011). Experience-driven procedural content generation. *IEEE Transactions on Affective Computing*, 2(3):147–161.

Ying, F., Wallis, A. O., Beguerisse-Díaz, M., Porter, M. A., and Howison, S. D. (2019). Customer mobility and congestion in supermarkets. *Physical Review E*, 100(6):062304.

Zhang, J., Yu, J., You, J., Tao, D., Li, N., and Cheng, J. (2016). Data-driven facial animation via semi-supervised local patch alignment. *Pattern Recognition*, 57:1–20.