

ATHABASCA UNIVERSITY

INERTIAL NAVIGATION SYSTEM USING AUGMENTED REALITY

TRANSFORMATION FOR CORRECTION AND COGNITION

BY

KENNETH MULDER

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE IN INFORMATION SYSTEMS

FACULTY OF SCIENCE AND TECHNOLOGY

ATHABASCA, ALBERTA

FEBRUARY, 2022

© KENNETH MULDER

**Approval Page**

The undersigned certify that they have read the thesis entitled

**INERTIAL NAVIGATION SYSTEM USING A MONOCULAR CAMERA AND  
FIDUCIAL MARKERS FOR CORRECTIONS AND SPATIAL COGNITION  
LEARNING TASKS**

Submitted by

**Kenneth Mulder**

In partial fulfillment of the requirements for the degree of

**Master of Science in Information Systems**

The thesis examination committee certifies that the thesis  
and the oral examination is approved

**Supervisor:**

Dr. Maiga Chang  
Athabasca University

**Committee Members:**

Dr. Larbi Esmahi  
Athabasca University

Dr. Mohamed Jemni  
University of Tunis

**External Examiner:**

Dr. Shein-Yung Cheng  
Chung-Yuan Christian University

December 20, 2021

## **Abstract**

The main goal of this research is to determine if an Inertial Navigation System (INS) can be corrected to effectively track a user's pose, position and orientation, by using a monocular camera with fiducial markers. In addition, this research determines if an augmented reality (AR) system interface that integrates spatial cognition learning tasks can test and train the user's spatial cognition. The implications of positive results are that the corrected INS poses on ubiquitous mobile devices provide reasonably accurate, precise, and reliable indoor navigation when a Global Navigation Satellite System (GNSS) is unavailable. This research investigates location-based services (LBS) for use in learning environments, tourism, and emergency services. The implementation technically enables the spatial cognition learning tasks. However, user tests results are still required to prove that user spatial cognition would improve and make navigating faster, which is a suggested area for future research.

*Keywords:* Path generation, Multiple travelling salesman problem, Contract net protocol, Distributed multi-agent systems, Calibration, Sensor fusion, Complementary filter, Augmented intelligence

## Table of Contents

Approval Page.....	ii
Abstract .....	iii
Table of Contents .....	iv
List of Tables .....	vi
List of Figures .....	viii
List of Symbols, Nomenclature, or Abbreviations .....	xi
Chapter I – Introduction .....	1
Motivation.....	1
Research Objectives.....	1
Research Issues .....	3
Thesis Organization .....	7
Chapter II – Spatial Cognition Learning Tasks .....	8
Scenario and Problems.....	8
Rationale .....	12
Chapter III –Spatial Cognition Learning Tasks Analysis and Design.....	23
Algorithms .....	23
Path Generation.....	23
Task Allocation.....	29
Calibration.....	33
Spatial Cognition .....	37
Navigation.....	41
Architecture.....	43
Workflow .....	47
Path Generation.....	48
Task Allocation.....	49
Calibration.....	51
Spatial Cognition .....	52
Navigation.....	53
Chapter IV –Spatial Cognition Learning Task Generation.....	55
Path Generation.....	56
Task Allocation.....	57
Calibration.....	58
Spatial Cognition .....	61
Navigation.....	64
Chapter V – Experiment and Discussion.....	68
Research Design and Hypotheses .....	68
Research Measurements.....	68
Hypotheses .....	70
Experiment Design.....	70

Data Collection .....	70
Performance Evaluation.....	71
Test Plan.....	72
Path Generation Test Scenario.....	72
Path Generation Test Phases .....	73
Task Allocation Test Scenario .....	73
Task Allocation Test Phases .....	74
Calibration Test Scenario.....	74
Calibration Test Phases .....	75
Spatial Cognition Test Scenario.....	76
Spatial Cognition Test Phases.....	76
Navigation Test Scenario.....	78
Navigation Test Phases .....	78
Data Analysis .....	79
Path Generation.....	80
Task Allocation .....	82
Calibration.....	89
Spatial Cognition .....	93
Navigation.....	97
Findings and Discussion .....	101
Path Generation.....	101
Task Allocation .....	102
Calibration.....	103
Spatial Cognition .....	104
Navigation.....	105
Chapter VI – Conclusions .....	108
Possible Contributions .....	108
Summary and Future Works .....	111
References .....	113
Appendix A: Gyro Allan Standard Deviation R Plot Script (Phidgets, 2017).....	122
Appendix B: Accelerometer Allan Standard Deviation R Plot Script (Phidgets, 2017) .....	123
Appendix C: XML Log File .....	125
Appendix D: Google Maps and AEGN Trip Plans.....	130
Appendix E: Test Scenarios.....	134
Appendix F: Path Generation Test Phases.....	136
Appendix G: Task Allocation Test Phases .....	139
Appendix H: Calibration Test Phases .....	143
Appendix I: Spatial Cognition Test Phases .....	148
Appendix J: Navigation Test Phases.....	153
Appendix K: Agents .....	156

Appendix L: IPIN 2018 Competition Results.....	160
Appendix M: Certification of Ethical Approval .....	162

**List of Tables**

Table 1 Tetrad for Hand-Held Tour Devices With AEGN .....	15
Table 2 SCORE View Properties.....	65
Table 3 IOF Symbols With Relative and Cardinal Text Direction Examples (IOF, 2018)....	65
Table 4 Google Maps (Google, n.d. b) and AEGN Path Distances and Times at Various Speeds .....	82
Table 5 Task Allocation Example Using CNP With CTM Calculations for Various Preferences .....	85
Table 6 Task Allocation Computational Complexity .....	87
Table 7 Performance Grades of Inertial Sensors (VectorNav, 2020, p.5) .....	89
Table 8 Error Terms by Sensor/Grade .....	92
Table 9 ART Pose and INS Errors at Finish Control Points.....	95
Table 10 Processing Time on Quaternion Generation (quaternion/s) (Michel, 2015, p.10) .....	97
Table 11 Sensors' Biases and Noises Consideration in Each Algorithm (Michel, 2015, p.7) .....	98
Table 12 INS, Gyro and GNSS Errors on Paths With a Travel Time Greater Than 180s.....	98
Table 13 Error Over Time by Sensor Grade .....	99
Table D1 Google Maps (Google, n.d. b) and AEGN Path Descriptions From Mcdonald's to Gogi .....	131
Table D2 Google Maps (Google, n.d. b) and AEGN Path Descriptions From Gogi to Stavro's.....	131
Table D3 Google Maps (Google, n.d. b) and AEGN Path Descriptions From Stavro's to Velvet.....	132
Table D4 Google Maps (Google, n.d. b) and AEGN Path Descriptions From Velvet to	

John's.....	133
Table D5 Google Maps (Google, n.d. b) and AEGN Path Descriptions From John's to 4 <sup>th</sup>	
Spot.....	133
Table E1 Test Scenarios.....	134
Table F1 AU Main Campus Manual Test Points.....	138
Table G1 4 Users Group Test Combinations.....	142
Table H1 Complementary Filter Tuning Tests (Distance to Drift Error > 3m at 1m/s) .....	147
Table J1 The Speeds/Paths with a Travel Time Greater than 180s .....	155
Table K1 Agent Types .....	158
Table K2 Agent Descriptions.....	159
Table L1 IPIN 2018 Competition Results with Reported Descriptions (Renaudin et al., 2019).....	160

## List of Figures

Figure 1 Uniform Field of View (UFOV) Task: Fixation (699ms), Stimulus (30ms), Mask (600ms) and Response.....	17
Figure 2 Mental Rotation Task (MRT) .....	18
Figure 3 Planar Fiducial Marker and Planar Art as a Fiducial Marker .....	19
Figure 4 Planar Fiducial Marker Constellation and Planar Art as a Fiducial Marker Constellation.....	19
Figure 5 Unweighted Areas Outside (Green) Make the Shortest Path Outside.....	25
Figure 6 Weighted Areas Outside (Green) Make the Fastest Path Inside (Pink) .....	25
Figure 7 Red Navmesh Wheelchair Path From Start Triangle to Finish Double Circle.....	26
Figure 8 Unity NavMesh Agent Properties.....	27
Figure 9 The Schedule Information Includes the Hours Open (Attend), the Worst Times (Avoid) and the Best Time (Assign).....	31
Figure 10 Jason Agent CNP Communications Example Where the Leader, a Teacher, Invites Student Users to Join a Group for Task Allocation.....	33
Figure 11 User-Conducted Calibration Maneuver (Martin, 2016, P. 92) .....	34
Figure 12 Gyro Allan Deviation Analysis (IEEE, 1998, p. 71).....	35
Figure 13 Unified Agent-Oriented Software Engineering Methodology (Dam & Winikoff, 2013, p. 687).....	44
Figure 14 Jade Methodology (Nikraz et al., 2006, p. 5) .....	45
Figure 15 AEGN DMAS Architecture .....	46
Figure 16 AEGN DMAS Workflow .....	48
Figure 17 Path Generation .....	49

Figure 18 Task Allocation .....	50
Figure 19 User Calibration.....	51
Figure 20 User Spatial Cognition .....	52
Figure 21 User Navigation .....	54
Figure 22 Screenshots of AEGN System .....	55
Figure 23 Screenshots of Path Generation .....	56
Figure 24 Screenshots of Task Allocation .....	58
Figure 25 Screenshot of Calibration .....	59
Figure 26 Screenshots of ART Pose and Spatial Cognition .....	62
Figure 27 Screenshot of User Spatial Cognition Learning Tasks .....	63
Figure 28 Screenshots of ART Pose and Navigation .....	64
Figure 29 Third (Allo) View .....	66
Figure 30 First (Ego) View.....	66
Figure 31 Top Forward (Geo) View .....	66
Figure 32 Top North (Geo) View.....	66
Figure 33 Top North (Geo) View With Straight Path.....	66
Figure 34 Distance and Direction, Direction and Compass Overview Examples.....	67
Figure 35 Scoring Fields for Answer Keys (Red Boxes), Answers (Yellow Boxes) and Scores (Blue Boxes) .....	77
Figure 36 Error Bubbles for GNSS (Red), INS (Green), Gyro (Blue) and Reference Agent (Green Hexagon) .....	80
Figure 37 Unity Agent Running Fast (4m/s) and Turning a 90° Corner .....	81
Figure 38 Unity Agent Walking Slowly (0.83m/s) and Turning a 90° Corner .....	81

Figure 39 Unity Agent Running Fast (4m/s), Stopping, and Turning a 90° Corner.....	81
Figure 40 Two User Group and One User Group Task Allocation Examples .....	83
Figure 41 Complexity Classification .....	88
Figure 42 Gyro Allan Standard Deviation .....	90
Figure 43 Accelerometer Allan Standard Deviation.....	91
Figure 44 Calibration Settings .....	92
Figure 45 Google Pedestrian Photosphere (Google, n.d. a) and Maps (Google, n.d. b).....	94
Figure 46 Best Fit Lines Graphing Error Over Time by Sensor Grade (Table 13) .....	99
Figure D1 AEGN Map Path (4.0m/s) .....	130
Figure D2 AEGN Map Path (0.83m/s) .....	130
Figure D3 Google Maps Path (Google, n.d. b) .....	130
Figure H1 Walking Velocity Fluctuations (Renaudin et al., 2019, p. 148604) .....	147
Figure K1 AMUSE Platform, System Containers (Green Nodes) and Service Agents Which Run Continuously .....	156
Figure K2 AMUSE Platform, Back-End Coordinator AMUSE Containers (Green Nodes) and Agents Which Run When the App is Active .....	157
Figure K3 AMUSE Platform, Back-End Coordinator Jason Containers (Green Nodes) and Agents Which Run When the Task Allocation is Active .....	157
Figure K4 AMUSE Platform, Back-End Communicator JADE Containers (Green Nodes) and Agents Which Run When the App is Active .....	157

**List of Symbols, Nomenclature, or Abbreviations**

Abbreviation	Expanded description
$a$	complementary filter coefficient
A*	A Star algorithm
AEGN	Allo-Ego-Geo-centric Navigation
Allo	third-person reference frame view
AMUSE	Agent-based Multi-User Social Environment
ANAV	Allo-centric NAVigation
AOSE	Agent-Oriented Software Engineering
ART	Augmented Reality Transformation
ARW	Angle Random Walk noise
AU	Athabasca University
BI	Bias Instability
CNP	Contract Net Protocol
DMAS	Distributed Multi-Agent System
$dt$	complementary filter time interval
Ego	first person reference frame view
ENAV	Ego-centric NAVigation
$f$	Accelerometer acceleration
FOV	Field Of View
Geo	top (down) forward or top (down) north reference frame view
Gyro	Gyroscope
IA	Augmented Intelligence
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
IOF	International Orienteering Federation
IPIN	Indoor Positioning and Indoor Navigation
JADE	Java Agent Development Environment
Jason	Java-based Agentspeak interpreter used with SACI for multi-agent distribution Over the Net
LBS	Location-Based Services
MAE	Mean Absolute Error
MOL	Method Of Loci
MP	Mount Pleasant
MRT	Mental Rotation Task
mTSP	Multiple Travelling Salesman Problem
NavMesh	Navigation Mesh
NP	Non-deterministic Polynomial time
$\omega$	Gyroscope rotation
PCD	Path Control Description
PITF	Path Integration Task to the Finish
PITS	Path Integration Task to the Start
SCORE	Spatial COgnition RatE
SLAM	Simultaneous Localization and Mapping
$\tau$	complementary filter time constant
TSP	Travelling Salesman Problem
UFOV	Uniform Field Of View
VRW	Velocity Random Walk noise
WADE	Workflows and Agents Development Environment
WOLF	WORkFLOw engine

## **Chapter I – Introduction**

This research focuses on an inertial navigation system using a monocular camera and fiducial markers. The first section of chapter I discusses the motivation for this research. Then the second, third, and fourth sections of chapter I describe the research objectives, issues, and explain the structure of this thesis, respectively.

### **Motivation**

The main topic of this study determines if an INS can be corrected to effectively pose, position and orientation, using a monocular camera with fiducial markers. In addition, this study determines if an AR system interface that integrates spatial cognition learning tasks can test and train the user's spatial cognition to improve system performance. The implications of positive results are that the corrected INS poses on ubiquitous mobile devices provide reasonably accurate, precise, and reliable indoor navigation when a GNSS is unavailable. Positive results indicate that indoor navigation is possible and can increase widespread adoption if planar art is used as fiducial markers to support corrections of cumulative INS errors. This low-cost design could lead to a commercially viable solution that would be especially competitive during an economic recession when it is difficult to rationalize expensive hardware and software. This research investigates location-based services (LBS) in learning environments, tourism, and emergency services.

### **Research Objectives**

This research investigates the following four objectives to overcome the challenges of navigating interior spaces using mobile devices.

The first objective is to enable navigating indoors using a hand-held android device. Wu et al. (2011) designed a mobile learning application using the current landmark or allo-centric navigation (ANAV) system for mobile devices with neither a built-in GNSS receiver nor a

compass. Wu et al. suggested the four primary issues to enhance the functionality and increase the application's performance are the shortest path, object shaping, obstacle avoidance and visibility.

The path generation algorithm addresses these issues, but rather than the shortest path, this algorithm calculates the fastest path because route speeds vary. This makes navigating in the tight confines of interior spaces more effective. Navigating indoors uses an INS with a camera and fiducial markers for corrections. This research develops a system to calculate the user's pose based on INS readings. The accuracy and precision of the pose are intermittently calculated to compare with other poses calculated from markers scanned with a camera. The scanned pose is used to correct the current pose, and path integration is continued from that point.

The second objective is to identify map discrepancies using a Project Tango Tablet. Simultaneous Localization and Mapping (SLAM) can quickly build a three-dimensional (3D) spatial model that can then be used to define landmarks for user reference corrections (Gurau & Nuchter, 2013). The Mantis Vision MV4D core 3D engine and depth sensing platform used in the Tango tablet (Galan, 2015) does not have depth range of survey quality lidar sensors, but it is adequate for small indoor spaces with a range of 0.5m-4.5m (Mantis Vision, 2013) and it is both affordable and compact. The scanned point cloud is used to help identify landmark discrepancies that are corrected and abstracted using universally standardized symbology to represent features.

The third objective is to personalize the guided tour by adapting the model for each user. For instance, the navigation messages convert the distance into the number of steps for that user depending on their stride. As well, visibility analysis is calculated based on the users' height. If a user is in a wheelchair, then they are guided along wheelchair accessible paths.

The fourth objective is spatial cognition learning tasks that should make navigating faster. Users are less dependent on the system if they are trained to have a higher spatial cognition. They

understand it quicker and refer to it less frequently, navigating faster.

### **Research Issues**

Quantitative analysis is used to evaluate the accuracy, precision, and error rate of the Allo-Ego-Geo-centric Navigation (AEGN) system posing. The spatial cognition tests are like the tests run by Feng et al. (2007). To achieve these features, this study seeks to answer the following research issues using a distributed multi-agent system (DMAS) (Appendix K).

#### **Technology Objective 1: Navigate Indoors Using a Hand-Held Android Device.**

**Issue 1: Fastest Path.** ANAV only considers the directions using two-dimensional (2D) situated objects. The system lacks performance because it does not calculate the fastest path to the destination. This research uses a four-dimensional (4D) spatial, temporal model to determine the optimal path.

**Issue 2: Object Shaping.** The object shaping issue involves how to calculate the navigation path around curved lines and non-quadrilateral shaped objects. The current system cannot handle complex lines and shapes. This research utilizes more complex geometry and topology when calculating 3D navigation paths to increase the efficiency of the suggested path. This research investigates running complex geometric routing using a Unity agent (Table K1, Table K2).

**Issue 3: Obstacle Avoidance.** The obstacle avoidance issue is generalized to address how to avoid obstacles. Areas may be impassable because there are hazards (e.g., fire), closures (e.g., construction), weather (e.g., severe storms) or restrictions (e.g., security). This research does not investigate automatically determining which areas are impassable, but if they are manually set, then the MAS avoids them when recommending a path. For example, if all entry/exit points are assigned a wheelchair accessible attribute, then wheelchair users only use the accessible points, and the others are ignored as obstacles.

**Issue 4: Visibility.** The visibility issue uses landmarks and boundaries that are along the recommended path. It is important for the navigation landmark to be visible. This research uses a Vuforia service (Vuforia, 2020) to analyze planar art along the recommended path to determine which can be used as fiducial markers. Art is ideally suited for landmark-based navigation because art installations' scale, viewpoint, and illumination (Sala et al., 2006) are intentionally salient for user and camera visibility. Using existing art avoids the need to set up and maintain fiducial markers. The system assumes that it is being used inside and the users have normal vision. Therefore, the system does not consider the effects of time or weather on visibility, such as night or fog. The system assumes the earth's curvature does not affect visibility and the space is flat.

One navigation strategy uses two landmarks aligned to indicate the correct course. For example, Captain Vancouver utilized the alignment of Passage Island and Anvil Island to determine the passage course to avoid the shallows off Vancouver, British Columbia. This strategy is not possible everywhere because a second landmark must align with another, so multiple landmarks are used at control points where available.

**Issue 5: Drift.** The fifth issue is reducing the cumulative INS error by correcting the 3D pose using a monocular camera and fiducial markers. The application generates better poses after the user takes a picture of the fiducial marker. Augmented Reality Transformation (ART) Pose uses the planar art pose recorded in the 3D model as a reference measure to transform the user's pose based on the camera's perspective.

### **Technology Objective 2: Map Using a Project Tango Tablet (Google, 2015).**

**Issue 1: Initialization.** The application checks for device motion, poor lighting, available space, clear cameras, and area localization before mapping.

**Issue 2: Processing.** The application processes frames in 33ms providing a display rate of

30 frames per second.

**Issue 3: Gaps.** The user fills areas that appear empty because of scanning issues caused by omissions, infrared light, bright sunlight, light bulbs, or very dark, shiny, and transparent materials.

**Issue 4: Symbology.** The map uses standard International Orienteering Federation (IOF) symbology as a lingua franca to make it universally understandable. Art photographs were taken and stored with their 3D poses as fiducial markers. Existing signs can also make excellent landmarks because they are placed in prominent locations with pertinent information. Additional fiducial markers can be added if the art or signage is not adequate to ART Pose.

### **Application Objective 3 Personalized Guided Tour.**

**Issue 1: Navigation Messages.** The navigational messages help guide the user on their learning path providing hints when required. Suppose the user is familiar with the environment and proves that they know where they should go based on their responses to the spatial cognition learning tasks. In that case, the Unity agent does not need to provide any intermediary guidance messages unless the user goes in the wrong direction. Providing hints only when required should take less time because the user does not waste time with the system, plus it is a good exercise of their spatial skills. Four tasks test the user's Spatial COgnition RatE (SCORE). A SCORE of four correct presents the user with the minimized interface. On the other hand, if a user has zero correct, then they are provided with the most navigational guidance. SCOREs between zero and four get a graduated level of navigational assistance according to their need for help. The navigational messages are generated when a user selects a target destination.

**Issue 2: Safety and Security Messages.** The system directs the users to the nearest exit in an emergency, personalized by their ability and location. For example, if a user is in a wheelchair, then they are guided along wheelchair accessible paths to an exit that is particularly valuable in an

unfamiliar environment. Once the users are outside, the application guides users to a muster point. The leader can send safety and security messages at any time.

***Issue 3: Learning Tasks Messages.*** The group leader configures learning tasks in the Coordinator Schedule Information interface, and group members can update their allocated event tasks. The application displays an ordered task allocation for each group member in the User Information interface, and the details are available in the Schedule Information interface. Learning task messages are available when the server finishes processing the allocations.

***Issue 4: Personalized Space.*** The guided tour is personalized by adapting the instructions for each user. For instance, the navigation messages can convert the distance into the number of steps for that user depending on their stride. The user's stride/step length is approximated by default to 0.413 (Hongu & Wise, 2009) times the user's height. The stride/step length equates to the reach/push of users in wheelchairs. The user's stride/step length is used to determine the size of the square pyramidal frustum indicating the user's field of view (FOV), which is elevated to the user's height. The default stride/step length can be changed for users with a gait variation from orthotics or prosthetics. The default reach can also be overridden if the user's reach is impaired.

**Application Objective 4 Spatial Cognition Learning Tasks.** Only the experimental group users get the spatial cognition learning tasks when they select a destination target for testing and training. The control group only gets spatial cognition learning tasks at the beginning and end of ten navigation cycles for testing purposes.

***Issue 1: Uniform Field of View (UFOV) Task.*** The UFOV task tests if the user understands which direction to go to the next landmark on the learning path. This task presents a radar screen that flashes the direction of the next landmark, and then the user must pick that direction.

**Issue 2: *Mental Rotation Task (MRT)*.** The MRT task is used to test if the user understands the path from their current location to the next destination. This task presents a rotated and skewed wireframe of the path to the next landmark and three other wireframe paths. The user must pick the rotated and skewed wireframe of the path to the next landmark.

**Issue 3: *Method of Loci (MOL) Task*.** This helps prepare or review the learning tasks. It allows the user to make and place notes within the 4D model before, during and after navigation to help remember what to do or what was done.

### **Thesis Organization**

Chapter II reviews relevant literature in spatial cognition to identify important problems in the field. The review will lead to the reasons for this research project focused on spatial cognition. Chapter III discusses the analysis and design of the DMAS based AEGN system. It examines the workflows and use cases for the methods and algorithms to generate the spatial cognition learning tasks. This chapter also provides an architectural framework for storing and representing the spatial relations of objects. Chapter IV explains with running examples of how the front and back end of the AEGN system work together on the dataset for the overall scenario to generate the spatial cognition learning tasks. Chapter V will discuss the AEGN system experiment.

## **Chapter II – Spatial Cognition Learning Tasks**

Chapter II reviews relevant literature in spatial cognition to identify important problems in the field. The review will lead to the reasons for this research project focused on spatial cognition. The first section of chapter II introduces the scenario and problems with spatial cognition. The second section of chapter II discusses important concepts, including the cognitive model, the temporal model, fiducial markers, AR, and spatial cognition learning tasks, to present the rationale for this research project.

### **Scenario and Problems**

"Are we there yet?" is a trivial question most kids ask too often to the annoyance of their parents. Still, it is a nontrivial question depending on who is asking the question and what the situation is when the question is asked. For instance, the question is important if it is a disabled person trying to find the closest emergency exit in a fire, especially if the fire and smoke block the most immediate exit (Klippel et al., 2010). As well, it may be a nontrivial solution to determine a precise interior location and direction to the destination because GNSS, cellular and wireless signals are degraded, jammed or unavailable inside of buildings. A combination of devices needs to be utilized to determine position inside tight interior spaces (Koppers, 2009). This study enhances an ANAV with augmented reality fiducial markers, spatial cognition learning tasks, and a cognitive, spatial, temporal model.

First, this study begins creating a cognitive model by adding an egocentric navigation (ENAV) system (Fleming, 2005) to an ANAV. ANAV is based on external direction references such as the cardinal directions north, south, east, and west. In contrast, ENAV is based on internal direction references such as the relative directions left, right, forward, and backward. The ENAV uses an INS to navigate with dead reckoning or path integration to determine the user pose as they

move. Path integration integrates the motion senses or sensors along paths to determine the direction and distance from a starting point. Darwin (1873) noted the ability for animals to dead reckon, and Murphy (1873) referred to this as an integrating process. This navigation strategy would be referred to later as path integration (Etienne & Jeffery, 2004; Mittelstaedt & Mittelstaedt, 1980) to denote the integration of motion senses to sense the path direction back to the original starting point. The corrected INS positions are compared to GNSS (geo-centric) positions to evaluate their precision and accuracy. AEGN allows the users also to use GNSS.

Animals use visual (optic flow), proprioceptive (body pose) (Lackner, 1988), and vestibular (body motion) senses, while the ENAV uses a camera and INS as comparable sensors. Path integration is subject to drift in both animals and ENAV caused by disorientation and errors in senses and sensors, respectively. The cumulative drift error is calculated to compare the accuracy of the pose of the fiducial markers. The user's pose calculated by path integration is corrected at fiducial markers, and the path integration continues based on the updated pose. The fiducial markers are located at control points where the user may need directions.

Tolman (1948) suggested that animals store spatial information in a cognitive or mental map. This study creates a cognitive model analogous to the biological, mental representation of the spatial data in animals and humans (Maniatakis et al., 2011). For instance, additional discoveries related to spatial cognition include place cells by O'Keefe and Nadel (1978), head direction cells by Ranck (Taube et al., 1990), boundary cells by O'Keefe and Burgess (Barry et al., 2006; O'Keefe & Burgess, 1996), and grid cells by Edvard and May-Britt Moser (Hafting et al., 2005). These brain cell functions combine into an internal posing system that gives us our sense of direction. The cognitive model abstracts the functions as landmarks, directions, and boundaries on a hexagonal grid, respectively.

Second, this study adds time to the cognitive model. The algorithm used to determine the navigation path was based on ease instead of performance. Wu et al. used a basic directional algorithm to calculate the path to the destination. The algorithm cannot navigate well within the tight confines of indoor floors, hallways, and doors. This research investigates using a more efficient algorithm using a 4D model. The cognitive 3D model was combined with a time schedule to create a cognitive 4D model of the environment. For instance, if a user were navigating between displays in a museum, the system would determine where and when to go to minimize time navigating and maximize time learning. At times, the utility of learning might be increased during docent presentations or decreased when it is busy.

Third, this study adds fiducial markers to the cognitive 4D model. The Android based system developed by Wu et al. utilized Quick Response (QR) code markers to determine the user's position. The user could replace the pre-stored and pre-calculated position with the more accurate position of the QR code. This research investigates using a camera to scan fiducial markers to determine the 3D pose. Fiducial markers are more effective at determining pose because their design is simpler than QR codes, increasing the range and precision for calculating the relative poses of the marker and camera (Tsai, 2012). Landmark information is available from the database and 3D model.

Fourth, this study adds augmented reality to the cognitive 4D model using the fiducial markers as targets. Lu et al. (2011) proposed a DMAS with a context-awareness knowledge structure to store learning tasks and an assignment generator to create activities appropriate for a user role, theme, and location. They also suggested improvements by adding storytelling and rewards to help motivate the user. Storytelling is achieved in part by following the layout of a museum display which often is in a logical order such as chronological or geographical. Rewards

are added with additional information in the form of AR. Lu et al. (2014) suggested that in learning games, “Augmented reality may help in enhancing the perceived usefulness of context-awareness” (p. 113). Li et al. (2013) also found that AR motivated users to learn by grabbing their attention like a pop-up book (Horner, 2012).

This research adds AR spatial cognition learning tasks. These tasks are added with the fiducial markers that provide the sub-meter precision and accuracy required to determine the image pose effectively. We only use AR at the fiducial marker orientation points and not while moving because it has been shown to have mixed success in transit when sub-meter posing is difficult to achieve (Duenser et al., 2012; Wen et al., 2013a). AR is used to make map interpretation less difficult by removing the mental map rotation and map symbol correlation to the surrounding real world. However, we use AR to make the initial orientation more challenging by adding spatial cognition learning tasks to improve users’ spatial knowledge (Wen et al., 2014b). Users’ spatial knowledge is tested with spatial cognition learning tasks to assess the users’ sense of direction. Once the user has entered their answer for the direction and distance to the next landmark, the application adapts the navigation tool appropriate for their classified level of spatial knowledge (Wen et al., 2013b). For example, a user demonstrating higher spatial knowledge is only given a direction bearing, while lower-level users are also given a top forward map. If a user needs additional guidance, then they can switch to a lower-level interface. However, a skilled user might navigate from the landmark orientation points without referencing the device, eliminating distractions and delays. We also experimented with implementing the top-down view as a panoramic picture projected onto a radar overview (Duenser et al., 2012) because Mulloni et al. (2012) found it was the fastest way for a user to interpret orientation.

Finally, this study adds spatial cognition learning tasks using an ART Pose to initialize the

tasks and answers. Wu et al. suggested that an ANAV could help users locate and learn materials faster in a museum or zoo. This study investigates the suggestions of Wu et al.. It improves the ANAV performance and functionality based on their suggested issues and spatial cognition learning tasks. For example, recent research on spatial cognition indicates that GNSS decreases users' spatial cognition learning by increasing dependence on the device and decreasing spatial awareness (Field et al., 2011). This study designs an interface that provides the user with a spatial layout rather than just turn-by-turn instructions to help increase the user's spatial awareness.

Another recent spatial knowledge discovery has been that women typically have less spatial cognition than men, but this can be overcome by playing video games (Feng et al., 2007). The UFOV and MRT tasks from the video game used by Feng et al. are incorporated into the AEGN interface for selecting a direction and path, respectively. These spatial cognition learning tasks are required when navigating with AEGN.

An additional feature added to this design is a simple note that can be associated with landmarks to facilitate the MOL mnemonic developed by ancient Greek and Roman scholars (Bower, 1970). This feature helps the users memorize what they have learned by remembering the places they went. This system, modeled on human spatial cognition, should reinforce users' natural spatial cognition and make them more independent at navigating.

### **Rationale**

Stephen Hawking (Hawking et al., 2014) and Elon Musk (2014; as cited in Wastler, 2014) have warned that artificial intelligence (AI) poses a threat to humanity as in the movies *Transcendence* and *Terminator*, respectively. Elon Musk (2014) warned that "There are some scary outcomes, and we should try to make sure the outcomes are good, not bad" (as cited in Wastler, 2014, para. 3). We can attempt to avoid the science fiction rise of the machines by applying the

bioethical principle of doing no harm or nonmaleficence. Computer learning has led to a convergence of AI and human intelligence as it has been used to improve both programs and people. However, people have become less autonomous as they have become increasingly dependent on computers, while AI improvements have made machines more autonomous. When using autonomous machines to maintain human autonomy, it is important for people to know when and how to take over manual controls (Kamphorst, 2012). A good example of this shift in autonomy has been documented in the airline industry's transition from manual to automated flight. Airline pilots are losing the knowledge and skill required to know when and how to take over from the autopilot.

The real threat to people exists when we depend on computer automation for critical processes, and we cannot manually take over when required. For instance, the Federal Aviation Administration (FAA, 2013a) has issued a safety report recommending that pilots practice more manual flight skills rather than relying on flight automation. They have identified increased manual errors (FAA, 2013b). The National Transportation Safety Board issued a safety alert related to two cases when the pilots did not realize they were at the wrong airport until they landed. These cases were dangerous because the runways were shorter than required for a normal landing of their aircraft and the air traffic control messages they were receiving were for traffic at another airport (NTSB, 2014). The pilots had to use emergency braking to stop on the shorter runways. The aircraft might have crashed if the weather was not optimal or if there were some other planes on those runways. Colgan Air Flight 3407 and Korean Air Lines Flight 007 are direr examples of autopilot related incidents that led to fatalities (Konnikova, 2014). These incidents might have been avoidable if the pilots knew when and how to take over from the autopilot.

The problem with people becoming overly dependent on computer guidance has been

found to a wider extent with the general use of LBS with ubiquitous GNSS-enabled devices (Field et al., 2011; Wen et al., 2014a). People can become fixated on the computer map representation and lose consciousness of real landmarks, just as Jean Baudrillard (1981) described hyperreality:

“Today abstraction is no longer that of the map, the double, the mirror, or the concept.

Simulation is no longer that of a territory, a referential being, or a substance. It is the generation by models of a real without origin or reality: a hyperreal. The territory no longer precedes the map, nor does it survive it” (p. 1).

A recent example of this occurred in Chicago when a man followed GNSS automated navigation messages past warning signs, around barricades, and off the end of a ramp being demolished, killing his wife (Healy, 2015). The degradation of pilot and driver knowledge and skills from an increasing dependence on plane and car automation accelerates the reasons to transition to pilotless planes and driverless cars. If human autonomy is abdicated and atrophies below a minimum level, then a guardian is required to take care of them (Casner et al., 2014; Kamphorst, 2012). If pilots and drivers cannot maintain their autonomy, then their role as the passengers’ guardians will be replaced.

It is ironic and dangerous to be misled by hyperrealistic computer navigational messages. People need to figure out when to use their own sense of direction with the computer in the background to help, not hinder. McLuhan wrote, “media is the message” (McLuhan & Fiore, 1967, p. 2), implying that the media messages the message in such a way that the meaning is changed. McLuhan used a tetrad diagram to illustrate how communication technology transforms messages using the gestalt figure ground method from the perspective of the background historical, technological context (Bolton, 2007). For example, the “pro et contra” of the transition from tours using hand-held devices to tours using hand-held devices with AEGN can be found in Table 1.

**Table 1***Tetrad for Hand-Held Tour Devices With AEGN*

“Pro”		“Contra”	
Figure (media/communication technology) hand-held tour devices with AEGN		Background (historical, technological context) hand-held tour devices	
AEGN		Hyperreality	
Enhances		Reverses	
Retrieves		Obsolesces	
Personal tour guide		Spatial cognition	

Table 1 shows how new technology using AEGN enhances hand-held tour devices, but the benefits may be reversed if a hyper realistic user interface misleads the user from reality and spatial cognition becomes obsolete. SLAM was used to help identify 3D mapping discrepancies. The 3D model provides surveyed artificial landmarks, or fiducial markers, for reference control points to correct the user pose. The system messages give the users and leader safety and security messages to give them the confidence and independence to explore a predefined space. The enhanced system retrieves more localized, current knowledge that is a benefit of having personal tour guides that were replaced by hand-held tour devices.

McLuhan & Fiore (1967) also wrote the following:

“All media are extensions of some human faculty— psychic or physical. The wheel ... is an extension of the foot, the book is an extension of the eye...clothing, an extension of the skin...electric circuitry is an extension of the central nervous system. Media, by altering the environment, evoke in us unique ratios of sense perceptions. The extension of any one sense alters the way we think and act— the way we perceive the world. When these ratios change, men change” (pp. 26-41).

Technology extends our senses, putting things virtually in front of our eyes and at the tips of our fingers, extending our proprioception. We sense the pose of our surroundings like we sense the pose of our bodies, making our environment immediately accessible. D.C. Simpson (1962)

described this as Extended Physiological Proprioception (EPP) in his study of patients' ability to perceive the tip of prosthetic limbs compared to this study where the prosthetic is the hand-held device. EPP is used in this research to help the users develop their cognitive map.

O'Keefe and Nadel (1978) physically located the cognitive map place cells in the hippocampus by showing that they fired when a subject was in a particular location. Woollett and Maguire (2011; as cited in Jabr, 2011) showed that the hippocampus grows with increased spatial cognition learning when they studied taxi drivers memorizing the London streets and landmarks. Conversely, they found that the hippocampus size decreased in retired London taxi drivers suggesting that less spatial cognition leads to grey matter loss. As well, injuries and diseases affecting areas of the brain associated with spatial cognition have been shown to increase disorientation. The potential physiological effect of either brain growth or atrophy makes it essential to conduct the study in an ethical manner to ensure nonmaleficence. For example, stereoscopic interfaces should be avoided with epilepsy patients (Koenig, 2012).

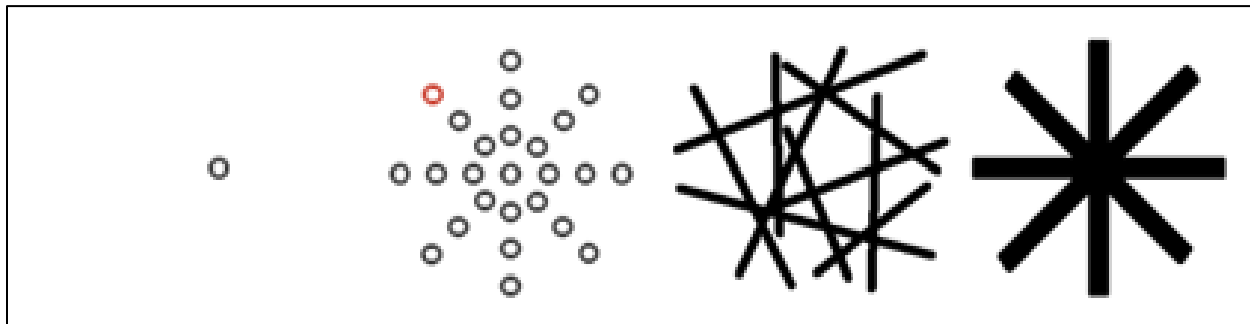
A user extended physiologically by a prosthetic device is a cyborg or cybernetic organism (Clynes & Kline., 1960). Cybernetic originates from the ancient Greek word *kybernētikē*, the art of steering or navigation (Davis, 2009). Ancient Greek navigators, *kybernētai*, were masters of multiple disciplines, including astronomy, meteorology, geography, and geometry. These disciplines' increasing depth and breadth make it difficult for a person to master the summation of all the scientific knowledge today. However, INS, AR and GNSS devices can help users respectively with internal (Ego-centric) (Figure 30) and external (Allo-centric) (Figure 29) directions in a global (Geo-centric) (Figures 31-33) context by encapsulating some of this navigational knowledge. The AEGN abbreviation is pronounced like Aegean, the sea where the *kybernētai* learned their navigation strategies. Likewise, the navigational system combined with

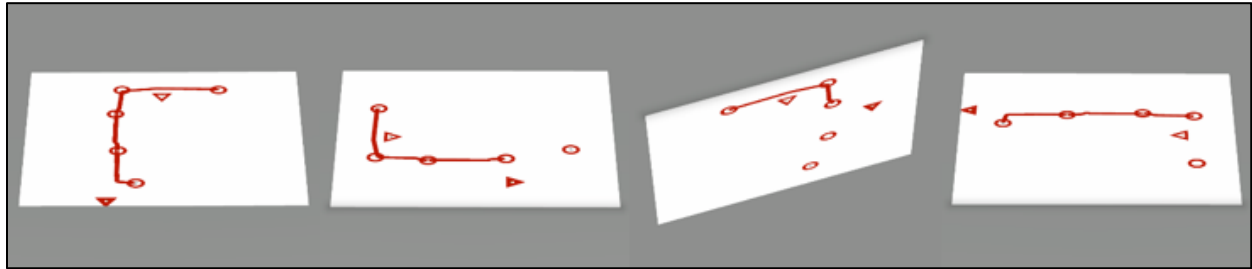
spatial cognition learning tasks also allows the users to learn navigation. For example, one strategy commonly used by kybernētai was coastal navigation, pilotage to maintain natural and artificial landmarks in sight such as islands and lighthouses (Davis, 2009). In a similar fashion, users learn how to use natural and fiducial landmarks to navigate with the visual aid of AEGN.

O’Keefe and Nadel (1978) also suggested that the spatial cognition learning of a cognitive map supports the MOL for location-based learning as well. Cicero attributed the mnemonic MOL to the very unpleasant experience by the Greek poet Simonides of Ceos (Cicero, 55/1942). He was at a banquet and left the building just before it collapsed and crushed all his friends. Simonides had to remember where everyone was sitting because they were unidentifiable. He was able to identify them by placing mental images of his friends in the location of the banquet hall in his mind’s eye. He later realized this method could be used for remembering other information. Other studies have also shown that spatial cognition learning tasks can improve people’s spatial cognition. Feng et al. (2007) proposed a video game to enhance user spatial cognition. Figure 1 and 2 shows the UFOV and MRT, respectively.

## Figure 1

*Uniform Field of View (UFOV) Task: Fixation (699ms), Stimulus (30ms), Mask (600ms) and Response*



**Figure 2***Mental Rotation Task (MRT)*

In the UFOV task the user is shown a sequence of screens and then must pick which direction had the location of the selected landmark. The MRT requires the user to choose one screenshot view that displays the selected path that has been rotated and skewed amongst the four views. Both tasks were effective at increasing the user's spatial cognition. Several other studies have been made investigating how to implement indoor mapping and navigation with hand-held devices, which can be used for learning to improve spatial cognition. The UFOV, MRT and MOL tasks are like Koenig's (2012) Pointing Task, Virtual Navigation Task and Virtual Memory Task for orientation, navigation, and memory purposes, respectively. These tasks are reusable learning objects that can save time setting up the learning environment. Neale et al. (2013) also added a memory task feature called an interest point marker to remember spatial properties.

Tsai (2012) proposed using planar fiducials to determine the pose of a camera. Tsai reviewed several markers, including natural and artificial features but recommended planar fiducials (Figure 3). The large simple graphics optimized them for detection and posing at a greater range than the other options. Fiducials are artificial landmarks that can augment or fill in for real landmarks when they are not available where required. Tsai also suggested using reflective materials to make the fiducial easily detectable in variable light conditions. A cluster or constellation of fiducial markers can be used to increase the accuracy and range for determining a

relative pose even if some markers are obstructed (Figure 4).

**Figure 3**

*Planar Fiducial Marker and Planar Art as a Fiducial Marker*



**Figure 4**

*Planar Fiducial Marker Constellation and Planar Art as a Fiducial Marker Constellation*



Klippel et al. (2010) proposed a standard for You-Are-Here (YAH) maps. Klippel recommended consistency, patterns, alignment, and a current location marker would help orient the user. Klippel reviewed some YAH maps in the same building and found variations and inconsistencies between maps and even within maps where the legend details differed from the

map. Another important feature for users is patterns. For instance, staircases and hallways are typically in the same location and orientation on different floors. These features are important to make familiar to the user instead of the locations of individual cubicles that would add unnecessary clutter. As well, Klippel suggested dynamic YAH maps for further research could provide user and context aware information. For instance, a dynamic map directs people who can walk to the closest staircase and people in wheelchairs to the nearest ramp in this research.

Sjanic (2017) used a monocular camera and an INS to determine navigation on a landmark map. He solved the SLAM problem of mapping and navigating an environment. These calculations can be used to determine the camera's pose relative to a landmark.

Huseth et al. (2011) proposed a Geospatial Location Accountability and Navigation System for Emergency Responders (GLANSER) that is an INS corrected by Radio Frequency (RF) signals for GNSS-denied environments. However, the same challenges for GNSS positioning exist with the RF signal diminishing with increased obstructions. The prototype that they have developed fuses INS, Doppler radar, GNSS, Ultra-Wide Band ranging radio, barometric pressure sensor and a communications radio. Their design assumes no knowledge or preparation of the interior environment before entering, which eliminates setting up fiducial or QR code markers. This system is relatively expensive to be widely accepted.

Anyplace Internet-based Indoor Navigation (IIN) (Zeinalipour, 2016) uses a hybrid Wireless Local Area Network (WLAN) and Inertial Measurement Unit (IMU) navigation. A WLAN heatmap and IMU are used to recognize and track the user's location between points of interest in stacked 2D top-down view drawings. By comparison, this research uses a calibrated and corrected INS for navigation through 4D models to landmarks with first, third and top views. As well, the path generation, navigation and task allocation are automated. The communication

component in this research also helps users coordinate their movement. However, the most significant difference is the addition of spatial cognition learning tasks to this research, which should improve the users' sense of direction. This research could benefit by adding features such as the hybrid WLAN and IMU navigation, Google Maps integration and drawing imports.

This research mitigates the hardware expenses by developing a DMAS that can be deployed on users' own hand-held android devices except for the device used to help identify discrepancies in the 3D map. We use a Project Tango Tablet (Google, 2015) to quickly create the 3D model (Koenig, 2012) in Unity (Unity, 2015). Rapidly creating 3D models helps individualize appropriate contextual environments (Wasinger, 2014). The Tango tablet uses a near infrared scanner and cameras for 3D SLAM. Mantis Vision makes the Tango 3D sensor, MV4D, like the Prime Sense 3D sensor used in Microsoft's Kinect. The 3D model combined with a time schedule creates the 4D environment used to calculate the navigational paths and messages. The users navigate the space with help from their hand-held device when requested, and the device keeps track of their pose using the INS corrected by ART Pose. Vuforia is used to process the planar art as fiducial markers and the associated AR (Vuforia, 2020). If the users know where they are going, it should be faster for them to continue directly to their destination without checking the hand-held device (Rosen, 2014). If they are going off track, then the system suggests a correction. The user can check a simple and consistent map of their location that is minimalized and abstracted to reflect the important features of cognitive maps such as landmarks, directions, boundaries, and hexagonal grid coordinates. These features are enough to support spatial cognition without creating a distracting hyperreality. The map uses the IOF map symbols (IOF, 2018) that have been designed to be unambiguous to users of all languages. Spatial cognition learning tasks help develop the user spatial cognition by using the MOL to enhance the user recollection of the learning task. This

navigation system is novel in that it intends to improve both artificial and human intelligence.

Augmented Intelligence (IA) enhances the user's cognition and makes them more autonomous from the system (IEEE, 2021).

### **Chapter III –Spatial Cognition Learning Tasks Analysis and Design**

Chapter III discusses the analysis and design of the DMAS based AEGN system. It examines the workflows and steps in the methods and algorithms. This chapter provides the structure of storing and representing the spatial relations of objects and the steps in the algorithms and methodology to generate the spatial cognition learning tasks. The first and second sections of chapter III describe the AEGN system algorithms and the system architecture framework, respectively. The third section of chapter III presents the DMAS workflows and use cases.

#### **Algorithms**

This section describes the AEGN system algorithms. The path generation subsection discusses the first step using Unity's A\* algorithm to generate all the possible paths between landmarks. The task allocation subsection discusses the next step in which the path lengths are used by a Contract Net Protocol (CNP) algorithm to determine the task allocations between group users based on the users' first preferences and using brute force to allocate the remaining tasks based on the minimum total group travel time. The calibration subsection discusses how to calibrate the INS. The spatial cognition subsection discusses how the users can select their assigned tasks that generate orientation details to test and train the user how to navigate to the task by adapting the navigation interface based on their SCORE. Finally, the navigation subsection discusses how the interface tracks the user's pose based on the sensor readings processed through an INS Complementary filter algorithm.

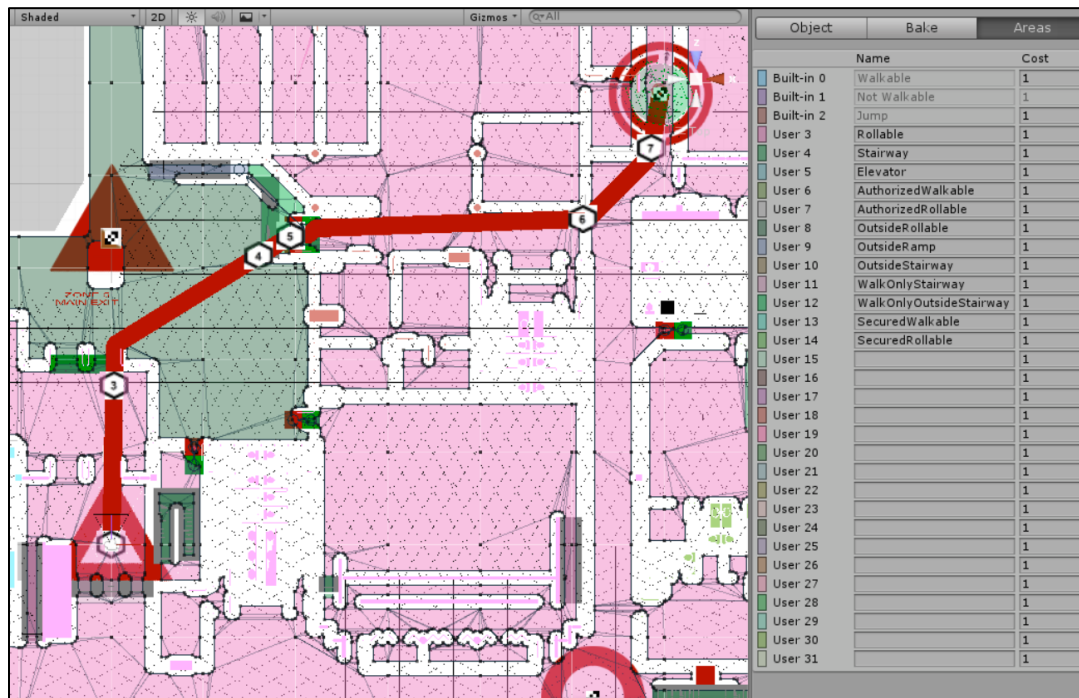
#### **Path Generation**

Unity's A\* algorithm is run to determine all the possible paths between landmarks as edges and nodes for two alternate weighted asymmetric/directed Hamiltonian graphs. The alternative paths are based on walking and rolling motion modes to allow the solution to adapt to user abilities.

It is asymmetric/directed because the paths between any two nodes may not be the same because there are one-way doors. There are also one-way fastest routes outside where user routes pass pedestrian street crossing buttons before crosswalks but not after. It is Hamiltonian because all nodes are connected to all other nodes, allowing a path to visit all nodes once. It is weighted because sections of the path may be slower. For instance, washrooms, kitchens, or laboratories may have wet floors, and caution warrants walking slower. A weighted graph assigns a number, or weight, representing a cost to an edge between vertices. Weights are implemented in Unity as area costs which make the distance of an edge, or path, across an area appear equal to the distance multiplied by the cost. For example, an area cost of 2.0 will make the distance between vertices, or nodes, appear to be two times as long. The Navigator applies higher costs to outdoor areas where users may be slowed down by the weather, such as rain or snow, when the users could need extra time to get an umbrella or jacket. The effect of the weighted area cost can be seen in the AU Main Campus building model between the main lobby (zone 5) and the library (zone 3). The shortest path is to go outside across a courtyard to the exterior library doors (Figure 5). However, applying a weighted area cost to the area outside makes the fastest path inside the main hallway (Figure 6). In Unity, the costs must be 1.0 or greater, so the effect always makes the path appear longer (Unity, 2016a). Tuning the effect of costs is by trial and error to generate the desired behavior of the path generation.

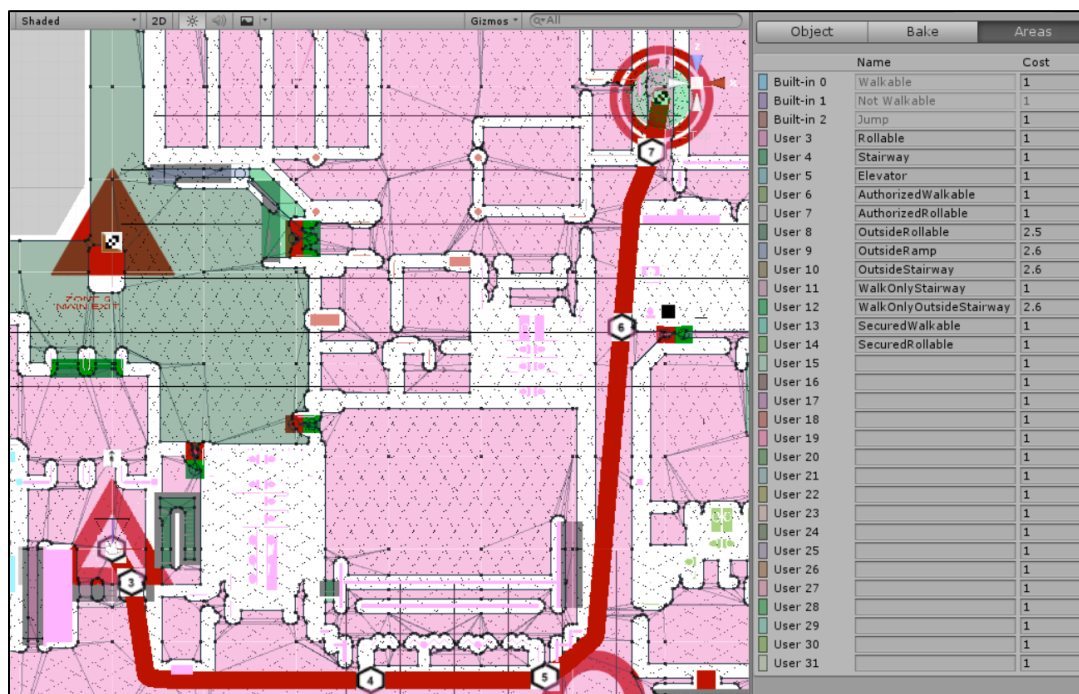
**Figure 5**

*Unweighted Areas Outside (Green) Make the Shortest Path Outside*



**Figure 6**

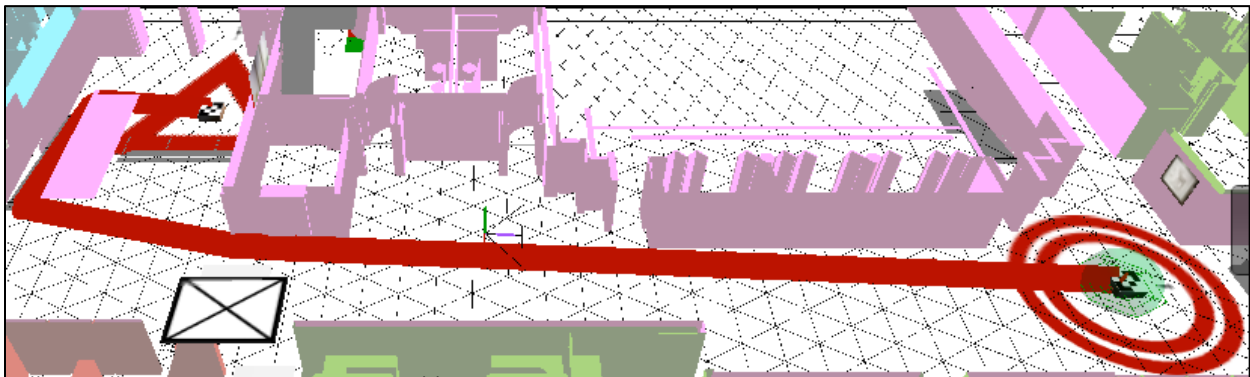
*Weighted Areas Outside (Green) Make the Fastest Path Inside (Pink)*



The navigation mesh (NavMesh) was generated in Unity using Filmbox (fbx) files exported from AutoDesk 3DS Max dwg files. The Project Tango tablet was used to scan areas with discrepancies to help build a very accurate rollable floor space (i.e.,  $\pm 2\text{cm}$ ) to ensure the paths were passable for a wheelchair (Figure 7). The distances are real world measurements, not geodesic or Euclidean. The Tango NavMesh was generated by building voxels, 3D pixels, from the 3D scanned point cloud. This generation is done in three steps: rasterization into voxels, extraction of walkable surfaces, and generation of a NavMesh. Comparing the Tango NavMesh to the AutoDesk NavMesh helps identify discrepancies in vertical and horizontal distances. The A\* algorithm then used the NavMesh to generate paths by selecting the closest visible corners of a corridor made from the NavMesh voxels.

**Figure 7**

*Red Navmesh Wheelchair Path From Start Triangle to Finish Double Circle*



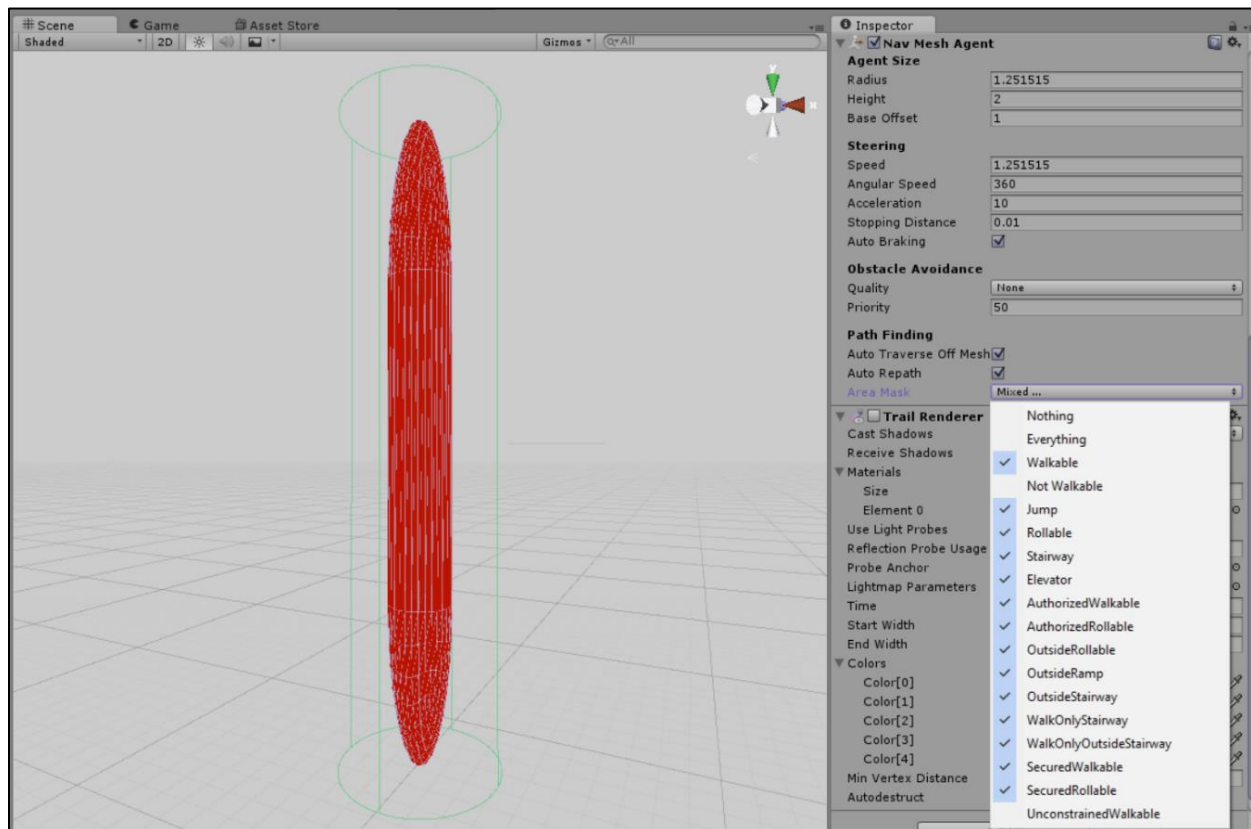
Understanding how the NavMesh is generated is important to configure the process efficiently. Decreasing the size of the voxels increases the density and optimization of the NavMesh and path, respectively but at the expense of processing time and storage. For example, halving the voxel size requires four times more processing time and storage.

The recommended default size is 12 voxels per Unity agent radius (Figure 8) to produce an accurate and efficient NavMesh (Unity, 2016b). The processing time and storage expense can

also be limited by scanning only the corridors that require discrepancy checks. This efficiency can be achieved by planning the scan by evaluating the site and target landmarks to minimize the time to scan, generate and store the navigation paths. Care should also be used to avoid scanning small obstacles such as wire fencing because the scanner will not detect it, and it will be processed as navigable. Also, hot, dark and glass surfaces will not scan with the IR sensor, but these problems might be mitigated by scanning at night with artificial light and closed curtains. The scanned area combined with building drawings should produce an accurate NavMesh and an acceptable approximation of the optimal paths using the A\* algorithm method. The navigation path algorithm provides the fastest path, object shaping, obstacle avoidance, and visibility issues.

**Figure 8**

*Unity NavMesh Agent Properties*



This algorithm improves the performance of the basic directional algorithm used by Wu et

al. to calculate the path to the destination. The algorithm determines navigation paths well within the tight confines of indoor floors, hallways, and doors. This algorithm efficiently navigates the 4D model. For instance, costs can be applied in advance to areas to make Unity agents take longer to pass through to model situations where the fastest path may not be the shortest path.

---

**ALGORITHM 1:** Path Generation (Figures 17, 23)

---

**Input:** NavMeshes [walkable, rollable], Tasks

**Output:** All task paths' distances and sprites for walking and rolling

```

1: For i = 1 to 6 do
2:   For j = i+1 to 6 do
3:     Foreach NavMesh[k]
4:       Move Unity agent using Unity's A* algorithm from Task[i] to Task[j] on NavMesh[k]
5:       Calculate distance
6:       Create sprite
7:     EndForeach
8:   EndFor
9: EndFor
10: return All distance table records and jpg sprites

```

---

For example, if there are six tasks, then there will be 60 paths between all the tasks, 30 for walking and 30 for rolling if the user is in a wheelchair. There will be five Unity agents starting at each of the six tasks, and each Unity agent will be assigned the destination goal of one of the five other tasks that has not yet been generated. The paths are bidirectional, so there are two paths between all destinations to account for the possibility of one-way routes.

The Unity agents first generate the paths on the walkable NavMesh and then generate the paths on the rollable NavMesh for wheelchairs to avoid rolling obstacles like stairs. The distances and sprites for each path will be saved to be used later for the task allocation and the MRT spatial cognition task, respectively. The path generation is used again in the Navigator for guidance between control points and ad hoc paths to anywhere on the NavMesh with the addition of a personalized height constraint to avoid any bottlenecks from height obstructions.

### **Task Allocation**

While the path generation is optimal, the task allocation is not. However, it is ethical by balancing equity and efficiency. The task allocation problem is a generalization of the Travelling Salesman Problem (TSP) based on path generation's weighted asymmetric Hamiltonian graphs. This problem is an extended multiple TSP (mTSP) (Khamis et al., 2015). It is multiple because it can calculate the allocation for one, two or three users. It extends the problem constraints by considering the physical capability of the user's motion mode. This feature makes the task allocation more equitable because it can adapt to the user's physical abilities (i.e., wheelchair users). In addition, the problem solution uses a combination of the CNP (Smith, 1980) and brute force to balance equity and efficiency. CNP is useful to allocate tasks in a DMAS. This solution uses CNP to allocate one task preference to each back-end proxy Jason agent (Figure K3, Table K1, Table K2), representing each respective user, before using the brute force process to reduce the size of the problem. Allowing each user to select a preference provides an equitable reward for the user's group participation by allowing them to pick the task or location of interest to them. If the users select familiar tasks or locations, then that can add to the overall efficiency in the task or travel time (Bernasco, 2019; Kyritsis, 2018). The remainder of the tasks is optimally allocated using brute force to determine the most efficient overall group travel time by iterating through every ordered combination of the permutations to find the minimum time. While the CNP is complicated with many messages between multiple front-end mobile AMUSE (Agent-based Multi-User Social Environment) agents (Figure K1, Figure K2, Table K1, Table K2), the burden of handling these messages is mitigated by using automated back-end proxy Jason agents to represent the users in negotiations once they selected their preference.

The complicated CNP is also offset by the logical simplicity of the brute force solution to

loop through the remaining permutations. The brute force method guarantees the overall group minimum travel time for the remaining tasks. CNP adds an additional benefit of allowing groups to self-organize. However, self-interested users may manipulate the CNP to the detriment of the overall efficiency. AEGN combines the CNP with the Clarke Tax Mechanism (CTM) (Ephrati & Rosenschein, 1991) to ensure tasks are allocated fairly. The CTM measures the social welfare maximization of the group task allocation. CTM is calculated using the CNP travel times. Large differences in CTM between users can indicate that tasks are not being evenly shared and may require redistribution to ensure users cooperate rather than compete to complete tasks.

This brute force calculation minimizes group travel time using the distance table of the tasks and the user speeds. All users start from the same location, but the velocity based on their stride and cadence calculated from their height will be different. Allocation will send the taller users to the tasks farther away. Users in wheelchairs will have the distance converted to a push equivalent to a stride. As well, users in wheelchairs may have different distances because the path to avoid obstacles such as stairs may be longer. This algorithm can find the task allocation in a reasonable time by constraining the number of users and tasks to three in a group.

This scenario, where three users pick task preferences and allocate the remaining three tasks, can be practically used for the task allocation in a short morning scrum or stand-up meeting of less than 15min to assign daily work. A group is any three Coordinator users. A leader is any Coordinator user who invites other Coordinator users to join a group. There are six tasks that are defined by the leader in the Coordinator schedule information, one task for each location. When the users accept the invitation, the schedule information is copied to their own schedule (Figure 9). They can modify the tasks allocated to them, which are also updated for the other users.

**Figure 9**

*The Schedule Information Includes the Hours Open (Attend), the Worst Times (Avoid) and the Best Time (Assign)*

**Schedule Information**

**31 AU Main Campus**

1 University Dr, Athabasca, AB

facility and safety orientation

26-10-2019 9:00 for 08:00 hrs

**5** Event: Task 5 ▼

Main Reception, Executive and Library

assign Explore evacuation routes to the

from 10:00 for 01:00 hrs

attend Open

from 9:00 for 08:00 hrs

avoid Lunch

from 12:00 for 01:00 hrs

**ALGORITHM 2:** Task Allocation (Figures 18, 24)

Input: Paths, Groups, Tasks, Users, Leader

Output: Task Allocations

```

1:  start Leader agent to coordinate the CNP
2:  Do while Leader Jason agent not shutdown
3:    Foreach Groups[i]
4:      Foreach Tasks[j]
5:        Foreach Users[k]
6:          Leader Calls For Proposals (CFP) from Users[k] for all Groups[i] Tasks[j]
7:        EndForeach
8:        Wait 3600s for proposals
9:        Leader receives ready messages from Users
10:       Leader accepts or rejects proposals
11:       Leader offers task allocations which minimize travel time for Users with CTM
12:     EndForeach
13:   EndForeach
14: EndDo
15: Foreach Groups[i]
16:   Foreach Users[j]
17:     start Users[j] Jason agent
18:     Do while Users[j] Jason agent not shutdown
19:       Users[j] sets motion mode as walking or rolling and length of stride or push
20:       Users[j] sends motion mode and length as offer for task
21:       Users[j] accepts or rejects proposal
22:     EndDo
23:   EndForeach
24: EndForeach
25: return Task Allocations

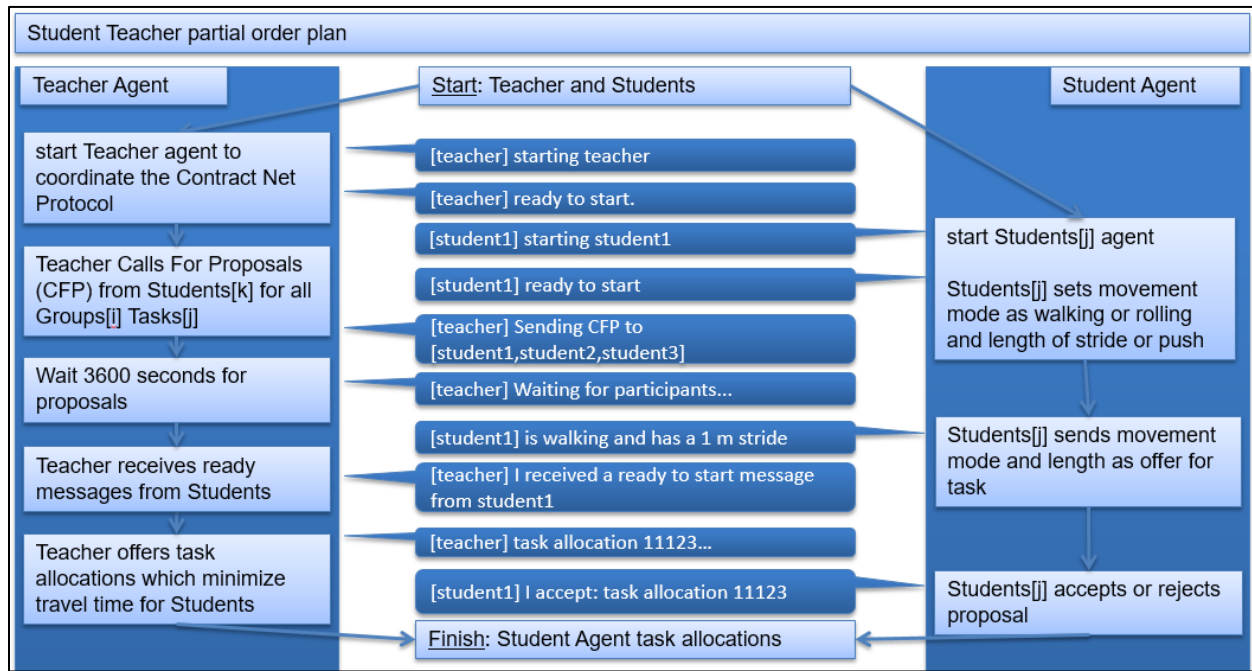
```

For example, the algorithm works well within a few minutes to process the allocation of five tasks to a group of three users. First, the group leader, Jason agent, starts and announces that it is ready to begin. The two other user Jason agents also start and announce that they are ready as well. The leader makes a Call for Proposals by announcing the tasks and waits for proposals. Since the users are starting from the same location, the objective of minimizing travel time is differentiated by the user's speed which is assumed to be calculated based on whether the user is walking or rolling and their length of stride or push. The user proposal is an announcement of their mode and length of movement. For instance, user 1 proposes to walk with 1m strides. The leader acknowledges receiving the user proposals, calculates, and announces the task allocations. The

task allocation cycles through all the permutations of paths for the users to complete the tasks and selects the one that minimizes travel time while ensuring that each user has at least one task assigned. The user task allocation is denoted by a numeric string of the user numbers associated with the tasks. For example, the string 11123 indicates that user 1 has been allocated tasks one to three and tasks four and five, which have been allocated to users 2 and 3, respectively. The process is completed when the users accept the task allocation. The partial order plan illustrates the order of communications to complete the contract negotiations (Figure 10).

**Figure 10**

*Jason Agent CNP Communications Example Where the Leader, a Teacher, Invites Student Users to Join a Group for Task Allocation*



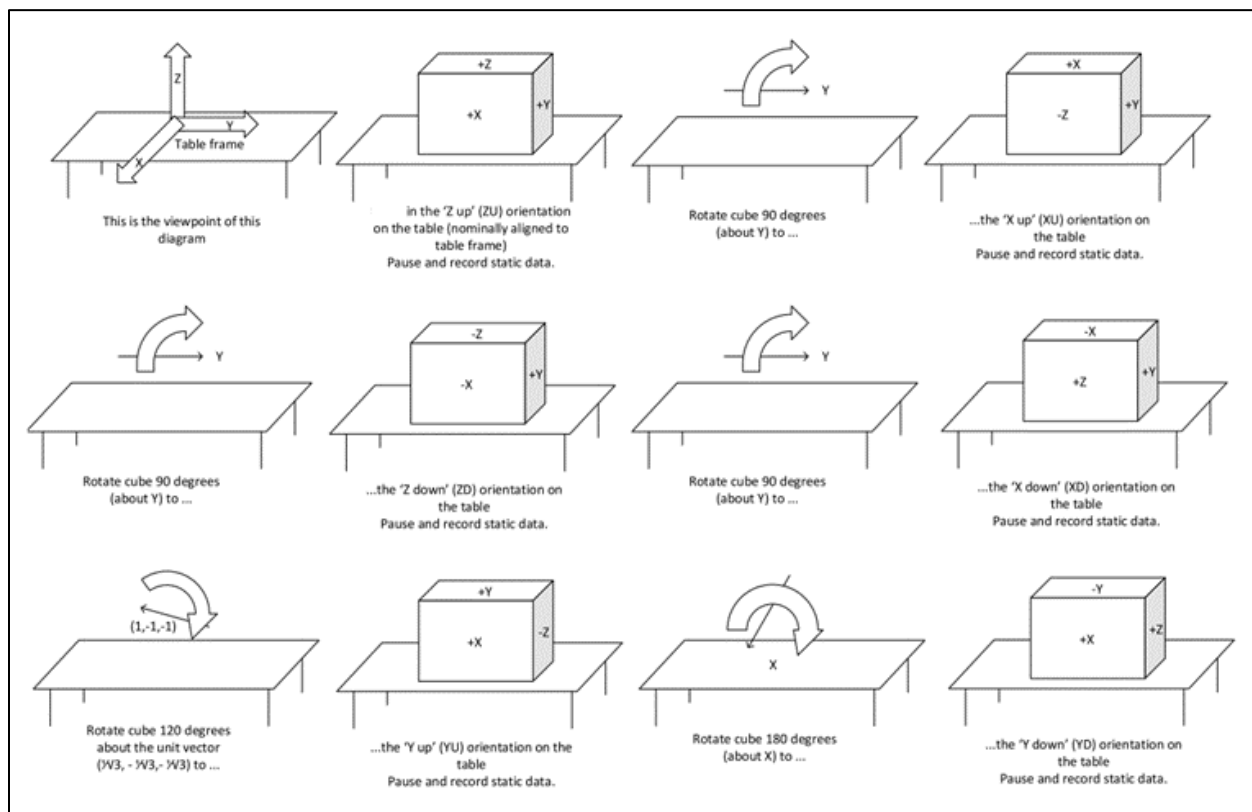
### Calibration

With calibration, the drift is minimized on all axes for both the gyro and accelerometer. The calibration should be performed whenever the device has experienced a physical or temperature shock. For example, if the device has been dropped, then the sensors may have been

knocked out of alignment, or if the device's temperature has changed, then the IMU sensitivity will also change. The device should be allowed to warm up in the environment being navigated before calibrating. If the device was calibrated at room temperature and then the user walked into a standing freezer or outside during the winter, then it would be necessary to recalibrate to maintain accuracy and precision. It is a simple calibration that does not require a special rig (Figure 11).

**Figure 11**

*User-Conducted Calibration Maneuver (Martin, 2016, P. 92)*

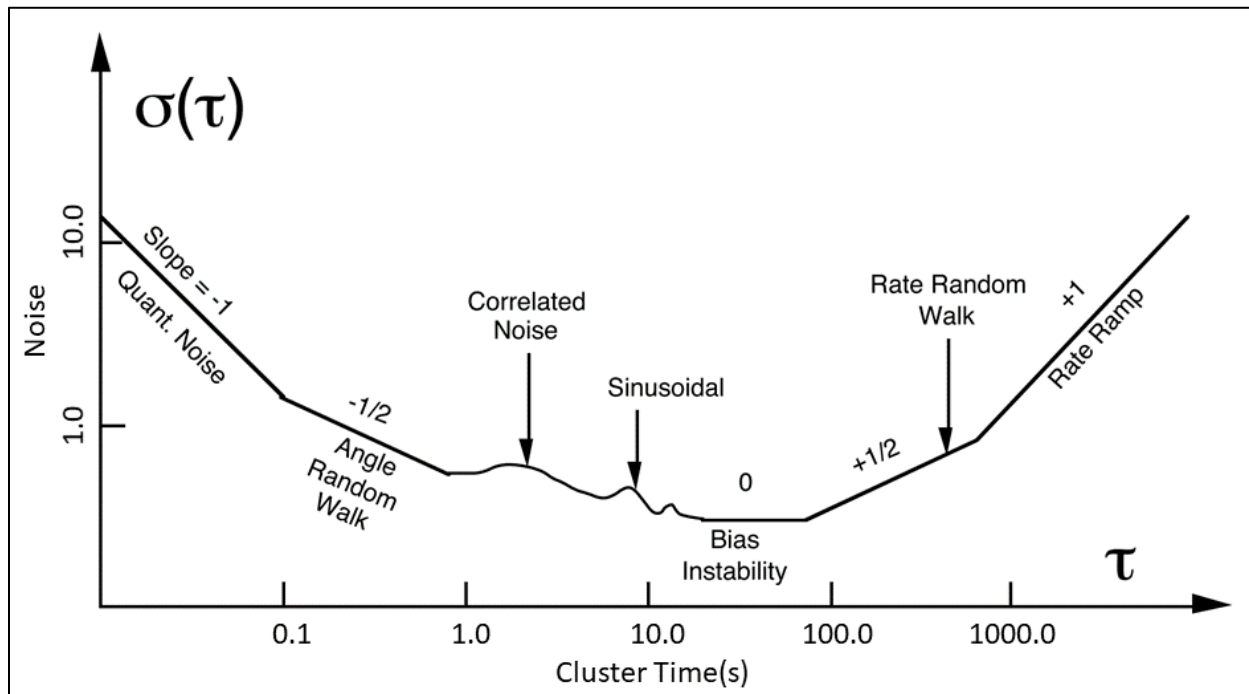


The calibration will calculate zero and first order errors. These include the bias for both the gyro and accelerometer and the accelerometer scale-factor and cross-coupling errors, and the gyro g-dependent errors. A windowsill or door jamb are usually level enough to help brace the device to hold it steady while the calibration is run. The calibrated gyro and accelerometer readings are used in the navigation algorithm. The calibration requires the user to hold the device in both

directions of the three axes for a short cluster time determined by an Allan deviation analysis (Figure 12) to minimize sensor bias noise. Hou (2004) showed that the gyro bias instability (BI) and Angle Random Walk (ARW), as well as the accelerometer BI and Velocity Random Walk (VRW) could be determined using Allan deviation analysis. Calibrating with Martin's Technique One reduces the effect of noise on bias by averaging the sum of static sensor readings in all directions over the longest BI cluster time for both the accelerometer and gyro.

**Figure 12**

*Gyro Allan Deviation Analysis (IEEE, 1998, p. 71)*



The complementary filter time interval ( $dt$ ) is the sample rate, 0.04s or 25Hz, of the gyro, and it is used to determine the change in the angle before the low frequency noise is removed by the high pass filter. The vertical stabilization allows the accelerometer reading to be adjusted to remove the acceleration from gravity before the high frequency noise is removed with the low pass filter. The complementary results of the high and low pass filters are then added to create an updated attitude, which is multiplied by the estimated speed to determine the change in position.

The new position and attitude are then used to update the pose as represented by the frustum on the user interface. The drift is also limited by constraining, or map matching, the vertical position to the navigable surface plus user height. If the position drifts more than 3m from the reference point along the fastest path, then the horizontal position can also be map matched to the reference point.

---

**ALGORITHM 3:** Calibration (Figures 19, 25)

---

**Input:** Gyro ( $\omega$ ) Cluster Time, Accelerometer ( $f$ ) Cluster Time

**Output:**  $\omega$  log file,  $f$  log file,  $f$  bias,  $f$  scale-factor & cross-coupling errors,  $\omega$  bias,  $\omega$  g-dependent errors

```

1: While Calibrate scene open
2:   If  $f$  is running
3:     current axis = sensor axis pointing up (sensor z axis opposite of unity z axis)
4:     If warming up and time < 1800s
5:       Write time,  $f$  readings to  $f$  log file
6:     EndIf
7:     If calibrating and time <  $f$  Cluster Time
8:       mean  $f$  current axis = Sum ( $f$  readings)/time
9:     ElseIf
10:       $f$  bias =  $f$  readings / 6
11:       $f$  scale-factor & cross-coupling x axis errors = mean  $f$  x axis / 2
12:       $f$  scale-factor & cross-coupling y axis errors = mean  $f$  y axis / 2
13:       $f$  scale-factor & cross-coupling z axis errors = mean  $f$  z axis / 2
14:    EndIf
15:  EndIf
16:  If  $\omega$  is running
17:    If warming up and time < 1800s
18:      Write time,  $\omega$  readings to  $\omega$  log file
19:    EndIf
20:    If calibrating and time <  $\omega$  Cluster Time
21:      mean  $\omega$  current axis = Sum ( $\omega$  readings)/time
22:    ElseIf
23:       $\omega$  bias = mean  $\omega$  / 6
24:       $\omega$  g-dependent x axis errors = mean  $\omega$  x axis / 2
25:       $\omega$  g-dependent y axis errors = mean  $\omega$  y axis / 2
26:       $\omega$  g-dependent z axis errors = mean  $\omega$  z axis / 2
27:    EndIf
28:  EndIf
29: EndWhile
30: return  $f$  bias,  $f$  scale-factor & cross-coupling errors,  $\omega$  bias,  $\omega$  g-dependent errors

```

---

### **Spatial Cognition**

The spatial cognition algorithm trains and tests the user's orientation to determine the appropriate navigation interface for their cognitive ability. The basic premise for training the user's spatial cognition is learning by rote, or repetition. The spatial cognition tasks draw the user's attention towards the start, current, and following landmarks. Then the SCORE from the spatial cognition tasks is used to adjust the navigational help provided so the user must continue to think about where they are going. The pose determined while scanning the AR marker at a control point initializes the spatial cognition algorithm. The next four steps apply only to the control group for testing and to the experiment group for both training and testing. These four steps are spatial cognition learning tasks to help orient and navigate to the next landmark. The first step for all users is a path integration test relative to the start landmark. This task involves pointing the camera in the direction of the marker and entering the number of steps of the direct distance to the marker regardless of any obstacles. In the second step the system shows the user the path and description. This path display includes first person, third person, and top views (Figures 29-33). The path uses standard IOF map symbology with red triangles, circles, and double circles representing the start, control and finish markers respectively connected by a standard IOF red line path (Figure 7). The cognitive 3D model presented in the map displays a simplified view of the hexagonal grid, boundaries, and landmarks to reduce laborious drafting details and unnecessary viewing distractions. For instance, it avoids showing an obstruction by displaying the navigable path around the object. Detail images are provided for important landmarks such as the planar art fiducial markers. Photos of existing signs can also be added to provide important directional information. The following two steps are the UFOV and MRT learning tasks. Finally, the experiment and control group are given a path integration test relative to the next landmark. The test results average

is used to calculate the user's SCORE.

---

**ALGORITHM 4:** Spatial Cognition (Figures 20, 26-27)

---

**Input:** landmarks, lastMark, agent, pathSprites

**Output:** SCORE

```

1:   buffer = 1.5m
2:   mark = selectMark(landmarks)
3:   i=1
4:   lastControl = agent.position, agent.dir
5:   While path (mapView (top, north, none, cardinal, curved), UnityAgent)
6:     If distance (agent, control) < buffer
7:       PCD[i,1] = symbol
8:       PCD[i,2] = symbol text
9:       PCD[i,3] = relativeDir (dir (lastControl.position, agent.position), lastControl.dir)
10:      PCD[i,4] = cardinalDir (dir (lastControl.position, agent.position), north)
11:      PCD[i,5] = relativeDirSymbolText(PCD[i,3])
12:      PCD[i,6] = cardinalDirSymbolText(PCD[i,4])
13:    EndIf
14:    i=i+1
15:    lastControl = UnityAgent.position, UnityAgent.dir
16:  EndWhile
17:  mark.CardinalDir = cardinalDir (dir (agent.position, mark.position), north)
18:  MRTask [1] = screenSprite(path(mapView(third), UnityAgent))
19:  MRTask [2] = resizeSkewSprite (MRTask [1])
20:  MRTask [3] = pathSprites (agent.position, lastMark)
21:  MRTask [4] = pathSprites (lastMark, mark)
22:  distance = setDistance ()
23:  if selectUnits () = steps
24:    distance = distance*agent.movementIncrement
25:  EndIf
26:  score [1] = distanceScore (distance, startMark.Distance)
27:  startMark.RelativeDir = relativeDir (dir (agent.position, startMark.position), cameraDir)
28:  score [2] = dirScore(startMarkRelativeDir)
29:  Print agent.position
30:  For i = 1 to i length
31:    print PCD[i,1] PCD[i,6] PCD[i,2]
32:  EndFor
33:  print UFOVTask(mark.CardinalDir)
34:  MRTPath = printRandom4DigitPermute(MRTask)
35:  UFOVCardinalDir = selectUFOVCardinalDir ()
36:  UFOVRelativeDir = relativeDir (UFOVCardinalDir, mark.CardinalDir)
37:  score [3] = dirScore (UFOVRelativeDir)
38:  score [4] = 0
39:  If selectMRTPath () = MRTPath
40:    score [4] = 4
41:  EndIf

```

```

42: distance = setDistance ()
43: if selectUnits () = steps
44:     distance = distance*agent.movementIncrement
45: EndIf
46: score [5] = distanceScore (distance, mark.Distance)
47: mark.RelativeDir = relativeDir (dir (agent.position, mark.position), cameraDir)
48: score [6] =dirScore(mark.RelativeDir)
49: SCORE = score.sum()/score.count()
50: Switch SCORE
51:     Default 0: mapView (top, forward, distance/dir radar, relative, curved)
52:     Case 1: mapView (top, forward, dir radar, relative, curved)
53:     Case 2: mapView (top, north, compass, cardinal, curved)
54:     Case 3: mapView (top, north, compass, none, curved)
55:     Case 4: mapView (top, north, compass, none, straight)
56: EndSwitch
57: path (mapView, agent)
58: return SCORE

```

---

Algorithm 4 shows the process for determining the SCORE and the appropriate map view.

The path and PCD are created as the Unity agent moves past control points in the NavMesh. The path is generated by Unity's A\* algorithm. The path starts with the current Unity agent position and direction (dir) and moves to the landmark (mark) selected from a list of allocated landmarks (Lines #5-16). The description includes symbols and text in relative and cardinal directions. For example, suppose the Unity agent is heading west towards a control point in the northwest. In that case, the cardinal direction symbol points to the upper left but the relative direction symbol points to the upper right. Line #17 determines the cardinal direction of the UFOV to test the user (Figure 1). Lines #18-21 create the sprites for the MRT buttons. Sprites are bitmap icons for buttons. All these sprites are rotated and skewed where one of the sprites is generated from a screenshot of the path generation, and the other three are different paths (Figure 2).

Lines #22-28 and #42-48 are the start and finish path integration tests, respectively. These test the user's sense of direction relative to the start and next landmarks based on their perception of the respective landmark's distance and direction. The user can choose meters or their own step

length as a unit of measure. For instance, if the user selects steps (e.g., 0.5m), then the distance (e.g., ten steps) is converted to meters before scoring the user's distance estimate (e.g., 5m). The distance is scored between zero and four points inversely proportional to the difference between the user's estimate and the actual distance. If the user estimates 5m and the distance is 5m, then they would score four, but if they estimated 10m or more, then they would score zero. On the other hand, an estimate of 2.5m or 7.5m would score two. Similarly, the direction is also scored depending on how close the user estimates relative to the actual direction by pointing the camera in the direction of the landmark. If the user estimates north and the direction is north, then they would score four, but if they estimate south, then they would score zero. An estimate of east or west would score two.

The control point description, UFOV and MRT are then printed out on the device screen in Line #29-34. The order of the MRT buttons is printed randomly. Then the user selects the direction in the UFOV and the path in the MRT. The UFOV is scored similarly to the path integration direction (Line #37). The MRT is scored four points for a correct selection and zero for any other choice (Lines #38-41).

Finally, the scores are averaged to determine the user's SCORE (Line #49). Then the SCORE is used to determine the map view appropriate for the user's level (Lines #50-56). The user should be motivated to score high and then be challenged by a map view which minimizes distraction by providing only the basic information the user requires. For example, a user SCORE of four would display (Line #57) a north up view with a straight path which would require them to rotate the map to their environment and find the appropriate curved path around obstacles themselves.

### **Navigation**

The navigation interface will display navigational information based on the SCORE from the spatial cognition tests. A direction bearing, top forward or top-down view as a panoramic picture projected onto a radar overview is also provided based on the SCORE from highest to lowest, respectively. All views also include IOF control path descriptions for users unfamiliar with the IOF symbology (Table 3) (IOF, 2018). This interface then displays the user's pose based on the sensor readings processed through a Complementary filter. While the Extended Kalman Filter (EKF) is the defacto standard method for sensor fusion, the complementary filter requires fewer processing resources to produce similar results, making it more suitable for mobile applications (Mahony et al., 2008). Fused gyro and accelerometer sensor readings are recorded as the INS poses. These poses, along with the GNSS positions and AR poses, will also be recorded for comparison when the mobile device camera recognizes the control markers. For example, the navigation starts with an initial pose from an AR marker. This AR marker could be at a control point used for spatial cognition learning tasks or only a pose correction point along the route. The position and angle are initialized from this pose. An AR correction should be performed whenever the distance to the ground truth reference Unity agent exceeds 3m.

The movement is constrained to the mode of walking or rolling. Movement is assumed to be forward, so only the accelerometer values for the forward axis are used. The horizontal direction equals the attitude calculated by the complementary filter, not sideways or backwards. Any movement side to side or up and down, such as moving hands, is assumed to cancel over time. Average speed is assumed to be between 0.83m/s and 1.67m/s and is estimated based on the user's height calculation of stride or push length per second. For example, assuming an average height of 1.65m, 200/60 steps/s, and an average stride of 0.413 times height would estimate a speed of

1.38m/s. Starting and stopping activity is recognized by speed changes forward and backward, respectively exceeding a speed change threshold of approximately 0.25m/s. People can start or stop nearly instantaneously when walking or rolling, and they usually walk or roll with the same speed. We therefore set the speed to the average speed when starting or 0 when stopping. Speed changes below the threshold adjust the speed incrementally. The pose is updated every 0.04s unless the user pauses by turning off the clutch. The pause is useful to prevent any drift from the speed calculations when the user is stationary, but the user orientation is still tracked and can drift. Drift caused by bias and noise is also minimized by using clustered and calibrated accelerometer and gyro readings. The navigation time cluster is the same for both the gyro and accelerometer because they are fused. The navigation time cluster is also shorter than the bias noise time clusters to keep the calculations responsive to movements, but at 1.0s, it can still minimize quantization, ARW and VRW noise.

---

**ALGORITHM 5:** Navigation (Figures 21, 28-34)

---

**Input:** movement mode and length, initialization pose, time cluster, constant and interval

**Output:** pose

- 1: Estimated speed = stride or push  $\frac{\text{length}}{s}$
  - 2: Position = position(initialization pose)
  - 3: Angle = angle(initialization pose)
  - 4: Time constant =  $\tau = 0.7s$
  - 5: Time interval =  $dt = 0.04s$
  - 6: Filter coefficient =  $a = \frac{\tau}{\tau + dt}$
  - 7: **Do while** navigation Unity agent not shutdown or paused
  - 8:   Angle = (angle + clustered calibrated gyro \* dt)
  - 9:   High pass filter =  $a * \text{angle}$
  - 10:   Low pass filter =  $(1 - a) * \text{clustered calibrated accelerometer-gravity}$
  - 11:   Attitude complementary filter = high pass filter + low pass filter
  - 12:   Position = position + attitude \* estimated speed
  - 13:   Pose = position + attitude
  - 14: **EndDo**
  - 15: **return** pose
-

## Architecture

This section introduces the system architecture framework. DMAS is ideally suited for this system designed to optimize allocations. The mental state of agent-oriented programming (AOP) extends the biologically analogous concept of spatial cognition. As well, agent communication enables negotiation and collaboration between multiple users. These AOP attributes are implemented in the Beliefs-Desires-Intentions (BDI) architecture of the AgentSpeak AOP language. This project utilizes Jason, an AgentSpeak interpreter, for the CNP task allocation because Jason has a plugin for the Eclipse integrated development environment to aid development and debugging. This project distributes Jason over the network using JAVA Agent Development framework (JADE) (Figure K1, Figure K4, Tables K1, Tables K2) instead of Simple Agent Communication Infrastructure (SACI) because it provides additional tools to monitor the agents.

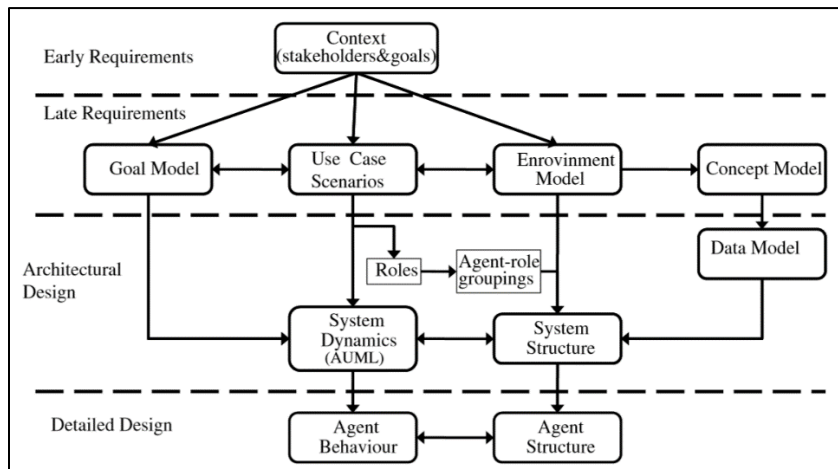
While JADE provides a comprehensive formal methodology to guide the analysis and design of DMAS, it lacks the tools to support the process. JADE is used for the distribution, communication, discovery, and behavior of agents. WADE (Workflows and Agents Development Environment) (Figure K1, Tables K1, Tables K2) extends JADE with a workflow engine and an Eclipse plugin (WOLF). WADE is used for the development and administration of fault tolerant workflows. The WADE Workflow Status Manager Agent (WSMA) persists the workflow state in the H2 database using Hibernate by deploying the WADE Persistence Add-On. WADE services run on Apache Tomcat to provide a web service to allocate tasks to the users dynamically. The AMUSE variation of WADE is implemented for user management of registration, authentication, communication, and coordination. Communication between activities on an android device is passed with the intents. All server development with JADE, Jason, WADE and AMUSE is on Eclipse using Java Development Kit (JDK) 1.7. The Communicator and Coordinator android

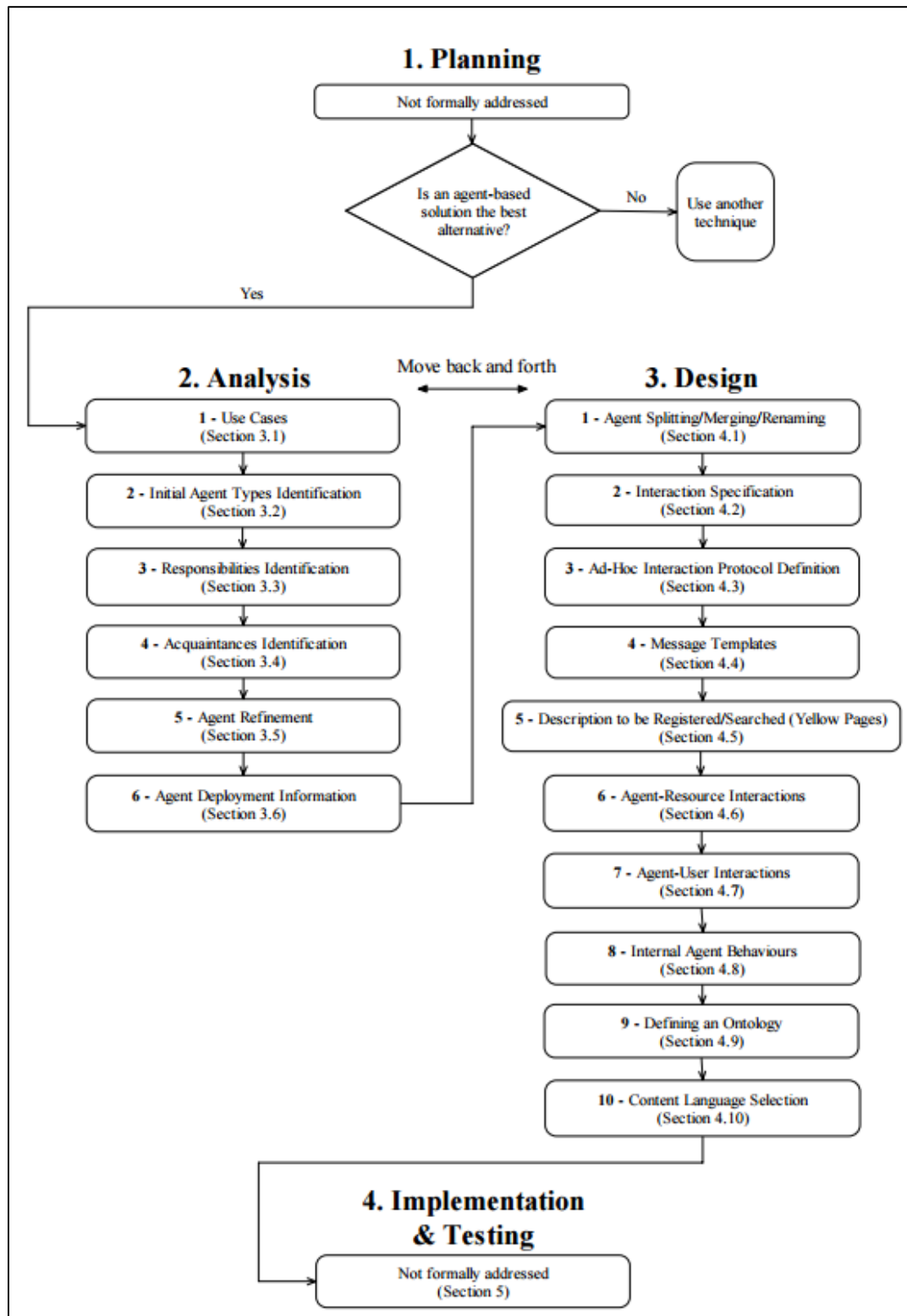
applications are developed in Android Studio 4.1.2 with Android Software Development Kit (SDK) 10 and Android Native Development Kit (NDK) 21.1.6352462. The Navigator android application is configured in Unity, scripted with C#, and then built as a Google Android project to be imported into Android Studio to generate an Android App Bundle (AAB). The Communicator and Coordinator are also generated as AAB before being released on Google Play.

The JADE methodology also recommends selective use of the sections as required. There are several agent-oriented software engineering (AOSE) methodologies with tools available with varying strengths and weaknesses. Dam and Winikoff (2013) (Figure 13) have suggested a Unified AOSE methodology (UAM). This project uses a subset of the JADE methodology (Figure 14) sections augmented with WOLF, another AOSE tool, to achieve a complete set of AOSE design tools. The added benefit of using WOLF for the workflows is that it provides a natural transition to migrate from the Jason and JADE environment to the next version using WADE code that can be generated by WOLF workflows. The WOLF Interaction Description Framework implements the Model View Controller (MVC) pattern using information, visualizer, and constraint elements, respectively, in WOLF workflow code (Bergenti & Caire, 2013).

**Figure 13**

*Unified Agent-Oriented Software Engineering Methodology (Dam & Winikoff, 2013, p. 687)*



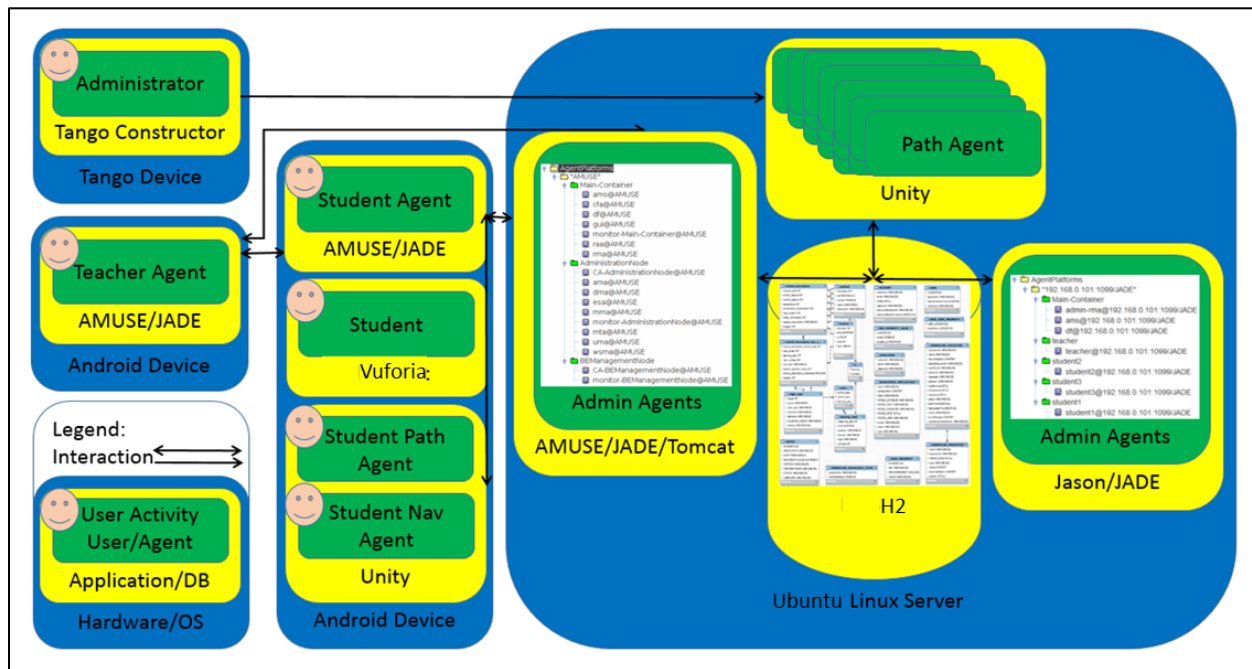
**Figure 14***Jade Methodology (Nikraz et al., 2006, p. 5)*

This project selects sections of the Jade methodology based on the UAM. It implements those details using the Prometheus Design Tool (PDT) because it has an Eclipse plugin that works well with Jason agent design paradigms (Kaim & Lenar, 2008; Padgham & Winikoff, 2003). While this methodology and tool do not generate Jason code, it improves the quality of the design and reduces the time required to make it.

To further improve the quality of the architectural design (Figure 15) and reduce the time, the project utilizes the Project Tango Development kit and Vuforia to develop the mobile client in Unity. The graphics are also abstracted to minimize the demands of collection and presentation. For example, augmented reality requires less time and effort to build than virtual reality (VR), a significant impediment to implementing VR. This method has the additional benefit of presenting a less distracting interface to the user. These hardware and software components significantly decrease the time to produce high quality and quantity graphical input and output.

**Figure 15**

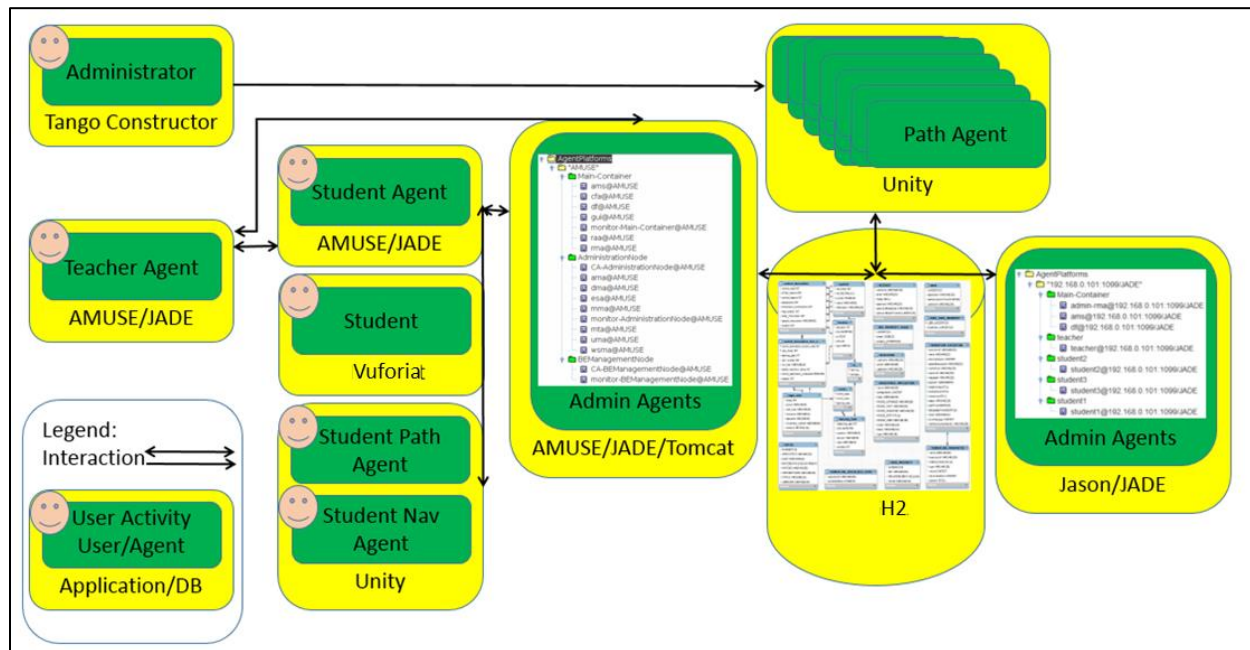
*AEGN DMAS Architecture*



There are administrative and user groups of agents running on the Linux server and android devices, respectively (Appendix K). The Ubuntu Linux server was built on Oracle VM VirtualBox for the required web, application and H2 database server. User agents do not have direct access to the database for security reasons. The AMUSE user manager agent acts as a go between to ensure that data access is secure. The schedule information of the 4D data model is persisted in the database. The 3D spatial information is created with AutoCAD dwg and Wavefront obj files from AutoDesk 3DS Max and Tango Constructor (Google, 2016), respectively. The dwg files contain the latest 2D building drawings extruded into 3D objects, and discrepancies are corrected with the obj data. The spatial model is then exported as an fbx file and imported into Unity. The JADE/WADE/AMUSE activities run on android devices in the Communicator and Coordinator applications. Unity activities run in the Navigator application.

### **Workflow**

This section presents the DMAS workflow and use cases. The workflow (Figure 16) explains how back-end services and the system work together. The use cases show how the inputs are modified to generate the outputs that can become inputs for parallel or follow-up use cases. The use cases are what users see in each step of the designed system. There are five use cases described in the following subsections: path generation, task allocation, calibration, spatial cognition, and navigation. The types of users include students/users, teacher/leader, and administrator. The teacher/leader differs from students/users in that they initially set up the schedule information and invite students/users to join the group. The administrator sets up and loads the system's 3D model. A student/user can only belong to one group at a time.

**Figure 16***AEGN DMAS Workflow*

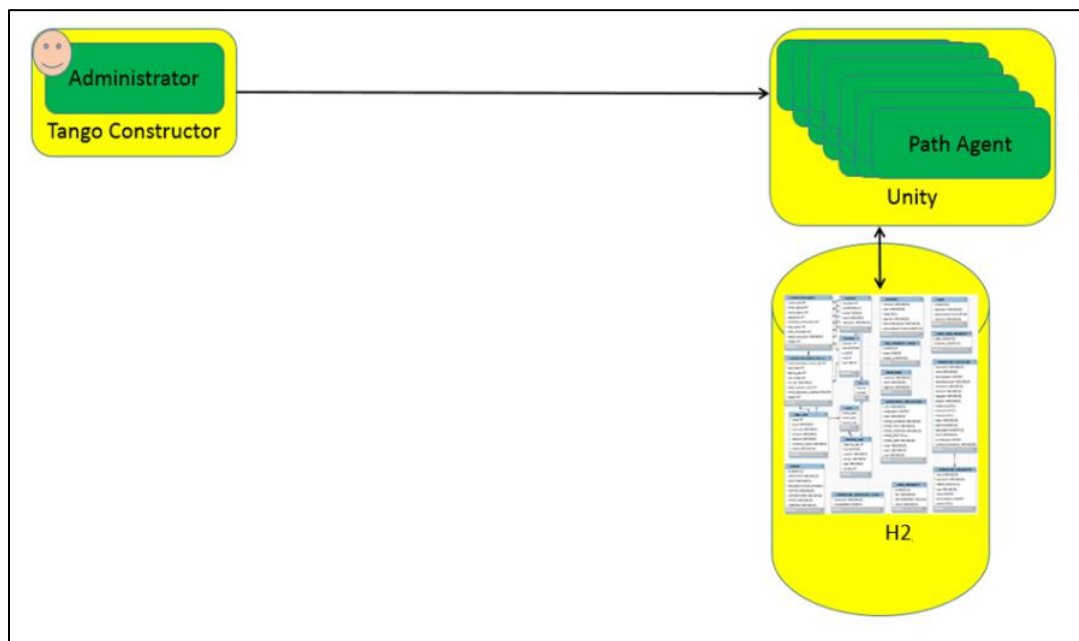
### Path Generation

This subsection introduces path generation (Figure 17) when the administrator creates and configures the indoor map using AutoDesk 3DS Max and the Project Tango Tablet. The workflow to create the spatial cognition data model has been optimized to build the model quickly. The administrator uses the Constructor application on the Project Tango Device to scan the floor in areas with map discrepancies to correct the model. The planar art fiducial markers are photographed, their poses scanned or measured and then added to the data model. Time can be saved by planning the scan to avoid hot, dark, cold, small (i.e., wire) and movable objects to avoid the need for editing or rescanning the area. The quality of the scan can also be improved by holding the device steady during initialization and then moving at a constant speed and processing 30 frames per second while scanning. The scan will be exported from the Project Tango Device as an obj file and loaded into AutoDesk 3DS Max. The corrected 3D model will be exported as an fbx

file and imported into Unity. A NavMesh will be generated, and IOF start, control and finish target symbols will be added. Walking and wheelchair Unity agents will be added for every start and finish target. Running the project will generate all the possible navigation paths. The start, finish, walking path length, and wheelchair path length will be loaded into the H2 database. This data provides us with the cognitive 3D model.

**Figure 17**

## *Path Generation*



## **Task Allocation**

This subsection presents task allocation (Figure 18), which divides the tasks amongst the group members and specifies the order of tasks which is the fastest path for each user.

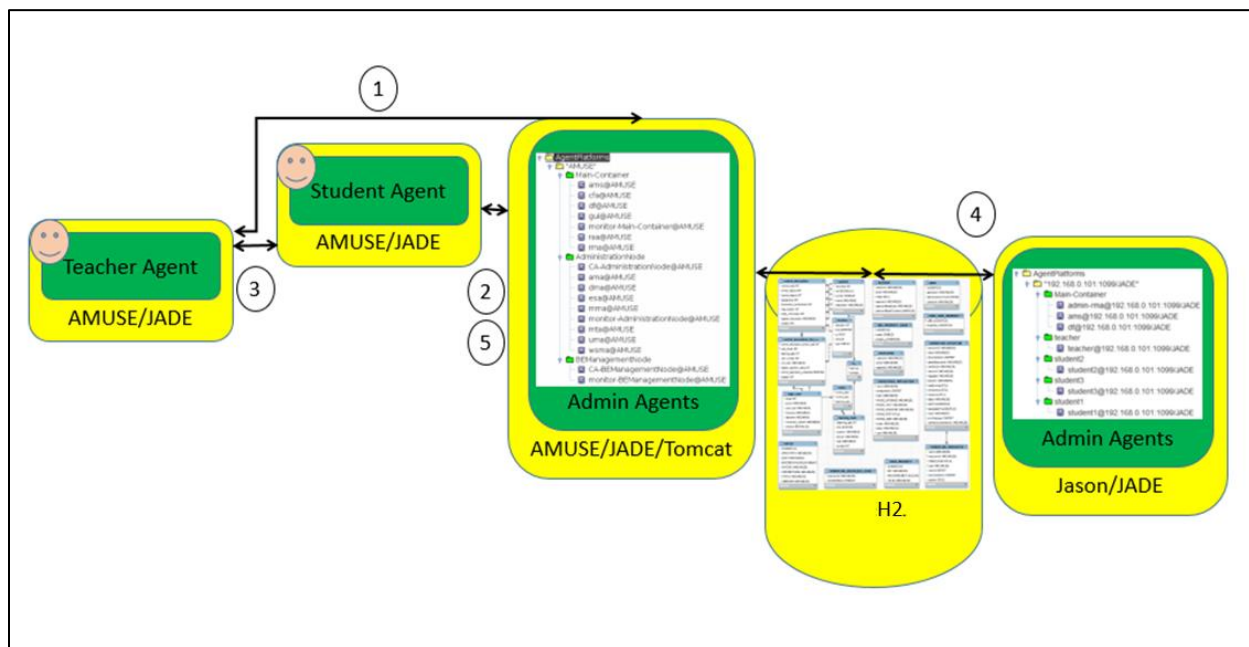
- 1) Sending and receiving personalized communications requires the leader to set up the tasks and invite users after they are logged in.
- 2) At the same time, users register and authenticate to the system.
- 3) The leader and users can then communicate with each other using the

Communicator and Coordinator apps.

- 4) The task allocation also uses the user personalization, group, and task information to allocate the tasks.
- 5) The user task allocations are then available in the Coordinator app's User and Schedule Information activities.

**Figure 18**

*Task Allocation*



The administration, registration, authentication, and communication will be implemented with AMUSE, allowing users to register their username and password. User personalization is configured with a WADE workflow, allowing the user to set whether they will walk or use a wheelchair. Task allocation will be implemented with Jason and is automatically triggered after users select their task preference. The task allocation determines the learning task message based on the personalized method of moving (i.e., walking or rolling). The location of the target task determines the navigational messages that are also personalized by the user's SCORE. The leader

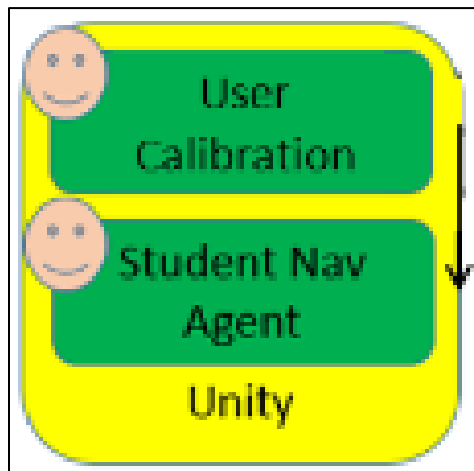
can use the communications to send safety and security messages to the users.

## Calibration

This subsection describes calibrating the accelerometer and gyro in a hand-held android device (Figure 19). The calibration provides us with more accurate and precise accelerometer and gyro readings. A user can conduct the procedure quickly without a specialized rig. The calibration starts warming up the device by running the sensors and logging the sensor readings. The sensors are affected by temperature, so they should be warmed up in the navigation environment. The device should also be recalibrated in the event of a physical shock, such as a drop which could knock the sensors out of alignment. Then the device calibration should be run while holding the device steady in both directions of all three axes. It is helpful to use a square and level surface such as a windowsill or doorjamb to brace the device to keep it still. All six directions must be held steady for about 20s. The calibration can be further optimized by using the Allan Variance analysis on accelerometer and gyro logs to find the minimum sensor bias times to replace the default values and minimize the effect of noise on calibration measurement. For example, the Alcatel 3V gyro has a noise reduction time cluster of 19s, so that calibration would take a minimum of 114s.

**Figure 19**

*User Calibration*

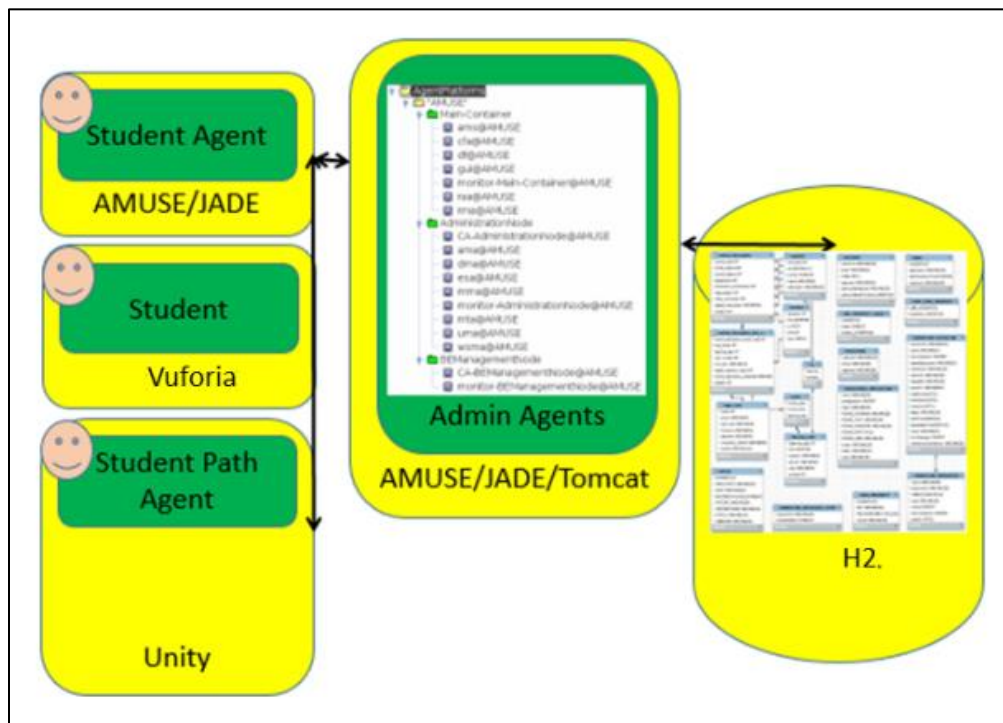


### Spatial Cognition

This subsection, spatial cognition (Figure 20), combines the 4D model, navigation path and personalized messages to generate the spatial cognition learning tasks. These tasks are used to test and train the user's spatial cognition. The learning tasks are initialized by scanning the AR marker at a control point to fix the user's pose. While scanning the AR marker, the user should hold the device steady while starting the navigation activity to ensure the best initialization of the spatial cognition learning tasks.

**Figure 20**

*User Spatial Cognition*



First, the user needs to select the start and the destination targets. The Path Integration Test tests the user's sense of direction and distance. This test requires the user to point the camera towards the start landmark and enter the number of steps/meters to reach it directly, assuming no obstacles exist. The navigation message is described to the user in the symbolic and textual format

of the IOF Path Control Description (PCD). The description includes information regarding the distance and direction of landmarks. The next step is the UFOV (Figure 1), that helps test and train the user regarding the direction to the next landmark. The user's mental map is tested by observing the path and then trying to select a rotated view of the path amongst a selection of similar paths that have been rotated and skewed. The MRT (Figure 2) helps test and train the user's mental map of the path. Finally, a Path Integration Test regarding the next landmark tests the user's sense of direction and distance to the destination landmark. The combined results of these tasks are used to calculate the user's SCORE, which is used to optimize the user's navigation path. These spatial tests and tasks are performed in the Unity Navigator app. The MOL task can be used whenever the user wants to record a note at a specific location.

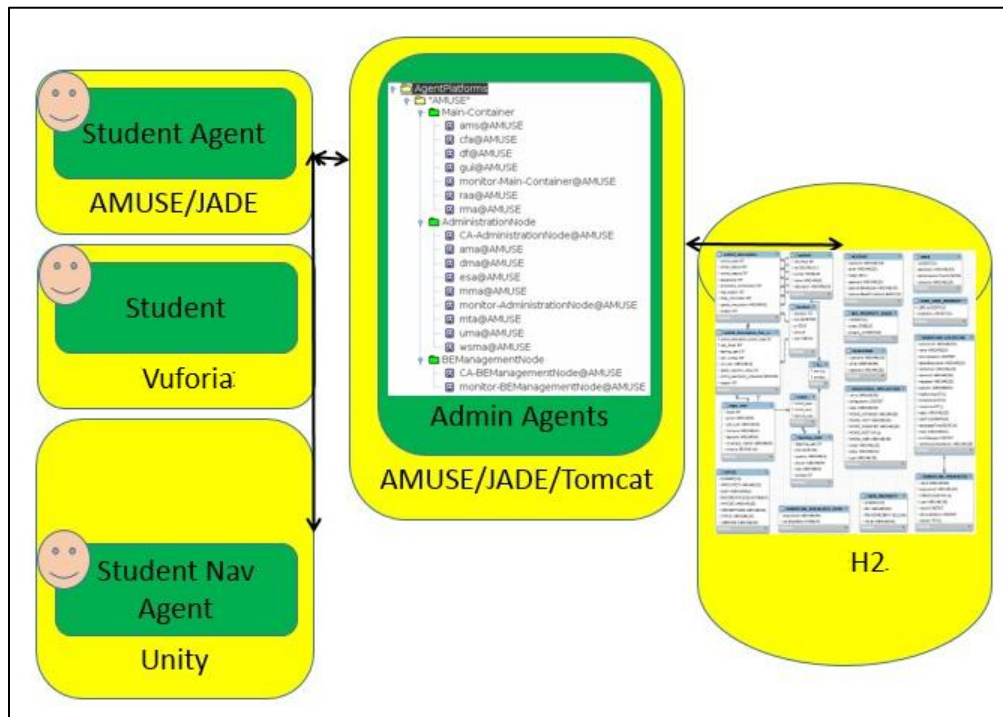
### **Navigation**

This subsection describes the 4D model for indoor navigation using a hand-held android device (Figure 21). This step provides us with an optimal navigation path. The path is optimized for the user's SCORE. The path starts from the AR marker initialization to correct any INS drift. The perspective and symbology used to display the path depend on the following five levels of spatial cognition: basic orienteering, distance pacing, direction bearing, map reading, and pathfinding. These levels equate to the IOF levels of white (very easy), yellow (easy), orange (medium), light green (hard), and dark green (very hard), respectively. The basic orienteering level provides the most intuitive and helpful interface, including a top view, forward facing map with control description text using relative instead of cardinal directions to describe a curved path with a radar overview indicating the relative distance and direction to the destination landmark. This level allows the user to become familiar with the basic mapping features. The distance pacing level provides the same map as the basic orienteering level, except that the radar overview does not

indicate the relative distance to the landmark. This level requires the user to determine their relative distance to the destination. The direction bearing level modifies the distance pacing level by replacing the radar overview with a compass and displaying the top view map in a north-up orientation with cardinal control description text. This level requires the user to determine the direction by rotating the map and becoming familiar with using a compass. The map reading level modifies the direction bearing level by removing the control description text requiring the user to read the map symbols. The pathfinding level modifies the map reading level by replacing the curved path with a straight path requiring the user to determine a path to the destination marker. A level five spatial cognition user practices the orienteering skills regarding distance, direction, map reading and pathfinding. This application helps the user find the fastest, curved, optimal path around obstacles.

**Figure 21**

*User Navigation*



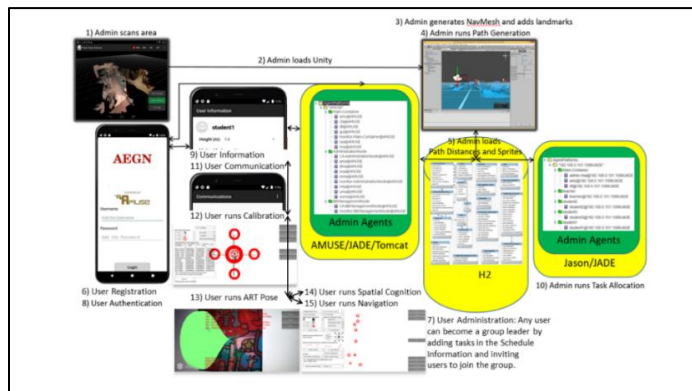
## Chapter IV –Spatial Cognition Learning Task Generation

Chapter IV explains with running examples of how the front and back end of the AEGN system work together on the dataset for the overall scenario to generate the spatial cognition learning tasks. It describes how the inputs are generated and altered and produce the intermediate and output data, respectively. The examples follow these inputs and outputs through successive and parallel workflow steps that the users see at each stage of the process (Figure 22).

Chapter IV is divided into path generation, task allocation, calibration, spatial cognition, and navigation sections. Path generation introduces creating and configuring the indoor map using AutoDesk 3DS Max and the Tango Tablet. Path generation provides us with the cognitive 3D model. Task allocation adds the schedule information to make a cognitive 4D model. It presents the navigation path to generate personalized guided tour messages. These include navigation, safety, security and learning messages. Calibration describes how to minimize accelerometer and gyro errors and noise. Spatial cognition combines the 4D model, navigation path, and personalized messages to generate learning tasks. Navigation describes how the 4D model provides us with an optimal path.

**Figure 22**

*Screenshots of AEGN System*

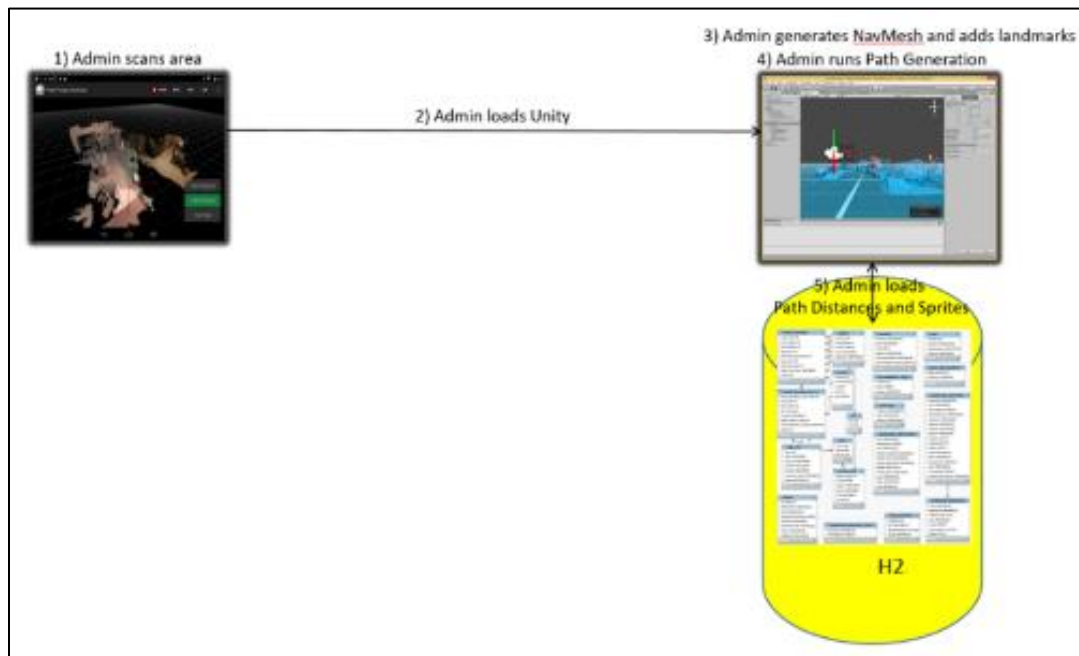


## Path Generation

This section introduces creating and configuring the indoor map using AutoDesk 3DS Max and the Tango Tablet (Figure 23). These steps solve the following issues in the first technology objective: fastest path, object shaping, obstacle avoidance and visibility.

**Figure 23**

*Screenshots of Path Generation*



- 1) Admin scans area: The administrator uses the Constructor application on the Project Tango Device to scan the floor in areas with map discrepancies to correct the model. The planar art fiducial markers are photographed, their poses scanned or measured and then added to the data model. The scan will be exported from the Project Tango Device as an obj file and loaded into AutoDesk 3DS Max with the building dwg to build the model and correct discrepancies.
- 2) Admin loads Unity: The corrected 3D model will be exported as an fbx file and imported into Unity.

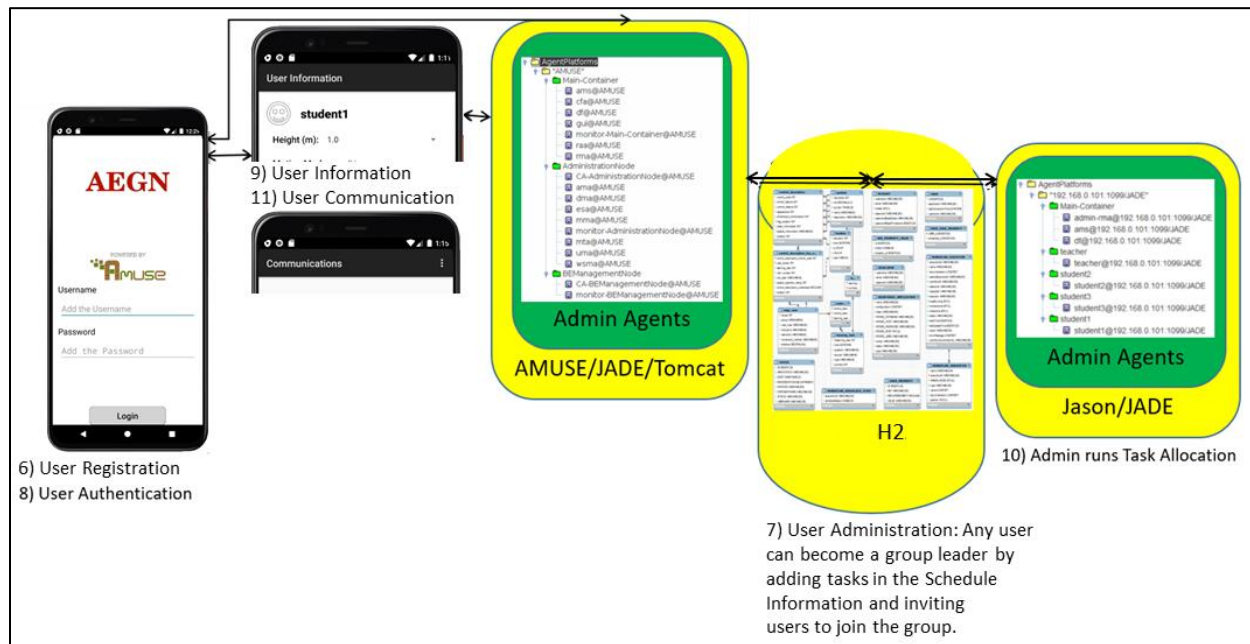
- 3) Admin generates NavMesh and adds landmarks: A NavMesh is generated, and IOF start, control and finish target symbols are added.
- 4) Admin runs path generation: All the possible navigation paths are generated.
- 5) Admin loads path distances and sprites: The start, finish, walking path lengths, and wheelchair path lengths will be loaded into the H2 database.

Path generation provides us with the cognitive 3D model.

### **Task Allocation**

This section presents the process to generate personalized guided tour messages (Figure 24). These include navigation, safety, security and learning messages.

- 6) User Registration: Anyone can become a user by entering a new username and password.
- 7) User Administration: Any user can become a group leader by adding tasks in the schedule information and inviting users to join the group.
- 8) User Authentication: The user enters their username and password.
- 9) User Information: The user enters their personal information, which allows the user to set whether they will walk or use a wheelchair and their height. Preferences are configured with a WADE workflow.
- 10) Admin runs Task Allocation: The admin Jason agent is triggered to automatically initiate the task allocation using the user personalization, group, and task information to allocate the tasks.
- 11) User Communication: The leader and users can then communicate with each other.

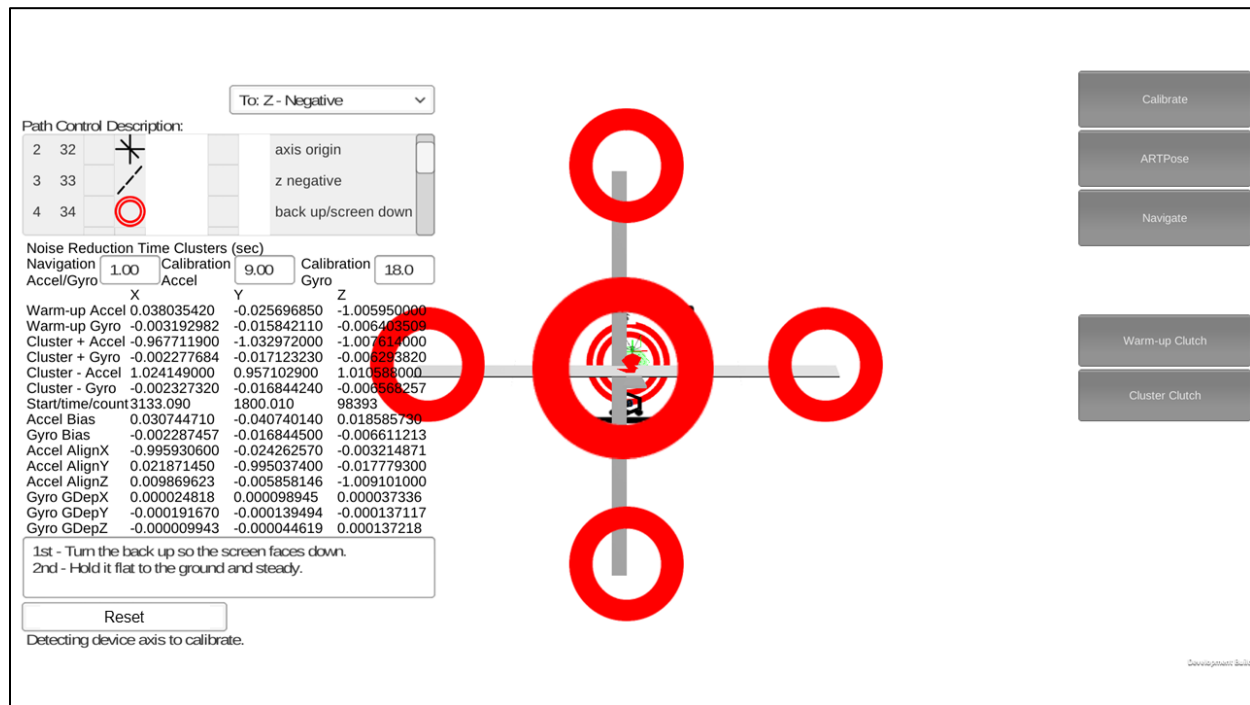
**Figure 24***Screenshots of Task Allocation*

## Calibration

This section describes calibrating the accelerometer and gyro in a hand-held android device (Figure 25). Calibration provides us with more accurate and precise accelerometer and gyro readings. A user can conduct the procedure quickly without a specialized rig to minimize the effect of deterministic and random errors with calibration and cancellation.

12) User runs calibration:

- a. Use the warm-up clutch to stabilize the device temperature and analyze the log file.
- b. Use the noise reduction time clusters to cancel stochastic noise.
- c. Use the cluster clutch to run the deterministic error calibration in both directions of all three axes.

**Figure 25***Screenshot of Calibration*

Calibration starts with a 30min warm-up which logs the gyro and accelerometer readings to a file. Calibration serves two purposes: to stabilize the device temperature in the environment and to analyze the characteristic noise property types and coefficients of the device sensors. Stochastic noise random errors cannot be calibrated but average towards zero over time. How much time is required and how close to zero noise can be reduced are useful specifications to characterize and calibrate sensors. Most manufacturers do not provide detailed specifications on the gyro and accelerometer in mobile devices. The Allan Variance analysis can be used to identify BI, ARW and VRW noise in consumer grade gyros and accelerometers (Hou, 2004). A logarithmic graph of the Allan deviation values square root of variance, makes it easy to determine the time required to reach minimum BI, ARW and VRW noise (Phidgets, 2017). The mean values of the gyro and accelerometer x, y, and z axes are used to identify the cluster times. If there is a

significant change in environment operating temperature, then rerun the warm-up and reanalyze the sensor specifications because they reduce noise in calibration and navigation. If the log is not used for analysis, then the device can be moved during warm-up. However, if the log is going to be used for analysis, then it can be left in any orientation, but it must be kept still. The warm-up is started by clicking on the Warm-up Clutch button. The start, time, count, warm-up x, y, and z readings for both the gyro and accelerometer will update the interface while running.

The stochastic noise cancellation uses the cluster times in calibration and navigation calculations. First, the minimum BI cluster times for both the gyro and the accelerometer are used when running the calibration. They are both used because the calibration is static, so the accelerometer and gyro calibration times can differ. Running the calibration with these time clusters reduces the effect of quantization, BI, ARW and VRW noise on the calculations. Second, navigation will use the same cluster time for the gyro ARW and accelerometer VRW. Navigation utilizes the same cluster time for both because it is a sensor fusion algorithm. The ARW/VRW cluster time is shorter than the BI cluster time, which keeps navigation responsive to sensor readings. Running the navigation with these time clusters reduces the effect of quantization, ARW and VRW noise on the calculations.

Deterministic error calibration minimizes the effect of bias, scale-factor, and cross-coupling accelerometer errors, as well as bias and g-dependent gyro errors. The device should be recalibrated if there has been any temperature change in the environment or in the event of a physical shock such as a drop which could knock the sensors out of alignment. Just warm-up the device if the device has been previously calibrated at the same environment temperature. Once the warm-up has been performed, the user can calibrate the device sensors in a couple of minutes without any special hardware or rig. Calibration is a six-maneuver static tumble test that measures

gravity's effect in both directions of all three axes of the gyro and accelerometer. It is helpful to use a square and level surface such as a windowsill or doorjamb to brace the device to keep it still. All six directions must be held steady for about 20s. The device displays a green bubble that floats to the top of the axis being calibrated. The calibration for the current direction is started by clicking on the Cluster Clutch button. The start, time, count, cluster direction x, y, and z readings for both the gyro and accelerometer will update the interface while it is running. When the cluster direction x, y and z readings are all populated, the accelerometer alignment matrix, gyro g-dependent matrix, and accelerometer and gyro bias vectors will be calculated. The Reset button zeroes all calibration fields. Once the device has been calibrated at the operating temperature, the inertial navigation will drift much less.

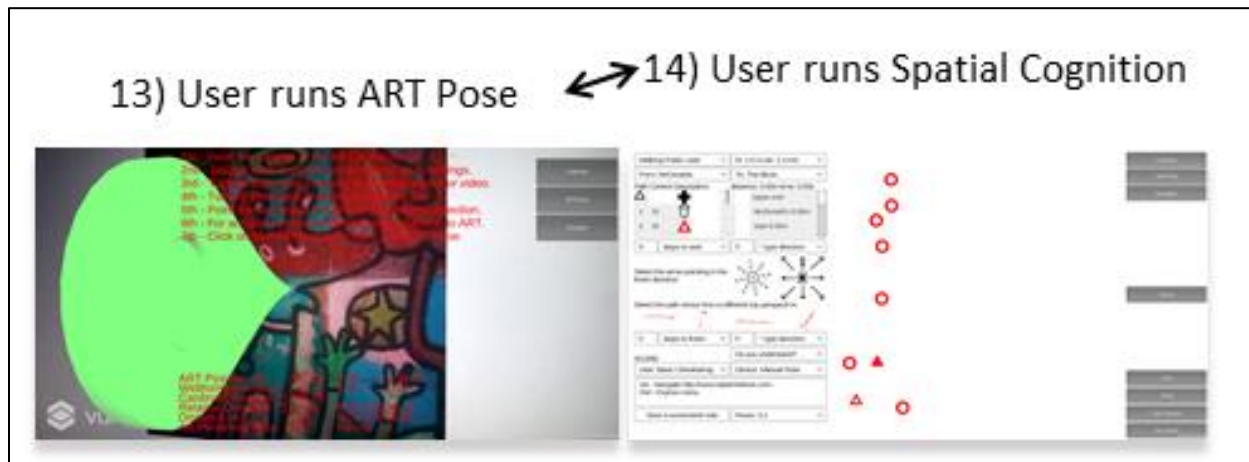
### **Spatial Cognition**

This section combines the cognitive 4D model, navigation path, and personalized messages to generate spatial cognition learning tasks. These tasks are used to test and train the user's spatial cognition (Figure 26).

- 13) User runs ART Pose: The learning tasks are initialized by scanning an AR marker at a control point to set the starting pose in the cognitive 4D model.
  - a. Point the camera at the ART in the model.
  - b. Double click anywhere on the screen to go to settings.
  - c. Turn on the Extended Tracking to view the model or video.
  - d. Turn off the Extended Tracking to view the pin.
  - e. Point the camera at the ART and in the desired direction.
  - f. For accuracy, keep the pin centered and perpendicular to the ART.
  - g. Click on Navigate to save your ART Pose and start the Navigate scene.

**Figure 26**

*Screenshots of ART Pose and Spatial Cognition*



14) User runs Spatial Cognition (Figure 27).

- a. The starting location is set from the ART Pose.
- b. Then the user needs to select the destination target to generate the navigation path and the spatial cognition learning tasks to test and train the user.
- c. The navigation message is described to the user in the symbolic and textual format of the IOF PCD.
- d. The Path Integration Task to the Start (PITS) requires the user to point the camera towards the starting landmark and enter the number of steps/meters to reach it directly, assuming no obstacles exist.
- e. The next step is the UFOV Task which requires the user to select the direction to the destination landmark.
- f. The MRT requires the user to select a path view amongst a selection of similar rotated and skewed paths.
- g. Finally, the Path Integration Task again, but this time to the Finish (PITF).

PITF requires the user to point the camera towards the next landmark, the destination, and enter the number of steps/meters to reach it directly, assuming no obstacles exist.

The combined results of these tasks are used to calculate the user's SCORE, which is used to optimize the user's navigation path. The navigation path and messages are personalized when the user's SCORE is recalculated to optimize the user's view. Unlike these tasks, which are tied to landmark locations, the MOL task can be used whenever the user wants to record a note at a specific location.

**Figure 27**

*Screenshot of User Spatial Cognition Learning Tasks*

1) From ART Pose

2) To

3) PCD

4) PITS

5) UFOV

6) MRT

7) PITF

Walking Public User ht: 1.5 m vel: 1.3 m/s

From: McDonalds To: The Block

Path Control Description: distance: 0.00m time: 0.00s

1 31 2 32

Upper end McDonald's 0.00m start 0.00m

0 steps to start 0 ° type direction

Select the arrow pointing in the finish direction

Select the path shown from a different top perspective:

0 steps to finish 0 ° type direction

SCORE: Do you understand?

User: Basic Orienteering Device: Manual Pose

1st - Navigate <http://www.eatdrinkblock.com>  
2nd - Explore menu.

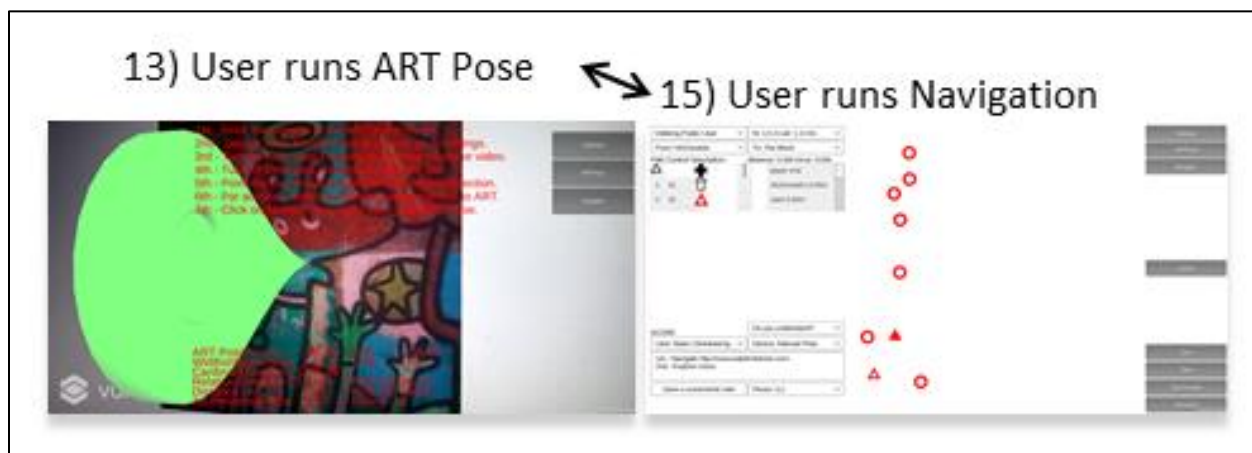
Save a screenshot note Floors: 0,1

## Navigation

Navigation is further improved by optimizing the user's navigation path with the user's SCORE (Figure 28). The user pose is corrected by scanning an AR marker following the instructions described in step 13.

**Figure 28**

*Screenshots of ART Pose and Navigation*



15) User runs Navigation: the view is adapted by the user's SCORE, but the view properties may be changed if desired.

- a. Change the SCORE to use the desired preset view properties (Table 2, 3).
- b. Select a perspective: First person, Third person, Top Forward or Top Down.
- c. 5When the clutch is activated, the Unity agent will trace the user's route.

The properties of the views related to the SCORE are illustrated in these examples. The perspective and symbology used to display the path depend on the following five SCORE levels.




## Table 2

### SCORE View Properties

Level	Direction text (Table 3)	View	Orientation	Path	Overview (Figure 34)
Basic orienteering	Relative	Top (Figure 31, 32, 33)	Forward (Figure 31)	Curved (Figure 31, 32)	Distance and direction radar
Distance pacing					Direction radar
Direction bearing	Cardinal		North (Figure 32, 33)	Straight (Figure 33)	Compass
Map reading					
Pathfinding					
	None				

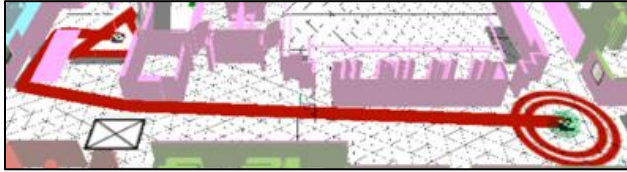
### Table 3

*IOF Symbols With Relative and Cardinal Text Direction Examples (IOF, 2018)*

IOF symbols	Relative	Cardinal
	Bottom right corner	Inside SE corner
	Front right doorway	NW doorway
	Front left doorway	NW doorway stairs top

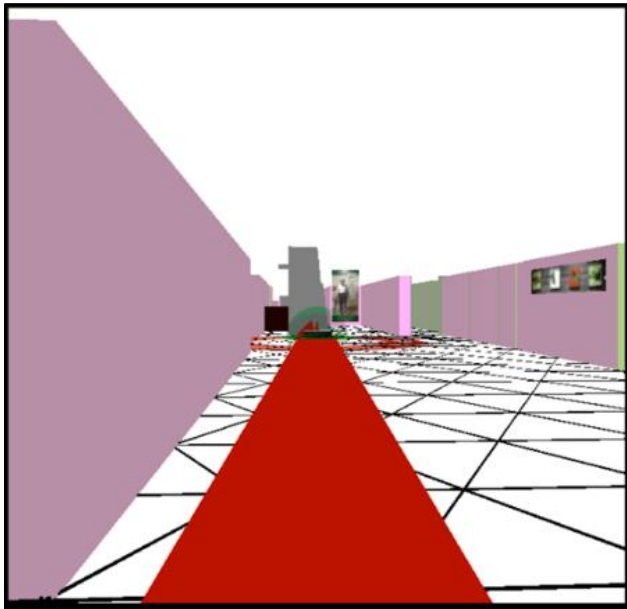
**Figure 29**

*Third (Allo) View*



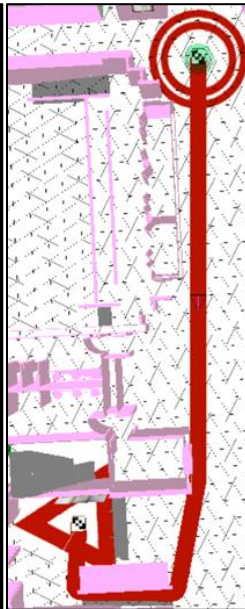
**Figure 30**

*First (Ego) View*



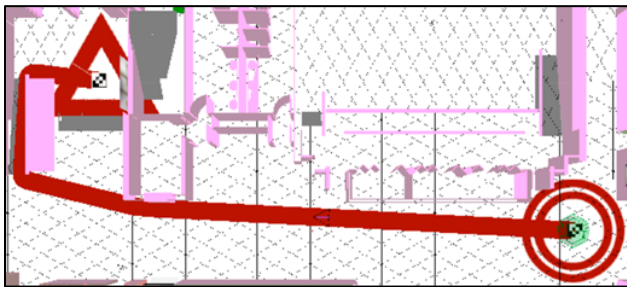
**Figure 31**

*Top Forward (Geo) View*



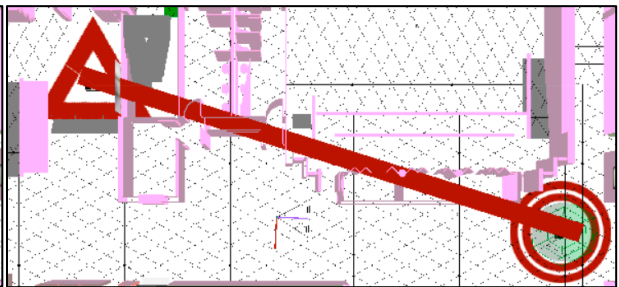
**Figure 32**

*Top North (Geo) View*



**Figure 33**

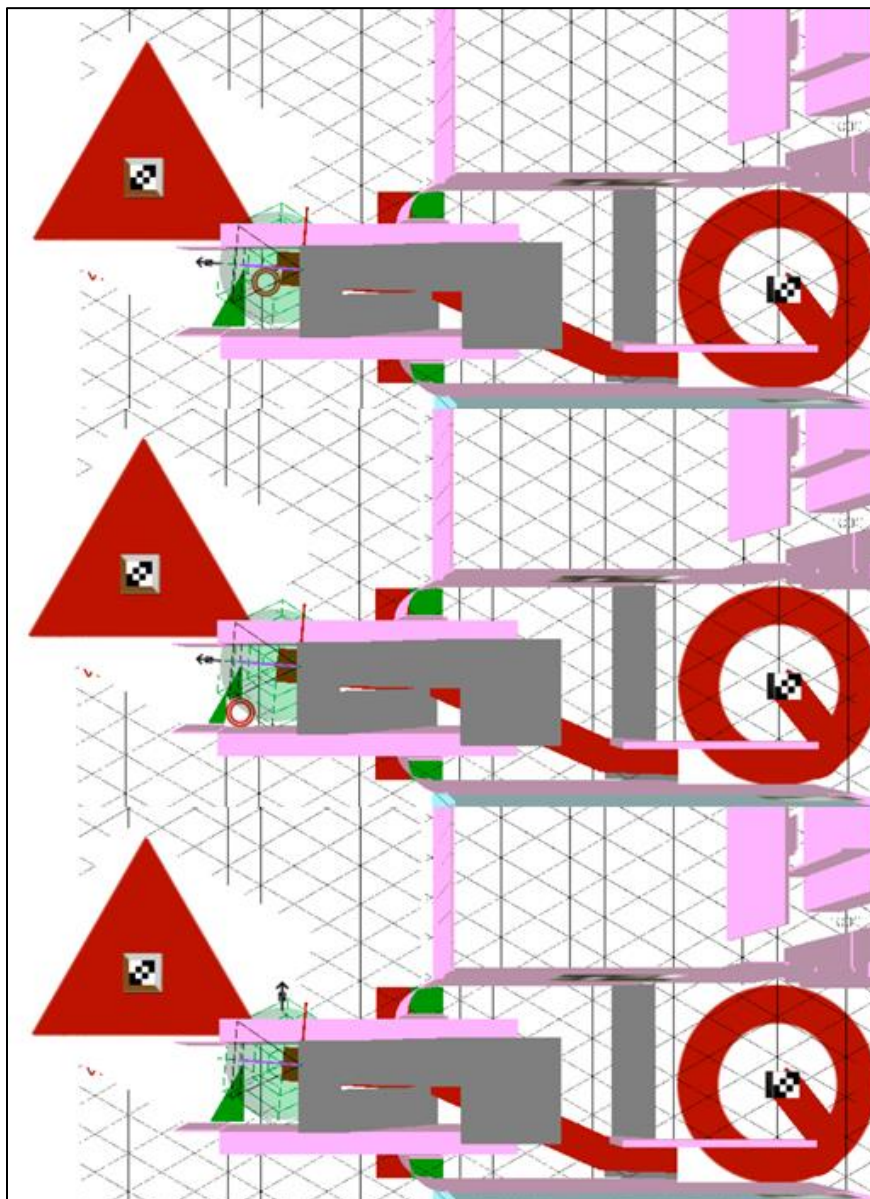
*Top North (Geo) View With Straight Path*



The red double circle finish symbol illustrates the overview distances and directions on the transparent compass disk. For distance, the relative distance of the double circle from the center to the edge of the disk indicates the user's distance along the path. For example, if the symbol is located at half the disk's radius, then the user is halfway to the finish. For the direction overview the double circle is always positioned along the perimeter of the disk.

**Figure 34**

*Distance and Direction, Direction and Compass Overview Examples*



## **Chapter V – Experiment and Discussion**

Chapter V discusses the AEGN system experiment. The first section of chapter V introduces the research design and hypotheses. The second section of chapter V, experiment design, describes the experiment and data collection. The third section of chapter V provides a detailed description of the test plan scenarios and phases. The fourth section of chapter V, data analysis, presents an evaluation plan for the quantitative analysis of the data collected. The analysis assesses the effectiveness and performance of the proposed methods. The fifth section of chapter V reports the findings from the data analysis and discusses the significance of those findings.

### **Research Design and Hypotheses**

This section describes the research design and hypotheses. This research design includes recording and analyzing quantitative measurements with the application. The research measurements subsection describes the quantitative application measurements of the application's algorithm performance for path generation, task allocation, calibration, spatial cognition, and navigation. The hypotheses subsection derives the hypotheses based on the research measurements.

### **Research Measurements**

This subsection describes the quantitative performance measurements of the algorithms for path generation, task allocation, calibration, spatial cognition, and navigation. Optimal performance is based not only on the fastest time to process but also on personal and environmental factors. Personal factors include security authorization, motion mode, user velocity, task preference and SCORE. Environmental factors are modelled by weighting outside paths to cost more. For example, a building with multiple wings might have shorter outside paths, and the additional weight on outside paths could make it take slightly more time, so the path generated

remains inside.

Path generation traces the paths for both walking and rolling between all six control points and records all 60 distances. Path generation also creates a snapshot of each path. The performance time to calculate a path at each control point is logged as CPU ticks.

Task allocation determines the optimal ordered user tasks based on group user preferences and velocity. It uses the CTM to calculate the time cost of the user's preference on the optimal group allocation time. The performance time to calculate the allocations and CTMs is logged in seconds.

Calibration determines the optimal time clusters to minimize BI, ARW and VRW noise based on many static readings. These time clusters are then used to calibrate zero and first order errors and reduce noise in the navigation algorithm.

Spatial cognition sets the optimal viewing properties of generated paths based on the user's SCORE. The SCORE is based on the four spatial cognition tasks for the destination control point selected by a user. Spatial cognition is set when the user answers the final SCORE question, "Do you understand?" and it is recorded in the log file (Appendix C). The performance time to calculate spatial cognition is logged as CPU ticks.

Navigation sets the horizontal direction of the user Unity agent based on the user's height and the Unity agent will trace the user's route at the INS adaptive velocity when the clutch is active. Whenever there is a new INS, or GNSS reading, then a log is recorded containing the following: motion mode, from location, to destination, PCD values, height, velocity, SCORE values, floors, android\_id, test\_num, INS pose, INS accuracy, INS time, Gyro pose, Gyro, Gyro time, GNSS position, GNSS accuracy, and GNSS time. For example, the first and last landmark should be in a space where GNSS satellite signals are available to provide a comparative

measurement for determining the INS drift. The performance time to calculate Navigation is logged as CPU ticks. The application's pose will also be recorded at fiducial markers based on the INS, and the AR corrected INS.

### **Hypotheses**

In this subsection, I propose hypotheses based on the research measurements. I used the research measurements that are described in the previous subsection to develop these hypotheses:

H1: Path generation finds the optimal path based on the motion mode online without pause in the GUI.

H2: Task allocation determines the optimal ordered user tasks based on group user preferences and time. The allocation takes less than 1min.

H3: Calibration sets the INS direction of movement along a route when the clutch with calibration is active for 180s online without pause in the GUI.

H4: Spatial Cognition is initialized with an ART pose and starts the optimal viewing properties of generated paths based on the user's SCORE online without pause in the GUI.

H5: Navigation is corrected with the ART pose and restarts navigation, after 180s the INS position is within  $\pm 3\text{m}$  of the actual position, it is more accurate than the gyro and GNSS positions, but less than the corrected INS poses, and the workflow to create the AR learning environment should be less than 4hr.

### **Experiment Design**

This section is divided into two subsections describing data collection and performance evaluation.

#### **Data Collection**

This subsection describes the process of data collection. The path generation, task

allocation, spatial cognition and navigation automatically log functionality and performance logs. Also, snapshots of the applications are taken to capture the application state. The data was uploaded for data analysis after running the experiment.

### **Performance Evaluation**

This subsection describes the technical based evaluation, which implemented the following:

- Similar systems used GNSS, gyro, and INS methods for the localization feature.
- A recording function to log performance time, accuracy, user time, and ease results.
- A detailed test plan to simulate using this system was divided into the following test scenarios: path generation, task allocation, calibration, spatial cognition, and navigation (Table E1).
- The evaluation methodology for these scenarios used the International Organization for Standardization 18305 (ISO, 2016) standard categories (Ahmed et al., 2020), which provided a generic template to analyze localization system requirements (Appendix E). The test method type was either simulated or real, and the test level was the overall system, internal components, or both. The test repeatability was either testing simultaneously or under the same conditions. Repeatability means the test can be repeated with the same procedures, location, operator, and equipment simultaneously or at different times under the same conditions. The test reproducibility means the data, methods and tools are fully described and available to different operators to test on different equipment.
- The test methodology also followed the Evaluating Ambient Assisted Living (EvAAL) test phases: preparation, execution, and dissemination (Barsocchi et al., 2012). The dissemination test phases describe what data was recorded.

- Individual systems were used individually, or multiple systems were used simultaneously based on the pre-defined scenarios in the test plan. Mobile applications ran on real Project Tango Tablet (Roberto, 2016) and Alcatel 3V and emulated Pixel, Pixel 2, Nexus 5 and Nexus 5X. Back-end processes ran on an Ubuntu server.

### **Test Plan**

This section provides a detailed description of the test plan scenarios and phases. Each test is aligned to evaluate a hypothesis. There are test scenarios and phases subsections for path generation, task allocation, calibration, spatial cognition, and navigation.

#### **Path Generation Test Scenario**

This subsection describes the Path Generation Test Scenario to evaluate if the path generation finds the optimal path based on the motion mode online without pause in the GUI as proposed by hypothesis H1. The test simulates one pedestrian walking and rolling. Simulation is automated between all the planned control points in both directions stopping at the finish control point in each path. Simulation is manual between unplanned points to test how the user personalization (mobility, security) and standard assumptions (safety, privacy, clearance, indoor) impact the path generation regarding objects affecting navigation (i.e., ramps, stairs, elevators, etc.). The Athabasca University Main Campus (AU) and the Mount Pleasant business strip (MP) cover an area of approximately 15000m<sup>2</sup> and 40000m<sup>2</sup>, respectively. The AU test was divided into two sets of walking and rolling points for six automated control points corresponding to the six zones and 257 manual points (Table F1) for specialized functionality. The automated path generation script creates snapshots of the paths. The manual tests included examining if wheelchair employees could be directed up ramps into the server room from zone 1 to 2 except when there was a fire which would activate the halon fire suppression system. The MP business strip had

automated walking and rolling tests for six restaurant entrance points which produced path snapshots which were examined to ensure the paths traced took the fastest routes using the available pedestrian crosswalks. Both sites, AU and MP, used a local site reference system with external overlays (i.e., aerial photos) imported after transforming latitude and longitude WGS84 coordinates. The AU 3D model was extruded from 2D AutoDesk drawings, and the MP 3D model was created with a Tango Tablet Constructor scan to verify the scale and position of aerial photos. Distance and time metrics were compared to Google Map Directions. CPU processing ticks were used as a performance metric for the A\* algorithm.

### **Path Generation Test Phases**

This subsection describes the Path Generation test preparation, execution, and dissemination phases (Appendix F). The test preparation involved importing the 3D models into the path generation scene. There was automated execution between the control points for a pedestrian walking and wheelchair rolling. Manual execution was required to test 257 other test points for various combinations of personalization and navigation features. The output was disseminated to the database, sprite atlas and performance logs for usage and analysis.

### **Task Allocation Test Scenario**

This subsection describes the Task Allocation Test Scenario to evaluate if the task allocation determines the optimal ordered user tasks based on group user preferences and time and that the allocation takes less than 1min as proposed by hypothesis H2. Brute force determines the optimal ordered user tasks based on group user preferences and velocity. The task allocation uses the CTM to calculate the time cost of the user's preference on the optimal group allocation time. Performance is evaluated based on comparing the distance and time results to the same order of destinations in Google Maps. In addition, the processing performance of the brute force mTSP

algorithm is evaluated for 1, 2 and 3 member groups in milliseconds.

### **Task Allocation Test Phases**

This subsection describes the Task Allocation test preparation, execution, and dissemination phases (Appendix G). Task allocation is currently only configured to process one site at a time, and while there can be multiple groups, a user can also only be in one group at a time. The data for the MP business strip was loaded for execution. The schedule information for the calendar event tasks for the MP control points was entered into AEGN Coordinator. The test preparation involved manually setting up groups of 1 to 3 consisting of different combinations of 4 users. The task allocation is automatically processed on the server after users select their preferences. The output was disseminated to the database for usage and analysis.

### **Calibration Test Scenario**

This subsection describes the Calibration Test Scenario to examine if the calibration sets the INS direction of movement along a route when the clutch with calibration is active for 180s online without pause in the GUI as proposed by hypothesis H3. This scenario involves simulated stationary and real pedestrians walking a predefined straight route to test the calibration settings. The simulated stationary test simultaneously runs the gyro and INS. The device is not rotated or accelerated as opposed to the real test. The real test runs the GNSS, gyro and INS and introduces additional device rotation and acceleration errors that are not included in the simple static calibration technique. While both tests are repeatable, only the simulated test is reproducible because the real test does not record all the continuous sensor data, which is affected by the rotation and acceleration of user handling. Both tests follow a straight path between the 4th Spot Restaurant, the 1st target, and its muster point, which was located 1km north, to extend the test distance and time. The extension gives the test several opportunities to automatically reset to reach

the 180s tracking goal within a 3m drift error. The drift error is measured relative to a virtual ground truth reference Unity agent, which moves at the expected velocity based on the user height along the path between the two targets. Time synchronization between the reference Unity agent and the real user walking is aided visually and audibly with the aerial photos, hexagonal grid map overlay, and a metronome for step cadence. The grid adapts to the user step length based on height. In the first-person perspective, the metronome ping (0.6s) indicates one step counted with one foot (left or right) to the center of the hexagonal radar grid location on the map overlay. The 3D ping sound gets louder as the user moves closer to the finish. As the user moves with the reference Unity agent, they can see their position relative to the landmarks around them, as seen in the aerial photos display where the reference Unity agent is located from a top forward perspective. Top forward is useful because the user can be zoomed out enough to see landmarks such as sidewalks, roads, and buildings but still see the orientation because the landmarks rotate around the reference Unity agent. General performance of the navigation algorithm is measured in quaternions per second.

### **Calibration Test Phases**

This subsection describes the Calibration test preparation, execution, and dissemination phases (Appendix H). Calibration is required after any physical or temperature shock which affect sensors. The calibration warm-up creates a log file for the Allan Variance analysis to determine the cluster times for the calibration. The calibration was run along each axis in both directions. A green bubble moves up the current axis. The smartphone was clamped on a level table, but any plumb vertical or flat level surface such as a door or window frame could be used. A series of simulated 1km static tests were used with time constants between 0.5 and 1.0 to determine the filter coefficient which would best balance the weight attributed to the gyro and accelerometer in

the complementary filter to yield the longest running time with a drift error less than 3m. A real 2km dynamic test was used to evaluate the calibration performance with the rotations and accelerations from user handling while walking. It is necessary to see the reference Unity agent relative to the landbase to synchronize the real and virtual positions. It is easiest to view the screen when the lighting is constant at twilight, night, or on a cloudy day to minimize reflected glare. The complementary filter helps by stabilizing the image perspective to allow the user to move and tilt the device either inadvertently or intentionally for the comfort of their arms, hands, or eyes (e.g., to avoid reflected glare on the screen). The output was disseminated to the player preferences and log files for usage and analysis. The raw log file is used for the Allan Variance analysis, the xml log file is used for functional analysis, and the performance log file is used for analyzing the algorithm.

### **Spatial Cognition Test Scenario**

This subsection describes the Spatial Cognition Test Scenario to examine if the spatial cognition is initialized with the ART pose and starts the optimal viewing properties of generated paths based on the user's SCORE online without pause in the GUI as proposed by hypothesis H4. It is useful to set the velocity to 1.2m/s because that makes the parallel lines of the grid 1m apart to help verify answers. This test evaluates the ART Pose and INS components one after another as part of the system under the same conditions. The test is repeatable but not reproducible because the rotations and accelerations of the user handling are not continuously recorded for replication of the test conditions. The processing performance is evaluated in milliseconds and CPU ticks.

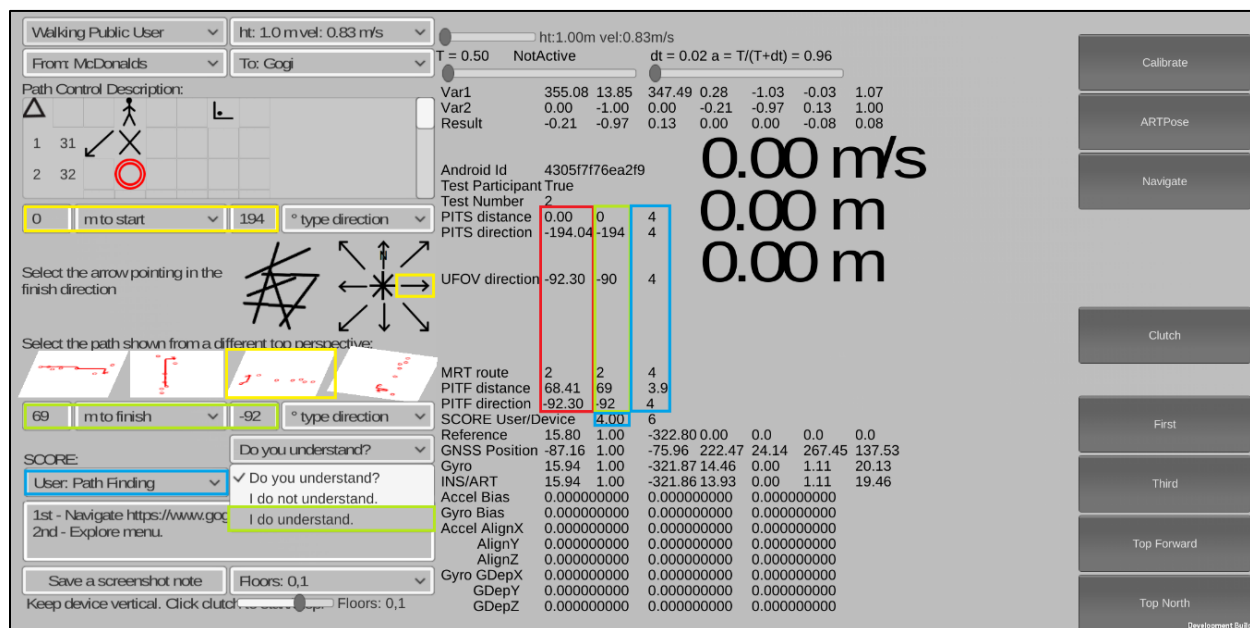
### **Spatial Cognition Test Phases**

This subsection describes the Spatial Cognition test preparation, execution, and dissemination phases (Appendix I). Preparation took about 4hr to take photos, measurements, and

then update the target database and add the new control points to the Unity Navigate scene. The application was recalibrated as well. This test was simulated using photos of the control points to initialize the start target using an ART Pose. Then a different destination target was selected to trigger the guiding Unity agent. The Spatial Cognition Learning Tasks were answered, and then this was repeated for each of the other target destinations before reinitializing the start target with each of the control point photos comprising a total of 30 test runs. This generates several questions to help you understand where you are going and determines how to display the map and PCD based on your understanding or SCORE (Figure 35). The PCD follows the IOF standard.

**Figure 35**

*Scoring Fields for Answer Keys (Red Boxes), Answers (Yellow Boxes) and Scores (Blue Boxes)*



The better you understand where you are going, the less information and distraction will be displayed. The objective of this app is to improve your mental map SCORE, so if the device SCORE is low for whatever reason, then it does not matter because you are not lost. You should only need to use this app when you are somewhere that is new to you, or it has

changed since you were last there. The 4D spatial cognitive model helps answer the questions. For example, to help understand distance and directions, it is useful to note that the parallel lines in the hexagonal grid are one stride apart. Those hexagonal lines form  $60^\circ$  angles, and all text is oriented so north is on the top. This can help you equate what you see on the map and what you see with your own eyes because your central binocular FOV is about  $60^\circ$  without eye movement,  $120^\circ$  with eye movement,  $180^\circ$  with monocular peripheral vision and  $360^\circ$  with head and trunk movement. The output was disseminated to the player preferences and log files for usage and analysis. The performance log file is used for analyzing the algorithm.

### **Navigation Test Scenario**

This subsection describes the Navigation Test Scenario to examine if the navigation is corrected with the ART pose and restarts navigation, after 180s the INS position is within  $\pm 3\text{m}$  of the actual position, it is more accurate than the gyro and GNSS positions, but less than the corrected INS poses, and the workflow to create the AR learning environment should be less than 4hr as proposed by hypothesis H5. This scenario involves a real pedestrian walking a predefined route to test the accuracy of the competing components and system. This repeatable scenario simultaneously tests the GNSS, gyro and INS as part of the system. It does not record the sensor reading continuously, so it is not reproducible. The scenario is run with a slower velocity of  $0.83\text{m/s}$  to make it easier to hold steady and handle the device. The application performance relies on synchronization with the reference Unity agent, the same as the calibration test. The general performance is also measured in quaternions per second and CPU ticks.

### **Navigation Test Phases**

This subsection describes the Navigation test preparation, execution, and dissemination phases (Appendix J). The Navigation test is like the Calibration real dynamic test except navigating

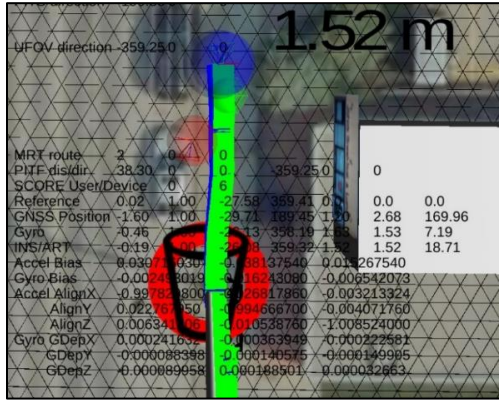
routes between the six control points. Each route selected is about 300m long to allow time to reach the goal of navigating 180s without drifting more than 3m (Table J1). Most of the preparation for this test was already made for the Calibration and Spatial Cognition tests. The preparation requires recalibration for the outdoor conditions. It was run early in the morning because if the experiment is run at night, twilight, or on a cloudy day, then it is easiest to view the screen without glare, and there are no shadows or glare from the sun, which might affect the AR target detection. It was also run in the spring before leaves had grown on the trees, and control points were set back away from buildings to allow the GNSS to work well for comparison. A slow walking speed of 0.83m/s was selected to help make it easier to handle the device in a steady manner. The output was disseminated to the player preferences and log files for usage and analysis. The xml log file is used for functional analysis, and the performance log file analyzes the algorithm.

### **Data Analysis**

This section presents an evaluation plan for the quantitative analysis of the data collected. The analysis will assess the effectiveness and performance of the proposed methods when available comparisons with different methods, such as other sensors and algorithms, will be made. Performance calculations will be made in seconds or ticks depending on the common measurement available for different methods. For example, competing navigation algorithms measure performance in terms of quaternions per second. Other comparison methods will be more descriptive, such as how correct and complete the path details of AEGN are compared to Google Maps. Screenshots are provided to illustrate path descriptions and displacement. For example, the best accuracy is readily apparent with error bubbles simultaneously illustrating the distance error from the reference agent position for competing methods (Figure 36). The data analysis is divided into path generation, task allocation, calibration, spatial cognition, and navigation subsections.

**Figure 36**

*Error Bubbles for GNSS (Red), INS (Green), Gyro (Blue) and Reference Agent (Green Hexagon)*



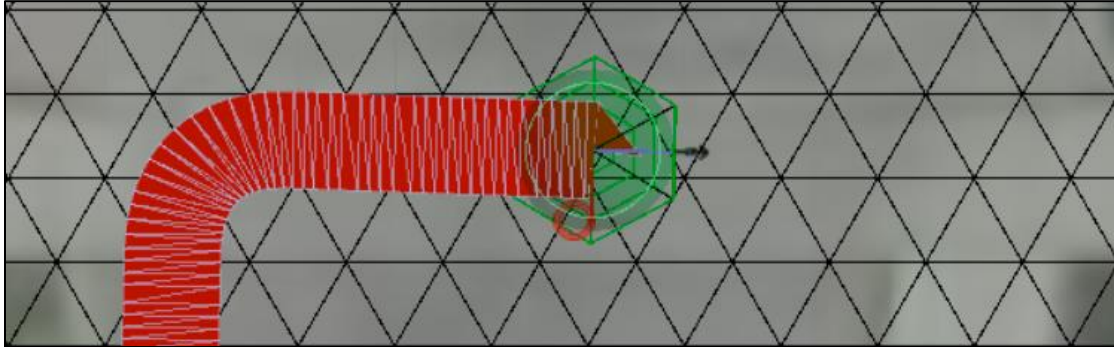
### Path Generation

The performance of the path generation algorithm was very fast,  $497 \pm 142$  ticks for each path. The algorithm's effectiveness is limited on longer paths in complicated environments with many routes to compare, so the Unity agent recalculates the path at control points, reducing longer paths into segments. This recalculation was not noticeable as the Unity agent traced the path even at the fastest walking speed. The speed of the Unity agent tracing the path (1.67m/s) was the fastest walking speed to reduce the processing time.

Using a faster speed to trace the routes can speed up processing, but it affects the path line. For example, a 4m/s speed changes the path, as can be seen in detailed Figures 37 and 38 (Full paths Figure D1 and D2), where the faster Unity agent takes a wider turn. The faster Unity agent takes a longer circumference on corners than a slower Unity agent, which takes a shorter, sharper turn. However, if the faster Unity agent stops before turning, then the path is not longer (Figure 39). Wider turns may or may not affect the overall length of the path if the turns are in the direction of the destination. If path generation needs to be faster, then force the Unity agent to stop before turning to create an accurate path. Changing the user's height while tracing a route will also add to the length of the route because the vertical move will be added to the 3D distance.

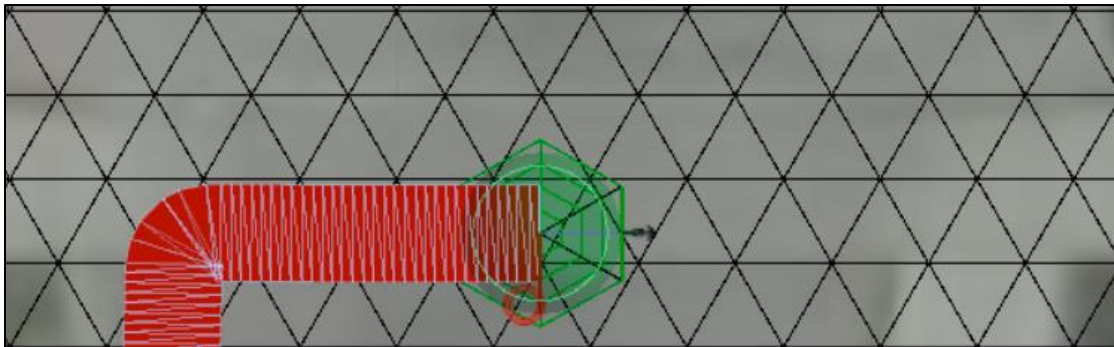
**Figure 37**

*Unity Agent Running Fast (4m/s) and Turning a 90° Corner*



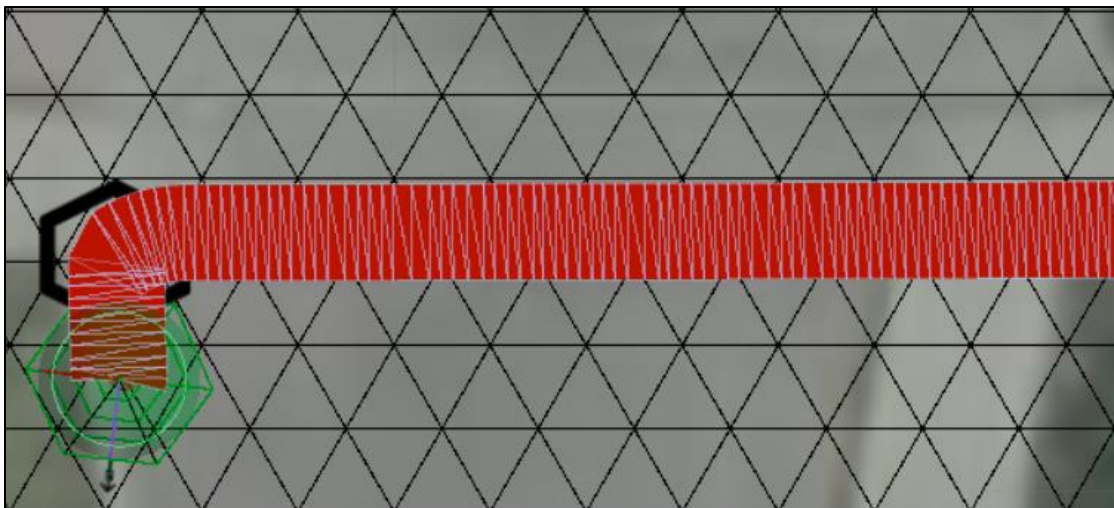
**Figure 38**

*Unity Agent Walking Slowly (0.83m/s) and Turning a 90° Corner*



**Figure 39**

*Unity Agent Running Fast (4m/s), Stopping, and Turning a 90° Corner*



The paths were more effective in modelling realistic walking routes than Google Maps. Google Maps routes paths down the center line of streets, indoor paths are limited to preset paths between points, and Google Maps does not model paths specifically for wheelchair-accessible routes. The AEGN fastest path for a single user to navigate to all six control points at 0.8343m/s was estimated to be 652.02m in 781s with a Mean Absolute Error (MAE) of  $0.06 \pm 0.04$ m or  $0.07 \pm 0.05$ s for the five path segments (Table 4). In comparison, the Google Maps fastest path for a single user to navigate all six control points was estimated to be an average speed of 1.41m/s to go 673m in 477s with a MAE of  $58 \pm 35$ m or  $64 \pm 34$ s for the five path segments. The high Google Maps errors were reflected in the summary estimates of 650m and 540s, which rounded the estimates to the closest 50m and 60s increments (Figure D3).

**Table 4**

*Google Maps (Google, n.d. b) and AEGN Path Distances and Times at Various Speeds*

Path	Estimated				Actual	
	Google Maps		AEGN (2.0m/s)		AEGN (0.8343m/s)	
	Distance (m)	Time (s)	Distance (m)	Time (s) (0.8343m/s) <sup>a</sup>	Distance (m)	Time (s)
0_TO_5 <sup>b</sup>	72	58	92.95	111.40	92.86	111.30
5_TO_4 <sup>c</sup>	108	60	142.37	170.64	142.25	170.49
4_TO_3 <sup>d</sup>	362	240	264.51	317.03	264.56	317.09
3_TO_2 <sup>e</sup>	33	59	113.89	136.50	113.86	136.46
2_TO_1 <sup>f</sup>	98	60	38.32	45.93	38.34	45.96
Total	673	477	652.02	781.48	652.46	782.00
MAE	58 $\pm 35$	64 $\pm 34$	0.06 $\pm 0.04$	0.07 $\pm 0.05$		

*Note:* <sup>a</sup>Path generated at 2.0m/s but navigated at 0.8343m/s. <sup>b</sup>Table D1. <sup>c</sup>Table D2. <sup>d</sup>Table D3.

<sup>e</sup>Table D4. <sup>f</sup>Table D5.

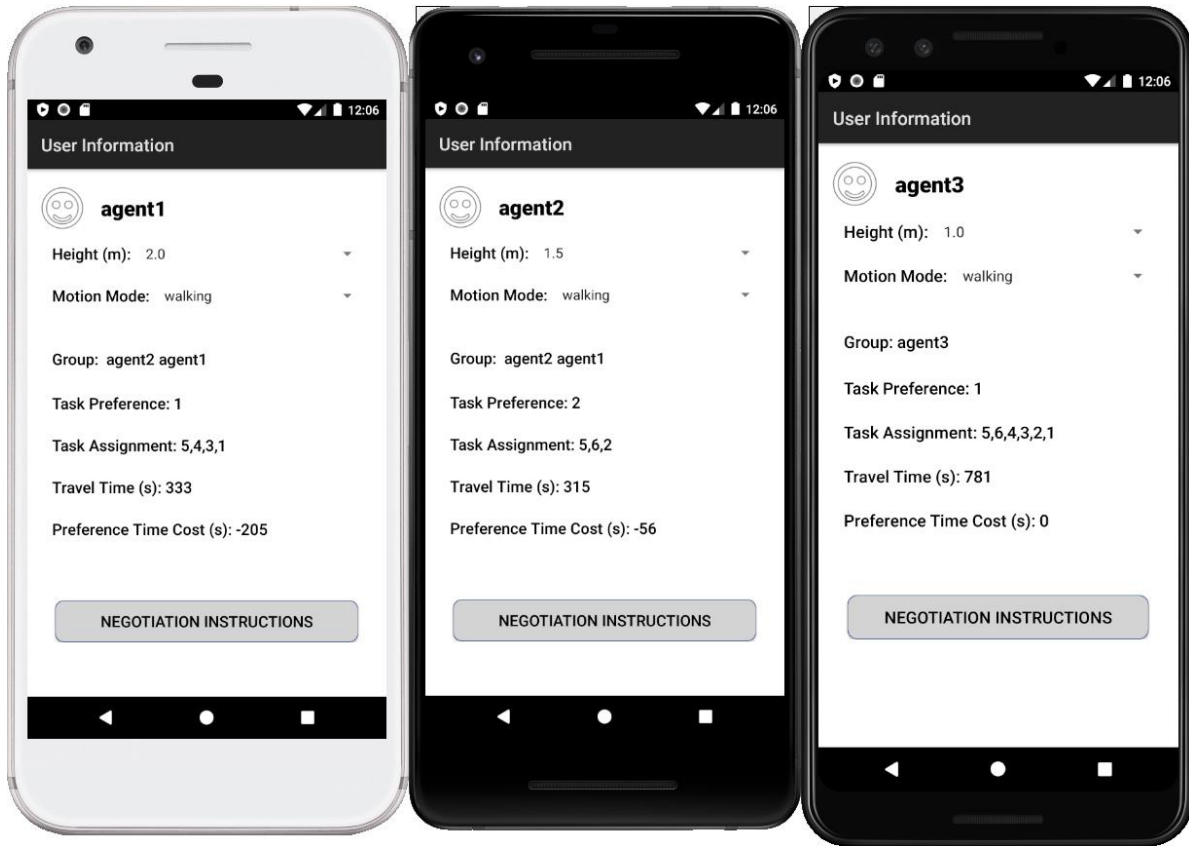
### Task Allocation

The performance of the task allocation algorithm was  $11.2 \pm 8.7$ s,  $14.4 \pm 2.7$ s, and  $20.2 \pm 2.8$ s for one, two and three user groups, respectively. The problem was constrained to three users with six destinations where one was fixed for all to start at, and each user chose one preference to reduce

the number of permutations with ordered combinations. For one user being allocated all tasks, the problem is simplified to order (5,6,4,3,2,1) of shortest travel time (781s) (Figure 40). This approximation of the travel time is only 0.5s faster than the actual travel time at 0.8343m/s (782s) (Table 4). AEGN is more effective than Google Maps by providing an ordered task assignment to minimize travel time. Google Maps also does not consider multiple users, wheelchair access or different speeds.

**Figure 40**

*Two User Group and One User Group Task Allocation Examples*



The task allocation outputs the assignment, travel time and preference time cost based on the users' height, motion mode and preference. All users start from the same target (i.e., 5), which is not processed in the permutations. Each user chooses a preference included in their own permutations but not the permutations of the other users. The CTM calculates the preference time

cost (pay), which is the difference between the sum of the other users' travel times for the current group (tax) and the alternative groups where the user does not participate (sum) (Table 5). In this example, time is the currency which measures the social cost of a user's benefit on the social welfare of the other users. When a user is allocated more tasks at the expense of other users who would have been allocated those tasks had the user not participated, the user who benefitted from the extra allocated time conceptually owes that to the users who lost out on that opportunity. These time costs do not necessarily need to be paid out, but this is a useful way to conceptualize what the preference time cost reveals as a task management tool to evaluate social welfare maximization. For example, the user assigned the tasks could be considered to cover a shift for another user who could pay back the benefit by trading a shift. In a practical application, time can be conceived as earnings balanced in the budget by either banking time or taking time off. No payments indicate users that do not affect the allocations of other users. Pivotal users have payments, and large payments may indicate some inequity in distributing tasks.

Suppose management assumes that users cooperate by revealing their preferences truthfully, and the process allocates tasks to minimize the group travel time. In that case, large payments might indicate a problem with user or environment settings. For example, if some users are much slower, lack security authorizations, or use a wheelchair in an environment where tasks are widely distributed in secure areas with accessibility challenges, then those users would likely get few tasks. Management could address these issues by starting from a different target, renting a motorized wheelchair, and requesting a security pass. Rerunning the task allocation would probably result in smaller payments or social costs indicating higher social welfare maximization.

If management assumes that users are not cooperating and are manipulating or gaming, the system to either acquire or avoid tasks, then they can investigate the settings and distribution. For

example, if there were a lone task in one direction and a cluster of tasks in the opposite direction, then if a user always picked the lone task, they would be less likely to be allocated any of the others. They could enforce the CTM payments to incentivize users to be truthful in their preferences or pay the social cost to ensure tasks are allocated fairly.

**Table 5**

*Task Allocation Example Using CNP With CTM Calculations for Various Preferences*

Jason agent	Three Jason agent group												
						Two Jason agent group							
										One Jason agent			
	Speed (m/s)	Pref	Alloc	Sum (s)	Tax (s)	Alloc	Sum (s)	Pay (s)	Tax (s)	Alloc	Sum (s)	Pay (s)	Pay (s)
1	<b>1.67</b>	<b>1</b>	<b>541</b>	<b>283</b>	<b>672</b>		<b>810</b>	<b>-138<sup>a</sup></b>					
2	<b>1.25</b>	<b>2</b>	<b>562</b>	<b>315</b>		<b>56421</b>	<b>453</b>		<b>357</b>		<b>781</b>	<b>-424<sup>b</sup></b>	
3	<b>0.83</b>	<b>3</b>	<b>53</b>	<b>357</b>		<b>53</b>	<b>357</b>			<b>564321</b>	<b>781</b>		<b>0<sup>e</sup></b>
1													
2										<b>564321</b>	<b>521</b>		<b>0<sup>e</sup></b>
3									<b>453</b>		<b>521</b>	<b>-68<sup>b</sup></b>	
1						<b>56421</b>	<b>340</b>		<b>357</b>		<b>781</b>	<b>-424<sup>c</sup></b>	
2					<b>640</b>		<b>697</b>	<b>-57<sup>a</sup></b>					
3						<b>53</b>	<b>357</b>			<b>564321</b>	<b>781</b>		<b>0<sup>e</sup></b>
1										<b>564321</b>	<b>390</b>		<b>0<sup>e</sup></b>
2													
3									<b>340</b>		<b>390</b>	<b>-50<sup>c</sup></b>	
1						<b>5431</b>	<b>333</b>		<b>315</b>		<b>521</b>	<b>-206<sup>d</sup></b>	
2						<b>562</b>	<b>315</b>			<b>564321</b>	<b>521</b>		<b>0<sup>e</sup></b>
3					<b>598</b>		<b>648</b>	<b>-50<sup>a</sup></b>					
1										<b>564321</b>	<b>390</b>		<b>0<sup>e</sup></b>
2									<b>333</b>		<b>390</b>	<b>-57<sup>d</sup></b>	
3													

*Note:* Preference (pref) and CNP task allocations (alloc) are task numbers. **Bold** values are related to the CNP calculations. *Italic* values are related to the CTM calculations. **Bold and italic** values are related to both the CNP and CTM calculations. <sup>a</sup>CTM pay for 3 Jason agent group (1,2, and 3). <sup>b</sup>CTM pay for 2 Jason agent group (2, 3). <sup>c</sup>CTM pay for 2 Jason agent group (1, 3). <sup>d</sup>CTM pay for 2 Jason agent group (1, 2). <sup>e</sup>CTM pay for individuals is 0.

Preferences reduce the number of permutations for the mTSP, while CTM adds permutations to the task allocation. CTM runs the alternate group permutations where each user does not participate. Table 5 illustrates this where the task allocations are run for each user in the three Jason agents' group and then run again in the two Jason agents' group for each Jason agent where they do not participate. Table 6 shows that only  $3*3!=18$  three Jason agents' mTSP permutations, but there are another  $3*2*4!=144$  CTM two Jason agents' permutations for a total of 162 permutations for a three Jason agents' group with six targets, one shared start target, and one preference for each user. The combined permutations are still less than the mTSP with six targets and one shared start target ( $3*5!=360$ ). Adding users with a preference reduces the number of mTSP permutations until the destination targets equal the number of users, at which point the problem is just a matter of selecting user preferences. Alternatively, the mTSP problem is reduced to a TSP problem with one user running solo, which has  $n!$  permutations for  $n$  destinations. The two Jason agents' group has the most permutations because the CTM must process two TSP alternative sets of permutations for each individual user to calculate the social cost. On a Deterministic Turing Machine (DTM), the most destination targets ( $n$ ) that could be theoretically processed for the TSP brute force computational complexity ( $O(n!)$ ) is  $10!$  or 3,628,800 permutations (Table 6). If the implementation achieved perfect optimization and processed one permutation/tick, then  $10!$  permutations would process in 0.36288s,  $11!$  in 4s,  $12!$  in 48s,  $13!$  in 10.38min,  $14!$  in 2.42hr and  $15!$  in 1.51 days. This performance is unrealistic in practice, even when processing the task allocation on the server side. In reality the processing was slowed down to 13 permutations/s by complications such as the CNP messaging. The brute force approach to mTSP only works for a small number of destinations and users. A tractable solution requires a suboptimal heuristic approximation to process more destinations and users in polynomial time.

**Table 6***Task Allocation Computational Complexity*

Targets (n)	Users (m)	mTSP <sup>a</sup>	CTM <sup>b</sup>	mTSP+CTM <sup>c</sup>
11	1	3628800	0	3628800
	2	725760	7257600	7983360
	3	120960	2177280	2298240
	4	20160	483840	504000
	5	3600	100800	104400
	6	720	21600	22320
	7	168	5040	5208
	8	48	1344	1392
	9	18	432	450
	10	10	180	190
10	1	362880	0	362880
	2	80640	725760	806400
	3	15120	241920	257040
	4	2880	60480	63360
	5	600	14400	15000
	6	144	3600	3744
	7	42	1008	1050
	8	16	336	352
	9	9	144	153
9	2	10080	80640	90720
	3	2160	30240	32400
	4	480	8640	9120
	5	120	2400	2520
	6	36	720	756
	7	14	252	266
	8	8	112	120
8	1	5040	0	5040
	2	1440	10080	11520
	3	360	4320	4680
	4	96	1440	1536
	5	30	480	510
	6	12	180	192
	7	7	84	91
7	1	720	0	720
	2	240	1440	1680
	3	72	720	792
	4	24	288	312
	5	10	120	130
	6	6	60	66
6	1	120	0	120
	2	48	240	288
	3	18	144	162

*Note:*  $n$  = number of targets, including the start target.  $m$  = number of users.

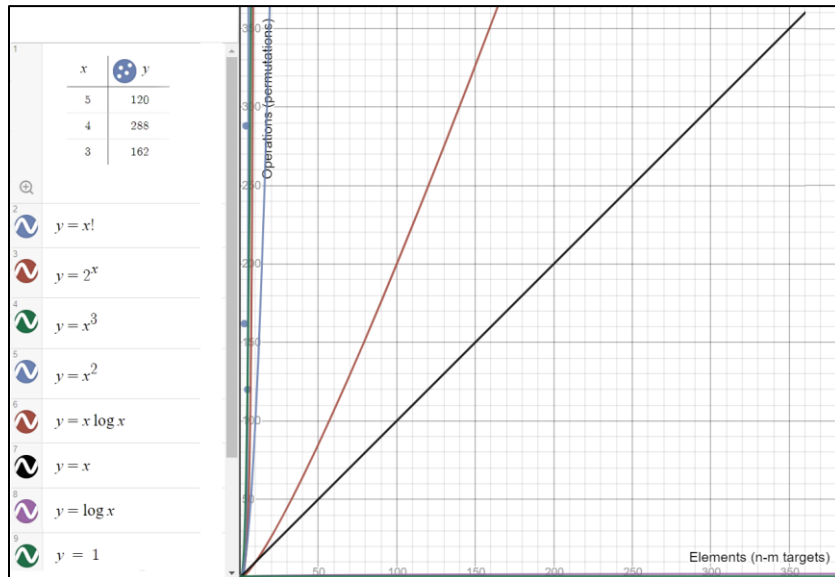
<sup>a</sup>mTSP permutations =  $O(m*(n-m)!)$ . <sup>b</sup>CTM permutations =  $O((m-1)*m*(n-m+1)!)$ .

<sup>c</sup>mTSP+CTM permutations =  $O(-m^3+m^2(n+2)-mn*(n-m)!)$ .

If the brute force optimization problem is considered a decision problem where the algorithm iteratively decides if each distance is less than the last calculated minimum distance, then it is also interesting to consider the complexity classification (Figure 41). Non-deterministic Polynomial time (NP) includes the orders of complexity from constant  $O(1)$ , logarithmic  $O(\log n)$ , linear  $O(n)$ , log linear  $O(n \log n)$ , quadratic  $O(n^2)$  and cubic  $O(n^3)$  from fastest to slowest. Superpolynomial time is the slowest, which includes the worse exponential  $O(2^n)$  and worst factorial  $O(n!)$  complexities. The mTSP, CTM and mTSP/CTM problems cannot be solved or verified in polynomial time, so their complexity classification is NP-hard and NP-complete, respectively.

**Figure 41**

*Complexity Classification*



*Note:*  $n$  = number of targets.  $m$  = number of users.  $x = n-m$ .  $y$  = permutations.

### Calibration

Without the calibration the velocity is extremely erratic. The performance of the calibration algorithm is good enough to make navigating short routes possible. For example, Woodman (2007) improved the INS drift error from 150m to 5m with a calibrated sensor fusion algorithm, but that was over only 60s. The desired 180s path within 3m error is extremely ambitious for a consumer grade INS. If the speed is off by 0.01m/s, then over half, 1.8m, of the 3m error budget is already used in 180s. This is intended to make a consumer grade INS perform at the low-end of industrial grade to provide better GNSS-denied navigation for use indoors (Table 7).

**Table 7**

*Performance Grades of Inertial Sensors (VectorNav, 2020, p.5)*

Grade	Cost (\$)	Gyro in-run bias stability (°/hr)	GNSS-denied navigation time	Applications
Consumer	< 10	-	-	Smartphones
Industrial	100-1000	<10.0	< 1min	UAVs
Tactical	5000-50000	< 1.0	< 10min	Smart munitions
Navigation	> 100000	< 0.1	Several hours	Military

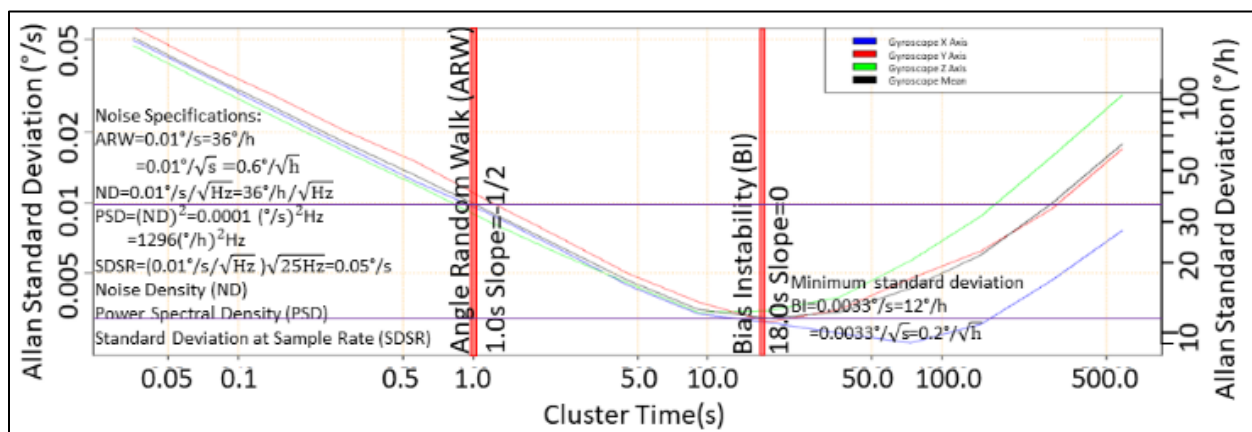
The simulated static and real dynamic tests found that the drift error could be limited to under 3m over a navigation time of 219s and 193s, respectively, similar to high-end industrial grade performance. The stationary calibration does not calculate the errors added by rotation and acceleration during dynamic conditions. While the user can try to hold the device steady, the real dynamic movement adds these errors decreasing the travel time within the 3m error bounds. In a dynamic scenario the user can compensate for these additional errors by turning the device in the opposite direction of the drift to keep on track with the reference Unity agent. If the application has a good static calibration and the user starts with the device pointed forward, then the drift can be expected to wander side to side while primarily being centered. Eventually the rotational drift to a side and the acceleration drift from the reference Unity agent will combine to exceed the 3m

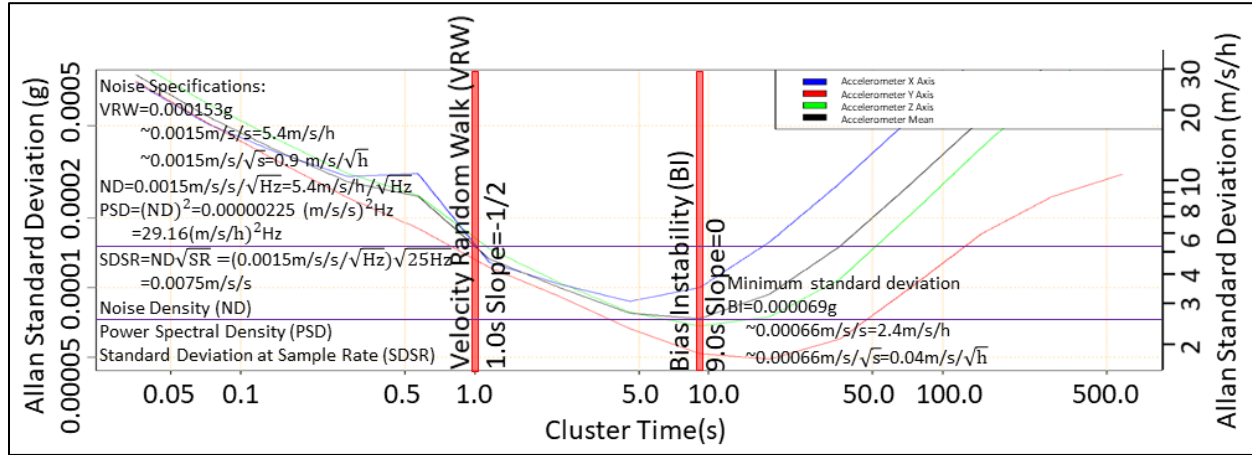
error budget.

The gyro bias of  $12^\circ/\text{hr}$  is used to grade the system as low-end industrial because the gyro bias has the greatest impact on the INS performance (VectorNav, 2020). The \$100 Alcatel 3V phone was also the same cost as a low-end industrial grade inertial sensor. The application needs to be calibrated and tuned well to achieve the best results. The calibration results are heavily dependent on temperature. The device must be warmed up and calibrated in the environment which will be navigated. For indoor navigation this requires recalibration when moving from a room temperature environment such as an office space to an area that is hotter or colder such as a furnace room or a walk-in freezer. The warm-up provides the raw sensor logs to characterize the gyro and accelerometer properties. The Alcatel 3V accelerometer and gyro had a sample rate of about 25Hz or once every 0.04s. A 0.5hr static data sample from the Alcatel 3V accelerometer and gyro was collected at room temperature. The Allan variance analysis was applied to the whole data set. A loglog graph of the three-axis gyro and accelerometer Allan standard deviation to cluster time (Figures 42 and 43) were made using R plot scripts (Appendix A and B).

**Figure 42**

*Gyro Allan Standard Deviation*



**Figure 43***Accelerometer Allan Standard Deviation*

The BI, ARW and VRW noise of the gyro and accelerometer are clearly identifiable on the graphs for the short and long term cluster times where the slope of the line is 0 for BI and  $\frac{1}{2}$  for ARW and VRW. Papoulis (1991) showed the Allan standard deviation error is equal to

$$\sigma(\delta_{AV}) = \frac{1}{\sqrt{2\left(\frac{N}{n} - 1\right)}} \sim \frac{1}{\sqrt{2\left(\frac{1800\text{seconds} \cdot 25\text{Hz}}{18\text{seconds} \cdot 25\text{Hz}} - 1\right)}} \sim 0.07 \sim 7\%$$

where N is the data set count and n is the cluster time count. The gyro BI is  $12^\circ/\text{hr} \pm 7\%$  or  $BI = 12 \pm 0.84^\circ/\text{hr}$  and its  $ARW = 0.6^\circ/\sqrt{\text{hr}} \pm 1.6\%$  or  $ARW = 0.6 \pm 0.01^\circ/\sqrt{\text{hr}}$ . The accelerometer BI is  $0.069\text{mg} \pm 5\%$  or  $BI = 0.069 \pm 0.0035\text{mg}$  and its  $VRW = 0.9\text{m/s}/\sqrt{\text{hr}} \pm 1.6\%$  or  $VRW = 0.9 \pm 0.015\text{m/s}/\sqrt{\text{hr}}$  (Table 8).

**Table 8***Error Terms by Sensor/Grade*

Sensor /Grade	Gyro BI (°/hr)	ARW (°/√hr)	Accelerometer BI (mg)	VRW (m/s/√hr)
MPU-9150 <sup>a</sup>	72000.00	0.30	103.0000	0.234
MPU-9250 <sup>b</sup>	5400.00	0.60	61.0000	0.174
MPU-9250 <sup>c</sup>	360.00	6.00	8.0000	0.174
Consumer <sup>d</sup>	100.00	2.00	10.0000	1.000
Xsens Mtx <sup>e</sup>	37.00	4.70	0.0290	0.084
MotionPak II <sup>f</sup>	14.00	0.50	0.0280	0.070
Alcatel 3V	12.00 ±0.84	0.60 ±0.01	0.0690 ±0.0035	0.900 ±0.015
Industrial <sup>d</sup>	10.00	0.20	1.0000	0.100
Tactical <sup>d</sup>	1.00	0.05	0.1000	0.030
Navigation <sup>d</sup>	0.01	0.01	0.0100	0.010

*Note:* <sup>a</sup>(Invensense, 2013). <sup>b</sup>(Martin, 2016). <sup>c</sup>(Invensense, 2016). <sup>d</sup>(VectorNav, 2020). <sup>e</sup>(Woodman, 2007). <sup>f</sup>(Hou, 2004).

The cluster times are used to minimize the effect of noise on calibrations. The static calibrations were run in both directions of the three axes for these cluster times. Once all the accelerometer and gyro readings were accumulated, the calibration calculated the accelerometer bias and alignment errors (scale-factor and cross-coupling) and gyro bias and g-dependent errors as seen in the calibration interface (Figure 44). The navigation cluster time is used to minimize ARW and VRW noise when navigating, and this is also when the errors are removed from the raw sensor readings.

**Figure 44***Calibration Settings*

Noise Reduction Time Clusters (sec)			
Navigation	1.00	Calibration	9.00
Accel/Gyro		Calibration	18.0
	X	Y	Z
Warm-up Accel	0.038035420	-0.025696850	-1.005950000
Warm-up Gyro	-0.003192982	-0.015842110	-0.006403500
Cluster + Accel	-0.967711900	-1.032972000	-1.007614000
Cluster + Gyro	-0.002277684	-0.017123230	-0.006293820
Cluster - Accel	1.024149000	0.957102900	1.010608000
Cluster - Gyro	-0.002327320	-0.016844240	-0.006583257
Start/time/count	3133.090	1800.010	45093
Accel Bias	0.030744710	-0.040740140	0.018585730
Gyro Bias	-0.002287457	-0.016844500	-0.006611213
Accel AlignX	-0.995930600	-0.024262570	-0.003214871
Accel AlignY	0.021871450	-0.995037400	-0.017779300
Accel AlignZ	0.009869623	-0.005858146	-1.009101000
Gyro GDepX	0.000024818	0.000098945	0.000037336
Gyro GDepY	-0.000191670	-0.000139494	-0.000137117
Gyro GDepZ	-0.000009943	-0.000044619	0.000137218

To tune the complementary filter, the static calibration test was run with time constants between 0.5 and 1.0 in 0.1 increments to find which one produced the longest runs with a drift error less than 3m. Since the warm-up log ran for 1800s and recorded about 45000 readings then the time interval  $dt = \text{time}/\text{count} = 1800\text{s}/45000 = 0.04\text{s}$  refresh time. The Filter coefficient  $a = \frac{\tau}{\tau + dt}$  so the only variable to tune is the time constant  $\tau$ . The optimal time constant for the Alcatel 3V was  $\tau = 0.7$ . This produced the best results ( $219 \pm 27\text{s}$ ) by balancing the weight of the gyro and accelerometer in the complementary filter. Other devices would have a different time constant depending on the time interval and on the characteristics of the installed gyro and accelerometer.

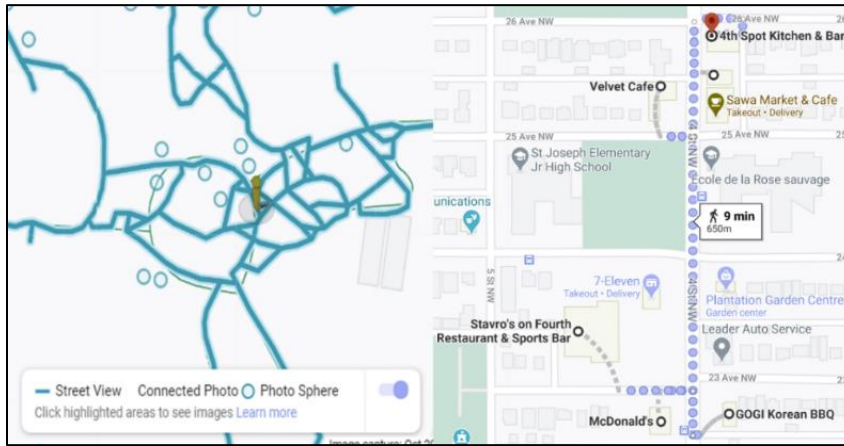
### **Spatial Cognition**

The performance of the spatial cognition algorithm was also good. There is no noticeable delay in generating the four tasks in 800 CPU ticks or less than  $0.08 \pm 0.02\text{ms}$ . The user can answer them relatively quickly, and then the interface is adapted to provide the right amount of information based on the user's SCORE. If the user wants more or less information, then they can manually change the SCORE, and it quickly switches because all the variations of the path descriptions are created at the same time. The adaptive PCD is also more descriptive, precise, and accurate than the Google Maps path description. Google has a photosphere and street map for people walking (Figure 45). The photosphere map suggests continuity in a network of lines, but it is just a collection of  $360^\circ$  photos at viewpoints. The street map suggests useful directions for pedestrians, but it uses the street network to route pedestrians down street centerlines and alleys instead of sidewalks. The Google path descriptions are also only as detailed as the street network using roads and street addresses to provide directions (Table D1-D5). AEGN uses a realistic walkable surface to model a walking path and the description of pedestrian landmarks such as crosswalks, staircases, and doors. As a result, the Google distances and times are accurate only within minutes and tens

of meters, whereas AEGN is accurate within seconds and meters. For example, the trip plan with Google Maps estimated 477s over 673m, but AEGN projected the trip to be 781s over 652.02m compared to the actual 782s over 652.46m (Table 4). The Google Maps summary distance and time rounded the estimate to 540s over 650m. The Google Maps descriptions also had a bent arrow indicating a turn but not indicating which direction.

**Figure 45**

*Google Pedestrian Photosphere (Google, n.d. a) and Maps (Google, n.d. b)*



AR is also used to help describe the pedestrian path, and it is used to initialize the Spatial Cognition tasks. There is some delay in switching between the ART Pose and Navigate scenes which could be removed if the scenes were integrated. As well, once a user switches from the ART Pose scene to the Navigate scene, the drift error begins to accumulate, so it is important to calibrate the application. The ART Pose errors are measured from surveyed reference positions pointed to the center of the ART target. The ART Poses provide more accurate positions and directions ( $0.25 \pm 0.06\text{m}$  and  $3 \pm 2^\circ$ ) than INS ( $1.68 \pm 0.76\text{m}$  and  $8 \pm 3^\circ$ ) (Table 9), so if a correction is required, then they should be used. There is a limit to the distance and angle that ART Pose can be used because as they increase, the FOV percentage decreases quickly.

**Table 9***ART Pose and INS Errors at Finish Control Points*

Finish target	ART pose				INS	
	Rating	FOV (%)	Error (m)	Error (°)	Error (m)	Error (°)
McDonalds	4	14	0.20	1	0.43	5
Gogi	2	7	0.21	3	2.06	8
Stavros	5	6	0.26	1	1.38	8
Velvet	4	27	0.29	1	2.36	4
Johns	4	37	0.19	7	1.43	7
4th Spot	5	31	0.43	4	2.44	13
MAE			0.25 $\pm 0.06$	3 $\pm 2$	1.68 $\pm 0.76$	8 $\pm 3$

ART Pose is accurate and precise because it is calculated directly from surveyed reference positions. INS calculations are relative to the last position and always accumulate drift errors. ART Pose, on the other hand, is limited by its viewing range and angle. While there can be some obstructions and it can be viewed from an angle, the optimal position and orientation to view 2D AR is from a distance roughly the same as the dimensions of the AR target and between 30° either side of center. This is not only for calculating the user pose but to view the AR media. If it is a 3D model, then it is useful to move in and out and side to side to view details from different perspectives, but in general the best view is when the AR Target fills the camera FOV. The best perspective for a large or small target would be farther or closer, respectively. The extreme limits set in the application for the near and far clipping planes are 0.01m and 20m, respectively, but 1m to 10m are the practical limits for ART ranging from signs to building facades. The size of the AR target should roughly correspond with the distance to the expected viewing perspective. For example, a sign 1m wide would be viewed well from 1m away or a 10m wide building entrance from 10m away. A 1m target could be detected 10m away, but a 1 to 10 ratio between size and distance is a good estimation of the distance limit. If there is difficulty in the AR target being detected, then moving in towards the front and filling the camera FOV will help. Once it is

detected, moving back to the desired location will usually keep tracking the AR target.

The Vuforia AR quality rating is a good measure of how well it will detect and track a target. A lower quality target will not be detected as well when viewed from a distance and angle. The ratings are automatically processed when images are uploaded to create a target based on the image detail, contrast, and patterns. Other details to consider when picking targets are permanence, lighting, visibility, flatness, and finish. Permanence, visibility, and lighting are good reasons to pick public art as targets because those are usually addressed when the art is installed. If these issues were considered, it would reduce the need for additional updates and illumination. The AR settings allow the flash to be turned on, which can be useful for small targets in close range in dimly lit conditions but try to avoid selecting poorly illuminated targets. Too much lighting from natural or artificial sources can compromise the target visibility with glare reflecting off glass or other shiny surfaces. Suppose the lighting changes the target depending on the time of the day or season. In that case, multiple images of the same target can be used for each scenario. Flat 2D targets work better than 3D targets for tracking and reduce the time to model in the application. The Vuforia target database can be dynamically loaded for different environments, such as lighting conditions. The workflow to create the AR learning environment with six targets is less than 4hr to perform the following tasks:

1. Pick targets as landmarks following the previous recommendations.
2. Add targets to blocks without any existing targets.
3. Take photos of the targets (preferably flat) illuminated, without obstructions or glare.
4. Measure the bottom edge of the target (cm), as well as the location and direction.
5. Clip and transform photos to make them have square (90°) corners. GIMP has a cage transform tool that works well to fix skewed images.

6. Upload the photos to <https://developer.vuforia.com/target-manager> (format: 24-bit < 2Mb  
RGB name = “target(integer)\_width(cm)\_rating(integer).jpg” e.g., 1\_100\_4.jpg for target=1, width = 100cm, and Vuforia rating = 4) and export the target database to a unitypackage.
7. Import the package into the Unity project with the building model.
8. Create sprites and AR targets in the ART Pose and Navigate scenes at control points.
9. Save the scenes, save the project, and rebuild the install.

### Navigation

The performance of the navigation algorithm was acceptable. There is no noticeable delay in generating a quaternion or rotation in  $9982 \pm 1183$  CPU ticks,  $0.99 \pm 0.12$ ms, or  $1002 \pm 119$  quaternions/s. The complementary filter algorithm can relatively quickly calculate the rotation of the device. Compared to five other attitude estimation algorithms, it was the slowest at 25% relative to the best (Table 10). Still, it is acceptable considering the application over-head from also rendering a real-time navigation application compared to the other processing times resulting from applications just logging the algorithm calculations.

**Table 10**

*Processing Time on Quaternion Generation (quaternion/s) (Michel, 2015, p.10)*

	Mahony	Madgwick	Fourati	Choukroun	Martin
Quaternions/s	2762	4052	2559	2148	1257
Relative to the best	0.68	1.00	0.63	0.53	0.31

Madgwick (2011) and Mahony (2008) also use quaternions to represent rotation in their complementary filters (Zhi, 2016). Only the Renaudin and AEGN algorithms correct bias and noise for the accelerometer and gyro (Table 11).

**Table 11***Sensors' Biases and Noises Consideration in Each Algorithm (Michel, 2015, p.7)*

	Gyroscope		Accelerometer		Magnetometer	
	Bias	Noise	Bias	Noise	Bias	Noise
Mahony	X					
Madgwick	X					
Fourati						
Martin	X		X		X	
Choukroun	X	X		X		X
Renaudin	X	X	X	X		X

The ground truth reference Unity agent is used to calculate the fastest path and error distance. The error distance is automatically logged while navigating when starting with ART Pose, recording GNSS, correcting INS error greater than 3m, timing 20s intervals and reaching the finish. Table 12 shows the mean results for the INS are better than both the gyro and GNSS. It also duplicates the dynamic calibration results averaging 193s before exceeding the 3m error budget in real conditions, walking through a neighborhood holding a smartphone. While the INS sample rate (25Hz) was slow compared to other INS implementations, the intermittent GNSS signals were exceedingly sparse, with only 30 readings in 37min over 1850m.

**Table 12***INS, Gyro and GNSS Errors on Paths With a Travel Time Greater Than 180s*

Path	FROM_TARGET	TO_TARGET	INS Mean error (m)	Gyro Mean error (m)	GNSS Mean error (m)	Distance <3m error (m)	Time <3m error (s)
3_TO_0	Velvet	McDonalds	1.48	5.80	1.50	239.96	288
0_TO_2	McDonalds	Johns	1.34	25.53	3.39	102.68	122
2_TO_5	Johns	Gogi	1.62	138.85	4.71	290.50	348
5_TO_1	Gogi	4 <sup>th</sup> Spot	1.66	3.31	2.98	105.45	126
1_TO_4	4 <sup>th</sup> Spot	Stavros	1.84	3.22	2.01	96.39	115
4_TO_3	Stavros	Velvet	1.23	32.55	3.81	129.81	156
Overall	Start (Velvet)	Finish (Velvet)	1.55 ±0.92	30.56 ±60.15	3.12 ±1.93	160.80 ±83.23	193 ±99

While the gyro bias was used to grade the system as a low-end industrial grade INS the error calibration and compensation resulted in navigation grade performance with a drift error less

than 3m over 180s (Table 13).

**Table 13**

*Error Over Time by Sensor Grade*

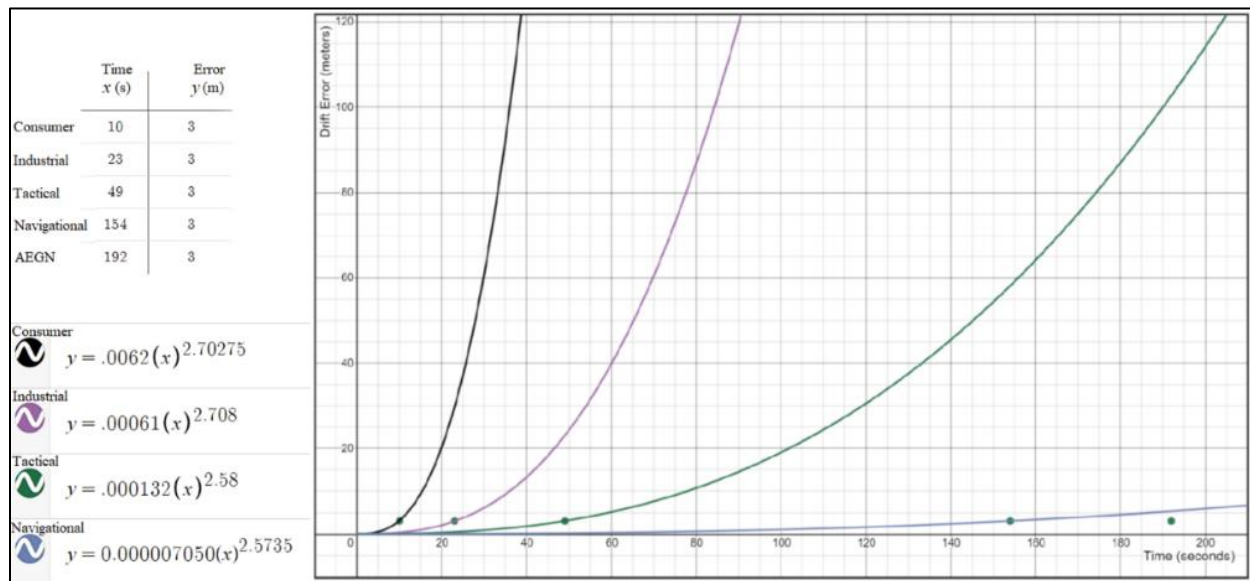
Grade	Error (m) over time					
	1s <sup>a</sup>	10s <sup>a</sup>	60s <sup>a</sup>	180s <sup>b</sup>	600s <sup>a</sup>	3600s <sup>a</sup>
Consumer	0.060	6.500	400.0	8000	200000	39000000
Industrial	0.006	0.700	40.0	800	20000	3900000
Tactical	0.001	0.080	5.0	<90	2000	400000
Navigation	<0.001	0.001	0.5	<5	100	10000

Note: <sup>a</sup>(VectorNav, 2020). <sup>b</sup>Estimated from best fit line in Figure 46.

Figure 46 visually depicts the difference in the drift errors of INS sensor grades. Consumer and navigation grade INS would be expected to exceed the error budget at 10s and 154s, respectively. AEGN INS lasted 193s before drifting 3m.

**Figure 46**

*Best Fit Lines Graphing Error Over Time by Sensor Grade (Table 13)*



AEGN also compared well to other non-INS indoor navigation solutions. A competition was held at the Indoor Positioning and Indoor Navigation (IPIN) 2018 conference to evaluate different solutions to indoor positioning in a real environment. They had two general tracks for

simulated and real tests for a shopping mall environment consisting of six or eight 3 to 8min routes with up to 17 reference points over 1000m (Appendix E). This was similar to the AEGN business strip street test of six 5 to 7min routes with up to 36 reference points over 1850m. The reference points were different for IPIN, using surveyed floor markers. In contrast, AEGN only used surveyed ground points at the start and finish and a moving virtual reference Unity agent along the route. IPIN used the 3rd Quartile metric to determine performance rank of the diverse competitors. The 3rd Quartile or 75th Percentile measured the distance error (meters) from the reference points for 75% of samples with a lower error. The shopping mall had a few floors, and this was also factored into the ranking with a penalty for not calculating the correct floor. While the AEGN business strip did not have multiple levels, the AU Campus tests had a few floors like the shopping mall, and the 3D model of the walkable surface was always in contact with the tracking Unity agent, so the level was always correct. The competing solutions used a variety of data (Wi-Fi, INS, GNSS etc.), methods and equipment (smartphones or laptops). While some competitors used signal sensors, they were not allowed to install additional beacons, or markers. Several competitors used different techniques at map matching or approximating the best real location of the user if the calculated location was not possible (e.g., in a wall). AEGN Auto INS is an extended version of map matching by determining the best-known position and direction when the navigation is determined to be incorrect (i.e., drift error > 3m). The difference in the simulated tests' lower drift errors compared to the real tests can probably be attributed to the real tests being performed in real-time and then submitted immediately after finishing the test, whereas the simulated test teams had three months to process, optimize and reprocess their results. AEGN's real test 75th percentile result, 1.14m, compares well even against the simulated test results with only one better IPIN result, 0.90m, for the Wuhan University team (WHU) (Table L1). While AEGN uses a smartphone

accelerometer, gyro, and camera with map matching the WHU team utilized a smartphone wi-fi, Geomagnetic, Magnetometer and Barometric sensors in addition to an accelerometer and gyro.

### **Findings and Discussion**

This section reports the findings from the data analysis and discusses the significance of those findings. AEGN is an INS using AR for corrections and spatial cognition learning tasks to improve the device sensor readings and the user's sense of direction. The system is designed for indoor navigation where GNSS signals are unavailable. The INS was also chosen for emergency situations when a power outage might also make wireless signals unavailable. INS was also needed to continuously track the user's direction for spatial cognition learning tasks. Neither GNSS nor wireless positioning can track directions as well as INS. INS calibration was added to make the low-cost INS readings accurate and precise enough to track a user walking 180s within a 3m error. When tracking exceeds the 3m error, AR is used to correct the position and direction. AR is also used to test and train the user with spatial cognition learning tasks. The spatial cognition learning tasks also need ordered combinations of task paths created by path generation and assigned with task allocation. As such, the key components of AEGN are path generation, task allocation, calibration, spatial cognition, and navigation. These findings are discussed in subsections for each of these components.

#### **Path Generation**

The results of the path generation test support the H1 hypothesis that it finds the optimal path based on the motion mode online without pause in the GUI. The automated and manual tests covered many areas and scenarios for both walking and rolling modes. The paths were generated online in approximately  $497 \pm 142$  CPU ticks which caused no noticeable pause in the GUI. There was no pause when passing another control point, even when recalculating the path. Recalculating

paths in more complicated environments is required for longer routes. For example, the path will not be solved when running 200m across a maze of office cubicles, but it will if it recalculates halfway. Many safety, privacy and security scenarios work avoiding hazards and inaccessible areas such as elevators in emergencies, washrooms, and server rooms without authorization. The navigation also personalizes routes based on user height and speed, avoiding low ceilings for tall users. Faster speeds (e.g., 4.0m/s) are not used to decrease the time required for tracing because wider turns change the estimated length and time of path. However, using the fastest walking speed (e.g., 1.67m/s) only contributes a MAE of  $0.06 \pm 0.04\text{m}$  or  $0.07 \pm 0.05\text{s}$  to each route which is minimal because the turning radius is very similar to the slowest walking speeds (e.g., 0.83m/s). This compares very well to the estimated paths of Google Maps, which have a MAE around a thousand times greater for distance ( $58 \pm 35\text{m}$ ) or time ( $64 \pm 34\text{s}$ ). The accurate and precise distance table is stored for use by the task allocation and route icons for the spatial cognition learning tasks.

### **Task Allocation**

The results of the task allocation test support the H2 hypothesis that it determines the optimal ordered combination of user tasks based on group user preferences and time in less than 1min. The performance of the task allocation algorithm was  $11.2 \pm 8.7\text{s}$ ,  $14.4 \pm 2.7\text{s}$ , and  $20.2 \pm 2.8\text{s}$  for various permutations, ordered combinations, of 6 tasks for 1, 2 and 3 person groups, respectively. The Task Allocation uses the distances from the path generation, so they have the same distance and time MAE. The algorithm uses brute force by comparing the combined time for every group permutation to guarantee optimal allocation. The number of permutations is reduced by first allocating the start and user preference tasks and then processing the remaining permutations with brute force based on the users' motion mode and speed to minimize group travel time. This reduces processing time and rewards the users for group participation. If the users select

familiar tasks and locations, then that can add to the overall efficiency in the task and travel time. If the remaining permutations exceed  $10!$  then the optimal brute force approach becomes intractable, and a suboptimal heuristic approximation is required. The user preference can also be used as an intuitive seed for a heuristic approximation. If the allocation is not equitably distributed, then the CTM will calculate a high preference time cost. An administrator can then investigate the task allocation to determine if any system or user issues need to be reconfigured to minimize the social costs. The task allocation is currently only configured to process one site and group per user at a time, but the algorithm allows multiple groups to be processed simultaneously. The AEGN Coordinator uses AMUSE to help users communicate when setting up self-organized groups, schedules, and preferences. Automated Jason agents representing user preferences handle the complex CNP messaging on the server. AEGN is more effective than Google Maps by providing an ordered task assignment to minimize travel time. Google Maps also does not consider multiple users, wheelchair access or different speeds.

### **Calibration**

The calibration test results support the H3 hypothesis that it sets the INS direction of movement along a route when the clutch with calibration is active for 180s online without pause in the GUI. The INS is characterized by a low-end industrial grade gyro  $BI=12\pm0.84^\circ/\text{hr}$  determined by an Allan deviation analysis of long-term sensor readings. The analysis also determines the time clusters required to minimize the effect of stochastic noise on calibration and navigation. The simple but effective six step static calibration is run with those time clusters without specialized equipment to minimize drift. The calibration compensates for accelerometer bias and alignment (scale-factor and cross-coupling), gyro bias and g-dependent deterministic errors. The navigation time cluster minimizes quantization, ARW and VRW stochastic noise.

Calibration does not account for changes in temperature, so the device needs to be warmed up at the temperature of the navigation environment before calibrating. Static tests with various time constants were used to determine the time constant (e.g.,  $\tau = 0.7$ ) to tune the complementary filter coefficient. The static calibration test can run over 180s with a drift error less than 3m ( $219 \pm 27$ s). This INS performance is short-term navigation or tactical grade. The calibration does not account for dynamic deterministic errors, so it did not perform as well, as expected, in dynamic conditions where it is subjected to errors added by rotation and acceleration ( $193 \pm 16$ 1s).

### **Spatial Cognition**

The results of the spatial cognition test support the H4 hypothesis that it is initialized with the ART pose and starts the optimal viewing properties of generated paths based on the user's SCORE online without pause in the GUI. The ART pose is accurate and precise enough to track the user's pose to initialize the tasks, answer key, and the user's answers. The MRT task buttons use sprites from the path generation. ART pose initializes the INS and start target, and the user selects the finish target, which creates the tasks and answer key. The four spatial cognition learning tasks can be quickly answered because they consist of only two multiple-choice, four numeric text inputs, and five dropdown fields. The questions can be answered and quickly SCOREd to set the optimal viewing properties for the user to begin navigating before the INS drifts. The SCORE sets the optimal viewing properties for the PCD and perspective. The PCD is adaptive and more descriptive than the Google path descriptions. For example, the PCD describes the walking routes along sidewalks rather than streets center lines. Just as the control points are where the path generation recalculates the path directions for the Unity agent, the control points serve as useful locations for the users to get their bearings. The spatial cognition learning tasks at the control points improve the user's sense of direction, augmenting intelligence.

## Navigation

The results of the navigation test support the H5 hypothesis that it is corrected with the ART pose and restarts navigation, after 180s the INS position is within  $\pm 3\text{m}$  of the actual position, it is more accurate than the gyro and GNSS positions, but less than the corrected INS poses, and the workflow to create the AR learning environment is less than 4hr. The ART pose is more accurate and precise (MAE:  $0.25 \pm 0.06\text{m}$  and  $3 \pm 2^\circ$ ) than the INS at the finish targets (MAE:  $1.68 \pm 0.76\text{m}$  and  $8 \pm 3^\circ$ ), so it is used to initialize the users pose. The ART pose accuracy and precision can be improved by choosing high quality AR targets that are good 24hr and all seasons. Indoor AR targets are easier to find with artificial light and no weather effects. Still, quality outdoor AR targets can also be found, such as internally illuminated signs in sheltered locations like building alcoves. Switching from the ART Pose scene to the Navigate scene needs to be refactored because it takes about 12s when it only takes about 4s to switch from Navigate to ART Pose. Most of the extra time is probably related to all the logging for test purposes and some for INS initialization. For example, automatically logging snapshots has already been disabled because they caused the GUI to pause. The time to reach the 3m error budget for the real live test ( $193 \pm 99\text{s}$ ) starting at one target and walking to another, stopping, and starting again at obstacles like crosswalks for cars with additional dynamic errors from acceleration and rotation while moving duplicated the result of the dynamic calibration tests ( $193 \pm 161\text{s}$ ). The drift MAE over 180s durations was  $1.55 \pm 0.92\text{m}$  while covering an average distance of  $160.80 \pm 83.23\text{m}$ . This performance compares well to the simultaneous drift MAE of the gyro ( $30.56 \pm 60.15\text{m}$ ) and GNSS ( $3.12 \pm 1.93\text{m}$ ).

The INS also compared very well to other navigation methods. Using similar testing methods to the IPIN 2018 competition, the INS 75TH Percentile error ( $1.14 \pm 0.65\text{m}$ ) is second only

to the best IPIN result. In particular, the design constraining the vertical to the 3D model NavMesh surface reduced the complexity of the INS to the horizontal components while guaranteeing the correct vertical position. The clutch enables the INS horizontal velocity threshold trigger to start, speed up, slow down and stop. The INS velocity is very sensitive, so it is necessary to calibrate the INS, tune the complementary filter, and move the device smoothly to avoid unnecessary jolts which might be falsely detected as a start or stop. For example, the INS does not track lateral, backward, or vertical velocity, and to eliminate those movements unintentionally, the device was mounted on a bicycle handlebar. Still, the vibrations caused too many fluctuations in speed. Walking at a slow speed (0.83m/s) helps make smooth operations easier. The clutch also starts and stops the virtual reference Unity agent keeping it synchronized with the user's real position. The dynamic virtual reference Unity agent saves a lot of setup time, reducing the outdoor survey points to the start and finish targets. The targets were surveyed with survey tape measurements and an infrared scanner to spot check discrepancies and align drawings with the aerial land base. The reference Unity agent is easiest to see by zooming out with the TOP FORWARD perspective, which shows the position relative to the surrounding aerial land base and the direction by rotating the whole scene. The image stabilization from the complementary filter allows the user to hold the device at any angle. This is more comfortable for holding and viewing to avoid glare to see the reference Unity agent. One problem with the reference Unity agent is where Off-Mesh links are used. The speed of the Unity agent is not controlled and will jump ahead, unnecessarily increasing the drift error.

The best navigation strategy uses ART pose and spatial cognition learning tasks to correct the initial INS sensor readings and the user's sense of direction. Then navigate through the most convoluted path sections first to get on a main path or corridor before the INS drifts much. Once

the user is on a main path or corridor, they may not even need to refer to the navigation device for help. This is a common navigation pattern where a user starts in office cubicles or store aisles, then navigates out to a corridor and into another set of cubicles or aisles. Egress and ingress of the cubicles and aisles are where navigation is most challenging. If the MAE is nearing 3m, then the user can choose the following map matching (constrained to the NavMesh) correction strategies:

- ART Pose -- when: intermittently at the start, restart, and finish; where: control point; why: relative to the surveyed reference point.
- Auto Pose – when: continuously following virtual reference Unity agent; where: along a route; why: fly through with first person perspective like a personal tour guide.
- Auto INS – when: intermittently following virtual reference Unity agent; where: along a route; why: track user pose but correct when 3m error budget exceeded.
- Manual Pose – when: intermittently while not following a route; where: user picks any destination point on a navigable surface; why: navigate ad hoc path.

A user can also counter-turn to the drift if they know where they are going until the destination is reached or the pose is corrected as described above. The sensors are still responsive but just out of alignment with the device orientation. This is like adjusting a compass for variations between the geographic and magnetic north poles because the magnetic north pole also drifts over time.

## Chapter VI – Conclusions

### Possible Contributions

The results support the first objective of enabling navigating indoors using a hand-held android device. The implementation enables the fastest path, object shaping, obstacle avoidance, visibility, and drift with a realistic NavMesh, optimal task allocation, adaptive interface, INS calibration, and ART pose. The results also support the second objective to identify map discrepancies using a Project Tango Tablet. The implementation manages initialization, processing, gaps, and symbology with localized scans, standardized symbology, and ART targets. The results also support the third objective to personalize the guided tour by adapting the model for each user. The implementation enables navigation, safety, security, and learning task messages in a personalized space with PCD, muster routes, communicator app, spatial notes, coordinator app and an adaptable perspective. The implementation technically enables the UFOV, MRT and MOL tasks. However, user test results are still required to prove that user spatial cognition would improve and make navigating faster.

Navigation indoors with a hand-held android device can be used in mines, factories, warehouses, malls, universities, hospitals during normal operations or in an emergency, for all ages and abilities. The augmented intelligence can be used for more than improving the user's spatial cognition. Some other examples include the following (IEEE, 2021):

- Virtual personal tour guide for distance education of art galleries, museums, and historical sites.
- Augmented reality tasks using multi-media to overlay on AR targets for educational learning, vocational training, or operational instructions.

For users, leaders and exhibitors, the interface can be used as a pedagogical tool for spatial

cognition and location-based learning tasks. The MOL task simultaneously improves memory of spatial cognition and location-based tasks. The benefits of preparing the system for an exhibit could be reused if the exhibitors supplied the 4D model to leaders, who could then tailor the model to fit the needs of an educational program. The fiducial markers could also be used more if they were included in the exhibit signage with human interpretable symbology to convey meaning to visitors without the system. The system is also beneficial to the user because it adapts to the user's needs to minimize distraction from their learning tasks. Reducing user distractions serves the following two purposes:

- 1) Exhibit displays are often laid out logically, such as chronological or geographical. If the user pays attention to the order of the displays, then they gain additional contextual knowledge of the subject they are learning.

- 2) If they look where they are going, they will build up a better cognitive map of the space. This will improve the user's ANAV and ENAV (Koutakis et al., 2013).

In addition, the safety and security features should give the users and leaders the confidence to let the users explore the learning area. The result is intended to help develop the user's cognitive map and sense of direction, so they spend less time travelling and more time learning or working.

For application developers, the application demonstrates how to develop an indoor group navigation system that could be translated into other potential applications such as tourism and emergency services. The cognitive 4D model using the universal IOF symbology also provides a useful example of how to abstract spatial features succinctly. The server coordinates the users by updating learning and guidance messages if the schedule or tasks are changed, allowing them to self-organize in groups. The schedule can be reused in many business applications where it is important to deploy resources effectively.

For academic researchers, this research presents a model for examining and developing AI in a fashion that improves human intelligence, augmented intelligence. This is not isolated to just the research fields related to the application development but also location-based field research that can benefit from using the application. The application involves interdisciplinary research in mathematics, computer, and cognitive sciences. These fields can add to deep and machine learning to analyze spatial data and cognition, respectively, to further enhance the application benefits. There are many location-based field research usages for architecture, art, history, geography, engineering, education, economics, commerce, and tourism. They can use the virtual navigation assistant to provide the directions and data they need when they need it. In general, augmented intelligence combines artificial and human intelligence, to produce better research results than they can separately (IEEE, 2021).

More specifically, for both developers and researchers, AEGN provides the following benefits:

- A framework for testing DMAS and mTSP concepts with real world models.
- A method to maximize low-cost sensor effectiveness with calibration.
- Simultaneous simulated and real testing of multiple navigation system components.
- An example of standardized test methodology to compare with other navigation solutions.
- A virtual reference Unity agent to save time setting up navigation application field tests and provides a continuous reference point to calculate MAE.
- An automated method to anonymously differentiate test and control users' functionality and data using the android id.
- Spatial cognition learning tasks enable clinical testing in real-world situations for everyday training, as a therapy, to improve people's sense of direction.

### Summary and Future Works

The main goal of this research is to determine if an INS can be corrected to effectively track a user's pose, position and orientation, by using a monocular camera with fiducial markers. In addition, this research determines if an AR system interface that integrates spatial cognition learning tasks can test and train the user's spatial cognition. The implications of positive results are that the corrected INS poses on ubiquitous mobile devices provide reasonably accurate, precise, and reliable indoor navigation when a GNSS is unavailable. Positive results indicate that indoor navigation is possible and can increase widespread adoption if planar art is used as fiducial markers to support corrections of cumulative INS errors. This low-cost design could lead to a commercially viable solution that would be especially competitive during an economic recession when it is difficult to rationalize expensive hardware and software. This research investigated LBS for use in learning environments, tourism, and emergency services. In conclusion, LBS using corrected INS with integrated spatial cognition learning tasks may increase not only the navigational performance but also the robustness of the system by enabling the users to function better spatially regardless of what technical spatial tools are available.

Suggested areas for future works include the following:

- Test the system for use as an occupational therapy aid for people suffering injuries or diseases affecting their spatial awareness, such as topographical disorientation disorder (Lim et al., 2010).
- Use deep learning to analyze user patterns to see if it is possible to predict user preferences.
- Implement a suboptimal heuristic machine learning algorithm for mTSP using the preference as a seed to replace the brute force approach for a larger number of destinations.
- Add support for dynamic obstructions (Canadian Traveler Problem) to force rerouting in

runtime.

- Add support for user skills, equipment, and availability to determine task allocation.
- The Communicator and Coordinator are configured to support Android's built-in text-to-speech and speech recognition. Add speech and voice support to the Navigator.
- Explore non-deterministic Turing machine or quantum computing to process simultaneous mTSP permutations.
- Model user preferences as Jason agent beliefs, desires, and intentions.
- Add the time to complete a task to the mTSP, a variation of the Travelling Purchaser Problem.
- Store calibration for various temperatures. Use a weather website for temperature if a temperature sensor is unavailable.
- Tile multiple sites overlaid on Open Street Map landbase.
- Increase the gyro and accelerometer sample rates to improve performance.
- Merge the ART pose and Navigate scenes to avoid delays in scene switching.
- Reenable Skyhook positioning.
- Add Wi-Fi positioning to create a hybrid with the INS.
- Crowdsourced: Wi-Fi heat map, Pedestrian traffic heat map, Magnetic heat map, Floor plans and discrepancies.
- Modify the INS to account for sideways and backward motion.
- Modify the testing to read the recorded IPIN 2018 dataset for simulations and comparisons.
- Improve velocity approximation to minimize false starts and stops.
- Add speed control to Off-Mesh links for virtual reference Unity agent.

## References

- Barry, C., Lever, C., Hayman, R., Hartley, T., Burton, S., O'Keefe, J., Jeffery, K., & Burgess, N. (2006). The boundary vector cell model of place cell firing and spatial memory. *Reviews in the Neurosciences*, 17(1–2), 71–97. <https://doi.org/10.1515/REVNEURO.2006.17.1-2.71>
- Baudrillard, J. (1994). *Simulacra and simulations*. University of Michigan Press. <https://doi.org/10.3998/mpub.9904>
- Bergenti, F. & Caire, G. (2013). Exploiting interactive workflows on android devices. <https://jade.tilab.com/wade/doc/tutorial/ExploitingInteractiveWorkflowsOnAndroidDevices.pdf>
- Bernasco, W. (2019). Adolescent offenders' current whereabouts predict locations of their future crimes. *PLOS ONE*, 14(1), 1–17. <https://doi.org/10.1371/journal.pone.0210733>
- Bolton, M. (2007, October). McLuhan for testers. *Better Software*, 14–15. <https://www.developsense.com/articles/2007-09-McLuhanForTesters.pdf>
- Bower, G. H. (1970). Analysis of a mnemonic device: Modern psychology uncovers the powerful components of an ancient system for improving memory. *American Scientist*, 58(5), 496–510. <http://www.jstor.org/stable/27829239>
- Casner, S., Geven, R., Recker, M., & Schooler, J. (2014). The retention of manual flying skills in the automated cockpit. *Human Factors*, 56(8), 1506–1516. <https://doi.org/10.1177/0018720814535628>
- Cicero, M. (1942). *On the orator: books 1-2*. (E. Sutton, Trans.). Harvard University Press. (Original work published ca. 55 BC) <https://www.loebclassics.com/view/LCL348/1942/volume.xml>
- Clynes, M., & Kline, N. (1960, September). Cyborgs and space. *Astronautics*, 26-27, 74-76. <http://web.mit.edu/digitalapollo/Documents/Chapter1/cyborgs.pdf>
- Dam, H. K., & Winikoff, M. (2013). Towards a next-generation AOSE methodology. *Science of Computer Programming*, 78(6), 684–694. <https://doi.org/10.1016/j.scico.2011.12.005>
- Darwin, C. (1873). Origin of certain instincts. *Nature*, 7(179), 417–418. <https://doi.org/10.1038/007417a0>
- Davis, D. L. (2009). *Commercial navigation in the greek and roman world* [Doctoral dissertation, University of Texas]. <http://hdl.handle.net/2152/18420>

- Ephrati, E., & Rosenschein, J. S. (1991). The Clarke tax as a consensus mechanism among automated agents. *Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI-91)*, July 14-19, 1991, Anaheim, California, USA, 173–178.  
<https://www.aaai.org/Papers/AAAI/1991/AAAI91-028.pdf>
- Etienne, A. S., & Jeffery, K. J. (2004). Path integration in mammals. *Hippocampus*, 14(2), 180–192. <https://doi.org/10.1002/hipo.10173>
- FAA. (2013a). *Safety alert for operators - Subject: Manual flight operations purpose: This SAFO encourages operators to promote manual flight operations when appropriate.*  
[http://www.faa.gov/other\\_visit/aviation\\_industry/airline\\_operators/airline\\_safety/safo/all\\_safos/media/2013/SAFO13002.pdf](http://www.faa.gov/other_visit/aviation_industry/airline_operators/airline_safety/safo/all_safos/media/2013/SAFO13002.pdf)
- FAA. (2013b). *Final report of the performance-based operations aviation rulemaking committee/commercial aviation safety team flight deck automation working group.*  
<https://skybrary.aero/sites/default/files/bookshelf/2501.pdf>
- Feng, J., Spence, I., & Pratt, J. (2007). Playing an action video game reduces gender differences in spatial cognition. *Psychological Science*, 18(10), 850–855.  
<https://doi.org/10.1111/j.1467-9280.2007.01990.x>
- Fleming, P. (2005). *Implementing a robust 3-dimensional egocentric navigation system* [Master's Thesis, Vanderbilt University]. <http://hdl.handle.net/1803/13632>
- Google. (n.d. a). [La Brea Tar Pits and Museum - Google Maps]. Retrieved April 16, 2021, from [https://www.google.ca/maps/@34.0632334,-118.3553216,3a,75y,350.07h,90t/data=!3m1!1e1!3m8!1st0Jb\\_IDTf5vnjIDrB6KeaA!2e0!6shhttps:%2F%2Fstreetviewpixels-pa.googleapis.com%2Fv1%2Fthumbnail%3Fpanoid%3Dt0Jb\\_IDTf5vnjIDrB6KeaA%26cb\\_client%3Dmaps\\_sv.tactile.gps%26w%3D203%26h%3D100%26yaw%3D306.49576%26pitch%3D0%26thumbfov%3D100!7i13312!8i6656!9m2!1b1!2i52](https://www.google.ca/maps/@34.0632334,-118.3553216,3a,75y,350.07h,90t/data=!3m1!1e1!3m8!1st0Jb_IDTf5vnjIDrB6KeaA!2e0!6shhttps:%2F%2Fstreetviewpixels-pa.googleapis.com%2Fv1%2Fthumbnail%3Fpanoid%3Dt0Jb_IDTf5vnjIDrB6KeaA%26cb_client%3Dmaps_sv.tactile.gps%26w%3D203%26h%3D100%26yaw%3D306.49576%26pitch%3D0%26thumbfov%3D100!7i13312!8i6656!9m2!1b1!2i52)
- Google. (n.d. b). [McDonald's to 4th Spot Kitchen & Bar - Google Maps]. Retrieved April 16, 2021, from <https://www.google.ca/maps/dir/McDonald's,+507+23+Ave+NW,+Calgary,+AB+T2M+1S7/GOGI+Korean+BBQ,+4+Street+Northwest,+Calgary,+AB/Stavro's+on+Fourth+Restaurant+%26+Sports+Bar,+4+Street+Northwest,+Calgary,+AB/Velvet+Cafe,+25+Avenue+Northwest,+Calgary,+AB/John's+Breakfast+%26+Lunch,+4+Street+Northwest,+Calgary,+AB/4th+Spot+Kitchen+%26+Bar,+4+Street+Northwest,+Calgary,+AB>
- Google. (2015). Project Tango. <https://developers.google.com/project-tango/>
- Google. (2016). Tango Constructor (Version 1). Google.  
<https://developers.google.com/tango/tools/constructor>

- Gurau, C., & Nuchter, A. (2013). Challenges in using semantic knowledge for 3d object classification. *Proceedings of the KI 2013 Workshop on Visual and Spatial Cognition (KI 2013)*, Koblenz, Germany, September 17, 2013, 29-35. <https://robotik.informatik.uni-wuerzburg.de/telematics/download/ki2013.pdf>
- Hafting, T., Fyhn, M., Molden, S., Moser, M.-B., & Moser, E. I. (2005). Microstructure of a spatial map in the entorhinal cortex. *Nature*, 436(7052), 801–806. <https://doi.org/10.1038/nature03721>
- Hawking, S., Rusell, S., Tegmark, M., & Wilczek, F. (2014, May 1). *Stephen Hawking: 'Transcendence looks at the implications of artificial intelligence - but are we taking AI seriously enough?'*. Independent. <http://www.independent.co.uk/news/science/stephen-hawking-transcendence-looks-at-the-implications-of-artificial-intelligence--but-are-we-taking-ai-seriously-enough-9313474.html>
- Healy, A. (2015, March 30). *Police: GPS may have directed Indiana couple off bridge, man survives 38-foot plunge*. Syracuse. [http://www.syracuse.com/us-news/index.ssf/2015/03/police\\_gps\\_may\\_have\\_directed\\_indiana\\_couple\\_off\\_bridge\\_causin\\_g\\_fatal\\_crash.html](http://www.syracuse.com/us-news/index.ssf/2015/03/police_gps_may_have_directed_indiana_couple_off_bridge_causin_g_fatal_crash.html)
- Hongu, N., & Wise, J. (2009). Pedometer and new technology - Cell phone & Google Maps: What you need and want to know. <http://hdl.handle.net/10150/146658>
- Horner, H., Walker, L., Bentall, D., & Dünser, A. (2012). Learning complex three dimensional concepts using augmented reality. *Proceedings of Society for Information Technology & Teacher Education International Conference (SITE 2012)*, Austin, Texas, USA, March 5, 2012, 4522–4525. <https://www.learntechlib.org/p/40324>
- Hou, H. (2004). *Modeling inertial sensors errors using Allan variance* [Master's thesis, University of Calgary]. <http://dx.doi.org/10.11575/PRISM/21729>
- Huseth, S., Dewberry, B., & McCroskey, R. (2011). Pulsed-RF ultrawideband ranging for the GLANSER GPS-denied emergency responder navigation system. *Proceedings of the Institute of Navigation - International Technical Meeting (ITM 2011)*, San Diego, California, USA, January 24-26, 2011, 1(January), 389–396. [https://www.researchgate.net/profile/Brandon-Dewberry/publication/280721975\\_Pulsed-RF\\_ultrawideband\\_ranging\\_for\\_the\\_GLANSER\\_GPS-denied\\_emergency\\_responder\\_navigation\\_system/links/55c2d0e608aea2d9bdbff292/Pulsed-RF-ultrawideband-ranging-for-the-GLANSER-GPS-denied-emergency-responder-navigation-system.pdf](https://www.researchgate.net/profile/Brandon-Dewberry/publication/280721975_Pulsed-RF_ultrawideband_ranging_for_the_GLANSER_GPS-denied_emergency_responder_navigation_system/links/55c2d0e608aea2d9bdbff292/Pulsed-RF-ultrawideband-ranging-for-the-GLANSER-GPS-denied-emergency-responder-navigation-system.pdf)
- IEEE. (1998). Standard specification format guide and test procedure for single-axis interferometric fiber optic gyros. *IEEE Std 952-1997*, 1–84. <https://doi.org/10.1109/IEEESTD.1998.86153>

- IEEE. (2022). *What is augmented intelligence?*. <https://digitalreality.ieee.org/publications/what-is-augmented-intelligence>
- Galan, W. (2015). *Project Tango tablet teardown*. iFixit. <https://www.ifixit.com/Teardown/Project+Tango+Tablet+Teardown/28148>
- Invensense. (2013). MPU-9150 specification product 4.3 revision. <https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-9150-Datasheet.pdf>
- Invensense. (2016). MPU-9250 specification product 1.1 revision. <https://invensense.tdk.com/wp-content/uploads/2015/02/PS-MPU-9250A-01-v1.1.pdf>
- IOF. (2018). International specifications for control descriptions. <https://buffalo-orienteeering.org/wp-content/uploads/2020/03/control-description-updated-A4-2019.pdf>
- IPIN. (2018). IPIN competition 2018. [http://ipin2018.ifsttar.fr/fileadmin/contributeurs/IPIN2018/COMPETITION/IPIN2018\\_CallForCompetition\\_v2.1.pdf](http://ipin2018.ifsttar.fr/fileadmin/contributeurs/IPIN2018/COMPETITION/IPIN2018_CallForCompetition_v2.1.pdf)
- ISO/IEC. (2016). *ISO/IEC 18305:2016 Information technology — Real time locating systems — Test and evaluation of localization and tracking systems*. <https://www.iso.org/standard/62090.html>
- Jabr, F. (2011, December 8). *Cache cab: Taxi drivers' brains grow to navigate London's streets*. Scientific American. <http://www.scientificamerican.com/article/london-taxi-memory/>
- Kaim, K., & Lenar, M. (2008). Modelling agent behaviours in simulating transport corridors using Prometheus and Jason. *Proceedings of the 2008 Conference on New Trends in Multimedia and Network Information Systems*, Amsterdam, Netherlands, June 30, 2008, 182–192. <https://dl.acm.org/doi/10.5555/1565754.1565774>
- Kamphorst, B. (2012). The primacy of human autonomy: Understanding agent rights through the human rights framework. *Proceedings of the 1st Workshop on Rights and Duties of Autonomous Agents (RDA2-2012)*, Montpellier, France, August 28, 2012, 885, 19–24. <http://ceur-ws.org/Vol-885/paper3.pdf>
- Gurau, C., & Nuchter, A. (2013). Challenges in using semantic knowledge for 3d object classification. *Proceedings of the KI 2013 Workshop on Visual and Spatial Cognition (KI 2013)*, Koblenz, Germany, September 17, 2013, 1055, 29-35. <https://robotik.informatik.uni-wuerzburg.de/telematics/download/ki2013.pdf>
- Klippel, A., Freksa, C., & Winter, S. (2010). You-are-here maps in emergencies — The danger of getting lost. *Journal of Spatial Science*, 51(1), 117–131. <https://doi.org/10.1080/14498596.2006.9635068>
- Koenig, S. (2012). *Individualised virtual reality rehabilitation after brain injuries* [Doctoral dissertation, University of Canterbury]. <http://hdl.handle.net/10092/6870>

- Konnikova, M. (2014, September 4). *The hazards of going on autopilot*. The New Yorker. <http://www.newyorker.com/science/maria-konnikova/hazards-automation>
- Koppers, J. (2009). *Location based services* [Master's thesis, Radboud University]. <https://www.ru.nl/publish/pages/769526/joostkoppersscriptie.pdf>
- Koutakis, P., Mukherjee, M., Vallabhajosula, S., Blanke, D. J., & Stergiou, N. (2013). Path integration: Effect of curved path complexity and sensory system on blindfolded walking. *Gait & Posture*, 37(2), 154–158. <https://doi.org/10.1016/j.gaitpost.2012.06.027>
- Kyritsis, M., Gulliver, S. R., Feredoes, E., & Din, S. U. (2018). Human behaviour in the euclidean travelling salesperson problem: Computational modelling of heuristics and figural effects. *Cognitive Systems Research*, 52, 387–399. <https://doi.org/10.1016/j.cogsys.2018.07.027>
- Lackner, J. (1988). Some proprioceptive influences on the perceptual representation of body shape and orientation. *Brain*, 111(2), 281–297. <https://doi.org/10.1093/brain/111.2.281>
- Li R., Zhang B., Sundar S.S., & Duh H.B.L. (2013). Interacting with augmented reality: How does location-based ar enhance learning?. *Proceedings of the IFIP Conference on Human-Computer Interaction (INTERACT 2013)*, Cape Town, South Africa, September 2-6, 2013, 8118, 616–623. [https://doi.org/10.1007/978-3-642-40480-1\\_43](https://doi.org/10.1007/978-3-642-40480-1_43)
- Lim, T.-S., Iaria, G., & Moon, S. (2010). Topographical disorientation in mild cognitive impairment: A voxel-based morphometry study. *Journal of Clinical Neurology*, 6(4), 204–211. <https://doi.org/10.3988/jcn.2010.6.4.204>
- Lu, C., Chang, M., Kinshuk, Huang, E., & Chen, C. (2011). Usability of context-aware mobile educational game. *Knowledge Management and E-Learning*, 3(3), 448–477. <https://doi.org/10.34105/j.kmel.2011.03.031>
- Lu, C., Chang, M., Kinshuk, Huang, E., & Chen, C. (2014). Context-aware mobile role playing game for learning – A case of Canada and Taiwan. *Educational Technology & Society*, 17(2), 101–114. [http://dx.doi.org/10.1007/978-3-642-38291-8\\_8](http://dx.doi.org/10.1007/978-3-642-38291-8_8)
- Madgwick, S. O. H., Harrison, A. J. L., & Vaidyanathan, R. (2011). Estimation of IMU and MARG orientation using a gradient descent algorithm. *Proceedings of the IEEE International Conference on Rehabilitation Robotics (ICORR 2011)*, Zurich, Switzerland, June 27-July 1, 2011, 1–7. <https://doi.org/10.1109/ICORR.2011.5975346>
- Mahony, R., Hamel, T., & Pflimlin, J.-M. (2008). Nonlinear complementary filters on the special orthogonal group. *IEEE Transactions on Automatic Control*, 53(5), 1203–1218. <https://doi.org/10.1109/TAC.2008.923738>

- Maniadakis, M., Tani, J., & Trahanias, P. (2011). Ego-centric and allo-centric abstraction in self-organized hierarchical neural networks. *Proceedings of the IEEE International Conference on Development and Learning, (ICDL 2011)*, Frankfurt am Main, Germany, August 24-27, 2011, 2, 1–6. <https://doi.org/10.1109/DEVLRN.2011.6037347>
- Mantis Vision. (2013). Mantis Vision’s MVC-F5 Hand-Held Imager. [http://www.sgmlightwave.com/wp-content/uploads/2014/06/Mantis\\_small.pdf](http://www.sgmlightwave.com/wp-content/uploads/2014/06/Mantis_small.pdf)
- Martin, H. (2016). *Overcoming the challenges of low-cost inertial navigation* [Doctoral dissertation, University College London]. [http://discovery.ucl.ac.uk/1524153/1/martin\\_pdfjoiner.pdf](http://discovery.ucl.ac.uk/1524153/1/martin_pdfjoiner.pdf)
- McLuhan, M., & Fiore, Q. (1967). *The medium is the message*. Gingko Press Inc.
- Michel, T., Fourati, H., Genevès, P., & Layaïda, N. (2015). A comparative analysis of attitude estimation for pedestrian navigation with smartphones. *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN 2015)*, Banff, Alberta, Canada, October 13–16, 2015, 1–10. <https://doi.org/10.1109/IPIN.2015.7346767>
- Microsoft. (2019). *DateTime.Ticks Property*. <https://docs.microsoft.com/en-us/dotnet/api/system.datetime.ticks?view=net-5.0>
- Mittelstaedt, M.-L., & Mittelstaedt, H. (1980). Homing by path integration in a mammal. *Naturwissenschaften*, 67(11), 566–567. <https://doi.org/10.1007/BF00450672>
- Mulloni, A., Seichter, H., Duenser, A., Baudisch, P., & Schmalstieg, D. (2012). 360° panoramic overviews for location-based services. *Proceedings of the Conference on Human Factors in Computing Systems (CHI 12)*, Austin, Texas, USA, May 5 - 10, 2012, 2565-2568. <https://doi.org/10.1145/2207676.2208645>
- Murphy, J. J. (1873). Instinct: A mechanical analogy. *Nature*, 7(182), 483. <https://doi.org/10.1038/007483b0>
- National Transportation Safety Board (NTSB). (2014). *Safety alert-landing at the wrong airport check and confirm destination airport*. <https://www.nts.gov/Advocacy/safety-alerts/Documents/SA-033.pdf>
- Neale, S., Chinthammit, W., Lueg, C., & Nixon, P. (2013). RelicPad: A Hands-On, Mobile Approach to Collaborative Exploration of Virtual Museum Artifacts. *Proceedings of the IFIP Conference on Human-Computer Interaction (INTERACT 2013)*, Cape Town, South Africa, September 2-6, 2013, 8117, 86-103. [https://doi.org/10.1007/978-3-642-40483-2\\_7](https://doi.org/10.1007/978-3-642-40483-2_7)
- Nikraz, M., Caire, G., & Bahri, P. (2006). A methodology for the development of multi-agent systems using the JADE platform. *Comput. Syst. Sci. Eng.*, 21. [https://jade.tilab.com/doc/tutorials/JADE\\_methodology\\_website\\_version.pdf](https://jade.tilab.com/doc/tutorials/JADE_methodology_website_version.pdf)

- O’Keefe, J., & Burgess, N. (1996). Geometric determinants of the place fields of hippocampal neurons. *Nature*, 381(6581), 425–428. <https://doi.org/10.1038/381425a0>
- O’Keefe, J., & Nadel, L. (1978). *The Hippocampus as a Cognitive Map*. Oxford University Press. <http://hdl.handle.net/10150/620894>
- Padgham, L., & Winikoff, M. (2003). Prometheus: a methodology for developing intelligent agents. *Proceedings of the first international joint conference on Autonomous agents and multiagent systems (AAMAS 02)*, Bologna, Italy, July 15-19, 2002, 1, 37-38. <https://doi.org/10.1145/544741.544749>
- Papoulis, A. (1991). *Probability, random variables, and stochastic processes* (4th ed.). McGraw-Hill. [https://www.academia.edu/35834044/Probability\\_Random\\_Variables\\_and\\_Stochastic\\_Processes\\_Fourth\\_Edition\\_Papoulis](https://www.academia.edu/35834044/Probability_Random_Variables_and_Stochastic_Processes_Fourth_Edition_Papoulis)
- Phidgets. (2017, November 29). *Allan deviation primer*. [https://www.phidgets.com/docs/Allan\\_Deviation\\_Primer](https://www.phidgets.com/docs/Allan_Deviation_Primer)
- Renaudin, V., Ortiz, M., Perul, J., Torres-Sospedra, J., Jiménez, A. R., Pérez-Navarro, A., Martín Mendoza-Silva, G., Seco, F., Landau, Y., Marbel, R., Ben-Moshe, B., Zheng, X., Ye, F., Kuang, J., Li, Y., Niu, X., Landa, V., Hacoheh, S., Shvalb, N., ... Park, Y. (2019). Evaluating indoor positioning systems in a shopping mall: The lessons learned from the IPIN 2018 competition. *IEEE Access*, 7, 148594–148628. <https://doi.org/10.1109/ACCESS.2019.2944389>
- Roberto, R., Lima, J. P., Araújo, T., & Teichrieb, V. (2016). Evaluation of motion tracking and depth sensing accuracy of the Tango tablet. *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality Workshops (ISMAR-Adjunct)*, Merida, Mexico, September 19-23, 2016, 231–234. <https://doi.org/10.1109/ISMAR-Adjunct.2016.0082>
- Rosen, J. (2014, November 10). The knowledge, London’s legendary taxi-driver test, puts up a fight in the age of GPS. *New York Times*. <https://www.nytimes.com/2014/11/10/t-magazine/london-taxi-test-knowledge.html>
- Sala, P., Sim, R., Shokoufandeh, A., & Dickinson, S. (2006). Landmark selection for vision-based navigation. *IEEE Transactions on Robotics*, 22(2), 334–349. <https://doi.org/10.1109/TRO.2005.861480>
- Sjanic, Z., Skoglund, M. A., & Gustafsson, F. (2017). EM-SLAM with inertial/visual applications. *IEEE Transactions on Aerospace and Electronic Systems*, 53(1), 273–285. <https://doi.org/10.1109/TAES.2017.2650118>

Skoglund, M. A. (2011). *Visual inertial navigation and calibration* (Publication No. 421464) [Doctoral dissertation, Linköping University].  
<http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-68858>

Smith, R. (1980). The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, C-29(12), 1104–1113.  
<https://doi.org/10.1109/TC.1980.1675516>

Taube, J. S., Muller, R. U., & Ranck, J. B. (1990). Head-direction cells recorded from the postsubiculum in freely moving rats. I. Description and quantitative analysis. *Journal of Neuroscience*, 10(2), 420–435. <https://doi.org/10.1523/JNEUROSCI.10-02-00420.1990>

Telecom Italia Lab. (n.d. a). JADE api (Version 4.5.0). Telecom Italia.  
<https://jade.tilab.com/doc/api/index.html>

Telecom Italia Lab. (n.d. b). WADE api (Version 3.6.0). Telecom Italia.  
<https://jade.tilab.com/wade/doc/api/index.html>

Telecom Italia Lab. (n.d. c). AMUSE api (Version 1.5.0). Telecom Italia.  
<https://jade.tilab.com/amuse/doc/api/index.html>

Tolman E. C. (1948). Cognitive maps in rats and men. *Psychological review*, 55(4), 189–208.  
<https://doi.org/10.1037/h0061626>

Tsai, D. (2012). *Autonomous vision-based docking of the tethered axel rover for planetary exploration* (Publication No. 1017486) [Master's Thesis, Luleå University of Technology].  
<http://urn.kb.se/resolve?urn=urn:nbn:se:ltu:diva-44210>

Unity. (2015). *Create and connect with Unity 5*.  
<https://docs.unity3d.com/540/Documentation/Manual/UnityRemote5.html>

Unity. (2016a). *Navigation areas and costs*.  
<https://docs.unity3d.com/530/Documentation/Manual/nav-AreasAndCosts.html>

Unity. (2016b). *Navigation overview*. <https://docs.unity3d.com/530/Documentation/Manual/nav-Overview.html>

Wasinger, R., He, H., Chinthammit, W., Collis, C., Duh, H., & Kay, J. (2014). The importance of “neighbourhood” in personalising location-based services. *Proceedings of the 26th Australian Computer-Human Interaction Conference on Designing Futures: The Future of Design (OzCHI 14)*, Sydney, New South Wales, Australia, December 2-5, 2014, 172–175.  
<https://doi.org/10.1145/2686612.2686638>

Wastler, A. (2014, June 21). *Elon Musk, Stephen Hawking and fearing the machine*. CNBC.  
<https://www.cnbc.com/2014/06/21/ays-artificial-intelligence-is-a-danger.html#>

- Wen, J., Deneka, A., Helton, W., & Billinghamurst, M. (2014a). Really, it's for your own good... making augmented reality navigation tools harder to use. *Conference on Human Factors in Computing Systems (CHI 14)*, Toronto, Ontario, Canada, April 26-May 1, 2014, 1297–1302. <https://doi.org/10.1145/2559206.2581156>
- Wen, J., Deneka, A., Helton, W. S., Dünser, A., & Billinghamurst, M. (2014b). Fighting technology dumb down: Our cognitive capacity for effortful ar navigation tools. *Proceedings of the Human-Computer Interaction. Applications and Services (HCI 2014)*, Heraklion, Crete, Greece, June 22-27, 2014, 525–536. [https://doi.org/10.1007/978-3-319-07227-2\\_50](https://doi.org/10.1007/978-3-319-07227-2_50)
- Wen, J., Helton, W., & Billinghamurst, M. (2013a). A study of user perception, interface performance, and actual usage of mobile pedestrian navigation aides. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, September, 2013, 57(1), 1958–1962. <https://doi.org/10.1177/1541931213571437>
- Wen, J., Helton, W., & Billinghamurst, M. (2013b). Classifying users of mobile pedestrian navigation tools. *Proceedings of the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration (OzCHI '13)*, Adelaide, Australia, November 25 - 29, 2013, 13–16. <https://doi.org/10.1145/2541016.2541073>
- Woodman, O. J. (2007). An introduction to inertial navigation. University of Cambridge Computer Laboratory Technical Report, *SAE Technical Papers*, 696. <https://doi.org/10.48456/tr-696>
- Woollett, K., & Maguire, E. A. (2011). Acquiring “the knowledge” of London’s layout drives structural brain changes. *Current Biology*, 21(24), 2109–2114. <https://doi.org/10.1016/j.cub.2011.11.018>
- Wu, M., Chang, M., Heh, J., & Kinshuk. (2011). Landmark-based navigation system for mobile devices have no built-in gps receiver and compass. *Proceedings of the 10th World Conference on Mobile and Contextual Learning, (mLearn 2011)*, Beijing, China, October 18-21, 2011, 250-258. <http://maiga.athabascau.ca/publication/Conference-2011-mLearn2011.pdf>
- VectorNav. (2020). *Inertial navigation primer*. <https://www.vectornav.com/resources/inertial-navigation-primer>
- Vuforia (2020). *Vuforia developer library*. <https://library.vuforia.com/getting-started/overview.html>.
- Zeinalipour-Yazti, D., Laoudias, C., Georgiou, K., & Chatzimilioudis, G. (2016). Internet-based indoor navigation services. *IEEE Internet Computing*, 1-6. <https://doi.org/10.1109/MIC.2016.21>
- Zhi, R. (2016). *A drift eliminated attitude & position estimation algorithm in 3d* [Master’s Thesis, University of Vermont]. <https://scholarworks.uvm.edu/graddis/450>

**Appendix A: Gyro Allan Standard Deviation R Plot Script (Phidgets, 2017)**

```
// https://www.r-project.org/
```

```
sd(data$AccX)
```

```
data <- read.csv("rawGyroLog.csv",header=T)
```

```
end <- length(data$Time)
```

```
frequency <- 1/mean(data$Time[2:end]-data$Time[1:(end-1)])
```

```
library("allanvar")
```

```
avar.data.x <- avar(data$GyroX,frequency)
```

```
avar.data.y <- avar(data$GyroY,frequency)
```

```
avar.data.z <- avar(data$GyroZ,frequency)
```

```
options(scipen=5)
```

```
plot(avar.data.z$time, sqrt(avar.data.x$av)*57.29578, log="xy", type="l",
```

```
col="blue", xlab="", ylab="")
```

```
lines(avar.data.y$time, sqrt(avar.data.z$av)*57.29578, col="green")
```

```
lines(avar.data.x$time, sqrt(avar.data.y$av)*57.29578, col="red")
```

```
lines(avar.data.x$time,
```

```
(sqrt(avar.data.x$av)*57.29578+sqrt(avar.data.y$av)*57.29578+sqrt(avar.data.z$av)*57.29578)/
```

```
3, col="black")
```

```
grid(equilog=FALSE, lwd=1, col="orange")
```

```
legend("topright", c("Gyro X Axis","Gyro Y Axis","Gyro Z Axis","Gyro Mean"),
```

```
fill=c("blue","red","green","black"))
```

```
title(main="", xlab="Cluster Time (s)", ylab="Allan Standard Deviation (°/s)")
```

**Appendix B: Accelerometer Allan Standard Deviation R Plot Script (Phidgets, 2017)**

```
// https://www.r-project.org/

data <- read.csv("rawAccelLog.csv",header=T)

end <- length(data$Time)

frequency <- 1/mean(data$Time[2:end]-data$Time[1:(end-1)])

library("allanvar")

avar.data.x <- avar(data$AccX,frequency)

avar.data.y <- avar(data$AccY,frequency)

avar.data.z <- avar(data$AccZ,frequency)

plot(avar.data.x$time,avar.data.x$av,type="l",col="blue",xlab="Sample Time (seconds, at ~25
samples/s)",ylab=expression(paste("Allan variance (", g^2,")")))

lines(avar.data.y$time,avar.data.y$av,col="red")

lines(avar.data.z$time,avar.data.z$av,col="green")

options(scipen=5)

plot(avar.data.x$time,avar.data.x$av,type="l",col="blue",log="xy",xlab="Sample Time
(seconds, at ~25 samples/s)",ylab=expression(paste("Allan variance (", g^2,")")))

lines(avar.data.y$time,avar.data.y$av,col="red")

lines(avar.data.z$time,avar.data.z$av,col="green")

options(scipen=5)

plot(avar.data.x$time,sqrt(avar.data.x$av),type="l",col="blue",log="xy",xlab="Sample Time
(seconds, at ~25 samples/s)",ylab="Allan deviation (g)")

lines(avar.data.y$time,sqrt(avar.data.y$av),col="red")

lines(avar.data.z$time,sqrt(avar.data.z$av),col="green")
```

```

options(scipen=5)

plot(avar.data.z$time, sqrt(avar.data.y$av), log="xy", type="l", col="red", xlab="", ylab="")

lines(avar.data.y$time, sqrt(avar.data.z$av), col="green")

lines(avar.data.x$time, sqrt(avar.data.x$av), col="blue")

lines(avar.data.x$time, (sqrt(avar.data.x$av)+sqrt(avar.data.y$av)+sqrt(avar.data.z$av))/3,
col="black")

grid(equilog=FALSE, lwd=1, col="orange")

legend("topright", c("Accelerometer
X Axis", "Accelerometer Y Axis", "Accelerometer Z Axis", "Accelerometer Mean"),
fill=c("blue", "red", "green", "black"))

title(main="", xlab="Cluster Time (s)", ylab="Allan Standard Devation (g)")

```

### **Appendix C: XML Log File**

The system writes to the xml log file when the following occurs:

- 1) Click on save a screenshot note,
- 2) Click on RESET,
- 3) click on “Do you understand?”,
- 4) GNSS recorded,
- 5) ART pose recorded,
- 6) INS reset after drift > 3m.
- 7) Reached with current target or finish.

Log file = AEGN + AndroidId () + .xml

- (1) MODE\_ROUTE
- (2) HEIGHT
- (3) VELOCITY
- (4) FROM\_TARGET
- (5) TO\_TARGET
- (6) TRAVEL\_DISTANCE
- (7) TRAVEL\_TIME
- (8) TRAVEL\_SPEED
- (9) 20 CONTROL\_ITEMS\_9-28
- (29) PITS\_DISTANCE\_ANSWER\_KEY
- (30) PITS\_DISTANCE\_USER\_ANSWER
- (31) PITS\_DISTANCE\_USER\_SCORE

## INS USING ART FOR CORRECTION AND COGNITION

- (32) PITS\_DIRECTION\_ANSWER\_KEY
- (33) PITS\_DIRECTION\_USER\_ANSWER
- (34) PITS\_DIRECTION\_USER\_SCORE
- (35) UFOV\_ANSWER\_KEY
- (36) UFOV\_USER\_ANSWER
- (37) UFOV\_USER\_SCORE
- (38) MRT\_ANSWER\_KEY
- (39) MRT\_USER\_ANSWER
- (40) MRT\_USER\_SCORE
- (41) PITF\_DISTANCE\_ANSWER\_KEY
- (42) PITF\_DISTANCE\_USER\_ANSWER
- (43) PITF\_DISTANCE\_USER\_SCORE
- (44) PITF\_DIRECTION\_ANSWER\_KEY
- (45) PITF\_DIRECTION\_USER\_ANSWER
- (46) PITF\_DIRECTION\_USER\_SCORE
- (47) UNDERSTAND
- (48) USER\_SCORE
- (49) DEVICE\_SCORE
- (50) FLOORS
- (51) USER\_ID
- (52) TEST\_NUM
- (53) DATE\_TIME
- (54) INS\_REFERENCE\_POSITION\_X

## INS USING ART FOR CORRECTION AND COGNITION

- (55) INS\_REFERENCE\_POSITION\_Y
- (56) INS\_REFERENCE\_POSITION\_Z
- (57) INS\_REFERENCE\_DIRECTION
- (58) INS\_POSITION\_X
- (59) INS\_POSITION\_Y
- (60) INS\_POSITION\_Z
- (61) INS\_DIRECTION
- (62) INS\_ESTIMATED\_ACCURACY
- (63) INS\_ESTIMATED\_ACCURACY\_TIME
- (64) INS\_POSITION\_ERROR
- (65) INS\_DIRECTION\_ERROR
- (66) ART\_REFERENCE\_POSITION\_X
- (67) ART\_REFERENCE\_POSITION\_Y
- (68) ART\_REFERENCE\_POSITION\_Z
- (69) ART\_REFERENCE\_DIRECTION
- (70) ART\_POSITION\_X
- (71) ART\_POSITION\_Y
- (72) ART\_POSITION\_Z
- (73) ART\_DIRECTION
- (74) ART\_COMBINED\_FOV
- (75) ART\_RATING
- (76) ART\_POSITION\_ERROR
- (77) ART\_DIRECTION\_ERROR

## INS USING ART FOR CORRECTION AND COGNITION

- (78) GYRO\_POSITION\_X
- (79) GYRO\_POSITION\_Y
- (80) GYRO\_POSITION\_Z
- (81) GYRO\_DIRECTION
- (82) GYRO\_ESTIMATED\_ACCURACY
- (83) GYRO\_ESTIMATED\_ACCURACY\_TIME
- (84) GYRO\_POSITION\_ERROR
- (85) GYRO\_DIRECTION\_ERROR
- (86) GNSS\_POSITION\_X
- (87) GNSS\_POSITION\_Y
- (88) GNSS\_POSITION\_Z
- (89) GNSS\_DIRECTION
- (90) GNSS\_HORIZONTAL\_ACCURACY
- (91) GNSS\_HORIZONTAL\_ACCURACY\_TIME
- (92) GNSS\_POSITION\_ERROR
- (93) GNSS\_DIRECTION\_ERROR
- (94) ACCELEROMETERBIASX
- (95) ACCELEROMETERBIASY
- (96) ACCELEROMETERBIASZ
- (97) GYROSCOPEBIASX
- (98) GYROSCOPEBIASY
- (99) GYROSCOPEBIASZ
- (100) ACCELEROMETERALIGNXX

- (101) ACCELEROMETERALIGNXY
- (102) ACCELEROMETERALIGNXZ
- (103) ACCELEROMETERALIGNYX
- (104) ACCELEROMETERALIGNYY
- (105) ACCELEROMETERALIGNYZ
- (106) ACCELEROMETERALIGNZX
- (107) ACCELEROMETERALIGNZY
- (108) ACCELEROMETERALIGNZZ
- (109) GYROSCOPEGDEPXX
- (110) GYROSCOPEGDEPXY
- (111) GYROSCOPEGDEPXZ
- (112) GYROSCOPEGDEPYX
- (113) GYROSCOPEGDEPYY
- (114) GYROSCOPEGDEPYZ
- (115) GYROSCOPEGDEPZX
- (116) GYROSCOPEGDEPZY
- (117) GYROSCOPEGDEPZZ

## Appendix D: Google Maps and AEGN Trip Plans

**Figure D1**

*AEGN Map Path (4.0m/s)*



**Figure D2**

*AEGN Map Path (0.83m/s)*



**Figure D3**

*Google Maps Path (Google, n.d. b)*



**Table D1***Google Maps (Google, n.d. b) and AEGN Path Descriptions From McDonald's to Gogi*

Google Maps	AEGN	
	Relative text	Cardinal text
<p><b>McDonald's</b> 507 23 Ave NW, Calgary, AB T2M 1S7</p> <p>1. Head east on 23 Ave NW toward 4 St NW 27 m</p> <p>2. Turn right onto 4 St NW 38 m</p> <p>3. Turn left 7 m Destination will be on the left</p> <p>58 s (72 m)</p> <p><b>GOGI Korean BBQ</b> 2320 4 St NW, Calgary, AB T2M 2Z6</p>	<p>Walking Public User ht: 1.0 m vel: 0.83 m/s</p> <p>From: McDonald's To: Gogi</p> <p>Path Control Description: distance: 92.86m time: 111.30s</p> <p>1 31 Lower Left corner</p> <p>2 32 Front parking xwalk</p> <p>3 33 McDonald's 0.00m</p> <p>4 34 start 0.00m</p> <p>5 35 Right S sidewalk 9.01m</p> <p>6 36 Front W sidewalk</p> <p>7 37 Front 2300 block 4th St</p> <p>8 38 Front 4th St S xwalk</p> <p>9 39 Right E sidewalk 62.87m</p> <p>10 40 Front Left parking xwalk</p> <p>11 41 Front Gogi 89.74m</p> <p>Front control 6 90.05m</p> <p>SCORE: Do you understand?</p> <p>User: Basic Orienteering Device: Auto Pose</p>	<p>Walking Public User ht: 1.0 m vel: 0.83 m/s</p> <p>From: McDonald's To: Gogi</p> <p>Path Control Description:</p> <p>1 31 SW corner</p> <p>2 32 N parking xwalk</p> <p>3 33 McDonald's</p> <p>4 34 start</p> <p>5 35 E S sidewalk</p> <p>6 36 SE W sidewalk</p> <p>7 37 E 2300 block 4th St</p> <p>8 38 E 4th St S xwalk</p> <p>9 39 S E sidewalk</p> <p>10 40 SE parking xwalk</p> <p>11 41 E Gogi</p> <p>E control 6</p> <p>SCORE: Do you understand?</p> <p>User: Direction Bearing Device: Auto Pose</p>

**Table D2***Google Maps (Google, n.d. b) and AEGN Path Descriptions From Gogi to Stavro's*

Google Maps	AEGN	
	Relative text	Cardinal text
<p><b>GOGI Korean BBQ</b> 2320 4 St NW, Calgary, AB T2M 2Z6</p> <p>4. Head west toward 4 St NW 7 m</p> <p>5. Turn right onto 4 St NW 38 m</p> <p>6. Turn left onto 23 Ave NW 63 m Destination will be on the right</p> <p>1 min (110 m)</p> <p><b>Stavro's on Fourth Restaurant &amp; Sports Bar</b> 2411 4 St NW Unit 101, Calgary, AB T2M 2Z8</p>	<p>Walking Public User ht: 1.0 m vel: 0.83 m/s</p> <p>From: Gogi To: Stavros</p> <p>Path Control Description: distance: 142.25m time: 170.49s</p> <p>1 31 Lower Left corner</p> <p>2 32 Back parking xwalk</p> <p>3 33 Gogi 0.00m</p> <p>4 34 control 6 0.00m</p> <p>5 35 Left E sidewalk 20.48m</p> <p>6 36 Front Left 2300 block</p> <p>7 37 Front Left 4th St S</p> <p>8 38 Front W sidewalk</p> <p>9 39 Front Right 23rd Ave V</p> <p>10 40 Front Right 500 block</p> <p>11 41 Left N sidewalk 58.58m</p> <p>12 42 Right parking xwalk</p> <p>13 43 Right parking xwalk</p> <p>14 44 Front Stavros 139.12m</p> <p>Front control 4 139.44m</p> <p>SCORE: Do you understand?</p> <p>User: Basic Orienteering Device: Auto Pose</p>	<p>Walking Public User ht: 1.0 m vel: 0.83 m/s</p> <p>From: Gogi To: Stavros</p> <p>Path Control Description:</p> <p>1 31 Lower Left corner</p> <p>2 32 Back Left parking xwa</p> <p>3 33 Gogi 0.00m</p> <p>4 34 control 6 0.00m</p> <p>5 35 Left E sidewalk 15.03m</p> <p>6 36 Front Left 4th St S</p> <p>7 37 Front Left 2300 block</p> <p>8 38 Front W sidewalk</p> <p>9 39 Front Right 23rd Ave V</p> <p>10 40 Front Right 500 block</p> <p>11 41 Left N sidewalk 52.98m</p> <p>12 42 Right parking xwalk</p> <p>13 43 Right parking xwalk</p> <p>14 44 Front Stavros 133.64m</p> <p>Front control 4 133.95m</p> <p>SCORE: Do you understand?</p> <p>User: Basic Orienteering Device: Auto Pose</p>

**Table D3**

*Google Maps (Google, n.d. b) and AEGN Path Descriptions From Stavro's to Velvet*

Google Maps	AEGN	
	Relative text	Cardinal text
<p>Stavro's on Fourth Restaurant &amp; Sports Bar 2411 4 St NW Unit 101, Calgary, AB T2M 2Z8</p> <p>7. Head east on 23 Ave NW toward 4 St NW 63 m</p> <p>8. Turn left onto 4 St NW 270 m</p> <p>9. Turn left Destination will be on the left 29 m</p> <p>4 min (350 m)</p> <p>Velvet Cafe 502 25 Ave NW, Calgary, AB T2M 2A8</p>	<p>Walking Public User ht: 1.0 m vel: 0.83 m/s</p> <p>From: Stavros To: Velvet Cafe</p> <p>Path Control Description: distance: 264.56m time: 317.09</p> <p>1 31 → Lower Left corner</p> <p>2 32 → Right parking xwalk</p> <p>3 33 → Stavros 0.10m</p> <p>4 34 → control 4 0.10m</p> <p>5 35 → Front Right parking</p> <p>6 36 → Front parking xwalk</p> <p>7 37 → Right S sidewalk</p> <p>8 38 → Left 500 block 24th Ave</p> <p>9 39 → Left 24th Ave W xwalk</p> <p>10 40 → Front W sidewalk</p> <p>11 41 → Front Right control 7</p> <p>12 42 → Front W sidewalk</p> <p>13 43 → Front control 9 205.45m</p> <p>14 44 → Front W sidewalk</p> <p>15 45 → Front Right control 8</p> <p>16 46 → Front 500 block 25th</p> <p>17 47 → Front 25th Ave W xwalk</p> <p>18 48 → Front W sidewalk</p> <p>19 49 → Front Left parking xwalk</p> <p>20 50 → Front Velvet 261.45m</p> <p>Front control 3 261.77m</p> <p>SCORE: I do understand.</p> <p>User: Basic Orienteering Device: Auto Pose</p>	<p>Walking Public User ht: 1.0 m vel: 0.83 m/s</p> <p>From: Stavros To: Velvet Cafe</p> <p>Path Control Description:</p> <p>1 31 → SW corner</p> <p>2 32 → E parking xwalk</p> <p>3 33 → Stavros</p> <p>4 34 → control 4</p> <p>5 35 → N parking sidewalk</p> <p>6 36 → N parking xwalk</p> <p>7 37 → E S sidewalk</p> <p>8 38 → N 500 block 24th Ave</p> <p>9 39 → N 24th Ave W xwalk</p> <p>10 40 → N W sidewalk</p> <p>11 41 → NE control 7</p> <p>12 42 → N W sidewalk</p> <p>13 43 → N control 9</p> <p>14 44 → N W sidewalk</p> <p>15 45 → NE control 8</p> <p>16 46 → N 500 block 25th Ave</p> <p>17 47 → N 25th Ave W xwalk</p> <p>18 48 → N W sidewalk</p> <p>19 49 → NW parking xwalk</p> <p>20 50 → NW Velvet</p> <p>NW control 3</p> <p>SCORE: Do you understand?</p> <p>User: Direction Bearing Device: Auto Pose</p>

**Table D4**

*Google Maps (Google, n.d. b) and AEGN Path Descriptions From Velvet to John's*

Google Maps	AEGN	
	Relative text	Cardinal text
<p><b>Velvet Cafe</b> 502 25 Ave NW, Calgary, AB T2M 2A8</p> <p>10. Head east toward 4 St NW 29 m</p> <p>11. Turn left onto 4 St NW 4 m Destination will be on the right</p> <p>24 s (33 m)</p> <p><b>John's Breakfast &amp; Lunch</b> 2614 4 St NW A, Calgary, AB T2M 3A1</p>	<p>Walking Public User   ht: 1.0 m vel: 0.83 m/s</p> <p>From: Velvet Cafe   To: John's Breakfast</p> <p>Path Control Description: distance: 113.86m time: 136.46s</p> <p>1 31 Lower Left corner</p> <p>2 32 Left parking xwalk</p> <p>3 33 Velvet 0.00m</p> <p>4 34 control 3 0.00m</p> <p>5 35 Front W sidewalk 9.55</p> <p>6 36 Front 500 block 25th</p> <p>7 37 Front 25th Ave W xwalk</p> <p>8 38 Front W sidewalk</p> <p>9 39 Front control 8 37.24m</p> <p>10 40 Front Left 4th St S xwalk</p> <p>11 41 Front Right E sidewalk</p> <p>12 42 Front 400 block 25th</p> <p>13 43 Front 25th Ave E xwalk</p> <p>14 44 Front E sidewalk</p> <p>15 45 Front John's 110.49m</p> <p>Front control 2 110.82m</p> <p>SCORE: Do you understand?</p> <p>User: Basic Orienteering   Device: Auto Pose</p>	<p>Walking Public User   ht: 1.0 m vel: 0.83 m/s</p> <p>From: Velvet Cafe   To: John's Breakfast</p> <p>Path Control Description:</p> <p>1 31 SW corner</p> <p>2 32 SW parking xwalk</p> <p>3 33 Velvet</p> <p>4 34 control 3</p> <p>5 35 SE W sidewalk</p> <p>6 36 S 500 block 25th Ave</p> <p>7 37 S 25th Ave W xwalk</p> <p>8 38 S W sidewalk</p> <p>9 39 SE control 8</p> <p>10 40 SE 4th St S xwalk</p> <p>11 41 E E sidewalk</p> <p>12 42 N 400 block 25th Ave</p> <p>13 43 N 25th Ave E xwalk</p> <p>14 44 N E sidewalk</p> <p>15 45 N John's</p> <p>N control 2</p> <p>SCORE: Do you understand?</p> <p>User: Direction Bearing   Device: Auto Pose</p>

**Table D5**

*Google Maps (Google, n.d. b) and AEGN Path Descriptions From John's to 4<sup>th</sup> Spot*

Google Maps	AEGN	
	Relative text	Cardinal text
<p><b>John's Breakfast &amp; Lunch</b> 2614 4 St NW A, Calgary, AB T2M 3A1</p> <p>12. Head north on 4 St NW toward 26 Ave NW 44 m</p> <p>13. Turn right onto 26 Ave NW 41 m</p> <p>14. Turn right 13 m Destination will be on the right</p> <p>1 min (98 m)</p> <p><b>4th Spot Kitchen &amp; Bar</b> 2620 4 St NW, Calgary, AB T2M 3A3</p>	<p>Walking Public User   ht: 1.0 m vel: 0.83 m/s</p> <p>From: John's Breakfast   To: 4th Spot</p> <p>Path Control Description: distance: 38.34m time: 45.96s</p> <p>1 31 Lower Left corner</p> <p>2 32 John's 0.00m</p> <p>3 33 control 2 0.00m</p> <p>4 34 Front Right E sidewalk</p> <p>5 35 Front check in 8.31m</p> <p>6 36 Front muster 8.91m</p> <p>7 37 Front Left parking xwalk</p> <p>8 38 Front 4th Spot 35.24m</p> <p>9 39 Front control 1 35.51m</p> <p>10 40</p> <p>SCORE: Do you understand?</p> <p>User: Basic Orienteering   Device: Auto Pose</p>	<p>Walking Public User   ht: 1.0 m vel: 0.83 m/s</p> <p>From: John's Breakfast   To: 4th Spot</p> <p>Path Control Description:</p> <p>1 31 SW corner</p> <p>2 32 John's</p> <p>3 33 control 2</p> <p>4 34 N E sidewalk</p> <p>5 35 N check in</p> <p>6 36 N muster</p> <p>7 37 NW parking xwalk</p> <p>8 38 N 4th Spot</p> <p>9 39 N control 1</p> <p>10 40</p> <p>SCORE: Do you understand?</p> <p>User: Direction Bearing   Device: Auto Pose</p>

**Appendix E: Test Scenarios****Table E1***Test Scenarios*

	Path generation		Task allocation	Calibration	Spatial cognition	Navigation	IPIN 2018
Device/ Application	Linux Server/ Database		Linux Server/ Database, Amuse, Jason				Smart phones, laptops/ Various apps
	Windows Desktop/ AutoDesk 3DS Max, Unity Editor, AEGN Navigator		Windows Desktop/ Android Studio				
	Android 8 physical/ Tango Tablet		Android 8 physical, emulated/ AEGN Coordinator	Android 8 physical/ AEGN Navigator	Android 8 physical/ AEGN Navigator	Android 8 physical/ AEGN Navigator	
Category							
Ref System <sup>a</sup> System <sup>b</sup> Points <sup>c</sup> overlay	local lat-long 3d model			local lat-long 3d scan	local lat-long 3d scan	local lat-long 3d scan	WGS84 lat-long 3D scan
Performance <sup>d</sup> dist metric m <sup>e</sup> dist Statistic <sup>f</sup> time metric s <sup>e</sup> time statistic <sup>f</sup> process time <sup>g</sup> set-up(m <sup>2</sup> /hr)	3D mag MAE, $\sigma$ mag MAE, $\sigma$ Ticks		mag MAE, $\sigma$ mag MAE, $\sigma$ ms	3D mag MAE, $\sigma$ mag MAE, $\sigma$ quat/s	3D mag MAE, $\sigma$ mag MAE, $\sigma$ ms, ticks	3D mag MAE, $\sigma$ ,Q3 mag MAE, $\sigma$ quat/s 1000	2D mag MAE,Q3    1000
Method type <sup>h</sup> level <sup>i</sup> repeatable <sup>j</sup> reproducible	sim comp same cond yes		sim comp same cond yes	sim/real comp/sys simult yes/no	real comp/sys same cond no	real comp/sys simult no	sim/real sys simult no
Test							
ETL <sup>k</sup> person <sup>l</sup> velocity(m/s)	auto walk/roll 4	man Table F1 1.7	auto walk/roll 0.83-1.7	auto/man walk/roll 0.83	man walk/roll 1.2	man walk/roll 0.83	auto/man walk
Site type <sup>m</sup> level area(m <sup>2</sup> )	AU,MP O,W,S/St 3/1 >10000	AU O,W,S 3 >10000	MP St. 1 >10000	MP St. 1 >10000	MP St. 1 >10000	MP St. 1 >10000	AU,MP O,W,S/St. 4 >10000
Point number accuracy(m)	6 <0.1	257 <0.1	6 <0.1	6 <0.1	6 <0.1	6 <0.1	180 <0.1

*Note:* <sup>a</sup>Ref = reference. <sup>b</sup>local = local site coordinate system and WGS84 = World Geodetic System 1984. <sup>c</sup>lat-long = latitude longitude. <sup>d</sup>dist = distance. <sup>e</sup>mag = magnitude. <sup>f</sup> $\sigma$  = standard deviation and Q3 = 3rd Quartile or 75th Percentile. <sup>g</sup>ticks = CPU ticks, 10000ticks=1ms (Microsoft, 2021) and quat = quaternion. <sup>h</sup>sim = simulated. <sup>i</sup>comp = component and sys = system. <sup>j</sup>same cond = same conditions and simult = simultaneous. <sup>k</sup>man = manual and auto = automated. <sup>l</sup>walk = pedestrian walking and roll = wheelchair rolling. <sup>m</sup>O = Office, W = Warehouse, S = Subterranean, and St. = Street.

## **Appendix F: Path Generation Test Phases**

### 1) Preparation:

- a) Open project and path generation scene in Unity Editor.
- b) 3D Models
  - i) AU Main Campus
    - (1) Import 3D model from AutoDesk 3DS Max into Unity.
    - (2) Scan control points, drawing gaps, and discrepancies in large building dwg with Tango Tablet ( $\sim 1000\text{m}^2/\text{hr}$ ).
    - (3) Import 3D models from Tango Tablet scan into Unity.
  - ii) Mount Pleasant business strip
    - (1) Take Google satellite snapshots to create a 3D model in Unity.
    - (2) Scan the sidewalks with Tango Tablet ( $\sim 1000\text{m}^2/\text{hr}$ ) and import into Unity to verify snapshot scale and position.
- c) Move the control points to the test point locations in each 3D model.  
Bake NavMesh in Unity.  
For performance testing uncomment stopwatch sections of GO0100 Script.cs.
- d) Build apk and install on android 8 device.

### 2) Execution:

- a) Automated path generation - run path generation scene in windows Unity Editor  
( $\sim 1\text{min}/\text{path}$  or  $\sim 72\text{min}$ ).
- b) Manual path generation - run motion tracking Navigate scene (AU Main Campus only).
  - i) To find an object search the project hierarchy for objects (i.e., door) and while the scene window is open, double click on the name to zoom to the object.

- ii) In the game window set the test value (Table F1).
  - iii) Manual test points with manual pose.
  - iv) If the object controls direction (i.e., secure door – red and green doormat) then test both directions.
- c) If any paths are not generated as expected,
  - i) then look for issue(s) in Object Navigation Area, Navigation Area Cost, NavMesh default Off-Mesh Links, Exit Off-MeshLink Tag, and/or Unity agent Area Mask
  - ii) Fix and retest object.
- 3) Dissemination:
  - a) Optimal distances
    - i) C:\Users\%username%\AppData\LocalLow\AEGN\Navigator\distanceArray.sql
    - ii) Make the starting control point 0 to make the data 0 based index (i.e., 5 to 0).
    - iii) Change higher keys accordingly (i.e., 6 to 5).
    - iv) Load distances into server database for task allocation.
  - b) Optimal paths
    - i) Copy sprites from C:\Users\%username%\AppData\LocalLow\AEGN\
      - (1) walking-" + fromTarget.name + "-" + toTarget.name + ".png
      - (2) rolling-" + fromTarget.name + "-" + toTarget.name + ".png
    - ii) Copy sprites to %Unity project directory%\Assets\Standard Assets\CrossPlatformInput\Sprites.
    - iii) Load sprites in GO1100\_Dropdown Atlas Manager. If any sprites are deleted, then reload the Atlas.
    - iv) Build AEGN Navigator for Orientation and Navigation tests.

## c) A\* Performance `

i) When the Unity agent path passes through a control point it triggers a recalculation.

ii) logged at each control point.

(1) A\*StartTime

(2) A\*Ticks

(3) A\*Milliseconds

(4) A\*StopTime

iii) Log file = AEGN + AndroidId () + .txt saved for performance analysis.

**Table F1***AU Main Campus Manual Test Points*

Personalization/Standards	Object type	Setting test value	Points
Mobility/security/safety	Doors	Walking public	19
		Walking public/muster	19
		Walking authorized	19
		Walking authorized/muster	19
		Rolling public	19
		Rolling public/muster	19
		Rolling authorized	19
		Rolling authorized /muster	19
Mobility/safety	Stairs	Walking	25
		Rolling	25
		Rolling/muster	25
	Elevators	Walking	1
		Walking/muster	1
		Rolling	1
		Rolling/muster	1
Mobility	Ramps	Rolling	10
Clearance	Low ceiling	2m height	1
Privacy	Washrooms	Walking	9
	Locker rooms	Walking	4
Indoor	Plaza	Walking	2
			257

### **Appendix G: Task Allocation Test Phases**

1) Preparation:

a) 4D Models – spatial-temporal

i) Mount Pleasant business strip

(1) Load distances into server database from path generation.

(2) Use AEGN Coordinator to enter the schedule information for the calendar event tasks for the MP control points.

2) Execution:

a) A user can only be in one group at a time.

b) Using AEGN Coordinator: Manually configure and run 1 user, 2 users and 3 users group test combinations (Table G1).

c) If a user wants solo navigation, then click “Invite selected users” without selecting any other users. This will start processing allocations for solo navigation.

d) Group selection starts with searching for users.

e) Enter a username and click the search button.

f) If the search finds a registered user with that name, then it will add them to the online or offline tab depending on if they are logged in or not.

g) Repeat the search if there are other users you would like to add to your group.

h) Groups can contain up to three people including yourself, so you can select 1 or 2 online users.

i) Click “Invite selected users” to join your group.

j) After you Confirm Inviting the users then you will automatically go to the destination negotiation page where you wait up to 300s for the other users to join.

- k) Other users will receive an invite to accept or reject joining your user group when their Welcome activity is open.
- l) When a user accepts the invitation then you will get a message announcing they have joined, and their name will be added at the top of the page.
- m) If a user rejects the invitation or if a user does not respond to the invitation, then a virtual player will replace them in the destination negotiation. Virtual usernames begin with a capital V like Vicky or Vaffa.
- n) If you only invite one other user, then the system will add a virtual user to the group.
- o) When there are three users then the destination negotiation will begin.
- p) Users' alternate turns selecting preferences from a row of 5 buttons for 60s each.
- q) The top bar will turn green when it is your turn.
- r) The top bar will turn red while you are waiting for your turn.
- s) Each player sees his/her marked spaces as red circles and other players as grey crosses.
- t) When all the destinations are selected then processing allocations starts on the server and a timer starts for 180s.
- u) A user will start solo navigation in the following situations:
  - i) If a user's time to select destinations expires (60s)
  - ii) If a user does not select any destination preference.
  - iii) If the leader leaves the group, then all group users go solo.
  - iv) If a user leaves the group.
- v) If a solo user leaves while allocations are processing, then processing will continue on the server.
- w) Processing allocations runs on the server and takes into consideration all the destinations,

paths, users, and their preferences to efficiently and equitably allocate the destinations.

- i) All users are assigned the same starting point (5) because it is a public space accessible to all users.
- ii) All users are also assigned their first selected destination as their preference as determined in negotiations.
- iii) The remaining destinations will be assigned to minimize the total group travel time.
- x) AEGN Coordinator will switch to the User Information activity after the 180s processing allocations timer expires.
- y) If the user is part of a group, then when processing allocations is completed the user's user information and schedule information will be updated
- z) If the user is a solo navigator, then when processing allocations is completed the user's user information will be updated.

### 3) Dissemination:

- a) Optimal ordered task allocation based on user preferences and group travel time.
  - i) `SELECT * FROM USER_PROPERTY WHERE KEY LIKE 'TASK_PREF'`
  - ii) `SELECT * FROM USER_PROPERTY WHERE KEY LIKE 'TASK_ALLOC'`
  - iii) `SELECT * FROM USER_PROPERTY WHERE KEY LIKE 'TRAVEL_TIME'`
  - iv) `SELECT * FROM USER_PROPERTY WHERE KEY LIKE 'TRAVEL_TAX'`
  - v) Follow the task allocated order when navigating with the AEGN Navigator.
- b) brute force mTSP plus CTM Performance
  - i) logged for each user in each group.

```
(1) SELECT      *      FROM      USER_PROPERTY      WHERE      KEY      LIKE  
      'PROCESSING_TIME%'
```

ii) Saved for performance analysis.

**Table G1**

*4 Users Group Test Combinations*

User	User1	User2	User3	User4
User1	X			
	X	X		
	X		X	
	X			X
	X	X	X	
	X	X		X
	X		X	X
User2		X		
		X	X	
		X		X
		X	X	X
User3			X	
			X	X
User4				X

## **Appendix H: Calibration Test Phases**

### 1) Preparation:

- a) For performance testing uncomment stopwatch sections of  
TangoDeltaPoseController0.cs.
- b) Build apk and install on android 8 device.
- c) Open MP version of AEGN Navigator and the calibrate scene.
  - i) Test type – simulation: indoor room temperature
  - ii) Test type – real: outdoor air temperature
- d) 3D Models
  - i) Mount Pleasant business strip
    - (1) Open MP version of AEGN Navigator and Calibrate scene.
    - (2) Warm-up Clutch for 30min
    - (3) Use Allan deviation analysis on raw gyro and accelerometer readings to  
determine cluster times.
    - (4) Run calibration.
      - (a) Warm-up Clutch for 30min
      - (b) Enter noise reduction time clusters from Allan deviation analysis.
      - (c) Z +
        - (i) 1st - Turn the back down so the screen faces up.
        - (ii) 2nd - Hold it flat to the ground and steady.
        - (iii) 3rd - Press Cluster Clutch and let the timer run out.
      - (d) Y +
        - (i) 1st - Turn the top of the screen up.

(ii) 2nd - Hold it perpendicular to the ground and steady.

(iii)3rd - Press Cluster Clutch and let the timer run out.

(e) X +

(i) 1st - Turn the right side of the screen up.

(ii) 2nd - Hold it perpendicular to the ground and steady.

(iii)3rd - Press Cluster Clutch and let the timer run out.

(f) Y -

(i) 1st - Turn the bottom of the screen up.

(ii) 2nd - Hold it perpendicular to the ground and steady.

(iii)3rd - Press Cluster Clutch and let the timer run out.

(g) X -

(i) 1st - Turn the left side of the screen up.

(ii) 2nd - Hold it perpendicular to the ground and steady.

(iii)3rd - Press Cluster Clutch and let the timer run out.

(h) Z -

(i) 1st - Turn the back up so the screen faces down.

(ii) 2nd - Hold it flat to the ground and steady.

(iii)3rd - Press Cluster Clutch and let the timer run out.

## 2) Execution:

a) Test type – simulation

i) Open MP version of AEGN Navigator and Navigate scene.

ii) Settings:

MODE\_ROUTE=Walking Public User

HEIGHT/VELOCITY=ht: 1.2m vel: 1.0m/s

FROM\_TARGET=From: 4<sup>th</sup> Spot

TO\_TARGET=To: Muster (distance~1km, time~17min)

T=Time constant ( $\tau$ ) in Table H1

dt=0.04s sample period from Allan Variance analysis sample frequency = 25Hz

UNDERSTAND=Yes

USER\_SCORE=User: Basic Orientation

DEVICE\_SCORE=Device: Auto INS

FLOORS=Floors: All

CAMERA\_TYPE=Top North

iii) Accelerate forward and then gently place down in the same orientation.

iv) Brace the device steady in a stationary position.

v) Decelerate quickly and stand steady to stop.

vi) Records xml log automatically

vii) It will stop moving when the ground truth reference Unity agent reaches the finish target.

b) Test type – real

i) Recalibrate if required.

ii) Open MP version of AEGN Navigator and Navigate scene.

iii) Settings:

MODE\_ROUTE=Walking Public User

HEIGHT/VELOCITY=ht: 1.2m vel: 1.0m/s

FROM\_TARGET=From: 4<sup>th</sup> Spot using ART Pose

TO\_TARGET=To: Muster

T=0.7 for longest Mean Distance in Table H1

dt=0.04

UNDERSTAND=Yes

USER\_SCORE=User: Basic Orientation

DEVICE\_SCORE=Device: Auto INS

FLOORS=Floors: All

CAMERA\_TYPE=Top North

- iv) Accelerate forward and then walk at a steady pace. The velocity then fluctuates in waves as the user steps and it works best if those waves have a small amplitude (Figure H1).
- v) Point the INS FOV in the direction of travel and try to hold the device steady when walking.
- vi) Records xml log automatically.
- vii) It will stop moving when the ground truth reference Unity agent reaches the finish target.

3) Dissemination:

- a) The system writes xml log file when the following occurs:
  - i) Click on RESET,
  - ii) Log file = AEGN + AndroidId () + .xml
- b) INS Performance
  - i) Logged for each generated quaternion.
    - (1) INSSStartTime

(2) INSTicks

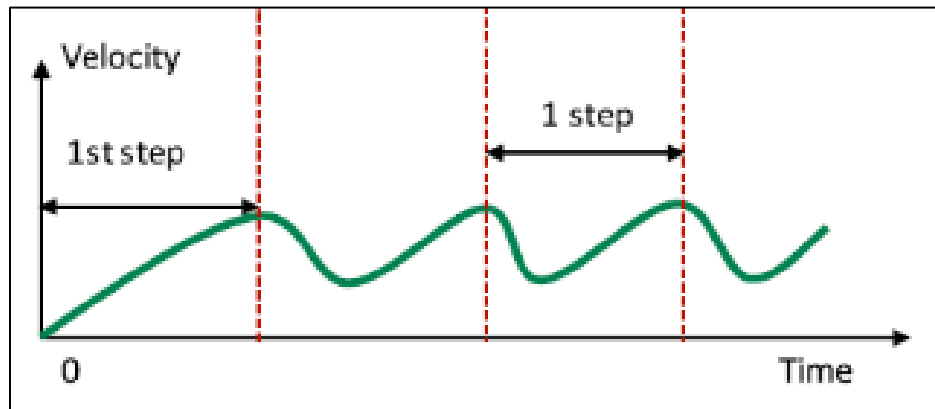
(3) INSMilliseconds

(4) INSStopTime

ii) Log file = AEGN + AndroidId () + .txt saved for performance analysis.

**Table H1***Complementary Filter Tuning Tests (Distance to Drift Error > 3m at 1m/s)*

Time constant ( $\tau$ )	Time interval (dt)	Filter coefficient (a)	Mean distance (m)
1.0	0.04	0.961538	99.60 $\pm 17.70$
0.9	0.04	0.957447	153.17 $\pm 50.82$
0.8	0.04	0.952381	163.50 $\pm 30.16$
0.7	0.04	0.945946	219.25 $\pm 27.32$
0.6	0.04	0.937500	122.75 $\pm 17.91$
0.5	0.04	0.925926	97.80 $\pm 6.39$

**Figure H1***Walking Velocity Fluctuations (Renaudin et al., 2019, p. 148604)*

### **Appendix I: Spatial Cognition Test Phases**

- 1) Preparation: (4hr)
  - a) Open MP version of AEGN Navigator and ART Pose scene.
  - b) 3D Models
    - i) Mount Pleasant business strip
      - (1) Path sprites and control points were already prepared in path generation.
      - (2) Configure the Unity project for ART Pose to transform the user's pose based on the camera's perspective.
      - (3) Open Unity project and ART Pose scene.
        - (a) Photograph planar art near control points, use Tango Tablet scan or measure for position, rotation, and scale (width and height).
        - (b) Open Target Manager on <https://developer.vuforia.com/vui/develop/databases> and add photos to the AUMainCampus database.
        - (c) Download the unitypackage database and import it into the unity project.
        - (d) Configure ARTargets and Sprites for each new photo.
      - (4) Open motion tracking Navigate scene.
        - (a) Add ARTarget sprites. Do not place sprites coincident because rendering will flicker which can cause some people to be disoriented. Set navigational objects at the measured coordinates and offset other rendered objects.
    - c) For Spatial Cognition performance testing uncomment stopwatch sections of GO9100\_Script.cs.
    - d) Open Navigate scene.
    - e) Enable GameObject\_Scoring and GameObject\_Errors to view scoring debug fields.

- f) Build apk and install on android 8 device.
- g) Open AEGN Navigator and calibrate scene for indoor room temperature. This is important because the gyro and accelerometer sensors are very sensitive to temperature.

### 2) Execution:

- a) Test type – simulation: indoor room temperature
  - i) Open MP version of AEGN Navigator and Navigate scene:
    - (1) MODE\_ROUTE=Walking Public User
    - (2) HEIGHT/VELOCITY=ht: 1.2m vel: 1.0m/s
  - ii) Open ART Pose scene
    - (1) FROM\_TARGET: ART Pose each of the 6 control points
      - (a) Point the camera at the ART photo.
      - (b) For accuracy keep pin centred to ART.
      - (c) Click on Navigate to save your ART Pose.
  - iii) Open Navigate scene:
    - (1) TO\_TARGET: each of the other 5 control points
      - (a) Select a new TO\_TARGET. If you select the current TO\_TARGET or select the FROM\_TARGET that is the same as the TO\_TARGET or muster, then nothing happens. This is to ensure that what is mapped corresponds to what was selected.
    - (2) PITS\_DISTANCE\_USER\_ANSWER:
      - (a) The first question asks for the distance and direction to the start or fire zone 5 control point at the front entrance.
      - (b) Set the distance to start.

- (c) Set the distance unit.
- (d) The distance can be entered in meters or steps.
- (e) Steps are averaged to be about 40% of your height.

(3) PITS\_DIRECTION\_USER\_ANSWER:

- (a) Set the direction to start.
- (b) Direction is entered in degrees from north where clockwise is positive.
- (c) There are options to enter the direction.
  - (i) typing or
  - (ii) manually clicking on the map on a point in the direction of the start or
  - (iii) ART Pose – follow on screen instructions to point towards the start.
    1. Point the camera at ART and in the desired direction.
    2. For accuracy keep pin centred to ART.
    3. Click on motion tracking Navigate to save your ART Pose.

(4) UFOV\_USER\_ANSWER:

- (a) The second question asks you to select the arrow pointing in the finish direction.
- (b) To help there is the flashing animation with a red dot in the direction to the fire zone that you selected to go to next.
- (c) Click on the arrow in that direction.
- (d) Set path.

(5) MRT\_USER\_ANSWER:

- (a) The third question asks you to select the path shown from a different top perspective.

(b) Icons show mini maps with just the circles of the control points and the triangle of the current muster point which will be close to your origin point. These look like star constellations such as Aries which like the Athabasca Main Campus building are both shaped like an L that has been rotated.

(c) Click on the mini map that shows the path for the control points you select to go from the origin and to finish.

(6) PITF\_DISTANCE\_USER\_ANSWER:

(a) The fourth question is like the first and asks for the distance and direction to the finish.

(b) Set the distance to finish.

(c) Set the distance unit.

(7) PITF\_DIRECTION\_USER\_ANSWER:

(a) Set the direction to finish.

(b) There are options to enter the direction.

(i) typing or

(ii) manually clicking on the map on a point in the direction of the finish or

(iii) ART Pose – follow on screen instructions to point towards the finish.

1. Point the camera at ART and in the desired direction.

2. For accuracy keep pin centred to ART.

3. Click on motion tracking Navigate to save your ART Pose.

(8) Confirm correct SCORE by viewing scoring debug fields (Figure 35).

(9) UNDERSTAND=Yes

(a) The fifth and final question simply asks if you understand where you are going.

(b) This clears the questions and calculates your score which determines the display of the PCD and the map.

(c) This also finishes the test and writes the details to the log.

(10) Confirm correct USER\_SCORE.

### 3) Dissemination:

#### a) Spatial Cognition tasks

##### i) Optimal viewing properties

(1) For each control point (1-6)

(a) For each question (1-5)

(i) check each answer to ensure the calculated SCORE is correct.

##### ii) The system writes xml log file when the following occurs:

(1) Click on “Do you understand?”,

(2) Log file = AEGN + AndroidId () + .xml

#### b) Spatial Cognition Performance

##### i) logged when the user answers the final SCORE question, “Do you understand?”.

(1) OrientStartTime

(2) OrientTicks

(3) OrientMilliseconds

(4) OrientStopTime

##### ii) Log file = AEGN + AndroidId () + .txt saved for performance analysis.

### **Appendix J: Navigation Test Phases**

- 1) Preparation: (in addition to Spatial Cognition Test preparation)
  - a) Open MP version of AEGN Navigator and Navigate scene.
  - b) For Navigation performance testing uncomment stopwatch sections of  
TangoDeltaPoseController0.cs.
  - c) Enable GameObject\_Scoring and GameObject\_Errors to view scoring debug fields.
  - d) Build apk and install on android 8 device.
  - e) Open AEGN Navigator and calibrate scene for outdoor air temperature. This is important because the gyro and accelerometer sensors are very sensitive to temperature.
- 2) Execution:
  - a) Test type – real: outdoor air temperature
    - i) Open MP version of AEGN Navigator and Navigate scene:
      - (1) MODE\_ROUTE=Walking Public User
      - (2) HEIGHT/VELOCITY=ht: 1.0m vel: 0.83m/s
    - ii) Open ART Pose scene
      - (3) FROM\_TARGET (see Table J1): ART Pose
        - (a) Stand at known ART reference location for FROM\_TARGET in Table J1.
        - (b) Point the camera at the ART object.
        - (c) For accuracy keep pin centred to ART.
        - (d) Click on Navigate to save your ART Pose.
    - ii) Open Navigate scene:
      - (4) TO\_TARGET (see Table J1):

Select a new TO\_TARGET from Table J1. If you select the current TO\_TARGET

or select the FROM\_TARGET that is the same as the TO\_TARGET or muster, then nothing happens. You must select a different zone than what is currently displayed to trigger the guiding Unity agent. This is to ensure that what is mapped corresponds to what was selected. The PCD follows the IOF standard.

(5) USER\_SCORE=User: Basic Orientation

(6) DEVICE\_SCORE=Device: Auto INS

(7) FLOORS=Floors: All

(8) CAMERA\_TYPE=Top Forward

(a) Top Forward allows the user to easily see rotations without zooming in so they can see the ground truth reference Unity agent as well as surrounding landmarks.

(b) Accelerate quickly and then walk at a steady pace to move forward.

(c) Point the INS FOV in the direction of travel and try to hold the device steady when walking.

(d) Decelerate quickly and stand steady to stop.

(e) Records xml log automatically.

(f) It will stop moving when the ground truth reference Unity agent reaches the finish target.

1) Dissemination:

a) Navigate

i) The system writes xml log file when the following occurs:

ii) Click on save a screenshot note,

iii) Click on “Do you understand?”,

## INS USING ART FOR CORRECTION AND COGNITION

- iv) GNSS recorded,
  - v) ART pose recorded,
  - vi) INS reset after drift > 3m.
  - vii) Every 10s after INS reset
  - viii) Reached with current target or finish.
  - ix) Log file = AEGN + AndroidId () + .xml
- b) Navigate Performance
- i) Logged for each generated quaternion.
    - (1) INSStartTime
    - (2) INSTicks
    - (3) INSMilliseconds
    - (4) INSStopTime
  - ii) Log file = AEGN + AndroidId () + .txt saved for performance analysis.
- c) Estimate setup time in minutes.

**Table J1**

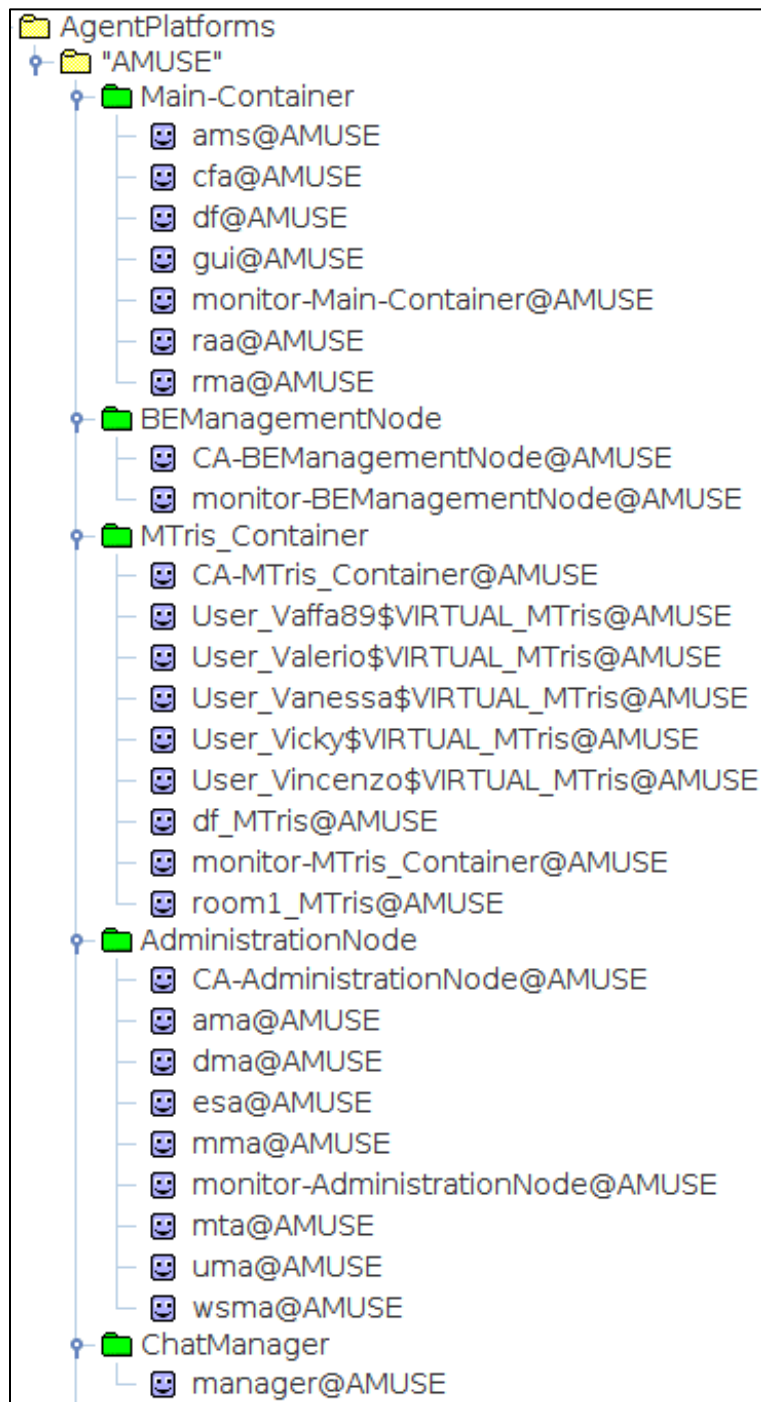
*The Speeds/Paths With a Travel Time Greater Than 180s*

Path	From Target	To Target	Ht (m)	Vel (m/s)	Distance (m)	Time (s)
3_TO_0	Velvet	McDonald's	1	0.83	296.51	357
0_TO_2	McDonalds	Johns	1	0.83	327.46	394
2_TO_5	Johns	Gogi	1	0.83	297.04	358
5_TO_1	Gogi	4 <sup>th</sup> Spot	1	0.83	337.42	407
1_TO_4	4 <sup>th</sup> Spot	Stavros	1	0.83	330.14	398
4_TO_3	Stavros	Velvet	1	0.83	264.51	319

## Appendix K: Agents

**Figure K1**

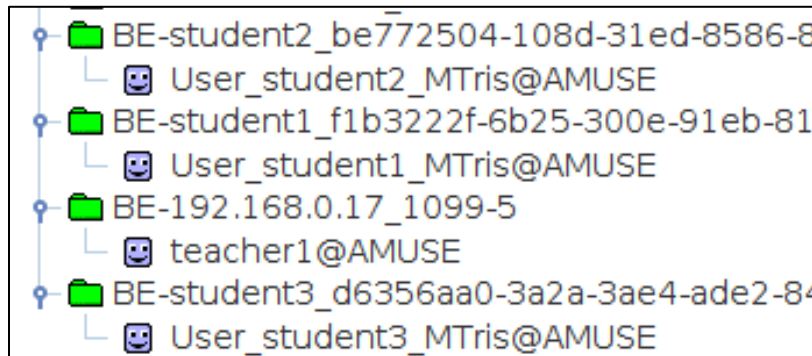
*AMUSE Platform, System Containers (Green Nodes) and Service Agents Which Run Continuously*



**Figure K2**

*AMUSE Platform, Back-End Coordinator AMUSE Containers (Green Nodes) and Agents Which*

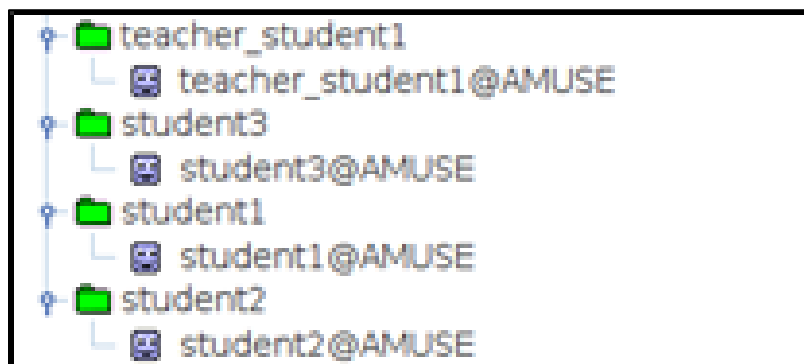
*Run When the App is Active*



**Figure K3**

*AMUSE Platform, Back-End Coordinator Jason Containers (Green Nodes) and Agents Which*

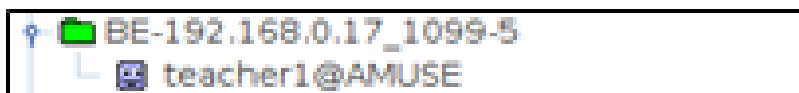
*Run When the Task Allocation is Active*



**Figure K4**

*AMUSE Platform, Back-End Communicator JADE Containers (Green Nodes) and Agents Which*

*Run When the App is Active*



**Table K1***Agent Types*

Container	Nickname	API	Agent type
Main-Container	ams	JADE <sup>a</sup>	Agent management system
	cfa	WADE <sup>b</sup>	Configuration
	df	JADE <sup>a</sup>	Directory facilitator
	gui	WADE <sup>b</sup>	Management
	monitor-Main-Container	WADE <sup>b</sup>	Container-monitor
	raa	WADE <sup>b</sup>	Runtime allocator
	rma	JADE <sup>a</sup>	Remote monitoring
BEManagementNode	CA-BEManagementNode	WADE <sup>b</sup>	Controller
	monitor	WADE <sup>b</sup>	Container-monitor
MTris_Container	CA-MTris_Container	WADE <sup>b</sup>	Controller
	User_Vaffa89\$VIRTUAL_MTris	AMUSE <sup>c</sup>	Virtual player
	User_Valerio\$VIRTUAL_MTris	AMUSE <sup>c</sup>	Virtual player
	User_Vanessa\$VIRTUAL_MTris	AMUSE <sup>c</sup>	Virtual player
	User_Vicky\$VIRTUAL_MTris	AMUSE <sup>c</sup>	Virtual player
	User_Vincenzo\$VIRTUAL_MTris	AMUSE <sup>c</sup>	Virtual player
	df_MTris	JADE <sup>a</sup>	Directory facilitator
	Monitor-MTris_Container	WADE <sup>b</sup>	Container-monitor
	Room1_MTris	AMUSE <sup>c</sup>	Board
AdministrationNode	CA-AdministrationNode	WADE <sup>b</sup>	Controller
	ama	AMUSE <sup>c</sup>	Application manager
	dma	AMUSE <sup>c</sup>	Developer manager
	esa	WADE <sup>b</sup>	Event system
	mma	AMUSE <sup>c</sup>	Match manager
	Monitor-AdministrationNode	WADE <sup>b</sup>	Container-monitor
	mta	AMUSE <sup>c</sup>	Match tracer
	uma	AMUSE <sup>c</sup>	User manager
	wsma	WADE <sup>b</sup>	Workflow status manager
ChatManager	manager	JADE <sup>a</sup>	Chat manager
BE-192.168.0.17_1099-#	Nickname (i.e., teacher1)	JADE <sup>a</sup>	BE Communicator user
BE-nickname_be#...	User_nickname_MTris	AMUSE <sup>c</sup>	BE Coordinator user
		JADE <sup>a</sup>	Communicator app
		AMUSE <sup>c</sup>	Coordinator app
teacher_nickname	teacher_nickname (i.e., student1)	Jason	Initiator (Group leader)
nickname	nickname (i.e., student2)	Jason	Participant (Group user)
na	na	Unity	Navmesh

Note: <sup>a</sup>(Tilab, n.d. a). <sup>b</sup>(Tilab, n.d. b). <sup>c</sup>(Tilab, n.d. c).

**Table K2***Agent Descriptions*

API	Agent type	Agent description
JADE <sup>a</sup>	Agent management system	The agent management and white page service
	Directory facilitator	The platform yellow page service
	Remote monitoring	A graphical platform management console.
	Chat manager	Tracks of all communicator agents.
	BE Communicator user	For communication with active communicator app
	Communicator app	Communicator app
WADE <sup>b</sup>	Configuration	Interacts with boot daemons and controls the application life cycle
	Management	WADE Management Console for starting/stopping an application, importing/exporting configurations and deploying workflows
	Container-monitor	Monitor, discover unreachable or dead containers.
	Runtime allocator	Controls agent pools and fault tolerance.
	Controller	Supervises the local container and WADE fault tolerance
	Event system	Implements the wade event system to suspend workflow
	Workflow status manager	Traces workflow executions and to persists state.
AMUSE <sup>c</sup>	Virtual player	Join negotiations even if only one user to encourage user to be honest about preference lest another agent picks it first.
	Board	Implements the room in the mtris game.
	Application manager	Manages the provided games and their lifecycle.
	Developer manager	Developer manager agent
	Match manager	Interfaces the client application with backend agents
	Match tracer	Provides persistent game state and restarts.
	User manager	Manages the profile of users and relationships.
	BE Coordinator user	For communication with active coordinator app
	Coordinator app	Coordinator app
Jason	Initiator (Group leader)	Runs when task allocation is active
	Participant (Group user)	Runs when task allocation is active
Unity	Navmesh	Navigator app

Note: <sup>a</sup>(Tilab, n.d. a). <sup>b</sup>(Tilab, n.d. b). <sup>c</sup>(Tilab, n.d. c).

**Appendix L: IPIN 2018 Competition Results****Table L1***IPIN 2018 Competition Results with Reported Descriptions (Renaudin et al., 2019)*

Track constraint	Team	Result		Reported description									
		75 <sup>TH</sup> Percentile (m)	MAE (m)	Map matching	Signal Sensor					Other Sensor			
					Wi-Fi	Geomagnetic	Bluetooth	Cellular	GNSS	Accelerometer	Gyro	Magnetometer	Barometric
Test type – simulated													
Smartphone	Five-WHU	0.90	0.70		X	X				X	X	X	X
	EGEC	1.30	1.10	X	X	X				X	X		X
	HFTS	3.60	3.00	X	X	X				X	X		
	ARARADS	4.20	10.80		X	X	X						
	UGENT	4.20	3.50	X	X					X	X		X
	YAI	6.20	4.60		X					X			
	TUM	7.60	6.30	X	X	X		X					X
	MCLIPS	8.80	8.00		X								
Foot-mounted	WHU	1.30	1.00			X			X	X	X		
	KIT	3.90	2.80						X	X	X		
	AOE	7.90	11.30							X	X	X	X
	YAI	90.80	66.10							X		X	
Test type – real													
Non-camera	KYUSHU	5.70	5.20	X						X	X	X	X
	SNU	6.80	4.50							X	X		
	ETRI	12.40	8.70							X	X		
	TUM	15.00	12.80	X						X		X	
	UPJS	37.50	30.30							X	X		X
Camera	ARIEL-2	11.70	9.80	X									
	ARIEL-1	15.20	11.70	X						X	X		
	ETRI	18.90	16.20										
	Google	-	-							X	X		

*Note:* Five-WHU=Wuhan University composed of Xingyu Zheng, Feng Ye and Jian Kuang.

EGEC=Zhenxing Ding and Feng Xu. HFTS=Stuttgart University of Applied Sciences composed of Stefan Knauth. ARARADS=Arara composed of Tomás Lungenstrass and Juan Pablo Morales.

UGENT=Ghent University composed of Jens Trogh and David Plets. YAI composed of Shih-

Hau Fang, Yu Tsao, Ying-Ren Chien, Shi-Shen Yang and Shih-Jyun Ye. TUM=Technical

University of Munich composed of Georgios Pipelidis and Nikolaos Tsiamitros. MCLIPS composed of Muhammad Usman Ali, Soojung Hur and Yongwan Park. WHU=Wuhan University composed of Yu Li, Xiaoji Niu and Jian Kuang. KIT=Karlsruhe Institute of Technology composed of Nikolai Kronenwett. AOE=Academy of Opto-Electronics composed of Wenchao Zhang, Xianghong Li and Dongyan Wei. KYUSHU=Kyushu University composed of Chuanhua Lu, Hideaki Uchiyama, Diego Thomas, Atsushi Shimada and Rin-ichiro Taniguchi. SNU=Seoul National University composed of Soyeon Lee, Blagovest Vladimirov, Eunyoung Cho, Sungwoo Jun, Changeun Lee, Sangjoon Park, So Young Park, Chan Gook Park, Yonghyun Lee, Jehyeok Rew, Changjun Park Hyeongyo Jeong, Jaeseung Han and Keumryeol Lee. ETRI=Electronics and Telecommunications Research Institute composed of SNU. UPJS=P. J. Šafárik University composed of Miroslav Opiela. ARIEL-2=Ariel University composed of Yael Landau, Revital Marbel, and Boaz Ben-Moshe. ARIEL-1=Ariel University composed of Vlad Landa, Shlomi Hacohen, Nir Shvalb and Boaz Ben-Moshe. Google composed of Ying Zhang.

## Appendix M: Certification of Ethical Approval



### CERTIFICATION OF ETHICAL APPROVAL

The Athabasca University Research Ethics Board (AUREB) has reviewed and approved the research project noted below. The AUREB is constituted and operates in accordance with the current version of the Tri-Council Policy Statement: Ethical Conduct for Research Involving Humans (TCPS) and Athabasca University Policy and Procedures.

**Ethics File No.:** 23059

**Principal Investigator:**

Mr. Kenneth Mulder, Graduate Student  
Faculty of Science & Technology\Master of Science in Information Systems

**Supervisor:**

Dr. Maiga Chang (Supervisor)

**Project Title:**

Inertial Navigation System Using a Monocular Camera and Fiducial Markers for Corrections and Spatial Cognition Learning Tasks

**Effective Date:** June 27, 2019

**Expiry Date:** June 26, 2020

**Restrictions:**

Any modification or amendment to the approved research must be submitted to the AUREB for approval.

Ethical approval is valid *for a period of one year*. An annual request for renewal must be submitted and approved by the above expiry date if a project is ongoing beyond one year.

A Project Completion (Final) Report must be submitted when the research is complete (*i.e. all participant contact and data collection is concluded, no follow-up with participants is anticipated and findings have been made available/provided to participants (if applicable)*) or the research is terminated.

**Approved by:**

**Date:** June 27, 2019

Carolyn Greene, Chair  
Athabasca University Research Ethics Board

---

Athabasca University Research Ethics Board  
University Research Services, Research Centre  
1 University Drive, Athabasca AB Canada T9S 3A3  
E-mail rebsec@athabascau.ca  
Telephone: 780.675.6718