Exploitation of Geographic Information Systems
for
Vehicular Destination Prediction

THESIS

Richard T. Muster, Capt, USAF

AFIT/GCE/ENG/09-05

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

AFIT/GCE/ENG/09-05

Exploitation of Geographic Information Systems for Vehicular Destination Prediction

THESIS

Presented to the Faculty
Department of Electrical Engineering and Computer Science
Graduate School of Engineering and Management
Air Force Institute of Technology
Air University
Air Education and Training Command
In Partial Fulfillment of the Requirements for the
Degree of Master of Science in Computer Engineering

Richard T. Muster, B.S.
Capt, USAF

March 2009

AFIT/GCE/ENG/09-05

Exploitation of Geographic Information Systems for Vehicular Destination Prediction

Richard T. Muster, B.S.
Capt, USAF

Approved:

/signed/ 21 Mar 2009

Maj Michael Mendenhall, PhD (Chairman) date

/signed/ 21 Mar 2009

Dr. Gillbert Peterson (Member) date

/signed/ 21 Mar 2009

Dr. Kenneth Hopkinson (Member) date

AFIT/GCE/ENG/09-05

Abstract

Much of the recent successes in the Iraqi theater have been achieved with the aid of technology so advanced that celebrated journalist Bob Woodward recently compared it to the *Manhattan Project* of WWII. Intelligence, Surveillance, and Reconnaissance (ISR) platforms have emerged as the rising star of Air Force operational capabilities as they are enablers in the quest to track and disrupt terrorist and insurgent forces. This thesis argues that ISR systems have been severely under-exploited. The proposals herein seek to improve the machine-human interface of current ISR systems such that a predictive battle-space awareness may be achieved, leading to shorter kill-chains and better utilization of high demand assets.

This thesis shows that, if a vehicle is being tracked by an ISR platform, it is possible to predict where it might go within a *Time Horizon*. This predictive knowledge is represented graphically to enable quick decisioning. This is accomplished by using Geo-Spatial Information Systems (GIS) obtained from municipal, commercial, or other ISR sources (e.g., hyperspectral) to model an urban grid. It then employs graph-theoretic search algorithms that prune the future state-space of that vehicle's environment, resulting in an envelope that constricts around all possible destinations.

This thesis demonstrates an 81 % success rate for predictions carried out during experimentation. It further demonstrates a 97 % improvement over predictions made solely with models based on vehicular motion. This thesis reveals that the predictive envelopes show immense promise in improving ISR asset management, offering more intelligent interdiction of targets, and enabling ground sensor-cueing. Moreover, these predictive capabilities allow an operator to assign assets to make precise perturbations on the battle-space for true event-shaping. Finally, this thesis shows that the proposed methodologies are easily and cost-effectively deployed over existing Air Force architectures using the Software as a Service business model.

Acknowledgements

The author would like to extend sincere thanks to the following individuals, without whose help, this thesis would not have succeeded. The members of the thesis committee, Maj. Michael Mendenhall, Dr. Gilbert Peterson, and Dr. Kenneth Hopkinson, for all of their guidance and mentoring. Maria Scalzo and Raj Malhotra of the Air Force Research Laboratories for their crucial technical advice on vehicle destination prediction. Dr. Steve Rogers, of the Air Force Research Laboratories, for providing a wealth of technical literature on the subject as well as practical advice. Dr. Devert Wicker of the Air Force Research Institute for his kind sponsorship of this work. Dr. Sanjeeb Nanda of SDS International Inc. for advice that led to the algorithmic methods proposed in this thesis. Douglass Abernathy of Lockheed Martin for sharing the technical details of Dynamic Tactical Targeting and for offering advice on technical aspects of the current work. Don Smith and Mark Smearcheck of the AFIT Advanced Navigation Technology lab for the use of the Leica GPS1200 surveying system (and advice on how to use it). Deanna Meekler, of Environmental Systems Research Institute, for outstanding software support. Dr. Juan Vasquez of Numerica Inc. for his advice on contemporary tracking methods. TSgt Edwin Velez of the 367 Training Support Squadron for the 3D Models used in some of the visualizations. The administrative staff of the TechEdge Collaboratorium for their tireless assistance in the office. Most importantly, my wife, whose forbearance and understanding permitted this work to proceed.

Richard T. Muster

Table of Contents

Page

List of Figures

Figure Page

Figure Page

Figure Page

List of Tables

List of Abbreviations

Abbreviation		Page
GIS	Geo-Spatial Information Systems	iv
UAS	Unmanned Aerial System	1-1
OODA	Observe, Orient, Decide, Act	1-2
ATO	Air Task Order	1-3
JFC	Joint Forces Commander	1-4
JFACC	Joint Force Air and Space Component Commander	1-4
COMAFFOR	Commander of Air Force Forces	1-4
GWoT	Global War on Terror	1-4
DAG	Directed Acyclic Graph	1-5
HUMINT	Human Intelligence	1-6
MASINT	Measurements and Signals Intelligence	1-6
SIGINT	Signals Intelligence	1-6
IMINT	Image Intelligence	1-6
OSINT	Open Source Intelligence	1-6
GEOINT	Geo-Spatial Intelligence	1-6
GLONASS	Global Orbiting Navigation Satellite System	1-6
IED	Improvised Explosive Device	1-9
EO	Electro Optical	1-11
SAR	Synthetic Aperture Radar	1-11
IR	Infra Red	1-11
CAOC	Combined Air Operations Center	1-15
DTT	Dynamic Tactical Targeting	2-18
CERDE	US Army Communications-Electronics Research, Development, and Engineering Center	2-20
PD-Map	Probability of Detection Map	2-25

Abbreviation		Page
JPADS	Joint Precision Air-Drop System	3-11
HERF	High Energy Radio Frequency	3-11
HPM	High-Powered Powered Microwave	3-11
UGS	Unattended Ground Sensors	3-11
ANT	Advanced Navigation Technology	4-14
V-RAMBO	Video Receiver And Monitor for Battlefield Operations	5-7
LSLG	Layered Sensing Leadership Group	1
COTS	Commercial off the Shelf	4
IAD	Integrated Air Defense	4
SaaS	Software as a Service	5
SOA	Services Oriented Architecture	5
DCGS	Distributed Common Ground Station	5

Exploitation of Geographic Information Systems for Vehicular Destination Prediction

I. Introduction

1.1 The Air Force has Failed the War-Fighter

The US Air Force has endured a searing *time 'o' troubles* over the past year that has seen, among other controversies, a crisis of culture as regards Intelligence Surveillance, and Reconnaissance (ISR) platforms. Secretary of Defense Robert M. Gates recently offered a stinging rebuke during an address to officers at the Air War College at Maxwell AFB Alabama in April 2008. In his remarks [2], he railed against the "old ways of doing business" that have hobbled efforts to put more ISR assets into the Iraqi and Afghan theaters. "We can do - and we should do - more to meet the needs of men and women fighting in the current conflicts while their outcome may still be in doubt." Secretary Gates followed up on this action by directing the standup of a UAS task force to address a very simple problem. That problem is that there are simply not enough ISR assets in the current theaters to support those campaigns. His injunction to this task force [3] was that it needed to "think outside the box about how to fast track more unmanned aerial vehicles and other ISR assets into the field." Its new charter was vastly instructive to his overall intent and included:

- Extending the operational limits of UAS's and other ISR platforms,

- Improving the efficiencies of training and test elements of ISR programs such as Predator pilot training,

- Ensuring the necessary bandwidth is provided to operate unmanned drones and other ISR assets, and

- Finding nontraditional capabilities of conventional aircraft like the targeting pods of F-16 Fighting Falcons to fill gaps in ISR coverage.

This recent undertaking inspires the central aim of this thesis which may be summarized as follows: it is to add a robust predictive capability to existing ISR platforms such that a remote observer can gain a richer decision-set to choose from during a given surveillance operation. It proposes that this goal ought to be achieved in real or near-real time, be generic to many of the ISR assets in the US inventory, and result in a specialized intelligence product that enables Effects-Based Operations.

Among the benefits of such an enhanced machine-human interface would be the ability to direct and divert high-value/high-demand assets in theater either for continued surveillance, interdiction, or other operations. It would, more generally, allow coalition forces to operate inside the enemy's proverbial "OODA Loop", especially when tracking a single, suspect vehicle. It would allow for sensor-cueing on the ground for greater battle-space awareness, and create target interdiction opportunities.

These concept hint at an even more powerful possibility: that the passive observer might cross the boundary to become an active participant. For if a predictive capability existed for a given vehicle with a well known trajectory, could not the observer know too the future effects of blocking a road, changing traffic light signals, staging a tactical team, or effecting some other perturbation of the system? This thesis proposes to take the Secretary of Defense at his word and, indeed, "think outside the box" with the view that the furtherance of these goals will have a *force multiplication effect* upon our ISR assets.

1.2 The Intelligence, Surveillance, and Reconnaissance Requirement

The first task for this thesis is to define the problem by decomposing it into its essential elements and to cloth it in a nomenclature. According to *Air Force Document 2-9 Intelligence, Surveillance, and Reconnaissance Operations*

> The goal of ISR operations is to provide accurate, relevant, and timely intelligence to decision makers. The Air Force best achieves this goal

> through effective employment of ISR capabilities, and by capitalizing on the interoperability existing among our ISR systems, as well as non-traditional sources, to create synergy through integration [4].

This reduces in practical terms to gathering spatio-temporal data from the battlespace, combining it with other available and relevant intelligence, and subjecting it to analysis to yield information which in turn enables superior decisioning. Again,

> Intelligence products must enable strategic, operational, and tactical users to visualize the operational environment systematically, spatially, and temporally, allowing them to orient themselves to the current and predicted situation to enable decisive action [4].

A further deconstruction might add that *intelligence* is the result of the integration, analysis, and interpretation of gathered data. By "gathered" we mean either by *surveillance*, which intends to persist over a single target in a sustained and systematic way, or by *reconnaissance*, which is more transitory and intends to discover an enemy's initial disposition.

If the first task was to embark upon definitions, then the second is to to prune the large spectrum of possibilities and choose a practical niche for our proposals. There is an intelligence requirement resulting in an Air Task Order (ATO) which results in a general navigation solution for the platform. This can vary widely dependent upon who the customer is, who or what the subject of observation is, and the spatio-temporal extent required. This results in an ISR platform being put into the air (or, if it has already been launched, reallocated for the mission), and it will view an area. We will refrain for the moment from stipulating that it is manned or unmanned, its sensor payload, its altitude, or any other unique characteristics. We will merely acknowledge that there is a broad taxonomy of platforms. Finally, there is an environment, which could vary from urban areas to desolate wildernesses. Tying all of these together, and of supreme importance to our subject, is the command and control element, again varying as to all of the previously enumerated items. We refer to the set containing the mission, the platform, the environment, and command and control as the *Mission Parameters*. The entire spectrum of all possible *Mission Parameters* for all possible contingencies is vast, though divisible into three general categories that correspond to the Three Levels of War. The first of these, is at the *strategic level* and includes high-altitude and space assets. The second, at the *operational level*, includes medium-altitude UAS's and multi-role platforms operated under the authority of a Joint Forces Commander (JFC), the Joint Force Air and Space Component Commander (JFACC), or the Commander of Air Force Forces (COMAFFOR). The third, at the *tactical level*, includes teleo-controlled air platforms that are deployed by the single infantry squad.

1.3 The Kill Chain

The careful follower of world events, especially of events pertaining to the Global War on Terror (GWoT), will have marked that the line between ISR activities and traditional combat operations has begun to blur. This may be attributed to the narrowing of the time-delay between information gathering, analysis, and exploitation for the purpose of shortening the proverbial *Kill Chain*. Defined as *find, fix, track, target, engage, assess*, it has generally been employed to understand the dynamics of target acquisition and prosecution. In the infancy of modern air power (specifically during World War II) the target acquisition portion of the *Kill Chain* might involve aerial reconnaissance, followed by days of photo-analysis, and the target prosecution portion would entail several squadrons of heavy bombers engaged in mass carpet-bombing, followed by more aerial reconnaissance for damage assessment. This could yield a *Kill Chain* with a duration of days to weeks.

In a nod to the verity that in today's world the object of a sortie will not always be to *kill*, a trend has emerged where the word "effects" is substituted for "kill". This modification allows for the inclusion of propaganda leaflets and electronic warfare in the operational tool-box. Though the expression *Effects Chain* will not be employed in this work, it is useful to bear in mind the motivation for its formulation. A maturation of this concept has been suggested by Rogers, et al. [5] that takes into account not only the occasionally non-lethal aspects of "effects", but also the cyber and electromagnetic domains. Named the *Modified Cyber Kill Chain*, this is expanded to mean *anticipate, interact, find, fix, target, track, engage, assess, anyway, anywhere, anytime* and coincides nicely with our stated goals. Moreover, its name is suggestive of the fact that its intent is to expand the battlespace to include *cooperative layered ISR* which combines ground and airborne sensor suites and algorithmic analysis to drive its anticipatory and interactive aspects. The practical results of this refinement take us beyond the ability of the World War II aircraft to carpet-bomb an entire municipality, or even a modern F-15E Strike Eagle to place an aimpoint on a target lit by a tactical laser. This allows interactions in the battlespace that include sensor cueing, ISR resource management, and also the pre-staging of tactical assets on the ground. The eyes, intellect, and implements of war, having formerly been separate, have begun to be tightly coupled and this in turn places an entirely new complexion on the *Mission Parameters*.

1.4 Defining the Mathematical and Representational Frameworks

In order to realize the stated goal of this study, we need to procure a *Mathematical Framework* to act as the underlying scaffolding for the enterprise. Also, it is necessary to express data that results from this calculus within a *Representational Framework* so that it may inform decisioning. Finally, an *Operational Framework* must be defined within which these proposals may be employed, consistent with *Mission Parameters* that can be expected in the GWoT. Let us first turn our attention to the computational considerations that will be incurred in the *Mathematical Framework*. It has been stated that real or near-real time video (irrespective of spectrum) and target tracking are assumed. A ready-made candidate for the mathematical requirement is Graph Theory. A graph is simply a collection of vertices connected by edges. This construct is used to model anything from wide area networks to air-traffic patterns, and comes with a rich set of well-understood algorithms for a variety of computational tasks. In this case, simple variant known as the *Directed Acyclic Graph* (DAG) will be employed. This is merely a collection of vertices connected by single-direction edges. Our tracklets exist in the mathematical space of DAGs where vertices are positions for a vehicle of interest, captured in each frame of the video, and the edges are *inferred* connectives.

Having achieved knowledge formulation mathematically, we must yet undertake knowledge representation. It is fortunate that an entire industry has sprung up around this problem. It is perhaps no coincidence that Geographic Information Systems (GIS) make extensive use of Graph Theory, together with relational databases, to depict spatio-temporal models at the micro, meso, and macro extent. These are employed for a diverse set of modeling requirements that extend over a geographic area and are useful for municipal planning, civil engineering, conservation, disaster-preparedness and many emerging military applications. The basic approach is to take GPS location measurements (usually differential corrected) for defining the vertices of a given structure, whether a road, a building, an underground conduit system, etc, and to represent these as tables of x, y, and z coordinates (corresponding to lattitude, longitude, and altitude respectively) in a database. These are then subjected to custom queries that render them as points, lines, and polygons in an OpenGL driven graphical user interface. Along with the x, y, and z coordinates are a host of relevant metadata that can be associated with one or more of these structures. Different constructs are then separated into layers and rendered on top of each other, yielding a context-rich, composite picture which can be subjected to ad hoc queries as the need arises.

GIS technology has matured to the point where it has begun to play an active role in intelligence activities. Those familiar with Human Intelligence (HUMINT), Measurements and Signals Intelligence (MASINT), Signals Intelligence (SIGINT), Imagery Intelligence (IMINT), and Open Source Intelligence (OSINT) might not be familiar with Geo-Spatial Intelligence (GEOINT) due to its relatively recent appearance on the stage. It was a natural consequence of American GPS, European Gallileo, and Russian GLONASS, that government and industry would begin to use this technology for municipal planning and management. This data has begun to accrete in large data-marts and to be more readily available. Combined with *ortho-rectified* satellite imagery for context, such products are now a common staple of map-finding and geographic search applications that can be had online. When properly combined with real-time ISR data, its object properties, such as the speed limits of roads, gradients of hills, land-use, and zoning, can be used to add context for the purpose of constraining future events.

1.5 Defining an Operational Framework

Now that *Mathematical and Representational Frameworks* have been defined, it must be decided what kinds of missions will be suitable candidates for these proposals. More succintly put, An *Operational Framework* must be defined. It has been suggested that the proposal should be generic to as many types of situations as possible, yet now we must add some constraints to that assertion. It is reasonable to begin with environments. These fall into two broad categories that roughly correspond to the two theaters of Iraq and Afghanistan. The former can be characterized as an urban setting with the infrastructure of a modern nation-state. The latter can be characterized as a rural setting whose infrastructure, modest even in peaceful times, exhibits all of the degradations of nearly three decades of war. ISR missions over these areas will be sufficiently different from each other that a solution for one might not necessarily be a solution for the other and so a choice must be made.

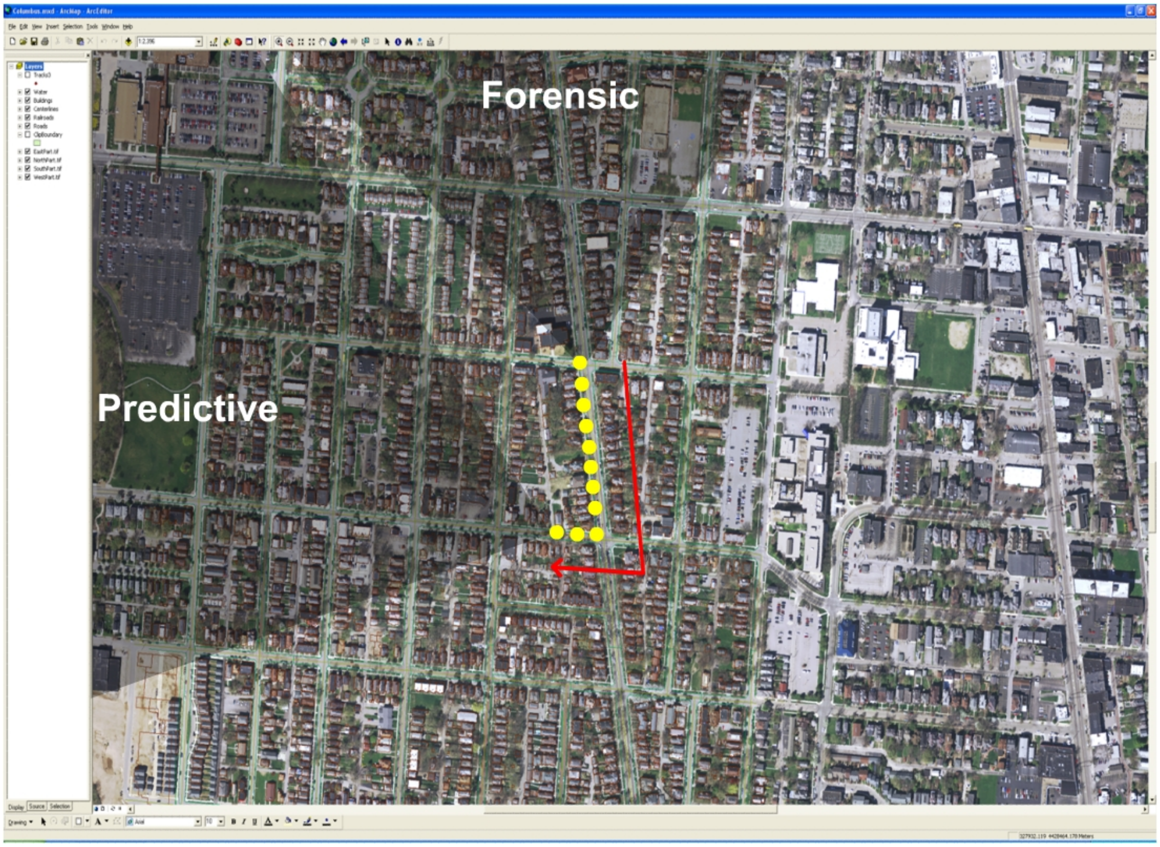
At the date of this writing (February, 2009) it would appear that *Operation Iraqi Freedom* is concluding successfully, and that *Operation Enduring Freedom* in Afghanistan stands to profit most from improvements. It therefore might surprise the reader that we will choose to exploit urban environments with this study. There are several reasons for this, and chief among these is that since world populations are migrating heavily to urban centers [6] it is reasonable to predict that the urban setting stands to be the primary battlefield of the 21st century. Also, the urban theater is more dynamic and the solution to such problems will invariably yield more information content. An example from Iraq: if a party of insurgents in a truck is spotted emerging from Fallujah, it will be of supreme importance to discover whether that vehicle is heading for a hidden weapons cache, for the staging area used by that group to launch terror attacks, or to a local safe-house. A counter-example for Afghanistan: if a party of Taliban fighters in a truck is spotted emerging from the Khyber Pass, it will surprise no one who is viewing the scene that the truck will deposit into the empty Dakka Plain and still be there an hour, or several hours, later. In Information Theory parlance, the former example contains more *information entropy*, which is a measure used to quantify the uncertainty of random variables. Ergo, answers to the question "where will they most likely be in twenty minutes" will have more information value in the former case. Our first constraint, then, will narrow our interest to urban and suburban environments.

1.5.1 Defining a Spatio-Temporal Niche . Until now the phrase *spatio-temporal* has been employed rather offhandedly, only because it is part of the ISR vernacular. However, our topic will benefit from a more exact definition of what it is that we mean by *space* and *time*. There necessarily exists a strong correlation between spatial extent and the level at which war is being waged. A very large macro-extent, say collected by a satellite, covers an entire geographic area and is more suitable for such tasks as *intelligence preparation of the battlespace* at the *strategic level*. A medium or meso-extent, say collected by an MQ-1 Predator, will cover an area consistent with a municipal area and is more suitable for operational and tactical operations. At the extreme end of this spectrum, at a micro-extent, are such platforms as the Army Raven which covers one city block and is used by single fire-teams in tactical situations. Our reliance upon the *Modified Cyber Kill Chain* and our determination to predict the future states of real and near-real time targets enables us to disqualify the satellites at the macro-extent which lacks the spatial fidelity (as well as temporal persistence) necessary for our goals. Although our proposal may admit the tactical level, we must be careful with the micro-extent because at that level dismounted troops are fighting small enemy groups in a drama that may conclude before even the most optimal predictive capabilities can be of use.

We will therefore be interested in the operational and tactical levels of war and this offers us our second constraint: spatially we are concerned with the meso-extent and admit the micro-extent when possible. This area will range from twenty city blocks to an entire municipal area.

Time can be deconstructed into past, present, and future. Dependent upon the requirement, different ISR missions and their associated platforms are concerned with different mixes of the three. For instance, the surveillance of a vehicle, carried out over the space of ninety seconds, would yield a collection of point measurements commonly referred to as a *tracklet* as in Figure 1.1. (This assumes a tracking capability) A mission primarily interested in forensics would seek to *project backwards* from this tracklet in order to decipher its past, as the silhouette pointing to the north illustrates. An example scenario for this might play out as follows: A vehicle suspected to have been involved with planting IEDs along a convoy route is spotted, tracked for a given time, but then the viewing asset is forced to leave the area. The resulting video could then be used by analysts to infer a past that would link that tracklet with the site of the planted IED. Now let us attempt the reverse of this scenario. This time, the airborn asset is more persistent and can afford to stay with the vehicle. The tracklet, representing the present, will grow with each frame of video and the observer might want to *project forward* in order to predict where the vehicle is going as the silhouette pointing to the west illustrates. Perhaps, for instance, the vehicle is on its way to plant an IED along a convoy route. The two scenarios bear an obvious symmetry because they are essentially the same. Note however that the former is less constrained by time and that the latter is time-critical and bears more on the *Modified Cyber Kill Chain*. This then suffices for our third constraint: temporally, we will only be interested in predicting the future during real or near-real time missions with the object of shortening the *Modified Cyber Kill Chain*.

1.5.2 Identification of Weapon Systems and CONOPS . It has been previously suggested that the mission niche for these proposals will include the operational


Columbus.mxd - ArcMap - ArcEditor
File Edit View Insert Selection Tools Window Help
1:2,396
Layers
Forensic
Predictive
Display Source Selection
Drawing
Arial

- ☐ Tracks3
- ☑ Water
- ☑ Buildings
- ☑ Centerlines
- ☑ Railroads
- ☑ Roads
- ☐ ClipBoundary
- ☑ EastPart.tif
- ☑ NorthPart.tif
- ☑ SouthPart.tif
- ☑ WestPart.tif

Figure 1.1: A vehicle tracklet in an urban environment with two analytic components. *Forensic* explores the *past*, and *Predictive* explores the *future*.

and tactical levels of warfare. A variety of platforms and sensor suites have been developed for these levels, along with communications and distribution architectures and receiver suites. A brief enumeration of these items will be useful later when outlining missions and CONOPS that correspond with our stated goals. In keeping with our stipulation of persistent, real or near-real time aerial surveillance, it is reasonable that the sensor suites that would be most useful would be electro-optical (EO), Synthetic Aperture Radar (SAR)[1], and Infra-Red (IR). The RQ-4A Global Hawk, MQ-1 Predator, and MQ-9 Reaper UAS systems contain all of these sensor suites. The JSTARS system, armed with a powerful SAR suite, is also a natural candidate. Targeting pods mounted on air superiority/dominance multi-role aircraft, including the LANTIRN pod and the SNIPER XR Advanced Targeting Pod are equipped with EO and IR and are also relevant. Finally, it has been proposed that low-flying, manned platforms with combinations of these sensor suites be provisioned to the theater, and these gain easy admittance.

The predictive capabilities proposed here will, by necessity, be general rather than specific. As a foreshadowing to the technical aspects of this work, it is useful to imagine the future state of a subject under surveillance as being represented by a bubble, superimposed on the GIS urban grid, and constricting around a predicted destination as the forecast becomes more accurate. If the subject has been under surveillance, then we may assume that tracking and targeting are already foregone conclusions. At any time, an aimpoint could be placed on the subject. The value from predicting where the target will be in the future, then, does not arise from the *target* and *engage* elements of the *Modified Cyber Kill Chain*. Rather, it involves the *anticipate* and *interact* elements. There are four capabilities that will be discussed that will be a consequence of the proposed *Predictive Battlespace Awareness*. These are:

- Sensor cueing

[1] At present, SAR is not capable of real or near-real time video. However, it shows promise in the near future of becomming so and we therefore include it in our list.

- ISR asset management and allocation
- Tactical pre-staging
- Event shaping

The first of these could mean, among a host of possibilities, that military, commercial, or municipal ground surveillance assets, are sequestered, made active, and monitored when a subject is predicted to enter their vicinity. Such ground sensors might include surveillance cameras, RFID monitors, GEO-Locators, Cell-phone triangulation equipment or any active or passive monitoring capabilities in the area. The second implies that another ISR asset may be vectored to the predicted area to assume surveillance of the subject. Its general intent speaks to one of the founding principles of this work, namely, of optimizing existing assets. The third could entail the insertion of a rapid reaction force, sent to ambush the target on the ground. This has the benefit of lethal discretion which could mitigate civilian casualties. It also allows for the physical detention of the target, which comes with the bonus of an additional intelligence yield, assuming effective interrogation. The fourth item speaks to the *perturbations of the system* alluded to earlier. Traffic lights in the predicted area could be controlled to shape where the target chooses to go. Roads could be barricaded and checkpoints set up. These actions could then be fed back into the system to further constrain the prediction such that, theoretically, the observer could become the controller.

1.5.3 A Bird's-Eye View of Ongoing Research. Finally, it is worth mentioning that the research and development to be described in this work do not exist in a vacuum, but are part of a greater research portfolio being conducted by Air Force Institute of Technology (AFIT) graduate students and sponsored by the Air Force Research Laboratories (AFRL) Sensors Directorate. This portfolio follows the general progression depicted in Figure 1.2 which relies upon the iterative refinements that operate on raw data, turning it into information, then into knowledge, and resulting in military capabilities. The incomplete list below will aid the reader in forming a broader context within which to place the current work.

- Studies performed last year demonstrated the possibilities of using scene context in GIS data-sets to aid in tracking a vehicle being surveilled from above [29]. The existence of a viable tracking capability is an essential assumption for the work to be described.
- Work is being conducted to exploit hyperspectral imagery such that feature extraction algorithms may be performed to discover roads, buildings, vegetation, and other features of interest [11]. This will allow GIS data-sets to be created from the air without friendly forces ever setting foot in a hostile environment.
- Work is also being conducted with hyperspectral imagery that allows feature extraction for the purpose of detecting human skin on the ground [12]. This will enable future efforts at dismount tracking capabilities where persons on the ground will not easily escape a persistent staring array in the sky.
- Research that coincides closely with this work is seeking to implement a *Reasoning Engine* to deduce semantics from a vehicular tracklet as it is surveilled from the air. This will create intelligence products in the form of early warning messages that may predict intentions and dispositions of enemies on the ground.
- Advanced Synthetic Aperture Radar (SAR) capabilities are being studied that might compensate for the native inabilities of this technology to resolve moving objects effectively [14]. The work is focusing on identifying and tracking *radar shadows* which are less variable and higher in fidelity and may lead to full SAR videography.

The kinship that these different research initiatives bear to one another is patent. They are all concerned with aerial sensing and most are associated with GIS technologies in one fashion or another. They rely upon graph-theoretic, probabilistic, and other artificial-intelligence/machine-learning disciplines with the object of enhancing the Layered Sensing construct for the purposes of greater Predictive Battle-Space Awareness.

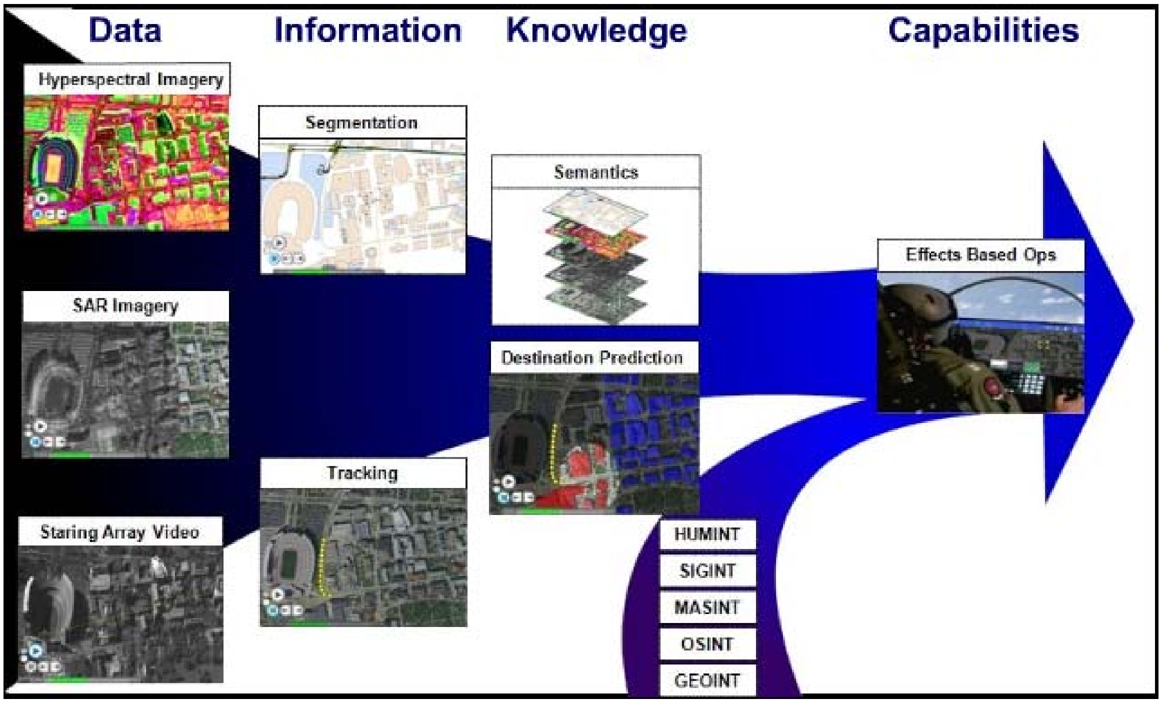


Figure 1.2: Ongoing research at The Air Force Research Laboratory Sensors Directorate focuses on a steady progression from *data*, *information*, *knowledge*, and *capabilities.*

1.6 Chapter I Summary

In the beginning of this chapter the goal for this thesis was stated as: to add a robust predictive capability to existing ISR platforms such that a remote observer can gain a richer decision-set to choose from during a given surveillance operation. It is now appropriate to nail down the specifics. The goal has been refined to include only urban or suburban environments. Strategic and extremely tactical platforms have been deemed inadmissible, in favor of medium-altitude, persistent aircraft equipped with EO, IR, or SAR. It has been limited to the meso-extent of twenty city blocks to an entire municipality, and constrained to be in real or near-real time in the expectation of shortening the *Modified Cyber Kill Chain*. It has stipulated the operational and tactical levels of war, which includes a spectrum beginning with the JFC, JFACC, or the COMAFOR in the Combined Air Operations Center (CAOC) and ending with the tactical operator on the ground. It has assumed a tracking capability organic to the software on the platform. It has proposed to use graph theory to model tracking data and that a GIS environment be used for representation. It has proposed to fuse the data taken from ISR with existing GEOINT to forge a predictive capability. It has suggested that this capability should exist within the *Layered Sensing* construct. Finally, it has predicted that this would be most feasible with the *Cloud Computing* business model. The ideas presented forthwith are intended to enlarge upon and augment the thinking that has come to dominate command and control at these levels with regards to ISR that specifies that a *Predictive Battle-Space Awareness* ought to

- Exceed human predictive capabilities
- Create early battlefield awareness
- Lead to rapid knowledge formulation

And most importantly, drive *Effects-Based Operations* [16]. To borrow from the nomenclature of the Prussian General Staff, it is the ambition of this thesis to add a bit of *fingerspitzengefuhl*, or fine finger-tip touch, to the machinery of battle-space management through the fusion of near or near-real time data from ISR platforms and GEOINT. It is ultimately hoped that a “synergy through integration” [4] will emerge that will have a force multiplication effect on ISR assets in theater.

II. Background

Having thoroughly described the problem space in Chapter I, the next order of business is to consider work that has been done in industry, the military, and academe. This chapter presents reviews of literature that have been published that describe approaches to our subject. It begins with the most general publications and ends with extremely specific approaches with the object of defining a baseline on which Chapter III elaborates on further. This baseline attempts to identify the *state-of-the-art* so that the approach developed throughout the rest of the thesis does not incur the reproach of having *re-invented the wheel.*

2.1 Representing Domain Knowledge for Traffic Modeling and Path Prediction

It is first appropriate to motivate the problem with an example of tracking entities in a spatio-temporal domain. The problem has received much attention from the dismount tracking perspective, and a brief exploration of the emerging methodology for this yields valuable insights into the major dynamics of the problem. In a seminal work on the subject [17], Makris and Ellis argue for a semantics generating engine that uses video frames of "captured activity" as input. Their proposed model is described as being "spatio-probabilistic" and is applied to pedestrian traffic areas. The spatial component relies on a topological background of vertices and edges, the former being defined as entry/exit zones, junctions, occlusion areas, likely stopping points, and areas where velocity vectors might change. (These also account for the natural bounds of the viewing area where a target will "appear" or "disappear".) The latter are defined as "paths", which are the edges that connect the vertices, or more succinctly, "roads", and "routes" which are the complete history of a "target" of interest and are aggregates consisting of one or more "paths".

The probabilistic component relies on Gaussian models which apply to the nodes of the graph. The researchers compared two clustering algorithms, k-Means and Expectation-Maximization to characterize entry/exit points of a scene with gaussian ellipses superimposed over the frames. Their research showed that the Expectation-Maximization algorithm provided a more exact model for a given point and also better filtered the inevitable noise from the data. It also demonstrated the feasibility of learning routes of behavior based on extended surveillance and assigning probabilities to newly acquired targets that they will enter/exit a given node. They point out that this could lead to identifying atypical behavior based on deviance from established norms learned by the model.

Next in support of this topic, the same authors expound upon the "learning" only hinted at in their first paper [18]. They observe that it is possible to implement learning with two competitive neural networks connected by a membrane of "leaky neurons". The first of these is proposed to "model the distribution of flow vectors", and the second to "model the trajectory distribution". The leaky neurons are intended to provide an element of hysteresis to the mechanism. As per their previous suggestions, this learning ability is intended to identify atypical behavior of a "target" based upon deviance from "learned" behaviors that are statistically common.

Makris and Ellis rely upon the same model and the same dichotomy developed in their previous paper cited above [17] consisting of "nodes", "paths", and "routes". While a target is acquired and monitored, its trajectory is recorded and matched to existing routes in the data-set. If it resembles an existing route below a predefined threshold, then it is attributed with that route tag. If not, then it causes a new route to be added to the data-set based upon its vectoral and scalar characteristics. (This assuming that it has endured the scruples of a preprocessing algorithm that attempts to eliminate noisy routes caused by indecisive vectors e.g., the target milling about). Additionally, the vertices corresponding to the newly added route are re-characterized by the addition of a weighting factor to a node variable that listens for such inputs. Hence, heavily travelled nodes increase in weight the more often they are traversed.

The result of this exercise is again a topological graph, though this one, dependent on the "learning period", discovers common routes taken by agents over a given topology. This ability allows for the possibility of distinguishing atypical behavior based on innovation that is detected in a given sequence of frames. Additionally, behavior that is classified as typical can benefit from a list of semantics for known routes, thereby yielding meaning to the interested observer. The only shortcoming of this approach, as the authors concede, is the dependence upon the “learning period”. This is a foreshadowing of the central difficulty associated with all of the probabilistic approaches considered during this research.

2.2 The Traditional Highway Management Approach to Traffic Modeling

This section considers traditional approaches for modeling traffic that are still extant in government and municipal planning and remains useful to our purposes for two reasons. First, it highlights the unwieldy nature of the traditional calculus which seeks to understand traffic modeling, which has always been poorly understood, in terms of fluid dynamics, which is better understood. However, the comparison of vehicular traffic flow to fluid dynamics is a flawed metaphor that breaks down at the discrete level. Second, it makes references to new techniques that we present in the next section. One of the traditional authorities on kinematic modeling of traffic deserves a brief description since it figures prominently in the final proposal.

The Highway Capacity Manual (National Research Council, Washington) [19] is an example of the traditional highway management approach and can inform our elementary dichotomy. It introduces the main parameters with which the traffic modeler needs to become conversant. The most elementary of these are *flow-rate, volume, speed, and density*, and they are expressed under two constraints, that of interrupted and uninterrupted traffic flow. The first of these two are closely related and are defined by the *Highway Capacity Manual* as:

Volume V: The total number of vehicles that pass over a given point or section of a lane or roadway during a given time interval; volumes can be expressed in terms of annual, daily, hourly, or sub-hourly periods.

Flow Rate FR: The equivalent hourly rate at which vehicles pass over a given point or a section of a lane or roadway during a given time interval of less than one hour, usually fifteen minutes.

The differences between V and FR, though slight, yields a useful relationship in the form of the *Peak-Hour Factor* (PHF) described as:

$$PHF = \frac{V/24\ hrs}{max(FR)} \tag{2.1}$$

Equation 2.1 is important when considering the capacity of a given road segment. If the road capacity is considered an ultimate threshold, then it can be compared to a PHF to predict congestion (though not catastrophic failure of the structural components of the road segment).

In addition to *Volume* and *Flow-Rate*, we are concerned with the cost, in terms of time, of traversing a road segment. A simple approach that employs the speed-limit SL and length ℓ of a road segment to define the cost Ω (in seconds) is defined as:

$$\Omega = (SL)^{-1}\ell \tag{2.2}$$

This could be regarded as a reasonably accurate measure if richer data did not exist. However, we prefer to employ simple statistics, if possible, to determine the cost Ω. A simple equation is provided by *HCM* that lends itself to our purpose [19]. *Space Mean Speed* (SMS) is often used to describe the (harmonic) mean speed of individual vehicles over a given length of road. This is given by:

$$SMS = \frac{i\ell}{\sum\limits_{i} r_i}, \tag{2.3}$$

where i is an observed vehicle, ℓ is the length of road segment traversed, r_i is the travel time of the i^{th} vehicle and SMS is measured in feet/second. As we prefer our units for Ω to be in seconds, this simple permutation suffices:

$$\Omega_{road} = (SMS)^{-1}\ell. \tag{2.4}$$

There are still other contributors to cost that have not yet been included. These may take a more various guise than many would expect, ranging from synchronized traffic lights, stop signs, yield signs, turn-prohibition signs, and lane restrictions. Additionally, there are school zones, railway crossings, and steep gradients. Moreover, the kinetic properties of the intersection includes a time penalty simply when the driver wishes to turn (especially left). To model these time-penalties in a purely deterministic way would, as before, be prohibitive. Instead, it suffices for our purposes to employ another statistic:

$$\Omega_{intersection} = \frac{\sum\limits_{i} r_i}{i}, \tag{2.5}$$

where the travel time r_i for i vehicles is averaged. Hence, we may define our simplified cost function as:

$$\Omega = \Omega_{road} + \Omega_{intersection}. \tag{2.6}$$

Finally, there is density. This parameter is vital to our model because it informs us as to flow rate, speed, headway between vehicles, and alludes to the general quality of traffic, e.g., the probability of vehicular accidents, delays, and traffic jams that can propagate along the length of a given roadway. This is given by:

$$D = (FR)(SMS) \tag{2.7}$$

where:

$$FR = \textit{Flow Rate} \text{ (vehicles/hour)}$$

$$SMS = \textit{Space Mean Speed} \text{ (miles/hour)}$$
$$D = \textit{Density} \text{ (vehicles/mile)}$$

2.3 Graph Theory

A graph is a pair of sets V and E such that the former represents vertices and E represents connectives between vertices [20]. The formal mathematical notation for this pair of sets is $G(V, E)$. An intuitive example for this construct might be cities as vertices V connected by roads as edges E. Another might be the hierarchical structure of a corporation where the CEO occupies the pinnacle and authority flows, pyramid like, to lower echelons in the organization. In this example, the personnel in the organization are the vertices V and the lines of authority are the edges E.

A key preoccupation of the graph theorist is the optimal traversal of a graph $G(V, E)$. The oft-cited metaphor for this is the *Travelling Salesman Problem* where there are a number of cities connected by roads and the salesman must devise an optimal route such that all cities are visited with the least amount of distance travelled. Here it is important to note the importance of *cost* which are numeric values attributed to edges E. *Cost* is the penalty for traversal of an edge between two vertices and can be denominated in distance, time, dollars, work, or a number of other examples. The (least costly) traversal of a graph is important for our purposes in order to facilitate a search of graph $G(V, E)$ for vertices of interest. It should therefore be unsurprising that a major focus of Graph Theory is search and this, in and of itself, is also a challenge to the scope of this work. However, a brief visitation of the concepts of search can facilitate the discussions in the remainder of this chapter and in Chapter III.

Search of a graph may be described by two broad categories, uninformed and Informed [21]. In each case, one starts at a root vertex V (say, the starting city for the traveling salesman) and begins a traversal of the graph such that neighboring nodes are interrogated. One form of *Uninformed Search* is *Depth-First Search* and stipulates that the search continue from the root vertex down the graph until a bottom extrema is reached where it then returns to the shallowest unexplored depth to begin down again. Usually recursive in execution, this approach iterates until the entire graph is covered. Note that with extremely deep (even infinitely deep) graphs, this search method can be unwieldy. Another, *Breadth-First Search*, is exactly converse in that it attempts to interrogate every vertex in adjacent levels before continuing to a deeper level. In each case, the *Cost* of edges E is never taken into account.

Informed Search, in contrast, employs *cost* considerations to decide upon the most judicious avenues of exploration and may also be guided by heuristics. The most simple example is a greedy search algorithm called *Best-First Search* that seeks the least costly combination of edges E to traverse to a goal. Russell and Norvig employ the evaluation function $f(n) = h(n)$ where $h(n)$ is the *estimated* cost from the start vertex to a destination vertex. This is a heuristic meant to anticipate the sum of shortest edges E between start and finish. A more sophisticated approach is A^* (pronounced A-Star) search algorithm, and employs evaluation function $f(n) = h(n) + g(n)$ where $g(n)$ is the path from the start node to a given node n, and $h(n)$ is the *estimated* cost from n to the goal. Our discussion in section 2.2, which discusses the cost of road-segments, is meant to facilitate this and similar functionalities.

In addition to search, there are more sophisticated traversals of graphs that yield more specialized results. One might want to know, for instance, a *Minimum Spanning Tree* in a graph. This would depict a subgraph $G'(V, E') \in G(V, E)$ where $G'(V, E')$ contains all the vertices $V \in G(V, E)$ but connects them with the minimum number of least costly edges $E' \subseteq E$. *Kruskal's Algorithm* is often employed for this. One might wish to know the *Shortest Path* between two nodes in a graph and *Dijkstra's Algorithm* is a popular approach to this requirement. Additionally, one might wish to know the optimal *Flow* that can propagate through a graph given the cost of its edges, and the *Max-Flow* algorithm is often employed to do this. We explore the latter two of these algorithms next as they bear heavily in Chapter III when we will be faced with a very large graph $G(V, E)$ and will have to make judicious traversals in order to obtain specialized results.

```
$Function\ Dijkstra$
    $S \leftarrow \phi$
    $Q \leftarrow\ v \in G(V, E)$
    $While\ Q \neq \phi$
        $do\ u \leftarrow\ Min(Q)$
            $S \leftarrow S \cup u$
            $for\ each\ v\ adjacent\ to\ u$
                $Relax$
```

Figure 2.1: *Dijkstra's Search* finds the least-costly path between two points in a graph $G(V, E)$.

2.3.1 Dijkstra . An excellent source for algorithms of this sort is *Introduction to Algorithms* by Thomas Cormen [30], et al. It describes *Dijkstra's Algorithm* which attempts to find the shortest path between two points. It employs a directed acyclic graph $G = (V, E)$ where V equals the set of vertices of the graph (for our purposes, intersections, sources and sinks of vehicular traffic, and the graphical extrema of our data-set) and E equals the set of edges of the graph. (For our purposes, the roads) It requires that all edges $e \in E$ have non-negative weights and that there exist two vertices v_s and v_t, that will be the source and terminal nodes, respectively, between which it would be desireable to determine the shortest path. This will seem to be a combination of *Breadth-First Search* and *A**. Its essential form [30], can be described by the pseudocode in Figure 2.1, where an empty or null set is denoted by ϕ, $\exists\ n\ e\ \in E$ and $\exists\ n + 1\ v\ \in V$

Two lists S, Q (in the form of minimum priority queues) are maintained while a traversal is performed from a starting node v_s for graph $G(V, E)$. Each vertex $v \in V$ is equipped with an initial estimated cost $d(v)$ to the goal vertex v_t. Q is populated with all $v \in V$ and ordered according to this value.

Beginning with v_s, the traversal interrogates each vertex, one level at a time, and pops the one with the lowest cost from Q and places it in S. Then a new cost for that v_n is calculated as the cost from $v_s - v_n$, denoted as $\pi(v)$, plus a new estimated cost heuristic from $v_n - v_t$ denoted as $d(v)$. This operation is described as "relaxing" the edges and elicits an image of a cloud starting at v_s and engulfing all vertices $v \in V$ until they have all been interrogated. At this point, list Q is empty and list S has been populated with an ordered list of vertices v such that vertices from $v_s - v_t$ constitute the shortest path.

2.3.2 Ford-Fulkerson Max-Flow . Another prominent algorithmic traversal in graph theory is the so called Maximum-Flow problem. As with Dijkstra, there are two nodes $v_s, v_t \in V$ for graph $G(V, E)$. Like before, $G(V, E)$ represents roads and intersections. In this case however, v_s is a source of flow (imagine commuters attempting to travel a road network during rush-hour), and v_t is a sink for that flow. Any discrete flow elements may enter the network through v_s and use any combination of directed edges through other vertices $v \in V$ in order to obtain the sink v_t. In this case, edges have a flow capacity C, rather than a cost, that may not be exceeded. The maximum-flow problem then is to find the maximum flow through a flow-network given these constraints.

One approach to the Max-Flow problem is the *Ford-Fulkerson* algorithm [30], and it can be described by the pseudocode in Figure 2.2, where each edge $e \in E$ is described by its respective vertex ordered pairs (u, v) and (v, u).

The essential idea is to discover a path through network $G(V, E)$ from source v_s to sink v_t that admits the most capacity c. This path is then subtracted from $G(V, E)$ to yield $G(V, E)_f$ which is a *residual* graph. Likewise, the capacity of this path p is subtracted from c to yield a residual capacity c_f.

Additional attempts are made on $G(V, E)_f$ until there are no more possible paths p. A brief explanation of the pseudo-code is as follows: during the first for-loop, all edges $e \in E$ are initialized to have zero flow, though they each have an integer capacity that has been previously assigned. Then, the while loop greedily searches for a path p from v_s to v_t such that it passes through the edges with some remaining capacity in residual graph G_f. If another path p is found, its capacity is subtracted

```
$Function\ MaxFlow$
    $For\ each\ edge\ (u, v) \in E$
        $Do$
            $f(u, v) \leftarrow 0$
            $f(v, u) \leftarrow 0$
            $G(V, E)_f \leftarrow G$
    $While\ \exists\ a\ path\ p\ from\ v_s\ to\ v_t\ in\ the\ residual\ network\ G_f$
        $Do$
            $c_f(p) \leftarrow min(c_f(u, v) : (u, v) \in p)$
            $For(each\ edge(u, v) \in p)$
              $f(u, v) \leftarrow f(u, v) + c_f(p)$
              $f(v, u) \leftarrow -f(u, v)$
```

Figure 2.2: *Max Flow* is a min-cut algorithm that searches the space in order to allow the most "flow" to travel between a source and sink nodes in $G(V, E)$.

from the flow $f(v, u)$ in the residual graph G_f and a flow capacity is added *in the other direction*. Hence, if a flow between two points was five units, and four units of flow were placed on the edge, then there would be a negative flow of four in the opposite direction and a positive flow of one in the original direction. This zero-sum approach allows the algorithm to search for paths until a bottleneck (known as the min-cut) is found, at which point it must stop. The flow at the min-cut will then be equal to the maximum flow through the network.

2.4 Temporal Statistical Models

If one considers the various decision points $v \in V$ in the graph $G(V, E)$ described for our paradigm, one cannot help reasoning that there exists a conditional dependence between different vertices in V. This is easy to illustrate. Imagine that there exist two major routes taken by commuters from a residential area to an industrial complex. One is a major highway and the other is a less known rural route. It is reasonable to expect that once one is on the highway, he will not divert to the rural route halfway between the residential area and the industrial complex. The same conjecture may be made for those who prefer the rural route. This implies that a vertex in a given path bear a statistical affinity for certain vertices and not for others. If that is so, it may be further conjectured that probabilistic inferences can be made as to which vertices an agent will visit based on vertices it has already been to, provided that those vertices have been attributed with conditional probability information. This observation demands that temporal statistical modeling be investigated more fully.

2.4.1 Markov Chains . The Markov Chain is the most simple spatio-temporal probabilistic model and is, essentially, a state machine [21]. It is represented by a temporal graph $G(V, E)$ where n vertices $v \in V$ represent n discrete states of a system and edges $e \in E$ connecting these states are transitional probabilities. The central idea behind Markov Chains is known formally as the *Markov assumption* which states that the probability of transition from state v_{t-1} to state v_t relies only upon v_{t-1} and upon no *a priori* information from earlier predecessors. This is expressed mathematically as $P(v_t|v_{0:t-1}) = P(v_t|v_{t-1})$. Just as adjacency matrices are used to depict connecting edges in graphs, so too, an $n \times n$ matrix A is used to describe the probabilities of state transition represented by V. Hence, each state is represented by a row in A where each row is a cumulative distribution function (CDF) that sums to one. (Bear in mind that a state may transition to itself.) The power of this concept resides in the fact that one may express the probability that $P(v_t|v_{t-1}) = A_{ij}$ where A_{ij} is the i^{th} probability of transition for the j^{th} state. (see Figure 2.3).

2.4.2 Hidden Markov Models . Hidden Markov Models build upon Markov Chains but with a further complication: the actual state is hidden [22]. Instead of referring to a state $v \in V$, one must presume that the states are impervious to direct observation or at least very obscure. Without directly sensing the states, we may directly observe one of the observations where there are m different possible observations associated with all states. This is denoted as $o_m \in \mathbf{O}$, $\mathbf{O}$ is a vector of many observations. It must be emphasized here that the set must be such that they could be observed, though not with equal probability, for each state. This added feature necessitates a second matrix of probabilities B, this one $n \times m$, where there

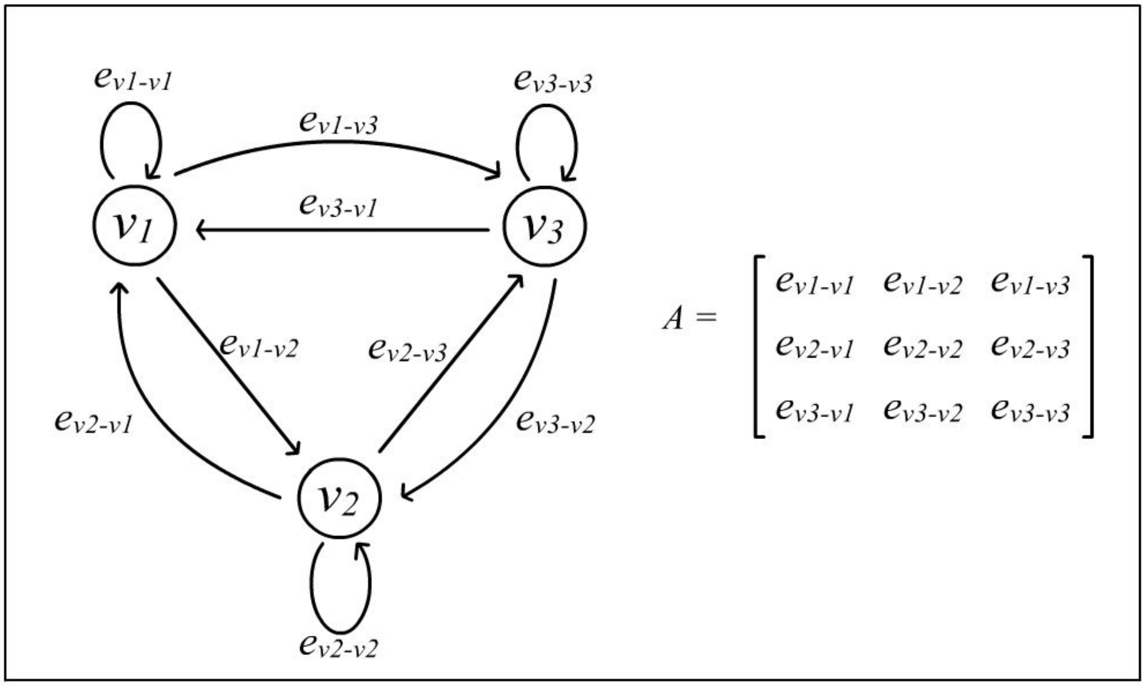
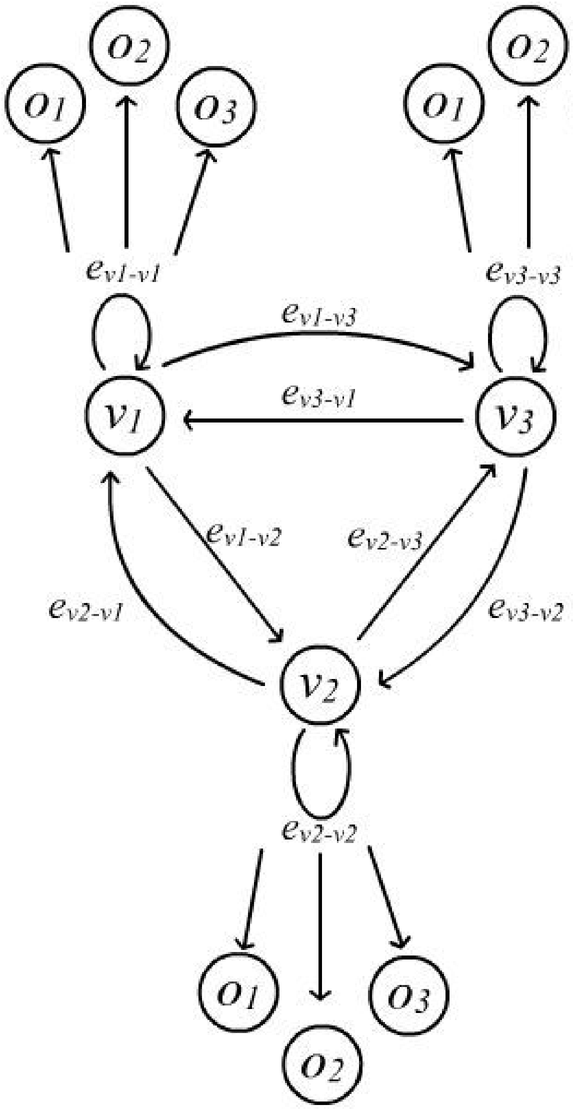

$e_{v1\text{-}v1}$
$e_{v3\text{-}v3}$
$e_{v1\text{-}v3}$
$e_{v3\text{-}v1}$
v_1
v_3
$e_{v1\text{-}v2}$
$e_{v2\text{-}v3}$
$e_{v2\text{-}v1}$
$e_{v3\text{-}v2}$
v_2
$e_{v2\text{-}v2}$
$A = \begin{bmatrix} e_{v1\text{-}v1} & e_{v1\text{-}v2} & e_{v1\text{-}v3} \\ e_{v2\text{-}v1} & e_{v2\text{-}v2} & e_{v2\text{-}v3} \\ e_{v3\text{-}v1} & e_{v3\text{-}v2} & e_{v3\text{-}v3} \end{bmatrix}$

Figure 2.3: A simple Markov Chain with three states.

are n rows of m length cumulative distribution functions, one probability for each observation *given the state* per $P(o_m|v_n)$ Finally, there is a vector π_n that accounts for all of the probabilities that the initial state will be the n^{th} state. We can bundle A, B, π as simply the model, Θ. See Figure 2.4.

A common example of such a situation [21] is the allegory of the man in the vault. The man never emerges from the vault and may not know the weather at any given day. The weather conditions are hidden states that can be either snowy, rainy, or sunny. The man has coworkers who come in from the outside and bring with them evidence of the hidden weather in their attire. Coats are worn on snowy days, umbrellas are brought on rainy days, and shorts are worn on sunny days. By observing these indications, the man might consult his matrix of conditional probabilities B and speculate with what certainty the state of the system.

Lawrence Rabiner enumerates three different problems that can be solved by Hidden Markov Models:



$$A = \begin{bmatrix} e_{v1\text{-}v1} & e_{v1\text{-}v2} & e_{v1\text{-}v3} \\ e_{v2\text{-}v1} & e_{v2\text{-}v2} & e_{v2\text{-}v3} \\ e_{v3\text{-}v1} & e_{v3\text{-}v2} & e_{v3\text{-}v3} \end{bmatrix}$$

$$B = \begin{bmatrix} P(o_1\backslash v_1) & P(o_2\backslash v_1) & P(o_3\backslash v_1) \\ P(o_1\backslash v_2) & P(o_2\backslash v_2) & P(o_3\backslash v_2) \\ P(o_1\backslash v_3) & P(o_2\backslash v_3) & P(o_3\backslash v_3) \end{bmatrix}$$

$$\boldsymbol{\pi} = \begin{bmatrix} P(v_1) & P(v_1) & P(v_1) \end{bmatrix}$$

Figure 2.4: A Hidden Markov Model with three states and three observations.

- Given a sequence of observations $\mathbf{O} = o_1, o_2, \ldots, o_t$, and the model θ, how can one decide the probability of the observation sequence $\mathbf{O}$?

- Given the sequence of observations $\mathbf{O} = o_1, o_2, \ldots, o_t$, and the model θ, how can we choose a corresponding state sequence $v_n = v_1, v_2, \ldots, v_t$ which optimally fits the evidence of $\mathbf{O}$?

- How do we adjust the model parameters Θ to maximize $P(\mathbf{O}\ |\Theta)$?

Of these, the most relevant to the discussion is the first problem. This one can be described by Equation (2.8):

$$P(\mathbf{O}|\Theta) = \sum_{all\ v_n} \pi_{v_1}[B_{v_1}(o_1)][A_{v_1 v_2}B_{v_2}(o_2)]\ldots[A_{v_{t-1}v_t}B_{v_t}(O_t)] \tag{2.8}$$

Where π_{v_1} is the probability of a particular state in vector π being the first state, and $B_{v_1}(o_1)$ is the probability of the first observation for its state from B, and $A_{v_1 v_2}B_{v_2}(o_2)$ is the probability from A of transition to the next state times the probability of the next observation from B. In practical terms this breaks down as multiplying the initial probability of the first state times all of the probabilities from B for the observations, times *all the permutations of all of the state transitions from A*. This actually tells us the probability of a given sequence of observations $\mathbf{O}$ given the model Θ (It is actually computationally unfeasible if done per Equation (2.1) and is made feasible by a technique known as *Forward-Backward Chaining*.

2.4.3 Kalman Filters . Next we consider an elaboration of the HMM's that allows for greater accuracies in the case of uncertain measurements. The Kalman Filter differs from HMM's in that, whereas the former is discretized, the latter is continuous [23]. The observational state spans over a Gaussian function, instead of there being a set of observations associated with hidden states. Moreover, the variance σ^2 of this Gaussian function depends upon the faith in that observation and its mean μ reflects the "best guess" of the actual state.

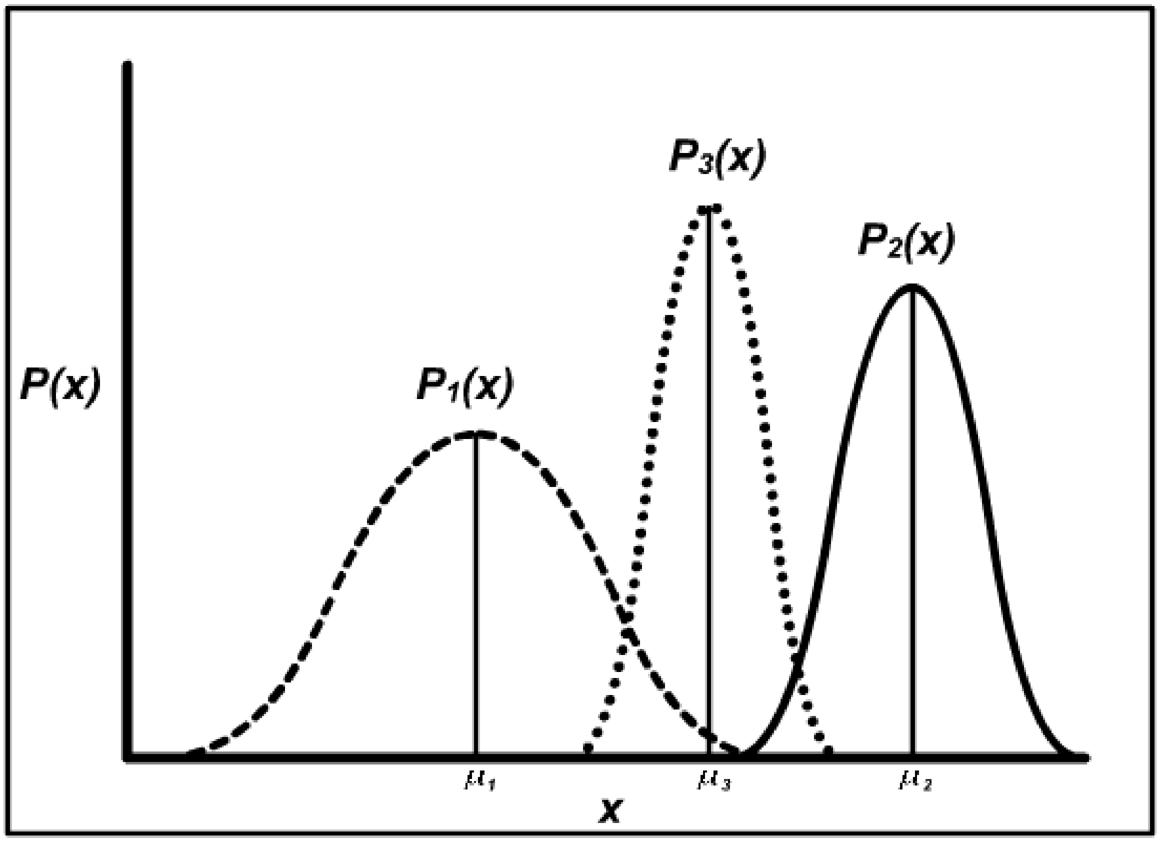


Figure 2.5: A Kalman Filter example where successive measurements from P_1 to P_3 enjoy improved confidence. Note the reduced variance from P_1 to P_3.

A commonly cited example (Maybeck) [23] is of mariners on a boat trying to calculate their distance with a seventeenth century sextant, compass, and the north star. An amateur takes a reading with his equipment in order to estimate their x position at $t = 1$. His measurement x_1 is not thought to correspond very well with their actual position because of the individual's inexperience, so its corresponding model $P_1(x)$ is assigned a large variance σ_1^2. Later, at $t = 2$, a more experienced seaman takes a measurement x_2 which is assigned a much smaller variance σ_2^2 for model $P_2(x)$. Rather than discarding the inexperienced seaman's reading, a third model $P_3(x)$ is created from each of the previous models though the contributions of the more accurate measurement are given more weight. The process continues with newer measurements and the model is subsequently updated per Equation (2.9) and Equation (2.10). See Figure 2.5.

$$\mu_{t+1} = \frac{(\sigma_t^2+\sigma_x^2)z_{t+1}+\sigma_z^2\mu_t}{\sigma_t^2+\sigma_x^2+\sigma_z^2} \tag{2.9}$$

$$\sigma_{t+1}^2 = \frac{(\sigma_t^2+\sigma_x^2)\sigma_z^2}{\sigma_t^2+\sigma_x^2+\sigma_z^2} \tag{2.10}$$

where σ_t^2 is the old variance, σ_z^2 is the variance of the new measurement, and σ_x^2 is the actual variance of the system. The system itself, just like the observations, can have a large variance, meaning that it is highly dynamic. An application for this for our vehicular path prediction could be, again, our cardinal headings, again with our ordered pair (x, y) representing the position of the target and ordered pair (v_x, v_y) representing the velocity of the target. Confidence in the successive measurements could be fed iteratively into a Kalman filter based upon feedback from the system or an outside source to ultimately shrink the variance (and hence the uncertainty) of the measurement.

2.5 A Review of Current Destination Prediction and Traffic Modeling Methodologies

The previous discussion arms us with the methodology for representing and analyzing the essential dynamics of our system. The next work considered by this research provides some of the missing pieces from the explanation above and drives the rest of the effort described in this thesis [24]. It proposes two methods of path prediction for a moving vehicle which could be combined for even better fidelity. The first method suggests representing the environment as a directed graph $G(V, E)$ (it also assumes the existence of a tracking system for the vehicle under surveillance). It attaches a value to the edges (which in our case are roads) based upon the cost of traversal and stipulates decision points at given intervals as nodes (in our case, traffic intersections, and other urban ingress and egress points). It then generates a set of possible paths within $G(V, E)$ that the vehicle can take. A time-horizon is defined as ΔT that constrains the paths according to time, though a distance-horizon could be employed instead. They propose an algorithm that prunes the paths from the vehicle's current position to a remote position if there are other paths from a previous position that would have been less costly.

For instance, an observed tracklet P of a subject vehicle is defined as a vector consisting of all of the measured positions of the subject as it travels in $G(V, E)$. From each observed position in P a set of n possible paths P_n can be generated according to the layout of $G(V, E)$. These will necessarily travel in all directions from that position and reach all of the possible destinations within time-horizon ΔT. A comparitor algorithm can then weigh all paths P_n such that different paths to the same destination are compared by their additive cost. If a vehicle travels from A to B, and a possible future destination is C, the cost $A - C$ will be compared to the cost $B - C$. If $A - C$ is found to be less costly then $B - C$, then destination C, along with path P_{B-C} are pruned from the problem space. The assumption of optimal driving demands that, had the subject intended to go to C, he would have proceeded from A. A practical motivation for this might be the case where position A leads to an expressway, and position B leads to an alley. This can be run iteratively as new positions p are added to P as the tracking continues. This uses a search heuristic that assumes driver behavior shaped by economy of time. We might easily enhance this search heuristic by adding a *Ford-Fulkerson Max-Flow* algorithm to predict traffic flow rates [30].

The second method that the authors propose is to use Markov Chains to predict where the vehicle may go. They stipulate a set of n states $v_n \in V$ where each state is a tuple of attributes known to the tracker. This tuple actually behaves like a *hyper-variable* and allows for many different forms of measurement to inform the model. These might be purely physical such as speed and velocity, or more behavioral such as aggressiveness, carelessness, etc. (the latter seem to require a great deal of judgmental interaction with the model). The important thing is that there is a discrete set of states defined by these tuples corresponding to possible measurements of the subject. Then, given a state v_t, at time $T = t$ and a matrix A describing transitional probabilities between states, the probability $P(v_{t+1}|v_t)$ will be found in the $(v_t)^{th}$ row and $(v_{t+1})^{th}$ column of A.

2.5.1 Dynamic Tactical Targeting . The *Dynamic Tactical Targeting* (DTT) work carried out by the Air Force Research Laboratory [25] is regarded as one of the most advanced efforts in destination prediction technology[1]. This milestone is important because it is the intent of this thesis to begin where others have left off in order to be certain that the work does not reinvent the wheel. The DTT effort, sadly, never attracted funding though it may live again in the guise of the present work which was influenced by its general approach. In addition to providing a robust tracking component for aerial targeting purposes, DTT employs primitive applications of the principles discussed above to location prediction. It accomplished this by defining a motion model representative of the top speed of the tracked subject vehicle

[1] Personal conversations with Mr. Douglas Abernathy of Lockheed Martin confirmed that DTT represented the latest research performed by the Air Force in vehicle destination prediction

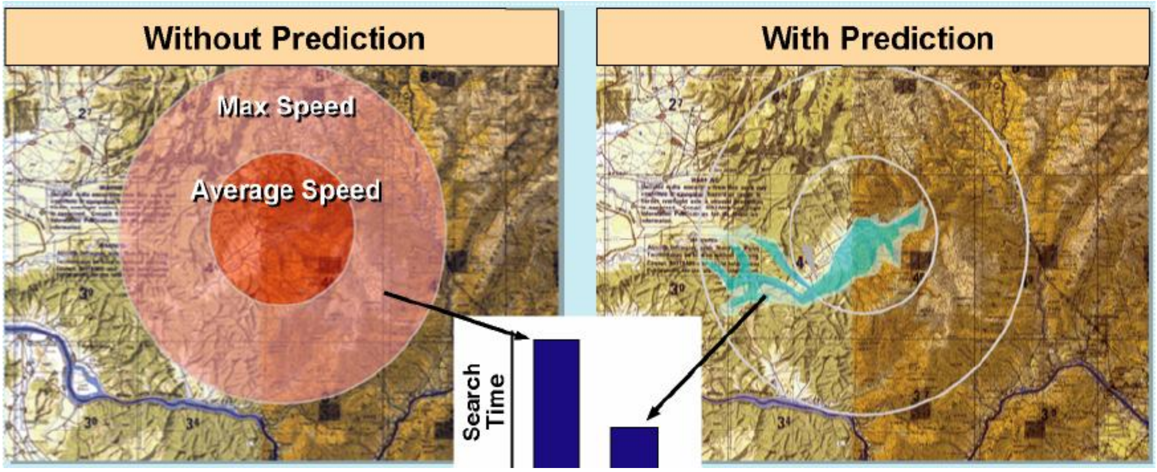


Figure 2.6: A Dynamic Tactical Targeting predictive envelope.

superimposed over a GIS terrain. Per Figure 2.6. The motion model assumes 360 degrees of freedom for the vehicle and equal resistance in all cardinal directions. Within this model is nested a second, smaller model representative of the average measured speed of the vehicle. The subject may be said to be bounded, within a given time horizon, by the annular region created by the outer and inner models.

Within this domain, an envelope is allowed to constrict around a smaller area based on *Dijkstra's Shortest Path* algorithm (section 2.3.1) carried out on a *Terrain Map* that overlays the area. Note from Figure 2.6 that this area is essentially of the first of the two types of environments described in Chapter I, or of a rural character. The *Terrain Map* defines the search-space as a pixel-grid whose weighted values represent "difficulty of movement". These measurements are taken with a "penetrometer", literally a cone-shaped device with graduated markings that is pushed into the earth at regular intervals. The amount of penetration corresponds to the "softness" of the earth, and hence to the "difficulty of movement". Areas such that are immune from this kind of measurement, such as bodies of water, forests, and other intractable areas are given higher values while roads and improved surfaces are given a low resistance values. The resultant *Terrain Map* is used with the real-time tracklet observables to inform *Dijkstra's Shortest Path* in defining the actual area where the vehicle may be reasonably expected to be within a given time horizon. While innovative, this model suffers from under-exploitation of the enormous data-sets available on urban environments. The exclusion of areas described as "municipal" in Section 1.5, is glaring in light of the richness of existing data and also the recent escalation in urban warfare [6]. This realization points the way to a great virgin territory upon which the efforts of this thesis may be applied.

2.5.2 Context and Fuzzy Representations . Another approach that has been made recently to this problem was written initially to address several perceived shortcomings with Data Fusion [26]. Its authors, Richard T. Antony of Fortune 500 technology company SAIC and Joseph A. Karakowski of the US Army Communications-Electronics Research, Development, and Engineering Center (CERDE) argue that the *Data Fusion* effort suffers from analytic shortfalls due to the failure to account for context inherent in rich data-sets. Their definition of context bears repeating in its entirety:

> "Context is considered to be any knowledge that potentially enhances the robustness of the objective products but that is not explicitly supplied as input."

The problem as described thus far admits for an observed signal (the tracklet) and a rich data-set (the GeoDatabase) but has neglected the potential of examining the layers of information *in respect to their adjacent data layers*. Mr. Antony and Mr. Karakowski argue that the *a priori* knowledge (in our case the tracklet and the GeoDatabase) can be combined to yield an *a posteriori* sum that is greater than its parts. They employ a construct dubbed *Context and Fusion Support Services* to offer four GeoProcessing services. These are: Search, Clustering, Correlation, and (Fuzzy) Set Operations.

The essential idea is to increase the domain-space dimensionality by adding context to spatio-temporal objects. According to the authors, context may be employed as constraints that prune the solution space of the problem, or as actual features represented in the knowledge domain. The latter of these two can assume the character of "fuzzy" modifiers whereby semantic information such as "near", "very near",

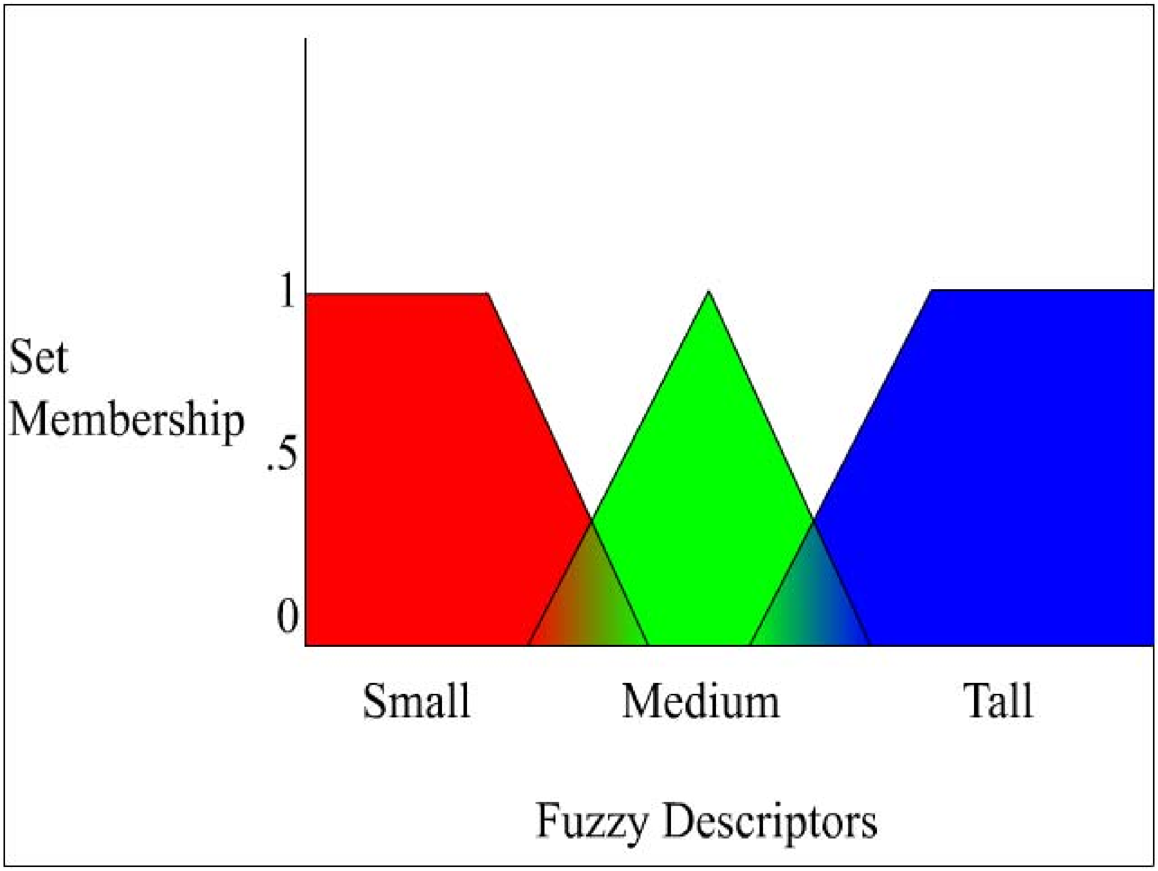


Figure 2.7: The union and intersection of "small", "medium", and "tall" using Zadeh's model [26].

and "coincident" can be encoded geospatially. Hence, the vagaries of human speech (especially of adjectives) may be captured by *Fuzzy Sets* and represented in a meaningful way. A graphical description first employed by L. A. Zedeh who introduced the notion in 1965 [27] is depicted in Figure 2.7. Here the uncertain nature of "short", "medium", and "tall" are shown as overlapping continua with intersecting regions. If each proposition is understood to have a range of 0-1, then even someone regarded as "short" might be able to claim a certain degree of "tall-ness", say a value of .02. Such a construct may be realized in spatial terms as shown in Figure 2.8 where a building may be surrounded by polygons that denote "near", "very near", and "extremely near". Note however, that unlike in Figure 2.7, there is no smooth continuum, but rather a step function that transitions from one set to the next.

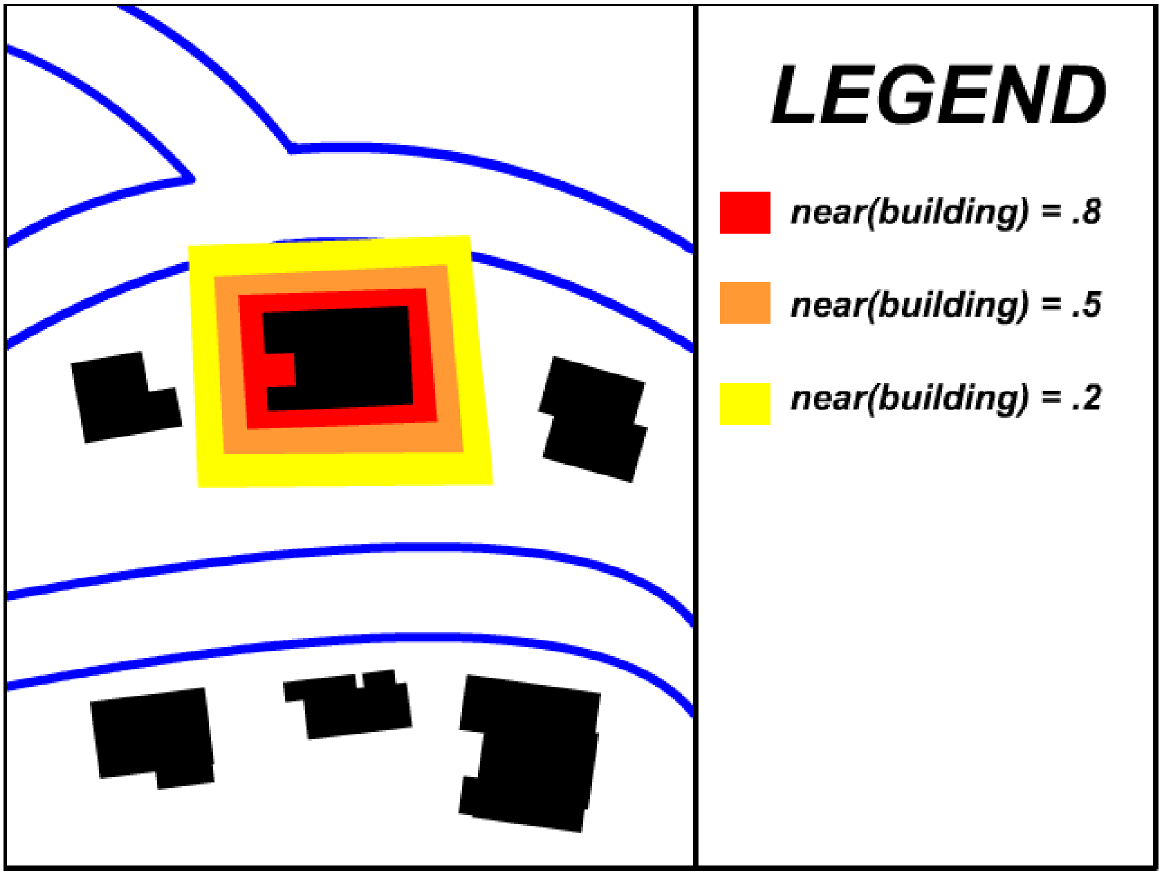


Figure 2.8: Fuzzy sets as represented as spatial features in a GeoDatabase.

Such a construct allows for the inclusion of HUMINT and SIGINT into our model if aided by *Natural Language Processing* such that GeoSpatial operations may be performed on *intercepted messages* as well as tracklets. What is more, this approach would enable strong correlation between contextual fuzzy sets and "crisp" non-fuzzy modifiers such as "in the parkway", "in the building", and "on the highway". Moreover, the *union, junction, and disjoint junction* of two or more "fuzzy" or "crisp" regions could yield semantics by calculation. Suppose that there are three regions representing three propositions that overlap and that the first two are "fuzzy", and the last is "crisp":

- A $= near(Road\ X) =\ .6$
- B $= near(Building\ Y) =\ .8$
- C $= in(Suburb\ Z) =\ True$

The area defined by $D = A \cap B \cap C$ would yield the semantic $D = near(Road\ X \wedge Building\ Y) = .7\ \wedge\ in(Suburb\ Z)$ where the "fuzzy" figure is the mean of the two "fuzzy" operands. This semantic enables primitive "understanding" at the machine level. Where before a possible tracklet would have yielded little without the eyes and direct attention of a live analyst, an event-listener can now warn of possible consequences and, more importantly, triage events of high interest to the analyst.

2.5.3 Microsoft Clearflow . The GeoProcessing services business model has recently surmounted the bounds of academic research and conferences and hit the marketplace. Microsoft ClearFlow technology [28], bundled with other map services as part of maps.live.com, actually allows web-enabled devices to interrogate a given urban extent in order to predict future traffic jams. In particular, the JamBayes service relies upon Gaussian mixture models that are trained with an exhaustive technique: drivers are hired to drive thousands of times throughout a given municipality with GPS logging devices. The models are then subjected to a Bayesian reasoning process that attempts to predict a "personality" for a road segment given the time of day.

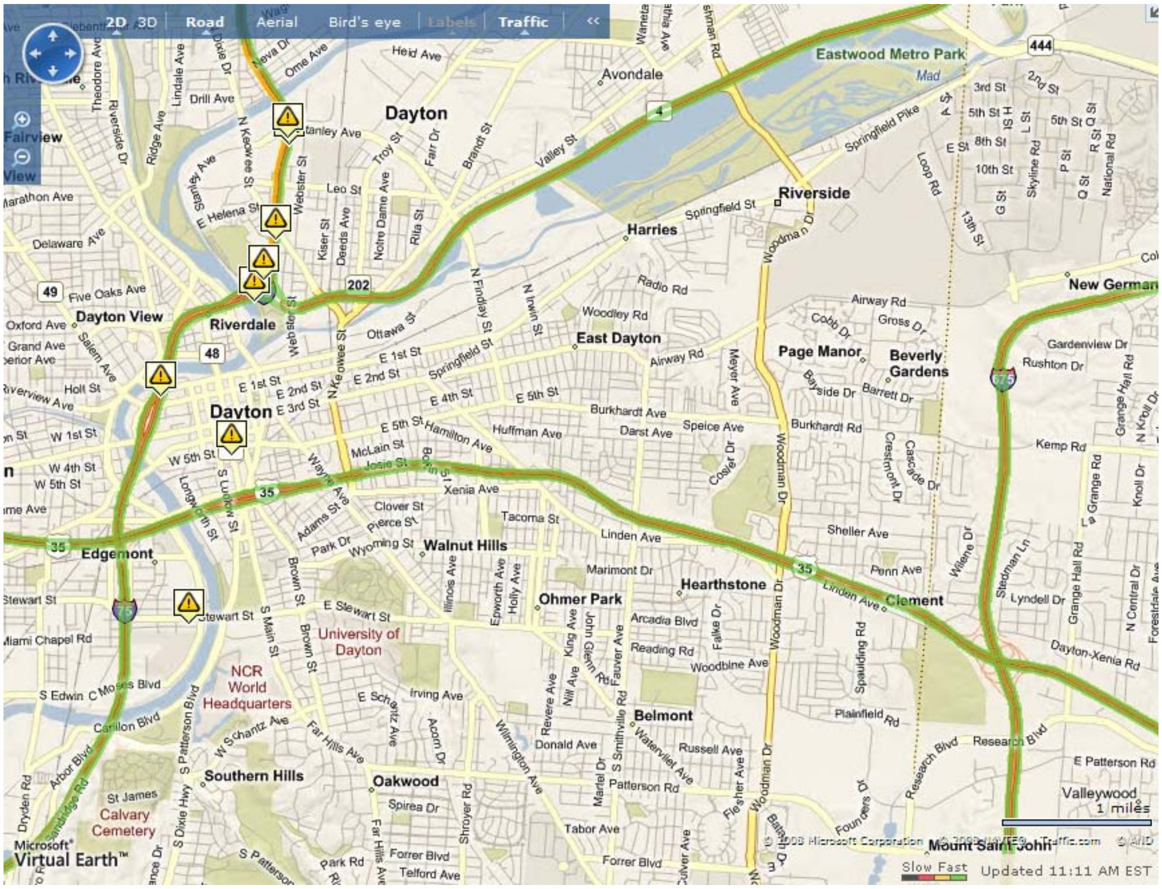


Figure 2.9: A screenshot of Microsoft maps.live.com with traffic services over the Dayton Ohio area.

Other *a priori* arguments passed to the system are weather, major sporting events, and other possibly disruptive phenomena.

The result, as Figure 2.9 demonstrates, is a map showing the roads in a given extent that are color-coded according to the prediction of traffic. Note that the legend in the lower right hand corner of Figure 2.9 depicts a linear gradient that depict traffic snarls that ranges from *green* (fast) to *amber* (slow), *red* (slower), and *black* (jammed). Note the exclamation point icons distributed around the arteries that denote municipal events and other context that could aid the users' decisions about route planning. These features correspond to the Mr. Antony and Mr. Karakowski's contextual "constraint" functionality.

2.5.4 A Contemporary Approach from the Air Force Institute of Technology. Contemporary work has been done in the area of tracking and prediction that avails itself admirably of GIS technology, and it deserves to be the capstone topic for this Chapter. Scott Pierce proposed in his 2008 AFIT thesis that GIS context could be used to filter false-alarms encountered while tracking a vehicle from the air [29].

The GIS context comes from two sources which are added to each other to form a *Probability of Detection Map* (PD-Map). The first would come from simple polygon representations of the road network of an area of interest. This is generated from a road-centerline GIS file where a buffer is created around the lines to approximate the 2D width of the roads. Figure 2.10 is the result.

The second, known as *Occlusion Masks*, were generated from known 3D height information of the buildings in an area of interest. A ray-tracing operation would be performed in a 3D environment consisting of buildings and the sensor, for discretized azimuth angles (model data created from LIDAR measurements of Columbus Ohio, by Wolpert Inc). The ray-trace results are stored in an *Oct-Tree* . A method of storing this information, an *Oct-Trees* stores data that describe volume the way *Quad-Trees* store data that describe area. As illustrated in Figure 2.11 and Figure 2.12, the *Oct-Tree* allows increasingly small voxels (3D pixels) to store the ray-trace information from the model. Either a value of '1' or '0' is given for each voxel dependant on whether the ray hit a building or not. The result, per Figure 2.13 is a binary map of the area *for that particular azimuth angle*. Assuming a constant altitude, 36 *Occlusion Masks* are created for every ten degrees of azimuth change.

Next, the PD-Map is generated by intersecting the *Road Mask* and the *Occlusion Mask*. The final result, as shown in Figure 2.14, is then used to filter change-detections collected over a series of frames at that azimuth. This results in a great reduction in the false-positives that result from parallax changes as the observing platform circles the area.

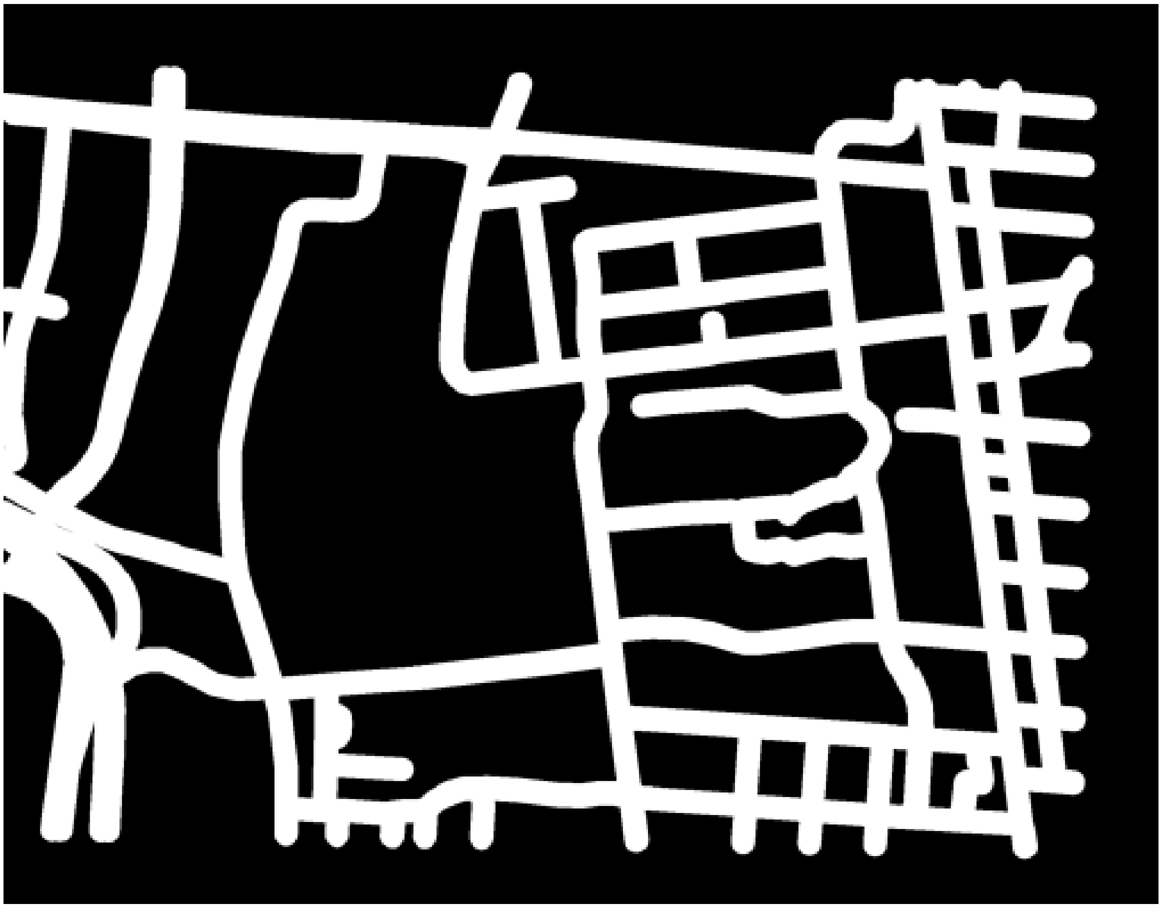


Figure 2.10: Pictured here is a *road mask* generated from road centerlines. White = 1 or “roadness,” and black = 0, “non-roadness.” (Illustration from *Context Aided Tracking and Track Prediction in Aerial Video Surveillance* [29])

.

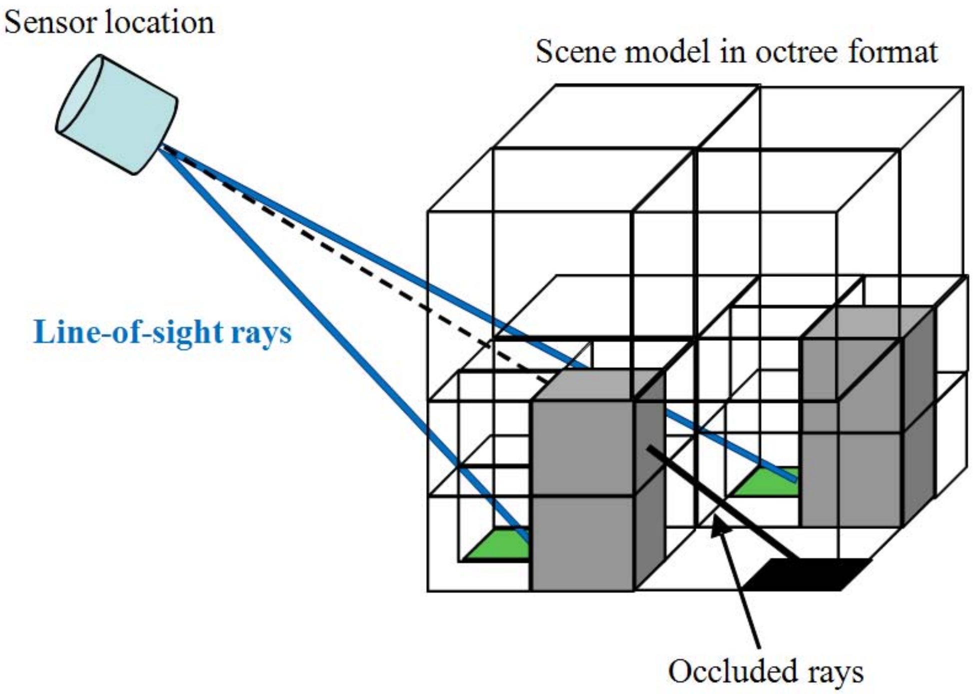


Figure 2.11: The ray-trace operation in the 3D model. Note that most of the voxels are empty. (Illustration from *Context Aided Tracking and Track Prediction in Aerial Video Surveillance* [29])

.

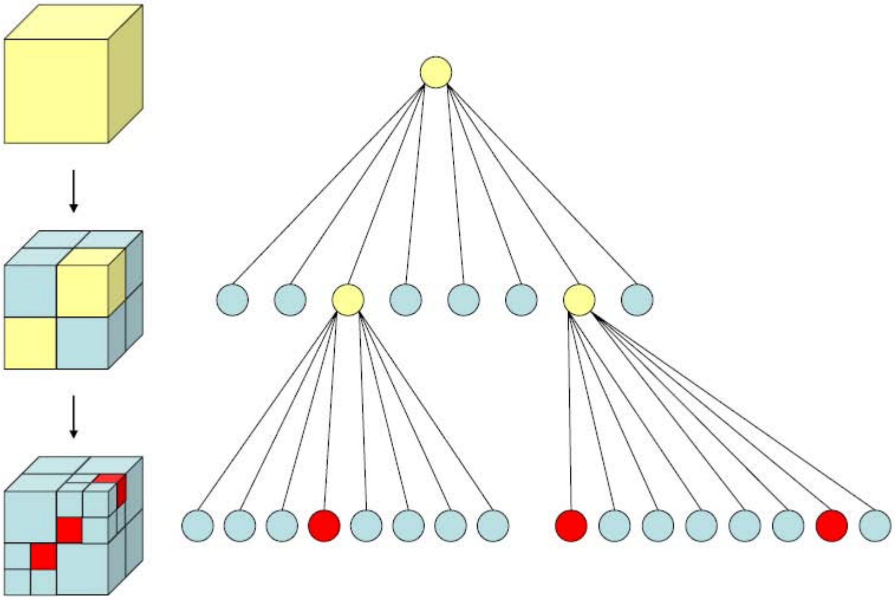


Figure 2.12: The *Oct-Tree* Allows for increasingly finer measurements of a volume to be stored in a lossless, efficient way. (Illustration from *Context Aided Tracking and Track Prediction in Aerial Video Surveillance* [29])

.

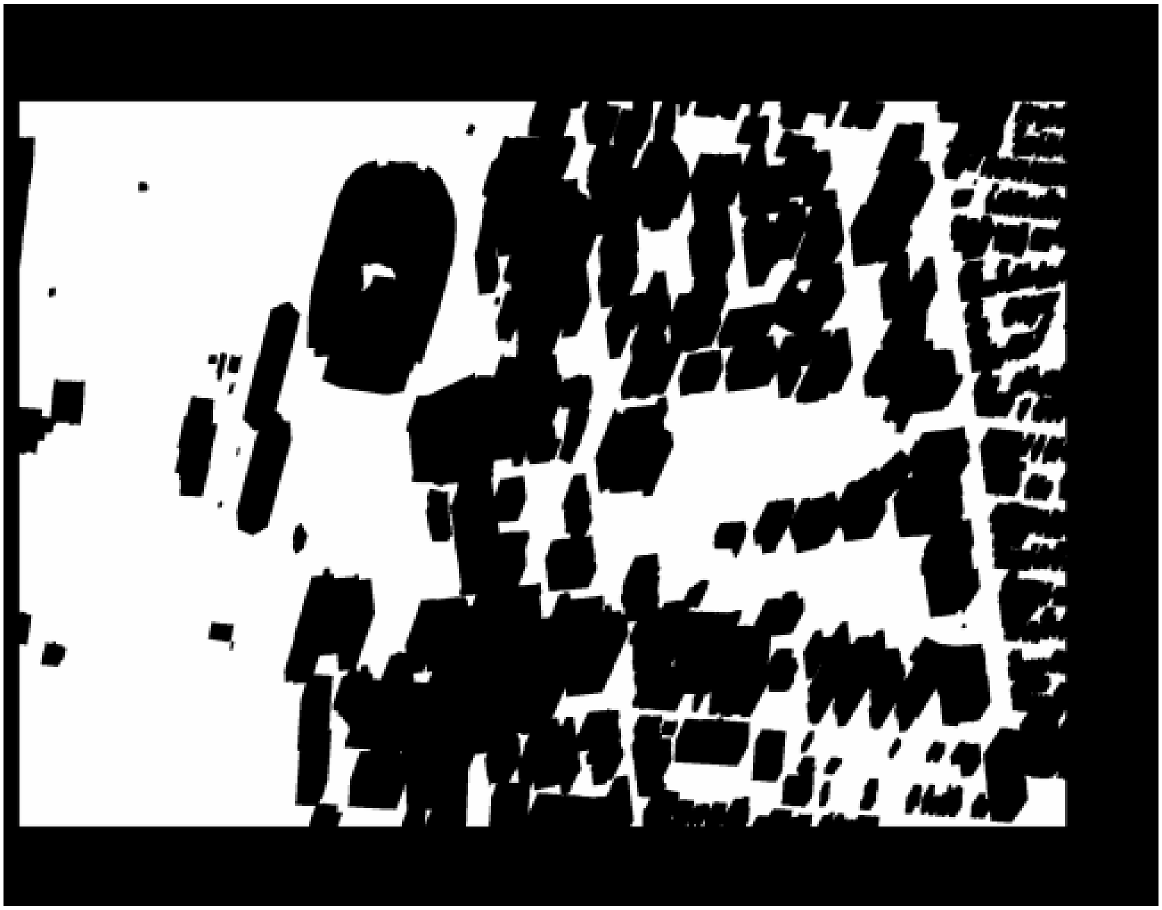


Figure 2.13: An *Occlusion Mask* created from ray-tracing. White = 1 = "non-occlusion area" and black = 0 = "occlusion area." (Illustration from *Context Aided Tracking and Track Prediction in Aerial Video Surveillance* [29])

.

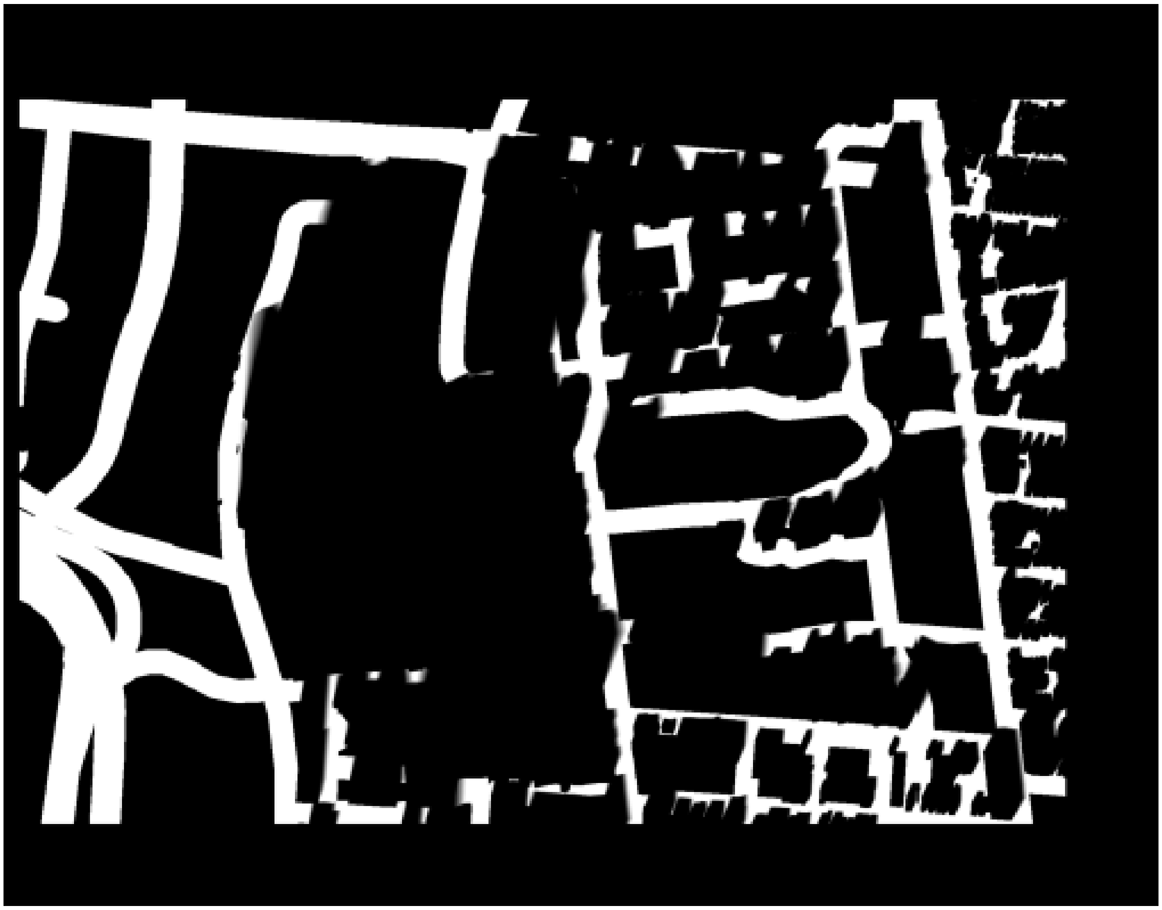


Figure 2.14: The final PD-Map. White = P(detection) = 1 and black = P(detection) = 0. (Illustration from *Context Aided Tracking and Track Prediction in Aerial Video Surveillance* [29])

.

2.6 Chapter II Summary

In this chapter many ostensibly disparate subjects were illustrated in such a way that a grand-unifying theme was allowed to emerge. In order that the work could begin on sound doctrinal footing, *Predictive Battle-Space Awareness* was introduced as the predicate of our undertaking, with the hoped-for consequent being *Effects-Based Operations* in the Battle-Space [16]. This, then, defined the *Operational Framework* within which the work intends to exist.

The problem was motivated by the works of Makris and Ellis [17], [18] which conveniently contained many of the key concepts that would be enlarged upon: tracking, feature recognition, learning, graph theory, and statistical modeling. To this, it was necessary to add the essential kinematic constraints of the system which we intend to model and found *The Highway Capacity Manual* useful to this end [19]. This proves doubly useful when we attempt to calculate the cost of road segments in succeeding chapters.

Next, the *Mathematical Framework* alluded to in Chapter I was treated with the examination of two (not mutually exclusive) approaches to interrogating this domain knowledge. First, *Graph Theory* was discussed along with the possibilities inherent in graph search and traversal. *Ford-Fulkerson* and *Dijkstra*, representative of these sorts of algorithms, were then illustrated. Second, *Temporal Statistical Models* were introduced with particular emphasis placed upon the works of Rabiner [22] and Maybeck [23]. Hidden Markov Models and Kalman filters, discrete and continuous approaches respectively, offer unique promise, especially when a predictive capability is desired in time.

Several case studies were then described with the intent that the preceding principles, and various combinations of them, could be viewed within our *Operational, Representational, and Mathematical Frameworks.* Work was described that proposed using graph theoretic and temporal statistical modeling to predict where a vehicle being tracked might go. The culmination of this approach, carried out by the Air

Force Research Laboratory, was the *Dynamic Tactical Targeting* suite [25], which, despite its eventual cancellation, succeeded in inspiring the current work.

To this basic underpinning, new thinking on the subject was added that takes into account context in order to yield semantics. The work of Antony and Karakowski [27] shows great promise when reasoning engines and natural language processing capabilities may be assumed. It is particularly important to our work that semantics may be used either to constrain the analysis of a domain, or may be instantiated as actual features for greater knowledge representation. Microsoft ClearFlow, an application that has enjoyed actual deployment in the market, was described [28]. This example illustrates the power of GeoProcessing combined with the *Software as a Service* business model. It also illustrates the labor-intensive aspects of employing the statistical approach because of the time and manpower required to build accurate models. Finally, the work of Scott Pierce was used to show practical GeoProcessing that provided real ISR solutions.

Chapter II, in summary, attempted to define the problem space by exploring the *Operational, Representational and Mathematical Frameworks* and the cutting edge thinking that has emerged recently in this field. In Chapter III we will be forced to narrow our emphasis to a subset of these myriad approaches and to combine what remains into an algorithmic approach that will readily fit with other work being carried out.

III. A Graph Theoretic Approach to Vehicular Destination Prediction

3.1 Introduction

The objective for Chapter III is to set forth a practical approach to solving the problem of vehicular destination prediction using the insights from Chapter II as a guide. Since many contemporary approaches were discussed, it is necessary first to admit those that are the most feasible and to discard the rest. This task reveals itself to be somewhat polar in nature since the two broad categories explored during the research were either graph-theoretic or probabilistic. Though these are by no means mutually exclusive, (in fact it is believed that they are highly complementary) to develop and demonstrate both would be prohibitive. Moreover, since a choice must be made, the graph-theoretic approach reveals itself to be the more attractive choice for two reasons. First, its reliance upon GIS data-structures that exist and are easily obtained make preparation of the data environment an easily surmountable task. In contrast, with the probabilistic approach, data models would have to be trained, thereby requiring an enormous data-collection effort. The second reason is that the graph-theoretic approach corresponds well to many efforts underway in the GIS community that will admit the use of already-developed tools and scripts for our own purposes. This consideration allows for a more powerful concept demonstration since much of the groundwork has already been accomplished.

To that end, the approach described in this chapter is primarily graph-theoretic (though with some probabilistic elements) and relies upon GIS technology and its abilities to operate on data-sets using well established theory in the field of computational geometry. We constrain a Dijkstra search in a municipal area from points in a tracklet until a given time horizon is reached. We use that search to define a geometry within which the subject may mathematically be allowed to travel to, within that time horizon. That geometry, called alpha hulls, also assists in down-sampling the tracklet in a reasonable way in order to reduce unnecessary computations. Next, the alpha hulls are further constrained, according to standard distance polygons created from the distribution of location centroids adjacent to the alpha hulls. Another operation will consider the overlap of these alpha hulls and derive negative space polygons which (it will be shown) will limit the mobility of the target if the assumptions, described next, hold. Also contingent on these key assumptions, will be the creation of tesselation products that will constrain the space further if the target has been deemed near the end of its journey. Figure 3.1 illustrates the essential flow. The process repeats until the tracklet finishes. Finally, we show that the process is able to generate certain semantics as to the subject's possible intentions. This chapter begins with a discussion of the problem-space and assumptions that can be made about it. It then describes the various algorithms used to interrogate the problem-space. Following this, the algorithms are fused to form a single process.

3.2 Types of Journeys and Their Implicit Assumptions

Former work discussed in Chapter II [24] hints at an approach to dynamic path prediction that deserves greater analysis. The central premise behind Weeks' and Nanda's proposals [24] was that a vehicle being tracked, and for which it would be desirable to employ a predictive capability, will exhibit a general economy when traversing a space between points A and B. This illustrates one of the great underlying dynamics with which this thesis will have to contend, namely that there are two broad categories of journeys. The first, which we shall refer to as the *Deliberate Journey* describes most road traversals which people make on a given day, with the object of attaining a destination in an optimal fashion. People commuting to their places of employment, driving to a market, or visiting friends are all common examples. It is important to note here that *Deliberate Journeys* are not constrained to always follow the same route in a city but, as per the discussion on Microsoft Clearflow in Section 2.5.3 [28] , may vary with traffic densities according to the time of day and week. The only salient characteristics of the *Deliberate Journey* are that there is a starting location A and an intended destination B, and the driver will wish to traverse that distance as optimally as he or she can.

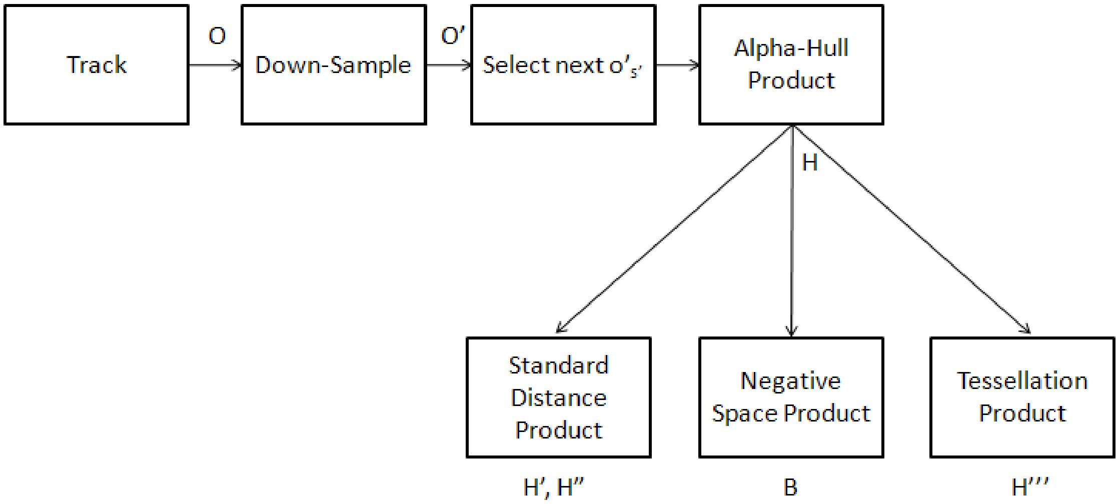


Figure 3.1: A generalization of the flow of the entire process. Variable and set descriptions follow in this chapter.

This begs the question "when would a driver not prefer the most optimal route when making a journey?" A brief reflection on one's own driving habits might recommend some examples. For instance, a tourist seeing the sights in a city or the proverbial Sunday driver will drive in circles without appearing to have any destination whatsoever. Similarly, mail carriers on their routes and policemen on patrol will fit into this category. This, then, illustrates the second broad category, which we will call here the *Circuitous Journey*. The object of the *Circuitous Journey* will not be to close the distance between locations A and B but will be to visit many destinations. It may also include journeys where no destination is intended at all. Instead, the journey itself might be the object. Naturally, if we wish to apply these descriptions to the real world, we will not wish to be constrained by such a rigid orthodoxy. It is important to note that there may exist combinations of these two types of journeys. For example, someone might suggest "Let's visit our friends in the next town. But since it is such a nice day, we'll take the scenic route." Hence, though we will treat tracklets as though they were one or the other, it is useful to understand them in terms of fuzzy-set membership rather than as either-or propositions.

As the object of this thesis is mainly military, the reader might wonder at the relevance of the present discussion. After all, we are interested neither in city commuters nor in sight-seers but, instead, in insurgent cells operating in urban environments. The point becomes relevant because, when tracking such enemies from staring platforms in the air, if a predictive capability is desired, it becomes necessary to treat the two types of journeys in different ways. If the assumption is made that the subject is undertaking a *Deliberate Journey*, then the possible options for his future destination is greatly constrained per Weeks and Nanda [24]. Conversely, if a *Circuitous Journey* can be established for the subject, the predictive capabilities will be diminished. However, with a *Circuitous Journey*, a semantic product can result. For instance, a subject might appear to be performing his own surveillance on an intended terror target. He or she might also be aware that he or she is under scrutiny and be practicing counter-surveillance techniques in order to diminish his observability. In these cases, the target of interest behaves in a manner that warrants further (human) analysis.

3.2.1 The Directed Destination Assumption. It now becomes necessary to formalize our assumptions so that they may be exploited in this thesis. Contingent upon our description of the two types of journeys, we rely on the assumption that the former type is, by far, the most common. To that end, we will call our first assumption *Directed Destination*, and it stipulates that a driver will take the least costly route between two points. The advantage of this assumption, is that it proposes per Figure 3.2, if a vehicle travels from point A to point B, then the distance represented by line-segment A-B, added to the distance from B to the distance to all other destinations, will necessarily be longer for some of the destinations than from A. Since the premise assumes that the vehicle will take the shortest path between two given points, then these destinations may be pruned from the space of possible destinations. A practical example might be when a vehicle passes the entrance of a highway, thereby precluding all possible destinations where the highway would have yielded a faster travel time than if the vehicle navigated the city blocks.

3.2.2 The Slow-Fast-Slow Assumption. The second, *Slow-Fast-Slow*, is an elaboration on the observation that city grids are laid out such that there are expressways and highways that allow for quick commutes along the cardinal points. This allows drivers to spend a minimal time navigating urban mazes with traffic lights and congestion in favor of traveling slowly only until the first on-ramp is encountered, then traveling quickly over most of the distance, and then exiting along an off-ramp to travel the final leg of the journey slowly again. *Slow-Fast-Slow* provides that if a vehicle has traveled (slow) to an on-ramp, and then traveled (fast) on an express artery, and then exited into a neighborhood (slow) then all other slow neighborhoods reachable from the fast artery may be pruned from the space. This is also a corollary of *Destination Directed* and Figure 3.3 demonstrates a reasonable case.

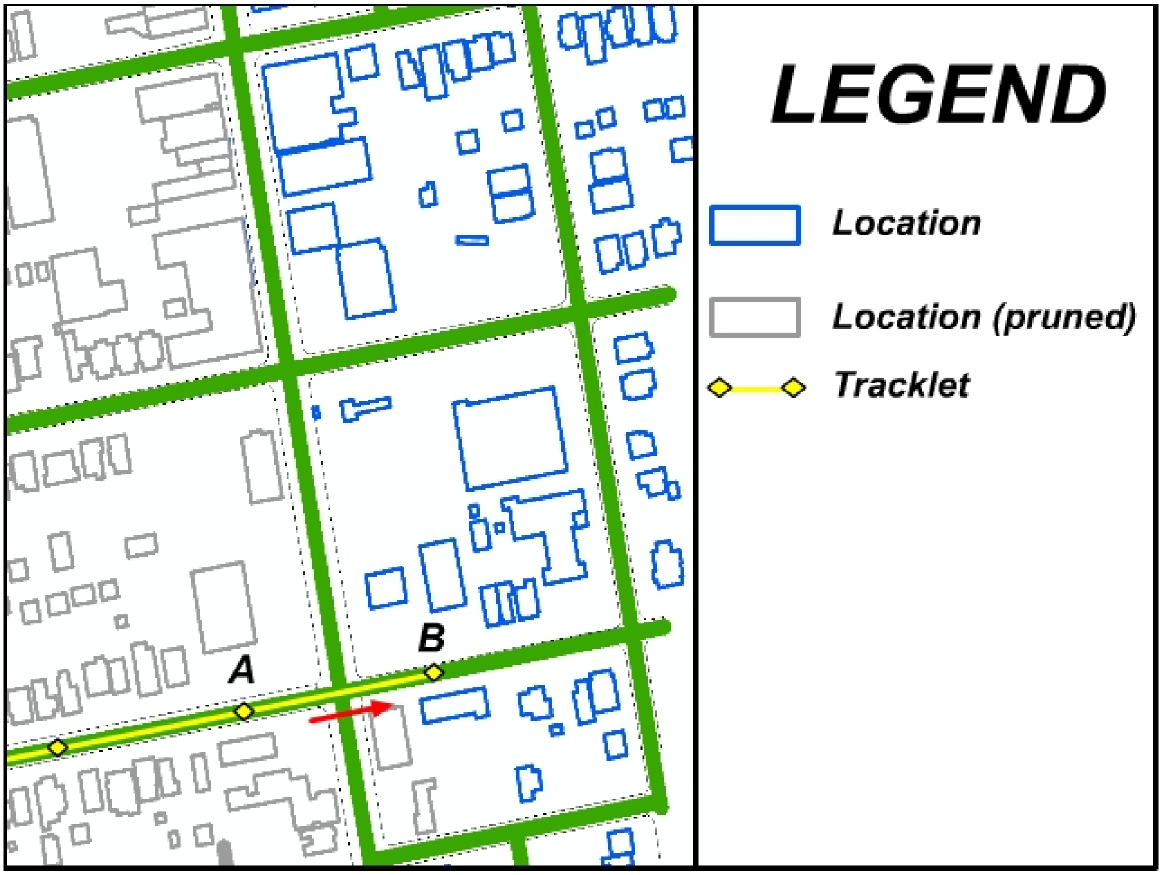


Figure 3.2: The *Destination Directed* assumption allows locations to be pruned.

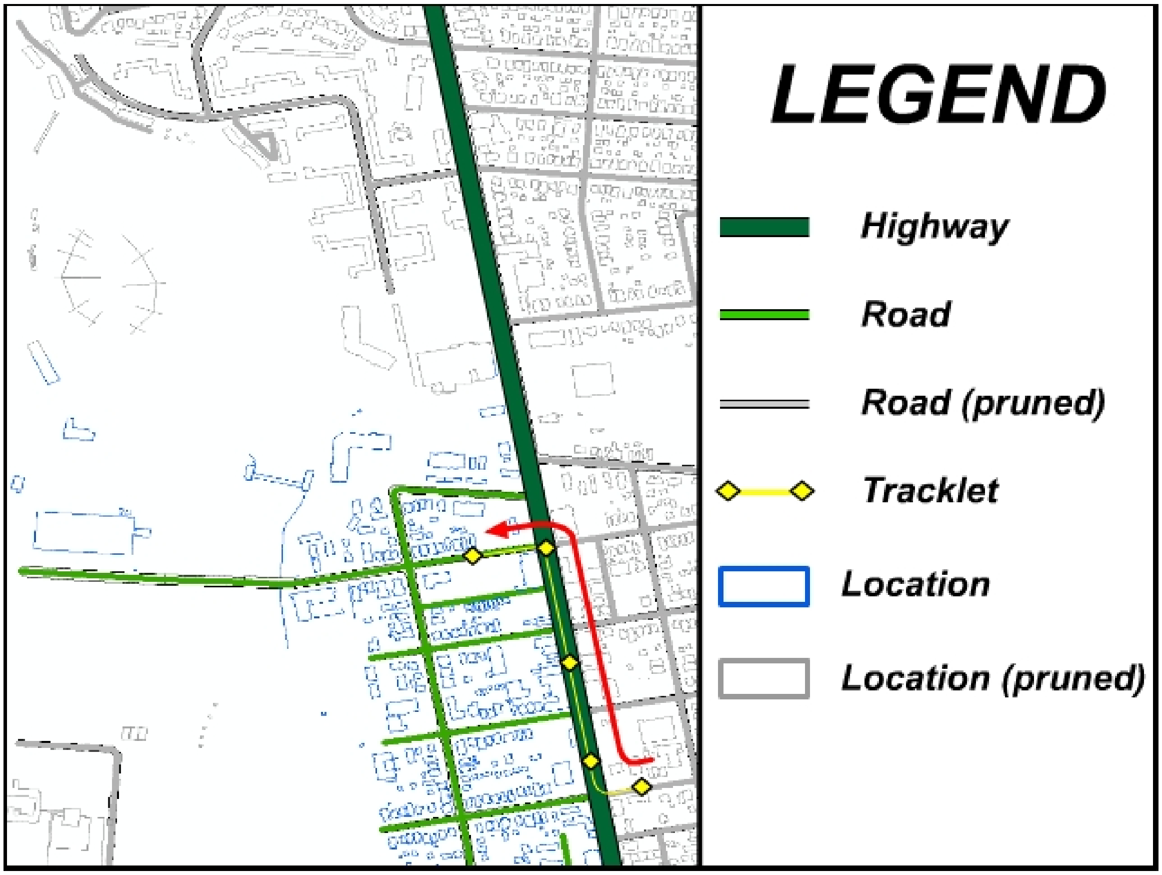


Figure 3.3: The *Slow-Fast-Slow* assumption allows entire neighborhoods to be pruned.

In the next sections, these assumptions are used to inform a practical algorithm that takes the real or near-real time tracking information from a vehicle and uses it to search the GIS layers of a municipal grid in order to bound an area containing its possible destinations for the observed points in the tracklet. The goals are, first, that this bounding geometry be as small as possible and, second, that it be as accurate as possible.

3.3 The Preparation of the Data-Sets

Since it is the object of this approach to operate in real or near-real time, it is prudent to attempt as much preparation of the urban data as possible. In order to do this, some mathematical rules of engagement must be set forth. Throughout this chapter, it will be necessary to operate on points, lines, and polygons because these are the essential elements of GIS systems. These objects are easily manipulated with set-theory operators if they are viewed as sets. We must therefore entertain the proposition that line objects and polygon objects are actually sets of points. This becomes particularly relevant when two data-sets of different dimensionality must be combined in an operation. For instance, imagine polygon A and point set B, and that these sets share some overlap in space. It is useful to define the spatial intersection of these regions such that all points B that fall within the space of polygon A are selected. This is reminiscent of classic Venn diagrams, and we may easily accomplish this operation with:

$$C = A \cap B \tag{3.1}$$

It is important to note here that operations between sets of differing dimensionality will *always result in a solution of the lowest dimensionality*. Hence, C will be a collection of points $C \in B \wedge C \in A$. Predictably, operations between sets of the same dimensionality always result in a solution of that dimensionality. Also, in order to follow the growing list of variables and sets, Table 3.1 in Section 3.4.9 will be useful as a reference.

3.3.1 The Road Network and the Municipal Data-Set . The first task then is to define a set of destinations D within the urban grid. These are composed of buildings, parks, city commons, park-and-rides, and any other point that could be regarded as the terminal node of a journey. Although it would be more spatially correct to represent these as polygons, it makes more sense computationaly to represent them as the centroids of those polygons. Next, a graph $G(V, E)$ is defined such that E are the road segments defined in the space. Edges E are directional so that, for two-way arteries, there are two edges of opposite directionality connecting two vertices. Vertices V are associated with intersections of these road segments. Furthermore, vertices V must be defined for terminal areas of the roads such as driveways and parking areas, and for the terminal extent of the data layers (where the data-set ends). Once this has been done, it is necessary to attribute road segments $E \in G(V, E)$ in such a way that their contiguous nature is apparent. In other words, all line segments belonging to a road called Oak Lane is given that naming metadata as well as its respective predecessor and successor vertices V. This linked-list morphology will aid later in our algorithm. Also, it is necessary to map the set of destinations D to $G(V, E)$ in a meaningful way. The relationship need not exist in an explicit data-structure since the two sets already enjoy a spatial coincidence. Instead, it is proposed here that a tessellation or k-nearest neighbor algorithms (commonly included in GIS development environments) are employed such that at least one $v \in V$ is associated with each $d \in D$. A final employment for tessellation (modified for our purposes from the Voronoi variant) will be to create a set W_n of tessellation regions about the exit ramps of n major expressways. A tessellation region (a polygon) $w \in W$ will be useful in determining the travel intentions of a vehicle that has exited a highway, given the *Slow-Fast-Slow* assumption.

3.3.2 The Tracklet of a Surveilled Vehicle . Tracking systems vary widely though their basic mode of operation coincides enough to permit us to work with generalizations. A tracker collects (for our purposes, at a rate of 1 Hz, or once per second) position information in the form of:

$$o_s = \{x_s,\ y_s,\ \upsilon,\ t\}\,, \tag{3.2}$$

where x and y are GPS coordinates, υ is velocity, and a t is the timestamp. We denote an observation as $o_s \in O$ for S samples where S is the total number of samples in O. We employ index s defined as

$$s = 1,\ 2,\ 3,\ \ldots,\ S. \tag{3.3}$$

As for the elements of tracklet O, o_s is a 4-tuple containing $x_s,\ y_s,\ \upsilon$, and t. These data are passed through a *Kalman Filter Constant Velocity Model*, undergo a ground-truth (registration) operation that registers them to the spatial database, and then added to set O such that

$$O = \{o_1,\ o_2,\ o_3,\ \ldots,\ o_S\}\,, \tag{3.4}$$

where $|O| = S$ and O is sub-scripted with index $s = 1,\ 2,\ 3,\ \ldots,\ S$. Finally, it is necessary to downsample O. This is denoted as O' such that

$$O' = \{o'_1,\ o'_2,\ o'_3,\ \ldots,\ o'_{S'}\}\,, \tag{3.5}$$

where $|O'| = S'$ and O' is sub-scripted with index $s' = 1,\ 2,\ 3,\ \ldots,\ S'$.

3.3.3 The Observation Spot . This thesis has proposed that its chief aim is to aid the surveillance platform in its mission. Chapter I proposed that a certain agnosticism be preferred as to which platforms and sensor payloads this effort was meant to profit. However, mission profiles vary widely and the proposals of this thesis are more useful to some than to others. Whether sensors are EO, IR, SAR, or a combination, there are two broad categories into which mission profiles fit [34] [35]. These may be generally defined as the persistent staring spot and the moving swath. The former, as depicted in Figure 3.4 and Figure 3.5, is literally an area that is viewed constantly for a period of time. This product may be thought of as a movie shot over a wide area that lasts as long as the platform stays in the area, and is consistent with our definition of surveillance in Section 1.2 (Recall, however the limitations posed by SAR). The latter, as depicted in Figure C.2, is a swath of mosaic-ed images taken along a flight path. This is less amenable to a real or near-real time intelligence collection and is more consistent with our definition in Section 1.2 of reconnaissance. This thesis focuses on the persistent staring spot since we have said that the intent is to aid real and near-real time operations. To that end, we define a coverage spot polygon C that is represented in our space that is four square kilometers. (This is the actual specification for EO and SAR spot-coverage on the Global Hawk [34]). Predictions undertaken in the space are given context by their spatial relation to C. A final note on geometry: the aperture shape for our sensors will dictate that C will be square for a single frame. However, a persistent staring spot will arise, per Figure 3.4 and Figure 3.5, from the aircraft circling a given point on the ground. This causes the picture frame to rotate around this center, making its effective geometry round. Hence, C are represented as a disk with a radius of 2 km.

3.3.4 Other Contextual Objects. Although this work is primarily interested in the context generated from the municipal data-set $G(V, E)$, the tracklet O, and the UAS coverage spot C, there are other possible context-adding objects that might be placed in the system for a greater semantic yield. Among these are the operational radii of dismount tactical elements, the drop zones of the Joint Precision Air-Drop System (JPADS), or even the effective radius of a stationary US Army sniper, all of which can be represented as polygon disks. Other polygon disks might include electromagnetic energy footprints to include cell-tower radii, High Energy Radio Frequency weapons (HERF), High-Powered Microwave weapons (HPM) effective areas [33], and electronic warfare applications such as radar/radio jamming fields of effect. In addition, Unattended Ground Sensors (UGS) and surveillance cameras are modeled as a disk, minus occluding objects such as buildings and trees. Finally, other forms of transportation networks may be modeled as we have done with the road network,

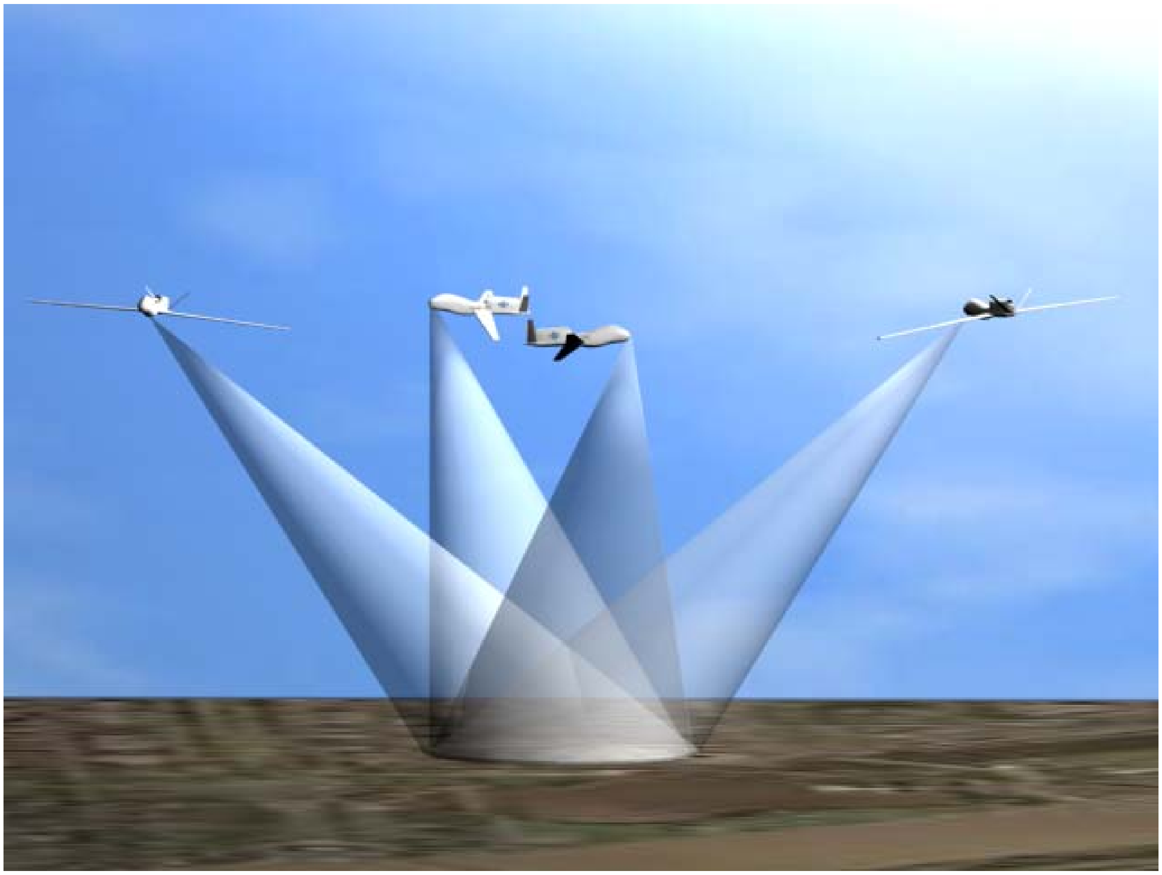


Figure 3.4: The observation spot of a Global Hawk UAS as viewed from the side.

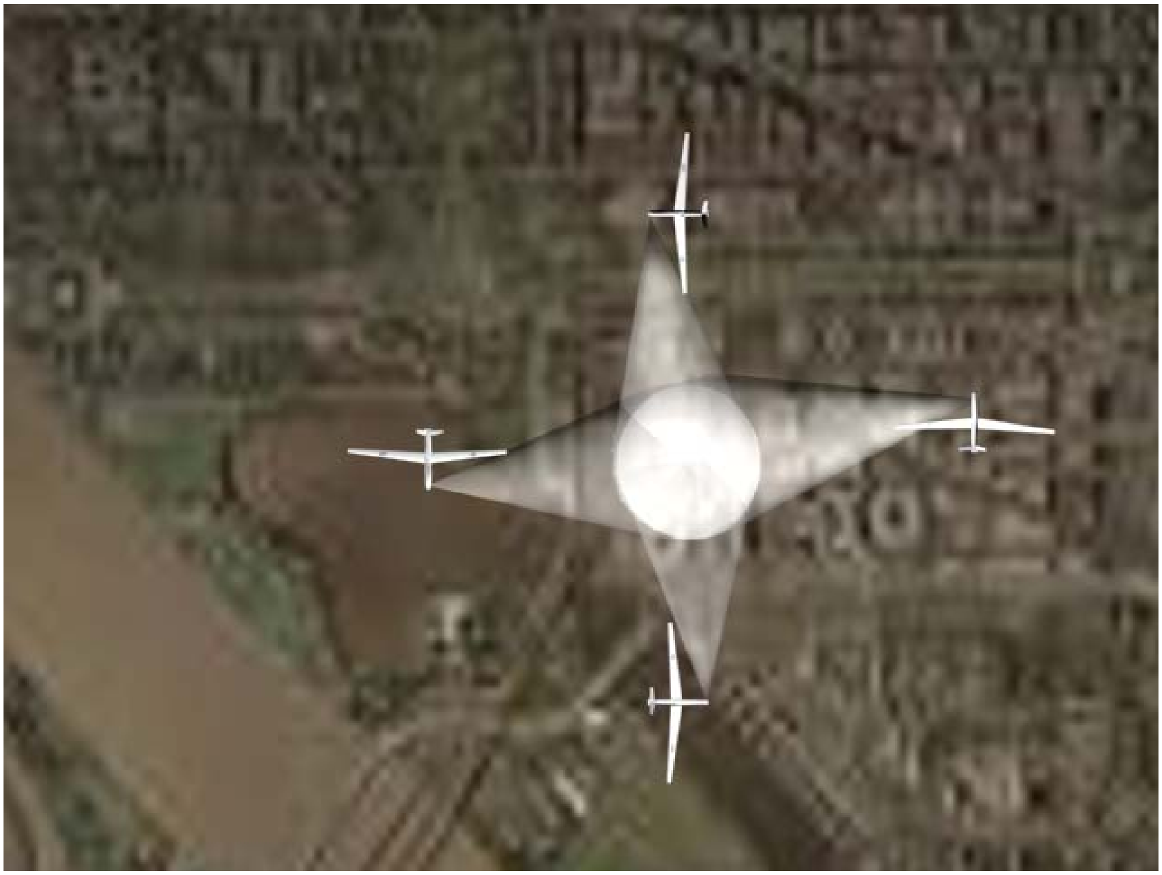


Figure 3.5: The observation spot of a Global Hawk as viewed from above.

to include river transport systems (such as the Tigris and Euphrates in Iraq) and underground tunnel networks (such as at the Egypt-Gaza border and in Viet Nam during our conflict there).

3.4 Development of a Vehicular Destination Prediction Algorithm

On tracking a vehicle from an airborne, staring array, tracklet O begins to grow at about 1 Hz from the first sample o_s at $s = 1$ (assuming a tracking capability). The intent is that certain points in O will serve as starting points for multiple Dijkstra searches of the space, extending radially from o_s until a time horizon T, using cost Ω in time, as described by Equation 2.2, Equation 2.3, and Equation 2.4. The intent here is for Dijkstra to be extended in each possible direction finding the shortest path to each extremity within T. For this, a bounding geometry becomes necessary to describe the results of that search. To that end, convex hulls and alpha hulls are described. Next, it would be infeasible to search from *every point*. Hence, a reasonable method for downsampling O is explored. Finally, there are some operations that can be made to final searches sets that enables them to be pruned for even sharper results. The standard distance and tessellation algorithms assist in this.

3.4.1 Deciding Upon Scope, and Extent. An elementary consideration that has to be made is the scope and extent of our search from a given observation point o_s. This may be done by utilizing a time-horizon T such that a search will not extend beyond T. Otherwise, the system might set forth (assuming limitless computational and storage capability) that "the vehicle will eventually visit Calcutta India or Vilnius Lithuania" even though the tracklet O began at the corner of Fifth and Main in Booneville Missouri. Statements like these would be practically devoid of any informational content relative to the situation. Hence, we must search within a bounds that yields a more meaningful result. T, necessarily ought to a reasonable fraction (say 20 %) of an actual journey that traverses a major city or the extent of a coverage spot. Experimental evidence elaborated on in Chapter IV justifies a time horizon T

of 1 to 5 minutes for the Dayton Ohio metropolitan area which provides the data-sets used to test these algorithms.

3.4.2 Searching the Space. In order to search the space, we use an algorithm that employs Dijkstra's shortest path search described in Section 2.3.1. The search begins from point $o'_{s'}$. The search must repeat until it has reached all reachable locations in the set of all possible locations D that it can travel to within time T and return a respective path for each one. In this case, observed point $o'_{s'}$ in the tracklet, the road network graph $G(V, E)$, the set of locations D, and time horizon T are passed to a function *radialDijkstra* such that the search continues seeking a shortest path for a given location $d \in D$ until it's cost $\Omega > T$. It then sweeps across all $|D|$ possible destinations and finds the shortest distance to those (here, we employ m as the index of reachable locations in D such that $m = 1,\ 2,\ 3,\ \ldots, |D|$). The result of each search is a path which we shall denote $p(o'_{s'},\ d_m)$, meaning an optimal route between $o'_{s'}$ and the m^{th} $d \in D$. Note that this function is indexed by s' because there exists one for every $o'_{s'}$ and by m because there will be one path from that $o_{s'}$ to each of the m locations. The set of all of these paths from an $o_{s'}$ is denoted with an upper-cased P and subscripted with s'. Therefore,

$$P_{s'} = \{p(o'_{s'},\ d_1),\ p(o'_{s'},\ d_2),\ p(o'_{s'},\ d_3),\ \ldots,\ p(o'_{s'},\ d_m)\}\,, \tag{3.6}$$

for the $|D|$ locations within T of $o'_{s'}$. The paths $p(o'_{s'},\ d_m)$ are graphs, though non fully-connected, sparse graphs that are directional and acyclic. The pseudocode for this algorithm in Figure 3.6.

3.4.3 Bounding the Space with Computational Geometry. In order to represent the set of paths P in a way that makes sense geometrically, it is important to bound that space with a polygon using computational geometry. However, such a polygon will be meaningless during analysis unless there is a simpler polygon that can be used as a basis of comparison. It makes sense to begin, then, with something simpler such as a kinetic motion model (demonstrated by the Dynamic Tactical Tar-

```
$Function\ radialDijkstra(o'_{s'},\ G(V,E),\ T,\ D)$
    $For\ all\ d_m \in D,$
        $\Omega_m \leftarrow 0$
        $While\ \Omega_m < T$
            $p(o'_{s'},\ d_m) \leftarrow Dijkstra(G(V,E),\ o'_{s'},\ D)$
            $\Omega_m += \ cost(p(o'_{s'},\ d_m))$
        $P_{s'} \leftarrow p(o'_{s'},\ d_m)$
    $Return\ P_{s'}$
```

Figure 3.6: $RadialDijkstra(o'_{s'},\ G(V,E),\ T,\ D)$ finds paths from $o_{s'}$ to all $d \in D$ within T.

geting (DTT) effort [25]). In a non-variable environment where a vehicle could drive unimpeded, at top speed, and in any direction (say the Utah Salt-Flats) it would be reasonable to define this model $mm_{s'}$ for an s' point, as a circle whose radius depends on the upper limits of a vehicle's possible speed V and a time horizon T. For this, a circular polygonal bounds is described as:

$$mm_{s'} = 2\pi \upsilon T \tag{3.7}$$

where, for our purposes, υ is the top velocity of an average sedan (say, 180 KPH), with the position of the vehicle at the origin, and there exists a set of models MM associated with each $o_{s'}$ where

$$MM = \{mm_1,\ mm_2,\ mm_3,\ \ldots,\ mm_{S'}\}\,. \tag{3.8}$$

The practical result of Equation 3.7 is a very primitive motion model that still serves to bound our space reasonably well. In fact, this defines the uppermost physical constraint for our vehicle and we will use this as a baseline for judging more sophisticated approaches (while explaining and justifying the down-sample approach below, we will employ it because of its simplicity). It will also find a use when we seek to create a figure of merit for our final product. This is called function $motionModel(h_{s'},\ T)$.

A better polygon for bounding the possible space where the vehicle might travel in T is the convex hull. This algorithm takes as its argument a collection of points in a 2-D vectoral plane and will be referred to as function $collectPoints(P_{s'})$. This function is given the points from the all of the shortest paths $P_{s'}$ just described for $radialDijkstra$ (Recall that $P_{s'}$ is really a collection of edges E and vertices V). In other words, all of the vertices from all of the paths in set $P_{s'}$ are collected as:

$$V_{s'} = \bigcup_{m} P_{s'}. \tag{3.9}$$

The convex hull algorithm then returns the set of points that bounds the set of points $V_{s'}$ and draws lines between them. An oft-used metaphor compares it to placing a rubber-band around a collection of pegs in a board and allowing it to constrict around the outermost pegs. An $O\,(n\ Lg\ n)$ implementation of this approach is Graham's algorithm [36]. This approach compares all points in a collection (represented in vectoral space) to a previously defined centroid and orders them according to increasing angle with the central point. It considers n points with its two nearest adjacent neighbors. Per Figure 3.7, if an interior angle θ formed by the point and its neighbors is strictly greater than π then it is considered *reflex* and cannot be a member of the convex hull. In Figure 3.7, the dotted line represents a convex hull for the points.

An even better approach, and the most useful to our purposes, will be an elaboration on the convex hull. The alpha hull is similar to its cousin in that outermost points in a set are used to describe a polyline that bounds that space. The alpha hull takes its name from the parameter α that may range as $0 \leq \alpha \leq \infty$. The formal definition [37] sets forth that there is an α-disk of radius α. For $\alpha = 0$ the disk is a point, and for $\alpha = \infty$, an open half-space. The α-disk is allowed to roll along the exterior of the data (in our case points $V_{s'}$) whereby edges are drawn between points that touch the α-disk. The practical result, rather than a smooth polygon, is an convex structure as per Figure 3.8. Dependent upon the size of α, there can be large fissures

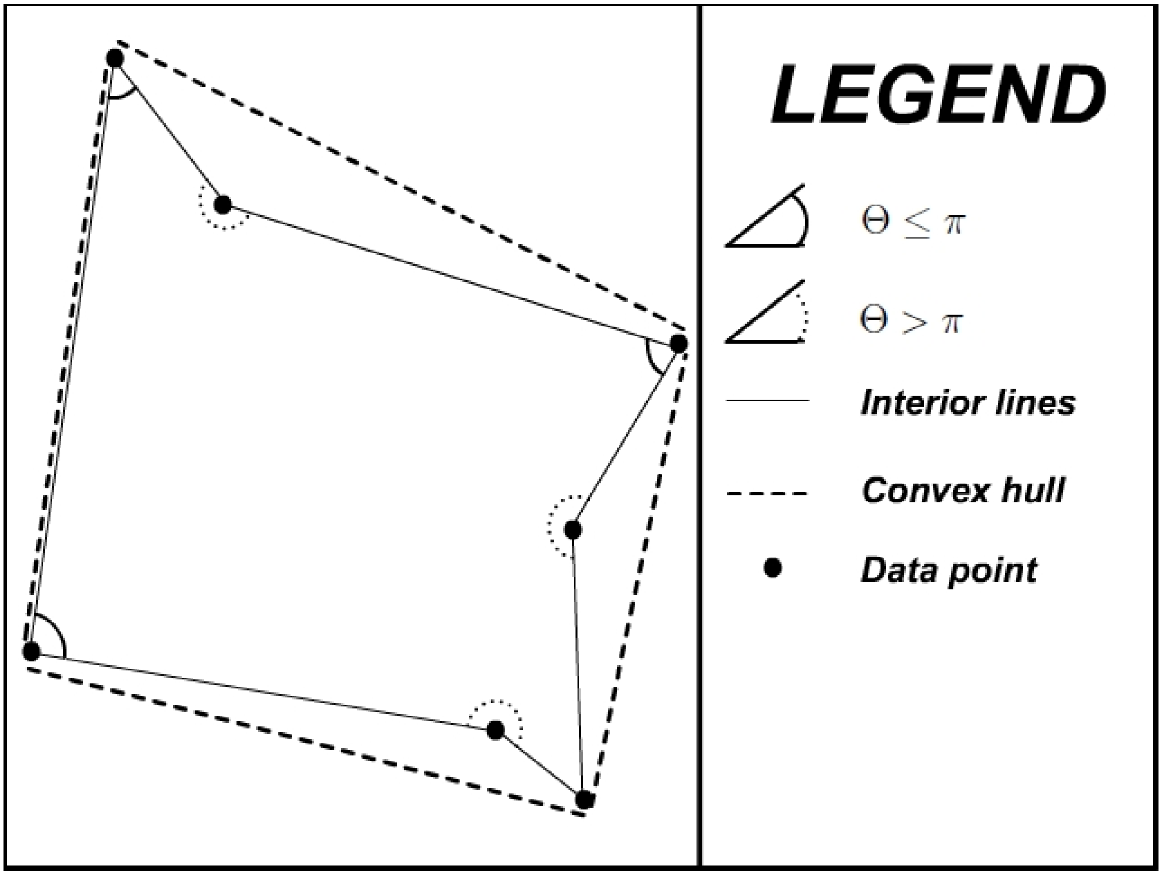


Figure 3.7: A convex hull where $EAB > \pi$ and is therefore *reflexive*.

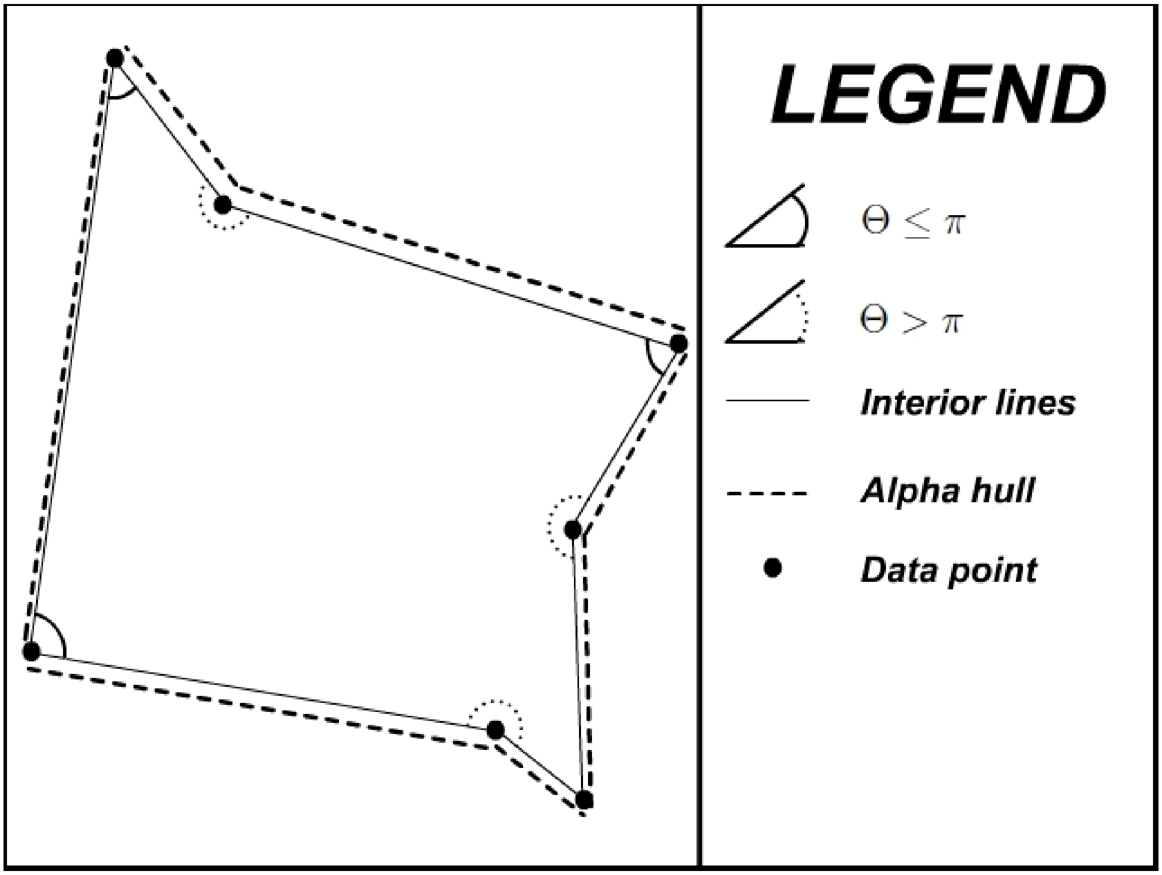


Figure 3.8: An alpha hull from the same data points. The α-disk is assumed to be small enough to include all data.

and even tunnels through the data points as the α-disk becomes smaller. (Note that the convex hull now reveals itself to be a special case of the alpha hull where $\alpha = \infty$).

This suits our purposes better because its area is smaller than that of a convex hull, and it is our goal to shrink the polygon that represents the vehicle's destination as much as possible. It is also a reasonable choice because the lines of the alpha hull conform more stringently to the road structure, thereby disallowing space that is devoid of infrastructure (pasture-land, lakes, etc) to be bounded. The set of all bounding polygons are referred to as H, meaning that for a given down-sampled tracklet O', set H contains each bounding polygon $h_{s'}$ for its associated $o'_{s'}$. The practical consequence of this will be an area that constricts around $o_{s'}$ according to all of the Dijkstra searches to the extent of T. Hence, as the track unfolds, newer polygons accompany tracklet O' yielding the practical information of where the vehicle can be expected within T. This is be denoted as $h_{s'} \leftarrow alphaHull(V_{s'})$ after each s' search and rendered over the GIS layers. Therefore we may write,

$$H = \{h_1,\ h_2,\ h_3, \ldots,\ h_{s'}\}\,. \tag{3.10}$$

3.4.4 Design Choices for Down-Sampling Tracklets . Next, it has been stated that it would be infeasible to search from each of the S samples in O and this deserves a more in depth discussion. Recall that it is one of the primary goals of the algorithm to prune from the space points that could have been reached more optimally than from earlier positions. Hence, it is a waste of processing time to search and compare from every point. This is so because, at a sample rate of 1 Hz, there will be many samples between two given nodes $v_k,\ v_{k+1}$ in our network graph $G(V, E)$. Moreover, the generative source of our signal is extremely noisy and non-linear. Consider, for instance, the accumulation of points at the same spot while the vehicle waits at a traffic signal. The spatial distribution of these points will differ wildly from those that are sampled while the vehicle is driving 100 KPH on the freeway. It is therefore impossible to apply a traditional downsample approach, such as decimation, to provide a meaningful down-sampling of the tracklet data.

Instead, we employ an approach that is more relevant to the spatio-temporal aspects of the tracklet O, and that makes use of our previous efforts. A simple deconstruction of the problem appears in Figure 3.9. This illustration assumes the simplest of possible alpha hulls, the round disk, instead of more complex instances, and reveals an interesting relationship. If o_1 is observed as the first point in the tracklet O, then its corresponding bounding polygon will be created as described in Equation 3.8. Succeeding samples that occur within that space will be of little interest to us because we have already predicted the space within that time-horizon T for o_1. It will therefore be possible to use the bounding polygon to mask and discard those intersecting points. In other words, we discard the observations $o \in O$ that exist *inside the polygon*, and then redefine its cardinality S. This is described mathematically as,

$$O = O - O \cap H, \tag{3.11}$$

and

$$S = |O|, \tag{3.12}$$

where the truncated O is renumbered 1, 2, 3, $\ldots$ S with $Function\ renumber(O,\ S$. The next point of interest to is the next point that appears outside the bounding polygon, or o_1 after the renumbering so that

$$o'_{s'+1} = o_1. \tag{3.13}$$

Such a process may continue until the vehicle reaches its destination. The functionality is referred to as function $downSample(O,\ h_{s'})$.

Thus far, three analysis products have been produced. The first of these is the downsampled tracklet O', with a cardinality of S'. The second is the set of bounding polygons H also with a cardinality of S'. The third is the set created by the intersection of H polygons as depicted in Figure 3.10. The natural consequence of these new polygons is a space that corresponds to our stated goal of finding areas that can be pruned because it would have been more optimal to travel to them from

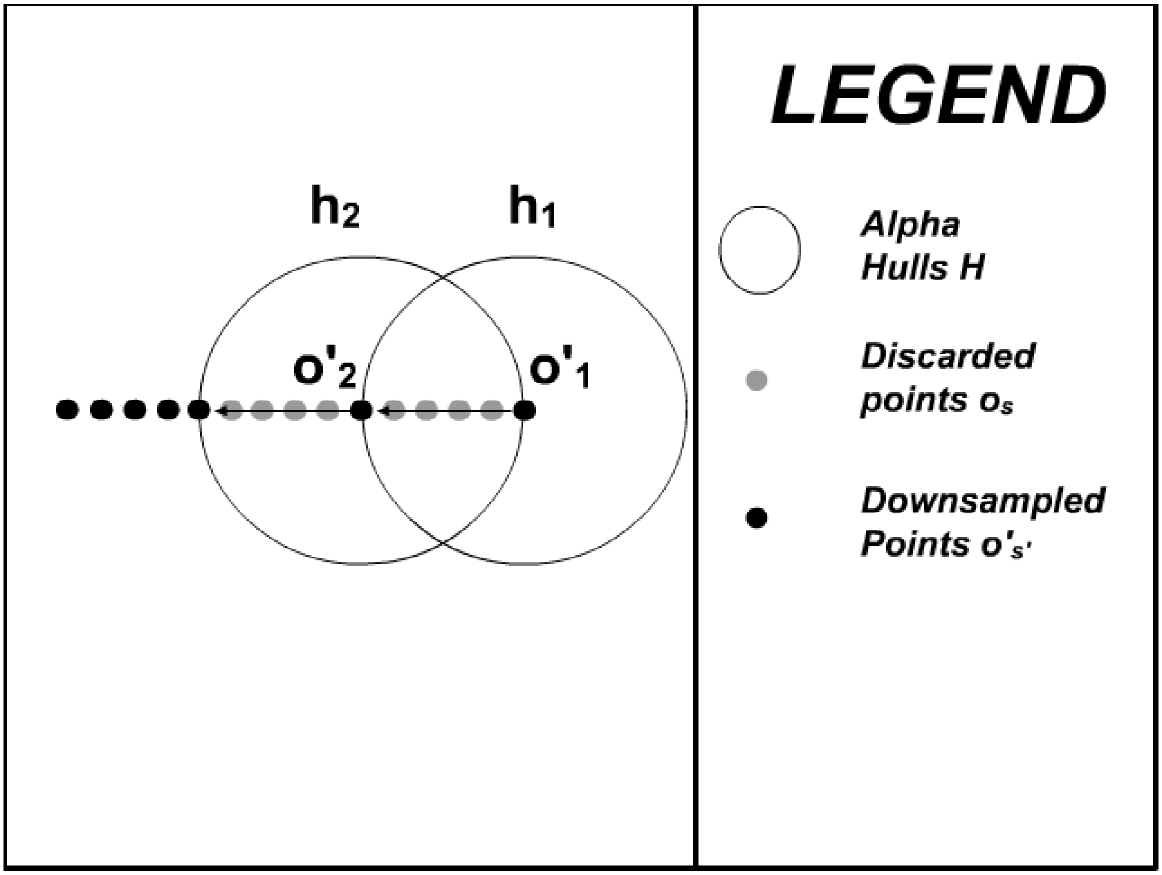


Figure 3.9: Down-sampling a tracklet with search-polygons is easiest to conceptualize with simple motion models.

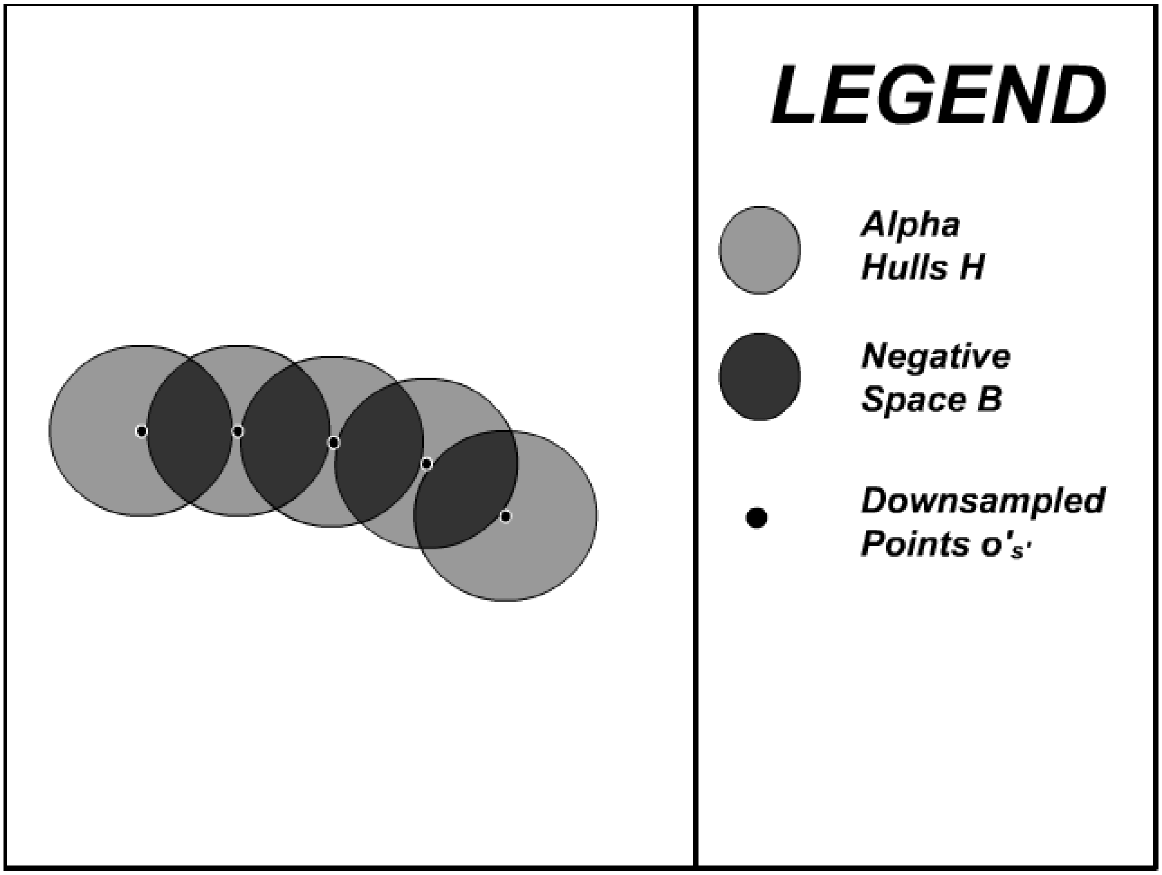


Figure 3.10: The negative space created by the $radialDijkstra(o'_{s'},\ G(V,E),T)$ and $downSample(h_{s'},O)$ algorithms.

$o'_{s'}$ than from $o'_{s'+1}$. This insight singlehandedly offers the benefits of the *Destination Directed* assumption. Given *Destination Directed*, we may state that the vehicle will not visit this space again. We will call the set of these intersections the negative space polygons B and for S' down-sampled points, there shall be $S'-1$ of them. The algorithm will be called function $negSpace(H)$ and is described by:

$$b_{s'} = h_{s'-1} \cap h_{s'}, \tag{3.14}$$

and

$$B = \{\phi,\ h_1 \cap h_2,\ h_2 \cap h_3,\ \ldots,\ h_{s'-1} \cap h_{s'}\}\,. \tag{3.15}$$

where there is a ϕ for the first value in the set because, for h_1, there is no predecessor to operate with.

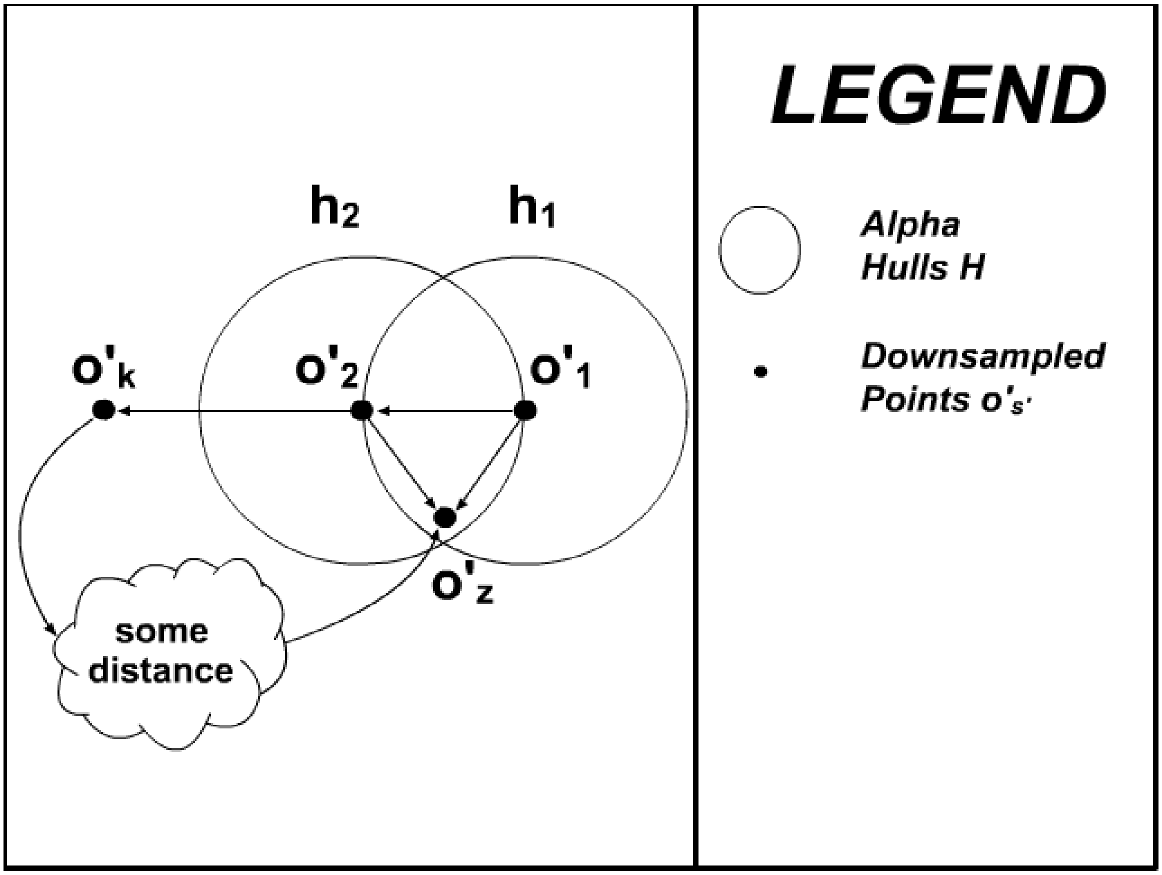


Figure 3.11: The *Directed Destination* assumption allows the negative space B to be pruned from the search space.

It still needs to be shown that the space formed by the intersection of two search-polygons would encompass the locations that could be pruned from the search-space due to the *Destination Directed* assumption. The proof for this assertion for which we will again use our simple version of the alpha hull and refer to Figure 3.11.

Definition: Let two disk-shaped polygons h_1 and h_2 with centroids o'_1 and o'_2 intersect such that the outer circumference of h_1 touches but does not go beyond o'_2 and the outer circumference of h_2 touches but does not go beyond o'_1. The negative space polygon b is the intersection of these two per $b = h_1 \cap h_2$.

Lemma: Let the cost Ω between any points $\left\{o'_{s'},\ o'_{s'+1}\right\} \subset O'$ be a function $\ell\left(o'_{s'},\ o'_{s'+1}\right)$ determined by the amount of time it takes to travel between $o'_{s'}$ and $o'_{s'+1}$. $\ell\left(o'_{s'},\ o'_{s'+1}\right)$ is not strictly equal to $\ell\left(o'_{s'+1},\ o'_{s'}\right)$. For a given $\ell\left(o'_{s'},\ o'_{s'+1}\right)$, a corresponding $\ell\left(o'_{s'+1},\ o'_{s'}\right)$ is not automatically assumed because of the possibility of one-way roads. Also, $\ell\left(o'_{s'},\ o'_{s'+1}\right) = 0$ if and only if $o'_{s'} = o'_{s'+1}$, otherwise, $\ell\left(o'_{s'},\ o'_{s'+1}\right) > 0$.

Proof: We may prove that if a vehicle passes point o_k that he will not visit any point in b by counter-example. If a vehicle does indeed visit the intersected space (say at o_z) after some k samples, then he must take one of two actions. The first would be to perform a U-turn and re-trace his steps from o_k back to o_2, in which case he may then traverse to o_z. The second choice is that the driver could make a turn at some future junction beyond o_k, acquire another artery that is headed back the direction he came, and continue until o_z is reached. The first case is non-optimal because the cost of the journey is $\Omega = \ell\left(o_1,\ o_2\right) + \ell\left(o_2,\ o_k\right) + \ell\left(o_k,\ o_2\right) + \ell\left(o_2,\ o_z\right)$ where he could have simply travelled $o_1 \rightarrow o_z$ with cost $\ell\left(o_1,\ o_z\right)$. Mathematically, $\ell\left(o_1,\ o_z\right)$ must be less than $\ell\left(o_1,\ o_2\right)$. That is because that term is the radius of the circles m_1 and m_2 (by our down-sample definition) and any chord formed from either o_1 or o_2 to a point inside this negative space b must be shorter since the radii of the two circles describes this space. The second case is non-optimal for the same reason: the driver would have incurred the cost $\Omega\ \ell\left(o_1,\ o_2\right) + \ell\left(o_2,\ o_k\right) + \ell\left(\textit{some distance}\right)$ where, again, the $\ell\left(o_1,\ o_2\right)$ alone is costlier than if he had travelled $o_1 \rightarrow o_z$ with cost $\ell\left(o_1,\ o_z\right)$. $\square$

Having defined sets H and B for s', these products are employed again by feeding them back into the process. B in particular offers a lucrative savings in computational time as regards our *radialDijkstra* search because, given the *Destination Directed* assumption, we are no longer required to search these regions. We may define a tabu list $\overline{G(V,E)}$ where vertices $v \in V$ and edges $e \in E$ incident with the set of polygons B are flagged so that they are no longer searched. This will be referred to as function $tabu(G(V,E),\ \overline{G(V,E)},\ B)$ is shown in Equation 3.16 and Equation 3.17.

$$\overline{G(V,E)} \leftarrow G(V,E) \cap B \tag{3.16}$$

The set of vertices $V \in \overline{G(V,E)}$ and the set of edges $E \in \overline{G(V,E)}$ may be removed from $G(V,E)$ so that extra computation is not expended on them in the future. This is done with set-wise subtraction per:

$$G(V,E) \leftarrow G(V,E) - \overline{G(V,E)} \tag{3.17}$$

The final product appears as shown in Figure 3.12, and having made the case, we employ a set of alpha hulls H. As a final note on set B, these products retain a certain *a posteriori* character due to the fact that, when moving from o_n to o_{n+1}, the corresponding b_{n+1} will not be available until the search that has generated h_{n+1} has been completed. However, during the upcomming discussion on the tabu list, B is shown to be immensely valuable to our process.

3.4.5 Narrowing the Result with the Standard-Distance Mask. Thus far, it has been demonstrated that a rich yield of knowledge may be acquired by considering layers in the data-set *relative to each other*. In particular, the tracklet O has been combined with the road network $G(V,E)$ in order to generate the three analysis products described above. However, the richness that typically exists in these municipal data-sets has not been exploited yet. For instance, topology, hydrology, demographics, or buildings have not been considered, or the possible contributions that they might make to predicting possible destinations of a vehicle. To that end,

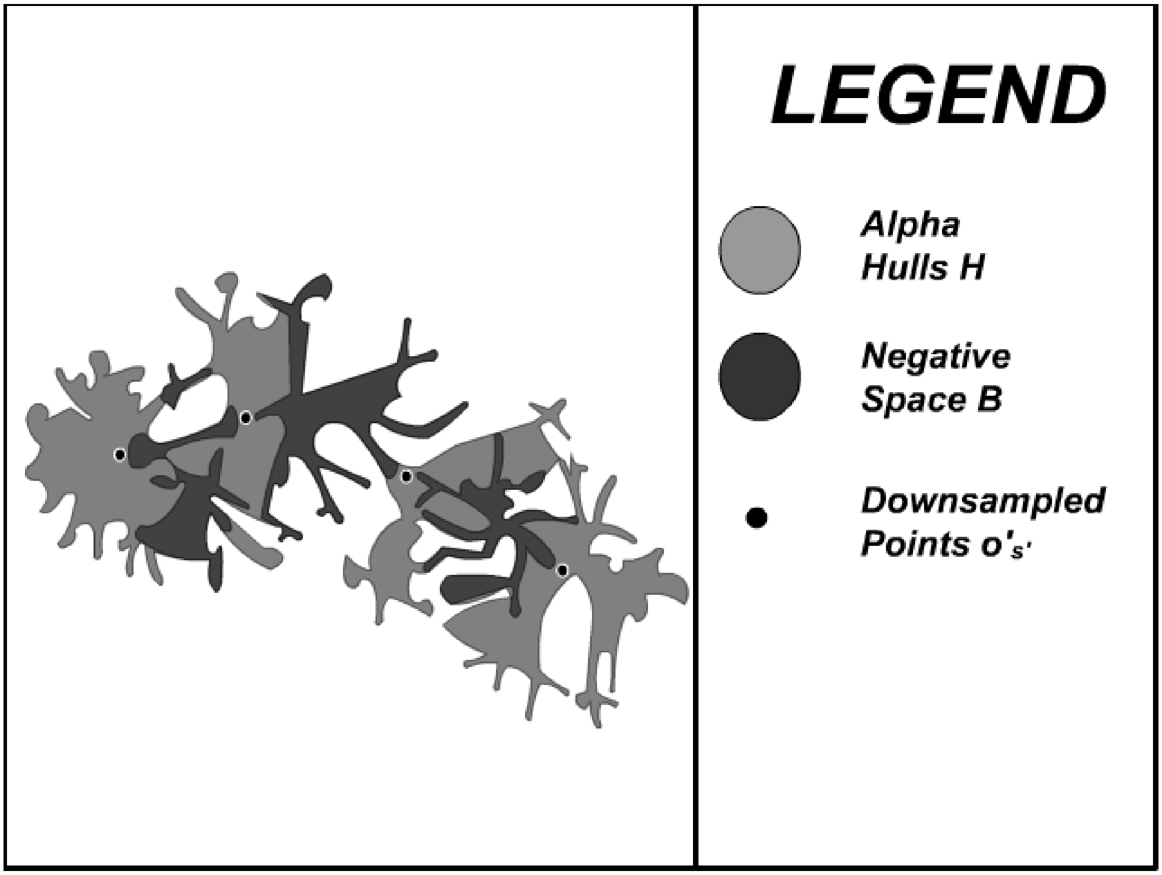


Figure 3.12: The negative space of set B created by five alpha hulls H.

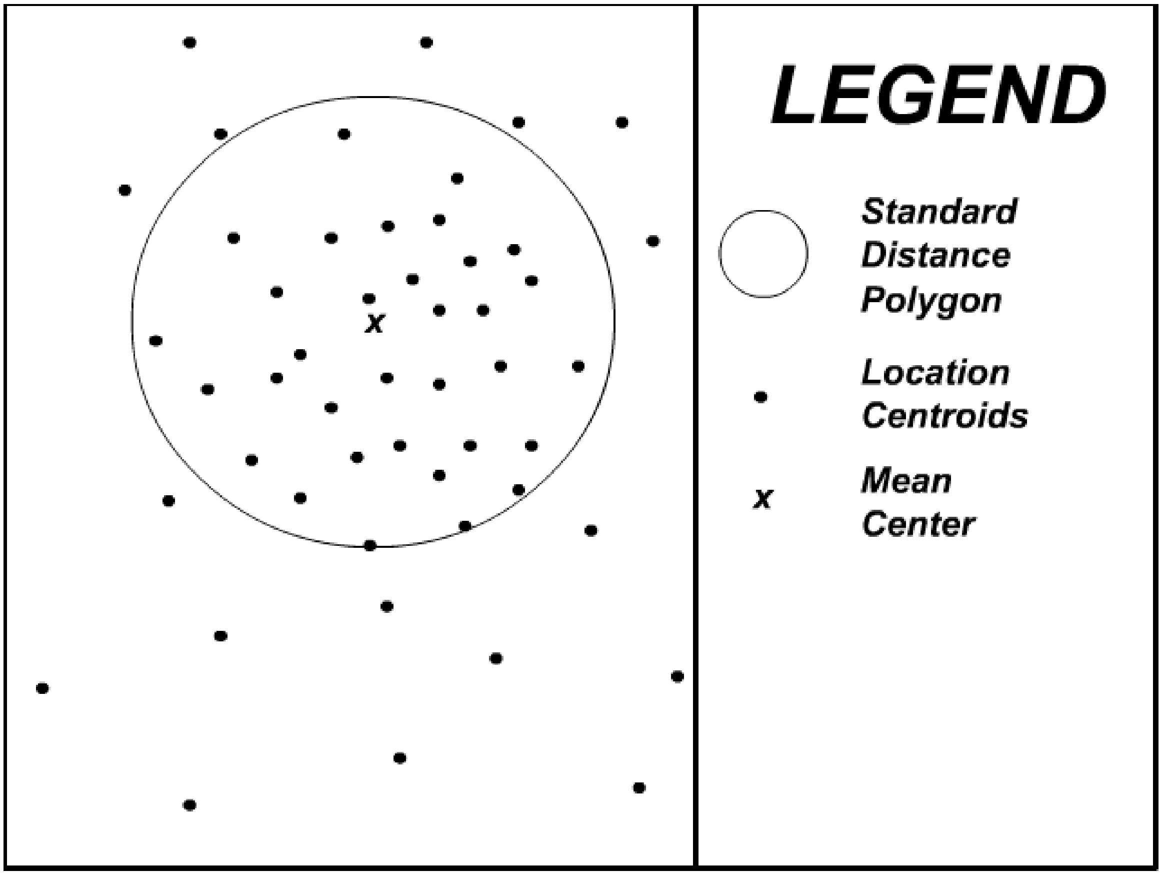


Figure 3.13: The first standard deviation standard distance polygon $a'_{s'}$ for a cluster of building centroids D.

known building locations are exploited in the next section in order to prune the alpha hulls in a meaningful way.

In Section 3.3.1 it was proposed that one of the pre-processing tasks that might be performed on the data-set would be to generate centroids for the facilities, buildings, and other possible destinations. This is done because it is easier for algorithms to operate on points than on polygons due to their reduced dimensionality. The information that is lost due to this abstraction is not serious and well worth the added functionality. When one considers a GIS map with road networks and building centroids, one is struck by the apparent clustering that the centroids are often capable of. This stands to reasons because the expression *urban sprawl*, often used to describe American cities, does not necessarily imply *uniform sprawl*. To wit, buildings cluster according to topology, zoning, proximity to road-networks, and many other variables, with the ultimate result being clusters of communities separated by less densely populated areas.

The polygon $h_{s'}$ may be used to select a set of location centroids $d_{s'} \subseteq D$ that fall within its bounds. It is referred to as function $selectD(D,\ h_{s'})$, and is described by Equation 3.18 as:

$$D_{s'} = D \cap h_{s'}. \tag{3.18}$$

It is proposed next that the relative density of a given set of possible destination $D_{s'} \subseteq D$ is expressed using the standard distance algorithm that is often used by the geospatial analysis community [38], [39]. This bears a close resemblance to typical standard deviation calculations, though Equation 3.19 and Equation 3.20 include a pythagorean calculation for x, and y and is multiplied with 2π to inscribe a circle. For a given set of points in a plane, a mean center $\bar{X}$ and $\bar{Y}$ is calculated and then a mean distance for the n points in $D_{s'}$ from that center. This results in a scalar that may act as a radius that inscribes a circle with the mean center at the origin (see Figure 3.13). The equations for the first and second standard deviation circles are shown in Equations 3.19 and 3.20.

$$a_{s'} = 2\pi \sqrt{\left(\frac{\sum\limits_{d_{s'}=1}^{n} (x_{d_{s'}} - \bar{X})^2}{n} \right) + \left(\frac{\sum\limits_{d_{s'}=1}^{n} (y_{d_{s'}} - \bar{Y})^2}{n} \right)}, \tag{3.19}$$

Note that the second standard deviation equation will merely be Equation 3.19 ×2.

$$a'_{s'} = 4\pi \sqrt{\left(\frac{\sum\limits_{d_{s'}=1}^{n} (x_{d_{s'}} - \bar{X})^2}{n} \right) + \left(\frac{\sum\limits_{d_{s'}=1}^{n} (y_{d_{s'}} - \bar{Y})^2}{n} \right)}. \tag{3.20}$$

where there are n locations $d_{s'} \in D_{s'}$, and $x_{d_{s'}}$, $y_{d_{s'}}$ are the coordinates for every $d_{s'}^{th}$ feature, and

$$\bar{X} = \frac{\sum\limits_{d_{s'}=1}^{n} x_{d_{s'}}}{n}, \tag{3.21}$$

$$\bar{Y} = \frac{\sum\limits_{d_{s'}=1}^{n} y_{d_{s'}}}{n} \tag{3.22}$$

are the coordinates for the mean center of the features. Moreover,

$$A_{s'} = \{a_1,\ a_2,\ a_3,\ \ldots,\ a_{s'}\}, \tag{3.23}$$

and

$$A'_{s'} = \{a'_1,\ a'_2,\ a'_3,\ \ldots,\ a'_{s'}\}. \tag{3.24}$$

where the "prime" notation is meant to distinguish these products from each other and has no correspondence to our previous usage with O, O', S, and S'.

The practical result of this exercise is a disk polygon centered over the densest part of the cluster of centroids. This can be accomplished for the first, second, or third standard deviation (though only the first two are employed here) denoted as

$a_{s'}$ and $a'_{s'}$. The first may be said to contain 68 % of the points, the second 95 %. (These functions will be called $stdDev1(d_{s'})\ and\ stdDev2(d_{s'})$) This is useful when one considers that, when an alpha hull $h_{s'}$ is created for a given $o'_{s'}$, it is merely showing where, in the space, the vehicle might travel in a given T. One might afterwards ask where in $h_{s'}$ does the density of possible locations make it most likely for that vehicle to go? $D_{s'}$ represents the possible locations that the vehicle might actually visit within T. A subsequent application of standard distance to $D_{s'}$ for the first and second deviations yield two disk polygons centered on the statistical mean center of $D_{s'}$. These may, in turn, be used to mask $h_{s'}$ as per Figure 3.14 and computed in Equation 3.25 and Equation 3.26. The result is a two-stage gradient of $h_{s'}$ where it may be said with 68 % accuracy that the vehicle travels within the first standard deviation, and with 95 % accuracy, within the second standard deviation. The two functions will be called $stdDevMask1(h_{s'},\ a_{s'})$ and $stdDevMask2(h_{s'},\ a'_{s'})$ and their respective equations are:

$$h'_{s'} = h_{s'} \cap a_{s'}, \tag{3.25}$$

$$h''_{s'} = h_{s'} \cap a'_{s'}. \tag{3.26}$$

Hence,

$$H' = \{h_1 \cap a_1,\ h_2 \cap a_2,\ h_3 \cap a_3,\ \ldots,\ h_{s'} \cap a_{s'}\}, \tag{3.27}$$

and

$$H'' = \{h_1 \cap a'_1,\ h_2 \cap a'_2,\ h_3 \cap a'_3,\ \ldots,\ h_{s'} \cap a'_{s'}\}. \tag{3.28}$$

Note, again, that the new superscripting convention is employed here. It is convenient to employ the "prime" superscript to describe successive manefestations of the same polygon. For example, H becomes H', H'', etc. It is important that this not be confused with our previous use of the "prime" superscript which was used to differentiate between a tracklet O and its down-sampled counterpart O'.

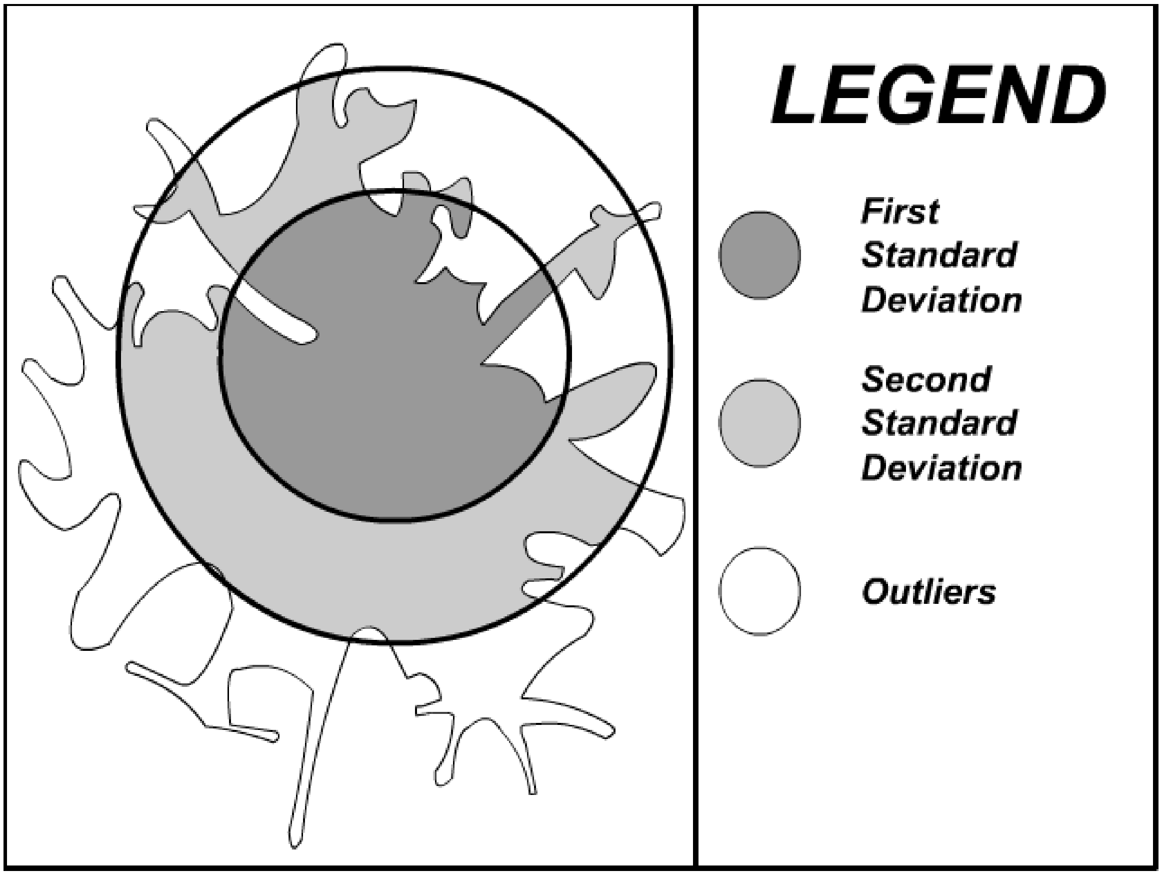


Figure 3.14: The alpha hull $h_{s'}$ masked by the first and second standard deviation standard distance polygons $a'_{s'}$ and $a''_{s'}$ result in clipped polygons $h'_{s'}$ and $h''_{s'}$.

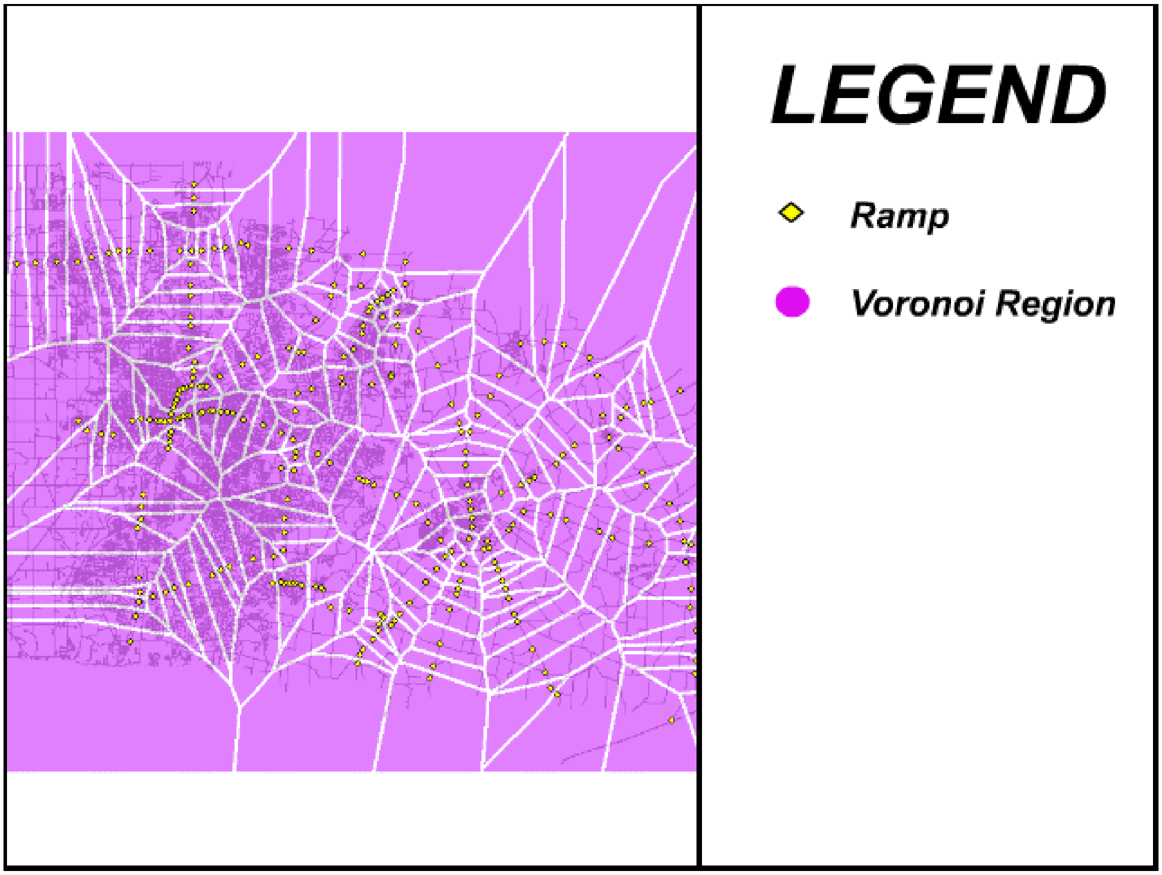


Figure 3.15: The Voronoi regions created from off-ramps over Xenia and Dayton Ohio.

3.4.6 Narrowing the Result with the Bi-Directional Search Tessellation . Until now, several advantages have been reaped through judicious use of the *Destination Directed* assumption, and through the statistical properties of point densities. We have yet to explore our second assumption *Slow-Fast-Slow*, and the possible knowledge that this might yield. Recall that this assumption was based on the observation that many *Deliberate Journeys* follow a sequence where a driver will travel on slower arteries only to enter an expressway, and then to exit to slow arteries again to finish the last leg of the journey. This can be exploited if an event-handler is able to evaluate the events as they unfold. When the handler detects that a vehicle has entered the final *slow* phase of the journey then it might attempt to restrict all searches within the Voronoi region of the exit ramp taken from the main artery. This seems to be a reasonable because our assumptions demand that, if presented with a choice, the motorist will take the exit that is nearest his goal.

The Voronoi tessellation (the dual of Delaunay triangulation), when given as an argument a set of points in a plane, results in polygon cells that encompass all of the area of that plane closer to that point than to any other. The boundaries between these cells are defined by the space in the plane equidistant between two points. As Figure 3.15 depicts, this may be practiced on points representing off ramps of highways with the result that each Voronoi region, represents a space that is closer to that ramp than to any other.

There is a subtle problem with the canonical Voronoi tessellation that is not readily apparent but causes it to fail in our purpose if not remedied. The other processes that have thus-far been described have utilized cost-functions rather than simple Euclidean distance to determine regions, as per the Dijkstra search above. That is because cost in time, rather than distance, is central to our assumptions. It is therefore desirable to contrive a modified tessellation such that the region defines an area of cost around a given off-ramp (we will still wish to call this a tessellation because it is our intent to segment a plane). This may be accomplished exactly as the searches that we have defined previously for alpha hulls where instead of providing points in down-sampled tracklet O', we provide the set of off-ramps. These searches may be allowed to continue until they reach the extent of $G(V, E)$, or until they collide with other regions (like bidirectional search [21]). In this way, all of the space is eventually be attributed exclusively to a given off-ramp point. In homage to the bidirectional search underpinnings of this approach, we will call this a BDS-tessellation.

Another useful refinement that we may add to this is that tessellated regions ought only to be created for a single major road or highway. This precaution prevents the tessellations from arteries travelling east-west and vice versa from interfering with those that travel north-south. The final outcome, as shown by the example for Xenia's US Route 72 in Figure 3.16, will be n sets of tessellations W_n where there are n major

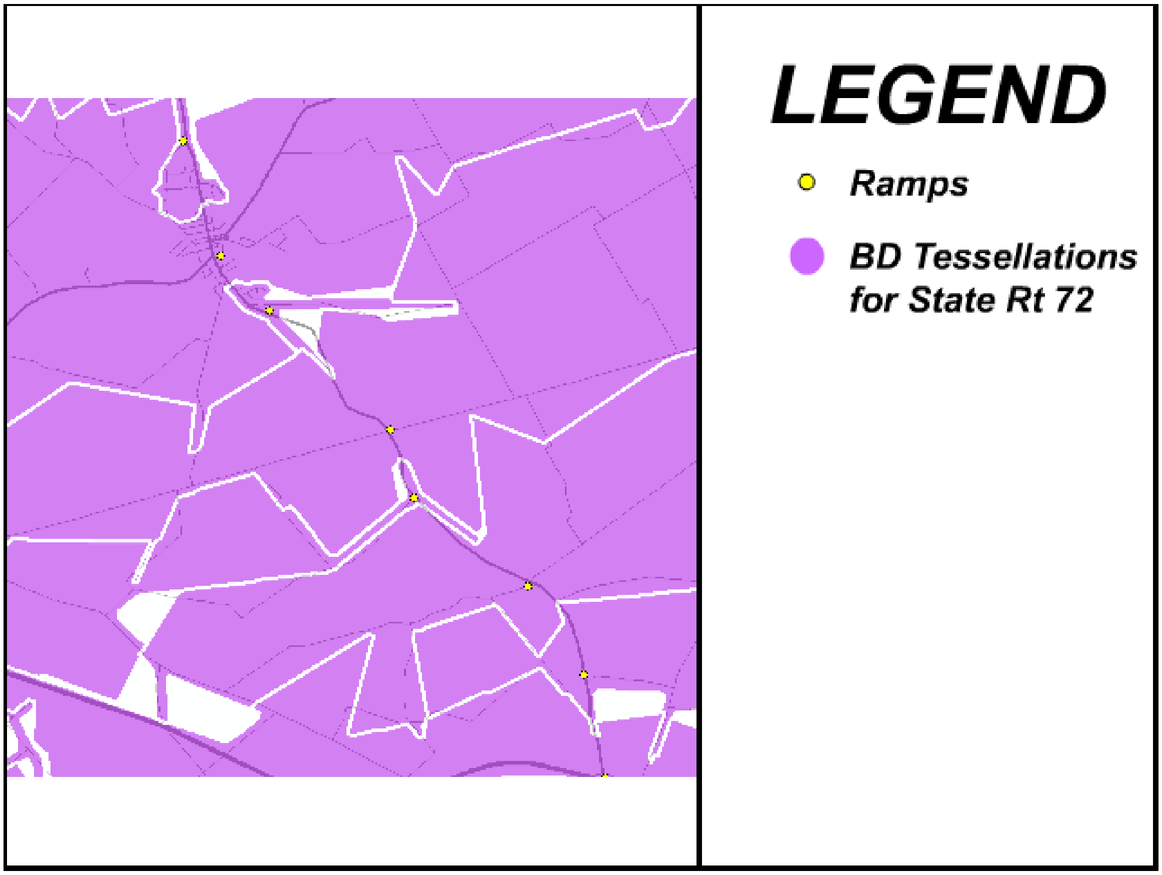


Figure 3.16: The bi-directional search tessellation $W_{US\ Rt\ 72}$ created by searching from each off-ramp until the extent of $G(V, E)$ is reached or a collision with another search occurs. Pictured here is the set of regions for US Route 72 traveling from Xenia Ohio to the southern reaches of the state.

roads and highways. The result, truer to our goals associated with the *Slow-Fast-Slow* assumption, is a string of regions that follow a major artery.

Contingent upon this description, it is expected that if a vehicle exits a major artery, it is because that ramp was the closest one to his intended neighborhood. Given the assumption of *Slow-Fast-Slow*, this is the final leg of his journey, is expected not to exceed the bounds of the polygon $w \in W$. Hence, this polygon becomes an additional constraint that can be used to narrow our search, given that a listener detects our stated conditions.

The state-machine that is called for must be able to listen to the disposition of the vehicle and to detect *slow-fast* and *fast-slow* transitions as they occur. Such a

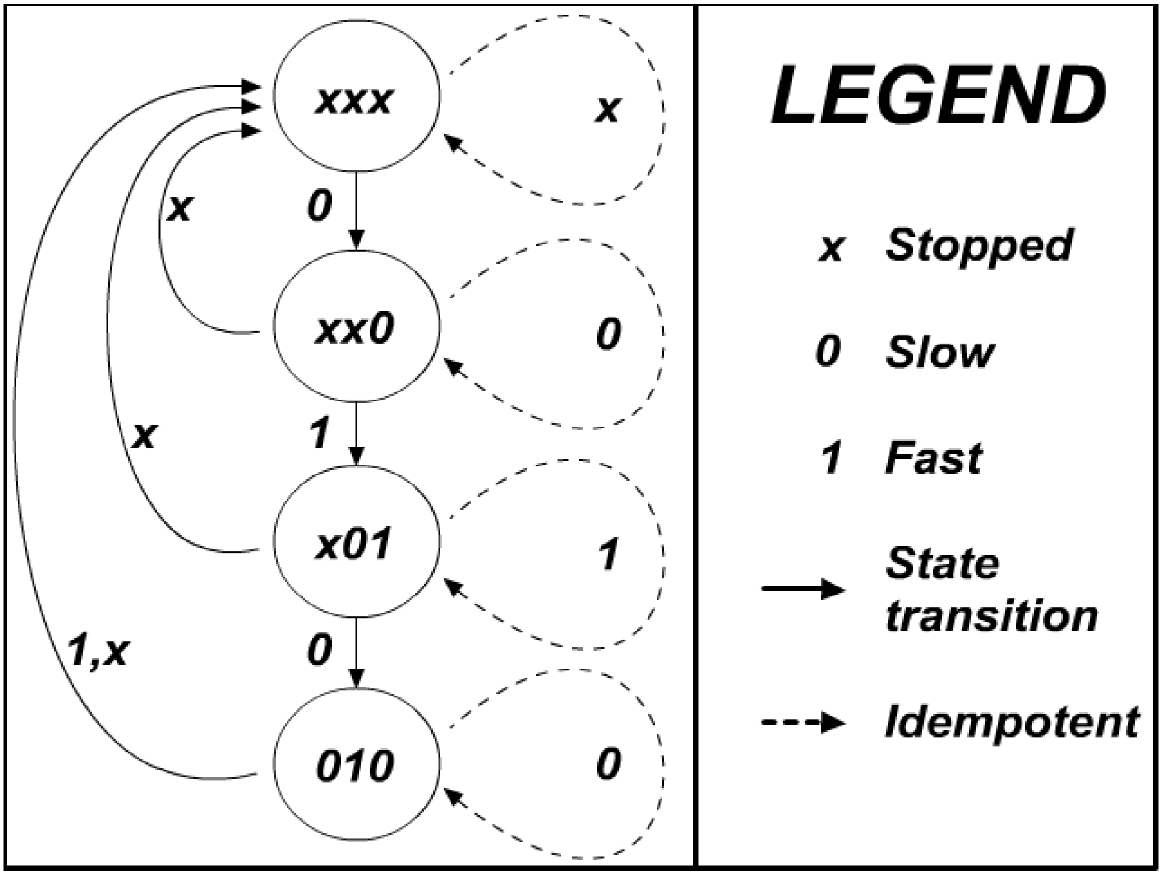


Figure 3.17: Pictured here is the state machine used for detecting the *Slow-Fast-Slow* condition.

construct may take the form illustrated in Figure 3.17. There are four states containing a three-tuple stack. The values for this stack may take the value X, meaning that the vehicle is stopped, 0, for a vehicle on a slow artery, and 1 for a vehicle that is on an express artery. The vehicle will start in the XXX *stopped* state. If it continues to be stopped, the first register will be operated on idempotently and there will be no transition. If it begins to drive, then a 0 will push onto the first position of the register and the state will transition to $XX0$, the *slow* state. It will never be allowed to have a 1 from the XXX position because it is assumed that no car will park along an expressway.

Having transitioned to state $XX0$, *slow*, the state implies that the car had been stopped but is now moving along a slow road. When a turn is made to another slow road, the transaction is again idempotent, and the system remains in state $XX0$. If a turn is made onto a fast road, then a 1 is pushed onto the register and the state transitions to the $X01$ *slow-fast* state. If the vehicle turns into a parking area, then all registers are popped and returned to the XXX state of *stopped*; the journey has ended without having ever been fast.

At the $X01$ *slow-fast* state, the system will again react idempotently if it continues along a fast road. If it turns onto a slow road, then a 0 is pushed onto the stack and the transition is made to the 010 *slow-fast-slow* state. If it stops (in the unlikely event that the vehicle experiences a collision or breaks down) then the journey is over and the state transitions back to XXX *stopped*.

At the 010 *slow-fast-slow* state, the event-handler will execute the tessellation mask on the next alpha hull $h_{s'}$ because the *Slow-Fast-Slow* assumption predicts that the vehicle has reached its objective neighborhood. If another slow road is reached, again the idempotent operation. If it stops, then it transitions back to XXX *stopped*. If it returns to a fast artery then the *Slow-Fast-Slow* assumption must be discarded and the process again transitions to XXX *stopped*. In this case, however, the semantic "vehicle is on a *Circuitous Journey*" is generated meaning that

```
$Function\ tessellation(stateMachine(O',\ G(V,E)),\ H,\ W_n)$
    $If(state(O',\ G(V,E)) =\ 010$
        $Return\ h_{final} \cap W_n$
```

Figure 3.18: $tessellation(stateMachine(O',\ G(V,E)),\ H,\ W_n)$ returns the tessellated polygon for the road and locale where the vehicle has exited an off-ramp.

possibly that the driver is lost, is touring an area, is surveilling an area, or practicing counter-surveillance techniques.

The practical outcome of our state machine having detected the *Slow-Fast-Slow* state of our state machine is that it may now dictate that searches only occur within that tessellated region $w \in W_n$ coincident with the off-ramp. This means that the $w \in W_n$ polygon may be used to mask the alpha hull $h_{s'}$ as the $a_{s'}$ and $a'_{s'}$ polygons did per:

$$H''' = h_{final} \cap w \tag{3.29}$$

Note that H''' is a singleton-set containing only one polygon (unlike H, H', and H''). Hence, there will not be s' of these, but one at the end of the journey. There will be a state machine function to listen to the system, and a function to execute Equation 3.29. We have also chosen to continue the "prime" convention. In this case, the triple prime is intended to denote the final manefestation of polygon H, in this case where it has been intersected with a tessellated region $w \in W_n$. This function is called by another function $state(O',\ G(V,E))$ $tessellation(stateMachine(O',\ G(V,E)),\ H,\ W_n)$, that intersects the proper tessellated region with H per Figure 3.18.

3.4.7 Negative-Space, the Tabu List, and the Generation of Semantics .

The cynic might point out that, should the vehicle being tracked violate *Directed Destination*, or its corollaries, that the algorithm could fail due to the driver reentering regions that have been deemed negative space B. This contingency can be mitigated if a mechanism is put in place that governs the pruning of $G(V,E)$ with negative

```
$Function\ NListener(O,\ B,\ s')$
    $Optimal \leftarrow True$
    $For\ all\ s'$
        $If\ o'_{s'} \bigcap_{s'} B\ \neq\ \phi$
            $G(V,E) \leftarrow G(V,E) \cup \overline{G(V,E)}$
            $Optimal\ =\ False$
    $Return\ G(V,E),\ Optimal$
```

Figure 3.19: $NListener(O,\ B,\ s')$ listens for a violation of the assumptions and handles it by adding the tabu list $\overline{G(V,E)}$ back to $G(V,E)$. It also disallows future use of the tabu list.

space polygons B. If a listener detects that the vehicle has entered a polygon B then the primary assumptions must be surrendered in a way that the set B no longer plays a role in adding points to the tabu list. This is readily accomplished if the tabu list $\overline{G(V,E)}$ is made empty per:

where *Optimal* is a boolean flag that governs the tabu list. When this occurs, the *Optimal* flag will also constrain the tabu list to be re-added to $G(V,E)$ because "all bets are off" as regards the *Destination Directed* assumption. When this scenario occurs, it is not to be regarded as a failure because, though enhanced predictive pruning is no longer possible, the semantic "vehicle is on a *Circuitous Journey*" is created as it was when *Slow-Fast-Slow* was violated in Section 3.4.6. This product may even prove more valuable than the others as it alludes to a possible casing event. Moreover, Dijkstra search products H, H', H'', and O' will still be valuable to Intelligence and Operations functions.

It remains, therefore, to make practical use of our analysis products H, H', H'', H''', and B. First, these products are intrinsically valuable in themselves. A single $h_{s'} \in H$ tells where a vehicle may go within time horizon T. H', H'', and H''' may be thought of as refinements on H. In the absence of any further down-sampled tracklet points o_{s+1}, these products will still be valuable within T and may even be allowed to grow as a function of T, though their value will decay as T becomes large.

Their value increases however with continued tracking until o_{s+n} because the proximity and overlap of H allows for the creation of B which is, in essence, a contextual refinement. B is valuable in that it tells us where the vehicle must not go, given *Directed Destination*. It becomes arguably more valuable when the vehicle enters that space due to the semantic "the vehicle is not on a *Deliberate Journey*."

There are other contextual refinements to be had, namely from the observation spot C. If an $h_{s'}$ is generated, then its area, when compared to the area of observation spot C, will lead to such conclusions as: "the vehicle has a 73% chance of remaining within the confines of C within T." This may be obtained by the simple ratio

$$P(remain) = \frac{\int h_{s'}}{\int C}, \tag{3.30}$$

Similarly, "the vehicle has a 27% chance of exiting the confines of C within T,"

$$P(exit) = \frac{\int (h_{s'} - C \cup h_{s'})}{\int C}. \tag{3.31}$$

Dependent on the needs of the user, such operations may be performed for H', H'' and H'''. We will refer to all such analysis as $generateSemantics(H,\ H',\ H'',\ H''',\ B,\ C)$.

3.4.8 Determining Figures of Merit . Figures of merit that are employed to judge the usefulness of these products can take several forms, though two are suggested. The first, demonstrated by the DTT effort [25], simply takes the ratio of the areas of alpha hull $h_{s'}$ and motion model $m_{s'}$ of Equation 3.7 such as

$$R = 1 - \frac{\int h_{s'}}{\int m_{s'}}, \tag{3.32}$$

where R is the amount of improvement from having used search to predict the vehicle's destination vis a simple motion model. This is essentially a judgement on the relative *smallness* (one of our stated goals) of the space within which it may be predicted where the vehicle can go. This measurement ranges as $0 < R \leq 1$ and the larger it is, the better the measure. It may be done for $H, H',\ H''$, and H'''.

Another figure of merit that speaks more to the accuracy of the prediction method is to ask how many points o_{s+T} actually fell within its predictive polygon. This may be done as

$$F = \frac{\{o_s,\ o_{s+1},\ o_{s+2},\ \ldots,\ o_{s+T}\} \cap h_{s'}}{\{o_s,\ o_{s+1},\ o_{s+2},\ \ldots,\ o_{s+T}\}} \tag{3.33}$$

Like R, this measurement ranges as $0 < F \leq 1$ and the larger it is, the better the measure. Again, this may be done for $H',\ H''$, and H'''.

Small values of R suggests that the search and its assumptions have not been aggressive enough and need to become more so. Small values of F suggests that the search has become too aggressive and needs to be relaxed. Taken together, R and F are highly complementary as high values for both will speak to the optimality and fine-tuning of the process.

3.4.9 Consolidation into a Single Process . It is now possible to consolidate these functionalities into a grand unifying theory. To that end, a simple pseudo-code representation of the process (shown in Figure 3.20), augmented by the flow-chart in Figure 3.21 (an elaboration on our original flow-chart shown in Figure 3.1), will illustrate how the moving parts fit together. Line (2) in the pseudo-code of Figure 3.20 initialize all product sets as empty sets. Lines (3-4) initializes indices s to $|O|$ and s' to 0 (Recall that the indices are $s = 1,\ 2,\ 3,\ \ldots,\ S$ and $s' = 1,\ 2,\ 3,\ \ldots,\ S'$). Line (5) constrains the function to continue until $S = 0$ because, as O undergoes the down-sample process, it is continually shortened and renumbered until it has zero points left. Line (6) constrains it to iterate for all S'. Line (7) copies the first (remaining) point from O and copies it to O', necessitating that S' increment by one in Line (8). Lines (9-11) accomplish the task of creating alpha hull H. Line (12) subtracts from O those points that intersect with H. Line (13) renumbers O after this so that Line (7) may perform its operation again during the next iteration. Line (14) generates motion models M for the calculation of the figure of merit R and also for use as a baseline. Line (15) calculates the negative space product B. Line (16) adds to the tabu list those $V \in G(V, E)$ that intersect B. Line (17) determines the set of building centroids $D_{s'}$ bounded by H. Lines (18-19) determine the standard distances A' and A'' for the first and second deviations, respectively. Lines 20-21 use these and H to create intersection products H' and H''. Line (22) conducts the state machine check to determine a *Slow-Fast-Slow* condition and generates the tessellation product H''' if that condition is found. Line (23) checks for a violation of the *Destination Directed* assumption and, if it detects it, adds the tabu list back to $G(V, E)$ and disallows any new additions to the tabu list (thereby also generating the semantic "Not a *Deliberate Journey*").

Finally, the process can be summed up according to variable, sets, and their purpose. For this, see Table 3.1.

3.5 Chapter III Summary

In this chapter we have proposed a search-based method for predicting where a vehicle may go within a given time-horizon T by exploiting the rich data-sets from commercial and municipal sources. The process relies upon two assumptions, *Directed Destination*, and *Slow-Fast-Slow*. Arguments passed to the process are a graph $G(V, E)$ representing the transportation network, a set of tessellation regions W_n around major off-ramps in that network, a set of location centroids D, an observed tracklet O, and a coverage spot C. The process returns a set of analysis products consisting of output polygons H, H', H'' and H''' that predict where the vehicle will be in T, and a semantic that could inform that "the vehicle is not on a *Deliberate Journey*." A methodology for criticism, in the form of metrics R and F, were described

Table 3.1: Variables and sets of the Destination Prediction Process.

Name	**Equations**	**Purpose**
$G(V, E)$	Defined Section 2.3	The road network expressed as a graph
$\overline{G(V, E)}$	Eqs. 3.16, 3.17	Subgraph of $G(V, E)$, Tabu space
O	Equations 3.4	Observed tracklet expressed as a 4-tuple
O'	Equations 3.5, 3.13	O downsampled by H
C	Defined Section 3.3.3	Polygon disk describing the coverage spot
W_n	Defined Section 3.4.6	The set of n tessellated regions for n roads
D	Defined Section 3.3.1	Set of centroids of all possible locations
$D_{s'}$	Equation 3.18	Subset of D representing $D \cap h_{s'}$
S	Equations 3.4, 3.12	Cardinality of O
s	Equation 3.4	Index of O
S'	Equation 3.5	Cardinality of O'
s'	Equation 3.5	Index of O'
B	Eqs. 3.14, 3.15	Negative space described by $h_{s'-1} \cap h_{s'}$
m	Equation 3.18	Index of $D_{s'}$
$P_{s'}$	Equation 3.6	The set of paths found by $radialDijkstra$ for a point $o'_{s'}$
$V_{s'}$	Equation 3.9	Set of all vertices $V_{s'} \in V$ from a $radialDijkstra$
A	Eqs. 3.19, 3.23	standard distance polygons (σ_1) of $D_{s'}$
A'	Eqs. 3.20, 3.24	standard distance polygons (σ_2) of $D_{s'}$
$\overline{X}$	Equation 3.21	X mean center of $D_{s'}$
$\overline{Y}$	Equation 3.21	Y mean center of $D_{s'}$
M	Eqs. 3.7, 3.8	Set of motion models for all $o'_{s'}$
H	Equation 3.10	Set of alpha hulls for all $o'_{s'}$
H'	Eqs. 3.25, 3.27	Set of alpha hulls masked by A for all $o'_{s'}$
H''	Eqs. 3.26, 3.28	Set of alpha hulls masked by A' for all $o'_{s'}$
H'''	Equation 3.29	Final alpha hull masked by W_n
R	Equation 3.32	Figure of merit: measure of smallness
F	Equation 3.33	Figure of merit: measure of accuracy

```
$Function\ DestinationPrediction(G(V,E),\ T,\ O,\ W, D)$
  $H,\ H',\ H'',\ H''',\ B,\ P_{s'},\ A,\ A',\ D_{s'},\ V_{s'},\ \overline{G(V,E)} \leftarrow \phi$
  $S \leftarrow |O|$
  $S' \leftarrow 1$
  $While\ S \neq 0$
      $For\ S'$
          $O' \leftarrow first\ table\ entry\ in\ O$                         (Equation 3.13)
          $S'\ ++$
          $P_{s'} \leftarrow radialDijkstra(o'_{s'},\ G(V,E), T)$
          $V_{s'} \leftarrow collectPoints(P_{s'})$                          (Equation 3.9)
          $H \leftarrow alphaHull(V_{s'})$
          $O \leftarrow downSample(h_{s'}, O)$
          $S \leftarrow renumber(O,\ S)$
          $M' \leftarrow motionModel(h_{s'}, T)$
          $B \leftarrow negSpace(H)$                                       (Equation 3.14)
          $\overline{G(V,E)} \leftarrow tabu(G(V,E),\ \overline{G(V,E)},\ B)$      (Equations 3.16, 3.17)
          $D_{s'} \leftarrow selectD(D,\ h_{s'})$                             (Equation 3.18)
          $A \leftarrow stdDev1(d_{s'})$                                    (Equation 3.19)
          $A' \leftarrow stdDev2(d_{s'})$                                   (Equation 3.20)
          $H' \leftarrow stdDevMask1(h_{s'},\ a_{s'})$                         (Equations 3.25, 3.27)
          $H'' \leftarrow stdDevMask2(h_{s'},\ a'_{s'})$                        (Equations 3.26, 3.28)
          $H''' \leftarrow tessellation(state(O', G(V,E)), H, W)$ (Equation 3.29)
          $Optimal \leftarrow NListener(O,\ B,\ s')$
```

Figure 3.20: $DestinationPrediction(G(V,E),\ T,\ O,\ W, D)$ combines all algorithms described into a single process.

that judge the smallness of the analysis polygons and their accuracy respectively. In the next chapter it will be shown how this process may be implemented in a GIS integrated development environment.

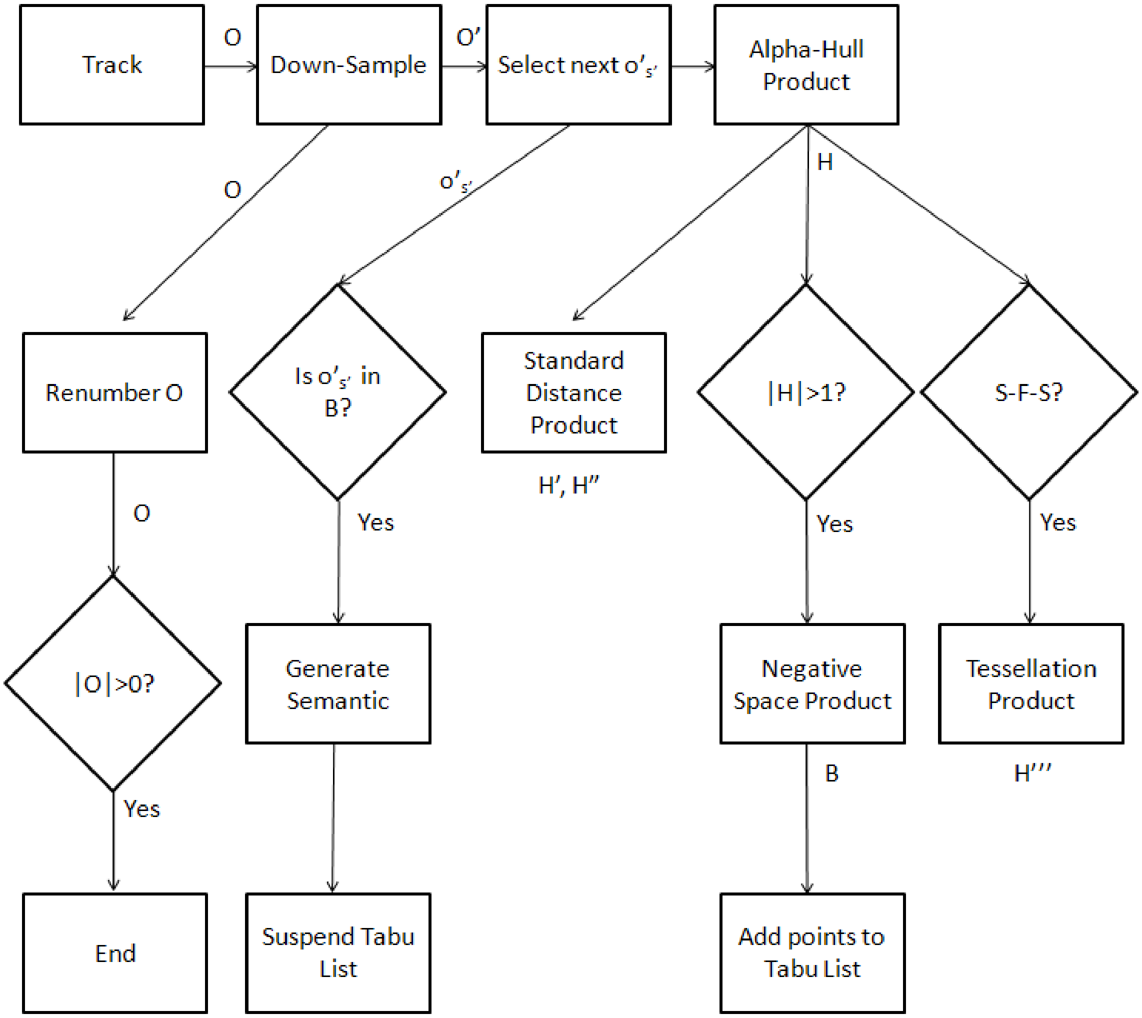


Figure 3.21: A more specific depiction of the flow of the entire process. Compare to Figure 3.1.

IV. Implementation, Assessment, and Results

4.1 Introduction

In this chapter the transition is made from the purely theoretical (as in Chapter III) and the practical, in the form of an actual software implementation. This task involves, first, exploring a suitable GIS development environment for the stated purposes. Second, data-sets have to be obtained from municipal sources for the creation of our graph construct $G(V, E)$, and also using GPS data-logger technology to simulate tracklets O. These data, in raw form, then need to be organized into a useable schema. Third, the algorithms and equations discussed in Chapter III need to be implemented in the GIS development environment. Fourth, the process has to be run, and the results tabulated and stored in an easily presentable format. Fifth, the process runs has to be evaluated by the figures of merit discussed in Section 3.4.8.

4.2 The ESRI ArcGIS Editor® and ArcCatalogue® Sortware Suite

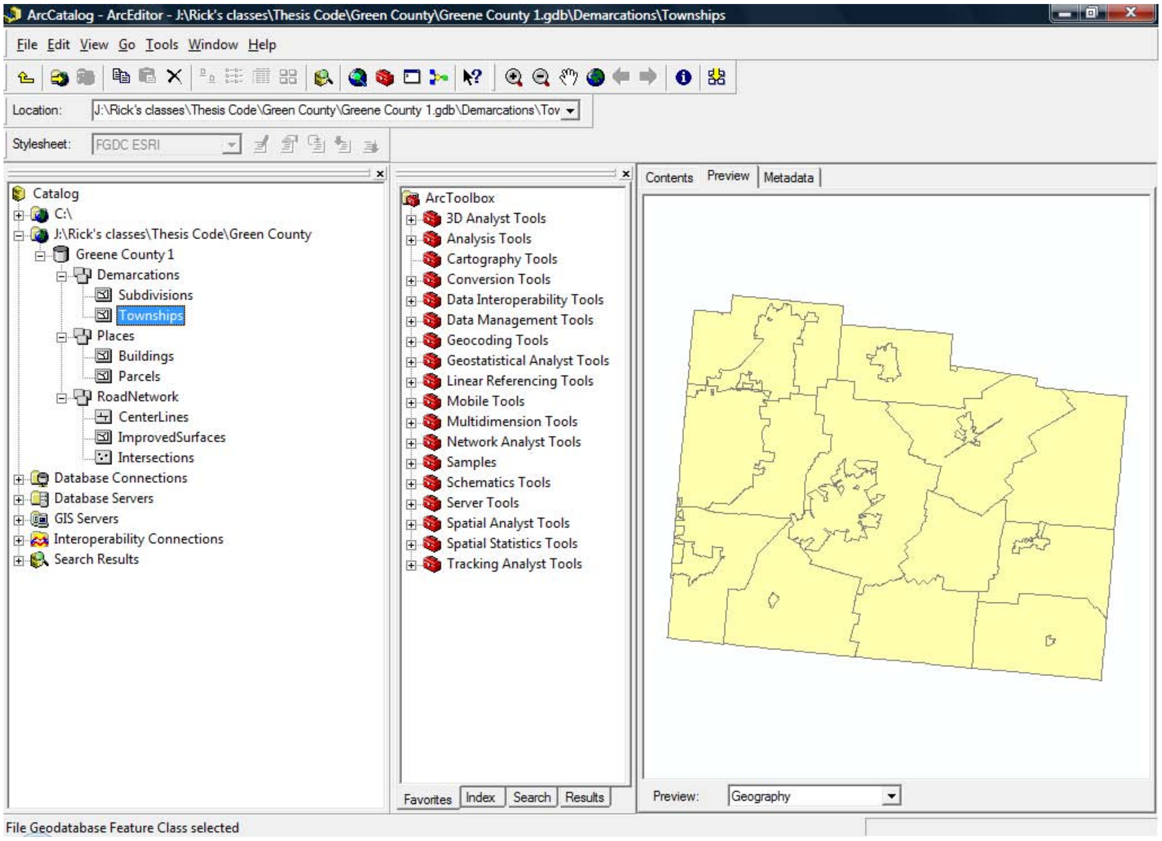
It now becomes necessary to explore a representational framework for the system which this thesis proposes to exploit and to implement the computational elements. It has been stated in Chapter I that our domain of interest is twenty city blocks to an entire municipal area encompassed by the meso and micro-scales there defined. It has also been proposed that the area of concern should be the urban environment. To that end, it is fortunate that, world wide, municipal planners have begun to represent their respective domains with Geographic Information Systems (GIS). In order to exploit this capability, this thesis will rely upon the ESRI ArcGIS Editor®.

GIS, at its most basic level, consists of data-tables containing latitudinal, longitudinal, and elevation information on features in a geographic extent. Based upon vector graphics, the most primitive form that this data can take is that of the point. The point is surveyed with GPS technology and then (ideally) post-processed with differential GPS correction software that gives it sub-centimeter accuracy. Salient examples might be geographic benchmarks, wells, and utility distribution man-holes. Points are subsequently able to be represented as lines (isolines, actually) that are simply points that have been connected, either by straight lines, or by Bezier curves. Examples might include the centerlines of roads, telecommunications distribution networks, and geographic boundaries. Predictably, the final geometry that these features can assume is that of the polygon, consisting of areas bounded by lines. Examples include land parcels, townships, cities, counties, states, and countries. All of these features may be attributed with meta-data such as names, capacities, and other semantic information that adds to the richness of the data-set. It is important to note that, although differential-corrected GPS is ideal, sometimes these data-sets are derived from legacy CAD drawings, ortho-imagery, and coverage area maps. It is therefore pertinent that two attributes for the data are the *lineage*[1] of the information (where did it come from?) and also its *tolerance*[2].

The ESRI suite employed for this thesis divides the functionalities of GIS into two main areas: database management, and presentation/geoprocessing. The former is handled with the ArcCatalog® and a brief digression into its layout is important for understanding this chapter. This application is the front-end for creating and managing GeoDatabases. Figure 4.1 illustrates one such structure for Greene County Ohio [40]. Nested within this structure are three sets of data tables (called feature data-sets) that contain shape-files representing different geographic structures. Shape-files may take the form point, line, or polygon. In this example, shape-files are nested with others that are similar inside feature data-sets. Hence, the RoadNetwork data-set contains shape-files labeled “ImprovedSurfaces”, “Centerlines”, and “Intersections”. The first is a polygon geometry, the second is a line geometry, and the third is a point geometry. Note that a fourth datastructure may exist within this construct called a *relationship class*. This has no geometric representation but is a table that allows different shape-files to be connected to each other by specific fields, either with

[1] The lineage of a data-set includes the source of the map, transformations that have been applied, and presentation specifications.

[2] The tolerance of a data-set specifies how accurate the position measurements are for the shape-files and their objects



Location: J:\Rick's classes\Thesis Code\Green County\Greene County 1.gdb\Demarcations\Tov

Stylesheet: FGDC ESRI

Preview: Geography

File Geodatabase Feature Class selected

Figure 4.1: ArcCatalog®: used for creation and management of geodatabases.

a one-to-one, one-to-many, or many-to-one cardinality. This functionality figures prominently in later discussions.

There is a presentation front-end for the ESRI GIS suite called ArcEditor®. Figure 4.2 illustrates *feature classes* represented as layers in the environment. The intuitive motivation for this is to invoke the old business model as an example. Before CAD and GIS, engineers represented coincident architectures as drawings on translucent sepia paper that could then be layered over each other, yielding relational semantics by their superposition. Similarly, representing our *feature classes* as layers allows us to include or omit those data-sets which are pertinent, zoom in to a relevant extent, and edit the points, lines, or polygons. A useful feature in this environment is the *editor* which allows rotational, translational, scaling, and other transforms to be executed on the features. When the edits are changed, the changes are *written to the data-set*. Additionally, a rich set of geoprocessing tools is included

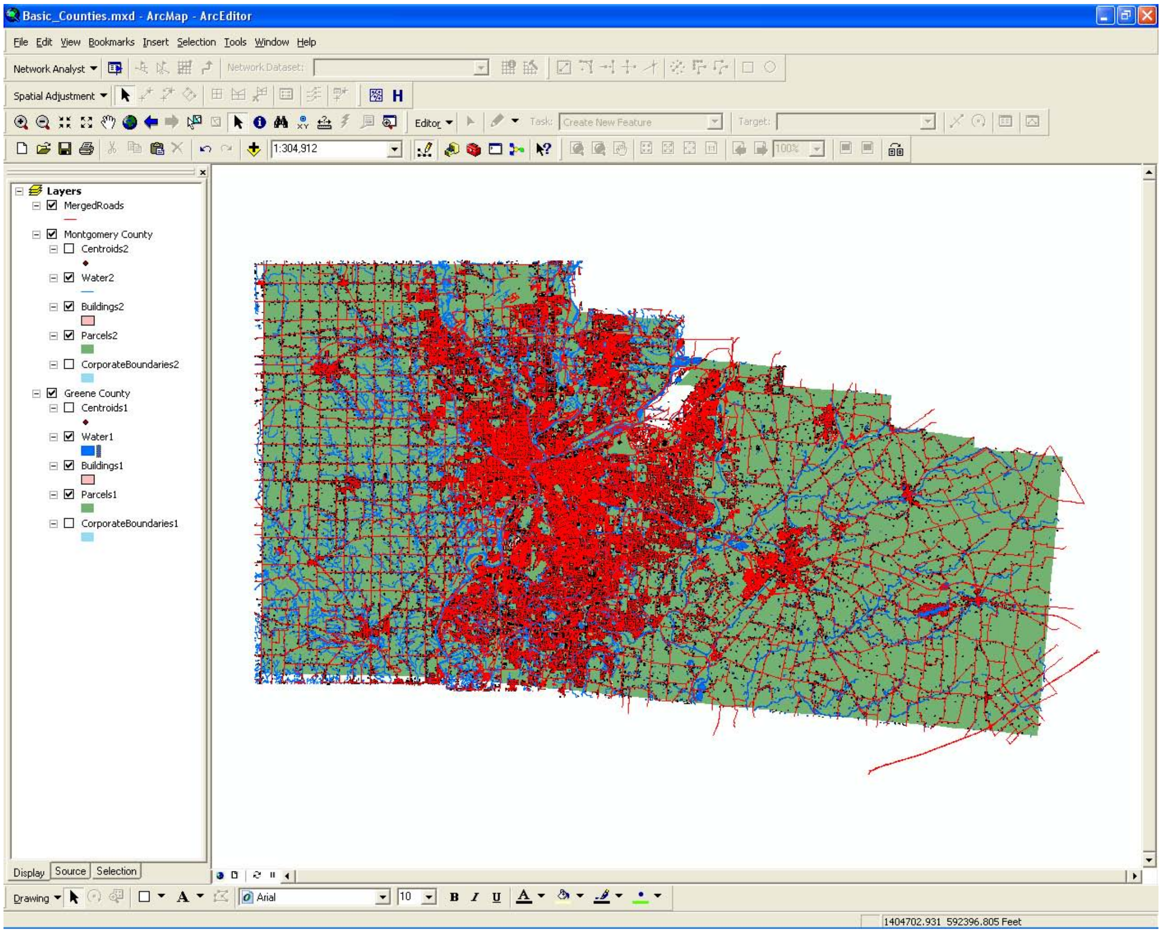


Figure 4.2: ArcEditor®: used for presentation and geoprocessing of geographic data-sets.

in this environment that allow for more sophisticated operations such as statistical clustering, network analysis, and 3D analysis. These functionalities may be used to build processes in a flow-control environment called Model Builder as is shown in Figure 4.3. This environment bears much in common with Matlab Simulink® and Labview® where functionalities are bundeled in icons that may be dragged into the environment, attributed, and "wired" to other data or functionalities. In Figure 4.3, the blue icons are data layers, the yellow icon is bundled functionality, and the green icon is an output data layer. Additional ad hoc functionality may be added by the user by writing Python® scriptlets that, when properly interfaced, behave as the native functionalities do. For our purposes the Model Builder is a prototyping environment used for demonstrating the basic concept. In order to provision a commercial product, one could employ the ArcEngine SDK tools, which, as part of an IDE that also includes Microsoft Visual Basic Studio® may facilitate a robust software-development and packaging capability.

Finally, some important conventions are in order to guard against confusion. Having moved from the theoretical (Chapter III) to the practical (this chapter) we are obliged to frame our discussion in the nomenclature of software and files rather than of set-theory. To that end, functions and variables described in Chapter III are left *italicized*, as they have been, and objects within ESRI Model Builder such as feature data-sets, variables, and shape-file products (*.shp) will be **boldfaced**. Also, rather than refer to singleton polygons h_1, h_2, h_3, $\ldots$, h_s belonging to feature data-set **H**, it will make more sense from a database perspective to say: **h_(1).shp**, **h_(2).shp**, **h_(3).shp**, . . . , **h_(s′).shp** by which we will mean the 1^{st}, 2^{nd}, 3^{rd}, through the s'^{th} shapefile are contained in the **H** feature data-set. Similarly, individual fields within a shape-file are referred to as **O_[1]** , **O_[2]** , **O_[3]** , . . . **O_[s]**. The difference between these two conventions is that some products may be saved as individual shape-files within a feature data-set (the former) and some may be saved all in one shape-file (the latter). The reason for the different modes of storage is purely operational: it is easier to pick a single shape-file, say **h_1.shp** from its fea-

Table 4.1: Shape-files and feature data-sets.

Theory	**Software**	**Type**	**Contains shape-files**
$G(V, E)$	**G_V_E**	FDS	**V.shp** , **E.shp**, **E-V.shp**
$\overline{G(V, E)}$	**V_tabu_(s’).shp**	SF	
O	**O.shp**	SF	
O'	**O_prime.shp**	SF	
C	**C.shp**	SF	
W_n	**W_(road)**	FDS	**W_Rt72.shp**, etc.
D	**D.shp**	SF	
$D_{s'}$	**D_temp.shp**	SF	
B	**B**	FDS	**b_(s’).shp**
A	**A**	FDS	**a_(s’).shp**
A'	**A_prime**	FDS	**a_prime_(s’).shp**
M	**M**	FDS	**m_(s’).shp**
H	**H**	FDS	**h_(s’).shp**
H'	**H_prime**	FDS	**h_prime_(s’).shp**
H''	**H_prime_prime**	FDS	**h_prime_prime_(s’).shp**
H'''	**H_prime_prime_prime.shp**	SF	

ture data-set **H** then it would be to select if from a shape-file that contained other polygons.

The discussion presented in the current chapter moves between theory (as discussed in Chapter III) and practical, and this allows for better distinction between the two. Table 4.1, comparable to Table 3.1, is included in order to alleviate the possible confusion that this switch may occasion. Note that some objects are of type SF, meaning shape-file, and that others are of type FDS, meaning feature data-sets.

4.3 Data Collection and Preparing the Data-Set

As set forth in Chapter III, it is necessary to obtain a GIS data-set of a municipality containing roads, buildings, and other infrastructure from either commercial or municipal sources. The work also requires some tracklets, and a coverage spot. The latter may be easily created in the environment from engineering specifications of ISR platforms (generalization from several is employed here, an observation spot

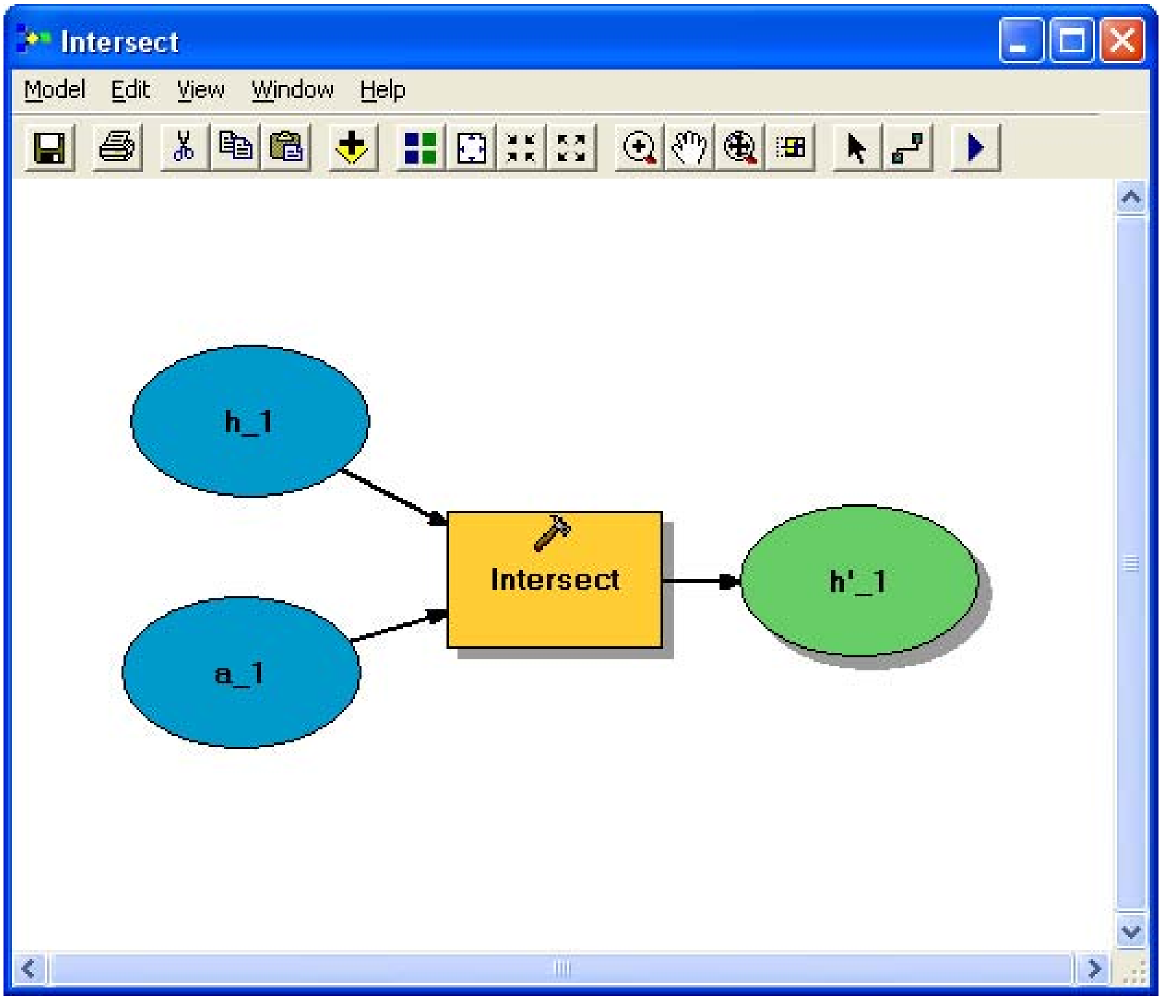


Figure 4.3: Model Builder: used inside ArcEditor® for building processes from native geo-processing functionalities.

of radius 2 km). The first two, however, present various challenges, and solutions to these challenges call for some elaboration.

4.3.1 Obtaining and Preparing Municipal Data-Sets . In the United States municipal data-sets are collected, maintained, and made freely available to the public at the county level of government. A typical county data-set contains road networks to include road centerlines and road polygons. They also contain polygons for buildings, bodies of water, land-parcels, and municipal corporate boundaries. These will cover the extent of the county, which for Ohio, is on average $900-1000$ km^2. Unfortunately, this means a roughly 30 km X 30 km square within which tracklets may be observed and travelling within such an extent might incur the criticism that either the tracklets are too short in duration or that they are overly contrived because they begin and end at the extrema of the county. The remedy for this is to join two county data-sets, and for our purposes, this meant the neighborhood of Dayton Ohio which includes Greene County and Montgomery County [40], [41]. Since it is the intent of this thesis to retain a thoroughly operational flavor, it bears mentioning that data collection in possibly hostile countries is not as easy. (It is proposed that hyperspectral imaging and feature-segmentation will readily accomplish the data collection if traditional sources are not available [11]). However, the county data will suffice for our demonstration.

The fact that there are two distinct data-sets forces a merger of two different schemas which presents a complication: there are no state or federally mandated standards and so different data-sets, though having much in common, are nevertheless often mismatched. For the purposes of this thesis, a schema that is as simple as possible will be preferred. Much of the columnar fields in the data-sets obtained from Greene and Montgomery county is for management and upkeep of the road network and therefore of little use to our purpose. Having discarded that which is of no use to our purpose, and having merged those columnar fields which are the same (though often named differently) the result is a schema that is employed to build the road

Table 4.2: GIS schema for a road network.

ID	**SL**	**LF**	**LT**	**RF**	**RT**	**CFCC**	**Name**	**Feet**	**Min**	**H**	**Z1**	**Z2**
1	25	301	399	300	398	A40	Walton St	420.04	0.1909	3	0	0
2	25	858	888	857	887	A40	Cherry St	568.60	0.2584	3	0	0
2173	25	0	0	0	0	A63	Ramp	527.65	0.2398	3	0	1
4191	65	0	0	0	0	A10	I 675	11945.05	2.0884	1	0	0
4191	65	0	0	0	0	A10	I 675	1831.54	0.3202	1	0	1
4334	55	425	435	426	435	A21	St Rt 235	1260.11	0.2605	2	0	0

network $G(V, E)$. Table 4.2 is representative of the Greene-Montgomery road line segments $E \in G(V, E)$. (Vertices V simply contains x and y coordinates).

Table 4.2 is representative of different types of roads and a brief list that describes their contents aids in understanding how $G(V, E)$ is created later.

- **ID**: Object ID. The key-field of the table.
- **SL**: Speed limit for that road segment.
- **LF**: Left from.
- **LT**: Left to.
- **RF**: Right from.
- **RT**: Right to.
- **CFCC**: Census feature class codes. Code for typifying road-type.
- **Name**: Name of the road segment.
- **Feet**: Length of the road segment in feet.
- **Min**: Cost Ω to traverse the road segment in minutes (per Equation 2.2).
- **H**: Hierarchy - 1 = highways, 2 = major roads, 3 = local roads.
- **Z1**: Start elevation.
- **Z2**: End elevation.

A few of the previously listed descriptions should have some more explanation. The fields “Left from,” “Left to,” “Right from,” and “Right to” are manifestations of the linked-list morphology alluded to in Section 3.3.1 and allow road segments to have predecessor and successor road segments. That there are two "lefts" and two "rights" is a testament to the two-way nature of many roads. (Those that are not will have nulls as values). Note that ramps and major highways in the data-set do not have these values. This is because their geometric structure is more naturally contiguous, and hence, less ambiguous than the municipal grid.

The CFCC value [42] is a US Census Bureau codification for infrastructure and is used in many GIS data-sets. The values are important to us because they inform whether the street is a bike-trail, alley, local road, major road, state route, US Highway, or a number of other possibilities. Table 4.3 illustrates all CFCC codes encountered in the Greene and Montgomery Counties data-sets. (Among other uses, this will be helpful in locating the ramps for the tessellated regions).

The $Z1$ and $Z2$ values are elevation data whose range may be $\{0,\ 1,\ 2\}$ and these are used to inform connectivity. Most road segments will have a 0 value meaning that they are at ground level. Those with Z fields with values of 1 and (extremely rarely) of 2 mean that they are elevated in the Z dimension. The practical advantage of this is that overpasses are able to be distinguished from those roads that pass underneath them. Otherwise, there would be no way to determine whether two perpendicular road segments were an intersection or an overpass when creating $G(V, E)$. For the Greene and Montgomery data-sets, these need to be identified because this data was not included. An effective approach is to create the fields, allow their defaults to be 0, and identify overpasses from their names and from visual inspection and to subsequently attribute the fields.

The Min value will be among the most important to us as it is a measure of cost Ω based upon Equation 2.2 to traverse that road segment in minutes. Note that these values were actually a part of the original data-sets, a fact that speaks to their universal applicability. The H value is a generalized hierarchy that seeks to classify roads (by their CFCC value) according to whether they are local roads, major roads or

Table 4.3: Partial CFCC Table.

CFCC	**Description**
A10	Primary road with limited access or interstate highway
A15	Primary road with limited access or interstate highway, separated
A20	Primary road without limited access, U.S. and State highway
A21	Primary road without limited access, U.S. and State highways, unseparated
A30	Secondary and connecting road, State and county highways
A40	Local, neighborhood, and rural road, city street
A41	Local, neighborhood, and rural road, city street, unseparated
A50	Vehicular trail, road passable only by four-wheel drive (4WD) vehicle
A60	Road with characteristic unspecified
A63	Access ramp, the portion of a road that forms acloverleaf or limited access
A73	Alley, road for service vehicles, usually unnamed, located at the rear of buildings

highways. This value was not an original part of the Greene or Montgomery data-sets and was created for presentation purposes that will be described shortly.

It bears mentioning that none of these fields were completely filled for each road segment. This may be attributed to the relative newness of GIS technology, the colossal scale of the data, and the inevitable budgetary shortfalls that confound all government organizations. Still, there is enough overlap in adjacent fields that allow reasonable deductions to be made on their behalf. For the Greene-Montgomery data-set, a script was built that, in the event that the speed-limit value was missing, would insert the average speed-limit for that CFCC into the field. A similar script was able to furnish the H values. The Feet and Min fields were similarly complementary.

4.3.2 Creating a Road Network $G(V, E)$. After having combined the Green and Montgomery road network data-sets, and attributed missing fields, it becomes a simple matter to create a shape-file network $G(V, E)$ using the ESRI *Network Dataset Wizard*. This wizard may be launched from ArcCatalog® and it accepts a line shape-file as its input. After importing the Greene-Montgomery road network into the wizard, *connectivity rules* are applied such that the road segments are made into edges E per their end points. Additionally, the Z fields are selected to grant elevation data as previously described. Next, the Min field is selected for the cost Ω of

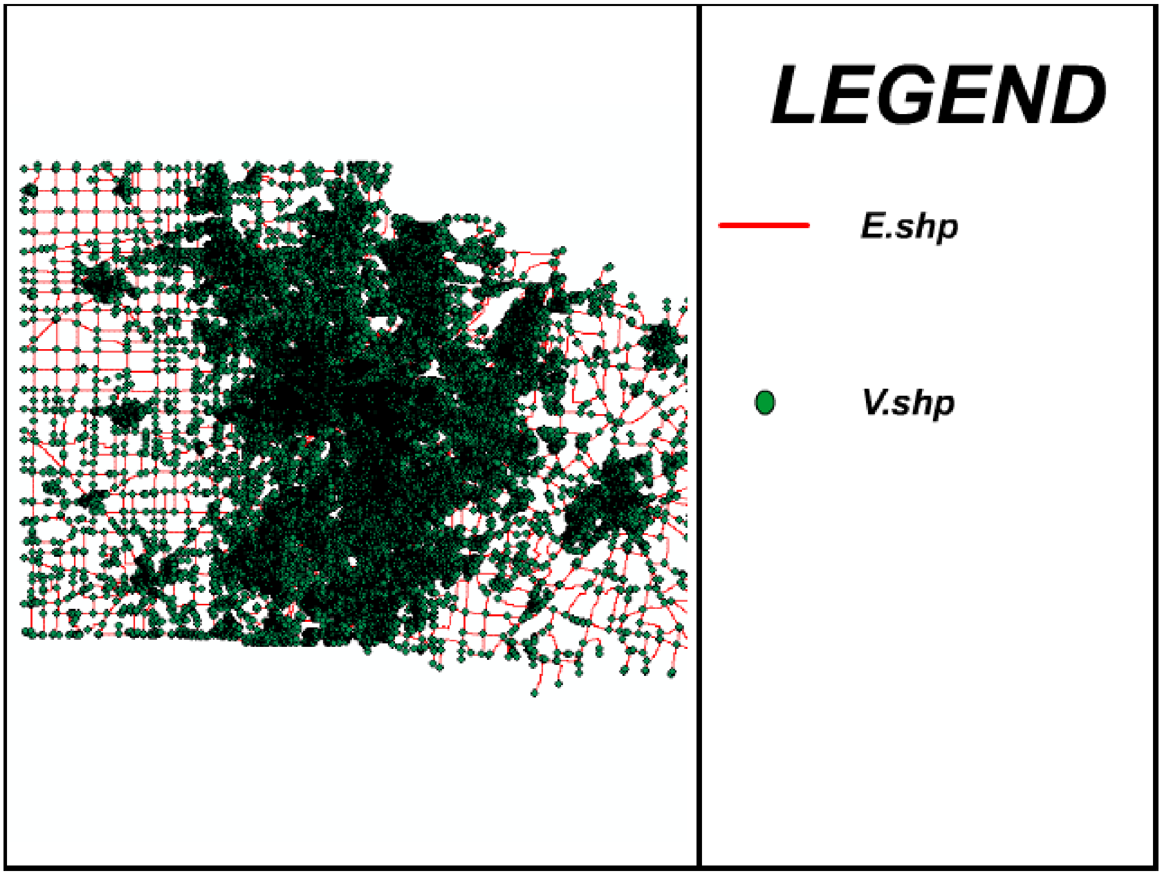


Figure 4.4: Graph $G(V, E)$ for the Greene-Montgomery data-set. The central density is Dayton Ohio and its suburbs.

edges E. Additional rules may then be chosen for adding an impedance for left-hand turns, thereby accounting, in part, for intersections. For the Greene-Montgomery road network, a 15 *sec* penalty was added for left turns so that Equation 2.6, which stipulated a cost $\Omega = \Omega_{road} + \Omega_{intersection}$ is invoked, though it is simplified to mean $\Omega = \Omega_{road} + \Omega_{left}$. The result, per Figure 4.4 and a close-up view in Figure 4.5, is a graph $G(V, E)$ created from the original data-set. Physically, this amounts to a point shape-file for V named **V.shp**, a line shape-file for E, **E.shp**, and a relationship-file that defines connectivity and we refer to it forthwith as **G(V.shp, E.shp)**. The schema for **E.shp** is as shown in Table 4.2 and the schema for **V.shp** contains object ID, lattitude and longitude. The relationship file, **E-V** preserves the $E - V$ link information and cost Ω values for **E.shp**.

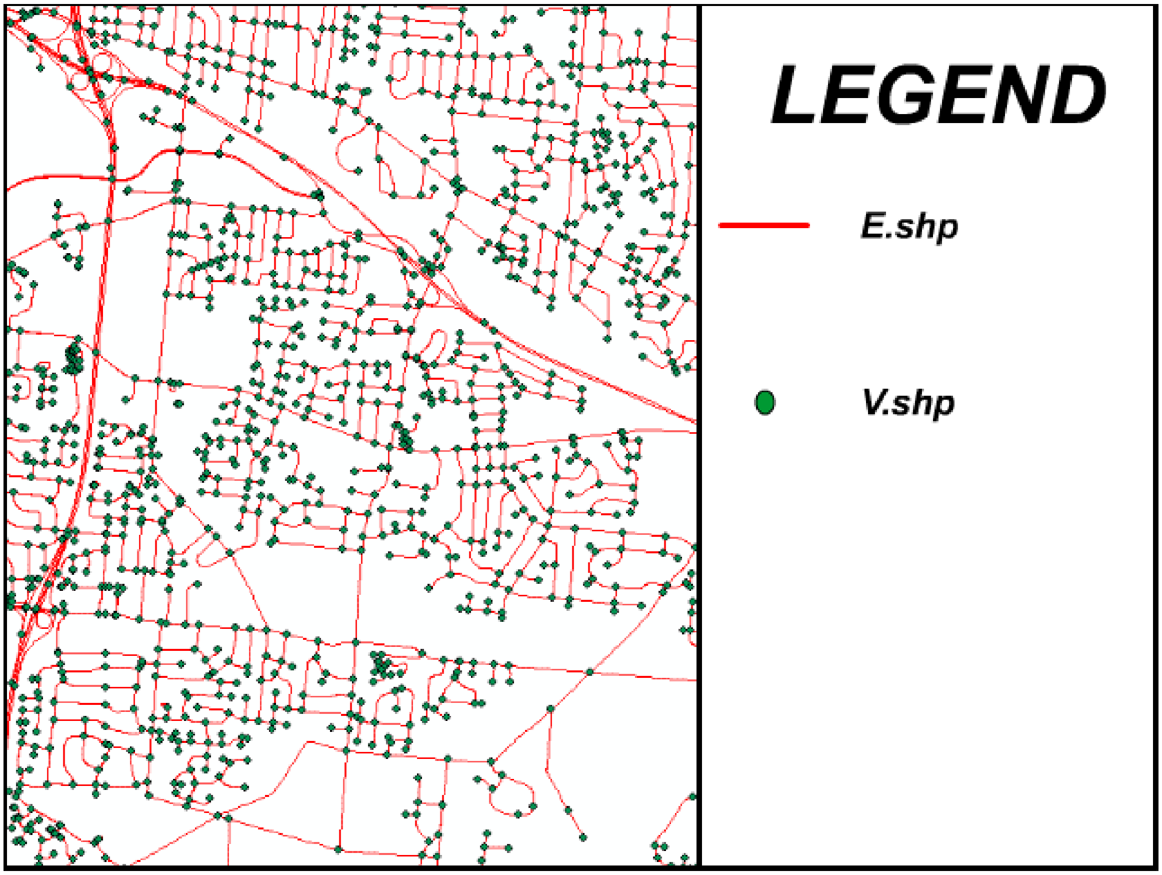


Figure 4.5: A close-up of graph $G(V, E)$ shows individual vertices V and edges E.

Table 4.4: GIS Schema for a Tracklet.

ID	Lat	Long	Speed (MPH)	Time (s)	Index
364	39.712385	-84.039297	55.734	836	836
365	39.712301	-84.039029	55.425	837	837
366	39.712132	-84.038495	54.960	839	838
367	39.712049	-84.038230	54.669	840	839
368	39.711803	-84.037442	54.000	843	840

4.3.3 Simulating an Observed Tracklet O. The next requirement was the tracklet O. Of course it will be impossible to recreate an ideal tracklet from the perspective of an actual airborne surveillance platform (such an item being necessarily scarce) and we will have to delude ourselves as to the real-time component. A reasonable approximation was made using a GPS tracker to log coordinates while driving in the Dayton area. A Leica GPS1200 Surveying System (see Figures. 4.6) was borrowed from the Air Force Institute of Technology *Advanced Navigation Technology* (ANT) laboratory for this purpose. Figure 4.7 shows how a GPS antenna was able to be fastened, via a magnet-mount, to the roof of the vehicle. The ANT lab also donated a GPS base-station to the effort so that differential GPS correction could be accomplished. The Leica GPS1200 logged points at 1 Hz (our assumed ISR sample-rate) and stored them as comma-delimited text files that were then post-processed, along with the base-station files, using Waypoint Products Group's *GrafNav* software. The result, per survey instructions input to the Leica GPS1200, was data corresponding to our 4-tuple in Section 3.3.2. This was x and y position, v velocity, and timestamp, previously defined as x_s, y_s, v, and t. This final product was easily imported into shape-files in ESRI ArcCatalog®. An example for the schema is shown in Table 4.4 (Note that the Time field has been rounded to the nearest second). (Note in the table that the Time does not proceed exactly at 1 Hz. This is due to occasional occlusions that prevent a reading. This is actually useful in that it affords a more realistic model).

Since the main object of this thesis is to predict possible destinations for a vehicle, given the assumption that the vehicle is embarked on a *Deliberate Journey*, nine trips were made and logged. The journeys were designed to cover the breadth and depth of the data-set space of Greene and Montgomery counties, and to range between 5 - 30 minutes in duration. They were allowed to traverse sparsely inhabited areas as well as dense urban areas. In order that the *Destination Directed* assumption would not be in doubt, the routes taken were planned in advance by a Garmin GPS navigator. Finally, in order that the tracklets are not viewed as being contrived to support the thesis, the journeys logged were either actual errands undertaken by the author, or trips between the TechEdge offices in Dayton and various McDonalds® restaurants in the data-space. These is all referred to (generically) as **O.shp** and is stored in the **O** feature data-set. A list of the tracklets are below.

- Xenia McDonalds 1 - AFIT
- TechEdge - Xenia McDonalds 2
- Tech Edge - Vandalia McDonalds
- Huber Heights McDonalds - TechEdge
- Jamestown McDonalds - Xenia McDonalds 1
- Author's Home - Montgomery County Animal Shelter
- Fairborn McDonalds - Jamestown McDonalds
- AFIT - TechEdge

4.3.4 Necessary Pre-Processing of the Data-Set . It was stated in Section 3.3.1 that some pre-processing is important for the computational tasks that have been set forth. In particular, there is a requirement for a centroids file (**D.shp**) generated from the building polygon shape-files, and there will also have to be a tessellated regions file (**W.shp**) created from the off-ramp locations. Fortunately, the GIS community is of a very open-source and collaborative character, and these functionalities are numerous and freely available. The first of these tasks, centroid generation, was the easiest and required nothing more than inputting the building

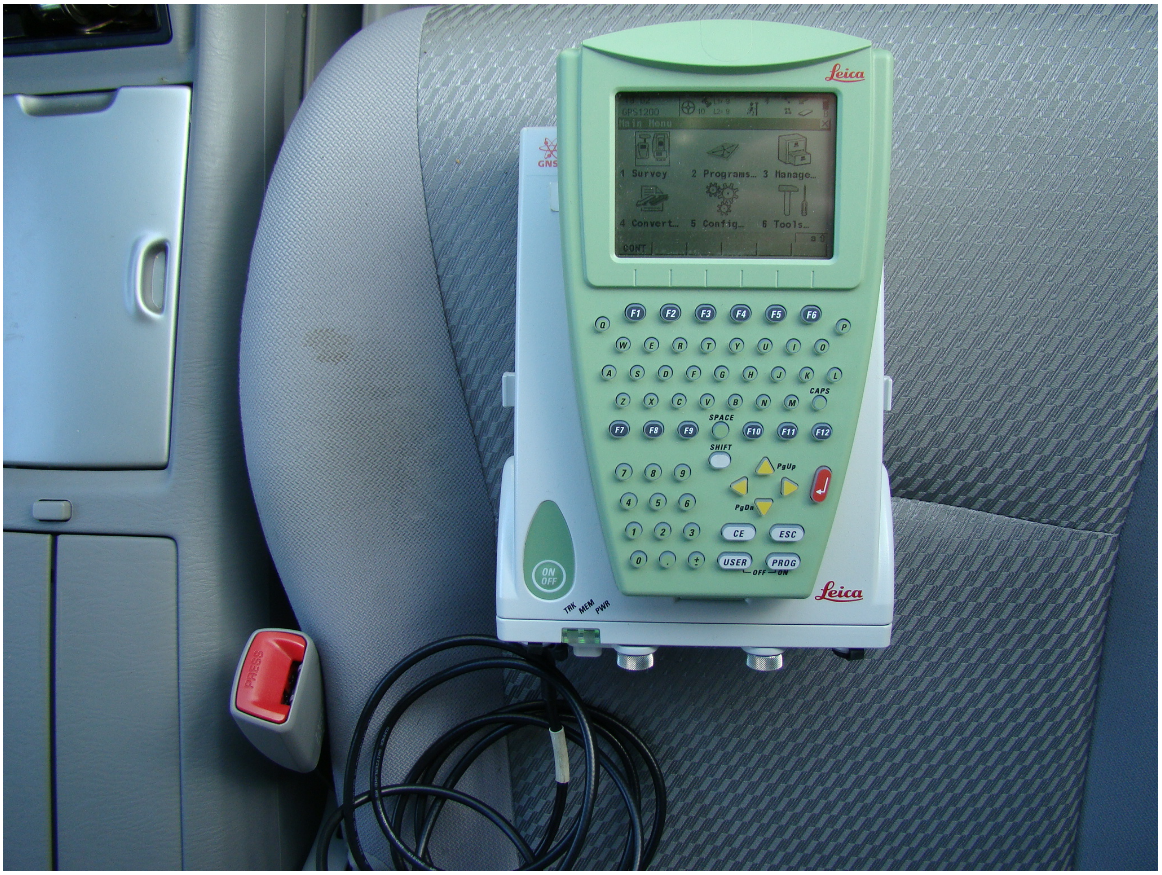


Figure 4.6: The Leica GPS1200 Surveying System on a data-collect.

shape-files from the respective counties into the script **getCentroids** [43]. The output were point shape-files and these were merged and named **D.shp.** The second task, tessellated regions, was more involved. First, a temporary point shape-file needed to be created and points added at all locations where off-ramps were found in the **E.shp** file. This task was aided by the fact that many of the road segments in **E.shp** were labled “ramp” under the Name field. Still, many were unlabeled and the only alternative was exhaustive visual inspection in the ArcEditor® environment. With all off-ramps identified and given a corresponding point in n temporary point shape-files, for the n major roads and highways, the **Solve Service Area** object was allowed to create a tessellated region shape-file **W_(n).shp** for each respective road. In order to accomplish this as described in Section 3.4.6, the time setting was set to equal 20 minutes, the estimated cost to traverse the length of the data-set. The option

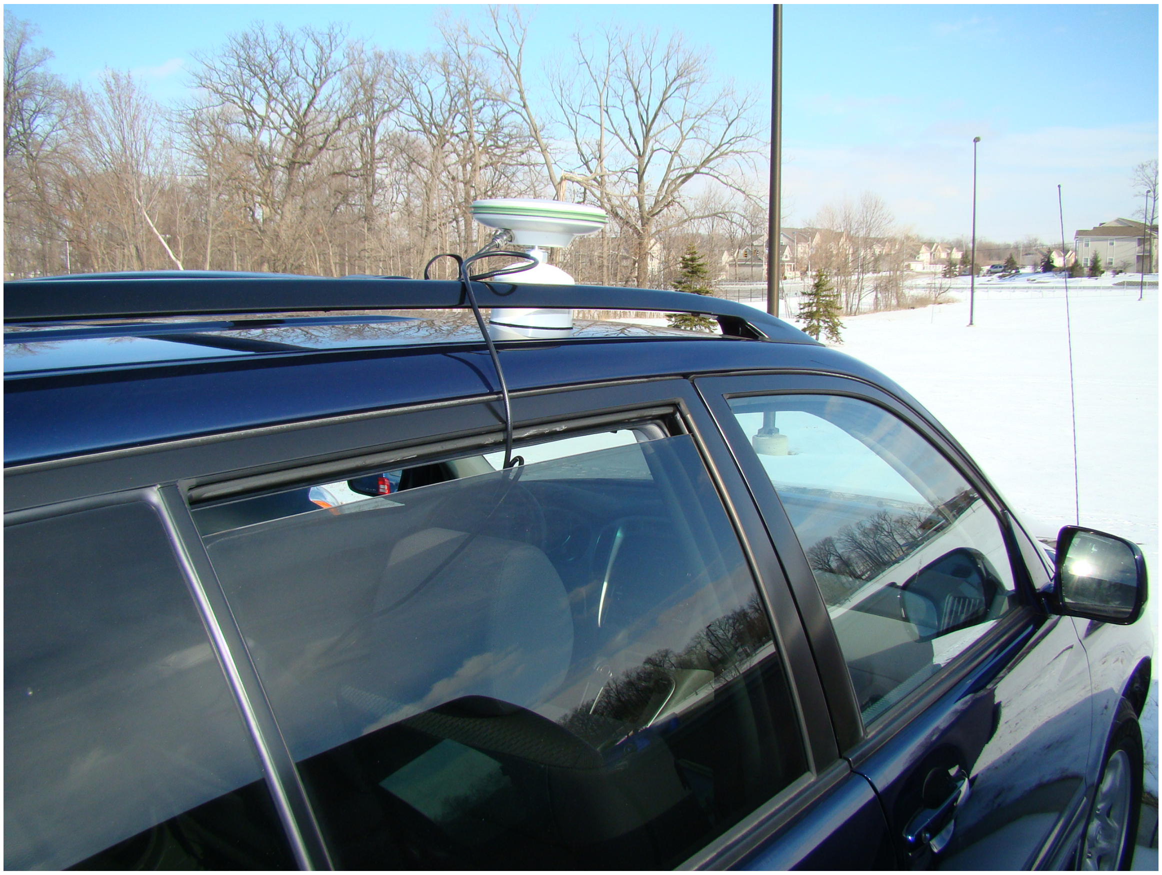
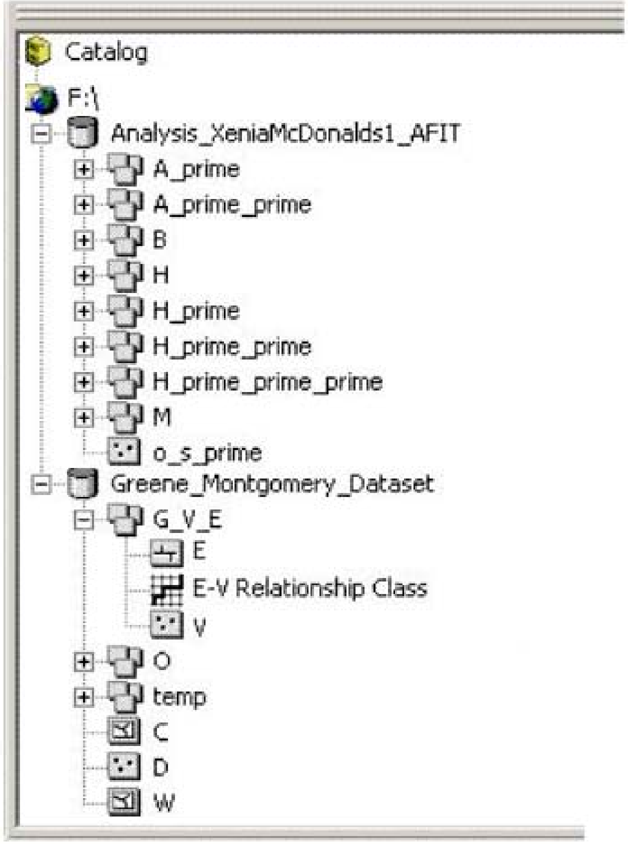


Figure 4.7: A magnet-mounted GPS antenna for the Leica GPS1200 Surveying System.

"Clip Intersections" ensured that when searches collided, they would cease, thereby forming a demarcation. Schemas for both **D.shp** and **W_(n).shp** consist of object ID, lattitude, and longitude.

4.3.5 Geodatabase Setup . It now becomes necessary to create and arrange some GeoDatabases to serve our experimentation purposes. These fall either into the category of "input" or "output" structures. The former contains the arguments and, the latter, the results of our process. Figure 4.8 depicts the *Greene-Montgomery* data-set which, per Sections 4.3.2, 4.3.3, and 4.3.4 contains: $G(V, E)$, O, $temp$, C, D, and W_n. Note the **temp** data-set: this is for computational housekeeping, to be described in great detail next, and includes such things as the tabu list.



```
Catalog
F:\
  Analysis_XeniaMcDonalds1_AFIT
    A_prime
    A_prime_prime
    B
    H
    H_prime
    H_prime_prime
    H_prime_prime_prime
    M
    o_s_prime
  Greene_Montgomery_Dataset
    G_V_E
      E
      E-V Relationship Class
      V
    O
    temp
    C
    D
    W
```

Figure 4.8: The Greene-Montgomery data-set and an analysis data-set. Individual shape-files may be stored in feature data-sets. An example of this is the **G_V_E** feature data-set which contains **E.shp**, **V.shp**, and the **EV Relationship Class**.

Similarly, there are GeoDatabases containing our analysis products, per Chapter III, contain A', A'', B, H, H', H'', H''', M, and o_s'. These are expressed as feature data-sets **A_prime, A_prime_prime, B, H, H_prime, H_prime_prime**, feature data-sets and **H_prime_prime_prime.shp** and **o_s_prime** shape-files. The feature data-sets contain polygon shape files named in lower-case and per their index **S'**. For instance, the contents of **H** are **h_1.shp**, **h_2.shp**, **h_3.shp**, . . . , **h_(s′).shp**. There will be eight analysis GeoDatabases for the experimentation tracklets enumerated in Section 4.3.3.

4.3.6 Presentation Setup. It is useful during the analysis and interpretation portion of our experiment to set up the environment such that the layers in ArcEditor® are suggestive of context. One way to accomplish this is to use the H field in the **E.shp** shape-file as described in Section 4.3.1. This represents a hierarchy ranging from $1-3$, determined by the *CFCC Table*, where 1 = highways, 2 = major roads, and 3 = local roads. In ArcCatalog® the **E.shp** properties may be altered such that this field is used to divide the data-set into *sub-types* which allow the road line-segments of **E.shp** to be rendered with different colors and weights as per Figure 4.9. This will allow immediate interpretations of the order "It appears that the vehicle is on a highway", or "it appears that the vehicle is in a residential area." On top of the **G(V.shp,E.shp)** layers (with the enhancement just described) are added layers for each of the analysis products from Section 4.3.5, tracklet **O.shp**, and coverage spot **C.shp**. Note that the gaudy choice of colors in Figure 4.9 is for illustrative purposes only. In order not to distract from other data that is presented, the roads will normally be displayed in gray-scale with varying thickness according to its hierarchy value.

4.4 Network Search and Analysis

Now it is possible to consider a practical implementation of the process described in Chapter III using the prepared data-sets and the Model Builder in ArcEditor®. The final process is illustrated by an 11x17 inch graphic of the flow structure that

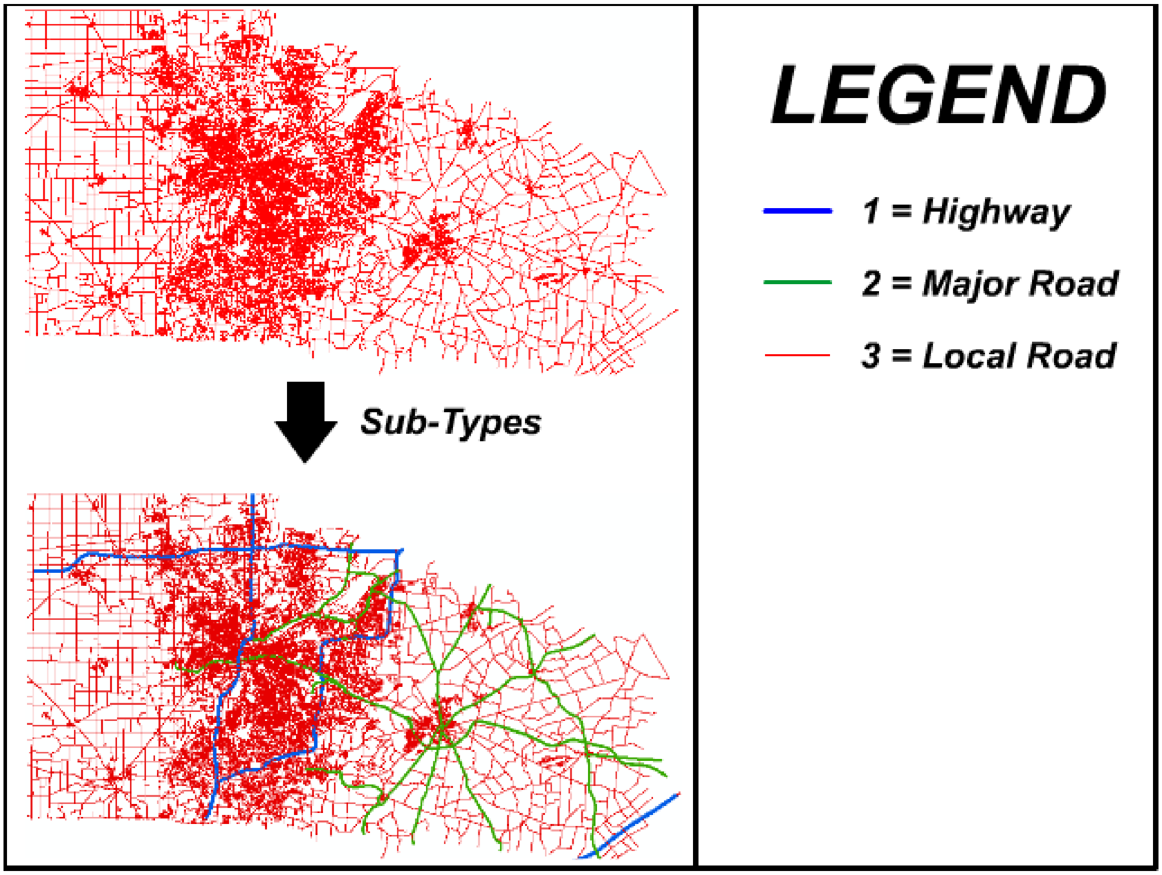


Figure 4.9: The division of edges $E \in G(V, E)$ into a hierarchical structure allows for greater context.

is attached as Figure 4.26 at the end of this chapter. The discussion will refer to Figure 4.26 in a piece-meal fashion, adhering strictly to the command-flow presented in the process pseudo-code *DestinationPrediction(G(V,E), T, O, W, D)* and illustrated in Figure 3.20 of Section 3.4.9. The easiest approach to adopt while following this discussion is to unfold the 11x17 inch diagram Figure 4.26 and to observe the thumbnail icons embedded in Figures 4.10-4.21 which depict, via silhouette, the portion of the diagram being discussed.

4.4.1 Operating on the First Point o_1 . Per line (2) in the pseudo-code in Figure 3.20, the process is obliged to start with all product feature data-sets and shape-files represented by H, H', H'', H''', B, $P_{s'}$, A, A', $D_{s'}$, $V^*_{s'}$, $\overline{G(V,E)}$, beginning as empty sets. This is a matter of course, for all shape-files have been created as per Figure 4.8 and they are all empty. In line (3) of Figure 3.20, S is attributed (before running the process) with the cardinality of the tracklet shape-file **O.shp**. This is a variable object denoted S in Figure 4.26, (the section silhouetted in Figure 4.10). Line (4) in Figure 3.20 stipulates that S' begin with a value of zero. This is accomplished by the functionality (to be described shortly) that regulates the iteration of variable **S'**.

Line (5) in Figure 3.20 stipulates that the process continue for all S, meaning in practical terms, that it cease when **O.shp** has been completely iterated. For this to happen, a Python® script, called $|\mathbf{O}| = \mathbf{0}$?, was created to check the cardinality of **O.shp** since it is incrementally deleted during the down-sample process. The location of this functionality is depicted in the lower-right portion of the thumbnail silhouette shown in Figure 4.10. Note the **Check** variable object created by the $|\mathbf{O}| = \mathbf{0}$? object. This is a boolean set to "False" when **O.shp** is exhausted. The Model Builder is paramaterized to listen to this variable and to cease when it is "False".

Line (6) in Figure 3.20 ensures that the process continues until all *S'* are iterated which results in most of the processes nesting in *S'*. Line (7) in Figure 3.20 calls for the first point in **O.shp** to be copied and placed in **O_prime.shp**. (Hence,

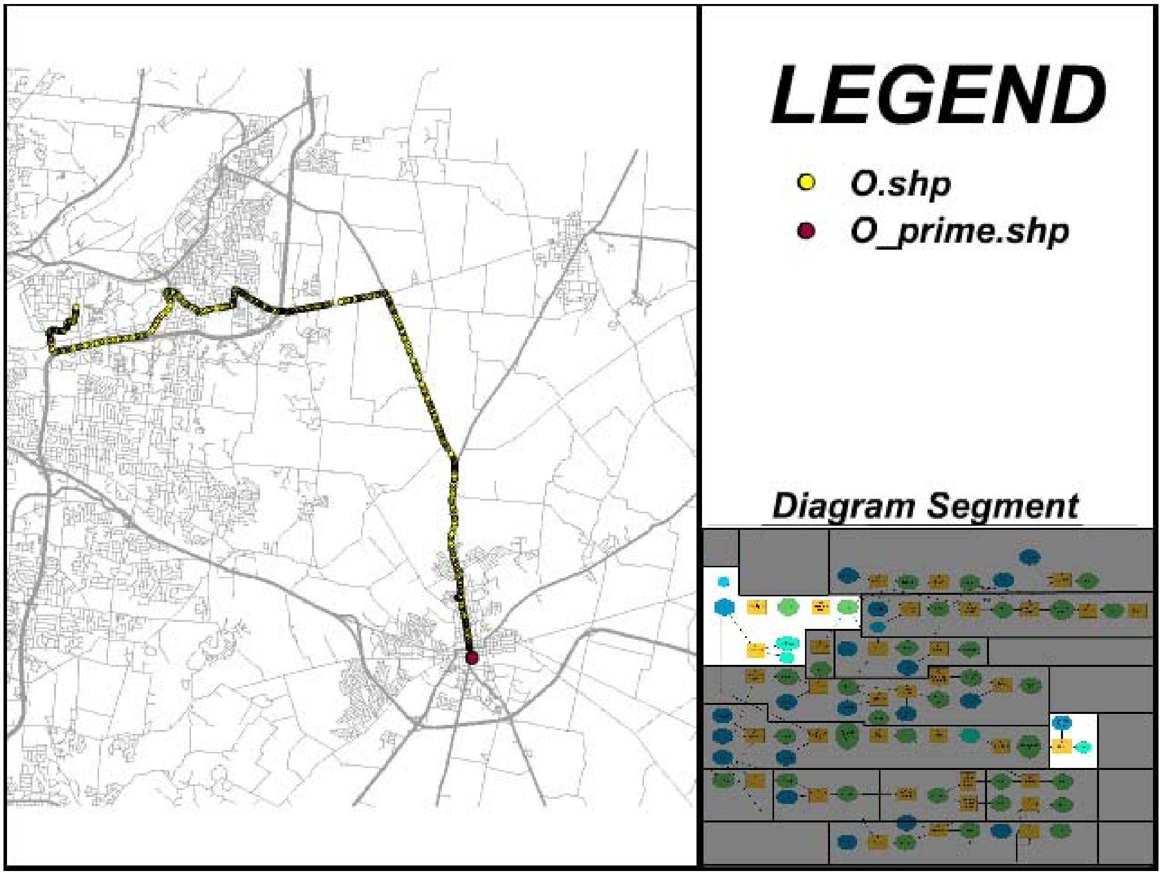


Figure 4.10: The first point of **O.shp** is automatically selected to be **O_prime** [**1**] **.shp**. Pictured is the Xenia-AFIT tracklet.

O_prime [**1**] **.shp** always equals **O** [**1**] **.shp**). Per the thumbnail silhouette in Figure 4.10, this is accomplished by a **Select** object (renamed **Select First o_s in O**) which takes as arguments data-set **O.shp**, and the name of the field to search (the *index* field). It selects the object with a '1' in this field and uses a **Copy** object to copy it to the **O_prime.shp** file. Figure 4.10 illustrates **O.shp** and **O_prime.shp**.

Line (8) in Figure 3.20 allows *S*' to increment for this operation. It is executed by the functional block **Increment S'** in the silhouetted region of the diagram in Figure 4.10. This is a Python® script that initializes *S*' to zero (Line (4) in Figure 3.20) and counts up for every iteration.

4.4.2 Creating Alpha Hulls H . Thus far, the algorithm has initialized, down-sampled the first point **O_prime** [**1**] **.shp**, and iterated *S*'. With **O_prime** [**1**] **.shp** determined, it may now be given to the $RadialDijkstra(o'_{s'},\ G(V,E),\ T,\ D)$ functionality to complete the alpha hull h_1 per Lines (9-11). For this, it was possible to utilize the **Network Analyst** tools in ArcEditor®. However, one important caveat to this function, as described in Chapter III, must be made for the practical implementation. The ESRI tool **Network Analyst** only operates on the *vertices* of graph $E \in G(V,E)$ and not on the locations D. Hence, a modification to the ideal case, as presented in Chapter III, will be made such that D is not considered and the function is understood to be $RadialDijkstra(o'_{s'},\ G(V,E),\ T)$. The sacrifice in fidelity that this imposes is not considered here to be large enough to impact the basic functionality or its results. It is important to bear this discrepancy in mind however when viewing the results.

The thumbnail silhouette in Figure 4.11 show where in the diagram this functionality appears. A **Create Network** object is employed to take arguments **V.shp** and **E.shp** (representing *G(V,E)*) and the time horizon T variable, whose value is set manually. This creates a network environment from the shape-files in preparation for the Dijkstra operation. Note the dotted arrow pointing from the **O_prime.shp** object to the **Create Network** object: where solid arrows represent command flow in Model Builder, dotted arrows represent constraints and this requires the operation to wait until the down-sample operation is complete. Once the network environment is complete, the output, labeled $\{\mathbf{G(V,E)}\}$, and **O_prime.shp** are combined in the **Add o'_s' to network** object. This is actually a **Add Locations to Service Area Network** object in the ESRI environment and allows the added point to become a starting point to search from along $\mathbf{G(V.shp, .shp)}$.

The output, labeled $\{\mathbf{G(V.shp, E.shp),\ o'\_s'}\}$, is combined with the tabu list **V_tabu.shp** with the **Add V_tabu to Network** object. This is actually an **Add Barriers to Service Area Network** object in the ESRI environment, and dictates that points from this shape-file will be regarded as obstructions when found to intersect any **V.shp** in the graph. (An explanation as to how the tabu list is generated will follow in Section 4.4.6). After this operation, the network, now la-

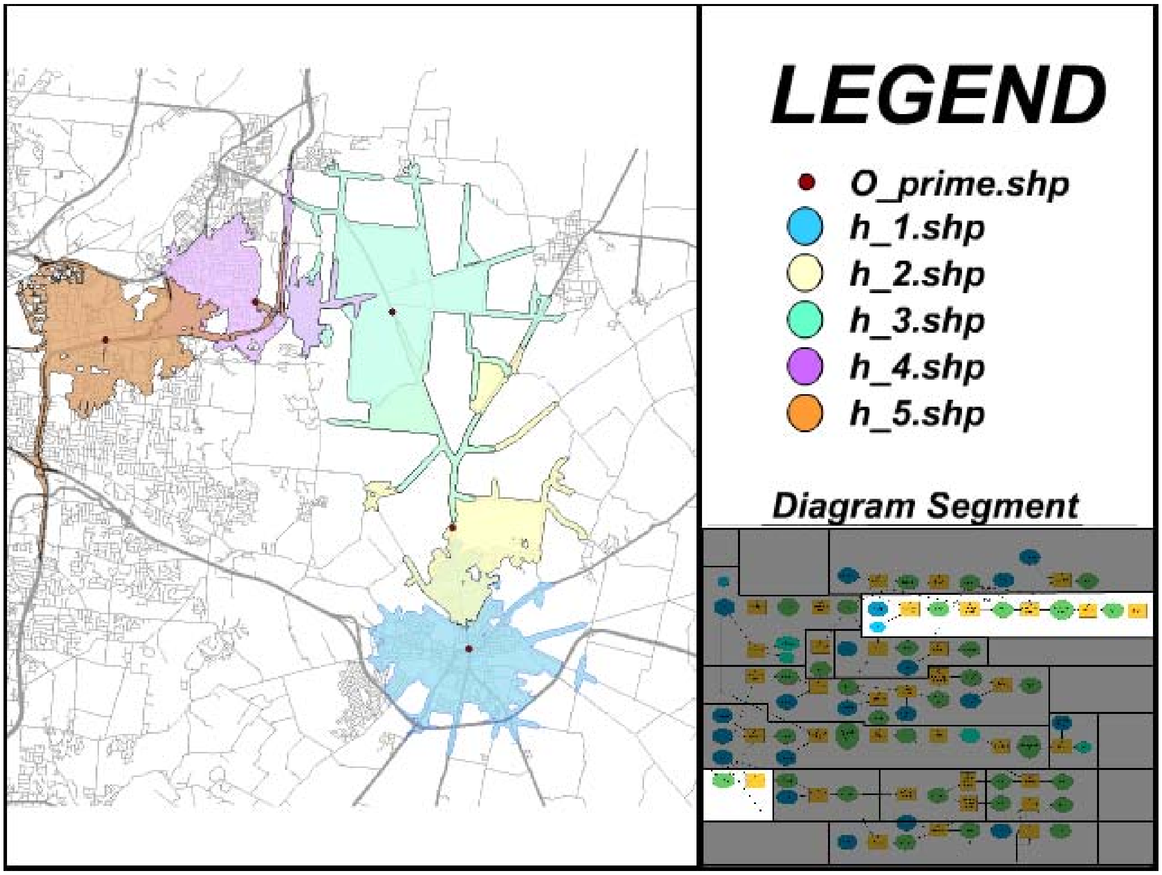


Figure 4.11: Five alpha hulls **h_(s').shp** are created and placed in feature data-set **H**. Pictured is the Xenia-AFIT tracklet at $T = 5$.

beled $\{\mathbf{G}(\mathbf{V}, \mathbf{E})$, $\mathbf{o}'\_\mathbf{s}'$, $\mathbf{V\_tabu}\}$, is complete and ready for the search within T to begin. The **Solve h_(s′)** object is actually an ESRI **Solve Service Area** object which was designed to determine how much area in a municipality could be serviced from a given location (fire stations and hospitals recommend themselves). Typically the **Solve Service Area** object will take many starting points (rather than our singleton **o_(s')**) and generate alpha hulls from a Dijkstra search in all directions within a T. This answers perfectly to our pseudo-code requirements, and the result is a single alpha hull which radiates from **o_(s')**. This, then, is selected and stored as **h_(s').shp** to feature data-set **H** where it will soon be put to use.

4.4.3 Downsampling the Tracklet to O' .

It has been described in Section 4.4.1 how the first table entry in tracklet **O.shp** will be copied to **O_prime.shp** but not how the actual down-sampling occurs (as the function calls for in Line (12) in Figure 3.20). It was stated in Section 3.4.4 that this would be done by masking part of **O.shp** with **h_(s').shp** per Equations 3.11, 3.12, and 3.13. Figure 4.12 shows what this will look like after **h_(s').shp** file has been generated. The thumbnail silhouette in this illustration shows the portion of the process diagram of Figure 4.26 where the **h_(s').shp** file is appended to **H_temp.shp** file (in the **temp** feature data-set in the Greene-Montgomery GeoDatabase). This singleton object is then given as an argument, along with **O.shp** to the **Downsample O.shp with h_(s′).shp** object which is merely an ESRI **Select with Polygon** object that has been renamed. The result is that all $O \cap h_{s'}$ are selected. Next, these are added to a **Delete Selected Points** object, with obvious results. This, then, is passed to the **Get |O|** object (renamed from an ESRI **Get Table Dimension** object) resulting in a new value for variable **S**. This constrains the **Renumber O.shp** functionality, which is another Python® scriptlet. This scriptlet performs the actions called for by Line (13) in Figure 3.20 and renumbers the index field in **O.shp** such that the first (remaining) point is numbered '1', the next '2', and so on. This is done so that when the **Select first o_(s) in O** functionality described in Section 4.4.1 is invoked again, it will select the point at the top of the table. The result, **Renumbered O**, constrains the scriptlet $|\mathbf{O}| = \mathbf{0}$? (also described in Section 4.4.1) that checks that $\mathbf{S} \neq 0$.

4.4.4 Generating Motion Models M. Line (14) in Figure 3.20 calls for a simple motion model (per Equation 3.7) which is used to compute figure of merit R as described in Section 3.4.7. The thumbnail silhouette in Figure 4.13 highlights the functionalities in the diagram where this takes place. The object **Create m_(s′).shp** (renamed from the ESRI **Buffer** tool) merely employs Equation 3.7 for a velocity $v = 100$ *mph* (Note that this is a conservative value. DTT used 130 *mph* [25]). In addition to velocity v, this functionality accepts **O_prime [s′] .shp** for each s', the result being as is shown in Figure 4.13. These products are saved as **m_(s').shp** shape-files in the **M** feature data-set in the Analysis geodatabase.

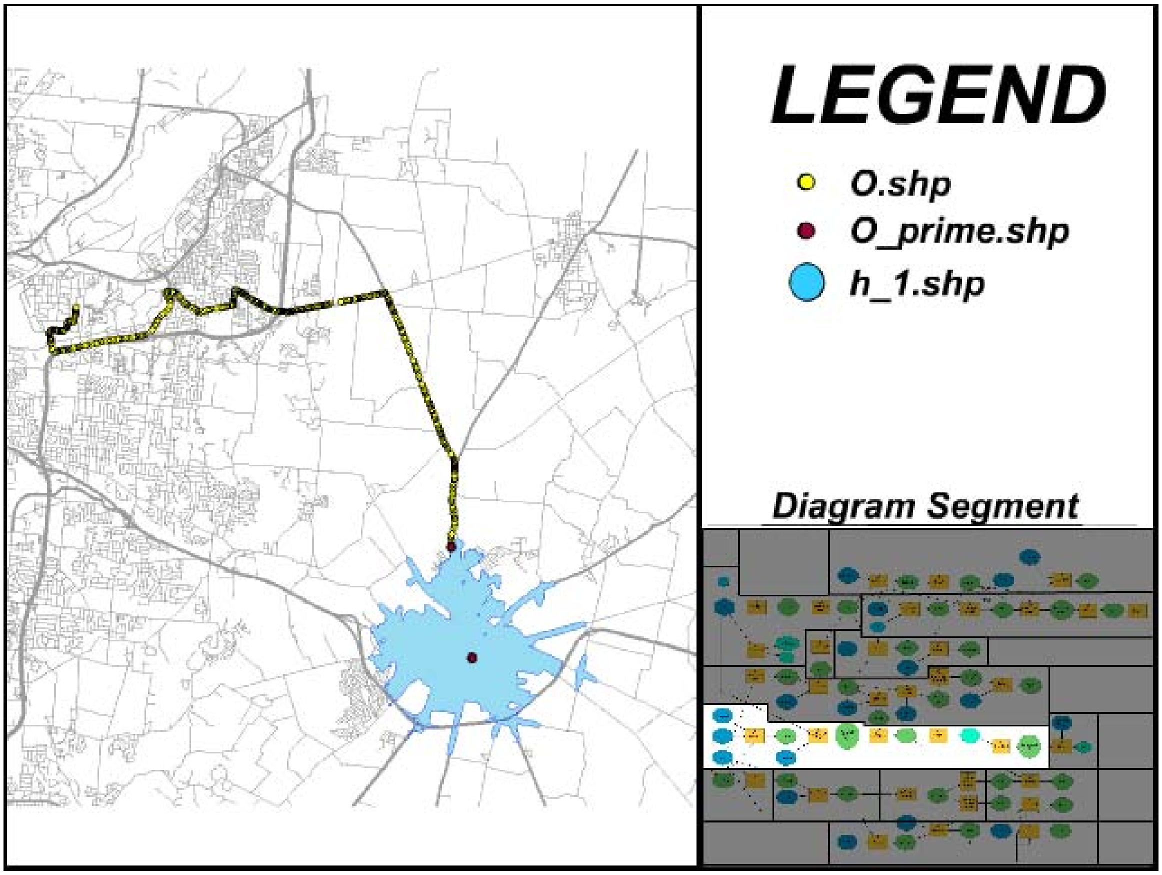


Figure 4.12: As each $h_n \in H$ is created, it may be used to downsample O by masking intervening points. Here **h_1.shp** for the Xenia-AFIT tracklet is used to determine **O_prime(2).shp**.

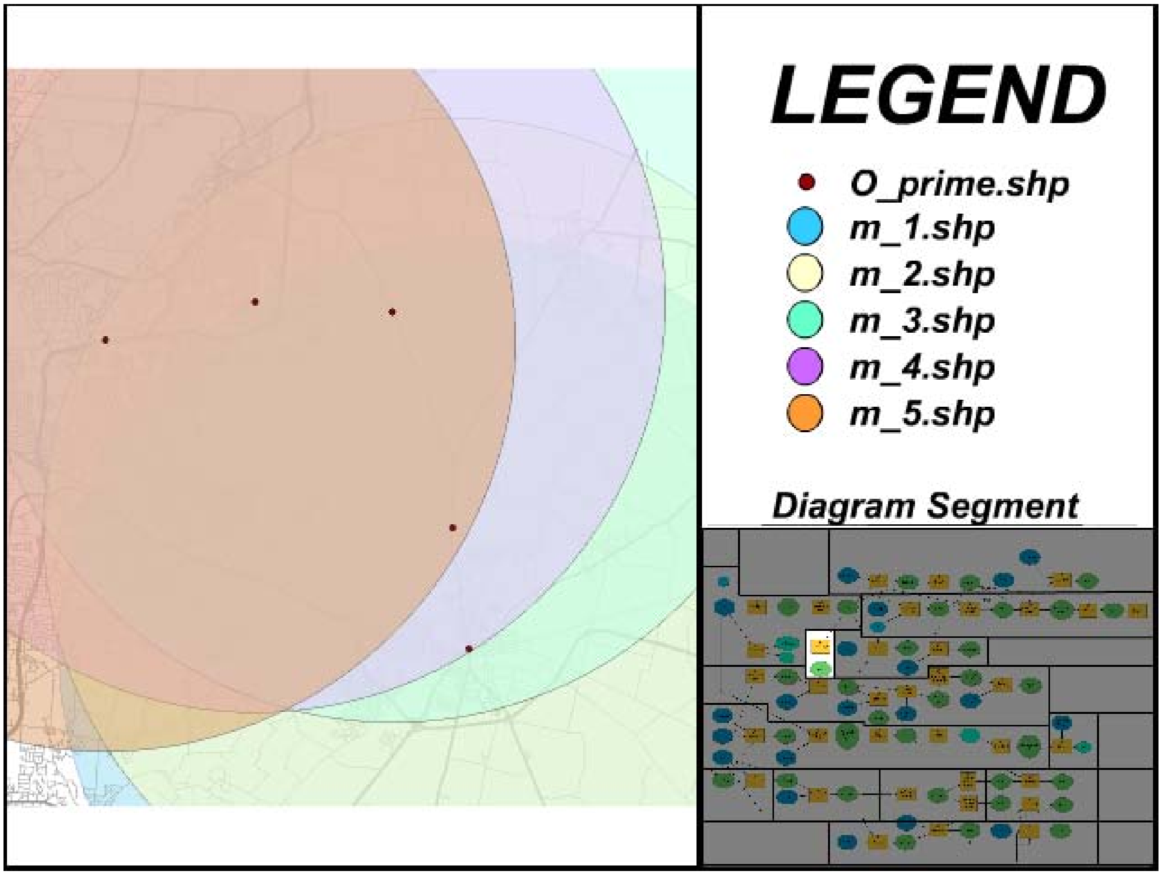


Figure 4.13: Once a **o_s'.shp** is calculated, polygon **m_s'.shp** is calculated. Pictured here is set **m_1.shp** for Xenia-AFIT tracklet at $T = 5$.

4.4.5 Creating Negative Space B. Line (15) in Figure 3.20 calls for the generation of the negative space. The thumbnail silhouette in Figure 4.14 shows that this occurs in the middle of the diagram. Before a general description, a design choice must be pointed out that is a consequence of the *a posteriori* nature of B as described in Section 3.4.4. A problem arises if one wishes to use a negative space b_2 between points o_1 and o_2 to define Tabu elements because that product would be $n+1$ iterations too late to inform the search that generates h_2. Hence, it can only be useful to the tabu list $\overline{G(V,E)}$ after the fact where it may participate, *a posteriori*, in pruning space $G(V,E)$. It will still retain its usefulness, per function $NListener(O,\ B,\ s')$ to determine the semantic "vehicle is not on a *Deliberate Journey*." Conversely, h_s may be used to select $\overline{V}$ from $V \in G(V,E)$ all vertices within its area to be added to the tabu list to the effect that when h_s is generated, it will not include the area of the negative space. This course of action is worse, because $h_1 \cap h_2$ will always equal 0 making $B = \phi$. This would cause $NListener(O,\ B,\ s')$ to never function. Hence, a decision must be made between the former, which ensures that set B is created at the cost of the *a posteriori* penalty, or the latter, which ensures greater run-time efficiency but offers no set B. For this process, the former approach was chosen because it is essential to generate B in order to provide cover for our key assumptions. Also, the larger tabu list $\overline{G(V,E)}$ that results from the latter approach is not be significant enough for that choice to be taken.

The process for creating a $b_{s'+1}$ product from $h_{s'}$ and $h_{s'+1}$ relies upon a queue-like construct consisting of two elements. Shape-files **b_temp_1.shp** and **b_temp_2.shp** (stored in the **temp** feature-set) are used to hold a copy of **h_(s').shp** and **h_(s' + 1).shp** respectively. The **Copy h_(s') to b_temp_1.shp** object accomplishes this once for each *S'* as is seen in the far-left and middle of the thumbnail silhouette in Figure 4.14. **b_temp_1.shp** and **b_temp_2.shp** are then added to an **Intersect** object where the result is stored in a final shape-file product **b_(s').shp**. This final product is connected by dotted arrow to constrain the next operation **Delete Features** which deletes the contents of **b_temp_2.shp**. The output from this, then, constrains the

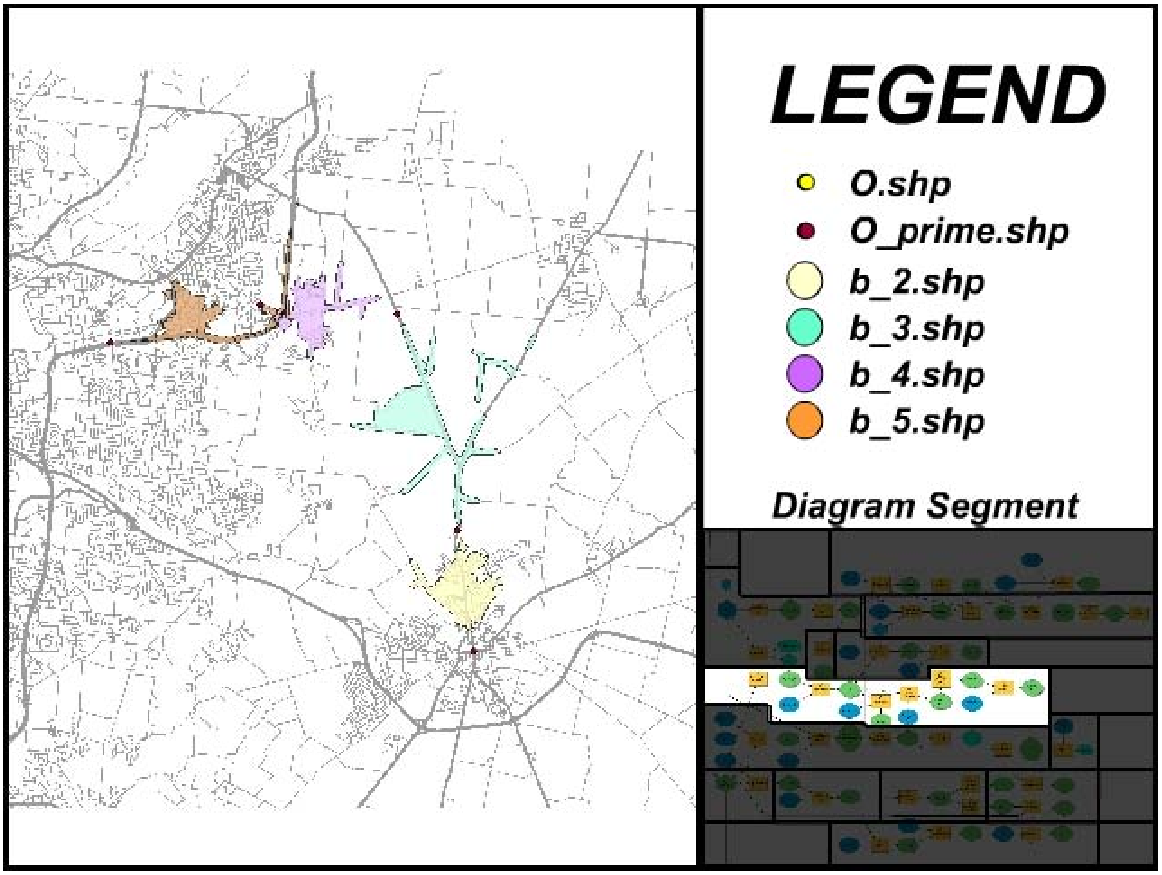


Figure 4.14: Notice that there is no **b_(1).shp**. Pictured are all **b_(s').shp** shapefiles in feature data-set **B** for the Xenia-AFIT tracklet at $T = 5$. When **O_(s').shp** and **O_(s' + 1).shp** are calculated, a polygon is created as **b_(s' + 1).shp**.

Copy b_temp_1.shp to b_temp_2.shp object, which as its name fortells, moves the contents of **b_temp_1.shp** to **b_temp_2.shp**. The result from this constrains a final **Delete Features** which deletes the contents of **b_temp_1.shp**. The practical result is a queue that passes successive **h_(s′).shp** products and intersects them. Note that on the first iteration, **b_temp_2** contains nothing and the first intersection results in **b_temp_1.shp** $= \phi$. A practical example may be shown in Figure 4.14. As a final operation, the **b_(s').shp** file will be appended after creation to a **B_temp.shp** file for future use in Section 4.4.9. (The difference between this and the **b_(s').shp** is that the former is a set stored in the **temp** feature data-set containing all $b_s \in B$ where the latter are individual shape-files for each individual polygon stored in the **B** feature data-set).

4.4.6 Working with the Tabu List $\overline{V}$ *.* Line (16) in Figure 3.20 calls for the addition of untraversable vertices to thetabu list (which for our purposes need only include vertices $\overline{V} \in V$). The thumbnail silhouette in Figure 4.15 reveals that this functionality may be found in the exact middle of our diagram. This simply takes the respective **b_(s').shp**, created in the last section, along with set **V.shp**, and applies them to a **Intersect** object. The result is a **V_tabu.shp** object that is stored in the **temp** folder.

4.4.7 Using Standard Distance to Prune the Space, Creating H' *and* H''*.* Lines (17-21) in Figure 3.20 are the successive steps required to generate the standard distance **h_prime_(s′).shp** and **h_prime_prime_(s′).shp** along with intermediate products **d_(s′).shp**, **a_prime_(s′).shp**, and **a_prime_prime_(s′).shp** and we begin with **d_(s′).shp**. As shown in the thumbnail silhouette in Figure 4.16, this process begins where the creation of **h_(s′).shp** left off. It takes **h_(s′).shp** and the set of building centroids **D.shp** and selects a subset of the latter with the former using a **Select D.shp with B_(s').shp** object (renamed from the ESRI **Select** object). The **d_(s′).shp** file is only temporary, and is stored in the **temp** feature data-set.

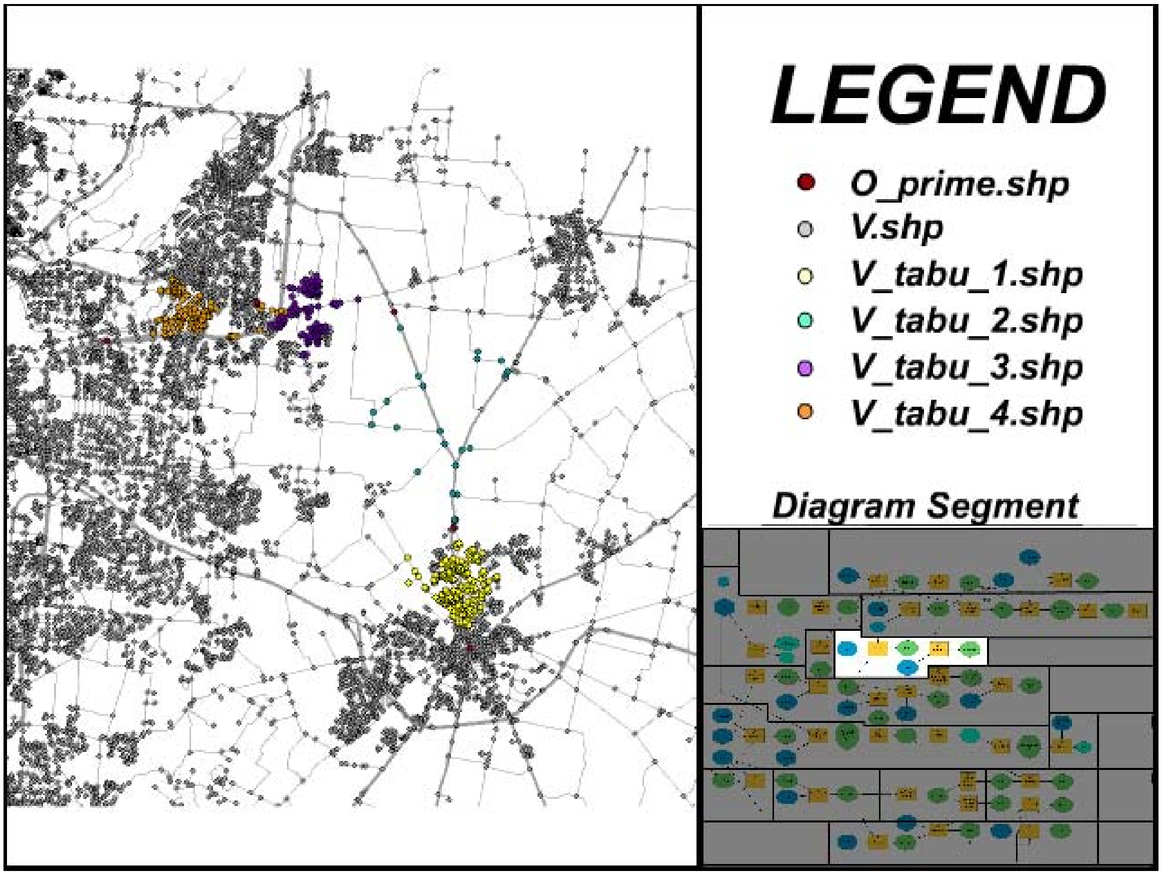


Figure 4.15: tabu list $\overline{V}$, saved as **V_tabu_(s').shp**, for the Xenia-AFIT tracklet at $T = 5$.

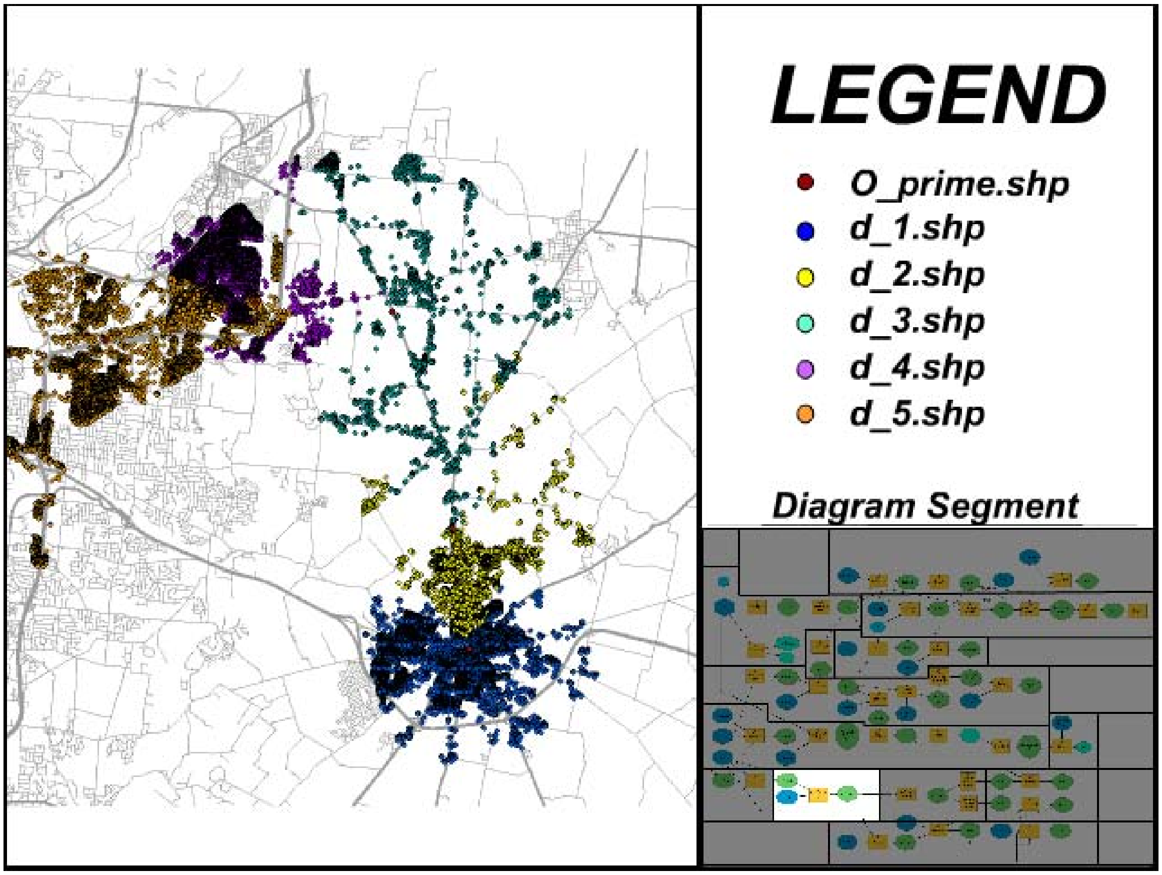


Figure 4.16: Selected points $\mathbf{d}\_(\mathbf{s}').\mathbf{shp}$ created by intersecting $\mathbf{h}\_(\mathbf{s}').\mathbf{shp}$ with $\mathbf{D}.\mathbf{shp}$. Pictured here are all $\mathbf{d}\_(\mathbf{s}').\mathbf{shp}$ for the Xenia-AFIT tracklet with $T = 5$.

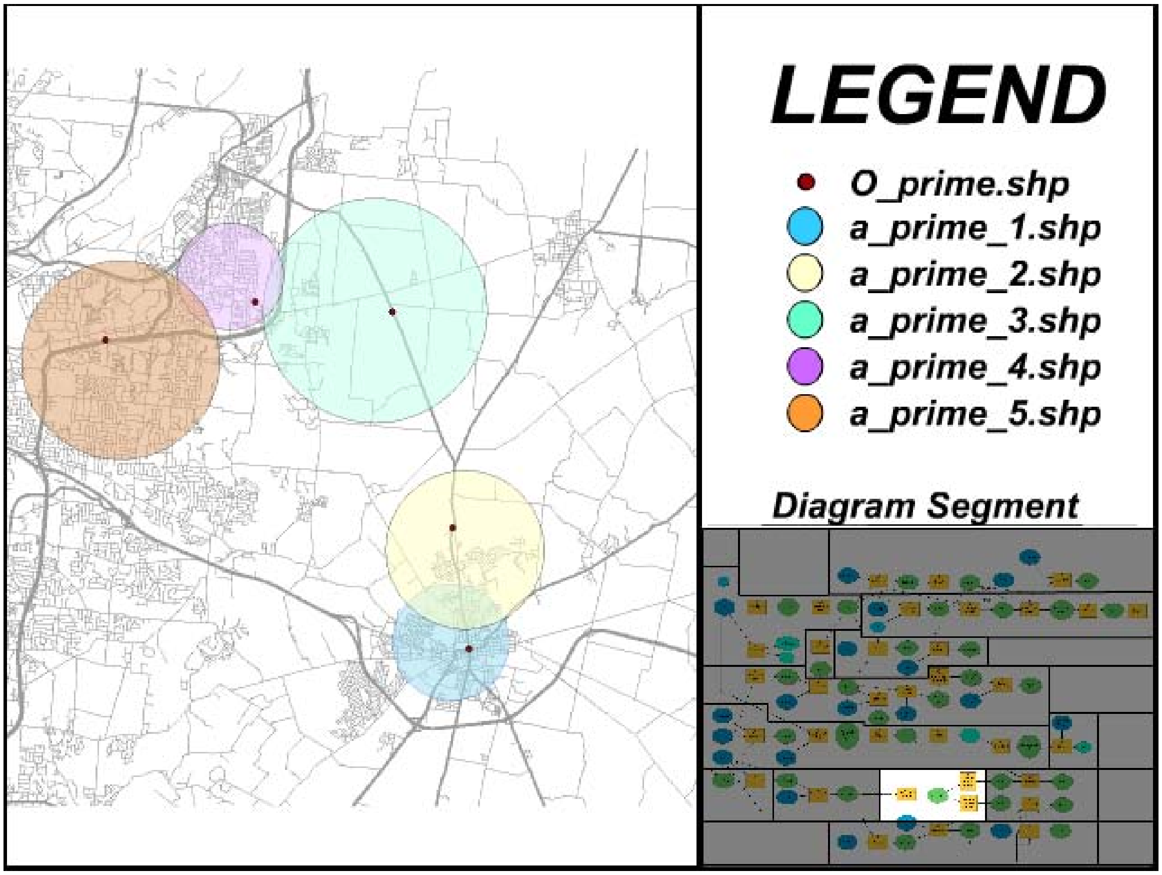


Figure 4.17: standard distance disks **a_prime_(s').shp** calculated from **d_**(s′)**.shp**. Pictured here are first standard deviation disks for the Xenia-AFIT tracklet at $T = 5$.

What follows is the implementation of the ESRI **Standard Distance** tool, and this will proceed in parallel for the first and second standard deviations as their functionality is the same. Lines (18, 19) in Figure 3.20 call for creation of **a_prime_(s').shp** and **a_prime_prime_(s').shp** shape-files and their location in the diagram may be inferred from the thumbnail silhouette of Figure 4.17. **d_**(s′)**.shp** is given as an argument to two **Standard Distance** instances where they are parameterized with 1 and 2 for their respective σ. The results are **a_prime_(s').shp** and **a_prime_prime_(s').shp**. Note in Figure 4.17 that the centers of these disks do not coincide with the points of **O_prime.shp** (their natural centroids). Instead, they can be seen to tend towards greater densities of roads (and hence habitation) which are depicted in gray-scale relief.

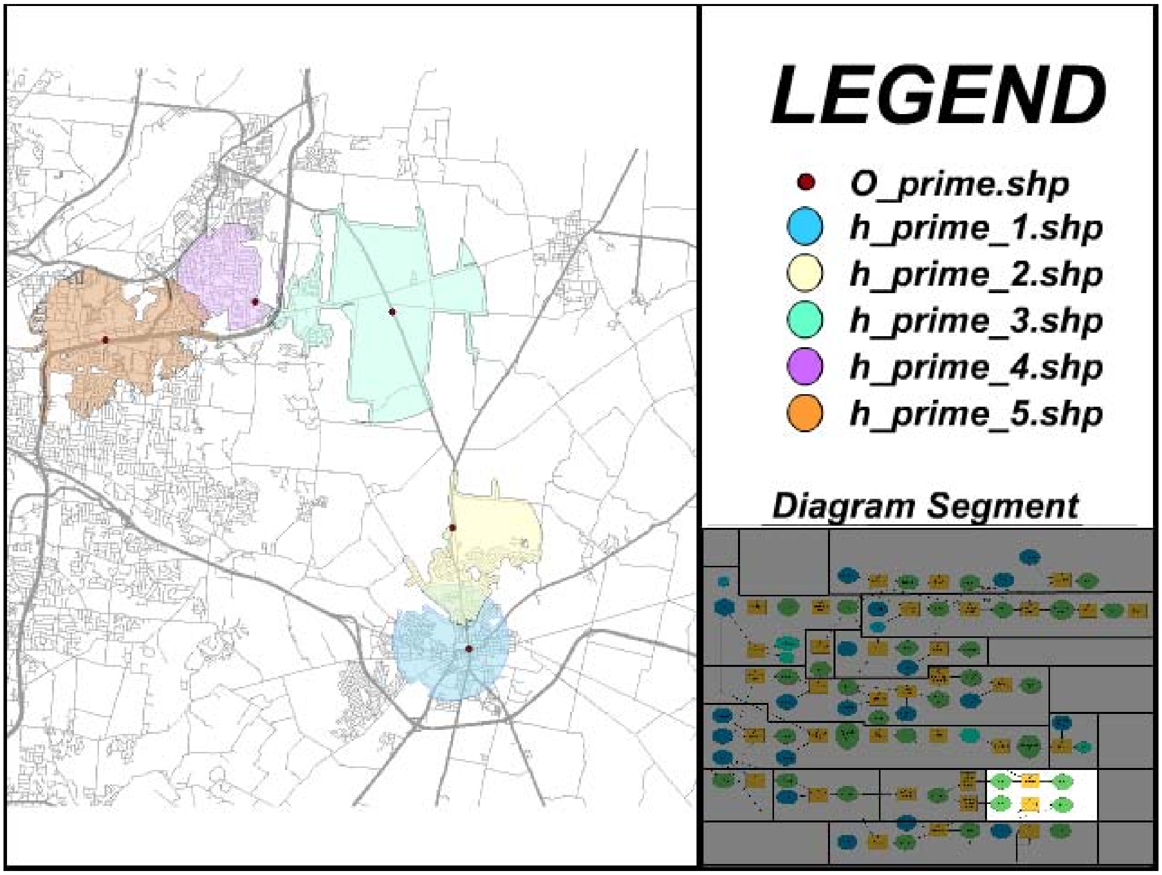


Figure 4.18: Final **h_prime_(s’).shp** products stored in feature data-set **H_prime**. Pictured is the Xenia-AFIT tracklet at $T = 5$. Note the reduced size of **H_prime** feature data-set from the **H** feature data-set shown in Figure 4.11.

Lines (20, 21) in Figure 3.20 call for the standard distance products to be intersected with **h_(s’).shp**, and the thumbnail silhouette in Figure 4.18 shows that these functionalities are at the right-bottom of the diagram. For this, intermediate products **a_prime_(s’).shp** and **a_prime_prime_(s’).shp**, along with **h_(s’).shp**, are given as arguments to **Intersect** functionalities, the results being written to **h_prime_(s’).shp** and **h_prime_prime_(s’).shp** in the **H_prime** and **H_prime_prime** feature data-sets respectively. The first standard deviation results from the Xenia-AFIT tracklet are depicted in Figure 4.18. Note that these resemble **h_(s’).shp** files depicted in Figure 4.11 except that they have been clipped and rounded at the edges.

4.4.8 Using Tessellated Regions to Prune the Space to Create H'''. Line (22) in Figure 3.20 calls for the $tessellation(stateMachine(O',\ G(V,E)),\ H,\ W\_n)$ function to be called. The thumbnail silhouette in Figure 4.19 and Figure 4.20 show that this functionality is at the bottom of the diagram. **E.shp** and **O.shp** are added to the functional block **stateMachine** where road segments **E.shp** are interrogated according to the *Hierarchy* field (per Table 4.2) and track point **O_prime** [**s**′] **.shp** is interrogated according to its velocity field (per Table 4.4). The operation (black-boxed) applies the $state(O',\ G(V,E))$ function as described in Section 3.4.6. The result is a boolean variable **SFS**, which corresponds to the flag that is returned from the state-machine checking the status of the $Slow-Fast-Slow$ state. This constrains an **Intersect** object that takes as arguments **h_(s').shp** and **W_n.shp**, writing the result to singleton set **H_prime_prime_prime.shp**.

Figure 4.19 shows tessellated regions shape-file **W_(n).shp** superimposed over **G(V.shp,E.shp)**. Superimposed over this is tracklet O. Figure 4.20 shows the final tessellated intersection of **h_5.shp** with **W_(n).shp** when the *Slow-Fast-Slow* state is detected. It can be seen that **O.shp** terminates (after exiting a major road) inside its predicted region.

4.4.9 Flow Control and the State Machine . Finally, Line (23) in Figure 3.20 calls for the $NListener(O,\ B,\ s')$ function described in Section 3.4.7 to be executed. The thumbnail silhouette in Figure 4.21 reveals that this functionality is located in the upper right corner of the diagram. First, **O.shp** and the **B_temp.shp** polygon are fed as arguments to an **Intersect** object. The result is **opCheck.shp** (stored in the **temp** feature data-set). This is interrogated by the **if neq {}** functionality (black-boxed) such that if **opCheck.shp** is an empty set, then output variable object **Optimal** will remain “true.” However, when tracklet **O.shp** is found to have intersected the negative space of **B_temp.shp** then **Optimal** will equal “false” necessitating a cascade of events as shown by the three conditional dotted arrows exiting this variable. First, **V_tabu.shp** is appended to **V.shp**. Second, the **Add V_tabu**

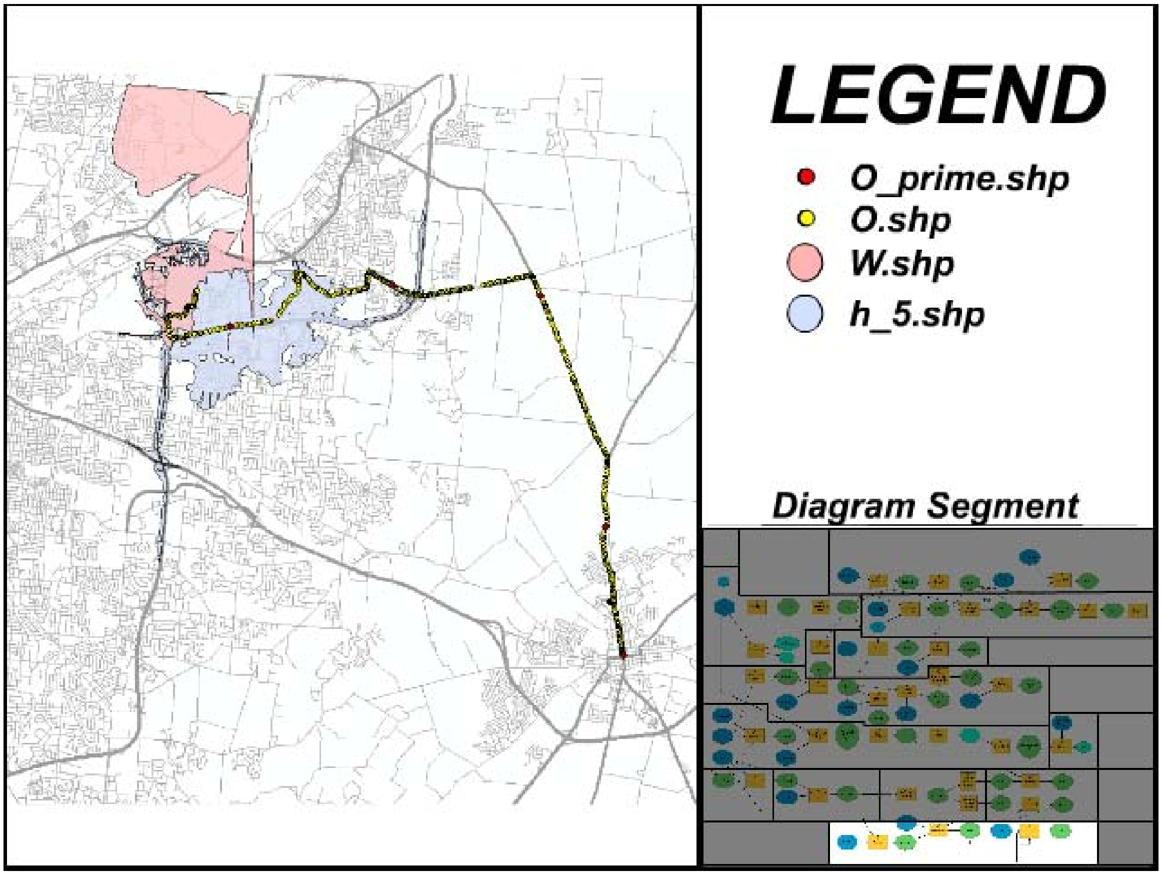


Figure 4.19: *Xenia-AFIT* tracklet file **O.shp** superimposed on Tessellated Regions file **W_(n).shp**.

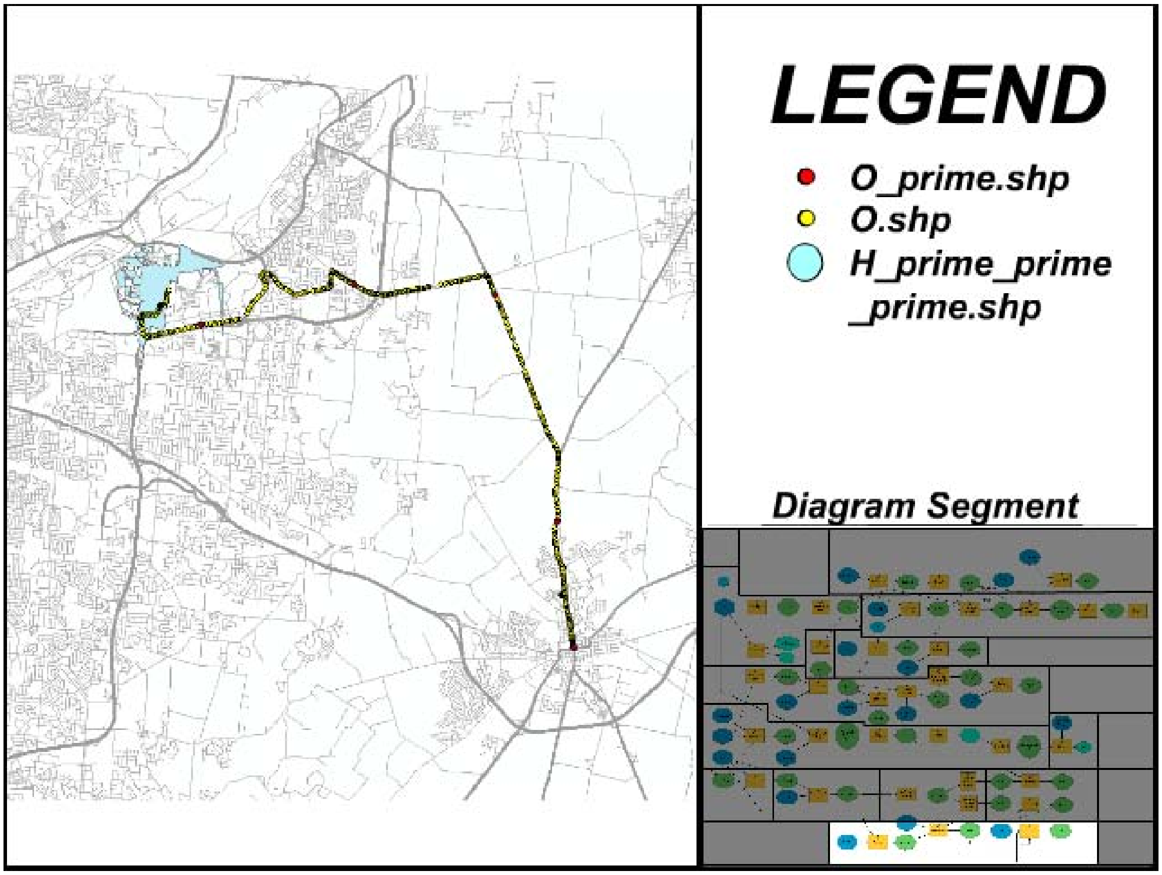


Figure 4.20: Polygon file **H_prime_prime_prime.shp** superimposed upon the final leg of the journey between Xenia and AFIT at $T = 5$.

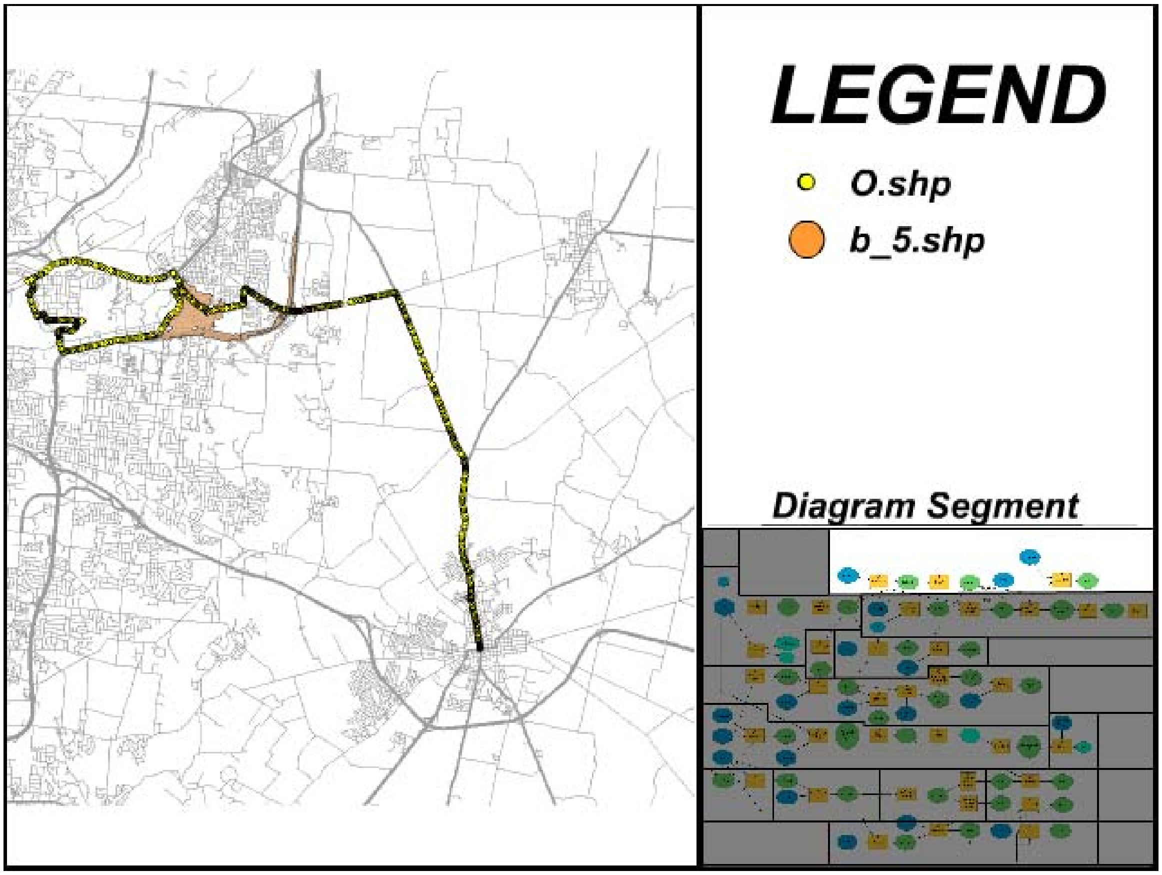


Figure 4.21: If the tracklet **O.shp** circles back over negative space **b_(5).shp**, the semantic "vehicle is not on a *Deliberate Journey*" is generated.

to network functionality discussed in Section 4.4.2 is disabled so that these are no longer added as barriers to the network. Third, the **Intersect** functionality discussed in Section 4.4.6 that creates the tabu list **V_tabu.shp** is disabled.

It is worth mentioning at this point that this event-handling is purely a design choice meant to demonstrate the over-arching concepts for searching this space. It has been stated that this event should generate a semantic which says "vehicle is not on a *Deliberate Journey*." However, there is deeper meaning in this. Though we do not discuss it in detail because it is beyond the scope of this work, others at AFIT [13] have implemented reasoning engines which specialize in this sort of event. In particular, if a reasoning engine has been ported to this process, it might also inform "vehicle is conducting a surveillance of an area."

4.5 Experimentation and Results

Having described the process in detail, conditions for the actual experimentation must be outlined. For each of the eight tracklets enumerated in Section 4.3.3, it was decided to conduct three process-runs each, one for $T = 1,\ 3$, and 5 minutes. It was recognized that the only parameters capable of variation for experimentation were Time Horizon T or the α parameter for the alpha hulls. The latter was ruled out because, if α becomes too small then the polygon develops tunnels and crevasses that fails to properly downsample the tracklet **O.shp**. In the other extreme, the polygon begins to resemble a mere convex hull, thereby thwarting our stated goal of attaining a small polygon. The desired geometry is a fairly amoeba-shaped polygon for which the ESRI **Solve Service Area** tool was judged to be sufficient. This leaves Time Horizon T and the three values listed above are reasonable fractions of the journeys which varied from ten to twenty-five minutes. Each of the twenty-four runs (three for each tracklet) were stored in its own GeoDatabase as illustrated in Figure 4.8. The runs took between one and two minutes for each tracklet on a Dell PowerEdge 2950 server-blade which is roughly a tenth of the actual time for the tracklets to unfold on the open road. This, despite the fact that the ESRI Model Builder is a non-optimized prototyping environment, ensures our stated goal of real or near-real time performance. Illustrations of the results for $T = 5$ may be seen in Figures A.1-A.8, in Appendix A and data tables listing R and F values for $T = 1,\ 3,\ 5$ are listed in Tables B.1-B.24 in Appendix B.

Having run the processes, it next becomes necessary to measure the value of our results. Eqs. 3.32 and 3.33 proposed in Section 3.4.8 will aid us in determining whether the results are useful as knowledge products or whether they should be altered, improved, or discarded altogether. Recall from Section 3.4.8 that a measure of efficiency was proposed, called R, and a measure of accuracy, called F. We turn next to the generation of these measures in order to judge the products that have been generated. These are summarized by the mean for each process-run in Table 4.5 in Section 4.5.3.

4.5.1 Final Measurements of Polygon Efficiency R . Recall that figure of merit R, described in Equation 3.32, called for the area of a search products H, H', H'', or H''' to be judged as a ratio with the area of the corresponding set of motion models M. The idea, borrowed from the original Dynamic Tactical Targeting (DTT) [25] experiments, suggests that, given a vehicle's potential for kinetic motion, expressed as M, we may use M as a baseline to judge the effectiveness of other more efficient approaches. Equation 3.32 may be expressed as function $R(h_{s'},\ m_{s'})$ and by this we mean to divide the area of alpha hull $h_{s'}$, by the area of motion model $m_{s'}$ and subtract the result from one.

A cursory examination of Tables B.1-B.24 reveal results in the range of 0.94 - 0.97, meaning that the alpha hulls were between three and six percent of their original possible areas. This is born out by Figure 4.22 which shows **h_5.shp** and **m_5.shp** for the *TechEdge-Xenia* tracklet at $T = 5\ min$. This is an excellent result when we recall that the DTT effort boasted ranges of 0.75 - 0.77 with a much less conservative motion model M [25]. We should expect even better savings with products H', H'', and H''', and indeed such was the case. However, in order to use function R we would need to devise a logarithmic scale in order to appreciate the results because they would range between 0.98 - 0.99 and vary in tenths and hundredths of a percent. However, since R is a measure of improvement, it would make sense to express it for H', H'', and H''', as $R(h'_{s'},\ h_{s'})$, $R(h''_{s'},\ h_{s'})$, and $R(h'''_{s'},\ h_{s'})$, meaning that the comparison is made between the other clipped polygons, not against motion model M, but against initial alpha hulls H. This approach provides a finer-grained look at the actual improvements over H and allow better judgments to be made as to their efficacy. Figure 4.23 shows an example of these polygons in relation to each other.

4.5.2 Final Measurements of Polygon Accuracy F . Measuring only the efficiency of the process does not satisfy the critic that the main goal of predicting the possible destination horizon of a vehicle has been accomplished. Thus far, we have been able to state that, per measurement R, we have been able to shrink an

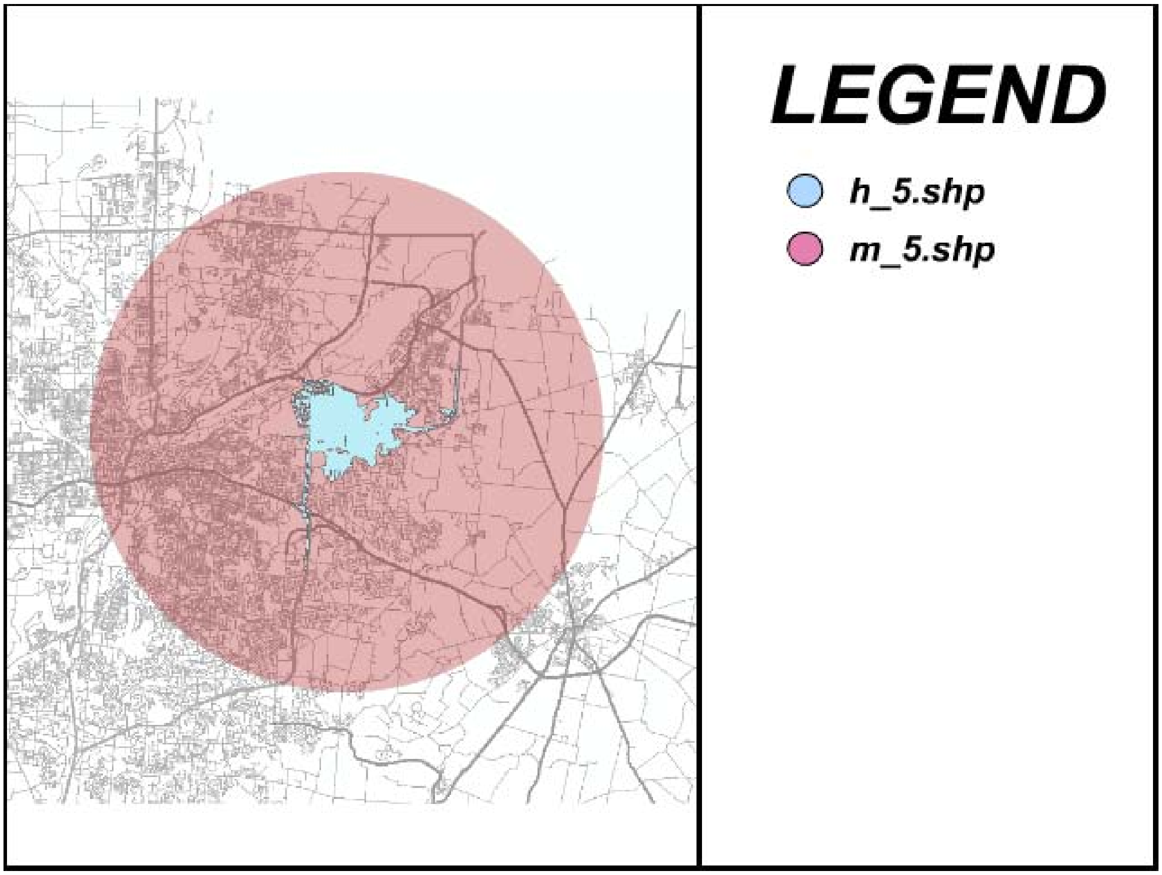


Figure 4.22: The efficiency function $R(h_{s'},\ m_{s'})$ is a measure of the ratio of alpha hull $h_{s'}$ and motion model $m_{s'}$. Polygons are **h_5.shp** and **m_5.shp** from the *TechEdge-Xenia* tracklet at $T = 5$.

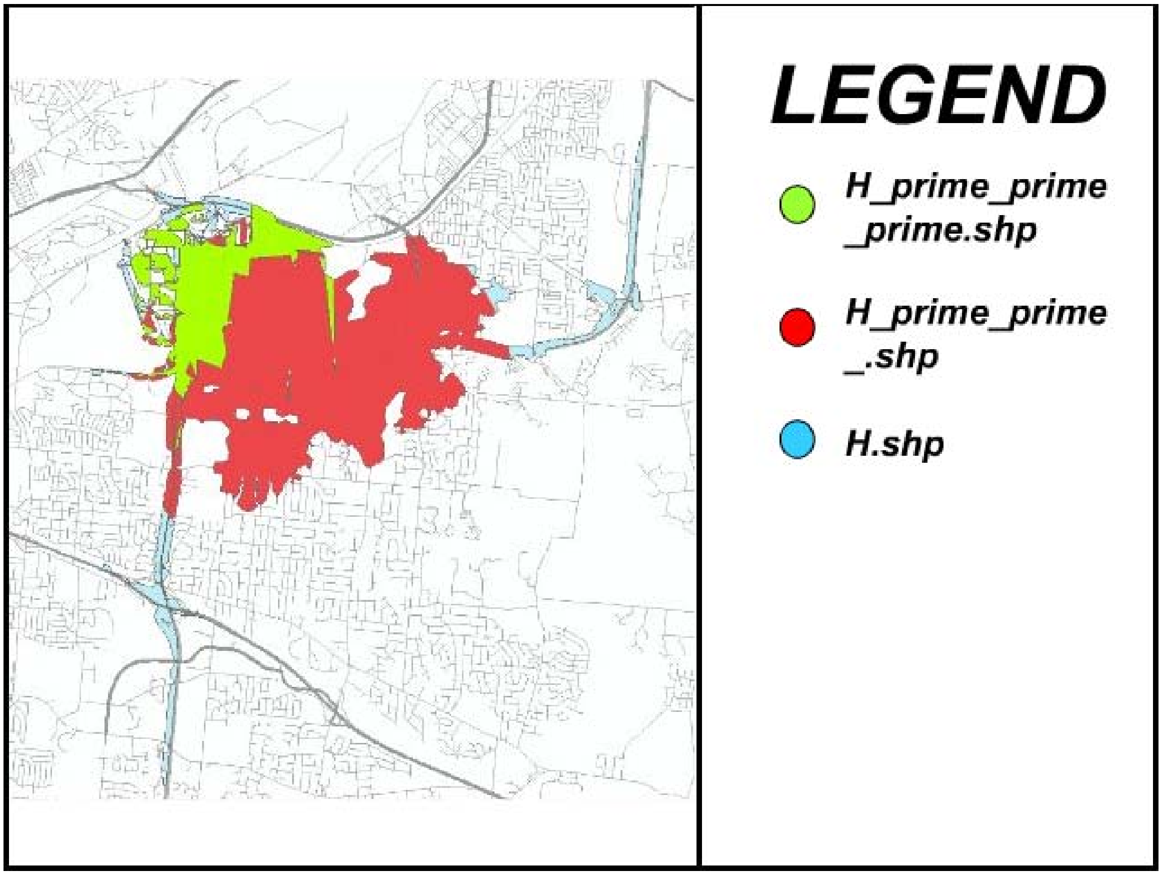


Figure 4.23: The efficiency functions $R(h'_{s'},\ h_{s'})$, $R(h''_{s'},\ h_{s'})$, and $R(h'''_{s'},\ h_{s'})$ are a measure of the ratio of polygons $h'_{s'}$, $h''_{s'}$, and $h'''_{s'}$ and the original alpha hull $h_{s'}$. Polygons are **h_5.shp**, **h_prime_5.shp**, **h_prime_prime_prime.shp** from the *TechEdge-Xenia* tracklet at $T = 5$. **h_prime_prime_5.shp** is not shown.

area, though we can not say whether it was worthwhile. Therefore Equation 3.33 is employed to calculate the accuracy F of the process-runs. This function is expressed as $F(O,\ O', H)$, $F(H')$, $F(H'')$, and $F(H''')$, and begins with the s'^{th} down-sampled point in **O'[s'].shp**. It subsequently finds its corresponding point in the full tracklet **O[s'].shp** and then counts up until the T^{th} point at **O[s' + T].shp**. Figures 4.24 and 4.25 show this portion of **O** in blue. F is simply the percentage of these points that fall within polygon $H,\ H',\ H''$, or H'''. Figures 4.24 and 4.25 show two possibilities. The former is a case when all of these points fall within the polygon and in this case $F = 1$. The latter is a case when some of the points exceed the boundaries of the polygon and $0 \leq F < 1$. Note that each measurement begins with downsampled point **O'[s'].shp** and may or may not exceed the $(s'+1)^{th}$ point. Whether it does or not, a new start is always made at the next downsampled point.

4.5.3 Experimental Results and their Interpretation . Tables B.1-B.24 in Appendix B document the results for both R and F for each of the eight tracklets for $T = 1,\ 3$, and 5 minutes and a sense of their appearance (for $T = 0$ only) may be gained from Figures A.1-A.8 in Appendix A. The mean values for each tracklet are summarized in Table 4.5 below. On inspecting the numerical data, there are several conclusions that may be reached. First, there are no obvious trends that emerge from this data as T is varied from $1,\ 3$, and 5 minutes or the process varies over the different tracklets. Second, the improvements were most marked from $R(H,\ M)$, though the $R(H',\ M)$ results are also substantial. It is no surprise that $R(H'',\ M)$ offers the least improvement.

What does surprise is that the improvement of $R(H''',\ M)$ is substantial when it worked properly. Not all tracklets resulted in a H''' (Jamestown-Xenia, and AFIT-TechEdge) because the vehicle never entered a major road or highway. Two (Home-Animal Shelter, TechEdge-Xenia) failed. The first failed because, after exiting a ramp, the vehicle continued into the tessellation region W_n and then exited it to continue for another several miles. The second failed because the tessellation region

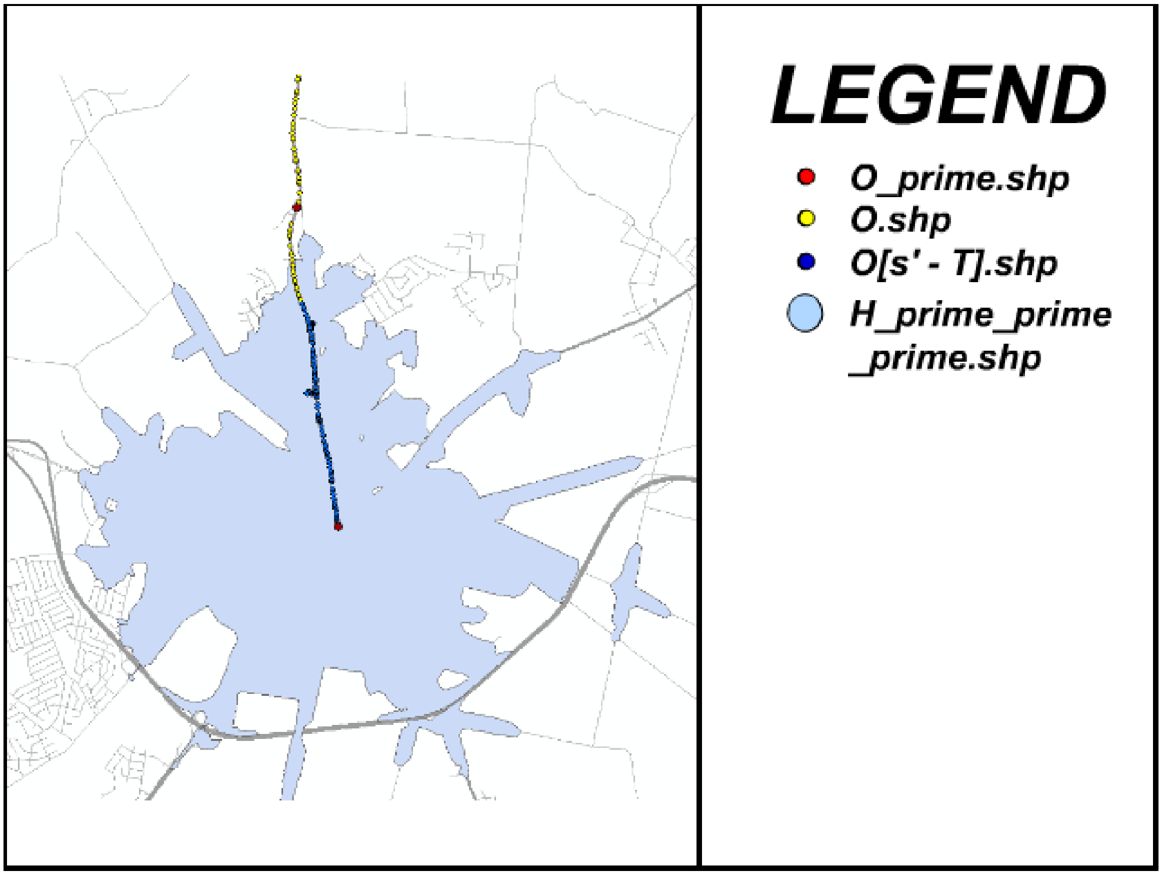


Figure 4.24: The accuracy functions $F(H)$, $F(H')$, $F(H'')$, and $F(H''')$ are a measure of points **O[s'].shp** to **O[s'+T].shp** that fall within the bounding polygons. Pictured is **O.shp**, **O_prime.shp**, **O[s'-T].shp**, and **h_1.shp** from the *TechEdge-Xenia* tracklet at $T = 5$. Note in this example that $F = 1$.

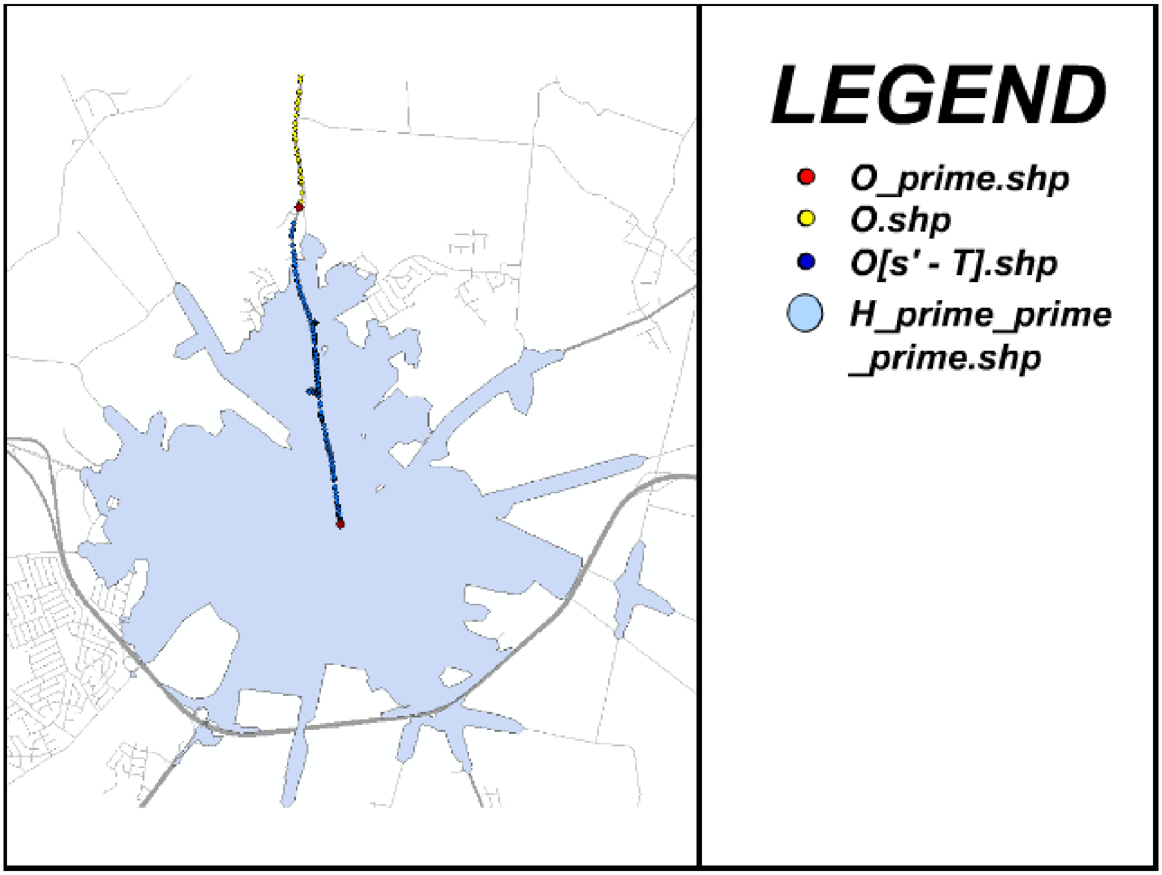


Figure 4.25: The accuracy functions $F(O,\ O', H)$, $F(O,\ O', H')$, $F(O,\ O', H'')$, and $F(O,\ O', H''')$ are a measure of points **O[s'].shp** to **O[s'+T].shp** that fall within the bounding polygons. Pictured is **O.shp**, **O_prime.shp**, **O[s'-T].shp**, and **h_1.shp** from the *TechEdge-Xenia* tracklet at $T = 5$. Note in this example that $F = 0.96$.

W_n did not enclose the area that could truly be said to bound its ramp because it happened at a terminal hub where many competing highways (and hence, their tessellation regions) competed destructively. However, the other four tracklets where this operation succeeded (Fairborn-Jamestown, Huber Heights-TechEdge, TechEdge-Vandalia, and Xenia-AFIT) experienced a combined mean improvement of 42 %, a marked enhancement.

Even better sense of these numbers can be made by exploring the accuracy F of these process-runs. It comes as no surprise that $F(H)$ was the most accurate, with a combined mean for all runs of 81 %. $F(H')$ delivered a combined mean of 56 % which was also expected. Remember that, since H' is created with the standard distance algorithm within the first standard deviation, the accuracy is anticipated to be within 68 %. A quick calculation with $F(H)$ corroborates this statement: $F(H) \times .68 = 57.1$ %. Likewise, $F(H'')$ yields a combined mean of 75 % and, with the second standard deviation within 95 %, $F(H) \times .95 = 76.9$ %. $F(H''')$ (when it applied and when it worked) offered an accuracy of $F(H''') = 82$ %.

Finally, an attempt must be made to determine why $F\,(H)$ stayed in the low eighties rather than being close to one hundred percent. The reason, simply put, is that, while carrying out the data-collections for tracklets O, the author exceeded the speed-limit often enough to confound the model. It should be emphasized that the author does not exceed the speed-limit gratuitously (or dangerously) but is akin to following the general flow of traffic. This exposes a central weakness of the model, which employed Equation 2.2 to determine the cost Ω of road-segments $E \in G(V, E)$. This equation asks for the length of the road-segment to be divided by the posted speed-limit, furnished as field **SL** in Table 4.2 for the road-segments E data-set. A more judicious approach might have taken into account the fact that many drivers drive five to ten KPH faster than posted speed-limits. Hence, the space mean speed approach of Eqs. 2.4 and 2.6 would have performed better in furnishing the field **SL** than the county data for the posted limits. If resources and time did not permit the statistical data collection involved with Eqs. 2.4 and 2.6, a simple buffer of five MPH might have sufficed to raise the accuracy F to close to one hundred percent.

With Efficiency R and Accuracy F figures of merit taken together, we may make the following conclusions:

1. Polygon H, a simple alpha hull created from a down-sampled point, offers a clear advantage over its siblings as being the best combination of efficiency and accuracy.
2. Polygon H', an alpha hull that has been intersected with a first standard deviation standard distance disk, is a useful knowledge product if accompanied with the caveat "within 68 % certainty".
3. Polygon H'', an alpha hull that has been intersected with a second standard deviation standard distance disk, is the least useful because, at the expense of great computation, it delivers the least amount of improvement over H, and still with the caveat "within 95 % certainty".
4. Polygon H''', a singleton set that is created if the *Slow-Fast-Slow* state has been detected by intersecting the corresponding Tessellation region with H, offers an excellent return for the computation, though it must be governed by a robust reasoning engine.

4.6 Chapter IV Summary

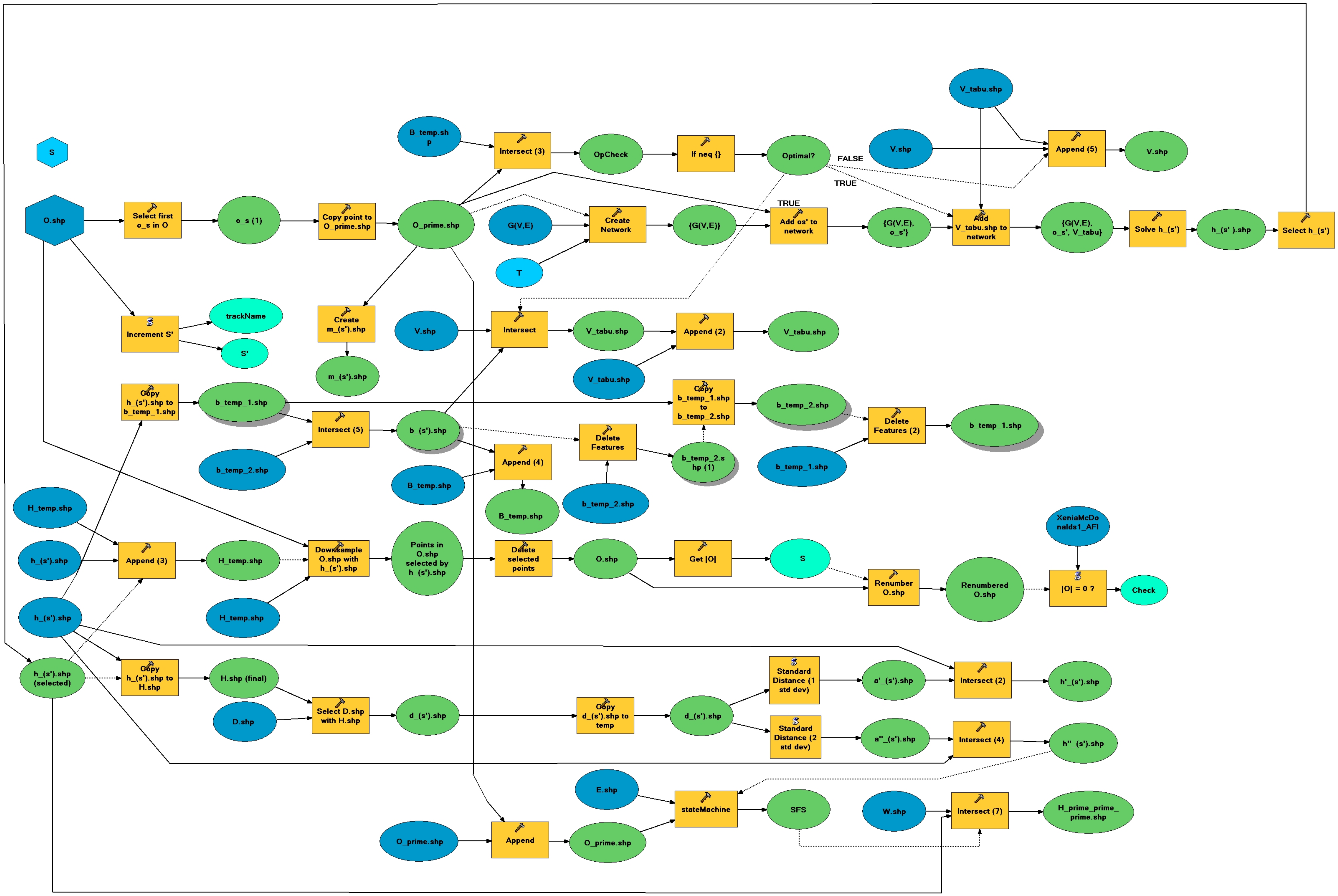
It has been shown in this chapter that powerful searches of an urban grid can be conducted by harnessing readily obtainable data-sets from either commercial or municipal sources (though we inferred that there were emerging military sources in Chapter I). These data-sets were then converted to a graph-theoretic model using the ESRI suite. The tracklets, though not real or near-real time, were easily simulated with the Leica GPS1200 Surveying System which resulted in 1 Hz tracklets made accurate with differential GPS correction. Search utilizing alpha hulls, standard distances and tessellation regions of the space yielded knowledge products in the form of

Table 4.5: Summary of Mean Results T = 1, 3, 5.

	R (Efficiency)				**F (Accuracy)**			
O	**H**	$\mathbf{H}'$	$\mathbf{H}''$	$\mathbf{H}'''$	**H**	$\mathbf{H}'$	$\mathbf{H}''$	$\mathbf{H}'''$
Afit-TechEdge (T = 5 min)	0.98	0.19	0.00	0.00	0.91	0.91	0.91	0.00
Afit-TechEdge (T = 3 min)	0.98	0.27	0.00	0.00	1.00	1.00	1.00	0.00
Afit-TechEdge (T = 1 min)	0.97	0.25	0.00	0.00	0.75	0.64	0.75	0.00
Fairborn-Jamestown (T = 5 min)	0.97	0.40	0.05	0.22	0.78	0.42	0.65	1.00
Fairborn-Jamestown (T = 3 min)	0.98	0.41	0.08	0.10	0.76	0.40	0.68	1.00
Fairborn-Jamestown (T = 1 min)	0.98	0.32	0.06	0.00	0.79	0.44	0.67	0.54
Home-Animal Shelter (T = 5 min)	0.97	0.39	0.03	0.00	0.73	0.40	0.73	0.00
Home-Animal Shelter (T = 3 min)	0.98	0.37	0.04	0.00	0.68	0.44	0.61	0.00
Home-Animal Shelter (T = 1 min)	0.97	0.37	0.05	0.00	0.64	0.41	0.58	0.00
Jamestown-Xenia (T = 5 min)	0.96	0.42	0.02	0.00	0.91	0.70	0.91	0.00
Jamestown-Xenia (T = 3 min)	0.98	0.39	0.08	0.00	0.92	0.62	0.87	0.00
Jamestown-Xenia (T = 1 min)	0.97	0.34	0.05	0.00	0.84	0.58	0.78	0.00
Huber-TechEdge (T = 5 min)	0.97	0.52	0.05	0.60	0.90	0.41	0.74	1.00
Huber-TechEdge (T = 3 min)	0.97	0.42	0.10	0.66	0.81	0.54	0.75	0.84
Huber-TechEdge (T = 1 min)	0.97	0.37	0.08	0.95	0.79	0.54	0.62	0.78
TechEdge-Vandalia (T = 5 min)	0.97	0.29	0.02	0.68	0.87	0.51	0.81	1.00
TechEdge-Vandalia (T = 3 min)	0.98	0.44	0.00	0.58	0.79	0.64	0.74	1.00
TechEdge-Vandalia (T = 1 min)	0.97	0.40	0.09	0.00	0.88	0.59	0.84	0.95
TechEdge-Xenia (T = 5 min)	0.97	0.35	0.02	0.74	0.69	0.48	0.58	0.00
TechEdge-Xenia (T = 3 min)	0.98	0.36	0.07	0.1	0.68	0.45	0.60	0.00
TechEdge-Xenia (T = 1 min)	0.98	0.36	0.07	0.32	0.58	0.37	0.52	0.00
Xenia-AFIT (T = 5 min)	0.96	0.37	0.05	0.81	0.99	0.72	0.96	0.78
Xenia-AFIT (T = 3 min)	0.97	0.35	0.03	0.41	0.92	0.59	0.81	1.00
Xenia-AFIT (T = 1 min)	0.97	0.23	0.00	0.00	0.84	0.69	0.84	0.00
Mean	0.97	0.36	0.04	0.26	0.81	0.56	0.75	0.41

polygons bounding the space where a surveilled vehicle could expected to be within a time horizon T. Finally, a mechanism for judging the efficiency and accuracy of these polygons was devised.

The next chapter will demonstrate that many of the approaches in this work are unique innovations that have exploited emerging technology in powerful and insightful ways. It will end with a brief look at possible work for the future and how this process may give a competitive edge to prospective war-fighters and their weapons systems. Finally, Appendix C will illustrate that there are many practical military applications for these products that may be employed, per our specifications in Chapter I, in the spectrum of the battlespace that ranges from the tactical to the operational levels of war.


S
O.shp
Select first o_s in O
o_s (1)
Copy point to O_prime.shp
O_prime.shp
B_temp.sh p
Intersect (3)
OpCheck
If neq {}
Optimal?
FALSE
TRUE
TRUE
V_tabu.shp
V.shp
Append (5)
V.shp
G(V,E)
Create Network
{G(V,E)}
Add os' to network
{G(V,E), o_s'}
Add V_tabu.shp to network
{G(V,E), o_s', V_tabu}
Solve h_(s')
h_(s').shp
Select h_(s')
T
Increment S'
trackName
S'
Create m_(s').shp
m_(s').shp
V.shp
Intersect
V_tabu.shp
Append (2)
V_tabu.shp
V_tabu.shp
Copy h_(s').shp to b_temp_1.shp
b_temp_1.shp
Copy b_temp_1.shp to b_temp_2.shp
b_temp_2.shp
Delete Features (2)
b_temp_1.shp
Intersect (5)
b_(s').shp
Delete Features
b_temp_2.s hp (1)
b_temp_1.shp
b_temp_2.shp
Append (4)
B_temp.shp
B_temp.shp
b_temp_2.shp
H_temp.shp
h_(s').shp
Append (3)
H_temp.shp
Downsample O.shp with h_(s').shp
Points in O.shp selected by h_(s').shp
Delete selected points
O.shp
Get |O|
S
Renumber O.shp
Renumbered O.shp
XeniaMcDonalds1_AFI
|O| = 0 ?
Check
H_temp.shp
h_(s').shp
h_(s').shp (selected)
Copy h_(s').shp to H.shp
H.shp (final)
D.shp
Select D.shp with H.shp
d_(s').shp
Copy d_(s').shp to temp
d_(s').shp
Standard Distance (1 std dev)
a'_(s').shp
Intersect (2)
h'_(s').shp
Standard Distance (2 std dev)
a''_(s').shp
Intersect (4)
h''_(s').shp
E.shp
stateMachine
SFS
W.shp
Intersect (7)
H_prime_prime_ prime.shp
O_prime.shp
Append
O_prime.shp

V. Conclusions

The exertions of Chapter V, by their technical nature, might have obscured the main objectives of this work, and it will be useful to render an accounting of what has been accomplished. Therefore, this chapter sums up the contributions of this work that are unique to the field and recommendations for future work.

5.1 Unique Contributions of this Work

It is worth sparing a few moments to enumerate the unique contributions that this thesis offers to the field. Recall that the baseline from which this effort began was the Dynamic Tactical Targeting project, managed by the Air Force Research Laboratory. The prime contractor for DTT was Lockheed-Martin, and though their objective was not primarily vehicle prediction (instead, it was tracking), they made the first serious attempt to utilize search-driven prediction using *Dijkstra's* algorithm.

Our improvements over this system are several. Among these, DTT employed a *Dijkstra's Grid* which compares roughly to our network $G(V, E)$ though it did not conform to roads, highways or other architecture. Instead, *Dijkstra's Grid* was a checkerboard lattice $D(V, E)$ where the vertices V represented ground hardness and edges E represented a cost function based upon the calculated difficulty to traverse its length according to the soil measured at vertices V. Values for V were collected over a test area with a *penetrometer*, literally a cone with graduated markings, that is pushed into the soil in order to measure its penetrability. The resulting square lattice $D(V, E)$ is then searched as we have done from a given point, yielding an alpha hull that constrains the area.

One obvious shortcoming for this approach is data collection. While it is feasible to conduct such a data-collect on a parcel of land in a rural area in the United States, it is decidedly less so in, say, the Kunar Valley in Afghanistan. The shear difficulty in collecting such a data-set over a wide area must have made certain customers balk at this approach. By way of comparison, our model $G(V, E)$ was built in less than a day from municipal sources and was superior in several respects. First, the data-set was less prone to variability (imagine how the DTT data-set would suffer during a rain storm). Second, our data-set was a far richer one, which allowed it to take more information into account. In addition to discovering the possible traversals of the space, our approach exploits the richness of the data-set by interrogating distributions of buildings (the **H_prime.shp** and **H_prime_prime.shp** products). In sum, though the DTT effort utilized GIS technology, it did not capitalize on the profound, though often latent, abilities of this technology to farm additional knowledge by considering *context*. The process described in this thesis not only uses a better data-set for the Dijkstra search, and exploit contextual elements, but it is also extensible enough to take on additional data, should it become available.

Another drawback to the DTT approach is that it did not search an entire tracklet as we have demonstrated, but it searched from a single point, the assumption being that if there was only one possible observation, then the DTT system could make a reasonable prediction from that point. The fact that our process down-samples a tracklet, with the very product that it works to create, is perhaps one of the greatest innovations of this thesis. Consider the economy of effort that this affords: it is akin to crossing a stream by creating one's own stepping stones from thin air. The practical benefit of this is that it allows entire tracklets to be considered rather than just stationary points. To that is added the shear practicality of the result. We may say during the process that "the vehicle will be within polygon **h_2.shp** from now until T." This allows effects to be vectored to the bounded area up until T, where, if subsequent target-prosecution is not as desired, the process may repeat because **h_3.shp** is created.

A significant consequence of being able to generate many alpha hulls from a tracklet, as it unfolds, is the ability to formulate the negative space entailed by our tabu list $\overline{G(V, E)}$. This innovation draws its inspiration from the work of Weeks and Nanda [24] discussed in Section 2.5, which led to the formulation of the two types of journeys, *Deliberate* and *Circuitous* and their consequent assumptions, *Destination Directed* and *Slow-Fast-Slow*. It is important to point out that the reason the experimentation and results analysis portion of Chapter IV did not test examples of *Circuitous* journeys is that the primary aim of the thesis was to develop a process for when the key assumptions held. It will be recalled from Section 4.4.9 that when a vehicle intersects a negative space polygon that the $NListener(O,\ B,\ s')$ event-handler will add the tabu list back to $G(V,E)$, an event that could not yield trend data that would be of any interest. That being said, this thesis refined the proposals of Weeks and Nanda into the formal assumptions described throughout, exploited the byproduct of the negative space, and dealt successfully with the possibility that the assumptions could at any time be violated. This last item led to the possibilities of semantics generation, an unlooked-for advantage that, nevertheless, allows the work to fit seamlessly with other work being carried out at AFIT [13] whose express object is indeed semantics.

The search and prediction portion of DTT suffered from several disabilities that made it, ultimately, infeasible. Chief among these is the difficulty in procuring data-sets that it required for *Dijkstra Searches*. Another was its inability to deal with true tracklets, essential for real or near-real time functionality. It was also an extremely proprietary, stove-piped system that would have required perpetual service-level agreements with the vendor until end-of-life, without the saving grace of being extensible, scalable, or interoperable with other systems. The work presented in this thesis, in contrast, relies on commercial off-the-shelf software. The ESRI suite is already employed world-wide by Air Force Civil Engineering and Communications activities and enjoys undisputed market-share for municipal and commercial GIS and geoprocessing. Moreover, the Software-as-a-Service business model proposed in Section C.0.1 would allow this solution to be provisioned over existing architectures such as Distributed Common Ground Station (DCGS) at a minimal cost, and practically out of the box. This business model would also provide coverage for a much larger customer base because the geoprocessing services could be streamed to any web-aware device. The next section will take these robust qualities into consideration as it explores some practical uses for the knowledge-products that are created from surveilled tracklets by this process.

5.2 Recommendations for Future Work

In the expectation that this work is to be of use to the Air Force Research Laboratory's Sensor Directorate at some future date, it is therefore worthwhile to embark on a brief discussion of improvements that may be made to the process and areas of future research. It will also be of immediate benefit to illustrate the ways in which this may be integrated with current projects that are underway. Finally, we would be remiss if we did not attempt to conjecture other possible applications in the battle-space (including joint applications) that might immediately prove to be of use in the current conflicts being waged by the United States.

5.2.1 Improving on the Process. It bears re-mentioning that this project was executed in ESRI Model Builder, essentially a prototyping environment. To that end, no attempt was made to optimize the execution of the process for faster execution times. Although the process managed to run in ten percent of the time that the simulated tracklets actually took to unfold, the premise has been that this process is intended to run in real or near-real time in order to facilitate fast decisioning. Therefore, chief among the improvements that are recommended for this process is that optimizing strategies are implemented in order to achieve this. These could include, since the primary focus is search, the use of poly-tree data structures with their attendant $O(log\ n)$ search mechanisms for faster *Dijkstra Searches* (as it stands, searches are tabular SQL searches of the data-sets). Next, because of the many intermediate shape-files created and then used by subsequent functionalities in the algorithm, the use of cache objects would speed the process and economize memory use. Furthermore, especially in the case of the standard distance algorithm, vectorized multiplication and addition would deliver a vast enhancement to the running time.

5.2.2 Integration with the Multi-Layered Sensing "Data-Table" Construct. A fully functioning implementation of this process may be realized by combining these functionalities with the "Data-Table" construct owned by the Air Force Research Laboratory Sensors Directorate. Pictured in Figure 5.1, this includes a MERL Diamond-Touch Table, a "touch-and-gesture" forty inch computer screen, laid horizontal as its name suggests. The data-table supports multiple users and is ideal for GIS applications, especially in collaborative situations analogous to Command and Control environments. The AFRL data-table's standard PC has NASA's World Wind 1.4 GIS software, a fully interactive 3D globe that may be navigated to a desired locale and serves primarily as a front-end presentation environment for GIS data-sets. World Wind 1.4 is an open-source, freely available project managed by NASA and supports either the Microsoft .NET or Java runtimes.

The open-source character of World Wind 1.4 causes it to be extremely extensible to other systems, and the process described in this work (using the ESRI products) may be augmented to it easily and cost-effectively. Our process was demonstrated using the ESRI Model Builder, a proto-typing environment, used because it was desirable to explore the full powers of geoprocessing without being encumbered with formal software packaging. For the purposes of adding it to the data-table construct, the ESRI ArcGIS Engine 9.3® would allow all the functionalities described (and more) to be bundled and deployed within World Wind 1.4 environment. ArcGIS Engine 9.3® is a Software Development Kit (SDK) containing a library of managed component-based classes called ArcObjects. There are versions for the Microsoft .NET Framework and Java, and it is supported by a robust developer community. ArcGIS Engine® is supported on Windows, Solaris, and Posix systems, allowing for cross-platform functionality. With access to a relational database management system, multiuser editing applications are also possible.

5.2.3 Other Air Force and Joint Applications. One theme that has played a recurring role in this work is the idea that the process we describe is generally agnostic

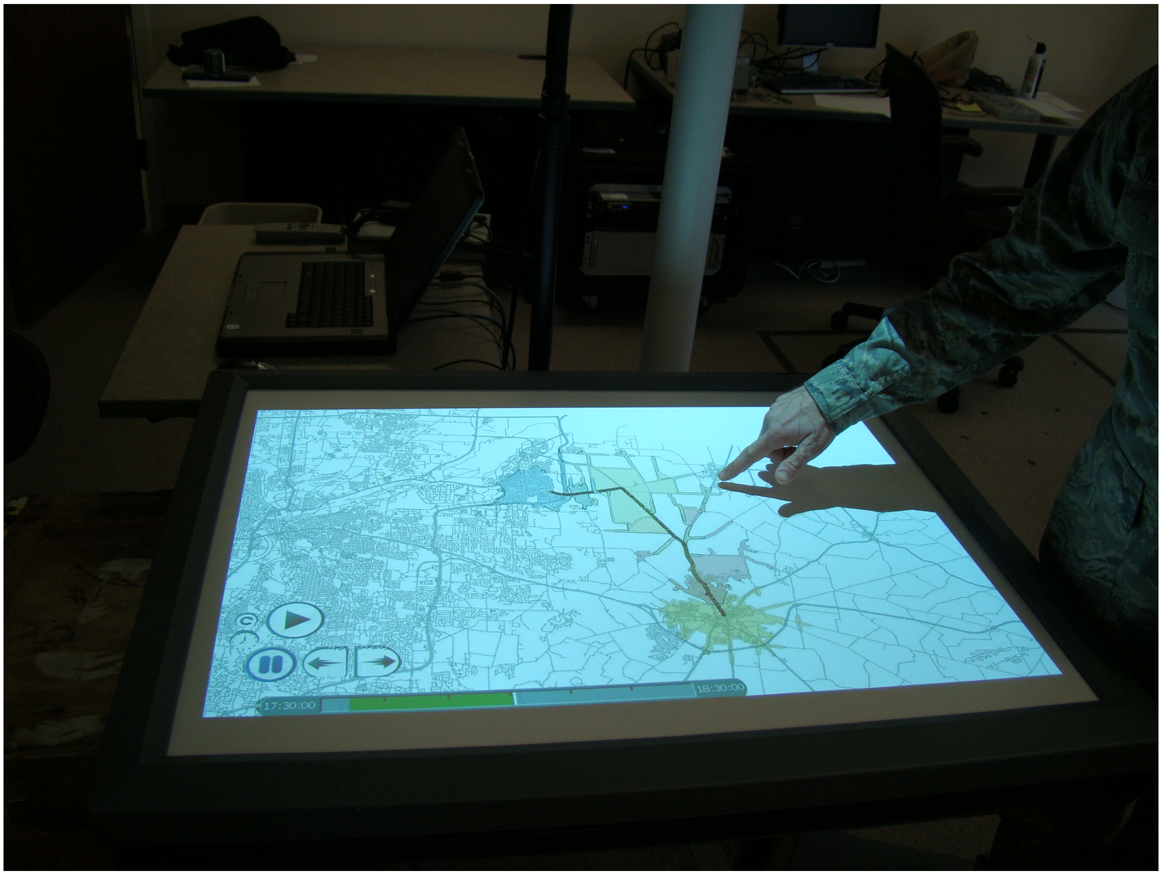


Figure 5.1: Pictured here is the MERL Diamond-Touch Table with World Wind 1.4 software. Vehicle tracklets and alpha hulls from our process are easy to view and manipulate collaboratively in this environment.

to particular ISR systems. This was no accident and accounts for our proposals of the Software as a Service in Appendix C business model, the wide spectrum of customer niches ranging from the tactical to the operational levels of war, and the layered sensing construct as prefaced in Chapter I. There exists a wide array of sensing platforms in use within the Air Force's arsenal that could profit from these proposals. The only constraints are that there be full-motion aerial videography (irrespective of spectrum), real or near-real time functionality, and a tracking mechanism for vehicles under surveillance. This description is general enough to admit a large (and growing) host of possible systems to include the Global Hawk, JSTARS, MQ-1 Predator, and MQ-9 Reaper. These systems exist solely for the ISR mission, though they may be joined by other specialized aircraft whose sensor suites are similarly powerful. In particular, the LANTIRN Targeting Pod and the Sniper XR Advanced Targeting Pod, for use on multirole fighter aircraft, are correspondingly well-endowed systems that could profit from such a Predictive Battle-Space Awareness. Rather than appearing on a data-table, our product would be equally informative within a Heads-Up Diplay giving instant feedback to Air Force pilots.

In addition to strictly Air Force systems, there are rich possibilities for joint application. Particularly, the US Army *Future Combat Systems* under contract to Boeing and Science Applications International Corporation (SAIC), would enjoy keen advantages if augmented with our process. This largely ground-based system of systems envisages unmanned aerial systems, unmanned ground systems, mobile-launch rocket systems, and other traditional weapons and support systems operating in a combined-arms construct. The US Army has displayed a keen aptitude for ISR-based missions of late: witness the Battle for Sadr City [45] in 2008 where UASs played a crucial role in the command and control decisioning that saw the eventual defeat of Moqtada al-Sadr's Mahdi Army. In addition to utilizing our data-table, within the *Future Combat Systems* environment, extreme tactical advantage could be gained if individual combat teams on the ground were equipped with wrist-worn devices such as the Israeli V-RAMBO, a wireless web-aware device radio-linked with UAS's and endowed with GIS capabilities. Since the *Future Combat Systems* envisions Raven and DragonEye UASs being organic to platoon and brigade command echelons, this becomes an extremely feasible proposal.

5.3 On Technology and Troops

The author would like to end this work by disclosing a view that, though a personal conceit, bears strongly on this work. Boeing, Lockheed-Martin, Northrop, L3-Titan Group, Hughes Missile Systems, and the other khanates of military procurement, while having served our country well, will never be able devise a weapon system that is as cunning, ruthless, and deadly as the US Air Force pilot, the Army combat soldier, the Navy sailor, and the Marine Corps rifleman. To borrow from one of the Air Force's preeminent thinkers, Colonel John Boyd, "People first, then ideas, then technology... *in that order*!" [44]. This work has proceeded with utmost faith in this axiom and it is the hope of the author that, rather than allowing ourselves to become too reliant on technology, that we ought instead to use technology to augment what is already our greatest asset: our war-fighters.

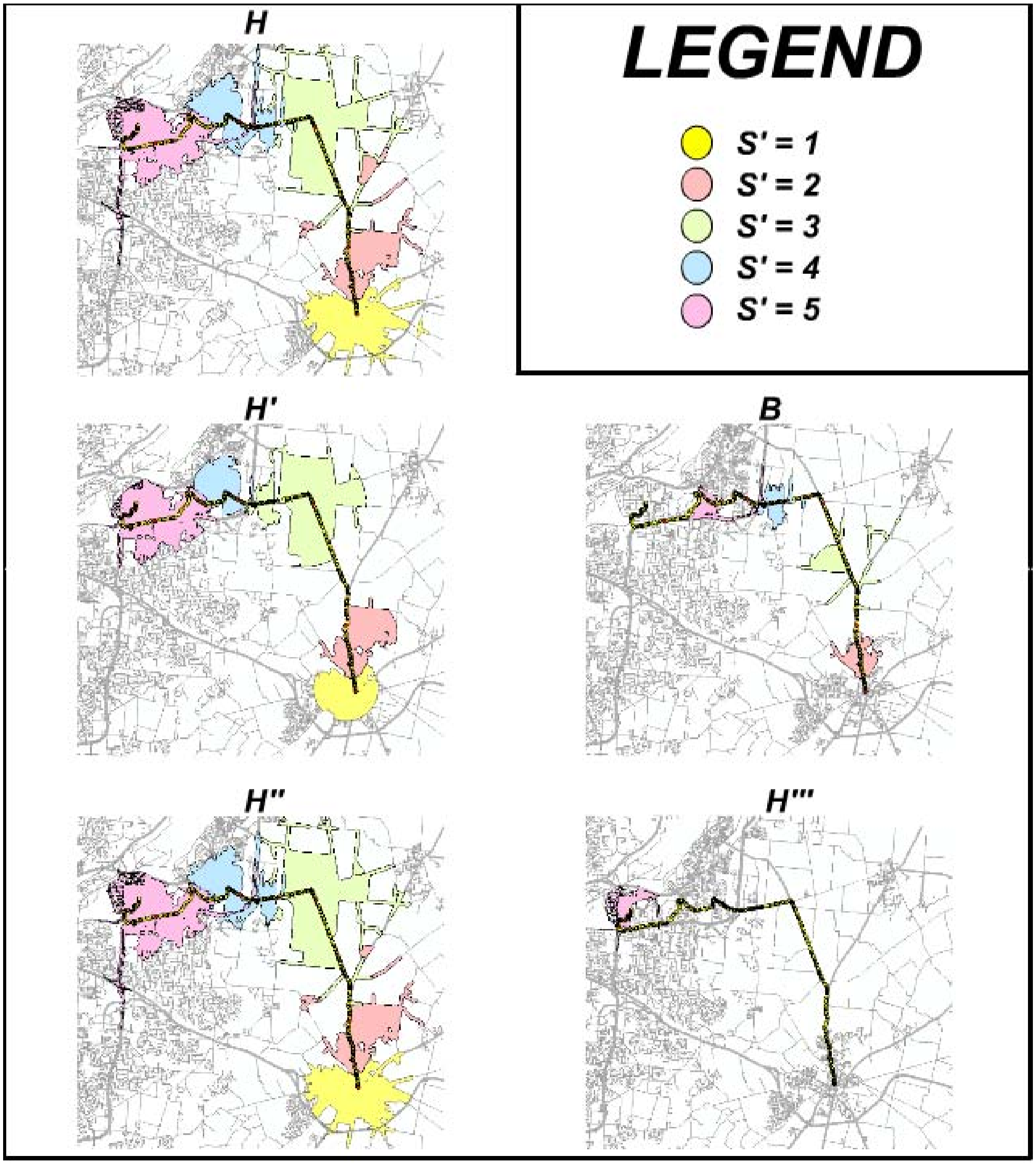


Figure A.1: Pictured here is the Xenia-AFIT run. Note that the **H_prime_prime_prime.shp** product is significantly smaller than the other bounding polygons.

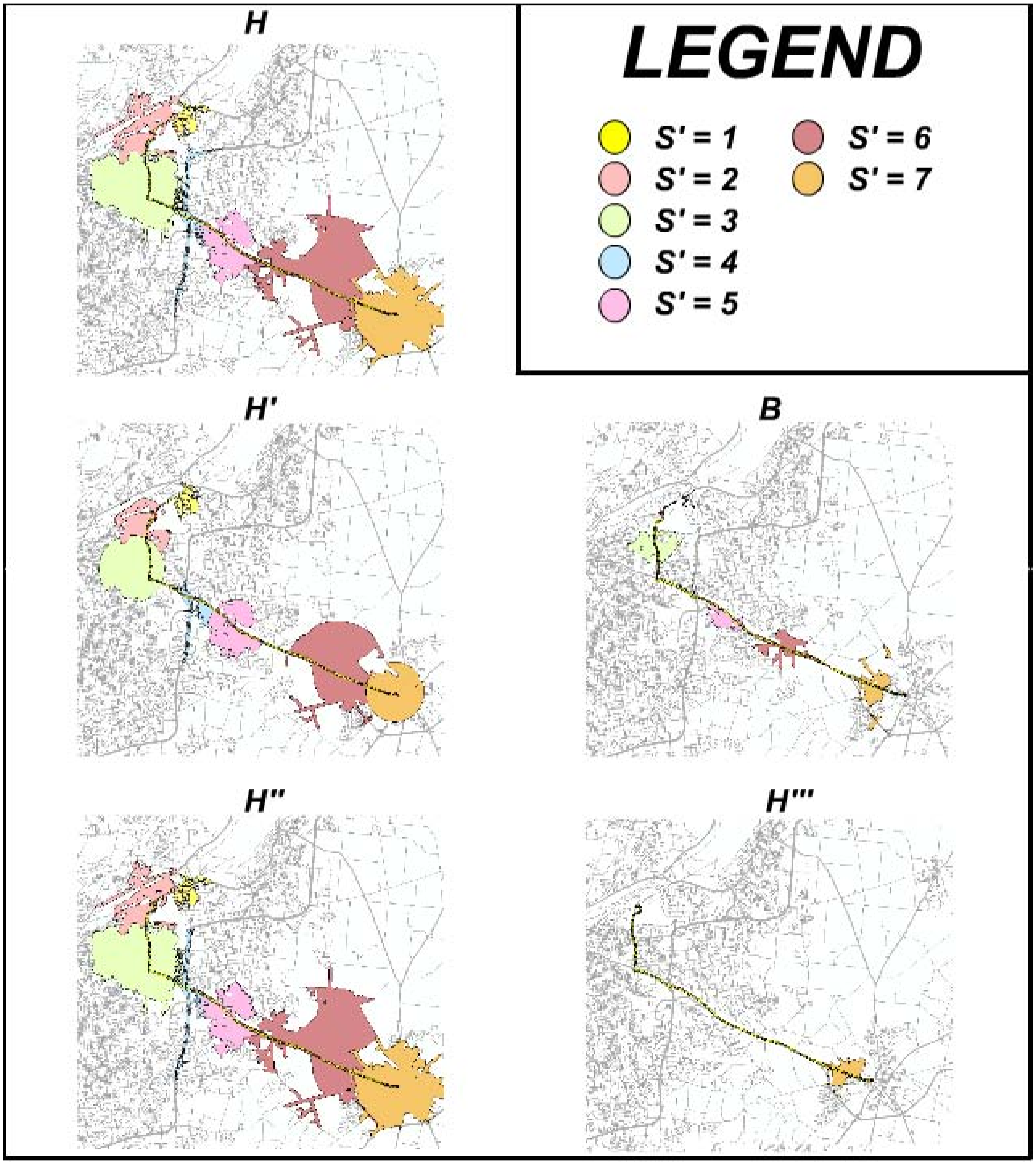


Figure A.2: Pictured here is the TechEdge-Xenia run. Note that in this case, **H_prime_prime_prime.shp** does not enclose the destination and therefore fails.

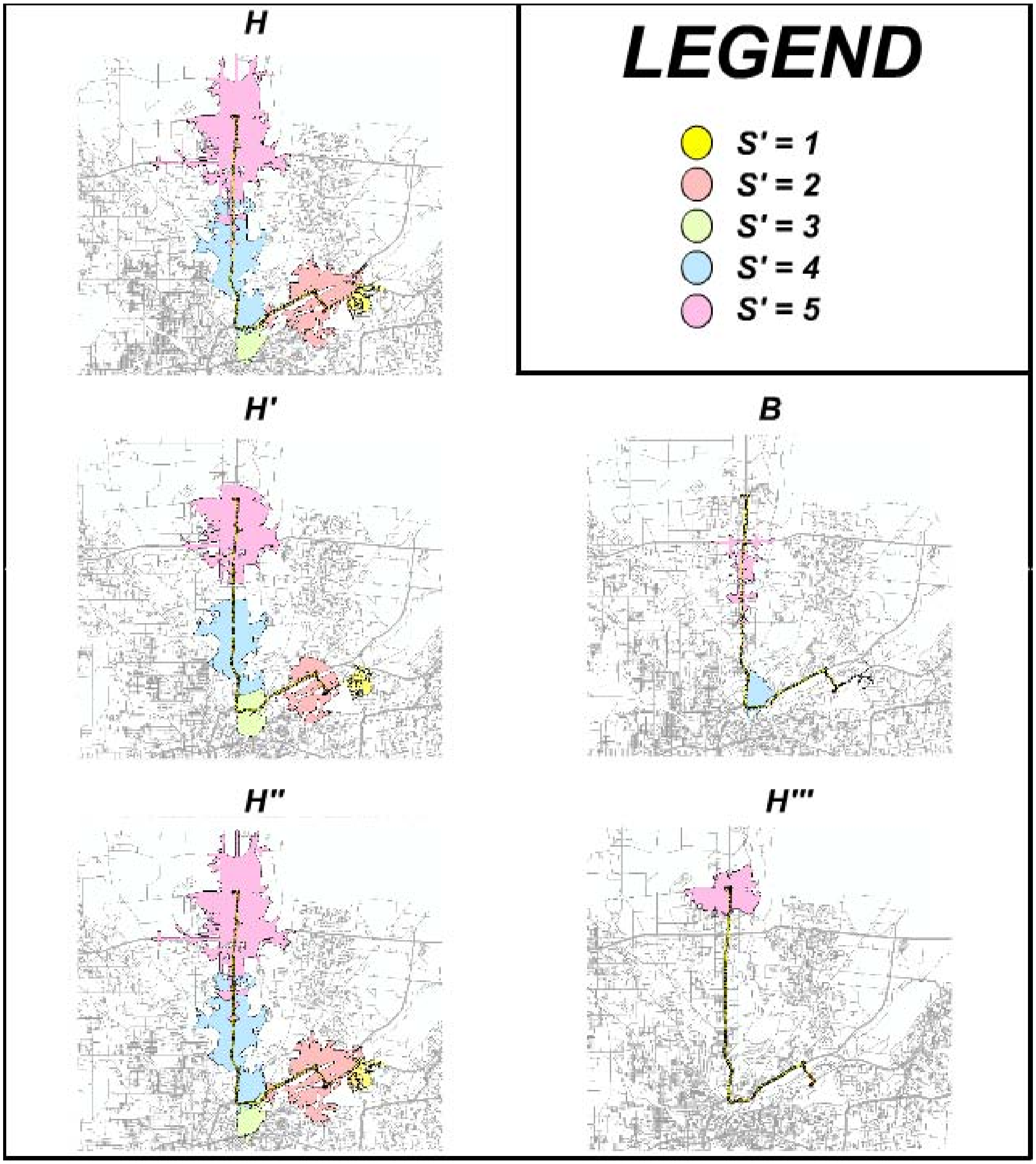


Figure A.3: Pictured here is the TechEdge-Vandalia run. Here, the **H_prime_prime_prime.shp** succeeds and represents considerable improvement over **H.shp**.

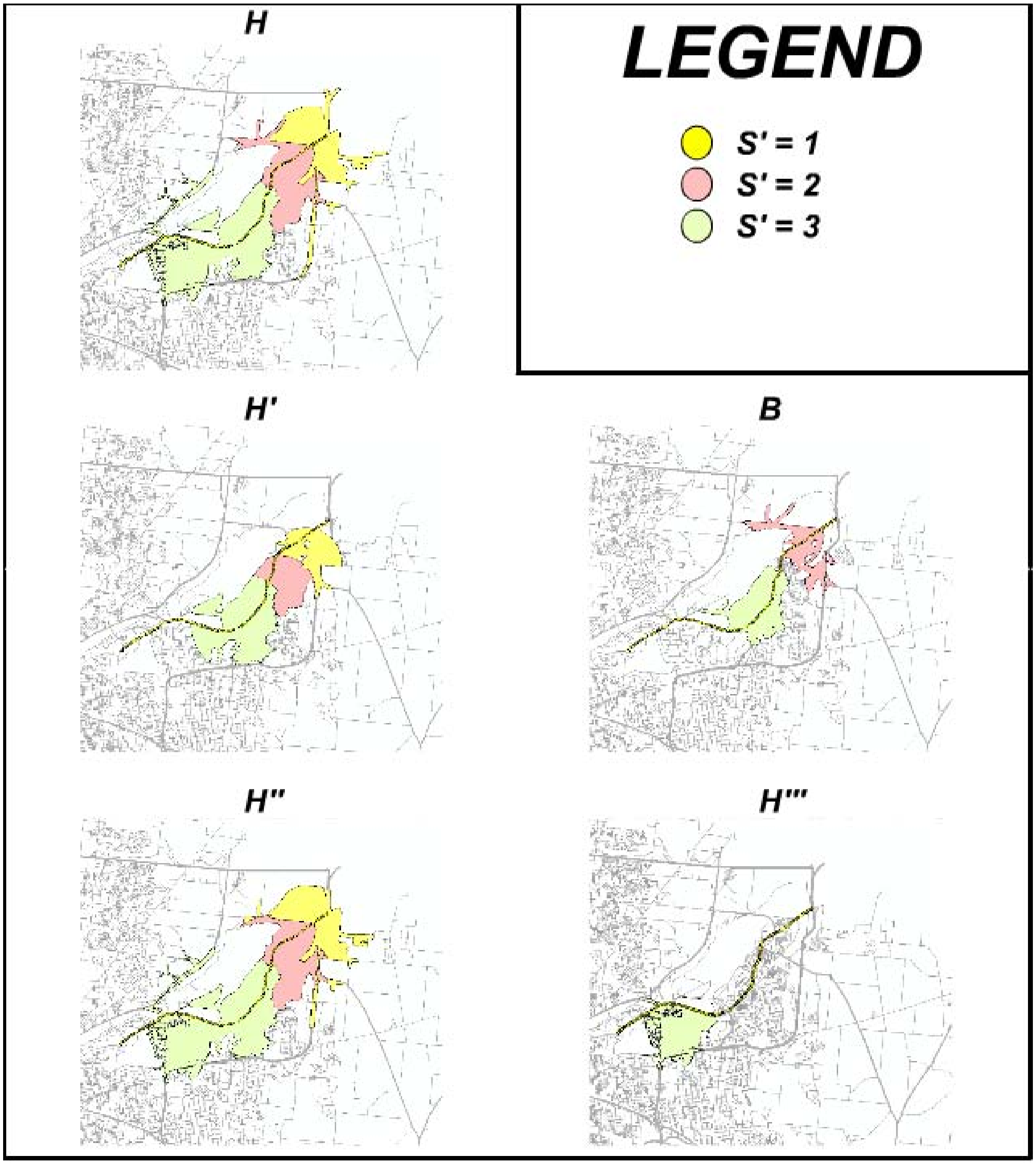


Figure A.4: Pictured here is the HuberHeights-TechEdge run. The **H_prime_prime_prime.shp** file appears to fail in this screen-shot. However, examination at enlarged extent reveals that the polygon encloses the entire tracklet.

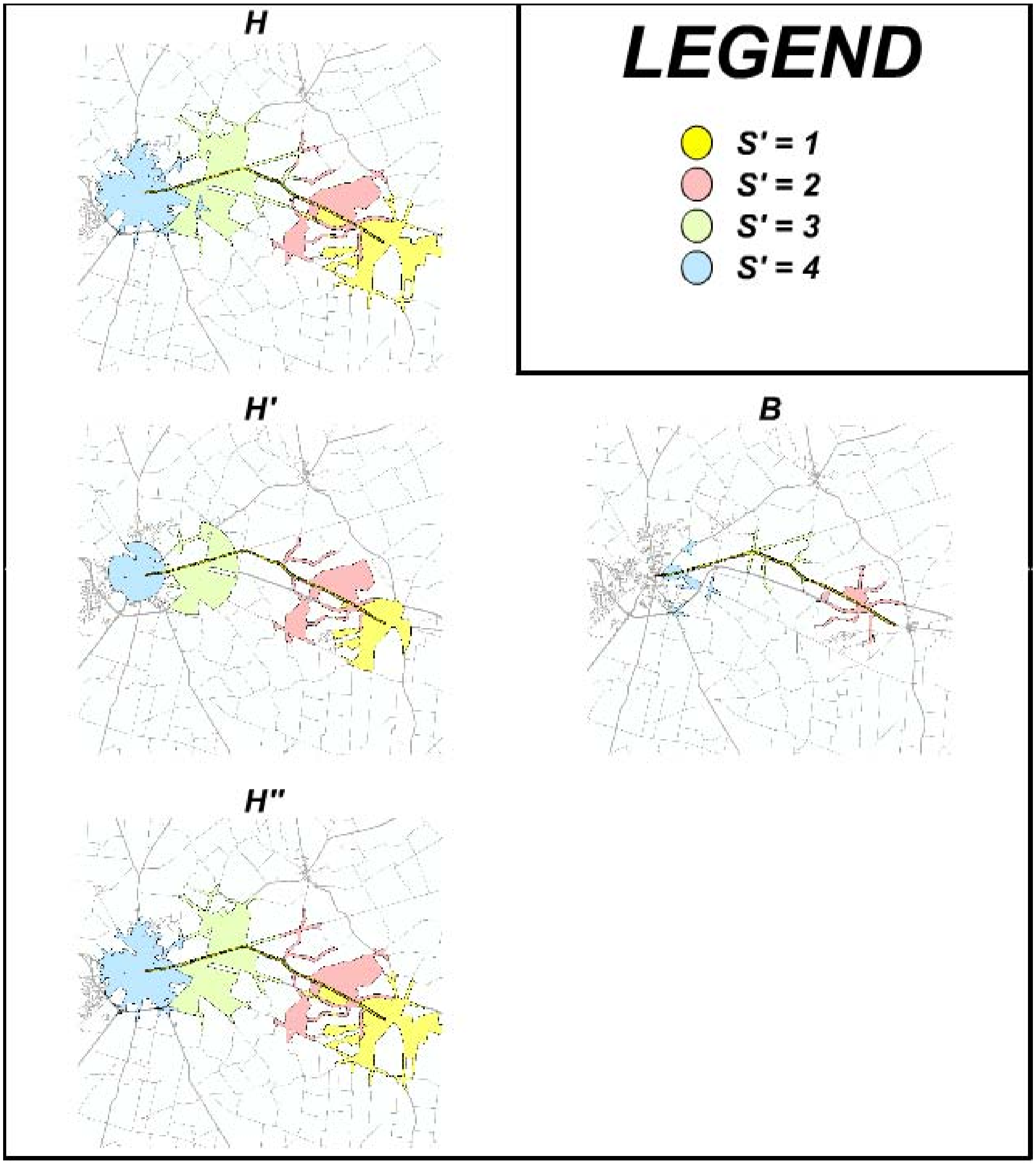


Figure A.5: Pictured here is the Jamestown-Xenia run. Note that no **H_prime_prime_prime.shp** was generated. This is due to the fact that the vehicle never entered a major road or highway.

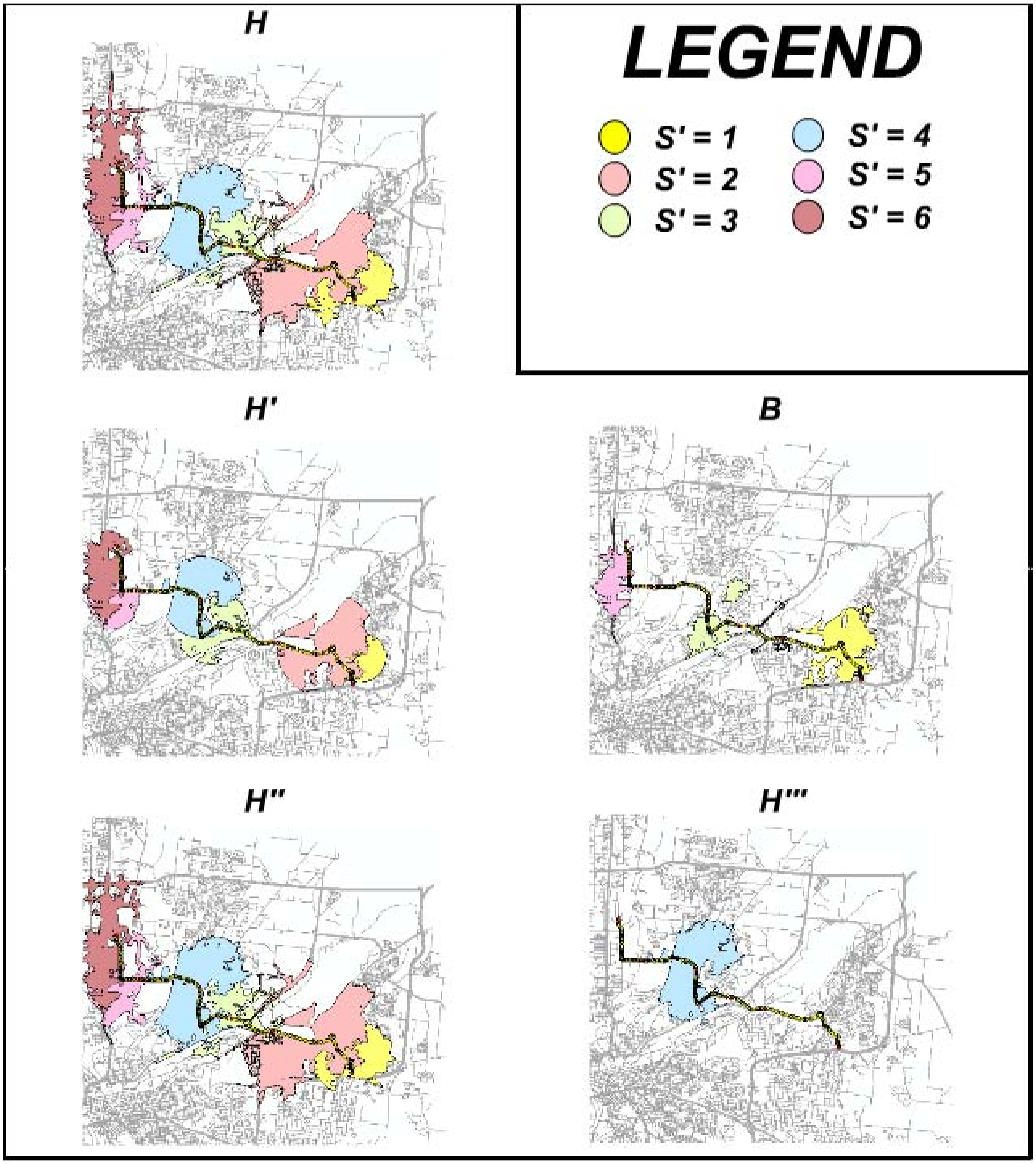


Figure A.6: Pictured here is the Home-Animal Shelter run. The **H_prime_prime_prime.shp** does not manage to enclose the final destination. It therefore fails.

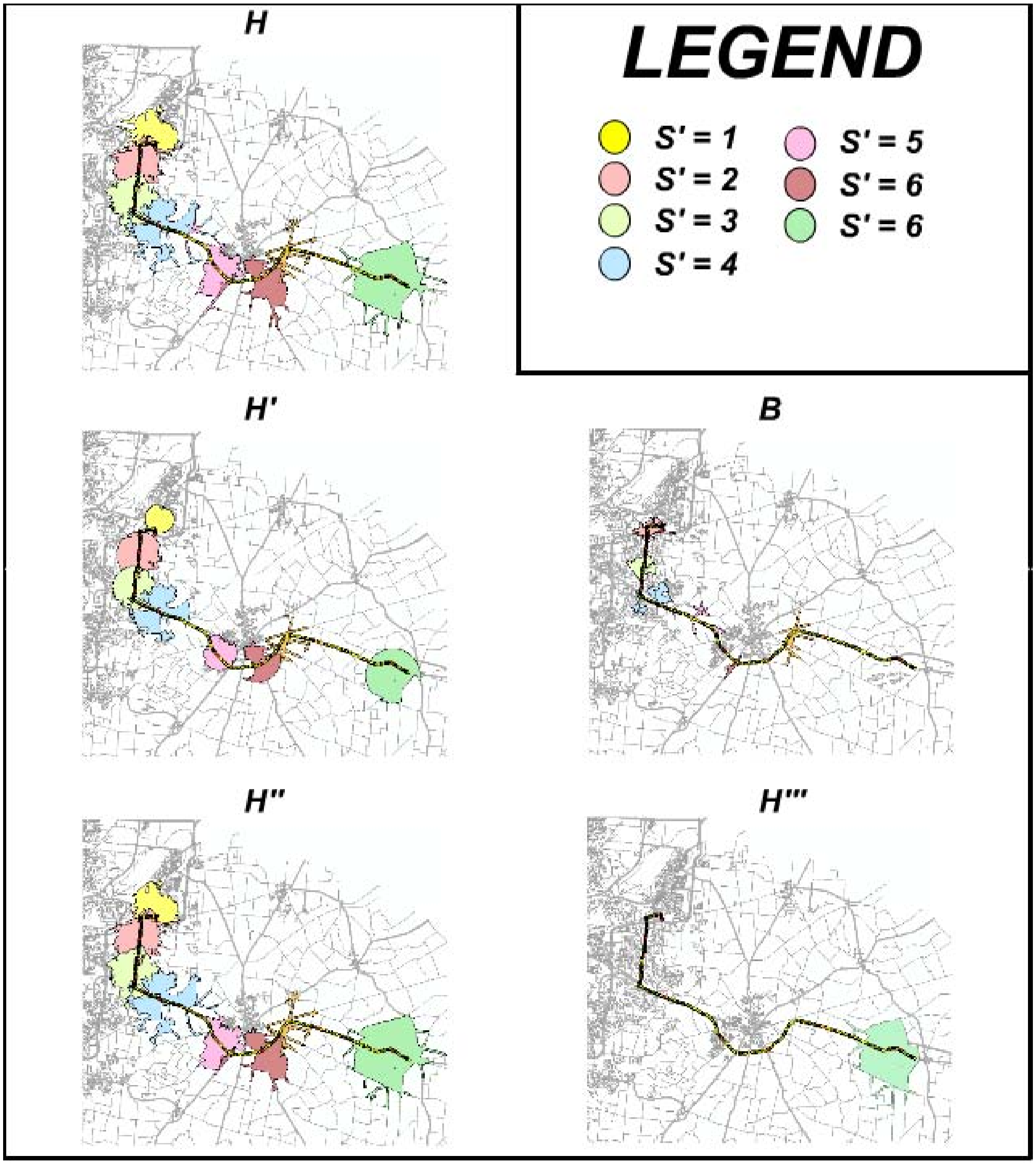


Figure A.7: Pictured here is the Fairborn-Jamestown run. The **H_prime_prime_prime.shp** succeeds but does not prune the space as well as others did.

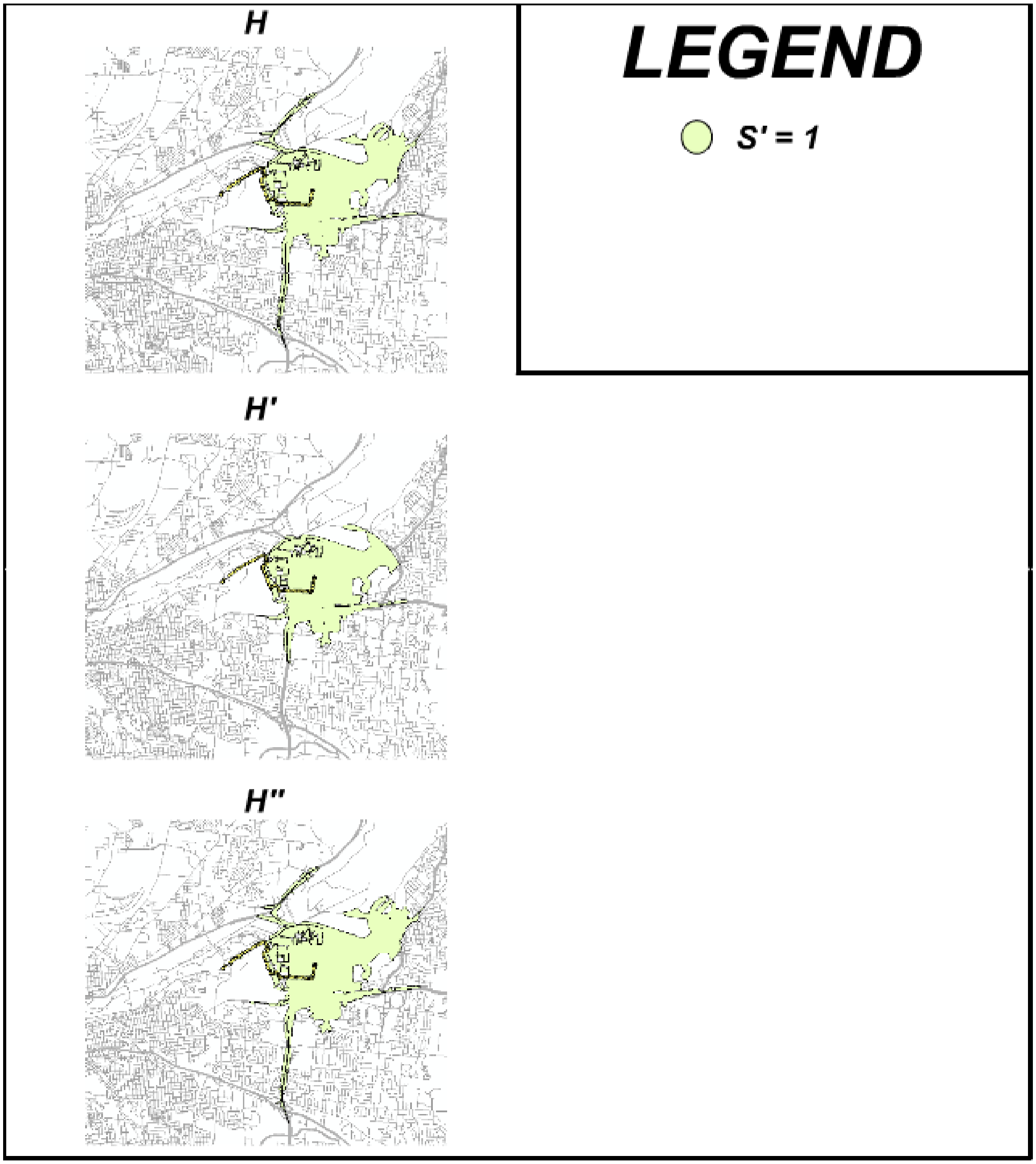


Figure A.8: Pictured here is the AFIT-TechEdge run. Like the Jamestown-Xenia tracklet, it does not enter a major artery and there is therefore no **H_prime_prime_prime.shp**. Additionally, there is no **B.shp**. Unlike the others, there was only one polygon **H.shp** generated for $T = 5$.

Table B.1: Xenia-AFIT T = 5min.

	R (Efficiency)				**F (Accuracy)**			
S’	**H**	**H’**	**H”**	**H”’**	**H**	**H’**	**H”**	**H”’**
1	0.96	0.39	0.00	N/A	1.00	0.29	0.37	N/A
2	0.97	0.38	0.05	N/A	0.82	0.22	0.49	N/A
3	0.97	0.36	0.00	N/A	1.00	0.73	1.00	N/A
4	0.97	0.20	0.00	N/A	1.00	0.94	1.00	N/A
5	0.97	0.37	0.10	N/A	1.00	0.92	1.00	N/A
6	0.97	0.33	0.01	N/A	0.84	0.68	0.84	N/A
7	0.96	0.60	0.10	N/A	0.69	0.30	0.63	N/A
8	0.98	0.26	0.01	N/A	0.96	0.23	0.96	N/A
9	0.97	0.25	0.01	0.41	1.00	1.00	1.00	1.00
Mean	**0.97**	**0.35**	**0.03**		**0.92**	**0.59**	**0.81**	

Table B.2: Xenia-AFIT T = 3min.

	R (Efficiency)				**F (Accuracy)**			
S’	**H**	**H’**	**H”**	**H”’**	**H**	**H’**	**H”**	**H”’**
1	0.96	0.39	0.00	N/A	1.00	0.29	0.37	N/A
2	0.97	0.38	0.05	N/A	0.82	0.22	0.49	N/A
3	0.97	0.36	0.00	N/A	1.00	0.73	1.00	N/A
4	0.97	0.20	0.00	N/A	1.00	0.94	1.00	N/A
5	0.97	0.37	0.10	N/A	1.00	0.92	1.00	N/A
6	0.97	0.33	0.01	N/A	0.84	0.68	0.84	N/A
7	0.96	0.60	0.10	N/A	0.69	0.30	0.63	N/A
8	0.98	0.26	0.01	N/A	0.96	0.23	0.96	N/A
9	0.97	0.25	0.01	0.41	1.00	1.00	1.00	1.00
Mean	**0.97**	**0.35**	**0.03**		**0.92**	**0.59**	**0.81**	

Table B.3: Xenia-AFIT T = 1min.

	R (Efficiency)				**F (Accuracy)**			
S’	**H**	**H’**	**H”**	**H”’**	**H**	**H’**	**H”**	**H”’**
1	0.96	0.07	0.00	N/A	1.00	1.00	1.00	N/A
2	0.96	0.18	0.00	N/A	1.00	1.00	1.00	N/A
3	0.98	0.15	0.00	N/A	1.00	1.00	1.00	N/A
4	0.98	0.34	0.00	N/A	0.55	0.10	0.45	N/A
5	0.98	0.25	0.00	N/A	0.95	0.55	1.00	N/A
6	0.96	0.36	0.00	N/A	1.00	1.00	1.00	N/A
7	0.96	0.32	0.00	N/A	1.00	0.42	1.00	N/A
8	0.95	0.37	0.00	N/A	1.00	0.58	1.00	N/A
9	0.97	0.12	0.00	N/A	1.00	1.00	1.00	N/A
10	0.95	0.17	0.00	N/A	1.00	1.00	1.00	N/A
11	0.96	0.44	0.00	N/A	0.76	0.66	0.76	N/A
12	0.97	0.31	0.00	N/A	1.00	1.00	1.00	N/A
13	0.98	0.10	0.00	N/A	1.00	1.00	1.00	N/A
14	0.98	0.25	0.00	N/A	0.83	0.83	0.83	N/A
15	0.97	0.12	0.00	N/A	0.67	0.63	0.67	N/A
16	0.95	0.20	0.00	N/A	1.00	1.00	1.00	N/A
17	0.94	0.26	0.00	N/A	0.67	0.33	0.62	N/A
18	0.99	0.18	0.00	N/A	0.32	0.12	0.32	N/A
19	0.99	0.14	0.00	N/A	0.36	0.25	0.36	N/A
20	0.99	0.33	0.00	0.00	0.71	0.36	0.75	0.00
Mean	**0.97**	**0.23**	**0.00**		**0.84**	**0.69**	**0.84**	

Table B.4: TechEdge-Xenia T = 5 min.

	R (Efficiency)				**F (Accuracy)**			
S’	**H**	**H’**	**H”**	**H”’**	**H**	**H’**	**H”**	**H”’**
1	0.96	0.52	0.00	N/A	0.54	0.00	0.00	N/A
2	0.94	0.33	0.00	N/A	0.96	0.89	0.96	N/A
3	0.98	0.37	0.05	N/A	0.61	0.44	0.61	N/A
4	0.99	0.27	0.07	N/A	0.34	0.32	0.34	N/A
5	0.96	0.40	0.00	N/A	0.74	0.14	0.54	N/A
6	0.98	0.36	0.00	N/A	0.61	0.61	0.61	N/A
7	1.00	0.16	0.00	0.74	1.00	1.00	1.00	Failed
Mean	**0.97**	**0.35**	**0.02**		**0.69**	**0.48**	**0.58**	

Table B.5: TechEdge-Xenia T = 3 min.

	R (Efficiency)				**F (Accuracy)**			
S’	**H**	**H’**	**H”**	**H”’**	**H**	**H’**	**H”**	**H”’**
1	0.97	0.31	0.00	N/A	0.74	0.26	0.44	N/A
2	0.98	0.51	0.39	N/A	0.80	0.71	0.80	N/A
3	0.96	0.42	0.06	N/A	1.00	0.77	1.00	N/A
4	0.99	0.28	0.09	N/A	0.37	0.19	0.37	N/A
5	0.99	0.33	0.04	N/A	0.40	0.16	0.37	N/A
6	1.00	0.38	0.03	N/A	0.28	0.21	0.28	N/A
7	0.96	0.29	0.00	N/A	0.42	0.01	0.16	N/A
8	0.97	0.38	0.00	N/A	1.00	0.53	0.84	N/A
9	0.98	0.34	0.00	N/A	0.75	0.69	0.69	N/A
10	1.00	0.44	0.06	0.10	1.00	1.00	1.00	Failed
Mean	**0.98**	**0.36**	**0.07**		**0.68**	**0.45**	**0.60**	

Table B.6: TechEdge-Xenia T = 1 min.

	R (Efficiency)				**F (Accuracy)**			
S’	**H**	**H’**	**H”**	**H”’**	**H**	**H’**	**H”**	**H”’**
1	0.99	0.00	0.00	N/A	1.00	1.00	1.00	N/A
2	0.97	0.83	0.00	N/A	0.46	0.31	0.46	N/A
3	0.97	0.21	0.00	N/A	1.00	0.24	0.67	N/A
4	0.95	0.22	0.12	N/A	0.67	0.67	0.67	N/A
5	0.95	0.49	0.00	N/A	0.56	0.36	0.56	N/A
6	0.97	0.30	0.25	N/A	0.86	0.14	0.86	N/A
7	0.99	0.57	0.00	N/A	0.71	0.54	0.71	N/A
8	0.99	0.38	0.00	N/A	0.32	0.05	0.05	N/A
9	1.00	0.29	0.00	N/A	0.40	0.40	0.40	N/A
10	1.00	0.27	0.12	N/A	0.39	0.09	0.30	N/A
11	0.99	0.48	0.05	N/A	0.36	0.36	0.36	N/A
12	0.99	0.29	0.69	N/A	0.25	0.25	0.25	N/A
13	1.00	0.85	0.01	N/A	0.32	0.24	0.32	N/A
14	0.96	0.21	0.00	N/A	0.35	0.04	0.15	N/A
15	0.97	0.52	0.00	N/A	0.71	0.71	0.71	N/A
16	0.97	0.10	0.00	N/A	0.88	0.08	0.42	N/A
17	0.98	0.43	0.00	N/A	0.78	0.57	0.78	N/A
18	0.99	0.22	0.00	N/A	0.68	0.68	0.68	N/A
19	0.99	0.24	0.00	N/A	1.00	0.74	1.00	N/A
20	1.00	0.00	0.00	0.32	0.00	0.00	0.00	Failed
Mean	**0.98**	**0.36**	**0.07**		**0.58**	**0.37**	**0.52**	

Table B.7: TechEdge-Vandalia T = 5 min.

	R (Efficiency)				**F (Accuracy)**			
S’	**H**	**H’**	**H”**	**H”’**	**H**	**H’**	**H”**	**H”’**
1	1.00	0.16	0.00	N/A	0.82	0.00	0.82	
2	0.98	0.41	0.01	N/A	0.80	0.74	0.80	
3	0.99	0.18	0.00	N/A	0.73	0.41	0.70	
4	0.96	0.37	0.07	N/A	1.00	0.40	0.76	
5	0.95	0.31	0.00	0.68	1.00	1.00	1.00	1.00
Mean	**0.97**	**0.29**	**0.02**		**0.87**	**0.51**	**0.81**	

Table B.8: TechEdge-Vandalia T = 3 min.

	R (Efficiency)				**F (Accuracy)**			
S’	**H**	**H’**	**H”**	**H”’**	**H**	**H’**	**H”**	**H”’**
1	0.98	0.35	0.00	N/A	1.00	1.00	1.00	
2	0.99	0.26	0.00	N/A	1.00	0.96	1.00	
3	0.99	0.36	0.00	N/A	0.27	0.27	0.27	
4	0.97	0.54	0.00	N/A	0.33	0.33	0.33	
5	0.98	0.01	0.00	N/A	1.00	0.84	1.00	
6	0.95	0.54	0.00	N/A	0.91	0.06	0.59	
7	0.95	1.00	0.00	0.58	1.00	1.00	1.00	1.00
Mean	**0.98**	**0.44**	**0.00**		**0.79**	**0.64**	**0.74**	

Table B.9: TechEdge-Vandalia T = 1 min.

	R (Efficiency)				**F (Accuracy)**			
S’	**H**	**H’**	**H”**	**H”’**	**H**	**H’**	**H”**	**H”’**
1	1.00	0.00	0.00	N/A	1.00	1.00	1.00	
2	0.98	0.89	0.66	N/A	0.57	0.03	0.03	
3	0.97	0.35	0.07	N/A	1.00	0.79	1.00	
4	0.98	0.35	0.00	N/A	1.00	0.92	1.00	
5	0.97	0.63	0.14	N/A	0.09	0.04	0.09	
6	0.99	0.36	0.00	N/A	0.95	0.62	0.95	
7	0.96	0.15	0.00	N/A	0.89	0.44	0.89	
8	0.98	0.48	0.00	N/A	1.00	0.93	1.00	
9	0.98	0.59	0.03	N/A	1.00	0.55	1.00	
10	0.97	0.26	0.00	N/A	1.00	0.31	1.00	
11	0.98	0.44	0.05	N/A	1.00	0.35	1.00	
12	0.94	0.31	0.00	N/A	1.00	0.71	1.00	
13	0.97	0.46	0.17	0.00	0.95	0.95	0.95	0.95
Mean	**0.97**	**0.40**	**0.09**		**0.88**	**0.59**	**0.84**	

Table B.10: Huber Heights - TechEdge T = 5 min.

	R (Efficiency)				**F (Accuracy)**			
S’	**H**	**H’**	**H”**	**H”’**	**H**	**H’**	**H”**	**H”’**
1	0.97	0.59	0.07	N/A	0.86	0.86	0.86	N/A
2	0.97	0.55	0.07	N/A	1.00	0.37	0.83	N/A
3	0.96	0.41	0.00	0.60	0.84	0.00	0.55	1.00
Mean	**0.97**	**0.52**	**0.05**		**0.90**	**0.41**	**0.74**	

Table B.11: Huber Heights - TechEdge T = 3 min.

	R (Efficiency)				**F (Accuracy)**			
S’	**H**	**H’**	**H”**	**H”’**	**H**	**H’**	**H”**	**H”’**
1	0.98	0.57	0.23	N/A	0.67	0.67	0.67	N/A
2	0.97	0.50	0.19	N/A	0.73	0.59	0.73	N/A
3	0.96	0.47	0.02	N/A	1.00	0.39	1.00	N/A
4	0.97	0.40	0.07	N/A	0.67	0.03	0.33	N/A
5	0.99	0.16	0.00	0.66	1.00	1.00	1.00	0.84
Mean	**0.97**	**0.42**	**0.10**		**0.81**	**0.54**	**0.75**	

Table B.12: Huber Heights - TechEdge T = 1 min.

	R (Efficiency)				**F (Accuracy)**			
S’	**H**	**H’**	**H”**	**H”’**	**H**	**H’**	**H”**	**H”’**
1	1.00	0.00	0.00	N/A	0.87	0.61	0.83	N/A
2	0.99	0.39	0.11	N/A	0.41	0.41	0.41	N/A
3	0.98	0.39	0.01	N/A	1.00	1.00	1.00	N/A
4	0.98	0.41	0.14	N/A	0.61	0.50	0.61	N/A
5	0.95	0.58	0.13	N/A	1.00	0.65	1.00	N/A
6	0.95	0.53	0.12	N/A	1.00	1.00	1.00	N/A
7	0.95	0.16	0.00	N/A	0.95	0.05	0.27	N/A
8	0.97	0.34	0.00	N/A	1.00	0.89	0.00	N/A
9	0.97	0.19	0.00	N/A	0.90	0.10	0.80	N/A
10	0.98	0.56	0.20	N/A	0.65	0.50	0.65	N/A
11	0.94	0.53	0.13	0.95	0.29	0.29	0.29	0.78
Mean	**0.97**	**0.37**	**0.08**		**0.79**	**0.54**	**0.62**	

Table B.13: Jamestown-Xenia T = 5 min.

	R (Efficiency)				**F (Accuracy)**			
	R (Efficiency)				**F (Accuracy)**			
S’	**H**	**H’**	**H”**	**H”’**	**H**	**H’**	**H”**	**H”’**
1	0.96	0.49	0.00	N/A	0.93	0.58	0.93	N/A
2	0.97	0.24	0.00	N/A	0.86	0.37	0.86	N/A
3	0.95	0.46	0.05	N/A	0.86	0.86	0.86	N/A
4	0.96	0.49	0.03	N/A	1.00	1.00	1.00	N/A
Mean	**0.96**	**0.42**	**0.02**		**0.91**	**0.70**	**0.91**	

Table B.14: Jamestown-Xenia T = 3 min.

	R (Efficiency)				**F (Accuracy)**			
S’	**H**	**H’**	**H”**	**H”’**	**H**	**H’**	**H”**	**H”’**
1	0.98	0.54	0.18	N/A	0.95	0.43	0.68	N/A
2	0.97	0.33	0.00	N/A	0.85	0.34	0.85	N/A
3	0.98	0.31	0.00	N/A	0.81	0.49	0.81	N/A
4	0.97	0.34	0.00	N/A	1.00	0.84	1.00	N/A
5	0.98	0.46	0.22	N/A	1.00	1.00	1.00	N/A
Mean	**0.98**	**0.39**	**0.08**		**0.92**	**0.62**	**0.87**	

Table B.15: Jamestown-Xenia T = 1 min.

	R (Efficiency)				**F (Accuracy)**			
S’	**H**	**H’**	**H”**	**H”’**	**H**	**H’**	**H”**	**H”’**
1	0.98	0.38	0.04	N/A	1.00	0.78	1.00	N/A
2	0.97	0.69	0.24	N/A	0.63	0.04	0.42	N/A
3	0.96	0.34	0.00	N/A	0.88	0.79	0.88	N/A
4	0.97	0.33	0.00	N/A	0.80	0.65	0.80	N/A
5	0.97	0.51	0.22	N/A	0.88	0.04	0.40	N/A
6	0.96	0.33	0.00	N/A	0.75	0.33	0.67	N/A
7	0.97	0.20	0.00	N/A	0.84	0.76	0.84	N/A
8	0.96	0.25	0.00	N/A	0.91	0.45	0.91	N/A
9	0.96	0.15	0.00	N/A	0.86	0.81	0.86	N/A
10	0.98	0.37	0.04	N/A	0.69	0.69	0.69	N/A
11	0.97	0.39	0.08	N/A	1.00	1.00	1.00	N/A
12	0.96	0.11	0.00	N/A	0.86	0.62	0.86	N/A
Mean	**0.97**	**0.34**	**0.05**		**0.84**	**0.58**	**0.78**	

Table B.16: Home - Animal Shelter T = 5 min.

	R (Efficiency)				**F (Accuracy)**			
S’	**H**	**H’**	**H”**	**H”’**	**H**	**H’**	**H”**	**H”’**
1	0.97	0.62	0.11	N/A	1.00	0.69	1.00	N/A
2	0.96	0.38	0.00	N/A	0.24	-1.13	0.24	N/A
3	0.98	0.32	0.00	N/A	0.81	0.81	0.81	N/A
4	0.97	0.31	0.00	N/A	0.51	0.43	0.51	N/A
5	0.98	0.34	0.03	N/A	0.83	0.60	0.83	N/A
6	0.98	0.36	0.02	Failed	1.00	1.00	1.00	Failed
Mean	**0.97**	**0.39**	**0.03**		**0.73**	**0.40**	**0.73**	

Table B.17: Home - Animal Shelter T = 3 min.

	R (Efficiency)				**F (Accuracy)**			
S’	**H**	**H’**	**H”**	**H”’**	**H**	**H’**	**H”**	**H”’**
1	0.98	0.44	0.02	N/A	1.00	0.97	1.00	N/A
2	0.96	0.39	0.09	N/A	0.74	0.01	0.31	N/A
3	0.99	0.48	0.10	N/A	0.60	0.46	0.60	N/A
4	0.97	0.45	0.03	N/A	0.64	0.21	0.55	N/A
5	0.98	0.21	0.00	N/A	0.59	0.48	0.59	N/A
6	0.99	0.31	0.00	N/A	0.65	0.65	0.65	N/A
7	0.99	0.33	0.02	Failed	0.57	0.31	0.57	Failed
Mean	**0.98**	**0.37**	**0.04**		**0.68**	**0.44**	**0.61**	

Table B.18: Home - Animal Shelter T = 1 min.

	R (Efficiency)				**F (Accuracy)**			
S’	**H**	**H’**	**H”**	**H”’**	**H**	**H’**	**H”**	**H”’**
1	0.98	0.38	0.00	N/A	1.00	0.76	1.00	N/A
2	0.94	0.26	0.00	N/A	0.88	0.46	0.88	N/A
3	0.95	0.61	0.14	N/A	0.79	0.00	0.29	N/A
4	0.97	0.40	0.10	N/A	0.89	0.61	0.89	N/A
5	0.98	0.53	0.09	N/A	0.46	0.46	0.46	N/A
6	0.98	0.20	0.00	N/A	0.47	0.47	0.47	N/A
7	0.97	0.10	0.00	N/A	1.00	1.00	1.00	N/A
8	0.98	0.24	0.00	N/A	0.59	0.21	0.59	N/A
9	0.97	0.37	0.00	N/A	0.58	0.58	0.58	N/A
10	0.98	0.26	0.00	N/A	0.67	0.44	0.67	N/A
11	0.99	0.59	0.24	N/A	0.21	0.21	0.21	N/A
12	0.98	0.59	0.11	N/A	0.67	0.04	0.44	N/A
13	0.97	0.46	0.00	N/A	0.56	0.30	0.44	N/A
14	0.98	0.23	0.00	Failed	0.27	0.27	0.27	Failed
Mean	**0.97**	**0.37**	**0.05**		**0.64**	**0.41**	**0.58**	

Table B.19: Fairborn - Jamestown T = 5 min.

	R (Efficiency)				**F (Accuracy)**			
S’	**H**	**H’**	**H”**	**H”’**	**H**	**H’**	**H”**	**H”’**
1	0.97	0.62	0.11	N/A	1.00	0.24	1.00	N/A
2	0.97	0.23	0.00	N/A	0.79	0.66	0.79	N/A
3	0.97	0.34	0.00	N/A	0.64	0.47	0.64	N/A
4	0.96	0.31	0.01	N/A	0.90	0.48	0.80	N/A
5	0.97	0.52	0.15	N/A	0.42	0.16	0.35	N/A
6	0.97	0.52	0.06	N/A	0.70	0.28	0.56	N/A
7	0.99	0.33	0.10	N/A	0.77	0.03	0.06	N/A
8	0.93	0.50	0.02	0.22	1.00	1.00	1.00	1.00
Mean	**0.97**	**0.40**	**0.05**		**0.78**	**0.42**	**0.65**	

Table B.20: Fairborn - Jamestown T = 3 min.

	R (Efficiency)				**F (Accuracy)**			
S’	**H**	**H’**	**H”**	**H”’**	**H**	**H’**	**H”**	**H”’**
1	0.98	0.44	0.02	N/A	1.00	0.51	1.00	N/A
2	0.97	0.31	0.01	N/A	0.95	0.50	0.95	N/A
3	0.97	0.29	0.00	N/A	0.49	0.39	0.49	N/A
4	0.97	0.23	0.00	N/A	0.82	0.51	0.81	N/A
5	0.99	0.37	0.07	N/A	0.64	0.14	0.56	N/A
6	0.96	0.44	0.10	N/A	0.82	0.25	0.56	N/A
7	0.99	0.77	0.39	N/A	0.54	0.43	0.54	N/A
8	0.98	0.54	0.11	N/A	0.37	0.00	0.19	N/A
9	0.99	0.32	0.02	N/A	0.96	0.47	0.84	N/A
10	0.98	0.27	0.02	N/A	0.74	0.26	0.54	N/A
11	0.99	0.48	0.05	N/A	0.74	0.31	0.66	N/A
12	0.97	0.45	0.04	0.10	1.00	1.00	1.00	1.00
Mean	**0.98**	**0.41**	**0.08**		**0.76**	**0.40**	**0.68**	

Table B.21: Fairborn - Jamestown T = 1 min.

	R (Efficiency)				**F (Accuracy)**			
S’	**H**	**H’**	**H”**	**H”’**	**H**	**H’**	**H”**	**H”’**
1	0.98	0.38	0.00	N/A	1.00	1.00	1.00	
2	0.96	0.42	0.05	N/A	1.00	0.05	0.57	
3	0.97	0.13	0.00	N/A	0.31	0.23	0.31	
4	0.97	0.12	0.00	N/A	1.00	1.00	1.00	
5	0.98	0.11	0.00	N/A	0.43	0.39	0.43	
6	0.97	0.13	0.00	N/A	0.52	0.44	0.52	
7	0.98	0.06	0.00	N/A	1.00	1.00	1.00	
8	0.99	0.50	0.03	N/A	0.43	0.07	0.43	
9	0.98	0.34	0.00	N/A	0.96	0.96	0.96	
10	0.95	0.51	0.12	N/A	3.00	0.25	1.50	
11	0.96	0.22	0.00	N/A	0.86	0.27	0.86	
12	0.97	0.93	0.79	N/A	0.82	0.32	0.64	
13	0.99	0.38	0.01	N/A	0.33	0.10	0.29	
14	0.98	0.23	0.00	N/A	0.74	0.26	0.74	
15	0.99	0.35	0.00	N/A	0.38	0.13	0.38	
16	0.99	0.30	0.00	N/A	0.50	0.46	0.50	
17	0.98	0.20	0.00	N/A	0.77	0.62	0.77	
18	0.98	0.64	0.23	N/A	0.78	0.04	0.35	
19	0.98	0.22	0.00	N/A	0.73	0.64	0.68	
20	0.98	0.22	0.00	N/A	0.73	0.38	0.73	
21	0.98	0.62	0.22	N/A	0.70	0.30	0.55	
22	0.98	0.00	0.00	N/A	0.70	0.70	0.70	
23	0.95	0.18	0.00	0.00	0.54	0.54	0.54	0.54
Mean	**0.98**	**0.32**	**0.06**		**0.79**	**0.44**	**0.67**	

Table B.22: AFIT - TechEdge = 5 min.

	R (Efficiency)				**F (Accuracy)**			
S’	**H**	**H’**	**H”**	**H”’**	**H**	**H’**	**H”**	**H”’**
1	0.98	0.19	0.00	N/A	0.91	0.91	0.91	N/A
Mean	**0.98**	**0.19**	**0.00**		**0.91**	**0.91**	**0.91**	

Table B.23: AFIT - TechEdge = 3 min.

	R (Efficiency)				**F (Accuracy)**			
S’	**H**	**H’**	**H”**	**H”’**	**H**	**H’**	**H”**	**H”’**
1	0.97	0.36	0.00	N/A	1.00	1.00	1.00	N/A
2	0.99	0.19	0.00	N/A	1.00	1.00	1.00	N/A
Mean	**0.98**	**0.27**	**0.00**		**1.00**	**1.00**	**1.00**	

Table B.24: AFIT - TechEdge = 1 min.

	R (Efficiency)				**F (Accuracy)**			
S’	**H**	**H’**	**H”**	**H”’**	**H**	**H’**	**H”**	**H”’**
1	0.92	0.22	0.00	N/A	1.00	1.00	1.00	N/A
2	0.95	0.12	0.00	N/A	1.00	1.00	1.00	N/A
3	0.99	0.55	0.00	N/A	0.67	0.19	0.67	N/A
4	0.99	0.05	0.00	N/A	0.58	0.46	0.58	N/A
5	0.98	0.48	0.03	N/A	0.23	0.18	0.23	N/A
6	0.98	0.06	0.00	N/A	1.00	1.00	1.00	N/A
Mean	**0.97**	**0.25**	**0.00**		**0.75**	**0.64**	**0.75**	

Appendix C. Business Model and Applications

Figure C.1: Layered sensing combines imagery from many sensing platforms.

C.0.1 A Proposed Business Model . It is reasonable to propose a general business model within which these ideas may operate. This undertaking need not begin from a zero baseline because much work has been invested recently in this regard. The *Air Force Research Laboratory*, in particular, has occupied the vanguard of this effort by developing the *Layered Sensing* construct. In a position paper meant to define *Layered Sensing* [7], the Layered Sensing Leadership Group (LSLG) define this construct as follows:

> Layered Sensing provides military and homeland security decision makers at all levels with timely, actionable, trusted, and relevant information necessary for situational awareness to ensure their decisions achieve the desired military/humanitarian effects. Layered Sensing is characterized

> by the appropriate sensor or combination of sensors/platforms, infrastructure and exploitation capabilities to generate that situation awareness and directly support delivery of "tailored effects".

The essential gist here is that reliable, accurate, and timely intelligence products must emerge from a single or multiple sources and be delivered to a "decision maker" for the purpose of prosecuting "tailored effects". It is prescient that their definition for "decision maker" is allowed to occupy a large spectrum of possibilities, which they define generally as "all blue forces". In other words, possible customers could include the JFACC in the AOC down to the Marine lance corporal on the streets of Fallujah. Figure C.1 illustrates that these intelligence products will be a laminate of various substrates coming from different platforms and sources. However, Figure C.1 does not (nor could it) represent all of the aerial sources of information. Additionally, it under-represents the *Cyber*, HUMINT, GEOINT, MASINT, IMINT, and OSINT contributions to the final "Synergy through Integration" [4] product. Examples that illustrate this construct are Figure C.2 and Figure C.3. The former combines aerial SAR and EO imagery taken from two different platforms for a greater semantics yield. The latter combines aerial EO and ground EO. The definition of *Layered Sensing* includes a list of guiding principles for these fused intelligence products. They must be:

- Persistent Coverage
- Wide Area Coverage
- Assured Global Access
- Engagement Quality Information
- Timeliness
- Trusted Sensing
- Information Triage
- Robust, Agile, and Adaptable

- Spectrum Dominance and Control
- Anticipatory Observations and Interactive Engagements
- Tailored Performance
- Affordable Open System Architecture

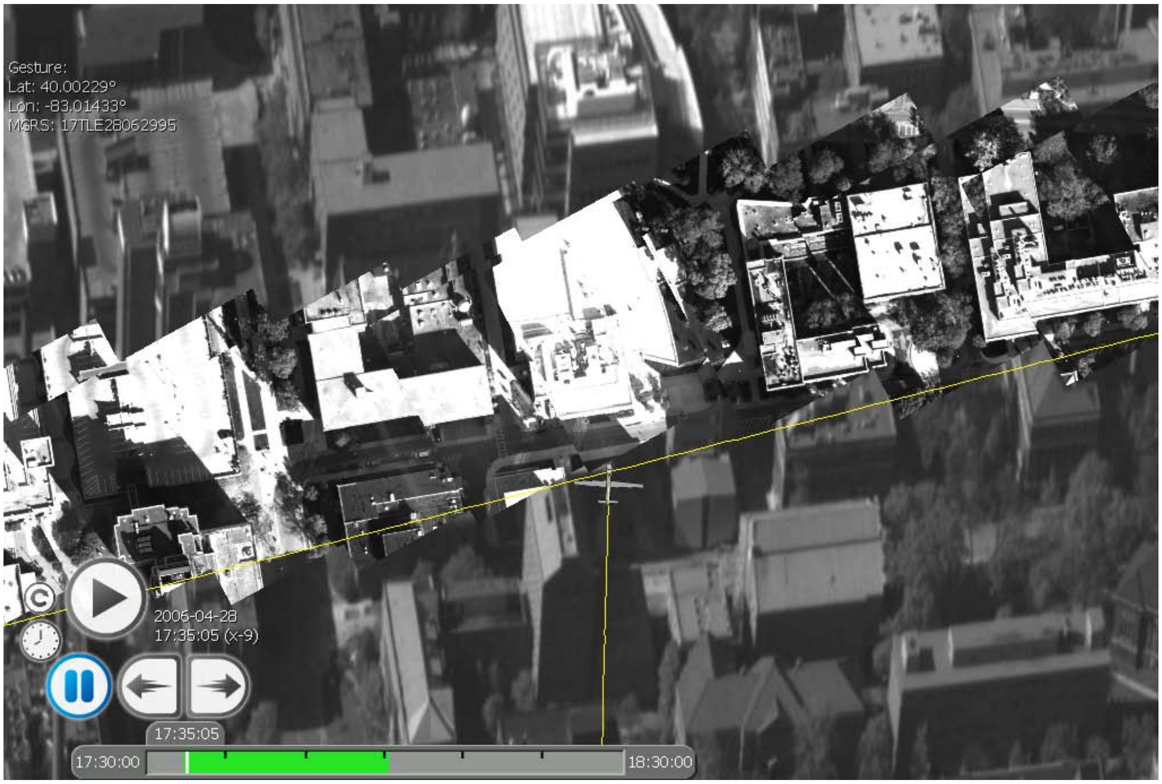


Figure C.2: Pictured here is an illustration of fused electro optical and synthetic aperture radar imagery.

This work will focus primarily on the *Anticipatory Observations and Interactive Engagements* topics from this list. However, the immediate discussion will explore the dynamics of the last topic *Affordable Open System Architecture*. It is here where our proposals will live or die based upon courses of action embarked upon by the Intelligence, Communications, Acquisitions, and other communities in the coming decade.

The LSLG posits that this architecture must be a "net-centric architecture" and of an "open standards" character. Also,

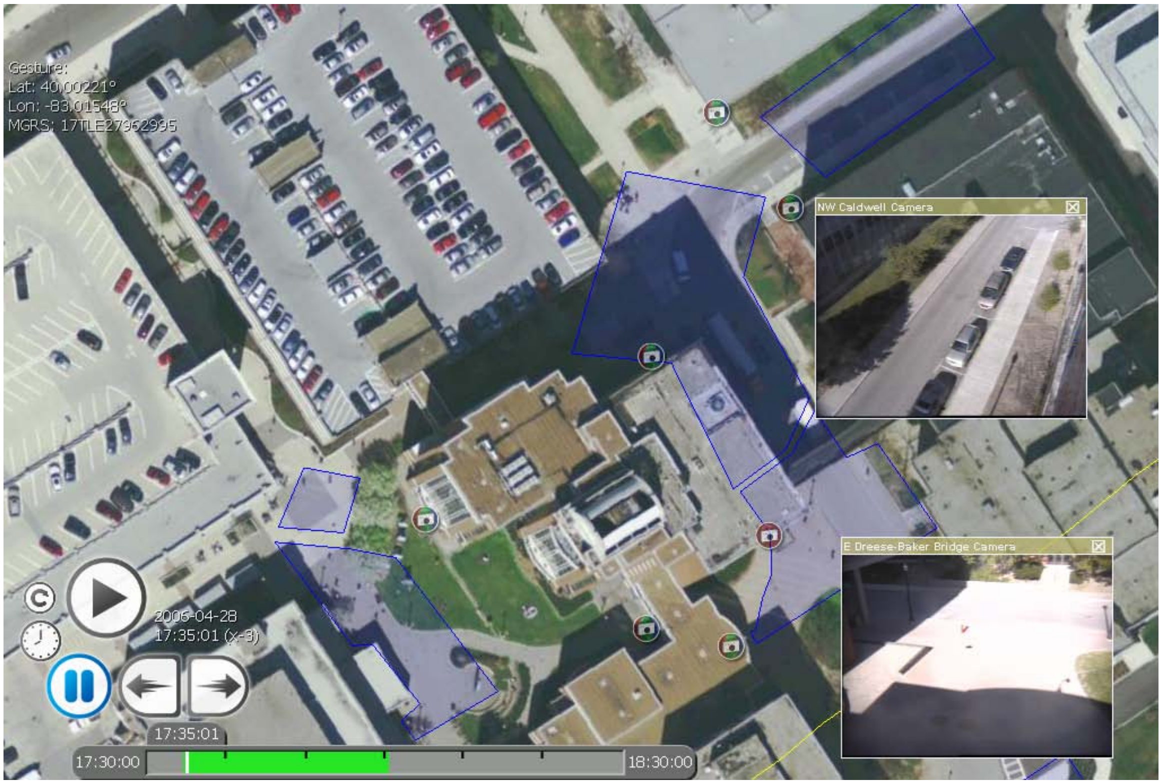


Figure C.3: Pictured here is an illustration of fused ground and aerial electro optical imagery.

> while *Affordable Open System Architecture* may use Commercial off the Shelf (COTS) components, it does not mandate nor require COTS.

While agreeing in principle with this, we offer the modest counter-example, per a report to Congress in 2000 [8], that the Chinese have provisioned a robust and formidable Integrated Air Defense (IAD) system based almost entirely upon COTS telecommunications components. This forces the admission that much can be accomplished (and at competitive costing) with such an approach. Additionally, the COTS approach forces the open standards regime required by *Layered Sensing*.

This requires us to peer a bit deeper into the realm of what is possible. Savy industry watchers will have marked a major paradigm shift in corporate IT lately in regards to enterprise data-processing. A serial of reports published in *The Economist* during October 2008 [9] illustrates the growing trend of *Cloud Computing* that has begun to manifest in the market. Because we have proposed major data collection, processing, presentation, and dissemination to a wide variety of customers in the battle-space, it behooves us to examine this trend so as to determine whether our model might exploit it.

The Economist cites Irving Wladawsky-Berger, a technologist at IBM, remarking that "in order for computing to reach a higher level, its (components) had to be commoditised." In other words, as the industry has matured, its essential functionalities in hardware and software needed the discreet packaging and interactive capability that cells in the biological world enjoy, and which they use to work together to form a system of systems. This system of systems, or *The Cloud*, behaves like the mainframes of computing antiquity, processing major computing workloads and then streaming the results as a service to client machines. The Software as a Service (SaaS) construct, which makes *Cloud Computing* possible, merely disaggregates the software and computational workload from the hardware. This computational workload becomes the feed-stock for virtualized computer images that perform the work, and then surrender the resources for reallocation. To the question "What would such a construct contribute to these proposals?" the answer is, simply, that a Services Oriented Architecture (SOA) would allow functionalities to be provided as services, rather than as monolithic software packages and thereby allow for a greater spectrum of customers to be serviced.

The Distributed Common Ground Station (DCGS) [10], used to disseminate imagery within the Intelligence community, has begun to flirt with this construct and might act as a test-bed for an enterprise *Cloud Computing* deployment. Actors in industry have already begun to fight in earnest for this market niche and Amazon, Google, and Microsoft have all provisioned some kind of cloud. Even more germane to our purpose, GeoProcessing functionalities being streamed to handheld devices is at the forefront of this trend [15]. That is because it would be infeasible for users to carry around the large GeoDatabases associated with GeoProcessing (a proposition that is often measurable in terabytes). This condition forced the SaaS business model upon this market niche almost immediately and offers us a clear choice.

C.1 Possible Applications in Tactical and Operational Battle-Space

The goals laid out in Section 1.1 called for the creation of a robust predictive capability which would augment existing ISR platforms such that a remote observer could benefit from a richer decision-set to choose from during a surveillance mission. The goals may be itemized more specifically as:

- Improve ISR management capabilities for greater asset efficiency
- Empower sensor-cueing on the ground
- Create target interdiction opportunities
- Create opportunities to perturb the battle-space to the advantage of friendly forces

All of these may result in Effects-Based Operations (EBO). Therefore, having spent a great deal of our time in the theoretical and software arenas, it will be rewarding to see this process and its results within the context of the tactical and operational levels of war.

C.1.1 ISR Platform Management. Recall that an airborne staring-array will have (if it is typical) a disk-shaped coverage area of the ground with a radius equal to roughly two kilometers. Figure C.4 illustrates this coverage area as **C.shp**, superimposed over the road network, a tracklet, and some other polygons. Note that if the controller of the airborne platform perceives that the surveillance target is about to exit the coverage area, then he/she will be confronted with the choice of altering the flight-plan in order to stay with that target or of maintaining the current position. The knowledge products consisting of alpha hull **h_(s').shp** and the set of all negative space **B.shp** may inform this decision in several way. First, the most recent alpha hull will predict (within time T) where that vehicle may go. The example in Figure C.4 implies that within T the coverage spot need not move. Additionally, close examination of the set of all negative space **B.shp** yields a valuable insight: it covers several of the major highways (in blue) within a reachable distance. By

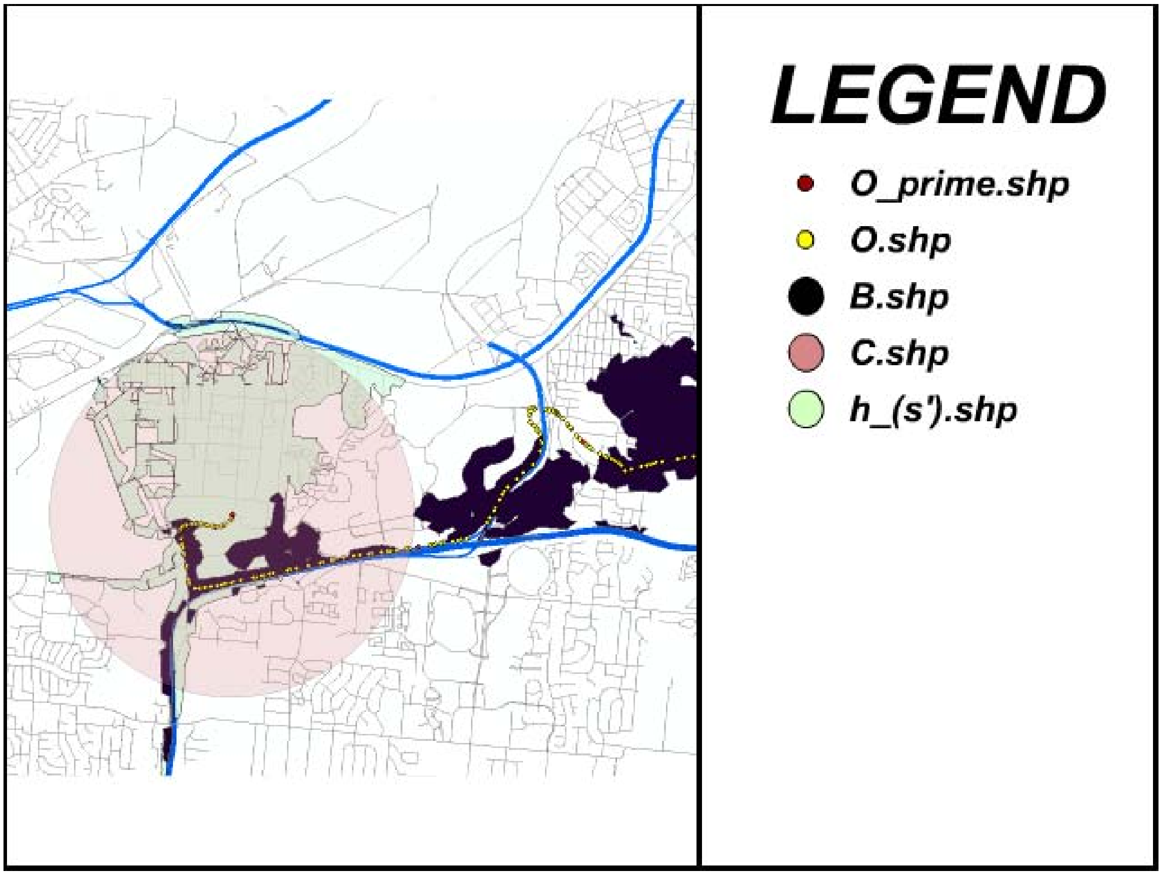


Figure C.4: It may be deduced from bounding polygon **h_(s').shp** and **B.shp** that the vehicle's possible destination is bounded by the highways (in blue) and that he will therefore remain within the local residential area.

Destination Directed (assumed to be in force unless the $NListener(O,\ B,\ s')$ event-handler informs otherwise), it may be concluded that the object of the surveilled vehicle will not be a highway on-ramp. This suggests that the coverage spot need not move because the vehicle will be navigating the small residential area bounded by the highways.

Another application involving the coverage spot **C.shp**, could be a situation where the spot must loiter over a given area despite a possible target exiting the bounds of observation. Such a case might result when there are more than one vehicles of interest. Normally, this would mean that the vehicle of interest would "fall off the radar" so to speak. Instead of allowing this, an alpha hull could be created exactly at the spot where the vehicle crossed the circumferential bounds of the spot as is

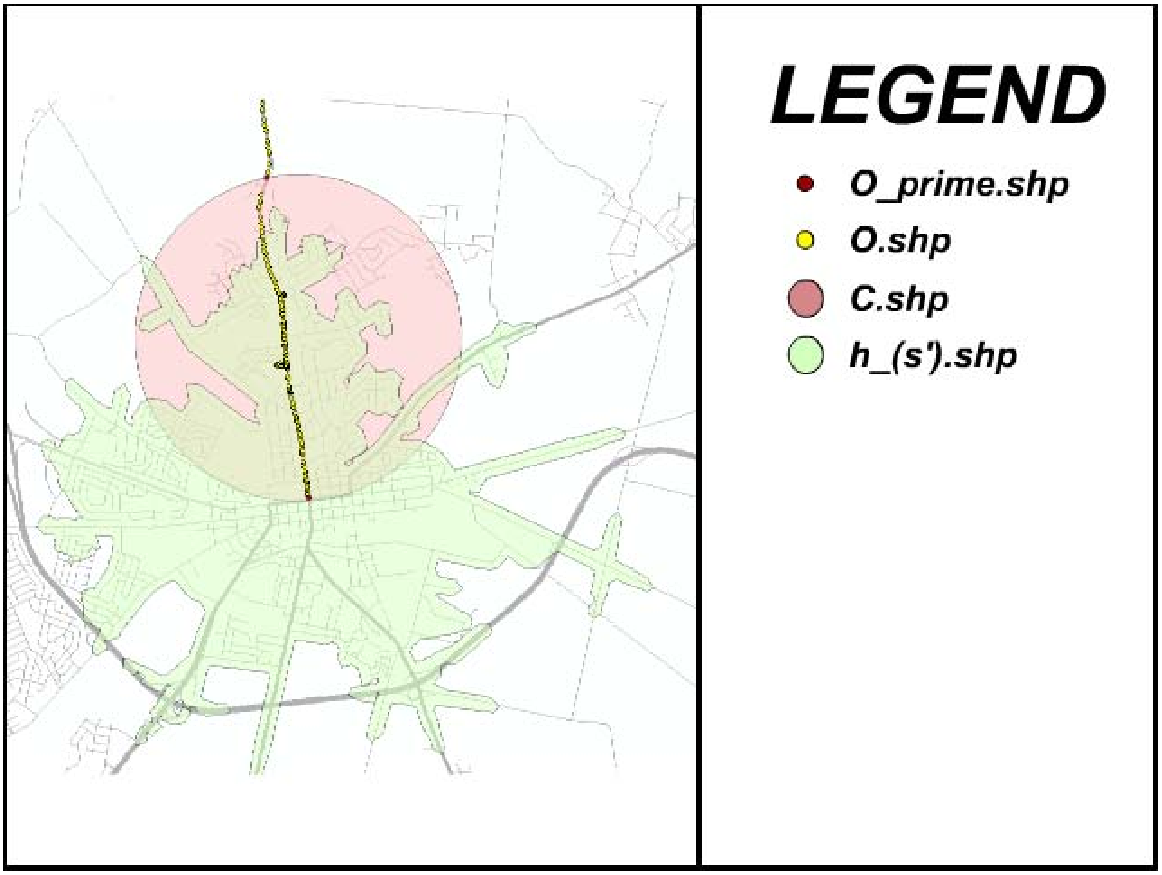


Figure C.5: If the vehicle appears to be exiting the bounds of the coverage space, represented by **C.shp**, and the coverage spot may not move, an alpha hull (**h_(s').shp**) may predict where the will be until T.

illustrated in Figure C.5. This would give observers until time T to determine a course of action that could include interdiction, or vectoring an additional ISR asset to the area.

C.1.2 Sensor Cueing. Per our discussion in Section C.0.1 on the layered sensing construct, it will always be desirable to add additional sensor capabilities to the fused data-environment. Assuming that there are surveillance cameras on the ground that are tightly-coupled to our process, we might initiate a cascade of events according to where the vehicle is perceived to be going. Figure C.6 illustrates a case in point. If a vehicle that is being tracked, and for which our process is generating bounding polygons **h_(s').shp** within T, then all blue-force ground sensors that fall

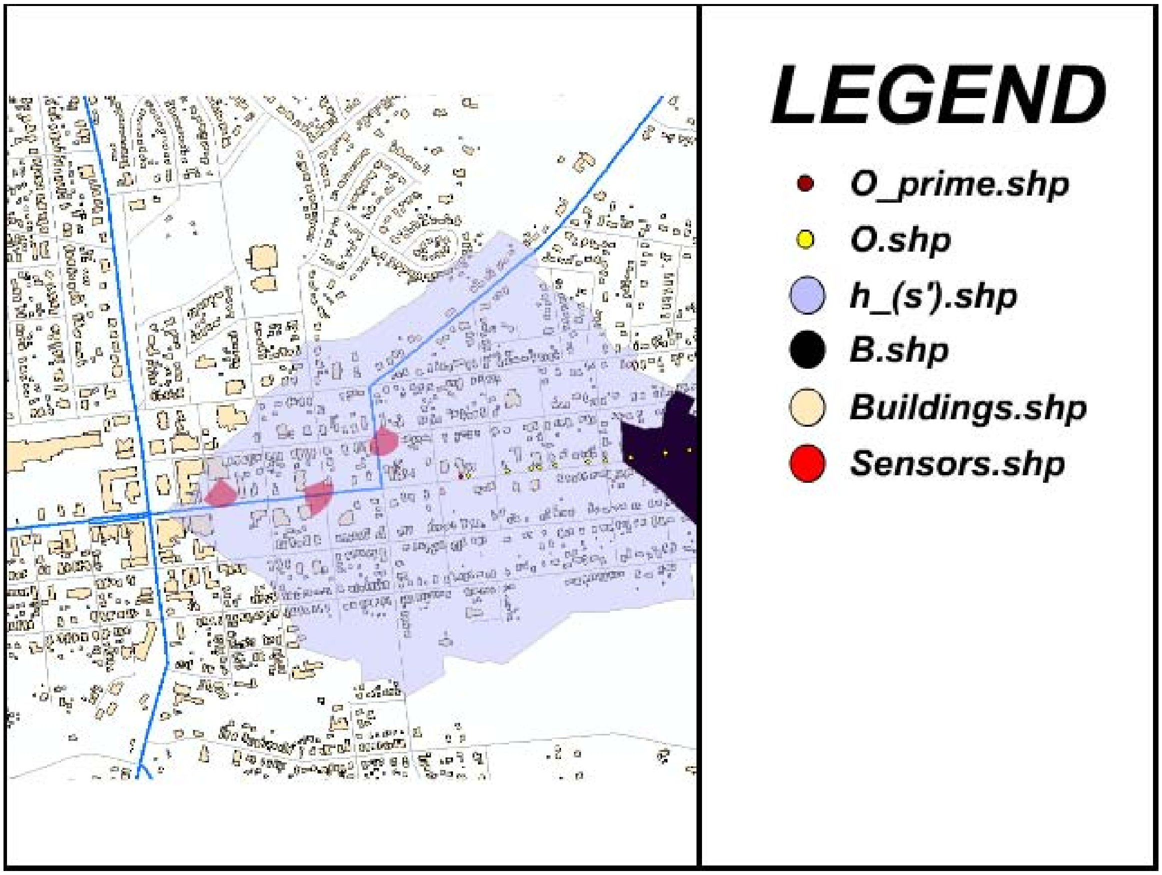


Figure C.6: In this scenario, the tracklet is about to enter an area with heavy surveillance camera coverage (**Sensors.shp**). Since these sensors are bounded by the current alpha hull **h_(s').shp**, they should be activated and closely attended.

within **h_(s').shp** should be switched on and attended. The red polygons in the figure represent surveillance cameras with average focal ranges and coverage over an area. If these sensors are manned by attentive intelligence operators, then more resolved imagery may be captured from the target in a timely enough manner to influence decisioning.

C.1.3 Interdiction. Central to our stated goal of Effects-Based Operations, will be a predictive awareness on the part of the observer. If it is known that the target is within the operational radii of in-theater offensive assets, then quick command decisioning may be undertaken. Figure C.7 illustrates a scenario where the target has entered a small hamlet and is within striking distance of two assets. The green disk

AC130.shp represents an *AC 130H Spectre* gunship, which travels 480 kilometers per hour. The operational radius was calculated using this peak velocity and a five minute travel time. It may be seen in this example that the gunship will be within range of the target before it is able to exceed its bounding polygon. Similarly, the blue disk **Stryker.shp** represents the operational radius of a *Stryker* armored combat vehicle with its light-mounted squad (traveling at its peak velocity of 72 kilometers per hour, and with a five minute travel time). Note that we employ a disk rather than an alpha hull; this is because the *Stryker* is an all-terrain vehicle. Again, it may be deduced that the *Stryker* may interdict the target before it can exceed its bounded area. This would prove even more useful in a non-ideal case. Imagine that the *AC 130H Spectre* gunship was ten minutes away and the *Stryker* was five minutes away. A joint commander would be able to quickly deduce that the *Stryker* squad enjoyed a greater possibility of successful interdiction, and therefore vector it, rather than the gunship, to the scene.

C.1.4 Perturbing the Battle-Space. The final, and most ambitious, of our stated goals was the possibility that the system user might exceed the bounds of observation and actually participate in the drama by adding precise perturbations to the battle-space. As an example, one might ask "what happens if we constrain traffic in certain areas?" The low-tech approach would be to set up roadblocks at strategic points in the road network. Nodes in our network (namely at intersections) may be "switched off" in such a way that is similar to our Tabu list. If arteries are closed at their intersections then they may, in fact, be added to the Tabu list with the effect of shaping the resulting alpha hull search. Figure C.8 illustrates the possibilities of such a scenario. The black area represented by the set of negative space polygons **B.shp** is created by placing barriers at key nodes in the road network $G(V, E)$. Wherever there is a barrier, the alpha hull (**h_(s').shp**) is blocked and does not grow beyond that point. Where, unimpeded, the alpha hull would have grown to cover all that is represented in black by **B.shp**, it may now only cover a fraction of the area. This

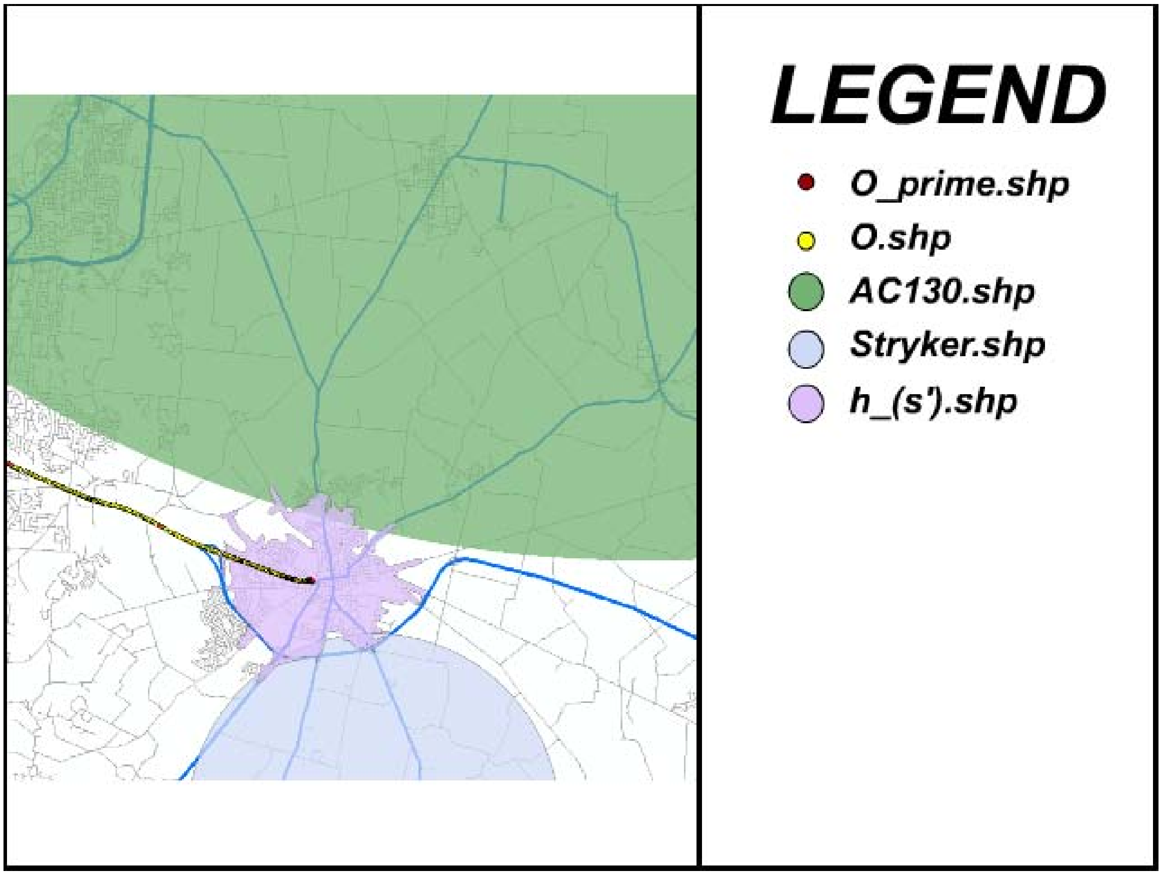


Figure C.7: In this scenario, the motion models of an *AC 130H Spectre* gunship (**AC130.shp**) and a *Stryker* armored combat vehicle (**Stryker.shp**) are seen in relation to a target's alpha hull.

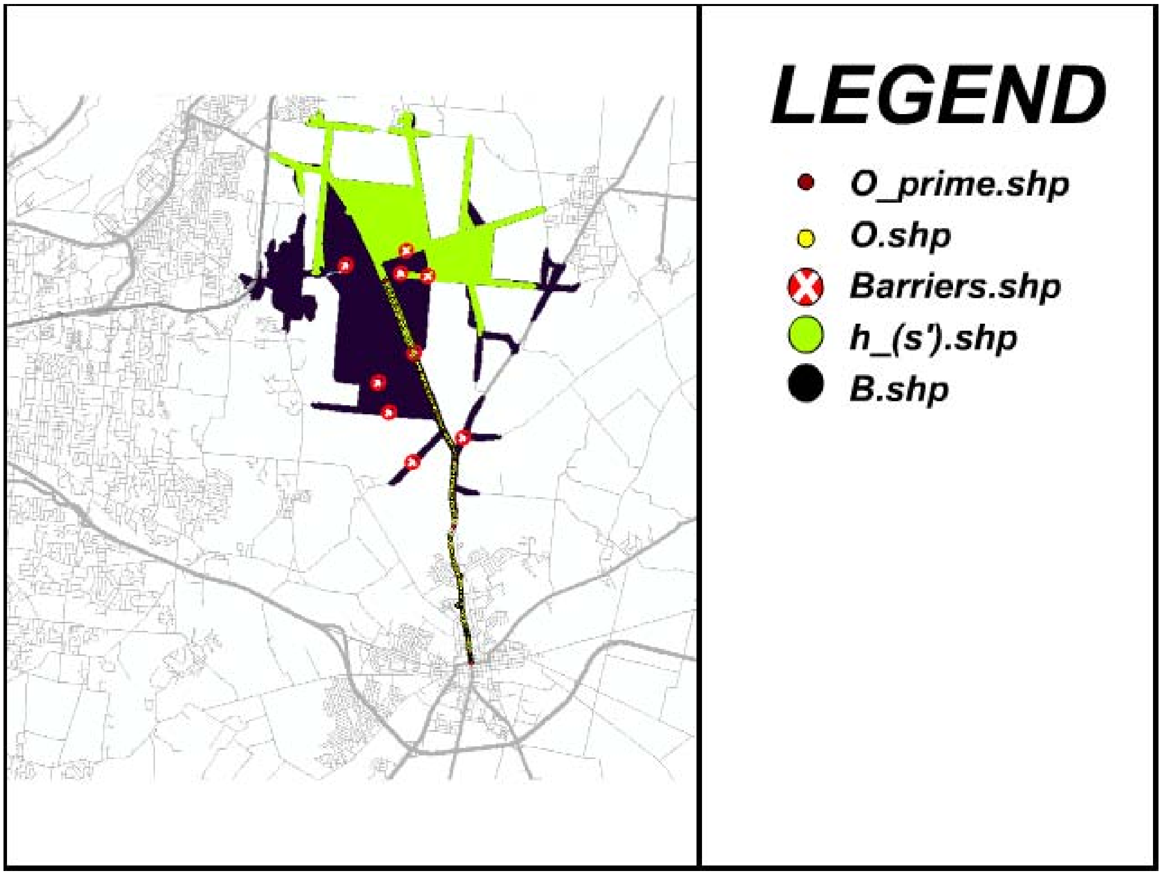


Figure C.8: Pictured here is an alpha hull that is not allowed to grow because impediments have been placed at key intersections in the road network. This allows the subject to be corralled into a smaller area.

has the practical effect of funneling the target within a smaller area, where he may be interdicted more easily (if, per the previous example, there are assets near) or observed more easily (if, per the first example, the ISR coverage spot may not move).

Bibliography

1. Woodward, Bob, *The War Within: A Secret White House History 2006-2008*, Simon and Schuster Adult Publishing Group, New York NY, 2008.

2. Gates, Robert M., *Remarks to Air War College*, US Air Force. Maxwell AFB, 21 Apr 2008.

3. Michael Hoffman, *Gates wants 240 Million more for ISR*, Air Force Times, 2008.

4. Air Force Doctrine Center, *Air Force Doctrine Document 2-9*, Joint Publication 2-01, Joint and National Intelligence Support to Military Operations, 2007.

5. Rogers, S. K., Sadowski, C., Bauer, K. W., Oxley, M. E., Kabrisky, M., Rogers, A. S. and Mott, S. D., *The life and death of ATR/Sensor Fusion and the hope for resurrection*, Proceedings of SPIE, Volume 6967, Mar 2008.

6. US Army, HQ TRADOC, *Field Manual 3-06 Urban Operations*, Department of the Army, 2007.

7. Rogers Steve, Mike Bryant, Brian Kent, and Michael Nowak, *LAYERED SENSING: Its Definition, Attributes, and Guiding Principles for AFRL Strategic Technology Development*, Sensors Directorate, Air Force Research Laboratory. 6.0th ed.

8. , United States Congress, *Annual Report on the Military Power of The People's Republic of China*, Report to Congress Pursuant to the FY2000 National Defense Authorization Act, 2000.

9. Siegele, Ludwig, *Special Report: Let it Rise*, The Economist, 23 Oct 2008.

10. Perera, Dave, *DCGS Integration Backbone forges joint ISR ground*, Defense Systems Magazine, 17 Nov 2008.

11. Howard, Torsten, *Automated Scene Segmentation in Hyperspectral Imagery*, Air Force Institute of Technology, Wright-Patterson AFB, March 2009.

12. Nunez, Abel, *Detection of Skin and its Chromophores in Hyperspectral Imagery*, Air Force Institute of Technology, Wright-Patterson AFB, March 2009.

13. Schmitt, Daniel, *Automated Knowledge Generation with Persistent Video Surveillance*, Air Force Institute of Technology, Wright-Patterson AFB, March 2009.

14. Donnell, Brian, *Using Shadows to Detect Targets in Synthetic Aperture Radar Imagery*, Air Force Institute of Technology, Wright-Patterson AFB, March 2009.

15. Shashi Shekhar, Ranga Raju Vatsavai, Xiaobin Ma, Jin Soung Yoo, *Navigation Systems: A Spatial Database Perspective*, Location-Based Services, 41-82, 2004.

16. Robert A. Piccerillo and David A. Brumbaugh, *Predictive Battlespace Awareness: Linking Intelligence, Surveillance and Reconnaissance Operations to Effects Based Operations*, 2004 Command and Control Research and Technology Symposium 2004.

17. Makris, D., and Ellis, T., *Finding Paths in Video Sequences*, British Machine Vision Conference (pp. 253–272). Manchester, UK: BMVC2001, 2001.

18. Makris, D., and Ellis, T., *Automatic Learning of an Activity-Based Semantic Scene Model*,London, UK: IEEE Computer Society, 2003.

19. National Research Council, *Highway Capacity Manual*, The National Academies, Washington D.C.,2000.

20. Diestel, Reinhard, *Graph Theory*, (3rd ed.), Heidelberg Germany, 2005.

21. Stuart Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach*,(2nd ed.), Upper Saddle River, NJ, 2003.

22. Lawrence R. Rabiner, *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*,Proceedings of the IEEE, vol. 77, pp. 257-286, 1989.

23. Maybeck, Peter S., *Stochastic Models, Estimation, and Control*,Academic PRess, New York, 1979.

24. Weeks, Joseph and Sanjeeb Nanda, *Contemporary Models for Path Prediction of Dynamic Entities*,Proceedings of the Spring 2006 Simulation Interoperability Workshop, vol. 2, pp. 479 487, 2006.

25. Lockheed Martin Corporation, *Dynamic Tactical Targetting (DTT) Target Motion Prediction Component*, Final Technical Report, Oct 2007.

26. Zadeh, L. A., *Fuzzy Sets*, Information and Control 8, 338-353, 1965.

27. Antony, Richard T. and Karakowski Joseph A., *Towards Greater Consciousness in Data Fusion Systems*, 2007 MSS National Symposium on Sensor and Data Fusion, 11-14 June 2007.

28. Horvitz, Eric and Apacible, Johnson and Sarin, Raman and Liao, Lin *Prediction, Expectation, and Surprise: Methods, Designs,and Study of a Deployed Traffic Forecasting Service*, Twenty-First Conference on Uncertainty in Artificial Intelligence, 2005.

29. Pierce, Scott, *Context Aided Tracking and Track Prediction in Aerial Video Surveillance*, Air Force Institute of Technology, Wright-Patterson AFB, March 2008.

30. Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, *Introduction to Algorithms*,(4th ed.) Cambridge, Massachusetts: MIT Press, 2003.

31. Shapshak, Mans, *Combining GIS and Semantic Technololgy to Create a Cultural Visualizer*, Directions Magazine, Tracasa Spain, 2008.

32. Schiller, Jochen, and Voisard, Agnes, *Location-Based Services*,Greensboro: Morgan Kaufmann, pp. 42 - 76, 2004.

33. Dunnigan, James, *USAF Building Disruptor Capability Right Now*, Strategy Page, 15 Jan, 2009.

34. Toole, Patrick A., *Sensor Models for Dynamic Tactical Targeting*, Toyon Research Corporation, 2006.

35. Chaput, Armand, *Conceptual Design of Unmanned Air Vehicle Systems*, The University of Texas Lecture Series, 2004.

36. Gries, David and Stojmenovic, Ivan, *A Note on Graham's Convex Hull Algorithm*,Information Processing Letters, vol. 25, pp. 323-327, 1987.

37. Marchette, David *Random Graphs for Statistical Pattern Recognition*, Wiley-Interscience, 2004.

38. Mitchell, Andy, *The ESRI Guide to GIS Analysis*, ESRI Press, Vol. 2, 2005.

39. Ebdon, David, *Statistics in Geometry: A practical Approach*, Wiley-Blackwell, Hoboken, NJ, Jan 1991.

40. Tomcisin, Steve, *Greene County Geographic Information Management System*, Raw data, 69 Greene Street, Xenia OH 45385.

41. Litvin, Joseph, *Montgomery GIS Records*, Raw data, 1850 Spaulding Road, Kettering, OH 45432.

42. Department of Commerce, Census Bureau, *U.S. Census Feature Class Codes (table)*, 2004.

43. Aniello, Pete, *Get Centroid*, Environmental Systems Research Institute, Available from: http://arcscripts.esri.com/details.asp?dbid=12781

44. Coram, Robert *Boyd: The fighter pilot who changed the art of war*, Little, Brown and Co., 2004.

45. Stahl, Leslie. "How Technology Won Sadr City Battle." 60 Minutes. CBS. New York, NY. 12 Oct. 2008.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)* 21–03–2009	**2. REPORT TYPE** Master's Thesis	**3. DATES COVERED** *(From — To)* July 2008 — Mar 2009

4. TITLE AND SUBTITLE

Exploitation of Geographic Information Systems
for
Vehicular Destination Prediction

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Muster, Richard T., Capt, USAF

5d. PROJECT NUMBER

08-240

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GCE/ENG/09-05

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

(MR. DEVERT WICKER)
Air Force Research Laboratory
BLDG 620, 2241 AVIONICS CIRCLE
WPAFB, OH 45433-7333
(937-904-9871 devert.wicker@wpafb.af.mil)

10. SPONSOR/MONITOR'S ACRONYM(S)

AFRL/RYAT

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approval for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This research proposes that vehicles in an urban setting subject to persistent Intelligence Surveillance and Reconnaissance will exhibit attributes that make it possible to predict their future position within a time-horizon. GeoSpatial Information Systems obtained from municipal, commercial, or hyperspectral sources may be used to model an urban grid and to make use of graph-theoretic search algorithms that can prune the future state-space of the vehicle's immediate environment. The results are representational polygons inside the GIS environment that constrict around the subject's likely destinations in the urban grid as the stream of observations continue to drive the process. These polygons predict where the subject may travel within the time-horizon, and also where they may not travel, given the assumption that they are moving along an optimal path between two points. It demonstrates a 81 % success rate for predictions carried out during experimentation in the Dayton Ohio area. It further demonstrates a 97 % improvement over predictions made with general motion models for moving vehicles.

15. SUBJECT TERMS

GIS, Dijkstra, voronoi, alpha hull, UAS

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Major Michael Mendenhall
U	U	U	UU	208	**19b. TELEPHONE NUMBER** *(include area code)* (937) 255-3636, ext 4614; email: michael.mendenhall afit.edu