University of Redlands

Crash Site Debris Recovery Mobile Application

A Major Individual Project submitted in partial satisfaction of the requirements
for the degree of Master of Science in Geographic Information Systems

by
Nicholas R. Janzen

Mark Kumler, Ph.D., Committee Chair

Pani Chakrapani, M.S.

Fon Allan Duke, M.B.A.

July 2012

Crash Site Debris Recovery Mobile Application

Copyright © 2012

by
Nicholas R. Janzen

The report of Nicholas R. Janzen is approved.

Fon Allan Duke, M.B.A.

Pani Chakrapani, M.S.

Mark Kumler, Ph.D., Committee Chair

July 2012

Acknowledgements

I would like to thank first and foremost my wife Mandi and my family. Without their help and support I would never have completed this project. I would also like to thank Steve Mesa for allowing me to work with his original idea, Fon Duke for promoting the project and setting up meetings to generate interest, Pani Chakrapani for his assistance in learning C#, Dr. Mark Kumler for his guidance in completing the project, Steve Paplanus for the programming help, and Cohort 20 for their support.

Abstract

Crash Site Debris Recovery Mobile Application

by
Nicholas R. Janzen

The Crash Site Debris Recovery Mobile Application is a prototype Windows Phone Application designed to assist crash investigators, first responders, and clean-up crews to locate aircraft crash debris in ground impact crash situations. The application, using a debris trajectory formula, the GPS location of the mobile device, a geoprocessing service, and user-defined variables of the affected aircraft and its crash characteristics, projects an estimated area of high debris concentration. In testing, the application has shown to be promising in furthering the development of mobile applications for use in airplane crash events.

Table of Contents

Chapter 1 – Introduction 1
1.1 Client 1
1.2 Problem Statement 1
1.3 Proposed Solution 1
1.3.1 Goals and Objectives 2
1.3.2 Scope 2
1.3.3 Methods 2
1.4 Audience 3
1.5 Overview of the Rest of this Report 3

Chapter 2 – Background and Literature Review 5
2.1 Search and Rescue 5
2.2 Debris Modeling 5
2.2.1 Debris Risk Assessment Model 5
2.2.2 Common Real Time Debris Footprint 6
2.2.3 Aircraft Accident Investigation Tool 6
2.2.4 Trajectory Analysis Program 7
2.3 Geographic Information Systems Integration 8
2.4 Summary 8

Chapter 3 – Systems Analysis and Design 9
3.1 Problem Statement 9
3.2 Requirements Analysis 9
3.2.1 Functional Requirements 9
3.2.2 Technical Requirements 10
3.2.3 Transitional Requirements 10
3.3 System Design 10
3.4 Project Plan 11
3.5 Summary 11

Chapter 4 – Database Design 13
4.1 Conceptual Data Model 13
4.2 Logical Data Model 13
4.3 Data Sources 15
4.4 Data Collection Methods 15
4.5 Data Scrubbing and Loading 15
4.6 Summary 15

Chapter 5 – Implementation 17
5.1 Debris Model Preparation 32
5.2 Developing a Geoprocessing Script Tool 17
5.3 Publishing a Geoprocessing Service 20
5.4 Developing a Windows Phone Application 22
5.1 Summary 28

Chapter 6 – Results and Analysis **29**
6.1 Crash Simulation 29
6.2 Test Case I: Boeing 737-200 32
6.3 Test Case II: Hawker Beechcraft 125-800A 34
6.4 Test Case III: Lockheed Martin F-22A Raptor 37
6.5 Test Case IV: Piper PA-31 Navajo 39
6.6 Summary 42

Chapter 7 – Conclusions and Future Work **43**
7.1 Conclusion 43
7.2 Future Work 43

Works Cited **45**

Appendix A. Sample Drag Coefficients **47**

Appendix B. CSDRMA Internal List of Aircraft Fixed Values **48**

Appendix C. Original GW-BASIC Code of the Trajectory Analysis Program **49**

Appendix D. CSDRMA Geoprocessing Script Tool Python Code **53**

Table of Figures

Figure 1-1: Project workflow. 2
Figure 4-1: Conceptual data model for project. 13
Figure 4-2: Diagram of the Colorado feature dataset. 14
Figure 4-3: Diagram of the Minnesota feature dataset. 14
Figure 4-4: Diagram of the Alaska feature dataset. 14
Figure 4-4: Diagram of the Hawaii feature dataset. 15
Figure 5-1: The CSDRMA geoprocessing script tool dialog box 20
Figure 5-2: Screenshot of the test geoprocessing service in the ArcGIS Services directory 21
Figure 5-3: CSDRMA loading page. 23
Figure 5-4: CSDRMA main page. 23
Figure 5-5: CSDRMA main page application bar and menu. 23
Figure 5-6: CSDRMA aircraft type section of the calculate pivot page. 24
Figure 5-7: CSDRMA flight info section of the calculate pivot page. 25
Figure 5-8: CSDRMA optional section of the calculate pivot page. 26
Figure 5-9: CSDRMA results section of the calculate pivot page. 27
Figure 5-10: CSDRMA calculate pivot page applicaton bar. 27
Figure 5-11: CSDRMA main page with projected debris field 28
Figure 6-1: Picture of the crash simulation location 29
Figure 6-2: Screenshot of the crash simulation CSDRMA projected debris field 31
Figure 6-3: Test Case I CSDRMA projected debris field compated to the known large debris 34
Figure 6-4: Test Case II CSDRMA projected debris field compated to the known debris field 36
Figure 6-5: Test Case III CSDRMA projected debris field compared to the known debris field 39
Figure 6-6: Test Case IV CSDRMA projected debris field compared to the known debris field 41

List of Tables

Table 1. Sample crash data locations 14
Table 2. CSDRMA geoprocessing script tool inputs. 18
Table 3. CSDRMA geoprocessing script tool non-spatial outputs. 19
Table 4. Geoprocessing service inputs 22
Table 5. Crash Simulation: specific crash variable inputs. 30
Table 6. Crash Simulation: non-spatial data outputs. 30
Table 7. Test Case I: specific crash variable inputs. 32
Table 8. Test Case I: fixed values for Boeing 737-200 33
Table 9. Test Case I: non-spatial data outputs. 33
Table 10. Test Case II: specific crash variable inputs. 35
Table 11. Test Case II: fixed values for Hawker Beechcraft Hawker 800 35
Table 12. Test Case II: non-spatial data outputs. 36
Table 13. Test Case III: specific crash variable inputs. 37
Table 14. Test Case III: fixed values for Lockheed Martin F-22A Raptor 38
Table 15. Test Case III: non-spatial data outputs. 38
Table 16. Test Case IV: specific crash variable inputs 40
Table 17. Test Case IV: fixed values for Piper PA-31 Navajo 40
Table 18. Test Case IV: non-spatial data outputs 41
Table 19. Measure of drag coefficients for selected aircraft (Scott, 2004) 47
Table 20. CSDRMA internal list of aircraft fixed values. 48

List of Acronyms and Definitions

AAIB	UK Air Accidents Investigation Branch
AAIT	Air Accident Investigation Tool
AGL	Above Ground Level
C_d	Drag Coefficient
CIC	Cranfield Impact Center
CRTF	Common Real Time Debris Footprint
CSDRMA	Crash Site Debris Recovery Mobile Application
DEBRA	Debris Risk Assessment Model
DMG	Desert Managers Group
FAA	Federal Aviation Administration
GIS	Geographic Information Systems
GPS	Global Positioning System
GW-BASIC	Version of the BASIC Programming Language
JBER	Joint Base Elmendorf-Richardson, Alaska
KRASH	Debris Modeling Formula Developed by the FAA
kts.	Knots
MDEP	Mojave Desert Ecosystem Program
MSL	Mean Sea Level
NASA	National Aeronautics and Space Administration
NTSB	National Transportation Safety Board
SAR	Search and Rescue
SDK	Software Development Kit
SOMTA	Seat Occupant Model Transport Aircraft
TAP	Trajectory Analysis Program
USAF	United States Air Force
Windows Phone Marketplace	Online Store for Windows Phone Applications

Chapter 1 – Introduction

This chapter explains the need for the Crash Site Debris Recovery Mobile Application (CSDRMA) and how it was developed. Sections 1.1 and 1.2 introduce the client and the problem that the CSDRMA seeks to solve. Section 1.3 presents the need for the CSDRMA, the goals, objectives, and scope of the project, and the methods that will be utilized to develop the applications. Section 1.4 defines the audience for this report and Section 1.5 concludes with an overview of the rest of this report.

1.1 Client

The client for this project is Mr. Fon Allan Duke, the Program Manager of the Mojave Desert Ecosystem Program (MDEP) and Department of Defense Coordinator for the Desert Managers Group (DMG). The MDEP and DMG were created for government agencies in the Mojave Desert area to "address and discuss issues of common concern" and to provide them "with solutions and services to better accomplish their assigned tasks" (Desert Managers Group, n.d.; Mojave Desert Ecosystem Program, n.d.).

A partner of these programs, Mr. Steve Mesa from the National Geospatial Intelligence Agency (NGA), originally proposed the creation of a mobile geographic information system (GIS) tool that could project the estimated debris field of an airplane crash. It was reasoned that armed with this tool, first responders, crash investigators, or other interested parties within the United States military could better perform their assigned tasks in responding to downed aircraft.

Mr. Duke assumed responsibility for the project for Mr. Mesa and functioned as the client for this project.

1.2 Problem Statement

There is a need for a dedicated mobile GIS tool that can assist crash investigators, first responders, and clean-up crews in locating areas of high concentration of debris after a ground impact airplane crash. Such a tool could enable these groups to locate possible survivors, retrieve debris critical to national security, clean up environmental hazards, secure precious metals used in sophisticated airplanes, and perform other related tasks more quickly and efficiently. The client has identified the governmental agencies involved in the MDEP and DMG that would benefit from such a tool.

1.3 Proposed Solution

The proposed solution to this problem was the Crash Site Debris Recovery Mobile Application (CSDRMA) for Windows Phone. The CSDRMA uses a debris trajectory formula, the GPS location of the mobile phone, and user-defined variables of the affected aircraft and its crash characteristics to project an estimated debris field. This section will further discuss the goals and objectives for the CSDRMA project (1.3.1), the scope of the project (1.3.2), and the methods to be used (1.3.3).

1.3.1 Goals and Objectives

The main goals for this project were to (1) develop a solution utilizing GIS software to identify possible areas of high debris concentration, (2) develop a mobile tool with the ability to run the same solution for different crash sites and crash characteristics, and (3) deploy the developed tool as an application for Windows Phone.

The objectives for this project were to (1) secure a formula that calculates aircraft debris trajectories, (2) develop a geoprocessing tool for testing use within Esri's ArcGIS, (3) obtain crash reports of four different airplane crashes and associated debris field shapefiles, and (4) deploy the prototype CSDRMA to the Windows Phone Marketplace.

1.3.2 Scope

Crash reports of four different airplane crashes and associated debris field shapefiles were provided by the client. A debris trajectory formula was also selected and developed into a geoprocessing tool for testing in Esri's ArcGIS against the four aircraft crashes. After testing, the developed geoprocessing tool and incorporated debris trajectory formula were adjusted to more accurately reflect test results.

The CSDRMA for Windows Phone was then developed based on the created geoprocessing tool with additional functionality added (i.e., GPS, internal reference of aircraft dimensions, etc.). A geoprocessing service was also developed and published for use by the Windows Phone based CSDRMA. The CSDRMA was finally deployed to the Windows Phone Marketplace.

1.3.3 Methods

The methodology for this project followed the iterative and incremental development workflow (Figure 1-1). This particular workflow was chosen because it is useful within the software development process given the many iterations of testing that need to be completed to ensure a product of high quality (OTS Solutions, 2009).

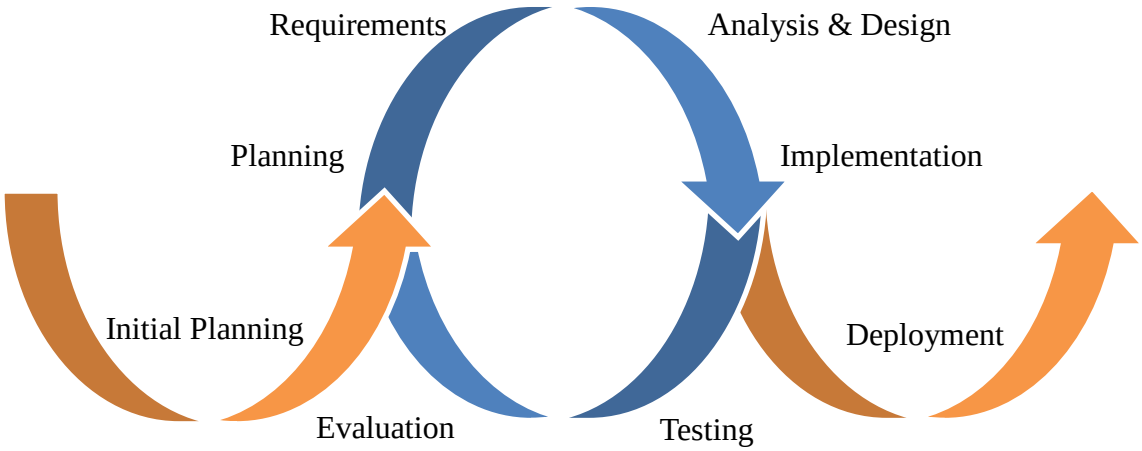

Figure 1-1: Project workflow.

The workflow, as applied to this project, first began with the initial planning of the CSDRMA, which entailed obtaining the necessary equipment, formulas, and the client-provided sample aircraft crash data to proceed. It then moved into the iterative cycle, which began with the planning stage wherein the concept of the geoprocessing tool version of the CSDRMA was planned. Next the requirements stage was completed, which analyzed the compatibility of the client's requirements with the tool concept. Afterwards the analysis and design stage was completed wherein the planned tool's functionality and design were analyzed. At the next stage, implementation, the tool was created. The geoprocessing tool was then tested against the sample aircraft crash data and the results were evaluated in the testing and evaluation stage. The iteration process was then completed again for developing the CSDRMA for Windows Phone based on the created geoprocessing CSDRMA tool. Once this process was completed, the CSDRMA for Windows Phone was deployed to the Windows Phone Marketplace.

1.4 Audience

The intended audience of this report understands GIS and the software utilized within the industry. They also will have an understanding of Windows Phone programming and ArcGIS Server concepts. Additionally, those familiar with aircraft crash investigations, recovery, or response will be familiar with many concepts covered within this report.

1.5 Overview of the Rest of this Report

The remainder of this report is comprised of chapters two through six. Chapter 2 gives a background of aircraft crash search and rescue (SAR) and crash modeling. Chapter 3 details how the CSDRMA was constructed. Chapter 4 includes the database design and organization of the sample airplane crash and testing result data. Chapter 5 details the implementation of the project and Chapter 6 presents the results and analysis of the project. The report concludes with Chapter 7 by identifying areas of future work.

Chapter 2 – Background and Literature Review

This chapter gives a background of the technologies and methods developed in the fields of search and rescue (Section 2.1), debris modeling (Section 2.2), and the incorporation of geographic information systems (GIS) into these fields (Section 2.3). These three areas represent the process of how airplane crash investigations have progressed from merely searching and finding crashes to analyzing and mapping crashes.

This chapter also highlights in Section 2.2 the Trajectory Analysis Program (TAP) utilized in the CSDRMA and other debris models that could be incorporated in future versions (see Section 7.1). The chapter concludes with a summary in Section 2.4

2.1 Search and Rescue

The scientific study of search operations began with a United States Navy research group during World War II. They developed a system of searching for the wakes of enemy warships using aircraft. Later, in the 1950s, the United States Coast Guard applied the research group's work towards water-based rescue applications which would come to be known as search and rescue (SAR) operations. With this adaptation and the subsequent work of scientists over the years, SAR methodology spread globally. SAR methods have since spread to land-based operations, and they remain in widespread use today (Cooper, 2000; Koester, Cooper, Frost, & Robe, 2004).

Though an important part of aircraft investigation and debris recovery, SAR methodology is not able to specifically identify areas of high concentration of debris, which is a requirement of the CSDRMA. SAR methodology could be an area of future functionality to be used in the CSDRMA to help first responders, investigators, and clean-up crews locate debris and search the project debris field (see Section 7.1).

2.2 Debris Modeling

Debris modeling, defined as a process that simulates a specific vehicle's crash dynamics, is a relatively recent development in the investigation of airplane crash events and even spacecraft crash events, especially after the 2003 NASA Space Shuttle Columbia disaster. With its application to airplane crash event investigations, debris modeling has the ability to simulate debris dispersions of a particular aircraft in various circumstances and conditions. With such a simulation, a fairly accurate projection of a debris field could be made and integrated into the CSDRMA, satisfying the requirement of the application to identify an area of high concentration of debris after an airplane crash event.

Subsections 2.2.1 through 2.2.4 examine the Debris Risk Assessment Model, Common Real Time Debris Footprint, Aircraft Accident Investigation Tool, and Trajectory Analysis Program debris models, as well as the feasibility of integrating each into the CSDRMA.

2.2.1 Debris Risk Assessment Model

The Debris Risk Assessment (DEBRA) model was developed by APT Research for use by flight safety analysts to assess the debris risk from a launch vehicle, which is defined

as “a rocket-powered vehicle used to transport a spacecraft beyond Earth’s atmosphere” (APT Research, Inc., 2012; Robledo, 2004; Launch Vehicle, 2012). The model combines “user inputs about nominal trajectory and failure mode information” to produce a hazard area while showing affected populated areas. Specifically, the model has the ability to project hazard areas for a vehicle explosion, malfunction, or engine failure and the number of expected fatalities from such an event (APT Research, Inc., 2012; Robledo, 2004).

Though DEBRA is a risk management tool for launch vehicles, there is a possibility that its underlying methods could be adapted to project a debris field for airplane explosions, malfunctions, or engine failures. Unfortunately, the DEBRA model is proprietary software and APT Research has shared very little about its internal functioning (Robledo, 2004). For this reason, a more in-depth review of the DEBRA model cannot be given, nor could it be incorporated into the CSDRMA at this time.

2.2.2 Common Real Time Debris Footprint

The Common Real Time Debris Footprint (CRTF) program was developed by ACTA for use in estimating the “range of free-fall, the mean impact locations, and impact dispersions of fragments” resulting from a launch vehicle breakup event (Lin, Larson, & Collins, 2003). The CRTF utilizes the ballistic coefficient of launch vehicle debris, which is the measurement of debris “slow[ing] in flight due to air resistance” (Courtney & Courtney, n.d.), and wind in determining a debris field. Additionally, the program runs in real-time and utilizes six different models to handle crash uncertainties (Sala-Diakanda, 2007; Robledo, 2004; Lin, Larson, & Collins, 2003).

Like the DEBRA model, the CRTF program was originally developed for launch vehicles but its underling methodology could be adapted for projecting the debris field of an airplane crash. Many studies have documented the results of the CRTF program, including one conducted on the NASA Space Shuttle Columbia disaster, but the CRTF is also proprietary software and knowledge of the internal functioning is limited and not able to be incorporated into the CSDRMA at this time (see Lin, Larson, & Collins, 2003).

2.2.3 Aircraft Accident Investigation Tool

The Aircraft Accident Investigation Tool (AAIT), as its name suggests, was developed specifically for airplane crashes and was developed in response to a 1989 Boeing 737-400 crash that occurred near Kegworth, England. The United Kingdom Air Accidents Investigation Branch (AAIB) and the United Kingdom Ministry of Defense commissioned the Cranfield Impact Center Ltd. (CIC) to develop the AAIT with assistance from the US Federal Aviation Administration (FAA) (Soltis, n.d.).

The AAIT is built upon KRASH, a program developed by NASA and the FAA that “analyses airframe crash effects” (NASA, 1999), and also incorporates the SOMTA (Seat Occupant Model Transport Aircraft) model for occupant simulation. With these incorporations and a series of user-inputs, including the selection of the affected airplane model from a built-in airplane list, the AAIT is able to simulate a crash event and report on the crash dynamics (Evans, 1996; Soltis, n.d.).

The AAIT has been successfully used in the crash investigations of other aircraft and has been documented extensively (Soltis). But its internal functioning is not openly available and it is not able to be incorporated into the CSDRMA at this time.

2.2.4 Trajectory Analysis Program

The Trajectory Analysis Program (TAP) was developed to determine safe distances between aircraft and spectators in an airshow environment. An FAA formula existed for such distances, but Mr. Hugh Oldham, the TAP developer, concluded that it did not provide for an adequate distance. As a result, Oldham expanded and customized a software program originally developed by the NTSB for airshow spectator distance calculations with input from the ISASI (International Society of Air Safety Investigators), aviation safety experts, and aerospace engineers and produced the TAP (Oldham, 1990).

The TAP projects scatter patterns of "in-flight airframe separations debris, specific to air-show environments" utilizing a ballistic trajectory, defined as "the trajectory traced after the propulsive force is terminated and the body is acted upon only by gravity and aerodynamic drag" (Robledo, 2004; Defense Technical Information Center, 2012).

The TAP model requires the following inputs:

1. Initial altitude of disintegration.
2. Initial density altitude. Density altitude refers to altitude "measured in terms of the density of the air rather than the distance from the ground" (Density altitude, 2012).
3. Altitude of Impact from ground level.
4. Wind velocity and direction.
5. Horizontal true airspeed at disintegration.
6. Rate of climb or sink at disintegration.
7. Weight of projectile.
8. Projectile drag coefficient (C_d). The measure of the "drag or resistance of an object in a fluid environment such as air or water" (see Appendix A) (Drag coefficient, 2012).
9. Projectile frontal area. The area exposed to airflow (Oldham, 1990).

The outputs of the TAP model are the following:

1. Horizontal distance from disintegration at impact.
2. Horizontal, vertical, and total velocities.
3. Terminal velocity.
4. Time to fall.
5. Flight-path angle at impact. The degrees below the horizon the aircraft is descending (-90 equals straight down) (Oldham, 1990).
6. Ground speed of projectile at impact and x and z components of that velocity. This can be used to calculate flight path (Oldham, 1990).

The TAP model in general has been researched very little and has never been documented being used to investigate an airplane crash event. But its internal functioning

and formulas are open, basic, and fairly simple to use (see Appendix C). For these reasons the TAP model was chosen to be integrated into the development of the CSDRMA. In future development, the TAP debris model could be substituted with another debris model with minimal modification to the CSDRMA. Re-programming the CSDRMA to handle additional parameters or formulas would be fairly uncomplicated.

2.3 Geographic Information Systems Integration

To extend the utility of an aircraft crash simulation generated by a debris model, the simulation can also be georeferenced, or mapped, via geographic information systems (GIS) software. Robledo performed such a task that showcased such an integration in mapping debris from the 2003 NASA Space Shuttle Columbia disaster in his disertation "Analysis and Integration of a Debris Model in the Virtual Range Project" (2004).

GIS software can be even further integrated into SAR operations. In a study conducted on the usefulness of GIS in SAR operations of a crashed Turkish Air Force airplane, Söylemez and Dr. Usul concluded that GIS analyses "can help to decrease search time and search efforts" in SAR operations (n.d.).Though this was a study of SAR operations, the CSDRMA has the potential to be applied within SAR operations. As a result, GIS software will be integrated with the TAP debris model within the CSDRMA in order to produce a georeferenced area of high concentration of debris, with the intention of decreasing search time and effort required by airplane crash investigators, first responders, and clean-up crews.

2.4 Summary

With a backdrop of SAR, debris modeling, and GIS fields, one can more fully understand how the CSDRMA has the potential to fit into the broader fields of airplane crash response, investigation, and debris recovery. Specifically this chapter found SAR operational methods not able to meet the CSDRMA requirement of being able to identify areas of high concentration of debris after an airplane crash event; though identified as an area of future development. The chapter also identified the TAP model as the only freely available and least complex debris model that could be integrated into the CSDRMA. The value of integrating GIS into the CSDRMA was also reviewed.

Chapter 3 – Systems Analysis and Design

This chapter highlights how the CSDRMA has been specifically designed to address the client's problem. Section 3.1 restates the problem being addressed, Section 3.2 provides an overview of the functional and non-functional requirements of the application, Section 3.3 presents the system design, Section 3.4 details the project plan, and Section 3.5 concludes with a summary of the items presented in this chapter.

3.1 Problem Statement

There is a need for a dedicated mobile GIS tool that can assist crash investigators, first responders, and clean-up crews in locating areas of high concentration of debris after a ground impact airplane crash. Such a tool could enable these groups to locate possible survivors, retrieve debris critical to national security, clean up environmental hazards, secure precious metals used in sophisticated airplanes, and perform other related tasks more quickly and efficiently. The client has identified the governmental agencies involved in the MDEP and DMG that would benefit from such a tool.

3.2 Requirements Analysis

This section reviews the functional (3.2.1) and non-functional requirements of the CSDRMA. The non-functional requirements are broken down into technical requirements (3.2.2), operational requirements (3.2.3), and transitional requirements (3.2.4).

3.2.1 Functional Requirements

The following are the functional requirements of the CSDRMA:

1. Require user input of crash characteristics. These characteristics specifically are the aircraft speed at time of impact, the aircraft directional heading, and the angle of descent. This allows the CSDRMA to be applied to different ground impact crash situations.
2. Allow for optional user input of additional crash characteristics. These include ground wind speed and direction, direction of terrain slope (i.e., upslope, downslope, none), and the angle of the terrain slope. With the inclusion of these values the accuracy of the estimated debris improves, but they can be omitted if unknown.
3. Provide a list of 16 aircraft that link to an internal list of associated aircraft dimensions and performance characteristics. This limits the number of required user inputs for values that remain constant for each aircraft type. Additionally, this limits user error and allows for a debris field to be calculated faster.
4. Allow for results to be saved. This enables the user to retrieve past inputs and results for further analysis.
5. Generate a map projecting the estimated debris field and display the real-time location of the mobile phone. This enables the user to navigate the crash scene while having a reference to the estimated debris field.

3.2.2 Technical Requirements

The following are the technical requirements of the CSDRMA:

1. Run as a native Windows Phone 7 application.
2. Develop, publish, and utilize a geoprocessing service to create and display an estimated debris field.

3.2.2 Operational Requirements

The following are the operational requirements of the CSDRMA:

1. Allow downloading by the public via the Windows Phone Marketplace.
2. Host the developed geoprocessing service utilized within the application on the client's server.

3.2.3 Transitional Requirements

The following are the transitional requirements of the CSDRMA:

1. Provide the client with the geoprocessing service materials for publishing on its own server.
2. Publish to the Windows Phone Marketplace.
3. Provide any necessary documentation to the client for inclusion on their website.

3.3 System Design

The CSDRMA was designed as a native Windows Phone 7 application that is downloadable from the Windows Phone Marketplace. Within the application itself, the main page was designed to show a map with the GPS location of the phone and serve as the main navigation page. From here, the user is able to access help documentation and the “calculate” page, in which crash characteristics and aircraft dimensions are set.

In the calculate area, a windows phone pivot page is utilized in which the user can select the type of aircraft involved in a crash and set the required values (i.e., the aircraft speed at time of impact, the aircraft directional heading, and the angle of descent). The application automatically populates the *aircraft speed at time of impact* field for user reference with a suggested speed based on the cruising speed of the selected aircraft; the user can easily change this field. Additionally, the user can set optional crash characteristics (i.e., ground wind speed and direction, direction of terrain slope, and angle of slope) on this pivot page.

The user is then able to tap the “Calculate” button in which the terminal velocity, time to ground impact, throw distance, angle of ground impact, speed of impact, and maximum altitude of aircraft debris are calculated using an internal reference of airplane characteristics and the debris trajectory analysis. The results are then displayed within the current window. In this window the user has the option of saving these results for later use and has the ability to view previously saved calculated results.

Once the calculation is complete, the user is able to tap the “Map” button which will navigate to the main page and display the estimated debris field on a basemap using the geoprocessing service hosted on the client’s server. The GPS location of the phone will still be displayed for ease in navigating the crash site and the estimated debris field.

3.4 Project Plan

The CSDRMA project plan shifted dramatically from what was originally planned. The main reason for this shift was that research was performed on software platforms which were originally thought to be able to meet the project requirements, namely Esri’s ArcGIS for Windows Mobile and Esri’s ArcPad. It was found that they lacked advanced customization capabilities which are critical to developing the CSDRMA. With further research it was found that a Windows Phone 7 native application using Esri’s ArcGIS Runtime Software Development Kit (SDK) for Windows Phone could meet the project requirements and was used accordingly.

Another reason for the shift was the lack of research on the availability of published geoprocessing services for publicly available use. It was discovered late in the project that there is a lack of these publicly available services and additional time had to be taken to create and publish a dedicated geoprocessing service for the CSDRMA.

Additionally, the development of a prototype CSDRMA python script tool and the Windows Phone CSDRMA took longer than expected. It was discovered late in the development of these solutions that it was necessary to convert degree values to radian values in order for the solutions to correctly project a debris field. This subsequently delayed the testing of the CSDRMA.

3.5 Summary

With an understanding of the client’s problem and an understanding of the functional and non-functional requirements, the CSDRMA was able to move forward with a clear direction. Subsequently, the CSDRMA was designed as a complete application that meets the client’s needs. Though there were a few setbacks with software platform changes and development delays, the project was carried out successfully.

Chapter 4 – Database Design

This chapter discusses the organization of the CSDRMA project. Section 4.1 reviews the conceptual data model for the project and Section 4.2 reviews the logical data model for data used for testing. Sections 4.3 and 4.4 review data sources and how data were collected. Section 4.5 details the process of how data were scrubbed and loaded.

4.1 Conceptual Data Model

The conceptual data model diagram for this project (Figure 4-1) illustrates the client's problem. The diagram details that in order to project an airplane crash debris field, an aircraft type, along with its associated characteristics, must first be selected. Next the flight characteristics must be input. Then, if available, the environmental characteristics are input. From these inputs, the debris field characteristics are calculated and used to create the estimated debris field.

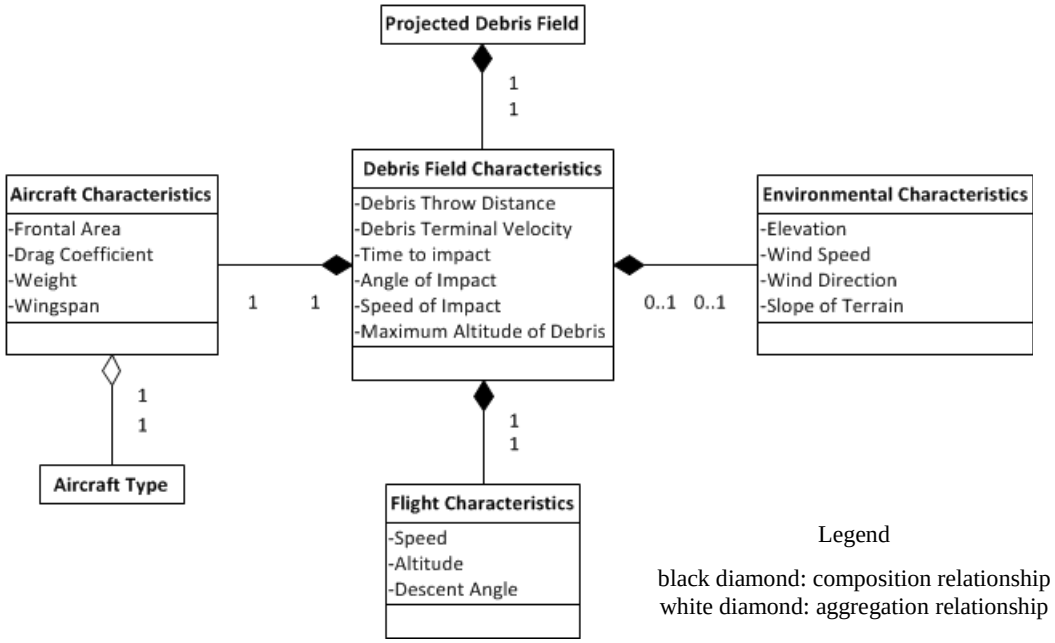

Figure 4-1: Conceptual data model for project.

4.2 Logical Data Model

A geodatabase was designed for the feature classes used in testing the CSDRMA and the feature classes test results. Specifically four feature datasets were created in order to organize the data by crash site. Each dataset was named after the U.S. state in which the aircraft crashed (see Table 1). Additionally, each feature class name contains the two-letter United States Postal Service abbreviation for state names and a standardized

description of the data (e.g., “CO_Crash_Site_point”). Figures 4-2 through 4-4 detail the organization within each feature dataset.

Table 1. Sample crash data locations.

Airplane Type	Location
Boeing 737-200	Colorado
Hawker Beechcraft 125-800A	Minnesota
Lockheed Martin F-22A Raptor	Alaska
Piper PA-31 Navajo	Hawaii

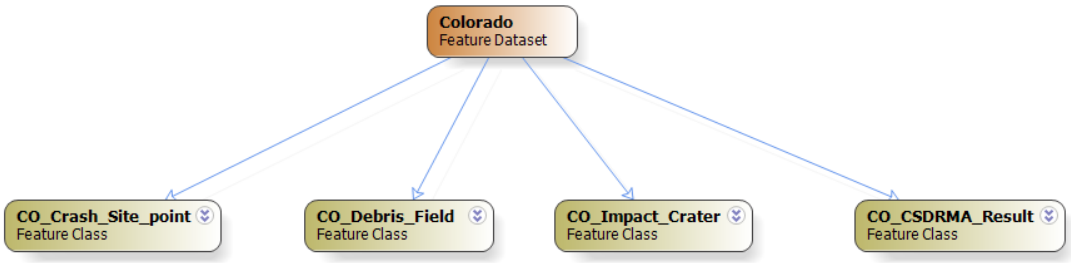

Figure 4-2: Diagram of the Colorado feature dataset.

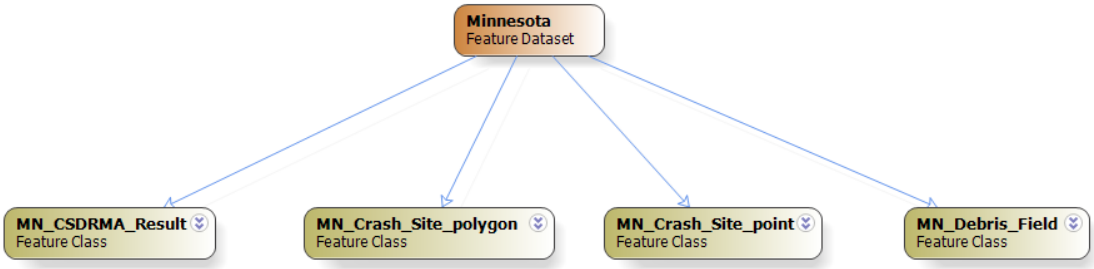

Figure 4-3: Diagram of the Minnesota feature dataset.

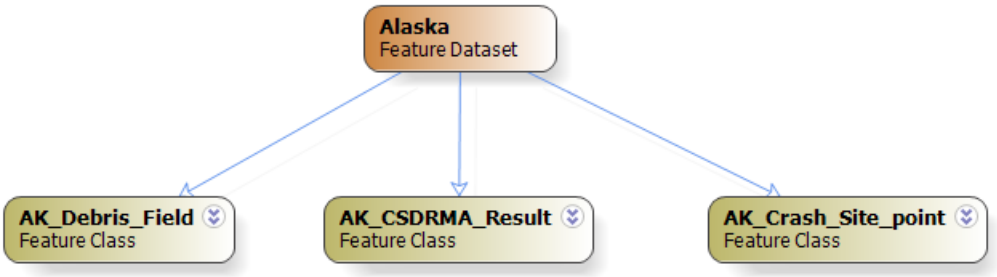

Figure 4-4: Diagram of the Alaska feature dataset.

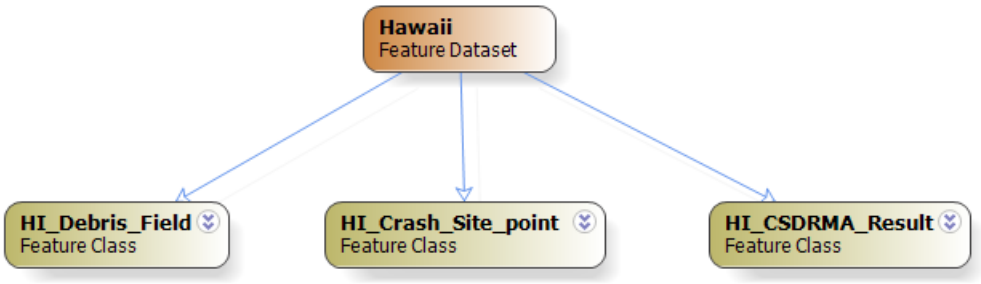

Figure 4-5: Diagram of the Hawaii feature dataset.

4.3 Data Sources

All feature classes used for testing the CSDRMA were provided by the client. The sources used by the client to create the feature classes for the Lockheed Martin F-22A Raptor crash came from a U.S. Air Force Aircraft Accident Investigation Board Report (United States Air Force Aircraft Accident Investigation Board, 2011) and a U.S. Bureau of Land Management Environmental Assessment (Hart, 2011) . The sources used to create the feature classes for the Boeing 737-200, Hawker Beechcraft 125-800A, and Piper PA-31 Navajo crashes came from the U.S. National Transportation Safety Board (2001a, 2001b, 2011).

4.4 Data Collection Methods

MDEP staff digitized the impact point location feature classes and debris field feature classes for the four sample crash sites using publicly available crash reports (see Section 4.3) and aerial imagery.

4.5 Data Scrubbing and Loading

Datasets received from the client were renamed to a standard naming convention using the two-letter United States Postal Service standard for state names based on the state in which the crash occurred. Additionally, a standardized description of the data was also included in the title of the individualized feature classes (e.g., “CO_Crash_Site_point”). The feature classes were stored in a geodatabase and organized by U.S. state location into feature datasets. The CSDRMA test results (in feature class format) were also loaded into their corresponding feature dataset. Corresponding crash reports and CSDRMA test summaries were additionally stored within the feature datasets for reference.

4.6 Summary

This chapter reviewed the conceptual data model of the client’s problem and how testing data and results were organized into a single geodatabase. Specifically, the data and results were organized into feature datasets based on the U.S. state in which the crash occurred. Additionally the crash reports and testing reports were loaded into the geodatabase.

Chapter 5 – Implementation

This chapter highlights the major steps taken to develop the CSDRMA for Windows Phone. Specifically, Section 5.1 reviews the debris model preparation that was completed and Section 5.2 details how the CSDRMA geoprocessing script tool was developed in order to be used for testing. Section 5.3 reviews the development and subsequent publishing of the geoprocessing service utilized by the CSDRMA for Windows Phone, and Section 5.4 details the development of the final product, the CSDRMA for Windows Phone. Section 5.5 concludes with a summary of the chapter.

5.1 Debris Model Preparation

The beginning of the CSDRMA project entailed conducting a review of available debris models for integration into the application. After an extensive review, the Trajectory Analysis Program (TAP) was chosen because of its internal functioning and formulas were open, basic, and fairly simple to use (see Section 4.2.2).

The particular copy of the TAP debris model used for the CSDRMA had several typographical, mathematical, and functional errors, and it was written using GW-BASIC programming code (see Appendix C). As this type of programming code is not compatible with the technologies utilized in this project, the TAP programming code was ported to Python programming code; additionally it was corrected and tested to ensure its functionality. Also, the original code called for an optional "density altitude" input that measured altitude "…in terms of the density of the air rather than the distance from the ground" (Density altitude, 2012). This input was eliminated as it was determined that crash first responders, investigators, and clean-up crews would not have this information readily available, and to reduce the number of required inputs.

Additionally, the TAP did not include any means to calculate the effect terrain had on the debris trajectory. As a result, a formula was created and incorporated into the converted Python programming code based on concepts discussed by Wood and Sweginnis in their book "Aircraft Accident Investigation, 2nd Edition" (2006).

5.2 Developing a Geoprocessing Script Tool

The Python programming code was further modified with the integration of ArcGIS functionality by means of the ArcPy site package. Specifically, the code was adjusted so that it first takes the geoprocessing script tool inputs, including an input point feature class of the airplane crash point of impact, and runs the modified TAP debris model (Table 2). Then it defines the spatial reference as WGS 84 using the European Petroleum Survey Group (EPSG) Geodetic Parameter of "4326", and defines the geoprocessing workspace as "in_memory". Next it creates a feature class and a table with four fields to hold longitude, latitude, length, and bearing values. Afterwards, the "Add XY Coordinates" geoprocessing tool is used to obtain the x and y coordinates from the input point feature class. Then, using the cursor function, the table is updated with the output values from the XY geoprocessing tool for the latitude and longitude fields, and the geoprocessing script tool inputs for the remaining fields. Next the "Bearing Distance To

Line" geoprocessing tool is run using the updated table. Finally, the "Buffer" geoprocessing tool is run on the "Bearing Distance To Line" output using $^{wingspan}/_{2}$ as the buffer distance. The output of the "Buffer" tool (i.e., the projected debris field) is then returned to the user along with a message detailing the non-spatial outputs generated by the modified TAP (Table 3).

Table 2. CSDRMA geoprocessing script tool inputs.

Required	Optional
Crash Location Feature Class	Ground Level Wind Speed (kts.)
Output Debris Field Feature Class	Ground Level Wind Direction (deg.)
Speed of Aircraft (kts.)	Angle of Terrain (deg.)
Altitude of Aircraft (ft. AGL)	Terrain Characteristic ("None", "Upslope", "Downslope")
Aircraft Heading (deg.)	
Descent of Aircraft (deg.)	
Frontal Area of Aircraft (sq. ft.)	
Drag Coefficient of Aircraft (C_d)	
Weight of Aircraft (lbs.)	
Wingspan (ft.)	
Ground Level (ft. MSL)	

Table 3. CSDRMA geoprocessing script tool non-spatial outputs.

Debris Terminal Velocity (kts.)
Time to Impact (sec.)
Debris Throw Distance (ft.)
Angle of Impact (deg.)
Speed of Impact (kts.)
Max Altitude of Thrown Debris (ft.)

Once the CSDRMA geoprocessing script tool programming code was integrated with GIS functionality, an associated geoprocessing script tool was created in Esri's ArcGIS (Figure 5-1). The parameters for the tool were the same as the CSDRMA geoprocessing script tool inputs found in Table 2. The tool was then ready for testing in ArcGIS (see Sections 6.1-6.4).

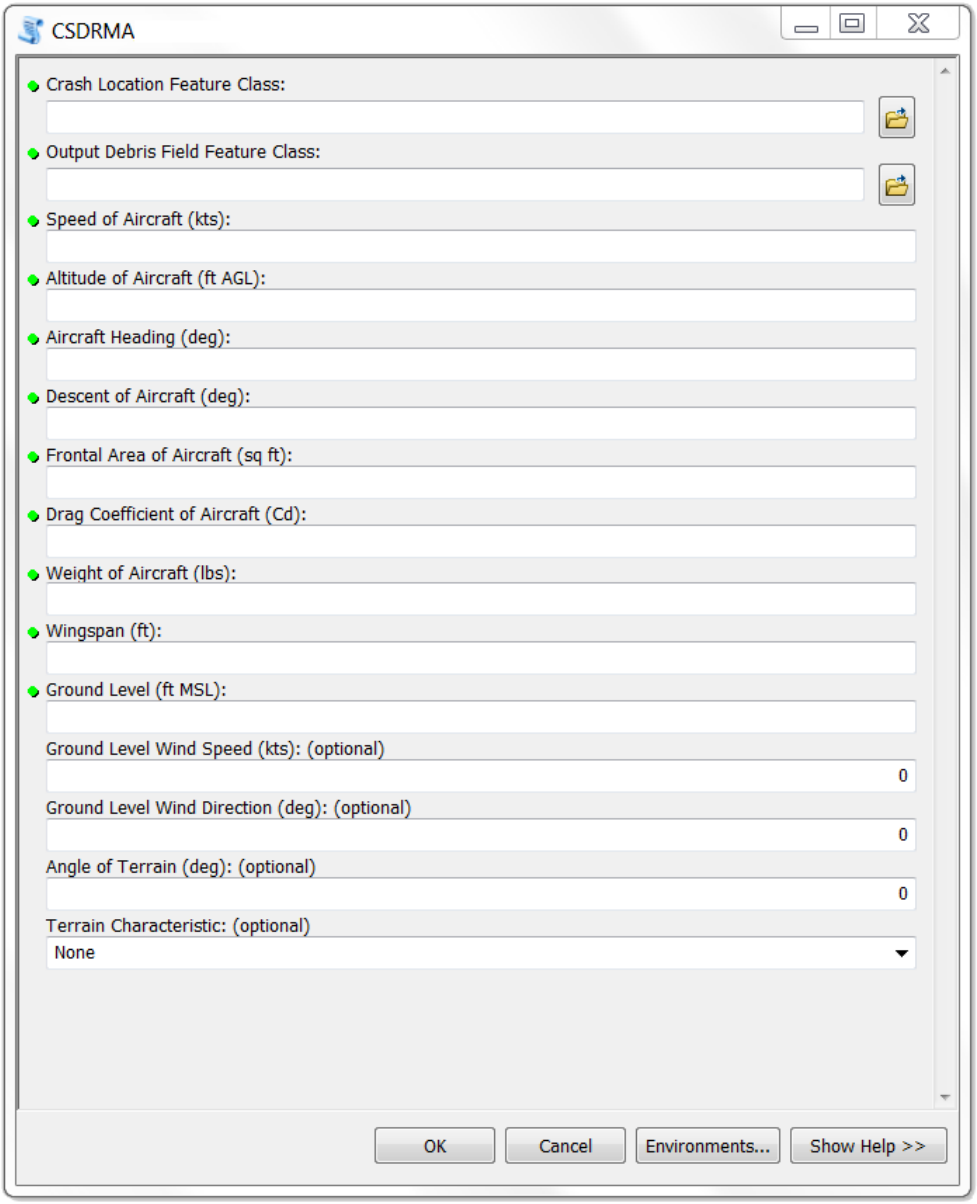

Figure 5-1: The CSDRMA geoprocessing script tool dialog box.

5.3 Publishing a Geoprocessing Service

Once the CSDRMA geoprocessing script tool testing was complete, a geoprocessing service was published for use in testing and developing the CSDRMA for Windows Phone (Figure 5-2). The service was designed to run asynchronously and complete similar geoprocessing tasks as detailed in Section 5.2 but using only the latitude and longitude of the point of impact rather than a point feature class, using fewer inputs (see Table 4), and the only output being the "Buffer" geoprocessing tool result (i.e., the projected debris field). Once the CSDRMA for Windows Phone and the geoprocessing service were working in harmony and producing results, a final version of the geoprocessing service was given to the client for hosting on their server. Subsequently, the CSDRMA for Windows Phone geoprocessing service URL (uniform resource

locator) was updated to connect with the geoprocessing service now hosted on the client's server.

Task: CSDRMA

Display Name: CSDRMA

Category:

Help URL: http://msgis-ags-1/arcgisoutput/C20-Nicholas_Janzen_CSDRMA/CSDRMA.htm

Execution Type: esriExecutionTypeAsynchronous

Parameters:

Parameter: Impact_Latitude
Data Type: GPDouble
Display Name: Impact Latitude
Direction: esriGPParameterDirectionInput
Default Value: 0
Parameter Type: esriGPParameterTypeRequired
Category:

Parameter: Impact_Longitude
Data Type: GPDouble
Display Name: Impact Longitude
Direction: esriGPParameterDirectionInput
Default Value: 0
Parameter Type: esriGPParameterTypeRequired
Category:

Parameter: Debris_Throw_Distance
Data Type: GPDouble
Display Name: Debris Throw Distance
Direction: esriGPParameterDirectionInput
Default Value: 0
Parameter Type: esriGPParameterTypeRequired
Category:

Parameter: Aircraft_Heading
Data Type: GPDouble
Display Name: Aircraft Heading
Direction: esriGPParameterDirectionInput
Default Value: 0
Parameter Type: esriGPParameterTypeRequired
Category:

Parameter: Aircraft_Wingspan
Data Type: GPDouble
Display Name: Aircraft Wingspan
Direction: esriGPParameterDirectionInput
Default Value: 0
Parameter Type: esriGPParameterTypeRequired
Category:

Parameter: Out
Data Type: GPFeatureRecordSetLayer
Display Name: Out
Direction: esriGPParameterDirectionOutput
Default Value:
Parameter Type: esriGPParameterTypeDerived
Category:

Supported Operations: Submit Job

Supported Interfaces: REST

Figure 5-2: Screenshot of the test geoprocessing service in the ArcGIS Services directory.

Table 4. Geoprocessing service inputs.

Point of Impact Latitude
Point of Impact Longitude
Debris Throw Distance (ft.)
Aircraft Heading (deg.)
Aircraft Wingspan (ft.)

5.4 Developing a Windows Phone Application

Once the CSDRMA geoprocessing script tool testing was completed and a test geoprocessing service was published, The CSDRMA for Windows Phone was developed. The CSDRMA was developed to open on the *loading* page (Figure 5-3) and then transition to the *main* page with an Esri aerial basemap of North America and display the phone's GPS location (displayed as a blue circle on the map) (Figure 5-4). From the application bar on this main page, the user can zoom in to their GPS location by pressing the *me* button (), define the location of the crash site by pressing the *impact* button (), project an airplane crash debris field by pressing the *calculate* button (), and access the help documentation by pressing the *help* button () (Figure 5-5). Additionally, on the application menu bar, which is accessed by tapping the three dot symbol () in the far-right of the application bar, the user can switch between Esri's imagery basemap and Esri's street basemap (Figure 5-5).

Figure 5-3: CSDRMA loading page.

Figure 5-4: CSDRMA main page.

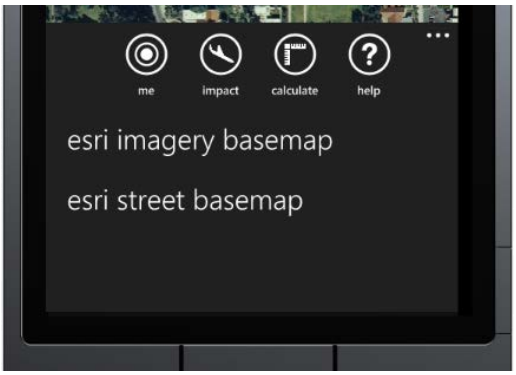

Figure 5-5: CSDRMA Main Page Application Bar and Menu.

Once the user sets the location of the airplane crash site (by tapping the *impact* button while the phone is positioned in the impact location and confirming the location in a pop-up dialog box), the projected debris field can be calculated by first tapping the *calculate* application bar button. The user is then brought to the *aircraft type* section of the *calculate* pivot page (a pivot page is a unique Windows Phone feature that allows for multiple sections of a page to be viewed by manually sliding the section from right-to-left while remaining on a main page) (Figure 5-6). Here the user is able to select from a list the type of aircraft involved in the crash event for which they wish to calculate a projected debris field. With this selection, fixed values of the selected aircraft will later be retrieved from an internal list and used in the modified TAP (see Appendix B).

Figure 5-6: CSDRMA aircraft type section of the calculate pivot page.

With the aircraft type selected, the user is then able to swipe the screen from right-to-left in order to navigate to the *flight info* section of the *calculate* pivot page (Figure 5-7). Here the user is required to enter the speed of the crashed aircraft, its heading, and the angle of its descent. These user inputs are restricted to numbers and the aircraft heading and angle of the aircraft's descent are also restricted by ranges, 1-360 and 0-90, respectively. If a user attempts to enter a value in violation of these restrictions, a pop-up dialog box appears, citing the error. Additionally for user reference, when the aircraft selection changes, the aircraft speed user input field is populated with the cruising speed of the selected aircraft (which is retrieved from the applications internal list of aircraft fixed values).

Figure 5-7: CSDRMA Flight Info Section of the Calculate Pivot Page.

With another right-to-left swipe, the user navigates to the *optional* section of the *calculate* pivot page (Figure 5-8). The user input values found here, as the section name suggest, are all optional and are used for obtaining a more accurate result. Specifically, the values are ground wind speed, ground wind direction, terrain angle, and aircraft terrain direction (*upslope* or *downslope*). Like the user input values in the flight info section, the user inputs values in this section are also restricted to numbers and the inputs measured in degrees have a range restriction of 1-360.

Figure 5-8: CSDRMA optional section of the calculate pivot page.

Again the user is able to swipe right-to-left to navigate to the *results* section of the *calculate* pivot page. When this section first appears no results are shown until the user presses the *Calculate* button, at which point the modified TAP is run using the user inputs and the constant values of the user selected aircraft. If any required inputs have been left blank by the user, a pop-up dialog box will appear with an error identifying the missing input and the calculation will be stopped. If no inputs are missing, all of the required and optional user inputs are displayed along with the modified TAP results (Figure 5-9). The specific modified TAP results are debris terminal velocity, time to impact, debris throw distance, angle of impact, speed of impact, and max altitude of thrown debris.

Figure 5-9: CSDRMA results section of the calculate pivot page.

Throughout the calculate pivot page, the application bar allows the user to navigate back to the main page by pressing the "map" button, and to access the application's help documentation by pressing the *help* button (Figure 5-10). The *save* and *open* buttons are only useful for the results section of the pivot page. Once a result is calculated, the user can press the save button and save the results to the phone's isolated storage (see Isolated Storage Overview for Windows Phone, 2012). The result is saved as a text file (.txt) with a name specified by the user. As the name suggests, the open button allows a user to open previously saved results and display them within the results section of the *calculate* pivot page.

Figure 5-10: CSDRMA calculate pivot page application bar.

Once the result is calculated, the *Map* button can then be pressed which will navigate the user back to the main page. At that time, the CSDRMA connects with the previously developed geoprocessing service, sends the service the required inputs, and receives the projected debris field from the service. The projected debris field is then overlaid on the selected Esri basemap (Figure 5-11).

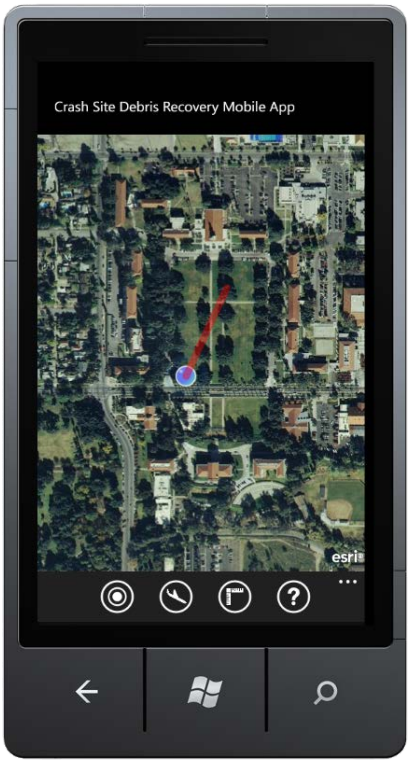

Figure 5-11: CSDRMA Main Page with Projected Debris Field.

As mentioned in Section 5.3, once the CSDRMA for Windows Phone was fully developed and producing acceptable results in testing (see Section 6.5), the geoprocessing server URL (uniform resource locator) was updated to connect with the final geoprocessing service being hosted on the client's server. Additionally, the CSDRMA Windows Phone was published to the Windows Phone Marketplace.

5.5 Summary

This chapter reviewed the major steps of the implementation of the CSDRMA project, namely debris model preparation, developing a geoprocessing script tool, publishing a geoprocessing service, and developing a Windows Phone application. Through these steps, the CSDRMA was developed, tested, and published to the Windows Phone Marketplace as the CSDRMA for Windows Phone.

Chapter 6 – Results and Analysis

This chapter details the results and analysis of testing the CSDRMA. Section 6.1 reviews a crash simulation conducted to determine if the CSDRMA for Windows Phone can run as a Windows Phone application. Sections 6.2 through 6.5 are test cases I-IV conducted in Esri's ArcGIS using a geoprocessing script tool version of the CSDRMA to determine its accuracy. Section 6.6 concludes with a summary of the chapter.

6.1 Crash Simulation

A crash simulation was conducted to test if the CSDRMA for Windows Phone can run as a Windows Phone application (no real airplane crash sites were readily available). The simulation occurred on the southwest corner of the University of Redlands Quad in Redlands, California (Figure 6-1) and involved a mock Cessna 310 airplane. An HTC Arrive mobile phone running Windows Phone 7.5 was utilized for this simulation.

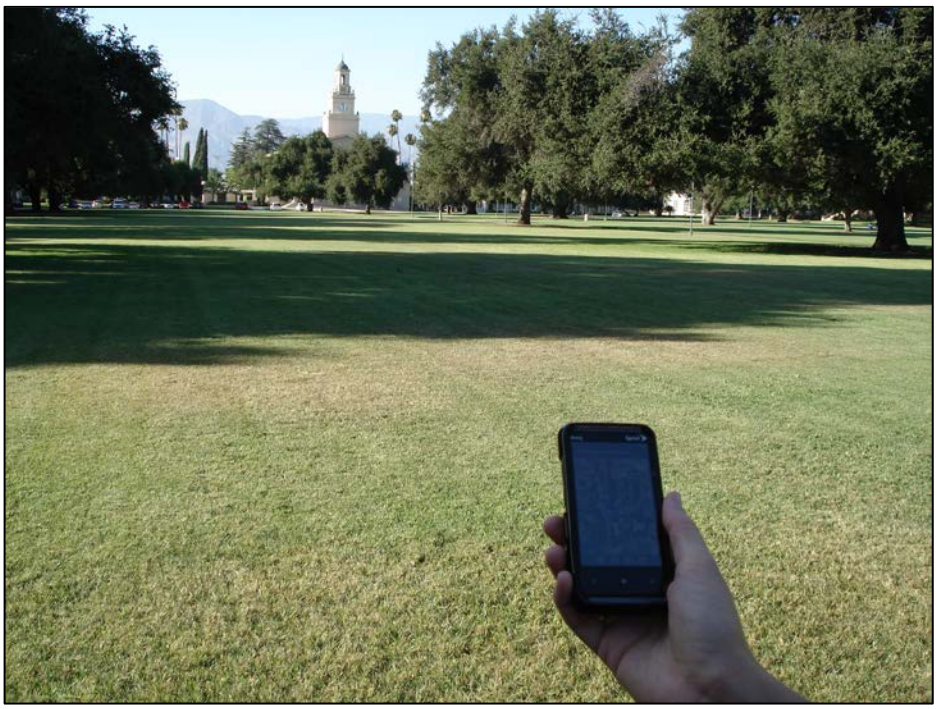

Figure 6-1: Picture of the crash simulation location.

Table 5 displays the specific crash variable inputs as required by the CSDRMA for Windows Phone, and Table 6 displays the non-spatial data outputs. Figure 6-2 shows the CSDRMA screenshot of the projected debris field for the mock crash event.

Table 5. Crash Simulation: specific crash variable inputs.

Specific Crash Variable	Variable Value
Speed of aircraft	98 kts.
Aircraft heading	32°
Angle of aircraft descent	44°
Ground level wind speed	3 kts.
Ground level wind direction	272°
Angle of terrain	0°
Terrain slope characteristic	None (flat)

Table 6. Crash Simulation: non-spatial data outputs.

Specific Non-Spatial Data Output	Output Value
Debris terminal velocity	622 kts.
Time to impact	7 sec.
Debris throw distance	828 ft.
Angle of impact	-44°
Speed of impact	109 kts.
Max altitude of thrown debris	198 ft.

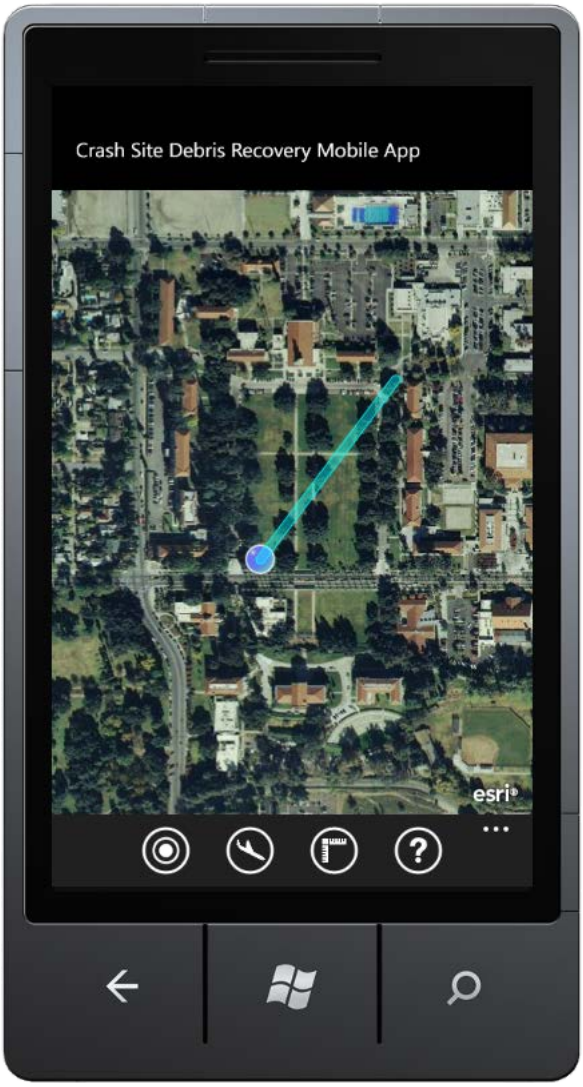

Figure 6-2: Screenshot of the crash simulation CSDRMA projected debris field.

The results of the crash simulation show that the CSDRMA for Windows Phone generated a plausible projected debris field for a Cessna 310 airplane crash. Additionally, the simulation proved that the mobile phone, internet connectivity, and the CSDRMA for Windows worked together correctly in a mobile environment. As a result, the crash simulation was deemed successful.

6.2 Test Case I: Boeing 737-200

The first test case used in testing the CSDRMA's accuracy in ArcGIS was that of a Boeing 737-200 crash that occurred on March 3, 1991. United Airlines flight 585 was preparing to land at Colorado Springs Municipal Airport, in Colorado Springs, Colorado, when there was a loss of control from an equipment malfunction. The plane crashed about 3.5 miles from the runway in a nearby park (National Transportation Safety Board, 2001a).

Information from the National Transportation Safety Board (NTSB) report on the crash, and aircraft characteristics of a Boeing 737-200, were acquired and used as inputs to the geoprocessing script tool version of the CSDRMA. Table 7 details the crash variables specific to the event, and Table 8 shows the fixed values for a Boeing 737-200 crash event. Table 9 shows the non-spatial outputs generated by the CSDRMA, and Figure 6-3 compares the Test Case I CSDRMA projected debris field with the known debris field.

Table 7. Test Case I: specific crash variable inputs.

Specific Crash Variable	Variable Value
Speed of aircraft	200 kts.
Altitude of aircraft	5,705 ft. AGL
Aircraft heading	20°
Angle of aircraft descent	80°
Ground level	5,704 ft. MSL
Ground level wind speed	22 kts.
Ground level wind direction	300°
Angle of terrain	0°
Terrain slope characteristic	None (flat)

Table 8. Test Case I: fixed values for Boeing 737-200.

Specific Fixed Value	Fixed Value
Frontal area of aircraft	1,098 sq. ft.
Drag coefficient of aircraft	0.03 C_d
Weight of aircraft	115,500 lbs.
Wingspan	93 ft.

Table 9. Test Case I: non-spatial data outputs.

Specific Non-Spatial Data Output	Output Value
Debris terminal velocity	1,144 kts.
Time to impact	20 sec.
Debris throw distance	1171 ft.
Angle of impact	-80.26°
Speed of impact	225 kts.
Max altitude of thrown debris	7,382 ft.

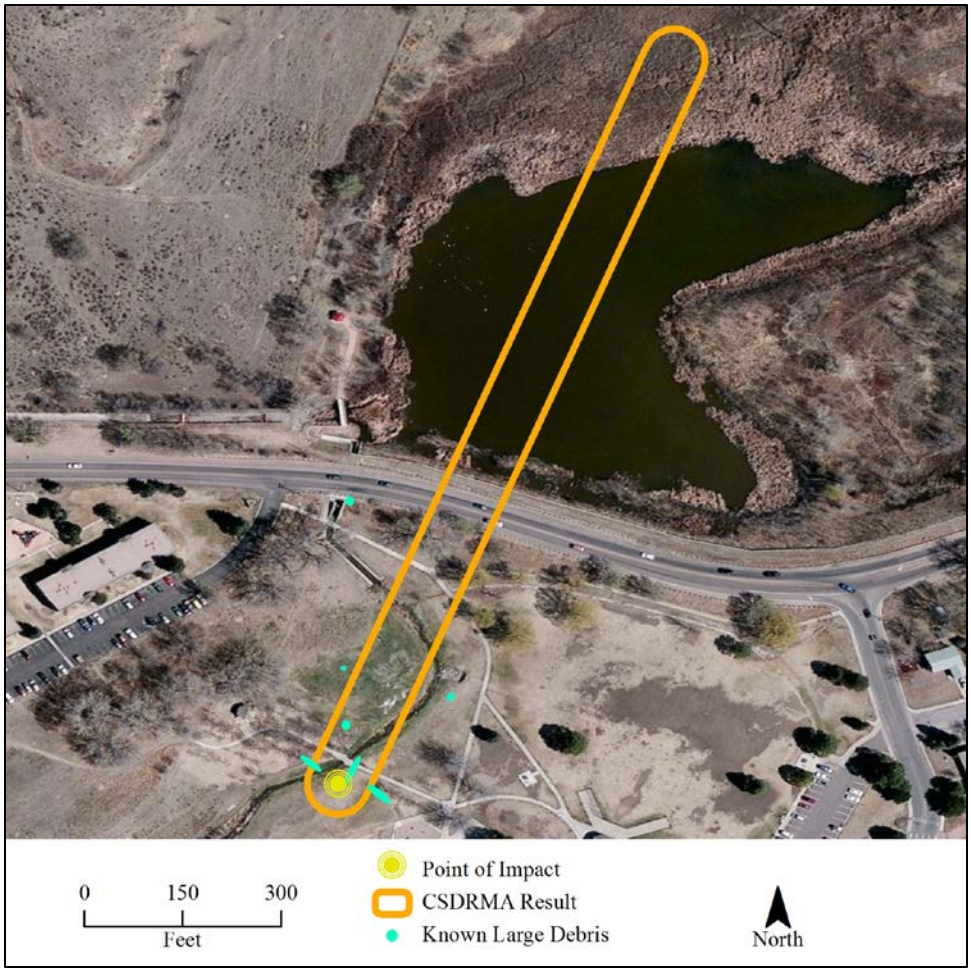

Figure 6-3: Test Case I CSDRMA projected debris field compared to the known large debris.

The results of Test Case I show that the CSDRMA projected debris field coincides with much of the known large debris (Figure 6-3). Additionally, the CSDRMA result does suggest that debris could have travelled into and over the body of water northeast of the point of impact. Although the CSDRMA projected debris results don't cover all of the known debris, debris is located only about 100 feet from the projected debris field which could easily be spotted in the open park environment by first responders, investigators, and clean-up crews, and as a result Test Case I was deemed successful.

6.3 Test Case II: Hawker Beechcraft 125-800A

The second test case used in testing the CSDRMA's accuracy was that of a Hawker Beechcraft 125-800A crash that occurred on July 31, 2008. East Coast Jets flight 81 attempted to land at Owatonna Degner Regional Airport, in Owatonna, Minnesota, but overran the runway and then attempted to regain altitude. In the process, the plane crashed about 2,000 feet from the end of the runway (National Transportation Safety Board, 2011).

Information from the National Transportation Safety Board (NTSB) report on the crash, and aircraft characteristics of a Hawker Beechcraft Hawker 800 (very similar to the Hawker Beechcraft 125-800A) were acquired and used as inputs into the geoprocessing script tool version of the CSDRMA. Table 10 details the crash variables specific to the event, and Table 11 shows the previously determined fixed values for a Hawker Beechcraft Hawker 800 crash event. Table 12 shows the non-spatial outputs generated by the CSDRMA and Figure 6-4 compares the Test Case II CSDRMA projected debris field with the known debris field.

Table 10. Test Case II: specific crash variable inputs.

Specific Crash Variable	Variable Value
Speed of aircraft	402 kts.
Altitude of aircraft	1,146 ft. AGL
Aircraft heading	130°
Angle of aircraft descent	0°
Ground level	1,145 ft. MSL
Ground level wind speed	17 kts.
Ground level wind direction	195°
Angle of terrain	0°
Terrain slope characteristic	None (flat)

Table 11. Test Case II: fixed values for Hawker Beechcraft Hawker 800.

Specific Fixed Value	Fixed Value
Frontal area of aircraft	353 sq. ft.
Drag coefficient of aircraft	0.02 C_d
Weight of aircraft	28,000 lbs.
Wingspan	44.5 ft.

Table 12. Test Case II: non-spatial data outputs.

Specific Non-Spatial Data Output	Output Value
Debris terminal velocity	1,106 kts.
Time to impact	0.02 sec.
Debris throw distance	169 ft.
Angle of impact	-0.68°
Speed of impact	462 kts.
Max altitude of thrown debris	11,450 ft.

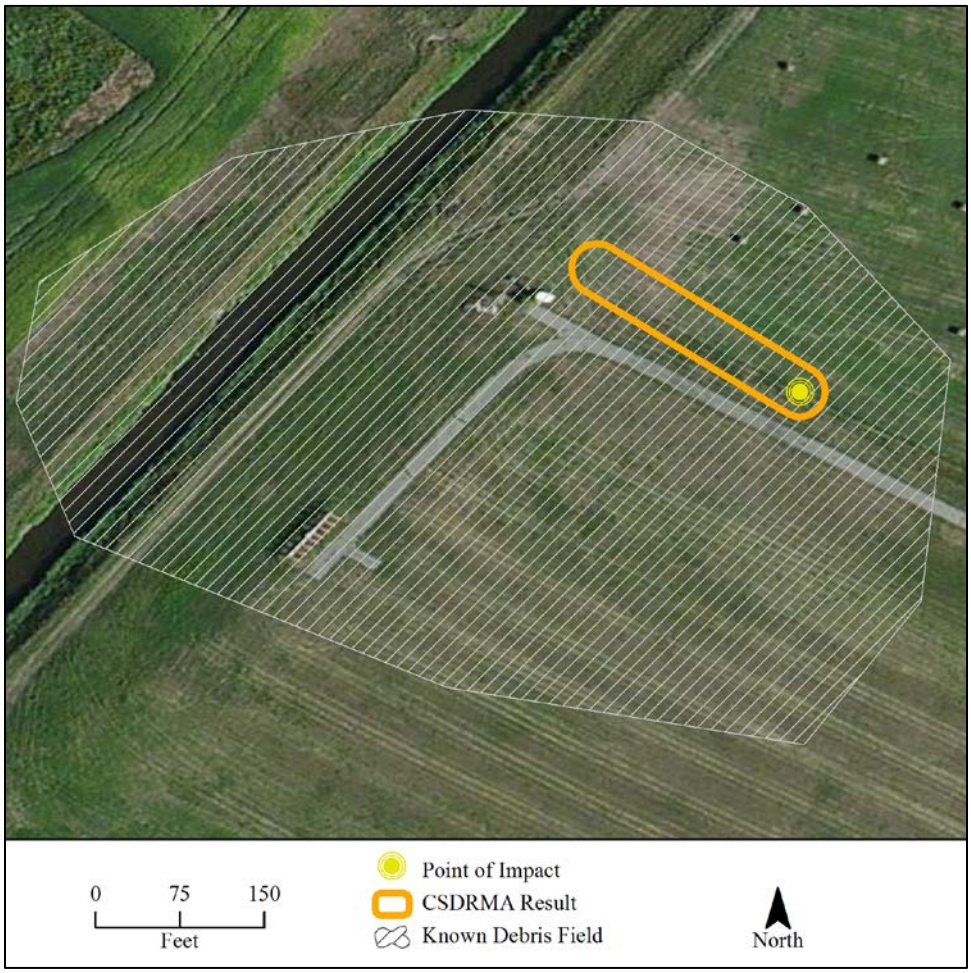

Figure 6-4: Test Case II CSDRMA projected debris field compared to the known debris field.

The Test Case II was unique in that the point of impact was not in the center of the debris field but rather more to the eastern side of the debris field. The results of this test case show that the CSDRMA did not take this factor into consideration and produced a projected debris field that covers only a small portion of the known debris field (Figure 6-4). The result would not have lead first responders, investigators, and clean-up crews to most of the debris, and as a result Test Case II was deemed unsuccessful.

6.4 Test Case III: Lockheed Martin F-22A Raptor

The third test case used in testing the CSDRMA's accuracy was that of a Lockheed Martin F-22A Raptor crash that occurred on November 16, 2010. The airplane, a United States Air Force (USAF) aircraft assigned to the 525th Fighter Squadron, 3rd Wing, Joint Base Elmendorf-Richardson (JBER), Alaska, experienced an equipment malfunction during a training mission. The pilot attempted to manage the malfunction but the plane crashed about 120 nautical miles from JBER on US public land (United States Air Force Aircraft Accident Investigation Board, 2011; Hart, 2011).

Information from the USAF Aircraft Accident Investigation Board report on the crash, and aircraft characteristics of a Lockheed Martin F-22A Raptor, were acquired and used as inputs for the geoprocessing script tool version of the CSDRMA. Table 13 details the crash variables specific to the event and Table 14 shows the previously determined fixed values for a Lockheed Martin F-22A Raptor crash event. Table 15 shows the non-spatial outputs generated by the CSDRMA and Figure 6-5 compares the Test Case III CSDRMA projected debris field with the known debris field.

Table 13. Test Case III: specific crash variable inputs.

Specific Crash Variable	Variable Value
Speed of aircraft	735 kts.
Altitude of aircraft	3,101 ft. AGL
Aircraft heading	315°
Angle of aircraft descent	48°
Ground level	3,100 ft. MSL
Ground level wind speed	0 kts.
Ground level wind direction	0°
Angle of terrain	0°
Terrain slope characteristic	None (flat)

Table 14. Test Case III: fixed values for Lockheed Martin F-22A Raptor.

Specific Fixed Value	Fixed Value
Frontal area of aircraft	840 sq. ft.
Drag coefficient of aircraft	0.02 C_d
Weight of aircraft	83,500 lbs.
Wingspan	44.5 ft.

Table 15. Test Case III: non-spatial data outputs.

Specific Non-Spatial Data Output	Output Value
Debris terminal velocity	1,289 kts.
Time to impact	53 sec.
Debris throw distance	3,8075 ft.
Angle of impact	-53.37°
Speed of impact	694 kts.
Max altitude of thrown debris	14,718 ft.

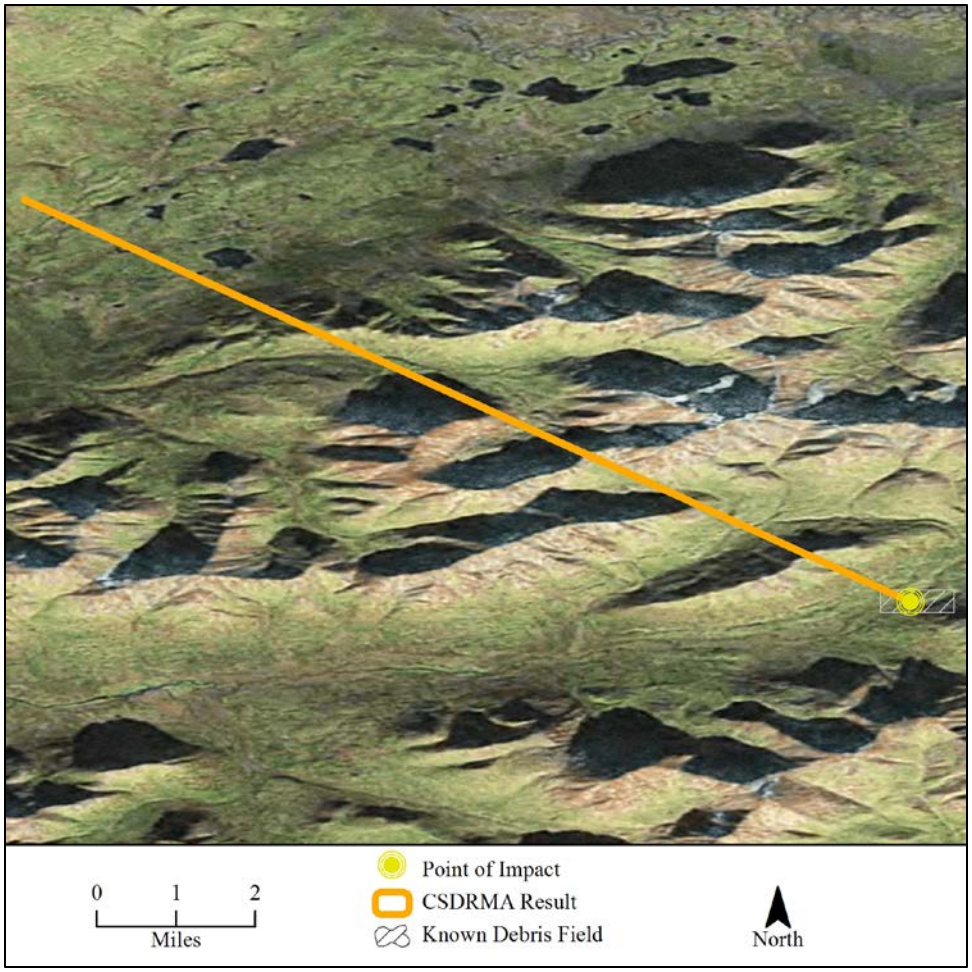

Figure 6-5: Test Case III CSDRMA projected debris field compared to the known debris field.

Because of the Lockheed Martin F-22 Raptor's high speed (735 kts., Mach 1.28) and steep aircraft descent angle (48°), a unique known debris field resulted that stretched only about 1,000 feet to the west and east and about 1,500 feet to the north and south of the point of impact. The CSDRMA was unable to produce a debris field that mimicked the known debris scattering, instead projecting a debris field that extended seven miles NW over mountainous terrain from the point of impact (Figure 6-5). Such a result occurred in part because of the CSDRMA's inability to take surrounding terrain into consideration. This result would not have lead first responders, investigators, and clean-up crews to most of the debris, and as a result, Test Case III was deemed unsuccessful.

6.5 Test Case IV: Piper PA-31 Navajo

The fourth test case used in testing the CSDRMA's accuracy was that of a Piper PA-31 Navajo crash that occurred on September 25, 1999. Big Island Air flight 58 was flying near the Mauna Loa volcano located on Hawaii Island, Hawaii, over cloud-covered mountainous terrain and crashed on the northwest slope of the volcano (National Transportation Safety Board, 2001b).

Information from the National Transportation Safety Board (NTSB) report on the crash, and aircraft characteristics of a Piper PA-31 Navajo, were acquired and used as inputs to the geoprocessing script tool version of the CSDRMA. Table 16 details the crash variables specific to the event and Table 17 shows the previously determined fixed values for a Piper PA-31 Navajo crash event. Table 18 shows the non-spatial outputs generated by the CSDRMA and Figure 6-11 compares the Test Case IV CSDRMA projected debris field with the known debris field.

Table 16. Test Case IV: specific crash variable inputs.

Specific Crash Variable	Variable Value
Speed of aircraft	735 kts.
Altitude of aircraft	3,101 ft. AGL
Aircraft heading	315°
Angle of aircraft descent	48°
Ground level	3,101 ft. MSL
Ground level wind speed	0 kts.
Ground level wind direction	0°
Angle of terrain	0°
Terrain slope characteristic	None (flat)

Table 17. Test Case IV: fixed values for Piper PA-31 Navajo.

Specific Fixed Value	Fixed Value
Frontal area of aircraft	840 sq. ft.
Drag coefficient of aircraft	0.02 C_d
Weight of aircraft	83,500 lbs.
Wingspan	44.5 ft.

Table 18. Test Case IV: non-spatial data outputs.

Specific Non-Spatial Data Output	Output Value
Debris terminal velocity	801 kts.
Time to impact	1 sec.
Debris throw distance	365 ft.
Angle of impact	-2°
Speed of impact	237 kts.
Max altitude of thrown debris	10,104 ft.

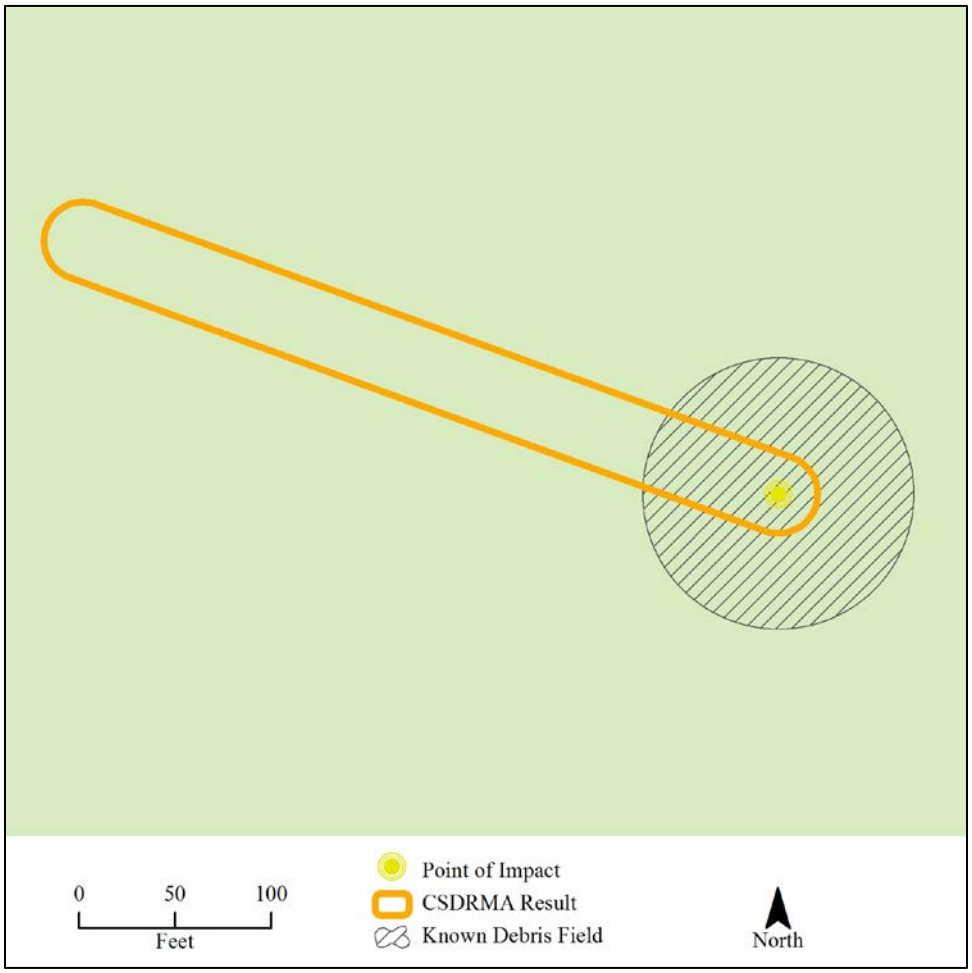

Figure 6-6: The Test Case IV CSDRMA projected debris field compared to the known debris field.

The results of Test Case IV show that the CSDRMA projected debris field covers only a portion of the known debris field (Figure 6-6). The results also suggest that debris could have travelled an additional 300 feet beyond the known debris field. Although the CSDRMA projected debris results don't cover the entire known debris field, only about 50 feet are left uncovered which could easily be spotted in the vegetation-free volcano slope environment of the Test Case IV by first responders, investigators, or clean-up crew members, and as a result Test Case IV was deemed successful.

6.6 Summary

This chapter reviewed the possibility of running the CSDRMA for Windows Phone as a Windows Phone application through a crash simulation and four test cases that determined the accuracy of the CSDRMA. The simulation and two out of the four accuracy test cases (Test Cases I and IV) proved successful.

Chapter 7 – Conclusions

Within this chapter, Section 7.1 reviews whether the CSDRMA project met the client's functional and non-functional requirements and gives a final conclusion of the project and Section 7.2 provides a description of possible areas within the CSDRMA project for future work.

7.1 Conclusion

The CSDRMA project can be deemed successful as it met all of the client's functional and non-functional requirements (see Section 3.2), and successfully proved that a mobile application can estimate an area of high debris concentration for use by first responders, crash investigators, and clean-up crew. Testing completed in this project revealed that the accuracy of the CSDRMA is not suitable for immediate deployment to airplane crash events. But with further research and development of the application and real-world testing, the CSDRMA could eventually become a very valuable tool in responding to, investigating, or cleaning-up debris after an aircraft crash event.

7.2 Future Work

Several areas of the CSDRMA project could benefit from more work and research. The first area is to incorporate a more appropriate debris model. The TAP is very simple and limited in the type of application to which it can be applied as it was originally designed for use in an airshow environment, in which debris is scattered from a mid-air collision or breakup. In these situations, debris may be thrown great distances if the airplane was climbing at a steep angle. Other debris models, such as those discussed in chapter 2 or one developed for ground-impact crash events, could be easily substituted in the application.

A second area is to develop a better way to utilize terrain data with the use of Digital Elevation Models (DEMs) or some other type of terrain data. As seen with Test Case III (Section 6.4), the CSDRMA was unable to take surrounding terrain into consideration when generating a projected debris field.

The third area of future work would be to include land cover data in the CSDRMA as crash characteristics can change in areas of varying vegetation and ground hardness.

Another area is to use the Universal Transverse Mercator (UTM) coordinate system within the CSDRMA in order to improve the spatial outputs.

The fifth area is to develop a way for first responders, crash investigators, and clean-up crews to mark and save debris locations within the CSDRMA. The results could be used in crash reports, conducting further analysis, or in communicating debris information with other involved parties.

The last area could be in developing the CSDRMA for Apple and Android mobile devices. This would allow the application to be utilized by more first responders, crash investigators, and clean-up crews.

Works Cited

ACTA. (n.d.). *Range Services*. Retrieved from ACTA: http://www.actainc.com/rangeservices.html

APT Research, Inc. (2012). *DEBRA*. Retrieved from APT Research, Inc.: http://www.apt-research.com/products/models/DebRA.html

Cooper, D. (2000). *The Application of Search Theory to Land Search: Adjustment of Probability of Area.* Retrieved from www.sarinfo.bc.ca/Library/Planning/AdjustmentofPOA.doc

Courtney, M., & Courtney, A. (n.d.). *The Truth About Ballistic Coefficients.* Retrieved from http://arxiv.org/ftp/arxiv/papers/0705/0705.0389.pdf

Defense Technical Information Center. (2012). ballistic trajectory. *DOD Dictionary of Military Terms*. Retrieved from http://www.dtic.mil/doctrine/jel/doddict/data/b/00611.html

Density altitude. (2012). *Wikipedia*. Retrieved from http://en.wikipedia.org/wiki/Density_altitude

Drag coefficient. (2012). Wikipedia. Retrieved from http://en.wikipedia.org/wiki/Drag_coefficient

Evans, A. (1996). The Air Accident Investigation Tool - Impact Tool and FDR Visualiser. *International Society of Air Safety Investigators 1996 Seminar Proceedings, Part I*, pp. 20 - 23.

Hart, J. (2011). *Environmental Assessment: Raptor Trail Access to F-22 Crash Site.* U.S. Department of the Interior, Bureau of Land Mangement (BLM).

Isolated Storage Overview for Windows Phone. (2012). *Microsoft Developer Network (MSDN)*. Retrieved from http://msdn.microsoft.com/en-us/library/ff402541(v=vs.92).aspx

Koester, R., Cooper, D., Frost, J., & Robe, R. (2004). *Sweep Width Estimation for Ground Search and Rescue.* Retrieved from www.uscg.mil/hq/cg5/cg534/nsarc/DetExpReport_2004_final_s.pdf

Launch Vehicle. (2012). *Encyclopædia Britannica*. Retrieved from http://www.britannica.com/EBchecked/topic/332323/launch-vehicle

Lin, M. Y., Larson, E. W., & Collins, J. D. (2003). *Columbia Accident Investigation Board Mishap Report, Volume II, Appendix D.16: Determination of Debris Risk to the Public Due to the Columbia Breakup During Reentry.* National Aeronautics and Space Administration (NASA). Retrieved from http://www.nasa.gov/columbia/caib/PDFS/VOL2/D16.PDF

NASA. (1999, March). Safety. *NASA-FAA Industry Roundtable.* Washington D.C. Retrieved from NASA-FAA Industry Roundtable: http://www.aeronautics.nasa.gov/events/showcase/safety.htm

National Transportation Safety Board. (2001a). *Aircraft Accident Report: Uncontrolled Descent and Collision With Terrain, United Airlines Flight 585, Boeing 737-200, N999UA, 4 Miles South of Colorado Springs, Municipal Airport, Colorado Springs, Colorado, March 3, 1991.* National Transportation Safety Board (NTSB), Washington D.C.

National Transportation Safety Board. (2001b). *Aircraft Accident Brief: DCA99MA088, Big Island Air flight 58, Piper PA-31-350, N411WL, near Volcano, Hawaii,*

September 25, 1999. National Transportation Safety Board (NTSB), Washington D.C.

National Transportation Safety Board. (2011). *Crash During Attempted Go-Around After Landing, East Coast Jets Flight 81, Hawker Beechcraft Corporation 125-800A, N818MV, Owatonna, Minnesota, July 31, 2008.* National Transportation Safety Board (NTSB), Washington D.C.

Oldham, H. E. (1990). *Aircraft Debris Trajectory Analysis*. Retrieved from Proairshow, LLC: http://proairshow.com/aircraft_debris.htm

Robledo, L. F. (2004). *Analysis and Integration of a Debris Model in the Virtual Range Project* (Doctoral dissertation, University of Central Florida). Retrieved from http://etd.fcla.edu/CF/CFE0000193/Robledo_Luis_F_200412_MS.pdf

Sala-Diakanda, S. N. (2007). *A Framework for the Assessment and Analysis of Multi-Hazards Induced Risk Resulting from Space Vehicles Operations* (Doctoral dissertation, University of Central Florida). Retrieved from http://etd.fcla.edu/CF/CFE0001606/SalaDiakanda_Serge_N_200705_PhD.pdf

Scott, J. (2004, July 11). *Drag Coefficient & Lifting Line Theory*. Retrieved from Aerospaceweb.org: http://www.aerospaceweb.org/question/aerodynamics/q0184.shtml

Soltis, S. J. (n.d.). *Overview of Usage of Crash Dynamic Anlytical Methods in Civil Aircraft Research and Certification Programs.* Federal Aviation Administration (FAA). Retrieved from http://www.fire.tc.faa.gov/2007conference/files/Crashworthiness/WedAM/Soltis Analysis/SoltisAnalysisPaper.pdf

Söylemez, E., & Usul, N. (n.d.). *Utility of GIS in Search and Rescue Operations Case Study.* Retrieved from http://proceedings.esri.com/library/userconf/proc06/papers/papers/pap_1908.pdf

United States Air Force Aircraft Accident Investigation Board. (2011). *United States Air Force Aircraft Accident Investigation Board Report: F-22A, T/N 06-4125.* United States Air Force (USAF), Aircraft Investigation Board.

Wood, R. H., & Sweginnis, R. W. (2006). *Aircraft Accident Investigation* (2nd ed.). Casper, Wyoming: Endeavor Books.

Appendix A. Sample Drag Coefficients

Table 19: Measure of drag coefficients for selected aircraft (Scott, 2004)**.**

Aircraft	C_d
Beechcraft Model 99	0.027
Boeing 747	0.0305
Cessna 172/182	0.027
Lockheed Martin F-104 Starfighter	0.048
Learjet 24	0.0216
Convair 880	0.024
Douglas DC-8	0.0188

Appendix B. CSDRMA Internal List of Aircraft Fixed Values

Table 20. CSDRMA internal list of airplane fixed values

Aircraft Type	Weight (lbs.)[a]	Wingspan (ft.)	Drag Coefficient (C_d)	Frontal Area (sq. ft.)[b]	Crusing Speed (kts.)
Aermacchi S-211	6,050	27.67	0.02	135.63	360[e]
Beechcraft Model 99	10,400	45.88	0.02	279.7	205
Boeing 747-200	833,000	195.66	0.03	1,098	481
Boeing 737-200	115,500	93	0.03[c]	1,098	421
Cessna 172 Skyhawk	2,450	36.08	0.02	174	122
Cessna 310	4,600	35	0.02	175	178
Convair 800	193,000	120	0.02	2,000	535
Douglas DC-8-32	310,000	142.42	0.01	2,771	511
Hawker Beechcraft Hawker 800	28,000	54.33	0.02	353[d]	402
Learjet 24	13,500	35.58	0.02	231.8	418
Lockheed F-104 Starfighter	13,500	35.58	0.04	196.1	1154[e]
Lockheed JetStar	6,050	27.667	0.01	840	438
Lockheed Martin F-22 Raptor	83,500	44.5	0.02[c]	840	1304[e]
McDonnell Douglas F-4 Phantom II	61,795	38.38	0.02	530	506
Piper PA-31 Navajo	6,500	40.67	0.02[c]	229	207
Ryan Navion	2,750	38.38	0.05	184	135

[a] Average weight unavailable, maximum take-off weight used
[b] Area of aircraft exposed to air flow, wing area used
[c] Value unavailable, estimated
[d] Value unavailable, value of similar "British Aerospace 125" aircraft used
[e] Cruising speed unavailable, maximum speed used

Appendix C. Original GW-BASIC Code of the Trajectory Analysis Program

The following is a copy of the original GW-BASIC code of the Trajectory Analysis Program (TAP) as developed by Hugh Oldham (1990) used in the development of the CSDRMA:

```
10:REM FULL TRAJETORY ANALYSIS PROGRAM FOR SCREEN DISPLAY
20:REM PROGRAM 1.10  8/20/90
30:REM FILE NAME "TRAJSCRN"
40:CLS
50:PRINT "TRAJECTORY ANALYSIS"
60:PRINT "FOR"
70:PRINT "AIRCRAFT DEBRIS"
80:PRINT " "
90:INPUT "INTIAL INDICATED AIR SPEED (KTS)";VEL
100:INPUT "INTIAL FLIGHT PATH ANGLE (DEG +/-)";ANGA
110:INPUT INTIAL FLIGHT PATH ALTITUDE (FEET AGL)";ALT
120:INPUT "INTIAL FLIGHT PATH DENSITY ALTITUDE IF DIFFERENT FROM
    INTIAL ALT ";DALT
130:IF DALT =O THEN DALT=ALT
140:PRINT "INTIAL DENSITY ALTITUDE ";DALT
150:INPUT "GROUND LEVEL (MSL FEET) ";GROUNDLEVEL
160:INPUT FLIGHT PATH COURSE MAG (DEG 001-360)";COURSE
170:IF COURSE <1 GOTO 160
180:IF COURSE >360 GOTO 160
190:INPUT "FRONTAL AREA OF DEBRIS (SQ FEET) ';FAREA
200:INPUT "DRAG COEFFICIENT OF DEBRIS (Cd)';CD
210:CDS=CD*FAREA
220:INPUT "WEIGHT OF DERBIS (LBS)";WT
230:INPUT "SURFACE WIND SPEED (KTS)";SWIND
240:INPUT "SURFACE WIND DIRECTION (DEG mAG 01 - 360)";DWIND
250:IF DWIND < 1 GOTO 240
```

```
260:IF DWIND > 360 GOTO 240
270:IF COURSE > DWIND THEN WINDC=COS(COURSE-DWIND)*SWIND
280:IF COURSE > DWIND THE WINDC=COS(DWIND-COURSE)*SWIND
290:IF SWIND >0 THE AWIND=WINDC+(ALT/30)^.26
300:IF SWIND=0 THE AWIND=SWIND
310:PRINT "HEAD WIND FACTOR AT FLIGHT PATH ALTITUDE ",AWIND
320:PRINT "HEAD WIND FACTOR AT SURFACE ",WINDC
330:PRINT "COMPUTE AIR MASS DENSITY AT ";DALT;" FEET MSL"
340:REM COMPUTE AIR MASS DENSITY IN SLUGS PER CUBIC FOOT
350:SLUGS=.002378*(1-(6.875*10^-6*ALT))^4.2561
360:GSLUGS=.002378*(1-(6.875*10^-6*GROUNDLEVEL))^4.2561
370:TVEL=(2*WT/(CDS*SLUGS))^.5
380:GLTEVL=(2*WT/(CDS*GSLUGS))^.5
390:PRINT "INTIAL TERMINAL VELOCITY (FPS) = ";TVEL
400:TVELKTS-TVEL*.5921052
410:GLTVELKTS=GLTVEL*.5921052
420:PRINT "INTIAL TERMINAL VELOCITY ";TVELKTS;" KTS"
430:TVELKTS=TVEL*.5921052
440:PRINT "GROUND LEVEL TERMINIAL VELOCITY ";GLTVELKTS," KTS"
450:PI=3.1416
460:TP=1!
470:DT=.05
480:WIND1=WINDC*6080/3600
490:T=0!
500:X=0!
510:Z=ALT
520:ANGCOR=PI/180!
530:DT2=DT*DT
540:ANG=ANGA*ANGCOR
550:REM CALCULATE TRUE AIRSPEED (FPS)
560:U=1.69*VEL*COS(ANG)
570:V=1.69*VEL*SIN(ANG)
580:PRINT " "
```

```
590:PRINT " "
600:PRINT " "
610:PRINT "  TIME     Z      Y     FPANGLE      KNOTS"
620:PRINT T,X,Z,ANGA,VEL
630:PRINT " "
640:W=WIND1*(Z1/30!)^.26
650:REM
660:REM CALULATE GROUND SPEED
670:UO=U-W
680:VO=V
690:REM CALCULATE DRAGE AND ACCELERATION
700:VEL2=U*U+V*V
710:IF U=0! THEN U=.01
720:FP=ATN(V/U)
730:FPANG=FP/ANFCOR
740:K=1!
750:IF U<0! AND V<0! THEN K=-1!
760:SLUGS=.002378*(1-(6.875-10^-6))^4.2561
770:DRAG=(SLUGS/2)*VEL2*CDS
780:AX=DRAG*COS(FP)*32.2*K/WT
790:AZ=-DRAG*SIN(FP)*32.2*K/WT-32.2
800:REM
810:REM CALCULATE VELOCITIES AND DISTANCES
820:UO=UO=AX=DT
830:V=V=AZ=DT
840:U-UO=W
850:VO=V
860:FPE=ATN(V0/UO)
870:FPANG=FPE/ANGCOR
880:IF UO<0! AND VO<0! THEN FPEANG=FPEANG-180!
890:X=X=UO*DT+.5*AX*DT2
900:Z=Z=VO*DT+.5AZ*DT2
910:Z1=Z
```

```
920:IF Z1<1! THEN Z1=1!
930:W=WIND1*(Z1/30!)^.26
940:T=T=DT
950:IF T<TP-.005 GOTO 980
960:PRINT CINT (T),X,Z,FPEANG,((UO*UO+VO*VO)^.5)*.592105
970:TP=TP=1
980:IFZ>GROUNDLEVEL GOTO 690
990:PRINT T,X,Z,FPEANG,((UO*UO=VO*VO)^.5)*.592105
1000:PRINT " "
1010:PRINT "DEBRIS TERMINAL VELOCITY ";GLVELKTS;" KTS"
1020:PRINT "TIME TO IMPACT     ";T;" SECONDS"
1030:PRINT "DEBRIS THROW DISTANCE    ";X;" FEET"
1040:PRINT "ANGLE OF IMPACT    ";FPEANG;" DEGREES"
1050:IMPACTA=(((UO*UO+VO*VO)^.5)*.68182)
1060:IMPACTB=(((UO*UO+VO*VO)^.5)*.592105)
1070:PRINT "SPEED AT IMPACT     ";IMPACTA;" MPH"
1080:BEEP
1090:INPUT "COMPUTE ANOTHER (Y/N) ";ANS$
1100:IF ANS$="Y" GOTO 10
1110:END
```

Appendix D. CSDRMA Geoprocessing Script Tool Python Code

The following is the CSDRMA Geoprocessing Script Tool Python Code used in testing the accuracy of the CSDRMA in Esri's ArcGIS:

```
#Crash Site Debris Recovery Mobile Application (CSDRMA) Geoprocessing Script Tool
#Trajectory Analysis for Aircraft Debris
#   for use as a tool script in ArcGIS 10.0
#   (requires feature class input of crash location)
#Nick Janzen
#MS GIS Program
#University of Redlands
#Redlands, California, USA
#July 2012
#Based on the "TAPS" program for GW-BASIC, Hugh Oldham, the
#  "Thrown Rotor Blade Trajectory Analysis" for Microsoft
#  Excel, T Watson (http://proairshow.com/aircraft_debris.htm),
#  and excerpts from the book "Aircraft Accident Investigation -
#  2nd Edition", Richard Wood & Robert Sweginnis
#

#Import Modules
import math, sys, arcpy, os

#ArcGIS Desktop Input
inputcrash = arcpy.GetParameterAsText(0)
workspace = arcpy.GetParameterAsText(1)
VEL = float (arcpy.GetParameterAsText(2))
ALT = float (arcpy.GetParameterAsText(3))
Course = float (arcpy.GetParameterAsText(4))
ANGA = float (arcpy.GetParameterAsText(5))
FAREA = float (arcpy.GetParameterAsText(6))
```

```
CD = float (arcpy.GetParameterAsText(7))
WT = float (arcpy.GetParameterAsText(8))
wingspan = float (arcpy.GetParameterAsText(9))
Groundlevel = float (arcpy.GetParameterAsText(10))
SWIND = float (arcpy.GetParameterAsText(11))
DWIND = float (arcpy.GetParameterAsText(12))
TANGA = float (arcpy.GetParameterAsText(13))
SLOPE = arcpy.GetParameterAsText(14)

#Input Checks

#Course Check I
if Course >= 1 and Course <= 360:
    Course = Course
else:
    arcpy.AddError("Error: Flight Path Course input must be 1-360")
    sys.exit()

#ANGA Check
if ANGA >= 0 and ANGA <= 90:
    ANGA = ANGA
else:
    arcpy.AddError("Error: Descent of Aircraft input must be 0-90")
    sys.exit()

#Course Check II
if Course < DWIND:
    WINDC = (Course - DWIND) * (math.pi/180) #convert to radians
    WINDC = math.cos(WINDC) * SWIND
else:
    WINDC = (DWIND-Course) * (math.pi/180) #convert to radians
    WINDC = math.cos(WINDC) * SWIND
```

```
#TANGA Check
if TANGA >= 0 and TANGA <=90:
    TANGA = TANGA
else:
    arcpy.AddError("Error: Angle of Terrain for Aircraft-Ground Impact input must be 1-
90")
    sys.exit()

#SLOPE Check
if SLOPE == "Upslope":
    ANGA = ANGA + TANGA
elif SLOPE == "Downslope":
    ANGA = ANGA - TANGA
elif SLOPE == "None":
    ANGA = ANGA
else:
    arcpy.AddError("Error: Aircraft-Ground Impact Characteristic input must be 'None',
'Upslope' or 'Downslope'")
    sys.exit()

#Preliminary Data Processing
wingspan = wingspan/2
CDS = CD * FAREA
SLUGS = 0.002378 * (1-(6.875**-6*ALT))**4.2561
GSLUGS = 0.002378 * (1-(6.875**-6*Groundlevel))**4.2561
TVEL = (2*WT/(CDS*SLUGS))**0.5
TVELKTS = TVEL * 0.5921052
GLTVEL = (2*WT/(CDS*GSLUGS))**0.5
GLTVELKTS = GLTVEL * 0.5921052
DT = 0.05
WINDone = abs(WINDC * 6080/3600)
T = 0
X = 0
```

```
Z = ALT
DTtwo = DT * DT
UVprep = ANGA * (math.pi/180) #convert to radians
U = 1.69 * VEL * math.cos(UVprep)
V = 1.69 * VEL * math.sin(UVprep)
W = WINDone * (Z/30) ** 0.26
UO = U
VO = V
Zmax = 0
FPEANG = 0

#Data Processing
while Z > Groundlevel:
    VELtwo = U*U+V*V
    if U == 0:
        U=.01
    else:
        U = U
    FP = math.atan(V/U)
    if U<0 and V<0:
        K = -1
    else:
        K = 1
    DRAG = (SLUGS/2)*VELtwo*CDS
    AX = -DRAG * math.cos(FP) * 32.2 * K/WT
    AZ = -DRAG * math.sin(FP) * 32.2/WT - 32.2
    UO = UO + AX * DT
    V = V + AZ * DT
    U = UO + W
    VO = V
    FPE = math.atan(VO/UO)
    X = X + UO * DT + 0.5 * AX * DTtwo
    Z = Z + VO * DT + 0.5 * AZ * DTtwo
```

```
    if Z > Groundlevel:
        T = T + DT
    else:
        T = T
    if abs(Z)>abs(Zmax):
        Zmax = Z  ,m * (180/math.pi) #convert to degrees
    else:
        Zmax = Zmax

#Output Processing
impactkts = (((UO*UO+VO*VO)**0.5)*0.68182)
FPE = FPE * (180/math.pi) #convert to degrees

#Spatial Processing
WKID = 4326 #WGS84
sr = arcpy.SpatialReference()
sr.factoryCode = WKID
arcpy.env.outputCoordinateSystem = sr
arcpy.env.workspace = "in_memory"
arcpy.CreateFeatureclass_management ("in_memory", "Bearing")
arcpy.MakeTableView_management ("Bearing","in_memory/BearingTable")
arcpy.AddField_management ("in_memory/BearingTable", "x_lon", "DOUBLE")
arcpy.AddField_management ("in_memory/BearingTable", "y_lat", "DOUBLE")
arcpy.AddField_management ("in_memory/BearingTable", "length", "DOUBLE")
arcpy.AddField_management ("in_memory/BearingTable", "bearing", "DOUBLE")
cur = arcpy.InsertCursor("in_memory/BearingTable")
row = cur.newRow()
arcpy.AddXY_management(inputcrash)
icrows = arcpy.SearchCursor(inputcrash)
for icrow in icrows:
    xlon = icrow.getValue("POINT_X")
    ylat = icrow.getValue("POINT_Y")
row.x_lon = xlon
```

```
row.y_lat = ylat
row.length = X
row.bearing = Course
cur.insertRow(row)
del cur, row
arcpy.BearingDistanceToLine_management("in_memory/BearingTable", "BearingLine",
"x_lon", "y_lat", "length","FEET", "bearing","", "","", sr)

#Spatial Outputs
buffer = arcpy.Buffer_analysis("BearingLine", workspace, str(wingspan) + " feet",
"FULL", "ROUND", "NONE")

#Output
arcpy.AddMessage("Debris Terminal Velocity (kts): " + str(TVELKTS))
arcpy.AddMessage("Time to Impact (sec): " + str(T))
arcpy.AddMessage("Debris Throw Distance (ft): " + str(X))
arcpy.AddMessage("Angle of Impact (deg): " + str(FPE))
arcpy.AddMessage("Speed of Impact (kts): " + str(impactkts))
arcpy.AddMessage("Max Altitude of Debris (ft): " + str(Zmax))
arcpy.AddMessage("Analysis Complete")
```