**Title of Project**

**Spatial Solutions**

****

**Patrick Sullivan-Project Manager**

**Sam Johnston-GIS Analyst**

**Andrew Goin-GIS Analyst**

**Tyler Armstrong-GIS Analyst**

**September 27, 2018**

**Table of Contents**

Introduction

 Summary…………………………………………………………………………………2

 Purpose…………………………………………………………………………………..2

 Scope……………………………………………………………………………………..2

Literature Review………………………………………………………………………………..2

 Importance of the Model Inventory of Roadway Elements…………………………..2

 Spatial Databases and Transportation Infrastructure………………………………..3

 Reducing Data Noise……………………………………………………………………4

 Automation of Data Extraction in GIS………………………………………………….4

Data & Methodology…………………………………………………………………………….5

Budget…………………………..………………………………………………………………..7

Time Table……………………………………………………………………………………….9

Expected Results………………………………………………………………………………12

Conclusion & Discussion………………………………………………………………………12

References..……………………………………………………………………………………13

1. **Introduction**

1.1 Summary

TxDOT (Texas Department of Transportation) is required to maintain records of the roads in Texas to provide other agencies with important information, receive funding, and data upkeep. The roads are expected to meet certain safety elements known as Model Inventory of Roadway Elements (MIRE). Of those elements, intersections and interchanges are two that need identifying among the other data. Right now, the problem is that there is no way to efficiently identify different intersections and interchanges, as well as attributes about those elements such as the angles and locations, without doing each intersection and interchange one by one. The implementation of a script to streamline this process would be ideal.

* 1. Purpose

Our group hopes to be able to design a script that fills the gaps in the MIRE attributes. A python script or model that can identify different intersection and interchanges would be substantial, but a script that could also find the angle of the intersections and provide unique identifiers is the ultimate goal. Once this script has been written, it is expected to be applied to other TxDOT maintained roads across all of Texas. Once these elements have been given the correct attributes, they can be shared with different agencies or among the state and federal government.

* 1. Scope

TxDOT provided us with data from all over Texas but asked us to limit it to one county from a list of 4 that they chose. Our group selected the Travis county as our study area. We will select our data from the main data set and focus on intersections and interchanges within. We chose this specific country because it is relatively close, so we can go and physically look at specific intersections if we need to, as well as some of our group members are familiar with the area.

1. **Literature Review**

2.1 Importance of the Model Inventory of Roadway Elements (MIRE)

This model was developed to standardize transportation network data structures concerning information varying from crash data to physical attributes of roadways so that it can be implemented at different levels of government without restricting the data usability (U.S. Department of Transportation 2010). This is all done with a goal of creating easily manageable data concerning the safety aspects of roadways. This is achieved by creating definitions and attributes that are uniform in nature so that they are more easily discerned by the analyst that will be using the data. MIRE has a total of 202 elements that are sectioned into three larger elements, roadway segments, roadway alignment, and roadway junctions (U.S. Department of Transportation 2010). For our purposes in this project we will only be focused on extracting elements having to do with roadway junctions, specifically intersections and interchanges.

2.2 Spatial Databases and Transportation Infrastructure

For our project and many like it, transportation network data is stored and analyzed with the help of a spatial database. The integration of spatial databases with the multiple software used in a GIS allow for more efficient processing of transportation network data, but more importantly it highlights the spatial relationship between these data points. Before GISs were implemented to manage transportation network datasets, data standards for the nation’s transportation networks were not uniform and did not implement a spatial attribute to manage the database (Thill 2000). This severely restricted the operability of the nation’s transportation network dataset, which inherently has spatial data associated with it. With this realization in the nineties, GIS based data management systems were implemented around the country to manage transportation network data with a focus on the spatial attributes associated with them (Thill 2000). This brought about new capabilities that previously were either impossible or so time consuming that they were not economically viable or necessary. The ability to create network models to represent the transportation infrastructure allowed for new safety data to be collected and furthermore allowed private companies and government agencies to plan routes and redirect traffic to create more efficient flow in the network (Thill 2000). Although our project does not deal directly with aspects of efficiency within the network, it does deal with extracting data from this network to further improve safety and meet the requirements set forth by the government. Without the implementation of a GIS to handle transportation data, this task would expend much more cost, inhibiting the growth and continued maintenance of the nation’s transportation infrastructure.

2.3 Reducing Data Noise

 Data noise in any dataset is a problem that analysts confront in almost every spatial analysis they perform. Our project is no exception, with a number of off-system roads that clog up the data-set with information that is not relevant to our end product. As illustrated in, “Automatic Urban Road Extraction from Digital Surface Model and Aerial Imagery”, reducing noise is paramount to providing an accurate analysis of whatever dataset you are using (Bethel et. al. 2006). The purpose of the article is focused on deriving polyline data from a raster image, which isn’t directly related to our project but their emphasis on reducing data noise is similar to the emphasis that we had to put on reducing noise. In Travis county there are more off-system roads than on-system, creating a lot of noise that we must eliminate before we can positively identify intersections of TXDot maintained roads. Similarly, in the article, they have objects in their raster image such as vehicles and parking lots that create a lot of noise and make the process of delineating and extracting roads much more difficult and provides a degree of in-accuracy to their analysis process (Bethel et. al. 2006). While we will not use a statistical process to eliminate our data noise, we will use attribute IDs to select and eliminate all off-system roads that will clog up our extraction process.

2.4 Automation of Data Extraction in a GIS

When processing large amounts of data in a GIS, time is an important variable. For many processes run by the mapping software, ours being extracting intersection and interchange information from a linear network layer, the actual task run by the machine is very repetitive and tedious. If done by hand, these tasks that are essential to any further analysis of the data would hinder the whole analytical process to extreme ends. This is why the utilization of the Python language in the mapping software has had such an impact on the implementation of GISs in various disciplines. In the paper, “New ArcGIS Tools Developed for Stream Network Extraction and Basin Delineations Using Python and JavaScript”, the authors show the reader how they automated what would usually be a time consuming and repetitive task through the use of coding languages (Abouelmagd et. al. 2016). Although stream extraction and basin delineation are not directly related to our project, the idea driving the need to automate this process is inherent in our task as well. While our study area is not nearly as large as the study area in the text, it is still quite sizeable, with a large amount of intersections that need to be processed and classified. Some of the processes involved in classifying these intersections and interchanges include determining the angles and amount of legs of these intersections and interchanges include. A process similar to this is illustrated nicely in the article, “A protocol to convert spatial polyline data to network formats and applications to world urban road networks”, where they superimpose a graph they generated onto the raster image of an intersection to assist them in defining the different nodes and edges the intersection has, creating topology among them (Karduni et. al. 2016). The use of Python alone would be sufficient to complete this project however, supplementing with Model Builder as the writers of the article did, will make it more user friendly for analysts throughout the country, providing a step by step illustration of what the Python script is doing (Abouelmagd et. al. 2016). Another step in the authors process of delineating watershed boundaries that could be applied to extracting intersections is the threshold value range of water flow to classify stream orders. While we would not be setting a continuous range of values, setting small separate ranges of degrees to classify intersection types with a Python script is an essential part of extracting and classifying intersections.

1. **Data & Methodology**

Our objective is to categorize different types of intersections in Texas. Therefore, we hypothesize the road data given to us from TxDOT can be categorized into different types of intersections. The data we will use in the project is a Texas road network shapefile and a Texas county shapefile provided by TxDOT. TxDOT told us the data is not always correct and we may need to clean up the data provided. Some programs needed for the TxDOT project are a python IDE and ArcGIS. We will use a combination of the model builder and python scripts. The algorithm that we are using is based from a forum (Bikash, 2012). Here is our current algorithm:

1. We will clip Travis county with the Texas road network shapefile.
2. The point intersect tool will be used to generate points where the road network lines collide.
3. Buffers will be generated where the intersect points are.
4. Streets will be clipped by the buffer.
5. The length of the street vectors will be measured using the line geometry tool.
6. Next, the field calculator will compute the angle between each line vector in the data set.
7. Using the 3 conditional statements through a python script, the program will use the number of lines and the angles measured to be recorded as either a T-intersection, a Y-Intersection, or a four-way intersection.
8. After categorizing the data, new points will be generated from the categorizes.

The algorithm outlined here will be used as our baseline algorithm. It is possible that any moment in time the algorithm is subject to change dramatically. For instance, we plan on excluding the buffer tool once we get a working model because of the amount of overhead produced by the processor. Furthermore, this algorithm does not address the missing roads. We will adapt along the way.

3.1 Flow chart

****

1. **Budget**

Table A-1. Project Budget.

|  |
| --- |
| Data Setup and Planning (Milestone 1) |
| Manager: | (2 hours/day \* 4 days) | 8 hours |
| GIS Analyst:  | (1 hours/day \* 4 days) \* 3 analyst  | 12 hours |
| Total Hours: |  | 20 hours |
| Hourly Pay: |  | $20 |
| Subtotal:  |  | $400 |
| Line Geometry (Milestone 2) |
| Manager: | (1 hours/day \* 7 days) | 7 |
| Analyst: | (1.5 hours/day \* 7 days) \* 3 analyst | 31.5 |
| Total Hours: |  | 38.5 |
| Hourly Pay: |  | $20 |
| Subtotal: |  | $770 |
| Identifying Real and False Intersections (Milestone 3) |
| Manager: | (1.5 hours/day \* 14 days) | 21 |
| Analyst | (1.5 hours/day \* 14 days) \* 3 analyst | 63 |
| Total Hours: |  | 84 |
| Hourly Pay: |  | $20 |
| Subtotal: |  | $1,680 |
| Grouping Intersections and Interchanges (Milestone 4) |
| Manager: | (1.5 hours/day \* 14 days) | 21 |
| Analyst: | (1.5 hours/day \* 14 days) \* 3 analyst | 63 |
| Total Hours: |  | 84 |
| Hourly Pay: |  | $20 |
| Subtotal: |  | $1,680 |
| Unique Identifiers and Verification (Milestone 5) |
| Manager: | (1.5 hours/day \* 3 days) | 4.5 |
| Analyst: | (1.5 hours/day \* 3 days) \* 3 analyst | 13.5 |
| Total Hours: |  | 18 |
| Hourly Pay: |  | 20 |
| Subtotal: |  | $360 |
| Optimization and Cleanup (Milestone 6) |
| Manager: | (1.5 hours/day \* 4 days) | 6 |
| Analyst: | (1.5 hours/day \* 4 days) \* 3 analyst | 18 |
| Total Hours: |  | 24 |
| Hourly Pay: |  | $20 |
| Subtotal: |  | $480 |
| Deliverables (Milestone 7) |
| Manager: | (1.5 hours/day \* 14 days) | 21 |
| Analyst: | (1.5 hours/day \* 14 days) \* 3 analyst | 63 |
| Total Hours: |  | 84 |
| Hourly Pay: |  | 20 |
| Subtotal: |  | $1,680 |
| Total: | : | **$7,050** |

1. **Time Table**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|

|  |
| --- |
| OCTOBER 2018 |
| Su | Mo | Tu | We | Th | Fr | Sa |
|  | 1 | 2 | 3 | 4 | **5** | 6 |
| 7 | 8 | 9 | 10 | 11 | **12** | 13 |
| 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 | **26** | 27 |
| 28 | 29 | 30 | 31 |  |  |  |

 |

|  |  |  |
| --- | --- | --- |
|  | October 1 | Project Presentation |
|  | October 2 | Data Setup, and Planning (Milestone 1) |
|  |  |
|  | October 5 | Deadline: Milestone 1 |
|  | October 6 | Line Geometry (Milestone 2) |
|  | October 12 | Deadline: Milestone 2 |
|  | October 13 | Identifying Real and False Intersections (Milestone 3) |
|  |  |
|  | October 26 | Deadline: Milestone 3 |
|  | October 27 | Grouping Intersections and Interchanges (Milestone 4) |
|  |  |

 |
|

|  |
| --- |
| NOVEMBER 2018 |
| Su | Mo | Tu | We | Th | Fr | Sa |
|  |  |  |  | 1 | 2 | 3 |
| 4 | 5 | 6 | 7 | 8 | **9** | 10 |
| 11 | 12 | 13 | 14 | 15 | **16** | 17 |
| 18 | 19 | 20 | 21 | 22 | 23 | 24 |
| 25 | 26 | 27 | 28 | 29 | **30** |  |

 |

|  |  |  |
| --- | --- | --- |
|  | November 9 | Deadline: Milestone 4 |
|  | November 10 | Begin: Unique Identifiers and Verification (Milestone 5) |
|  |  |
|  | November 12 | Deadline: Milestone 5 |
|  | November 13 | Begin: Optimization and Cleanup (Milestone 6) |
|  |  |
|  | November 16 | Deadline: Milestone 6 |
|  | November 17 | Begin: Deliverables (Milestone 7) |
|  |  |
|  | November 30 | Deadline: Milestone 7 |

 |

Milestone 1: Data Setup and Planning
All data needed to complete the project is provided by the client. There will be several team members working on derivatives of the same data. Special consideration for ensuring consistency, reliability and repeatability of results will be required. This period of planning will address these considerations by establishing best practices for ensuring data integrity and coordinating tasks to minimize conflicts when merging data.

 The ArcMap tools that will be used during the project will be decided on during this planning phase. The complexity of the project necessitates that the outputs of many tools be organized relationally in tables. These relationships will be defined initially during this phase and updated periodically as the project progresses. This milestone will be reached before October 5, 2018.

Milestone 2: Line Geometry

We will develop a method to calculate all angles between road centerlines of an intersect. The smallest of these angles will later be used for classification of the road intersection. We will then identify which road centerlines at an intersect are terminal or “passthrough.” This geometry information will later allow us to categorize each road intersection into geometry categories. This milestone will be reached before October 12, 2018.

Milestone 3: Identifying Real and False Intersections

ArcMap’s Intersect Tool creates point features at each place where two or more lines intersect. In the case of road centerlines, the tool does not consider physical realities such as tunnels and bridges.

A method to differentiate real and false line intersections, based on feature attributes and spatial location of features, will be developed to reach milestone 3. This milestone will be completed by October 26, 2018.

Milestone 4: Grouping Intersections and Interchanges

Many intersections and interchanges are comprised of multiple points of intersect. Using the intersects of cartographic lines (lines with attribute RTE\_RB\_NM ending in “-KG”) as a reference, points of road intersection and interchange in the immediate area can be grouped together by a unique identifier. This milestone should be reached by November 9, 2018.

Milestone 5: Unique Identifiers and Verification
 Each road intersection is required to have a unique identifier, along with a route and location descriptor. A lot of the work to needed to reach this milestone has been completed prior to starting on this milestone. This milestone will be reached by November 12, 2018.

Milestone 6: Optimization and Cleanup
 This project will depend heavily on tools in the Overlay, Proximity and Extract toolsets. The time these tools, particularly Clip, take to produce output increases exponentially according to the number of input features. This time can be minimized by reducing the input to these tools to features of interest only. For example, clipping only roads that intersect that are of interest to TxDOT is possibly faster than clipping every road that intersects.

All scripts and models used to complete this project are deliverable for the client. This milestone, while not necessary to the overall completion of the project, will increase reusability by the client. This milestone should be completed no later than November 16, 2018.

Milestone 7: Deliverables
The deliverables for this project are:
• Traditional county map showing the location of intersections and interchanges.

• ArcGIS Online (AGO) point feature service that is included on an AGO web map.
• Models and scripts used to complete the project
• Final report and presentation slides

This milestone will be reached when all deliverables are prepared and should be completed before November 30, 2018.

1. **Expected Results**

 Using the data provided by TxDOT, and more focused the Travis County area, our group will come up with a python script or a model that produces the desired attributes of the MIRE elements. After generating a script that is effective, we hope it can be implemented across all of Texas. This will be a huge time and money saver. Upon completion, we will deliver a map of the Travis County area intersections and interchanges and attributes pertaining to those, a point feature class containing all those locations, the script/model and the methods used.

1. **Conclusion and Discussion**

Our goal for this project is to make the streets of Texas a safer place. In order to accomplish this goal, we will categorize the Texas road network provide meaningful data to TxDOT. The categorized intersections will be used by TxDOT for MIRE regulations. Categorized data provided will be useful to MIRE because the database will allow government agencies to make faster and more coherent decisions.

The algorithm created will use a series of ArcGIS tools and a python selection script to categorize and produce a point database to solve TxDOT’s problem. We are starting small with a small test network, moving up to the county level, and then encompassing the entire state. Please refer to the methodology for further details.

Some issues involved with the current algorithm are data inconsistencies with the data provided and processor overhead with the resource heavy tools. The algorithm is not set in stone and will more than likely change throughout the course of the semester.

**8. References**

Abouelmagd, Abdou., Dietrich, Schroder., Michael, Marker., Omran, Abdel. New ArcGIS tools developed for stream network extraction and basin delineations using Python and javascript. *Computers & Geosciences* 94: 140-149. doi.10.1016

Altobello, Michael., DeLucia, Barbara Hilger., Lefler, Nancy., McGee, Hugh. Model Inventory of Roadway Elements. FHWA-SA-10-018, U.S. Department of Transportation Federal Highway Administration, Washington, DC.

Bethel, James., Junhee, Youn., Wonjo, Jung., Wonkook, Kim. Automatic Road Extraction From Digital Surface Model and Aerial Imagery. 47907-2051, School of Civil Engineering, Purdue University, West Lafayette, IN.

Bikash. 2012. Finding angle between intersecting features in two feature classes using ArcGIS Desktop and Python? *Geographic Information Systems Stack Exchange*. https://gis.stackexchange.com/questions/21038/finding-angle-between-intersecting-features-in-two-feature-classes-using-arcgis (last accessed 23 September 2018).

 Derrible, Sybil., Karduni, Alireza., Kermanshah, Amirhassan. A protocol to convert spatial polyline data to network formats and applications to world urban road networks. *Scientific Data* 3: 160046

 Thill, Jean-Claude. Geographic information systems for transportation in perspective. *Transportation Research Part C: Technologies* 8 (6):3-12. doi:10.1016/S0986-090X(00)00029-2