

Report Deliverable

Road Network Visualization - Phase 1

Authors:

Vincent Horosewski Jr.
Romeo Abraham
Shawn Allen
Dawn Marshall

Document ID: N2015-001

National Advanced Driving Simulator

2401 Oakdale Blvd.
Iowa City, IA 52242-5003
Fax (319) 335-4658

TABLE OF CONTENTS

Table of Contents	2
List of Figures.....	4
List of Tables	5
Executive Summary	6
1 Problem Statement.....	7
1.1 Objectives	7
1.1.1 Task 1 - State of the Art Review.....	7
1.1.2 Task 2 – Exploration of Problem Space	7
1.1.3 Task 3 – Report on Successes	8
2 State of the Art Review	9
2.1 Road vector data definition and data methods	9
2.2 Registration errors between road vector and raster datasets	9
2.3 Road vector data offsets.....	9
2.4 Geospatial data integration and extraction.....	10
2.5 Commercial products for raster-to-vector conversion	11
2.6 Improving road vector data offsets	12
3 Image processing Techniques.....	13
3.1 Introduction.....	13
3.1.1 Imagery concerns	13
3.1.2 Terminology.....	13
3.2 Color-based analysis	13
3.3 Statistical analysis.....	14
3.3.1 Clustering.....	14
3.3.2 Road space transformation.....	18
3.3.3 Shape detection	19
3.4 Artificial Intelligence-based approaches.....	20
3.4.1 Neural Networks and Deep Learning	20
3.5 Sample data.....	21
3.5.1 Original imagery	21
3.5.2 RGB space scatter plots	23
3.5.3 Segmentation by color similarity	26
3.5.4 Saturation and lightness correction.....	27
3.5.5 Segmentation by clustering.....	29

3.5.6	Road space matching	31
4	Using Existing GIS Tools.....	35
4.1	Steps or processes to automate aerial imagery	35
4.1.1	Prepare imagery for data extraction using ArcScan	35
4.1.2	Raster cleanup.....	36
4.1.3	Vector extraction.....	36
5	Summary	39
5.1	Proposed Next Step – Artificial Intelligence-based Techniques	39
6	References	40
7	Appendices	42
7.1	GNU Image Manipulation Program (GIMP) procedures	42
7.1.1	Select roadway	42
7.2	MATLAB code	42
7.2.1	Colorspace plot	42
7.2.2	Lighting correction and contrast enhancement	42
7.2.3	Cluster by color.....	43
7.2.4	Remove stray points from processed image	44

LIST OF FIGURES

Figure 1: (a) Original imagery, (b) Scatter plot in RGB space	16
Figure 2: (a) Enhanced imagery, (b) Scatter plot in RGB space.....	17
Figure 3: (a) Sample imagery, (b) Extracted road pixels.....	19
Figure 4: Sample raster images. (a) Sparse road density, (b) Low road density, (c) Moderate road density, (d) High road density.....	23
Figure 5: RGB space scatter plots. (a) Sparse road density, (b) Low road density, (c) Moderate road density, (d) High road density.....	25
Figure 6: Segmentation by color similarity. (a) Sparse road density, (b) Low road density, (c) Moderate road density, (d) High road density	27
Figure 7: Saturation and lightness correction. (a) Sparse road density, (b) Low road density, (c) Moderate road density, (d) High road density	29
Figure 8: Segmentation by clustering. (a) Sparse road density, (b) Low road density, (c) Moderate road density, (d) High road density	31
Figure 9: Road space matching. (a) Sparse road density, (b) Low road density, (c) Moderate road density, (d) High road density.....	33

LIST OF TABLES

Table 1: Advantages and disadvantages of the two approaches	35
---	----

EXECUTIVE SUMMARY

A review of the state of the art was conducted to understand the problem space and published methods of addressing key issues. The effort provided an understanding of registration errors between road vector and raster datasets, road vector data offsets, geospatial integration and extraction. Specifically, methods for improving road vector data offset were investigated. Additionally, commercial products for raster-to-vector conversion were considered.

Based on this review, two methods were explored, a method based on existing GIS tools and one based on image processing techniques.

Existing GIS tools can successfully extract road vector data from images of arbitrary complexity. However, this analysis requires significant manual refinement of intermediate results in order to obtain good results.

Image processing techniques can be leveraged to automate much of the process of road vector extraction. Obtaining good results, however, can be challenging on more complicated imagery. Additionally, complex analyses can require minutes per image, although this does not require human oversight. Exploration of this method indicates that some means of differentiating between roads and similar-looking features must be incorporated to fully automate the process of road vector extraction.

While presented separately, these two methods can be viewed as complementary approaches. That is, image processing techniques can be applied to imagery initially, and the results obtained can be plugged into GIS tools for refinement and additional processing. In this way, time spent on manually cleaning up the images will be reduced.

Proposed Next Step – Artificial Intelligence-based Techniques

Based on these results, we propose the investigation of artificial intelligence-based techniques as a next step in the exploration of automatically extracting data. Principally, this would involve the application of deep learning algorithms to the problem of identifying roads within images. The expected result would be the segmentation of a set of sample images that is sufficiently accurate to extract road centerlines without the need for any human intervention. Additionally, the question of divining road connectivity and traffic flow through the extracted data must be addressed. We expect these to be second-order difficulties, however. A statement of work and proposal would be developed in consultation with Rockwell personnel to ensure efforts align with their needs.

A rough order of magnitude for the proposed next step is \$75K -\$100K over the course of six to nine months. Based on current staffing allocations, we could begin this effort as soon as July 2015.

1 PROBLEM STATEMENT

This project seeks to define a better depiction of a road network than is currently available in commonly available vector datasets. Road networks consist of path data. Road path data commonly suffers from two significant flaws that must be addressed to function in data-centric simulation, where data from multiple sources (elevation, imagery and vector) are combined to produce the simulated scene rendering.

The first flaw is that vector datasets do not contain logical integrity for the road path network they describe. This can be in the form of overlapping vectors where a junction should be defined, or elevation differences such as bridges and overpasses. In order for simulated traffic to appear realistic during simulation, road paths must be logically connected. Typically road data contains no information about junction connections.

The second flaw is registration errors usually exist between vector and image datasets. This can be caused by several factors including overall data quality, projection and coordinate systems, different data sources and incomplete coverage.

An ideal solution would use existing data resources and correct for incomplete or invalid road networks, and align the road vectors with imagery that has been designated as the “ground truth” baseline for the combined dataset. This proposed solution would be required to generate an intermediate vector dataset that modifies the original vectors to satisfy error tolerance margins, and is used in the simulation workflow in place of the originating vector data.

1.1 Objectives

1.1.1 Task 1 - State of the Art Review

The goal of this task was to examine existing road network data as an alternative to modifying general vector data that is not already network data. Existing road network data (costs, availabilities and license agreements) was compared to creating and purchasing capabilities in order to modify general vector data into road networks. We examined whether existing road network data have the intersection definitions and attributes that are needed for real-time simulation rendering or required additional processing. The outcomes are this task was a report on the difference in cost and capabilities of: 1) purchasing new road network data (and modifying as necessary); and 2) defining a method of improving general vector data from non-road network data into network data.

1.1.2 Task 2 – Exploration of Problem Space

The goal was to employ vector adjustments to align properly with imagery data using methods identified in Task 1. This was explored through the development of algorithms to successfully generate logical connectivity and register output to image data. Where no satisfactory results were achieved, we identified any other information or methods needed to successfully determine logical connectivity.

1.1.3 Task 3 – Report on Successes

This report provides an overview on the level of success achieved and recommendations for implementation strategies and future tasks. Included is a description of the methods and algorithms employed, how they were successful in increasing proper alignment in imagery data, and recommendation for the next steps in the process based on issues encountered.

2 STATE OF THE ART REVIEW

2.1 Road vector data definition and data methods

“Road networks consist of nodes and road edges” (Yang et al, 2010). Due to its relative expensive nature, road network data collections are normally undertaken sporadically (Stefan et al, 2010). One major problem that content providers face is being certain of the quality of information when updating their current databases to a higher level of accuracy (Gautama and Borghgraef, nd). Aerial imagery is considered one of the standard data sources for the acquisition of topographic objects such as roads for use in Geographic Information System (Zhang et al, 2003). The significance of road extraction using aerial images is due to its ability to enhance the efficiency of map generation and revision; hence it can be very important in car navigation systems or emergency rescue system that requires instant or current maps (Zhang et al, 2003).

Producing road vector data that have a road-name attribute from raster maps is difficult because maps normally contain overlapping layers of geographic features such as road, contour lines and text labels. Hence, the map content is complex and presents challenges in converting the road geometry in raster maps to vector formats. Also, there is the additional problem of recognizing text labels of varied sizes using classic optical character recognition (OCR) techniques. After the vectorization of road geometry and text label recognition, challenges exist in associating individual road lines with the recognized labels (Chiang & Knoblock, 2012).

2.2 Registration errors between road vector and raster datasets

A fundamental issue for image registration is spatial pattern recognition in the road network, network matching, object extraction from imagery and image matching (Yang et al, 2010). “Traditional pixel based correspondence techniques are inadequate to solve this registration problem” (Brown, 1992). Feature based registration is the suggested approach to compare data sources that have different types of representations, in this instance comparing images to vector data (Gautama and Borghgraef, nd).

Usually, geo-referencing of the raster dataset using existing spatial data (such as a vector feature class) resides in the desired map coordinate system. The process entails locating a series of control points – known x, y coordinates – to correlate locations on the raster dataset with locations in the spatially referenced vector data, (Esri, 2008). “The geo-referenced data is only as accurate as the data to which it is aligned” (Esri, 2008).

2.3 Road vector data offsets

According to a study conducted by CIESIN (2013), there are five road network errors.

- ❖ Gaps in road network can result from unconnected segments.
- ❖ Road segments that lucidly represent roads on the ground but do not have attribute data and are not connected to other roads in the datasets.
- ❖ Randomly appearing short road segments far removed from the network without attribute data.

- ❖ Unconnected arcs without visual confirmation of its position or link to other roads in Google Earth but with attribute information.
- ❖ Visual inconsistency of the road's shape when comparing the dataset with Google Earth.

Road databases are assumed to include systematic and randomly distributed errors; the automated node detection method and the delineation routine are expected to take care of these types of discrepancies (Klang, 1998).

The extraction of road networks from imagery is likely to generate errors and may need manual intervention (Chen et al, 2006). This makes automation of the generation of road vector from raster data challenging. The main issue to address is differences between data representations. This can be in the form of different coordinate space or projections. Comparative analysis of geospatial vector data with images requires the information in the images to be described in object features terms (Gautama and Borghgraef, nd). The suggested system for change detection relies on feature based spatial registration, where detected features in the image are registered to corresponding features in the vector data.

2.4 Geospatial data integration and extraction

Geospatial data integration has become an important subject in GIS (Chen et al, 2006). As of 2005, despite much research on road extraction from a diversity of image sources, little research had been done on the fusion of different information sources for road extraction (Xiaoying & Davis, 2005). This has changed in recent years. LiDAR enables 3D surveying of the real world (Kumar et al, 2013). Xiong and Sperling (2004) proposed the amalgamation of road networks from different sources based on the patterns found in the road edges and nodes.

Tanaka and Kamimura (1993) presented a method of vectorization using the energy minimization principle. Their method entails thinning, line segment approximation and geometric transformation. After the line segment approximation stage, geometric transformations are applied to reform the skeleton vector obtained. The skeleton vector is a vector representation of the skeleton line. A cost function, that is used to quantify distortion levels which would be minimized if the distortion was reduced, is used in the process (Tanaka and Kamimura, 1993). Gold and Dakowicz (nd) showed that a point Voronoi diagram, coupled with the extraction of crust and skeleton information, is valuable in GIS. According to them, the labelled skeleton and Voronoi skeleton are useful in the extraction of features from digitized, scanned (raster) or point (vector) data.

The extraction of road vector information from raster maps is difficult since raster maps normally contain noise and roads often overlap with other features (Chiang et al, 2009). A thinning operator is capable of generating distorted lines around intersections, thus leading to inaccurate extracted road vector data if there is the absence of human adjustment (Chiang et al, 2009). The thinning operator is a process used to identify road centerlines.

Mena (2003), suggested the extraction of road from satellite imagery based on cues using color, texture pattern and monochromatic intensity. Vectorization algorithms based on skeleton extraction and graph theory, applying geometrical and topological adjustments

was presented as the way to create automated raster to vector conversion (Mena, 2006). This is a laudable submission since it improves the quality of the final output. Furthermore, for the generation of correct road vector network to be topologically accurate, mathematical morphology was suggested instead of non-conditional geometric adjustment (Mena, 2006). Mathematical morphology is a theory and technique for the analysis and processing of geometric structure based on set theory, lattice theory, topology and random functions (Wikipedia, 2015).

The vectorization algorithms proposed by Mena (2006) include: a raster cleaning stage, pixel classification and edge smoothing, skeleton extraction, graph generation and objects building, geometrical adjustment and topological adjustment.

The main advantages of automated road extraction are its uniform approach to different images, execution of processing operations in a short time and the capacity to extract roads not detected by human eyes (Gecen and Sarp, 2008).

Gecen and Sarp, (2008) used the line module of Geometica software in the automated extraction of a road in a study in Turkey based on four satellite images: SPOT, IKONOS, QUICKBIRD and ASTER. According to Gecen and Sarp (2008), there are basically four steps involved in the automated extraction of roads: roads sharpening, roads finding, roads drawing and relating extracted roads to each other.

Poullis (2014) presented an algorithm for road extraction from satellite images using tensor-cuts. This algorithm makes use of the combination of strengths of tensor encoding, global optimization using graph-cuts and feature extraction using Gabor jets. The Gabor filter is a linear filter used for edge detection. This algorithm is unsupervised and demands no thresholds. The framework requires no thresholds because all information is encoded using “tensorial representation and optimized using Graph-cuts”. This framework is suitable for remote sensing applications and road extraction using pre-processing of satellite images because they contain mainly linear features.

ArcScan provides a platform reputed to perform automatic or interactive raster-to-vector data conversion with high accuracy and little or no operator intervention during the data capture stage (Esri ArcGIS Online Resources, 2013).

2.5 Commercial products for raster-to-vector conversion

There are several products available with unique strengths and limitations. The focus will be on products that have the capability for road centerline vectorization. Vextractor and Raster-to-Vector products have this functionality. These two products employ line approximation algorithms that produce vector lines based on the locations of the pixels inside road areas. However, the line approximation algorithms fail to take into account the geometry and topology of the road network, and thus produce road vector data that are off-center (Chiang and Knoblock, 2011). R2V is an automated raster-to-vector conversion product from Able Software specializing in the digitizing of raster maps. R2V is designed for raster-to-vector conversion from maps and it is capable of handling map specific linear features, including curved roads and contour lines. According to Chiang and Knoblock (2011), in order to vectorize roads in raster maps using R2V software, the user must first of all provide labels of road colors or select one set of color thresholds to

identify the road pixels. The manual part of providing road pixels with only labels can be tedious and time-consuming, when using scanned maps with numerous pixels. When road pixels are extracted, R2V software can automatically trace the centerlines of the extracted road pixels and produce the road vector data. However, the centerline tracing function of R2V reacts to the road width if there is no manual pre-processing (Chiang and Knoblock, 2011).

2.6 Improving road vector data offsets

Usually, road databases are stored in a vector environment. GIS tools are preferable to detect and extract features of interest for updating (Klang, 1998). According to Fiset et al (1998) a map-guided approach is used to update the road network.

An algorithm for improving 2D road vector data using Airborne Laser-Scanning was put forward by Gopfert et al (2011), dependent on snakes (parametric active contours) and combines many concepts that concern internal energy, which maintains the original shape of the vector data through the expansion by terms and concepts of a ribbon.

Gautama and Borghgraef (nd) proposed a system for automated detection of change consisting of a low level feature detection process that extracts the junctions of road and a high level matching process that makes use of graph matching to find correspondences between the detected image information and the road vector data. This system relies on feature based spatial registration, where the identified features in the image are registered to corresponding features in the vector data.

The alignment of a road network to an image with higher accuracy will result in the improvement of its relatively poorer positional accuracy (Chen et al, 2006).

3 IMAGE PROCESSING TECHNIQUES

3.1 Introduction

3.1.1 Imagery concerns

The quality of any result obtained is highly dependent on the quality and resolution of the image being analyzed. While different image enhancement techniques exist, there is a fundamental limit to their effectiveness, namely the amount of information contained in the image. The image resolution must be sufficient to indicate the presence of roads strongly. Typically this means a resolution of one meter per pixel or finer. Additionally, the image content factors into the problem, as well. A busy complex scene, as one might find in an urban environment, defies simple analysis.

Orthographically projected imagery tends to make analysis easier, especially over larger regions. Although transformations can be applied to adjust the image projection, these require information about the viewing angle which may not be available after the fact.

3.1.2 Terminology

Splitting an image into discrete regions of interest that correlate to objects and features represents the first step towards automatically obtaining road centerlines. In the image processing field, this process is known as segmentation. Once segmented, it is often useful to apply meaningful labels to the various regions, a process known as classification. These represent the two most difficult aspects of the larger problem of automatically tracing a road within an image. Many methods exist to tackle these problems; here we will investigate color-based, statistics-based, and artificial intelligence-based approaches.

3.2 Color-based analysis

Color-based analysis can be used as a first attempt at segmentation and classification. Segmentation, in this approach, relies on the premise that adjacent objects frequently have different colors, so any color discontinuities between adjacent pixels may represent the boundary between objects. Classification can be achieved by making some assumptions about the colors of commonly seen objects. In overhead road images, there are typically about seven different colors represented. Regardless of whether they are asphalt or concrete, roads are generally a fairly pure shade of grey. Fields and forests will be green. Bare earth will typically be some shade of brown, and so on.

Segmentation and classification via color analysis generally fails with greyscale images, because greyscale only records one dimension of visual information, namely brightness. As such, there is no way to discriminate between equally lit surfaces, even when they represent completely disparate objects, e.g., road and trees. Having said that, the method can be used in very simple cases where the road stands out strongly against its environment, as with a desert scene.

Color imagery provides three dimensions of visual information, namely red, green, and blue intensities. This allows for the possibility of discriminating between similarly lit objects which were confounded in a greyscale image, so long as they have different hues.

In color-based analysis, segmentation can be considered an effort to reassign the colors of individual pixels based on a much reduced palette, e.g., all road pixels are grey, all house pixels are blue, all grass pixels are green, and so on. Anything that can be done to reduce the local variability of pixel colors increases the efficiency of the process, so long as care is taken to avoid erasing necessary detail.

Differences between adjacent pixels can be quite random due to atmospheric effects, lens aberration, normal discontinuities in the scene itself, and the finite resolution of the recording equipment. These differences manifest in the raster as “noise”. It is therefore advisable to use one of several methods to “de-noise” the image. Simply replacing pixels with the average of all pixels in its neighborhood can serve this purpose. The well-known Gaussian blur filter augments this approach by weighting adjacent pixel contributions by a factor derived from a 2D Gaussian distribution.

Wavelet decomposition appears to offer additional benefits over and above that of more traditional de-noising techniques. This technique applies a wavelet transform to generate a series of detail images that reject information that is above or below the current level being resolved. Removing the detail images that represent the finest details performs a conventional blurring operation. However, the detail images may also be edited rather than removed. This allows one to selectively remove noise or features in one region while preserving the rest of the image. Also, by editing the coarser detail images, wavelet decomposition can be used to correct for uneven lighting, as when a cloud crosses the sun.

The main drawbacks of color-based analysis are that the results are not particularly good in urban areas and that manual editing is frequently required due to differences in lighting, exposure, and atmospheric haze among individual pictures. An additional complication is the presence of large industrial or commercial buildings. These tend to be roofed with materials that are similar in hue to those used in roads in the sample imagery investigated.

3.3 Statistical analysis

These methods carry the advantage of being more amenable to automation. However, there still remains some amount of iteration and parameter tweaking.

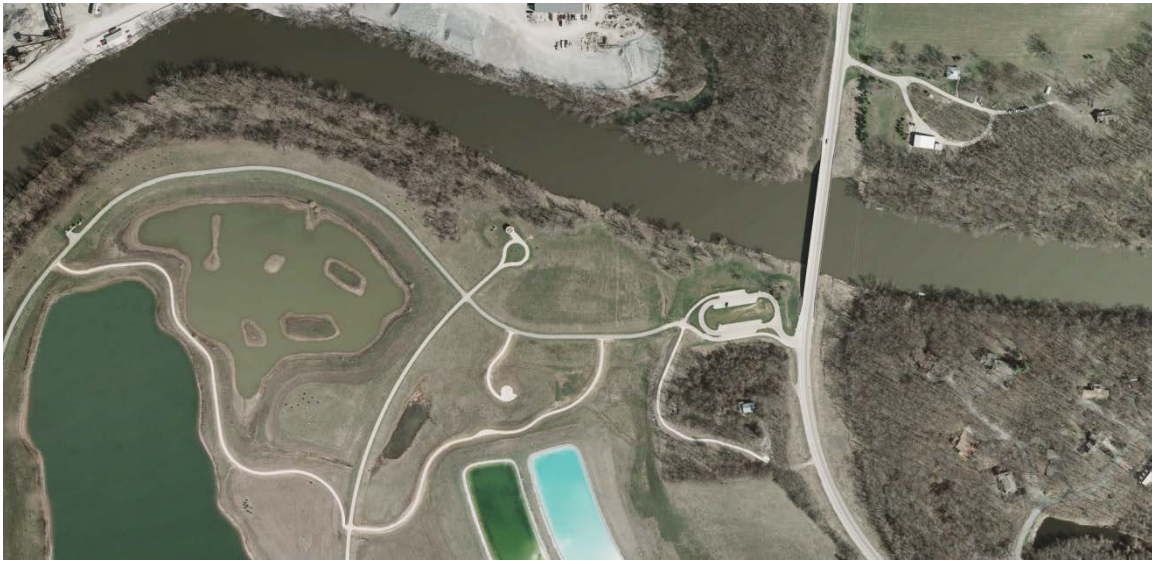
3.3.1 Clustering

Clustering refers the process of analyzing a dataset to find distinct or semi-distinct groupings within the data. Many clustering algorithms exist, but generally they define clusters according to one of three qualities: proximity to a cluster center, local point density, or agreement with a statistical distribution. Additionally, certain algorithms require that the number of clusters be specified beforehand. Performance varies according to the characteristics of the data, and what is the best algorithm for one application may not be the best for another. Here, the k-means algorithm was used, due to the fact that it finds wide support in statistical packages.

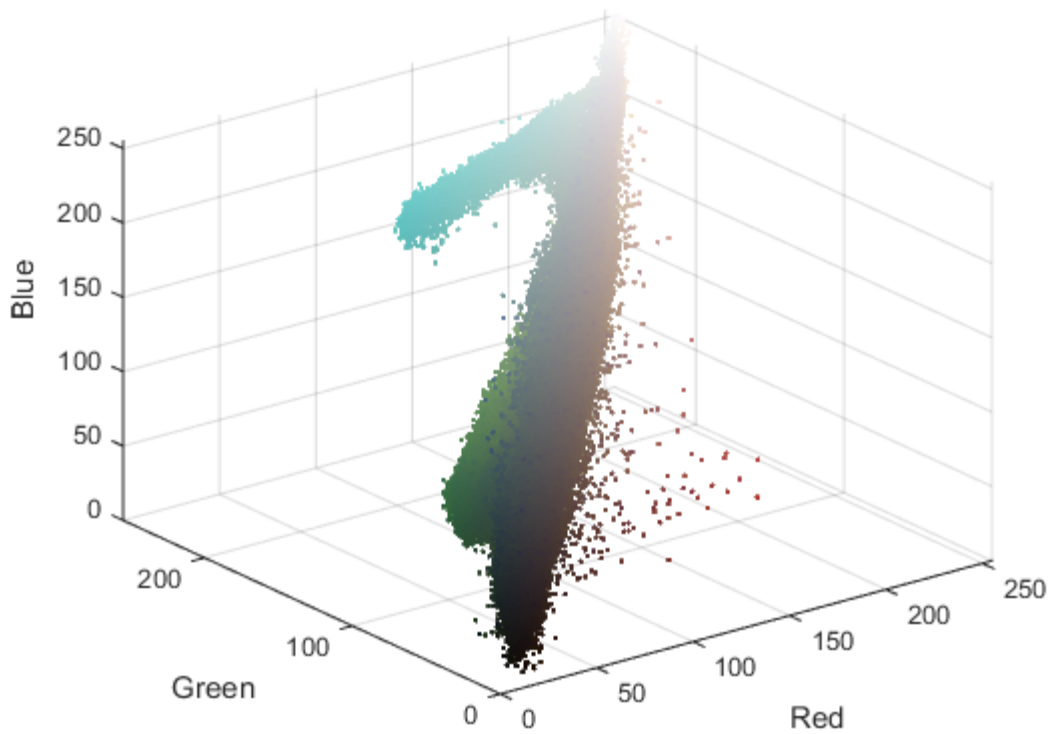
During this project, the clustering operation was done in color space, which places data in terms of its RGB values regardless of its original location within the image. In some ways, this application of clustering resembles an automated color-based approach, as

described previously. However, it is possible to augment this procedure by clustering the data in four or more dimensions, with the additional dimensions representing the location of the pixel in the original image or the connectedness of a pixel (within the original image) to other members of the cluster. This sort of approach can tax the clustering algorithm heavily, however, especially for large images. It may be possible to run the clustering in two steps, first by color, then by location or connectedness.

As k-means is a proximity-based algorithm, some pre-processing of the data was warranted. This is due to the fact that aerial imagery tends to be “washed out”, especially those in our sample set, which appear to have been taken in the fall. To see this effect, consider the images below.



(a)



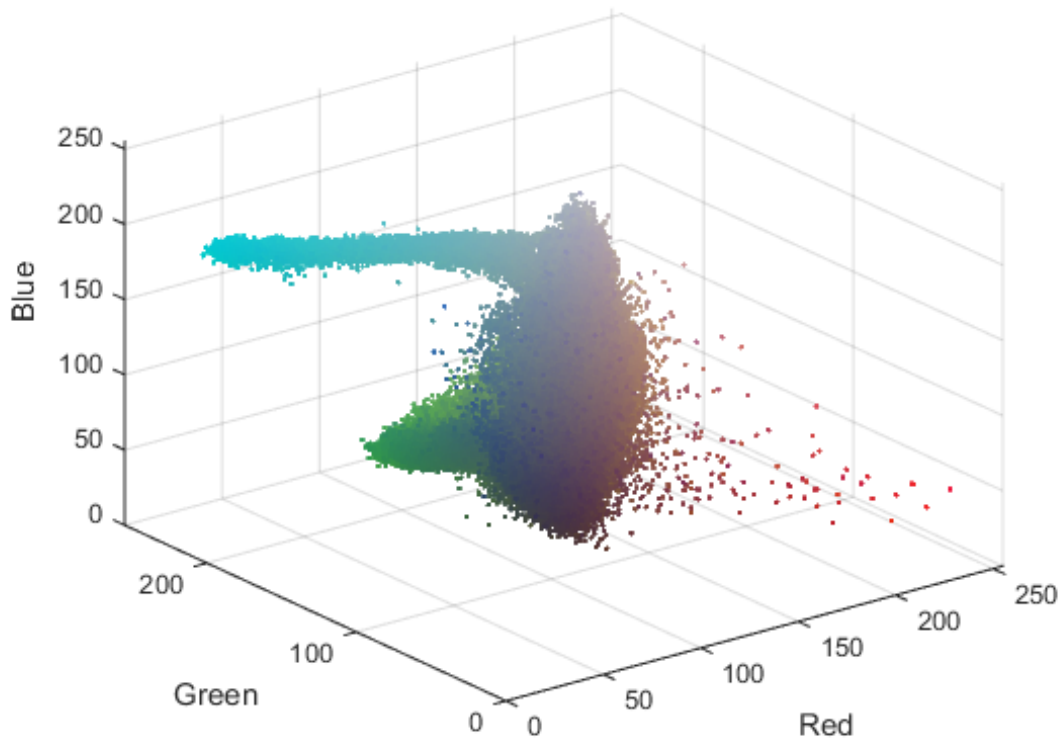
(b)

Figure 1: (a) Original imagery, (b) Scatter plot in RGB space

The uppermost image above represents a 3D scatter plot of individual pixels within the image below it. There are two promising clusters, but the larger part of the distinctiveness within the image falls along the line which represents shades of grey, due to shadows and highlights. This will make clustering difficult. To help remedy this problem, we can attempt to “spread out” the colors while reducing the extremes of lighting. In other words, we increase the overall color saturation while moving the lightness to the mean, as shown in the scatter plot below. The cyan and green clusters are now fully evident and blue and orange clusters starts to stand out as well.



(a)



(b)

Figure 2: (a) Enhanced imagery, (b) Scatter plot in RGB space

Typically, the cluster closest to grey represents the road. Also, the frequency of a particular color can indicate its classification. In urban and residential areas, the largest cluster generally represents the road. In rural areas, the reverse is true.

For algorithms that require a cluster count, some iteration is largely inevitable. Typically, using four clusters makes the best starting point. If after clustering finishes the cluster containing the roads also contains other features, more clusters are required. If the roads are spread between two or more clusters, decrease the cluster count.

3.3.2 Road space transformation

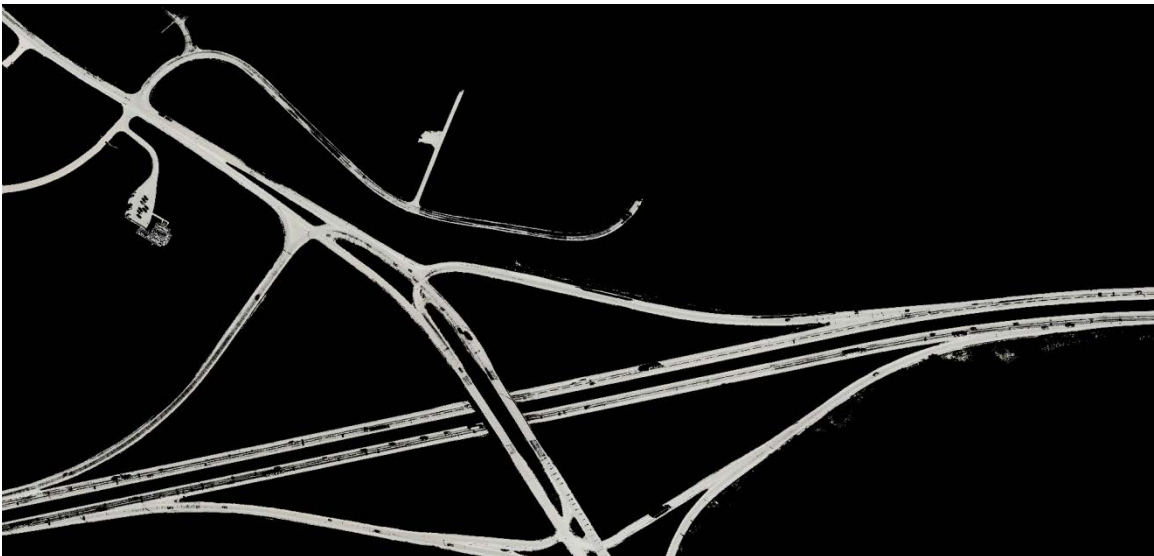
Another approach investigated relied on the use of principal component analysis to determine a pixel's membership in the road network. Principal components are defined as the eigenvectors of the covariance matrix of a dataset. These eigenvectors comprise a coordinate system that has some useful properties. Among them, the first principal component defines an axis along which the maximal change occurs in the data. More loosely, it represents that quality which is most indicative about the data.

The idea, then, is to consider only road pixels from a representative image or set of images. Principal component analysis is run on this data, and the resultant vectors define what we might call "road space." Once a candidate pixel is transformed into this coordinate system, its deviation from the primary axis can be used to decide how closely it conforms to the example road pixels.

The images below depict our sample road image and those pixels chosen for our representative set. Note that these road regions remain the same color as in the original image.



(a)



(b)

Figure 3: (a) Sample imagery, (b) Extracted road pixels

This approach has the advantage of requiring minimal effort after the initial road sample has been created, and the results are generally favorable, producing clearly identifiable road forms.

3.3.3 Shape detection

Investigating the applicability of this approach was infeasible due to time constraints, but it warrants brief mention nonetheless. One of the defining characteristics of roadways is a generally consistent shape, specifically rectangular on straight stretches or parabolic around curves. The Hough transform algorithm can detect simple shapes within an image.

Typically, lines and circles are the shapes being investigated, but other primitive shapes and simple arrangements of them are also possible.

The first step of this algorithm involves the use of an edge-detection filter, the most widely used being the Canny algorithm. This creates a black and white image of all the outlines of objects within the original image. Then, the Hough transform scans the boundary points and determines, for example, all the potential lines that this point could fall on. The number of lines is bound by some arbitrary resolution parameter. For each of these lines, a running tally is incremented. When all boundary points have been evaluated, those lines representing the highest tallies are considered to exist in the image.

The number of running tallies represents the limiting factor for this algorithm. It depends on the resolution desired (finer resolutions yield closer matches) and on the dimensionality of the shape being processed. Each dimension represents a degree of freedom for the shape in question. A circle can be fully determined by a center point and a radius. Therefore, the Hough transform operates in three dimensions. Lines, on the other hand, can be represented using only two measures: the angle and length of a vector radiating from the center and normal to it. Slope and intercept parameters are not used as the slope tends to infinity for vertical lines, making for very poor resolution near this extreme.

3.4 Artificial Intelligence-based approaches

One failing common to the above approaches is an inability to distinguish between roads and rooftops. In order to improve the performance in the discrimination and classification of roads in the case of close matches, some form of decision-making based in artificial intelligence techniques is wanted.

A series of simple observations about the qualities associated with a questionable region of pixels can help to clarify the situation. For instance, roads by their very nature are contiguous and proportionately longer than they are wide. Similarly, roads typically feature some lane markings on them. Rooftops tend to be oblong and localized. These facts can be accumulated and formalized using various techniques, but it can be difficult to obtain a comprehensive solution.

3.4.1 Neural Networks and Deep Learning

The field of neural networks has experienced something of a revival over the last ten years around the notion of “deep learning.” Deep learning builds on the legacy of previous research into neural networks but significantly improves on traditional techniques. This has allowed deep learning systems to tackle novel applications with increased success.

All along, pattern recognition has been one of the strengths of neural networks, and this is also true with deep learning techniques. Their performance in the segmentation of medical imagery has been surprisingly good, with some systems claiming “super-human” ability. Given this, the idea of using a deep learning system to identify roads in imagery seems a promising one.

Neural networks differ fundamentally from most other artificial intelligence techniques in that they can be adapted to respond in new ways to situations that were not present at

their creation. In some crude sense, they are able to learn. Therefore, you never have to “program” a neural network; rather you assemble one and then train it. This means that neural networks offer a significant advantage to end-users, namely they do not have to know anything about how the neural network operates. By the same token, designers of neural networks do not have to know anything about the phenomenon they are investigating. All that is required is that they share a data set that represents the intricacies of the phenomenon sufficiently well. This generally means many thousand to several million data points.

The basic unit of any neural network is the so-called perceptron, which is the artificial analog to a neuron. It is comprised of a number of input “sensors” which funnel data into a central accumulator. The inputs can be amplified or diminished independently. The accumulator then triggers when the aggregate input exceeds a threshold. So, the amounts of amplification assigned to the inputs and the activation threshold define how the perceptron works. Training, then, is the process of assigning appropriate values to the weights associated with each perceptron input.

Historically, neural networks have been trained using the back-propagation technique. Broadly speaking, one parcel of inputs is applied to the network inputs and then the resulting output is compared to the expected value. The difference is then used to revise the weights and thresholds within the perceptrons constituting the network. This is done in such a way that the largest changes are applied to those units most responsible for the erroneous output.

While this approach has yielded a number of successful applications, there are significant drawbacks. Training requires some manual intervention to indicate what the proper response should be. Furthermore, the training algorithm is not particularly amenable to parallelization which means that it can take a prohibitively long time to train complex neural networks on large data sets.

Deep learning carries the advantage that training can be done in an unsupervised manner. That is, the training set does not have to be “pre-digested” before being fed into the neural network. Data is simply fed into the network which then identifies features and relationships within the data. Also, training can be done in a piecemeal fashion, which permits large-scale parallelization.

While investigating these approaches was beyond the scope of this project, they represent the best hope for a comprehensive solution to automating roadway extraction from raster data.

3.5 Sample data

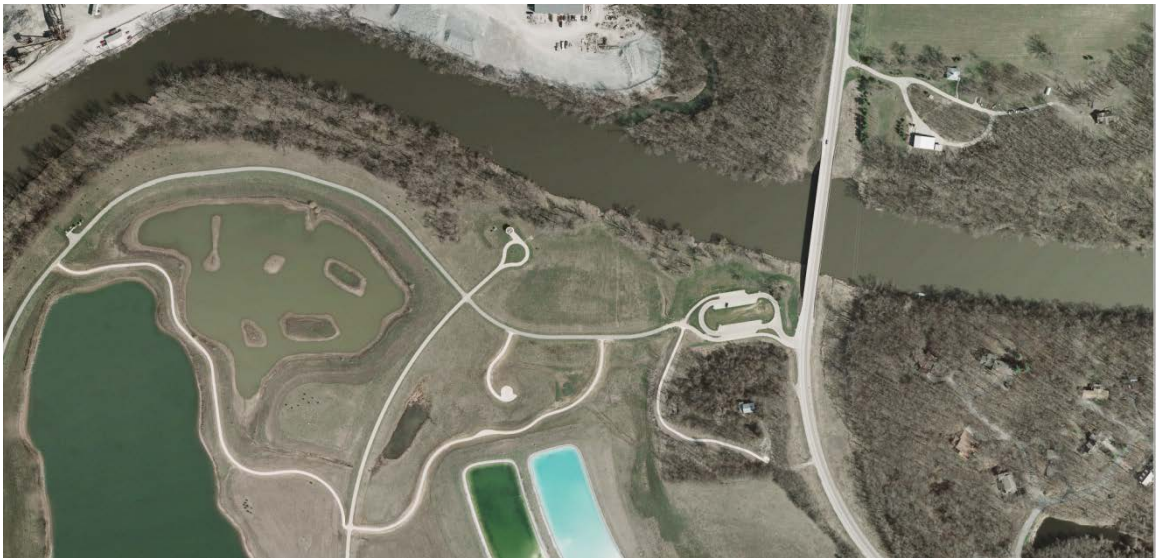
The following images depict a set of sample images and the results obtained from the methods described in [Image processing](#).

3.5.1 Original imagery

The sample images were chosen to reflect a spectrum of road network complexities. This complexity tends to dictate the difficulty of analysis.



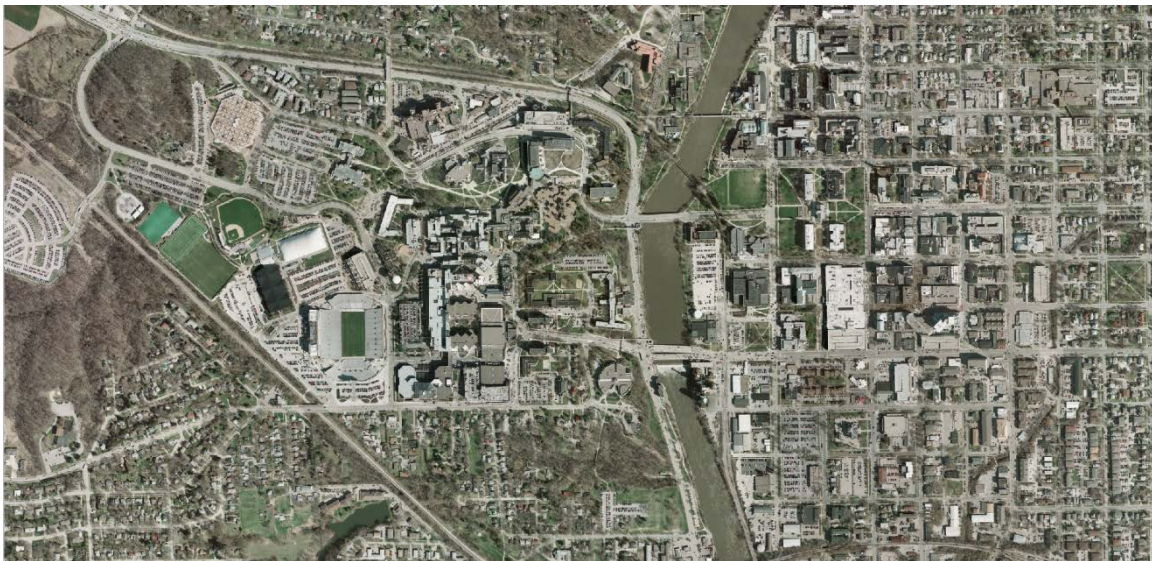
(a)



(b)



(c)

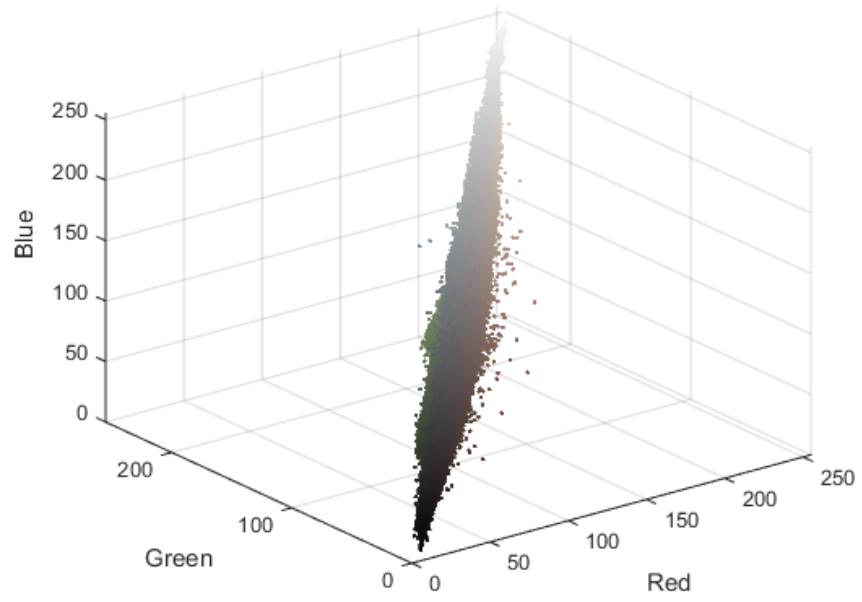


(d)

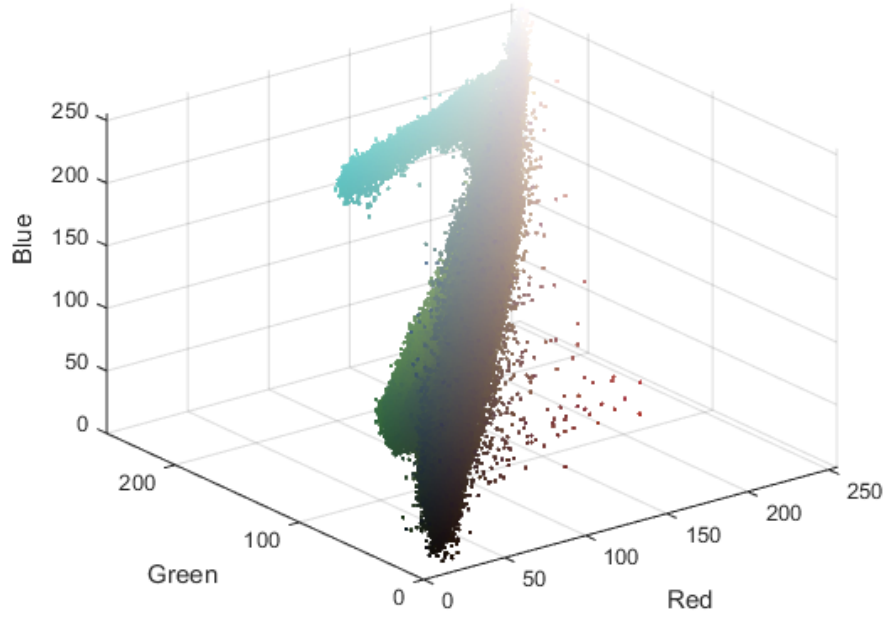
Figure 4: Sample raster images. (a) Sparse road density, (b) Low road density, (c) Moderate road density, (d) High road density

3.5.2 RGB space scatter plots

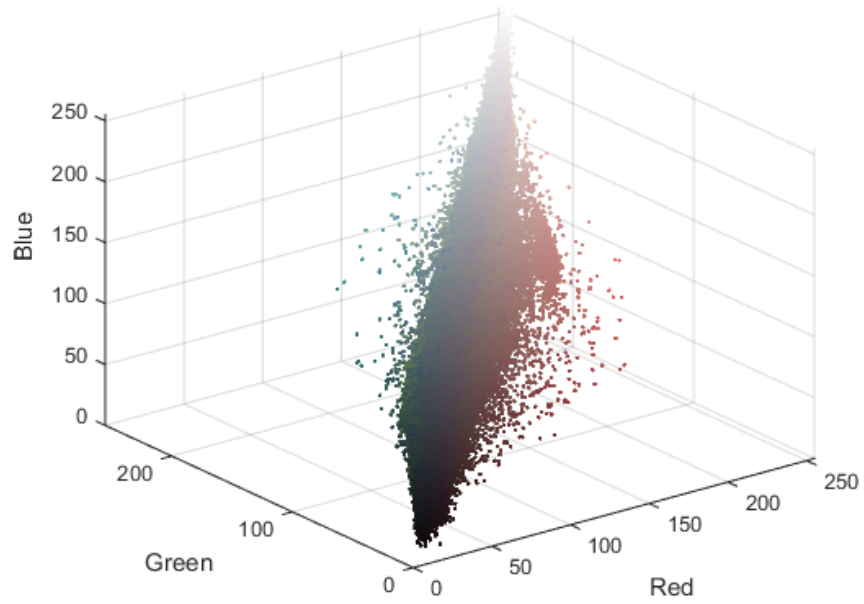
These scatter plots were formed by plotting each pixel of an image in three-dimensional space, where the axes reflect the red, green, and blue intensity of that pixel. For convenience, each pixel is plotted in its actual color. The



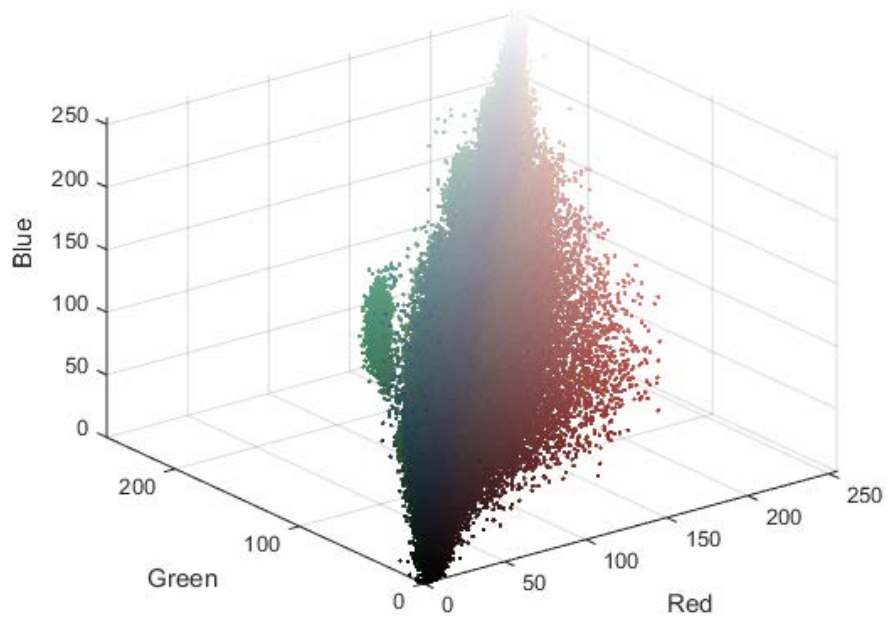
(a)



(b)



(c)

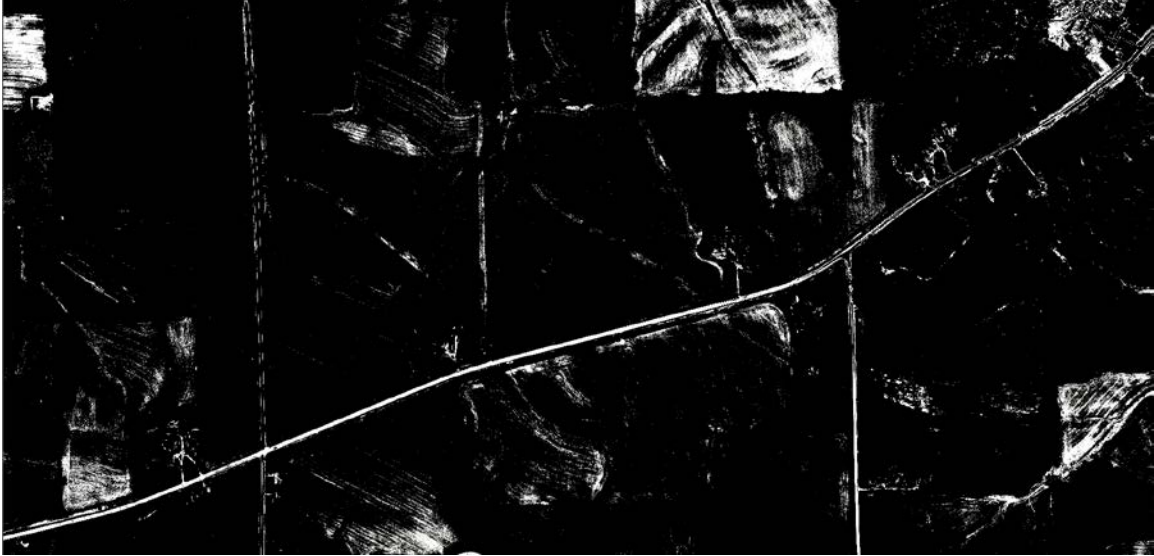


(d)

Figure 5: RGB space scatter plots. (a) Sparse road density, (b) Low road density, (c) Moderate road density, (d) High road density

3.5.3 Segmentation by color similarity

These images were using the procedure described in [GNU Image Manipulation Program \(GIMP\) procedures](#). They represent the performance of the simplest approach investigated and should be considered the baseline. Sharp road boundaries are present, but many false positives exist.



(a)



(b)



(c)



(d)

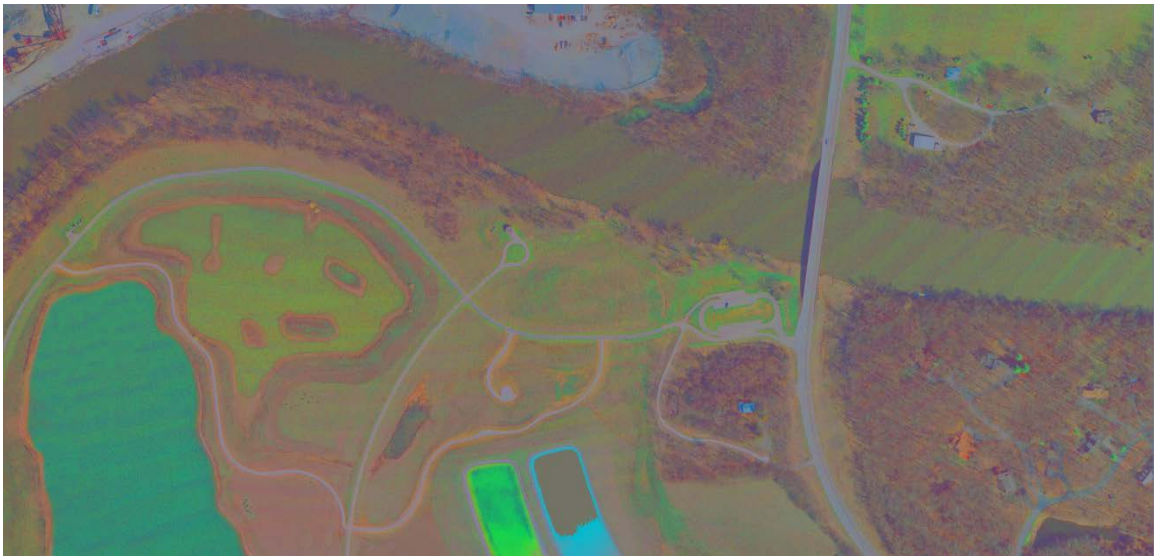
Figure 6: Segmentation by color similarity. (a) Sparse road density, (b) Low road density, (c) Moderate road density, (d) High road density

3.5.4 Saturation and lightness correction

These images show the original imagery after enhancement for clustering purposes. Here, the purpose is to differentiate regions while preserving essential color information.



(a)



(b)



(c)

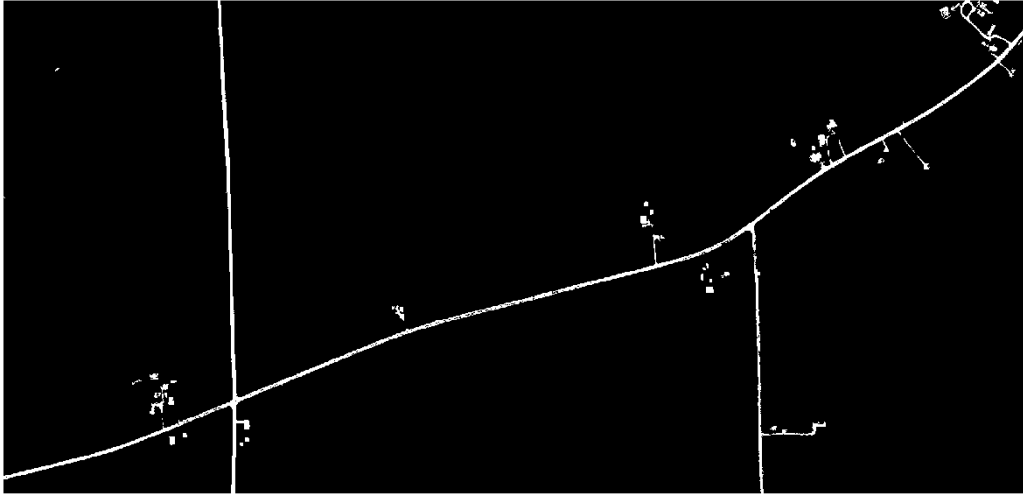


(d)

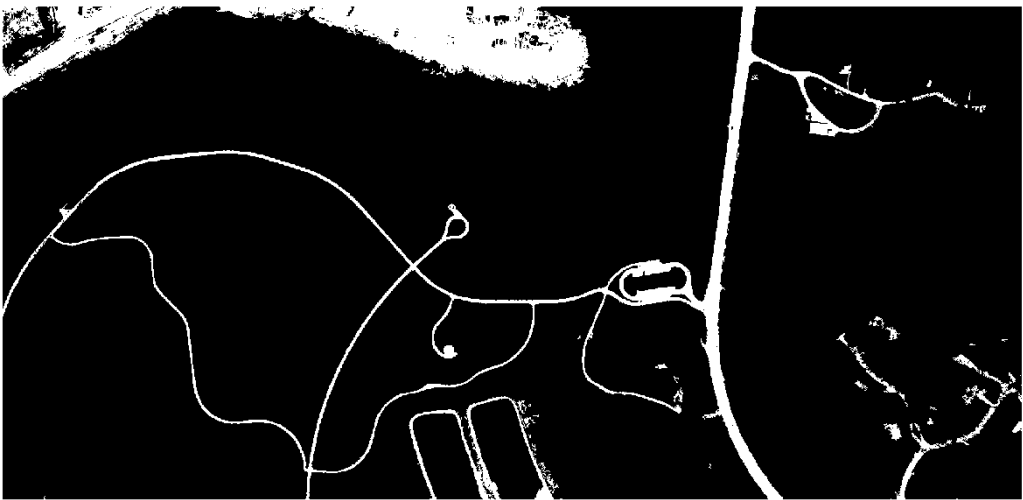
Figure 7: Saturation and lightness correction. (a) Sparse road density, (b) Low road density, (c) Moderate road density, (d) High road density

3.5.5 Segmentation by clustering

These results indicate the roads identified using the k-means clustering algorithm in RGB space on enhanced images. Typically, they show better detection of minor, outlying features and some small gains in the rejection of near misses. Segmentation by clustering performs better than segmentation by color similarity on disconnected road segments.



(a)



(b)



(c)

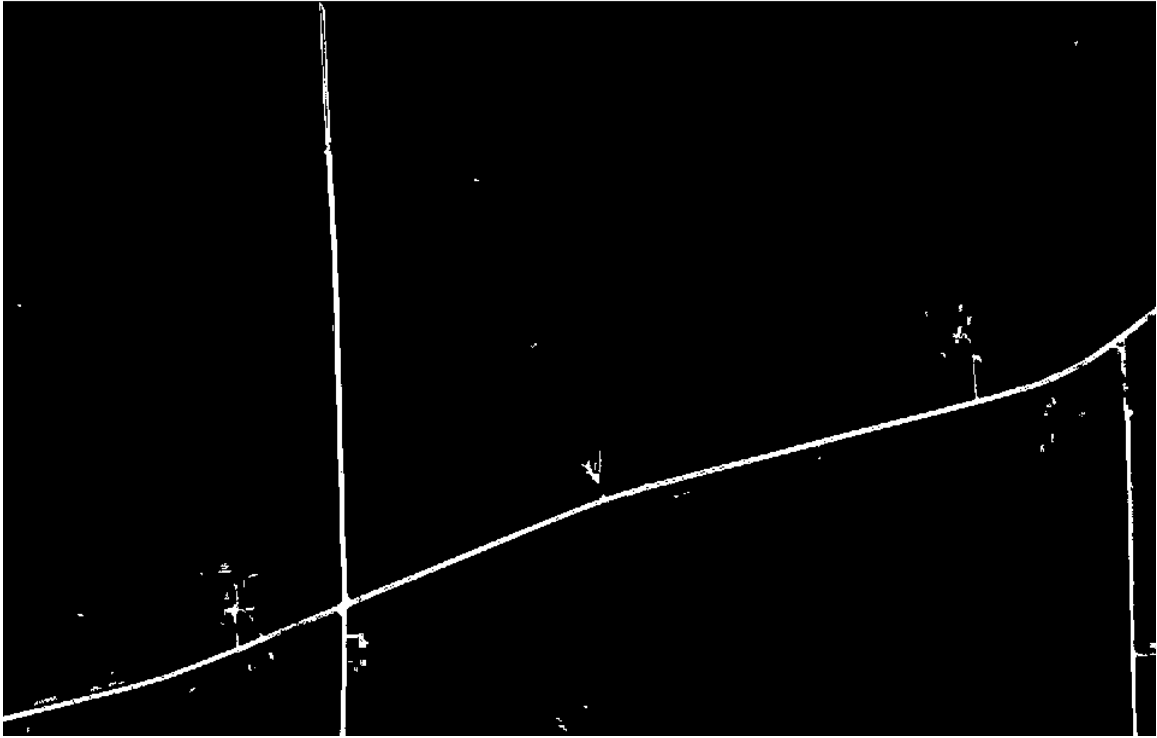


(d)

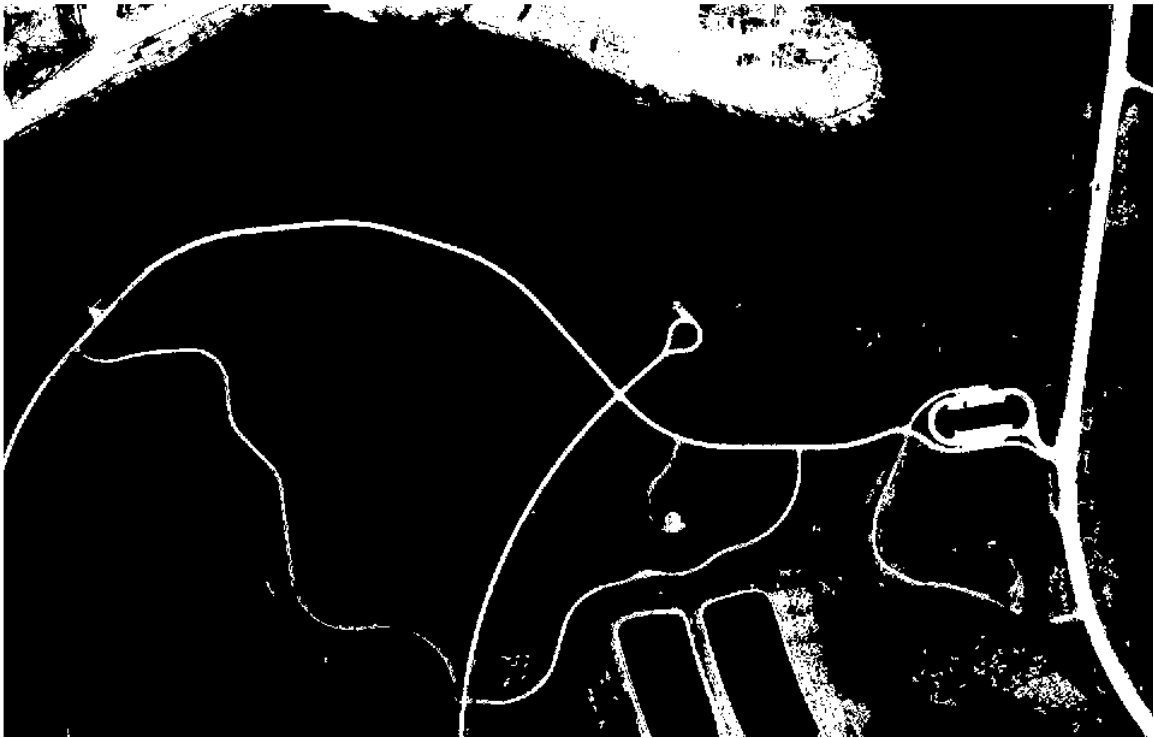
Figure 8: Segmentation by clustering. (a) Sparse road density, (b) Low road density, (c) Moderate road density, (d) High road density

3.5.6 Road space matching

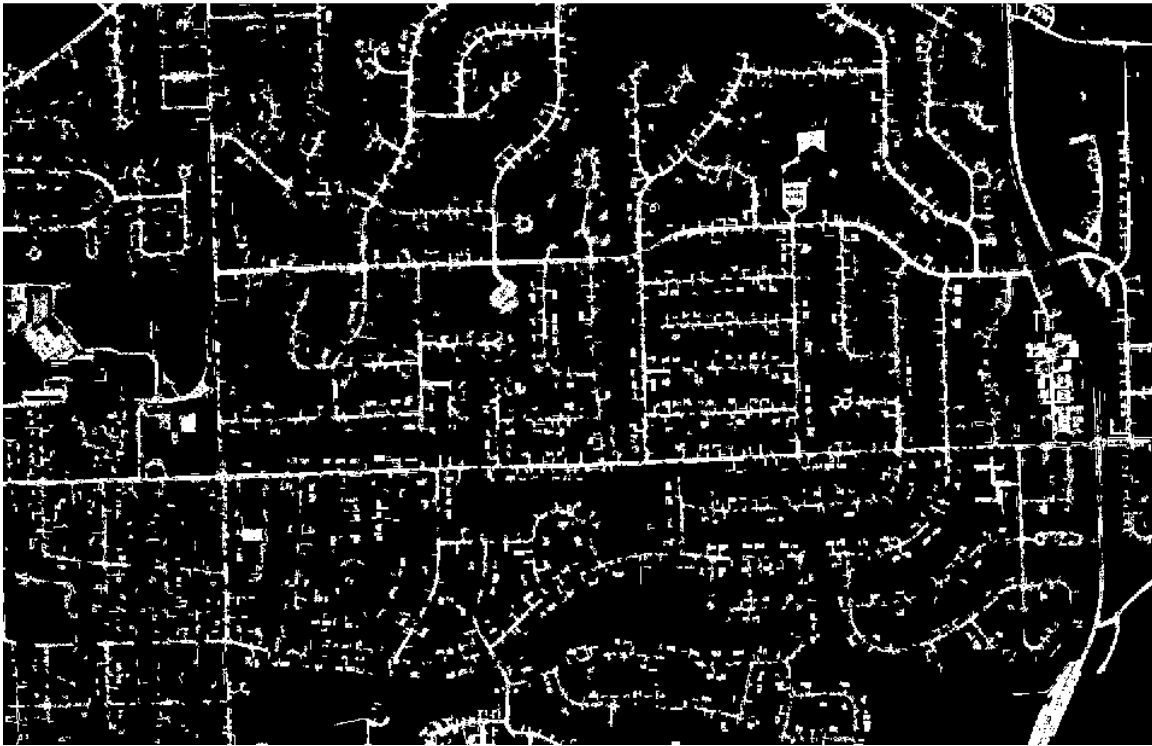
These results were produced using the road space matching algorithm. Here, the performance varies on a per-image basis. In particular, the presence of parking lots and concrete rooftops tends to result in false positives. As such, this algorithm produced the best results on the residential image and the worst on the urban image.



(a)



(b)



(c)



(d)

Figure 9: Road space matching. (a) Sparse road density, (b) Low road density, (c) Moderate road density, (d) High road density

4 USING EXISTING GIS TOOLS

From the review of literature, several approaches based on different algorithms have been suggested on how to improve road vector data. A majority of the approaches are based upon the extraction of road vectors from raster or aerial imagery. The caveat is that suggested algorithms success in road extraction is dependent on the type of data used.

Table 1: Advantages and disadvantages of the two approaches

<i>Approach</i>	<i>Advantages</i>	<i>Disadvantages</i>
Manual digitizing of road from aerial imagery	Easy and simple to use Accuracy depends on user	Laborious and time-consuming
Automated extraction of road vector data from aerial imagery	Applies uniform approach to different images: <ul style="list-style-type: none">- faster processing time- ability to detect roads not visible to human eye.	Higher possibility of error generation

As shown in Table 1, the simplest and most laborious approach is to digitize roads from a high quality aerial imagery. Road vector data generated from this source will be more accurate than road vector data from other sources. However, digitizing for road vector data for a large area is laborious and requires a lot of time. The way forward is to automate the process of improving road vector data using aerial imagery. The proposed method of exploring options of automation is scripting and model building.

Approach

We used the ESRI ArcGIS platform to generate improved road vector data from aerial imagery. ArcMap and ArcInfo provided geoprocessing tools for the extraction of road data from aerial imagery. The reason for use of aerial imagery as the starting point is because high quality aerial imagery serves as a good source of data for the generation of road vector data.

4.1 Steps or processes to automate aerial imagery

4.1.1 Prepare imagery for data extraction using ArcScan

ArcScan provides tools and algorithm resources to extract road vector data from aerial imagery. We used a process ESRI terms ‘raster pre-processing’ to prepare the imagery for data extraction. This process is heavily reliant on manually adjusting the raster, but ESRI creates a framework that can be used on other image resources beyond the test data. Thus, this process was deemed an appropriate approach for investigation. Preparing the image data for pre-processing required some image preparation steps.

First, we reclassified the aerial imagery into binary colors from its original color space to enable ArcScan to process the aerial imagery. When you add aerial imagery into ArcInfo or ArcMap, the aerial imagery shows three colors (red, green and blue) in the table of content section. The reclassify geoprocessing tool was used to reclassify the aerial imagery from three primary colors to two main colors. This is necessary because ArcScan requires binary color space to be able to process aerial imagery.

Second, the new binary colored image was used as a data source. Now that the colors are in binary form; roads, buildings and other features in the aerial imagery that are of similar material (concrete) appear as the same color.

4.1.2 Raster cleanup

Cleaning up the raster involves editing the data, essentially discriminating between road and non-road pixels. This is accomplished using the raster clean-up tool in ArcScan from the tool bar. Raster clean-up activates the raster painting tool which is used for editing the raster image. Alternatively the cell selection option can also be used to clean up unwanted cells or features in the aerial imagery.

Raster cleaning is the method for removing unwanted features and cells in the imagery. The erase tool of the painting tool bar was used with the magic erase tool to remove unwanted regions of the aerial imagery.

Alternatively, the select cell function (located under the cell selection option of the ArcScan toolbar dropdown options) can also be used to select and remove unwanted cells in the aerial imagery. This is done by entering the pixel size or number instead of entering total area of the raster pixels option. Selected cells can be erased by choosing raster cleanup >> dropdown option >> erase selected cells.

The raster cleaning step removed all unwanted features from the aerial imagery.

4.1.3 Vector extraction

ArcScan was then used to automatically generate all the road data from the aerial imagery. The generate features option under vectorization dropdown options of the ArcScan toolbar was used to create the road vector data. This section contains a step-by-step log of the process followed for automated digitizing.

4.1.3.1 Reclassify the raster or aerial imagery

Add the aerial imagery to ArcMap or ArcInfo

Click on Customize tab of ArcMap or ArcInfo, under the drop-down options, select extensions. Under extensions check Arcscan, 3D Analyst and Spatial Analyst to enable these tools to run.

Click on ArcToolbox, click on the plus sign beside the 3D Analyst Tools to open the options under it. Click on the plus sign beside Raster Reclass to open the options under it. Double click on Reclassify, which opens a Reclassify dialog.

In the opened dialog, under Input Raster – select the aerial imagery and click on the classify options. After the classification box opens, under classification method-select

Equal Interval and under Classes-select 2. Click on Okay and Okay to dismiss the opened dialogs. This will create a reclassify layer in the workspace.

Under Table of Contents, double click on reclassify layer that has just been created. In the Layer Properties box, click on the Symbology tab and change the colors to appropriate colors for easy identification; for example, black and white.

Click on catalog tab and create new shapefiles of a line and polygon and add them to the table of contents.

Alternatively, python codes which are provided below can be used to reclassify the aerial imagery by running the scripts in the Python idle program on a system with the appropriate libraries and ESRI ArcGIS modules.

4.1.3.2 Python Codes

```
# Import arcpy module
```

```
import arcpy
```

```
from arcpy import env
```

```
arcpy.env.overwriteOutput = True
```

```
# Check out any necessary licenses
```

```
arcpy.CheckOutExtension ("spatial")
```

```
# Local variables
```

```
Name of raster_tiff = "Name of raster.tiff"
```

```
raster_reclassify = "H:\\GIS\\folder\\name of file geodatabase\\raster_reclassify"
```

```
# Process: Reclassify
```

```
arcpy.gp.Reclassify_sa (name of the raster_tif, "Value", "1; 2", Reclassify_tif, "Data")
```

4.1.3.3 Parameterization of the core script

Replacing the file names of the data resources with variables will allow this script to be used to process other data resources than the sample.

Python is a high-level interpreted language; thus, the variables should be defined following the header portion of the python code (upstream from the #Process section).

4.1.3.4 Use Arcscan to automate the digitizing of road from the reclassified-raster

Click on the Arcscan tool to select it. Activate Editor and click on start editing. Under the ArcScan tool bar, click on Raster Clean-up tab, under the drop down options, click on start clean-up. Under the Raster Clean-up tab, this time click on Raster Painting Toolbar

and use the tools to erase and clean the unwanted objects. Alternatively, you can use cell selection by clicking on the cell selection tab. Under the cell selection drop-down arrow, click on select connected cells. In the opened select cells dialogue box, specify the raster pixels that you want to clean-up. After choosing the necessary values in the select cells dialogue box, click okay to effect changes. The highlighted unwanted parts of the raster or aerial imagery can be deleted by clicking on the raster clean-up tab on the ArcScan tool, under the dropdown options-click on start clean-up and after that under the same area (raster clean-up), click on erase selected cells.

After the raster preprocessing (clean-up), click on the vectorization tab found on the ArcScan open tool bar box. Under the vectorization drop-down options, click on generate features to create the road vector data. In order to make sure that generate features works properly, when you start editing choose the polyline option when prompted. Now click on the generate features option, in the generate features dialogue box, make sure that in the topmost template option-the polyline is selected and in the bottom template option-the polygon is selected. Click on Ok to generate the road vector data.

5 SUMMARY

In the current phase of work, two methods were explored, a method based on existing GIS tools and one based on image processing techniques.

Existing GIS tools can successfully extract road vector data from images of arbitrary complexity. However, this analysis requires significant manual refinement of intermediate results in order to obtain good results.

Image processing techniques can be leveraged to automate much of the process of road vector extraction. Obtaining good results, however, can be challenging on more complicated imagery. Additionally, complex analyses can require minutes per image, although this does not require human oversight. Exploration of this method indicates that some means of differentiating between roads and similar-looking features must be incorporated to fully automate the process of road vector extraction.

While presented separately, these two methods can be viewed as complementary approaches. That is, image processing techniques can be applied to imagery initially, and the results obtained can be plugged into GIS tools for refinement and additional processing. In this way, time spent on manually cleaning up the images will be reduced.

5.1 Proposed Next Step – Artificial Intelligence-based Techniques

Based on these results, we propose the investigation of artificial intelligence-based techniques as a next step in the exploration of automatically extracting data. Principally, this would involve the application of deep learning algorithms to the problem of identifying roads within images. The expected result would be the segmentation of a set of sample images that is sufficiently accurate to extract road centerlines without the need for any human intervention. Additionally, the question of divining road connectivity and traffic flow through the extracted data must be addressed. We expect these to be second-order difficulties, however. A statement of work and proposal would be developed in consultation with Rockwell personnel to ensure efforts align with their needs.

A rough order of magnitude for the proposed next step is \$75 -\$100K over the course of six to nine months. Based on current staffing allocations, we could begin this effort as soon as July 2015.

6 REFERENCES

Ahmadi F.F., Zoej M.J.V., Ebadi H. & Mokhtarzade M. (2008) The application of neural networks, image processing and CAD-based environments facilities in automatic road extraction and vectorization from high resolution satellites images. Retrieved from:

http://www.isprs.org/proceedings/XXXVII/congress/3b_pdf/109.pdf

Barua S. (2007) A generic framework for the application of graph theory to image processing.

Chen Ching-C. (2005) Automatically and accurately conflating road vector data, Street Maps and Orthoimagery. Retrieved from:

<http://search.proquest.com/pqdtft/docview/305428855/fulltextPDF/5C2911FCD6A74B5EPQ/1?accountid=14663>

Chiang Yao-Y. & Knoblock C.A. (2009) Extracting road vector data from raster maps.

Chiang Yao-Y. & Knoblock C.A. (2011) A general approach for extracting road vector from raster maps. Retrieved from:

<http://link.springer.com/article/10.1007%2Fs10032-011-0177-1#page-1>

Chiang Yao-Y. & Knoblock C.A. (2012) Generating named road vector data from raster maps. Retrieved from:

http://link.springer.com/chapter/10.1007%2F978-3-642-33024-7_5#page-1

Center for International Earth Science Information Network (2013) Methods used in the development of global roads open access data set (gROADS), version 1. Palisades NY: NASA Socioeconomic Data and Applications Center (SEDAC).

Doucette P., Agouris P., Musavi M. & Stefanidis A. (2000) Road centerline vectorization by self-organized mapping.

Esri ArcGIS Online Resources (2013). Retrieved from:

<http://resources.arcgis.com/en/help/main/10.2/index.html#//01m500000003000000>

Gautama S. & Borghraef A. (Nd) Detecting change in road networks using continuous relaxation labeling. Retrieved from:

<http://www.ipi.uni-hannover.de/fileadmin/institut/pdf/gautam.pdf>

Gecen R. & Sarp G. (2008) Road detection from high and low resolution satellite images.

Gold C. & Dakowicz M. (Nd) The crust and skeleton-applications in GIS.

Institute for Road Safety Research (2011) Data requirements for road network inventory studies and road safety evaluations-guidelines and specifications.

Jin X. & Davis C.H. (2005) An integrated system for automatic road mapping from high-resolution multi-spectral satellite imagery by information fusion. Retrieved from: <http://www.hamilton.ie/skirkland/SpectralRoad.pdf>

Karamanou A., Papazissi K., Paradissis D. & Psarianos B. (2009) Precise estimation of road horizontal and vertical geometric features using mobile mapping techniques.

Klang D. (1998) Automatic detection of changes in road databases using satellite imagery. Retrieved from: http://pdf.aminer.org/000/312/172/detection_of_roads_from_satellite_image_using_the_optimal_search.pdf

Li X., Lin H. & Zhao Y. (2005) A connectivity-based map matching algorithm

Mena J.B. (2003) State of the art on automatic road extraction for GIS update: a novel classification.

Mena J.B. (2006) Automatic vectorization of segmented road networks by geometrical and topological analysis of high resolution binary images. Retrieved from: <http://www.sciencedirect.com/science/article/pii/S0950705106001018#>

Poullis C. (2014) Tensor-cuts: A simultaneous multi-type feature extractor and classifier and its application to road extraction from satellite images.

Song W. (2011) Automated vector-vector and vector-imagery geospatial conflation. Retrieved from: <http://search.proquest.com/pqdtft/docview/305428855/fulltextPDF/5C2911FCD6A74B5EPQ/1?accountid=14663>

Wang Y., Chen Z., Cheng L., Li M & Wang J. (2013) Parallel scanline algorithm for rapid rasterization of vector geographic data. Retrieved from: <http://ac.els-cdn.com/S0098300413001465>

Yang B., Luan X. & Li Q. (2010) An adaptive method for identifying the spatial patterns in road networks.

Zhang Z., Zhang H. & Zhang J. (2003) Automatic road extraction based snake and line photogrammetry.

7 APPENDICES

7.1 GNU Image Manipulation Program (GIMP) procedures

7.1.1 Select roadway

- Load the image into GIMP.
- In the toolbox window, set foreground color to white and background color to black.
- In the toolbox window, click the Fuzzy selection or Select by color tool
- Click on a representative road pixel, holding the button down. Then, drag left/right to fine tune selection and let go.
- In the toolbox window, click the Bucket fill tool.
- SHIFT + click within the selection.
- CTRL + I to invert selection.
- CTRL + SHIFT + click in the inverted selection.

7.2 MATLAB code

These scripts require the Image Processing Toolbox and the Statistics Toolbox.

7.2.1 Colorspace plot

```
function colorspace(image)
    pixels = reshape(double(image), [], 3);
    colors = bsxfun(@rdivide, pixels, 255);
    figure;
    scatter3(pixels(:,1), pixels(:,2), pixels(:,3), [], colors, '.');
    axis([0 255 0 255 0 255]);
    xlabel('red');
    ylabel('green');
    zlabel('blue');
end
```

7.2.2 Lighting correction and contrast enhancement

```
function new_image = shade_and_enhance(image, factor)
    % shade_and_enhance collapses an image's luminosity to the mean
    % and enhances the hue.
    % Inputs:
    %   image: input image in RGB format
    %   factor: the degree of change (1 = no change)
    % Outputs:
    %   new_image: the resulting image
    if nargin < 2
        factor = 1;
    end

    pixels = double(reshape(image, [], 3));

    average = mean(pixels);
    deviations = bsxfun(@minus, pixels, average);
    principle_components = princomp(deviations);
    sun = principle_components(:, 1);

    pixels_in_sun_basis = deviations * principle_components;
```

```

    pixels_in_sun_basis(:, 1) = bsxfun(@rdivide, pixels_in_sun_basis(:, 1),
factor);
    pixels_in_sun_basis(:, 2:3) = bsxfun(@times, pixels_in_sun_basis(:, 2:3),
factor);

    new_deviations = pixels_in_sun_basis / principle_components;

    new_pixels = bsxfun(@plus, new_deviations, average);
    too_big = max(new_pixels, [], 2) > 255;
    too_small = min(new_pixels, [], 2) < 0;
    out_of_bounds = bsxfun(@or, too_big, too_small);
    for i = 1:size(new_pixels, 1)
        if out_of_bounds(i)
            new_pixels(i, :) = plane_line_intersect(sun, average, new_pixels(i,
:), average);
        end
    end
    new_image = uint8(reshape(new_pixels, size(image)));
end

function I = plane_line_intersect(n, V0, P0, P1)
    %plane_line_intersect computes the intersection of a plane and a segment
    % Inputs:
    %   n: vector normal to the plane
    %   V0: any point on the plane
    %   P0: start point of the segment
    %   P1: end point of the segment
    % Outputs:
    %   I: the point of interection
    u = P1 - P0;
    w = P0 - V0;
    D = dot(n, u);
    N = -dot(n, w);

    sI = N / D;
    I = P0 + sI .* u;
end

```

7.2.3 Cluster by color

```

function [clusters, centers, distances] = cluster_by_color(image, numClusters)
    imageSize = size(image(:, :, 1));
    RGB = double(reshape(image, [], 3));
    [C, centers] = kmeans(RGB, numClusters);
    clusters = zeros(numClusters, imageSize(1), imageSize(2), 3);
    distances = zeros(numClusters, 1);
    for i = 1:numClusters
        distances(i) = point_to_line(centers(i,:), [0 0 0], [255, 255, 255]);
        colors = zeros(numClusters, 3);
        colors(i,:) = [1 1 1];
        clusters(i, :, :, :) = ind2rgb(reshape(C, imageSize(1), imageSize(2)),
colors);
        figure;
        imshow(squeeze(clusters(i, :, :, :)));
    end
end

function d = point_to_line(pt, v1, v2)
    a = v1 - v2;
    b = pt - v2;
    d = norm(cross(a, b)) / norm(a);
end

```

7.2.4 Remove stray points from processed image

```
function new_image = despeckle(image)
    new_image = bwareaopen(im2bw(image, 0.5), 10, 4);
end
```