AUTOMATED VECTOR-VECTOR AND VECTOR-IMAGERY GEOSPATIAL CONFLATION

A Dissertation

Presented to

The Faculty of the Graduate School

University of Missouri-Columbia

In Partial Fulfillment

Of the Requirements for the Degree

Doctor of Philosophy

---

by

WENBO SONG

Dr. James M. Keller, Dissertation Supervisor

MAY 2011

The undersigned, appointed by the dean of the Graduate School, have examined the dissertation entitled

AUTOMATED VECTOR-VECTOR AND VECTOR-IMAGERY GEOSPATIAL CONFLATION

Presented by Wenbo Song,

A candidate for the degree of Doctor of philosophy

And hereby certify that, in their opinion, it is worthy of acceptance.

Professor James Keller

Professor Marjorie Skubic

Professor Curt Davis

Professor Chi-Ren Shyu

Professor Kannappan Palaniappan

ACKNOWLEDGMENTS

First, I would like to express my deepest gratitude to my advisor, Dr. James Keller for his intellectual and invaluable guidance during my doctoral study. It was my pleasure and honor to be his student. I would not have finished this work without his training and direction.

Also I would like to thank my doctoral committee members, Dr. Marjorie Skubic, Dr. Curt Davis, Dr. Chi-Ren Shyu and Dr. Kannappan Palaniappan for their insightful suggestions on my dissertation.

Special thanks to my boss Tim Haithcoat at MU Geographic Resources Center. This dissertation would not have been possible without his support. Also I would like to thank my colleagues at MU Geographic Resources Center for their help.

Finally, I would like to thank my family, my wife Jie Ning, my son David and daughter Jenna. They are always there to support and encourage me.

TABLE OF CONTENTS

LIST OF FIGURES

Figure **Page**

LIST OF TABLES

Table **Page**

AUTOMATED VECTOR-VECTOR AND VECTOR-IMAGERY GEOSPATIAL CONFLATION

Wenbo Song

Dr. James Keller, Dissertation Supervisor

ABSTRACT

With the rapid advance of geospatial technologies, the availability of geospatial data from multiple sources has increased dramatically. Integration of multi-source geospatial data can provide insights and capabilities not possible with individual datasets. However, multi-source datasets over the same geographical area are often disparate and do not match well with each other. Accurately integrating geospatial data from different sources is a challenging task. In this dissertation research, we proposed a set of innovative geospatial conflation algorithms to attack the multi-source geospatial integration/conflation problem. We developed a novel snake-based approach to conflate two vector road datasets which has several benefits over traditional conflation methods. Since feature matching is one of the most crucial subtasks of conflation, we proposed a new relaxation labeling-based point matching algorithm to provide an elegant and well-motivated solution to the conflation problem. For the vector-to-imagery conflation, we presented a comprehensive approach by integrating several vector-based and image-based algorithms including spatial contextual signature extraction, road intersections and terminations extraction, relaxation labeling-based point matching, piecewise rubber-sheeting transformation, and snake-based refinement. Finally we extended our road conflation approach to digital parcel map to make it consistent with high-resolution imagery. The experiments on real world geospatial datasets showed excellent results.

Chapter 1

Introduction

1.1 Map

We live in a spatial world and we use spatial information in our everyday life. For example, when we drive from a known place to another, we may only subconsciously use the ‘virtual map’ in our mind to get there. But when we visit an unknown place, we need the real paper road map or digital map on a Global Positioning System (GPS) navigation system to guide us. A map (paper or digital) has become an essential tool with variety of applications. By definition, a map is a graphical representation of the environment that shows the spatial features and their relationships [1]. It’s a depiction of all or part of the earth or other geographic phenomenon as a set of symbols and at a scale whose representative fraction is less than 1:1 [2].

There are many different types of maps such as road maps, land cover maps, thematic maps, topographic maps etc. A topographic map is a type characterized by large-scale detail and quantitative representation of relief, usually using contour lines in modern mapping. A topographic map shows both natural and man-made features [3]. The Canadian Centre for Topographic Information provides the following definition [4]: “A

topographic map is a detailed and accurate graphic representation of cultural and natural features on the ground."

Topographic maps have a wide variety of applications, from urban planning, engineering, energy exploration, natural resource management, to outdoor activities like camping, hiking and fishing, etc. A topographic map is often used as the base map for other thematic mapping and applications. It would be helpful to understand its mapmaking processes.

1.2 Classic Topographic Mapmaking

The U.S. Geological Survey (USGS) produced its first topographic map in 1879, the same year it was established. Most of the early maps were made using a classic mapping technique called plane-table surveying (figure 1.1). A plane-table is a portable drawing board which is fixed to a tripod with a sighting device used to observe the features of interest in the field. This is a method of survey that allows a surveyor to plot those features that could be seen and measured in the field and sketch the map by hand. Plane-table surveying remained the dominant USGS mapping technique until the 1940's, when it gave way to the aerial photographs and the age of Photogrammetry [5]. By definition, photogrammetry is the art, science, and technology of obtaining reliable information about physical objects and the environment through the processes of recording, measuring, and interpreting photographic images and patterns of electromagnetic radiant energy and other phenomena [6].

Figure 1.1 Plane-table surveying by USGS topographers (source: [5])

The use of aerial photographs for mapping was pioneered in the 1930's, when the USGS assisted the Tennessee Valley Authority in mapping its area of responsibility. This project was the first full-scale test of the use of aerial photographs in creating maps. Aerial photographs increased dramatically during World War II when its use proved crucial for gathering military intelligence. Aerial photographs and Photogrammetry led to a revolution in mapmaking. This change has significantly increased map coverage and enhanced map standardization [5].

However, producing an accurate topographic map is a long and complex process. It may take as much as 5 years from the identification of a mapping requirement to the printing of a large-scale map like one of the USGS 7.5-minute, 1:24,000-scale quadrangle maps. This process requires a team of professionals and a series of closely coordinated steps descried in the following [5].

Photographing the area

The first step in producing a topographic map is acquiring aerial photographs of the area being mapped. Each section of ground is photographed from two different angles to produce a stereoscopic three-dimensional image (figure 1.2). The sky must be clear, and the sun must be at the proper angle for the type of terrain being photographed. Seasonal factors must also be taken into consideration. For example, in areas where there are deciduous trees, the photos are usually taken between late fall and early spring when the trees are bare and the underlying ground features are more visible.

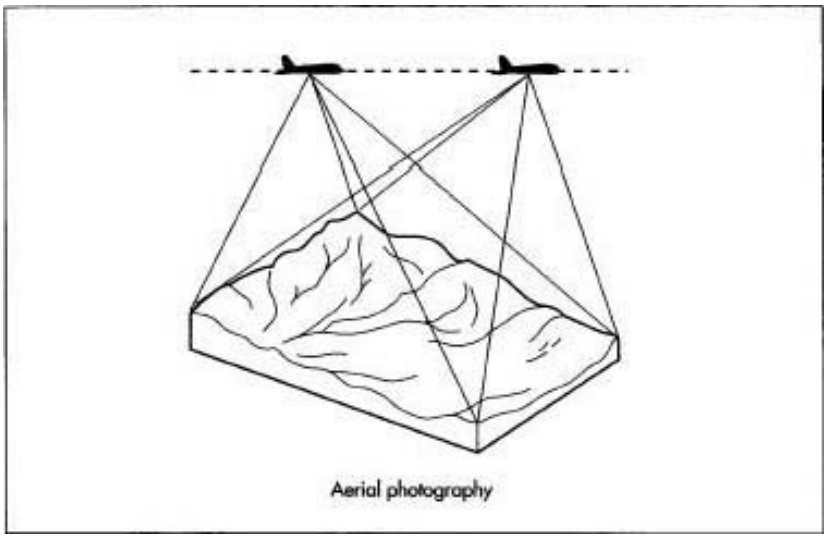


Figure 1.2 Overlapping aerial photographs provide stereoscopic coverage of areas to be mapped (source: [5])

Field surveying the control points and verifying the map features

After aerial photographs are obtained, field survey work may be required to establish and measure the map's basic control points. Horizontal control points are surveyed to determine the longitude and latitude, while vertical control points are surveyed separately to determine elevations. The control points help the map makers

correctly position the aerial photo images and become the framework on which map detail is compiled.

While the surveyors are in the field, they also identify objects that need visual verification. Some map features may require additional verification and the surveyors verify these features by talking with local residents or consulting local property records. The geographic names of place and political boundaries have to be collected and verified in the field.

Compiling the map manuscript

Upon completion of the field survey, a pair of overlapping aerial photographs showing the same ground area taken from different position along the flight line is placed in a special projector connected to a separate tracing table. The projected photographs are viewed through an optical system called a stereoscope that causes the left eye to see one photograph and the right eye to see another. The result is a three-dimensional view of the terrain from which a cartographer can draw a topographic map. Map features and contour lines are manually traced as they appear in the stereo model. As the operator moves a reference mark, the tracing is transmitted to the tracing table, producing the map manuscript.

Scribing, editing and printing the map

The finished map manuscript is photographed and a map-size film negative is made. This film negative is photochemically reproduced onto several thin plastic sheets coated with a soft translucent coating called a scribecoat. The plastic sheets are taken one

at a time and placed on a light table, where a soft light shines up through a white plastic surface. This illumination from below makes the lines of the map manuscript visible through the scribecoat. An engraver carefully cuts away the scribecoat along the lines and areas that are to be a certain color on the finished map. For example, one sheet will have all the lines for rivers, lakes, and other bodies of water that are to be blue. This process is repeated for each color [7].

After the scribed sheets are reviewed and edited several times, a color proof sheet is made by exposing each sheet under different color light to produce a color print that looks very much like the finished map. After further review and editing, the map is ready to be printed.

A press plate is prepared for each map color by exposing the scribed sheets. Paper is loaded into a lithographic printing press, and the first color is printed. The press plate and ink are changed and the paper is run through the press a second time to print the second color. This process is repeated until all the colors have been printed [7].

Figure 1.3 shows one portion of USGS 1:24,000 topographic map of Columbia, Missouri. Geospatial features are shown as points, lines or areas, depending on their size and extent. The build-up areas are shown in red and the individual buildings are shown as small black squares. Blue areas are lakes/ponds and green are vegetation. Red lines represent roads and the topographic contours are represented by brown lines.

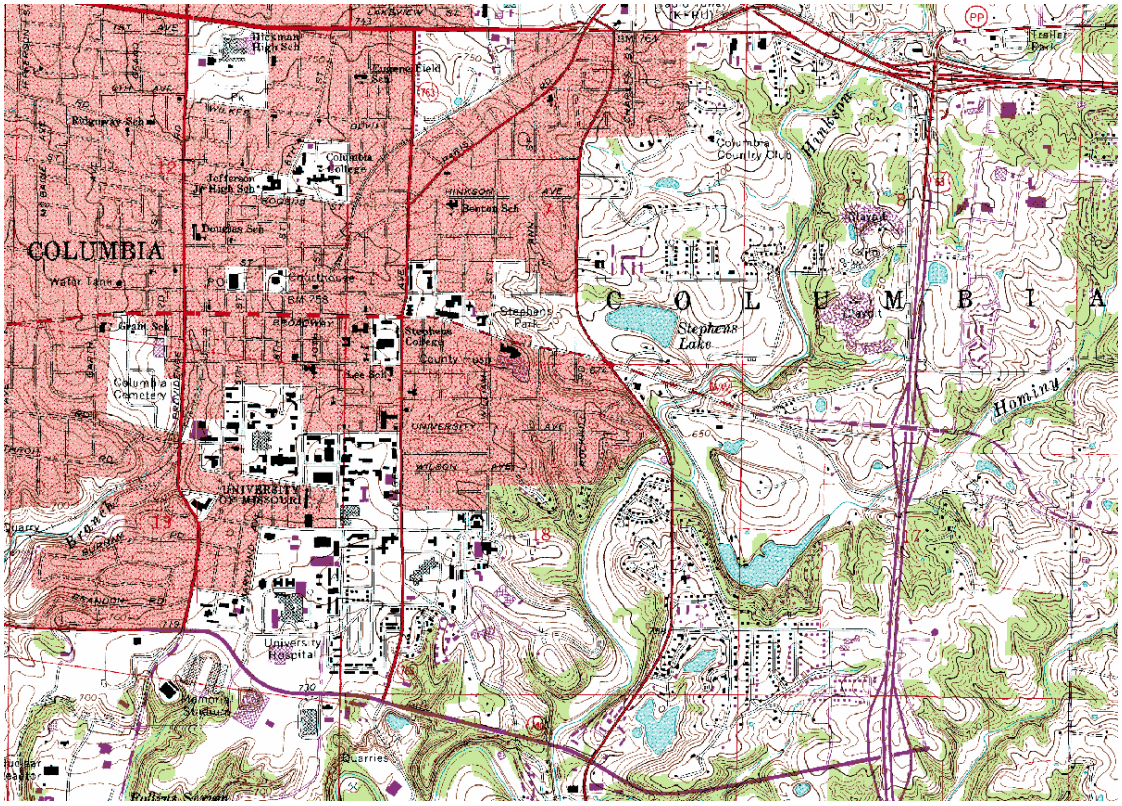


Figure 1.3 Portion of 1:24,000 USGS topographic map of Columbia, MO

The 1:24,000-scale, 7.5-minute topographic quadrangle map series includes about 53,000 map sheets which is the only uniform map series that covers the entire area of the continental United States in considerable detail. The 7.5-minute mapping program lasted from the mid-1940's until early 1990's. Since late 1960's, USGS had established a revision program to do minor or basic map revision using aerial photographs to update a subset of map features. Map producers have to manually revise cartographic features via photo interpretation. Geographic names and administrative boundaries are updated using information from local sources. This manual map revision is very expensive and time-consuming even for basic revisions. Basic revisions done with USGS government labor in 1999 required an average of 280 hours per quadrangle or approximately $17,000. Up

to 1400 1:24,000 map sheets can be revised each year. The median currentness date of the map series is 1979 even after map revision [8]. Now the USGS 1:24,000 topographic maps are more than 30 years old.

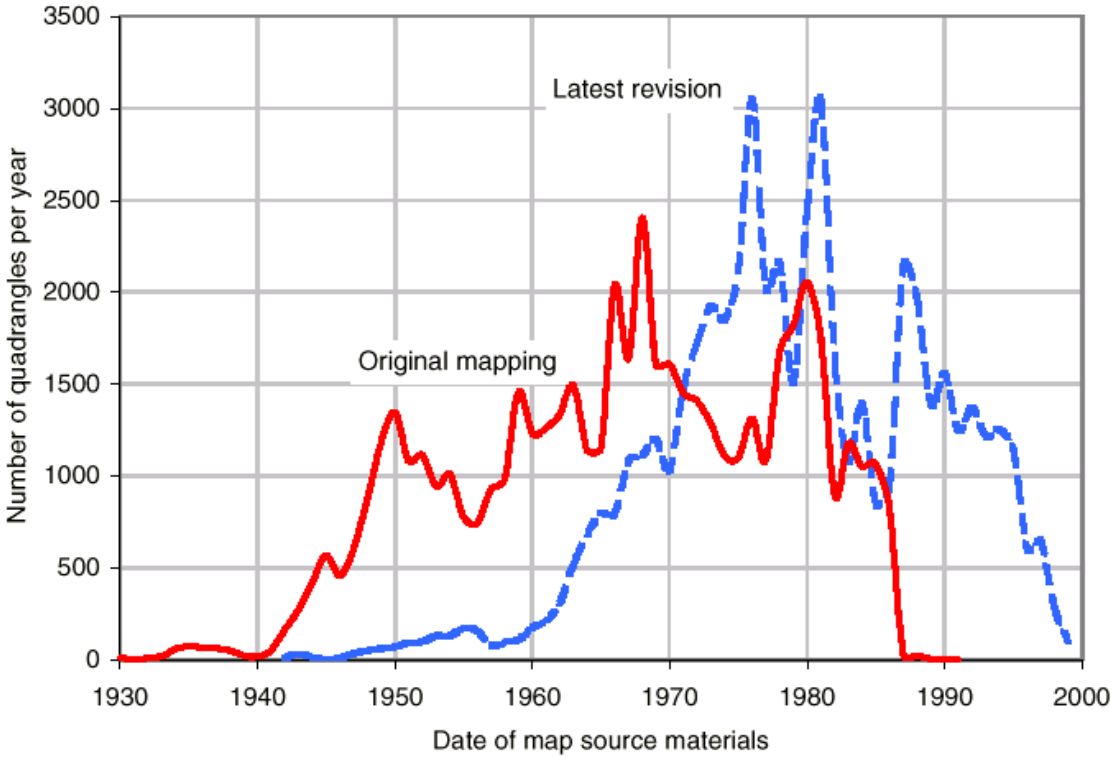


Figure 1.4 Currentness of Revised USGS Maps Compared to Original Maps (Source: [8])

1.3 Digital Mapping Revolution

With the rapid development of geospatial technology, many of the traditional mapmaking processes are being changed or eliminated by digital mapping such as a Geographic Information System (GIS). GIS is a collection of computer hardware, software, and geographic data for capturing, storing, updating, manipulating, analyzing, and displaying all forms of geographically referenced information [9]. Rapid advance of information technology and declining hardware cost have made GIS affordable to an

increasing wider audience. The use of GIS grew dramatically during the last three decades. It is now commonplace for government, academia, business, individuals to use GIS for many diverse applications. Many organizations now spend large amounts of money to create geospatial databases and build geographic information systems. To be used in GIS, the paper maps need to be converted into digital format. This conversion process is called geocoding. Studies have shown that finding the right maps, and converting these maps from paper to digital form by geocoding, takes up anywhere between 60% and 90% of both the time and money spent on a typical GIS project [2].

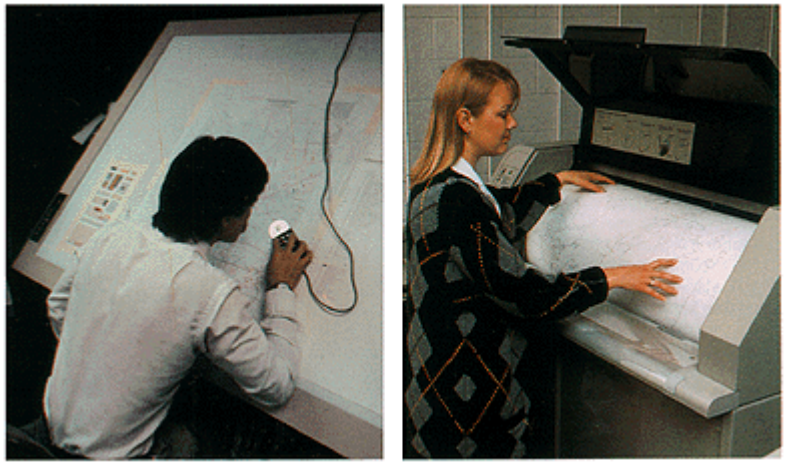


Figure 1.5 Vector digitizing with a tablet (left), raster scanning with a drum scanner (right) (Source: [10])

Digitizing and scanning are the two main geocoding technologies (figure 1.5). Digitizing mimics the way maps were drafted by hand and involves tracing the map feature using a cursor while the map is taped down onto a digitizing tablet. The digitizing tablet is a digital and electronic equivalent of the drafting table. The operator traces a map feature by hand, using a digitizer cursor to capture the locations of a string of points and

sends to the x, y units to computer [2]. A map can have 10 or more different layers such as roads, contours, boundaries, surface cover, and manmade features that require digitization.

Scanning is the second digitizing process. Scanning places a map on a glass plate and passes a light beam over it, measuring the reflected light intensity. The results are raster maps and can be converted into vector format. It needs further editing or tracking features on the computer screen.

After digitizing/scanning, several editing operations remain. For example, attribute codes must be added to identify what each digitized line or symbol represents. A variety of other tasks must be performed to ensure that information is complete and correct, including matching features with adjoining files, matching features relative to each other within the file, and controlling the accuracy of attribute coding and positions [2].

Once the digital geospatial data including both geometric and attribute information are ready, it's much easier to produce, revise, distribute and utilize digital maps. Paper maps can be printed. Figure 1.6 shows an example of digital Census 2000 block map of Columbia, MO.

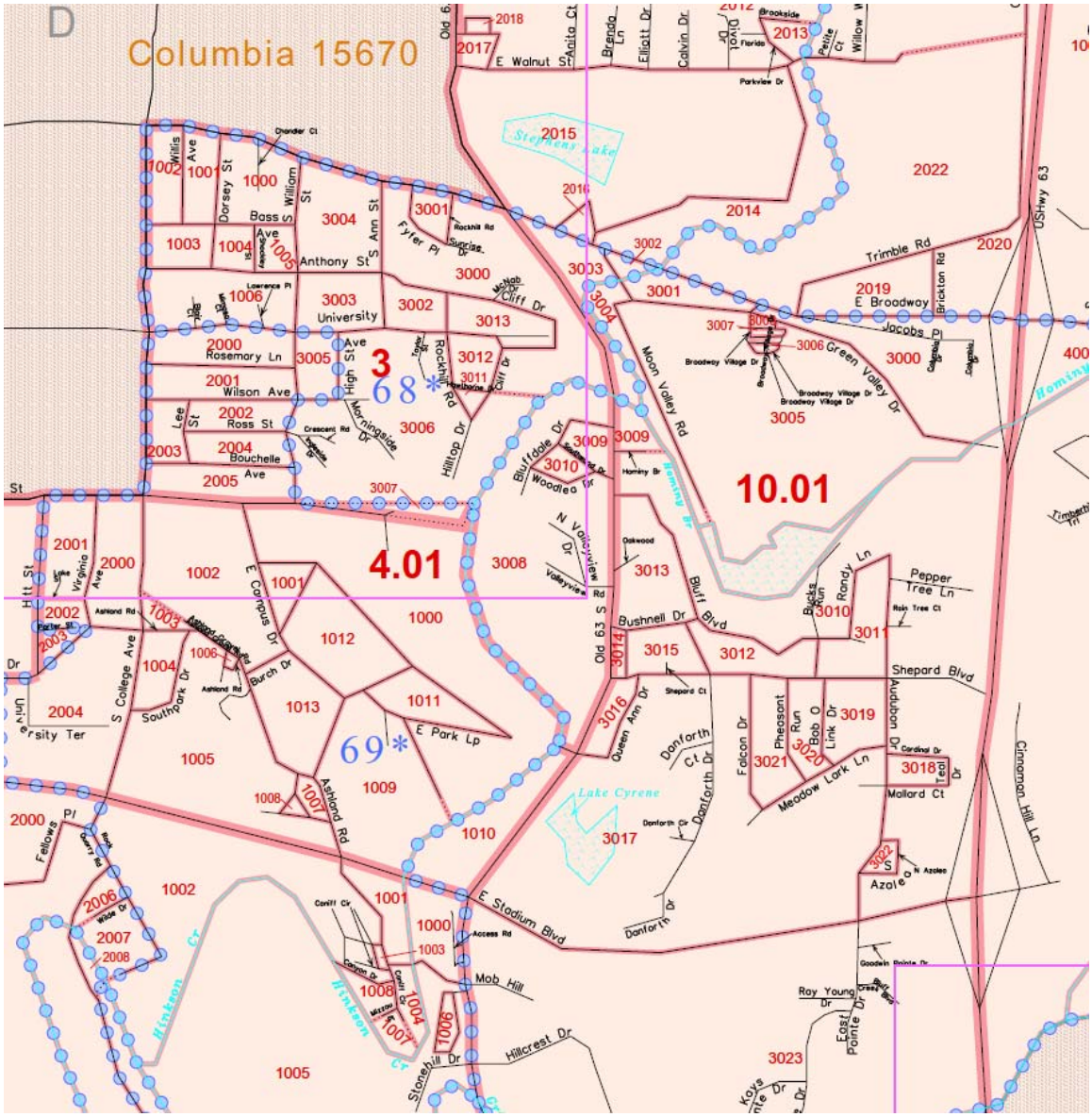


Figure 1.6 Portion of Census 2000 County Block map of Columbia, MO

1.4 Remote Sensing Imagery

The science and technology of remotely acquired data and information has undergone a revolution in the number and type of sensors, data availability, potential applications, and governmental and commercialization activities. There has been a massive increase in our ability to acquire radiometrically sensitive, geospatially-

referenced sensor data from aircraft, satellite and undersea instrument platforms [11]. With the rapid advancement of remote sensing technology, there are a variety of airborne and satellite remote sensing systems that have collected large volumes of imagery data, which is invaluable in monitoring the Earth system and the effects of human activities on the Earth.

Satellite Systems

In 1972 the first civil satellite, the Earth Resources Technology Satellite (ERTS-1, later renamed Landsat-1) was launched. This is one of a series satellites designed specifically to collect data of the Earth's surface and resources. The newest satellite in the series, Landsat 7, carries the enhanced thematic mapper plus (ETM+) with 30 m resolution visible and infrared bands and a 15 m resolution panchromatic band. A 30 m resolution ETM+ multi-spectral image of Columbia, MO is shown in figure 1.7. Landsat represents the world's longest continuously acquired collection of space-based moderate-resolution land remote sensing data. Nearly four decades of imagery provides a unique resource for agriculture, geology, forestry, regional planning, education, mapping, and global change research. Landsat data have been used by government, commercial, industrial, civilian, military, and educational communities in the United States and worldwide [11].

Figure 1.7 Landsat 7 ETM+ multispectral 30 m resolution imagery of Columbia, MO

Figure 1.8 Multispectral 4 m resolution IKONOS imagery of MU campus, MO

On Sept. 24, 1999 the world's first privately owned commercial, high-resolution imaging satellite IKONOS was launched by Space Imaging Inc. It has a 1 m resolution panchromatic band and 4 m resolution multi-spectral bands. A sample of IKONOS imagery of MU campus is shown in figure 1.8. Since then, several other high-resolution commercial satellite systems such as Quickbird, Worldview etc. were launched also. The world's highest resolution commercial earth-imaging satellite Geoeye-1 was launched in

Sept. 2008 with 0.4 m panchromatic band and 1.6 m multi-spectral bands. Many other counties such as France, Russia, China, India etc. also launched high resolution satellites. Table 1.1 lists several selected high resolution satellite systems.

Satellite	Pan Resolution (m)	MS resolution (m)	Country
Geoeye-1	0.4	1.6	US
WorldView-2	0.5	1.8	US
QuickBird-2	0.6	2.5	US
IKONOS-2	1.0	4	US
EO-1	10	30	US
Landsat 7	15	30	US
TopSat	2.5	5	UK
SPOT-5	2.5	10	France
Monitor-E	8	20	Russia
Resur DK-1	1	3	Russia
IRS Resource Sat-1	6	6,23,56	India
ALOS	2.5	10	Japan
KOMP Sat-2	1	4	Korea
Beijing-1	4	32	China
CBERS-3	5	20	China/Brazil

Table 1.1 Selected high resolution satellite systems

Aerial Photography

Although aerial photography cannot provide repeated global imagery coverage as satellites do, it can be customized according to users' requirements and collect local or regional imagery at even higher spatial resolution. Today aerial photography remains a fundamental element of many Earth inventory, environment, and resource applications. Airborne photographic systems routinely provide high spatial and spectral resolution data

(infrared, color, panchromatic) under requests of local, state, federal government or private companies. While commercial satellite imagery is not free to the public, users can often access aerial photographs acquired by federal programs free of charge.

USGS Digital Orthophoto Quadrangle (DOQ)

The National Aerial Photography Program (NAPP) is a multiple agency (U.S. Geological Survey, U.S. Department of Agriculture, U.S. Forest service) program coordinated by USGS that provides a standardized set of aerial photographs covering the United States. The program began in 1987 and ended in 2004. The NAPP photographs are available in black & white (B/W) or color-infrared (CIR), depending on location and date. The photographs are quarter-quadrangle centered (3.75 minutes of longitude and latitude in geographic extent) and are exposed at a nominal flying height of 20,000 feet above mean terrain with a 6-inch focal-length camera (photo scale = 1:40,000). They have 1-meter ground resolution [12]. Figure 1.9 shows an example of DOQ for MU campus.

The NAPP photography is the primary source used for the production of digital orthophoto quadrangle (DOQ) by scanning its transparency with a precision image scanner. A digital orthophoto is a geo-referenced image prepared from a perspective photograph, or other remotely-sensed data in which the displacement within the image due to sensor orientation and terrain relief has been removed. Orthophotos combine the image characteristics of a photograph with the geometric qualities of a map. The digital orthophoto is useful as a base layer of a geographic information system and as a tool for revising digital vector maps. Image processing algorithms can be applied to the image

data to support planimetric image classification, three-dimensional modeling, and many other spatial applications [12]. The DOQs are available for free download from USGS.

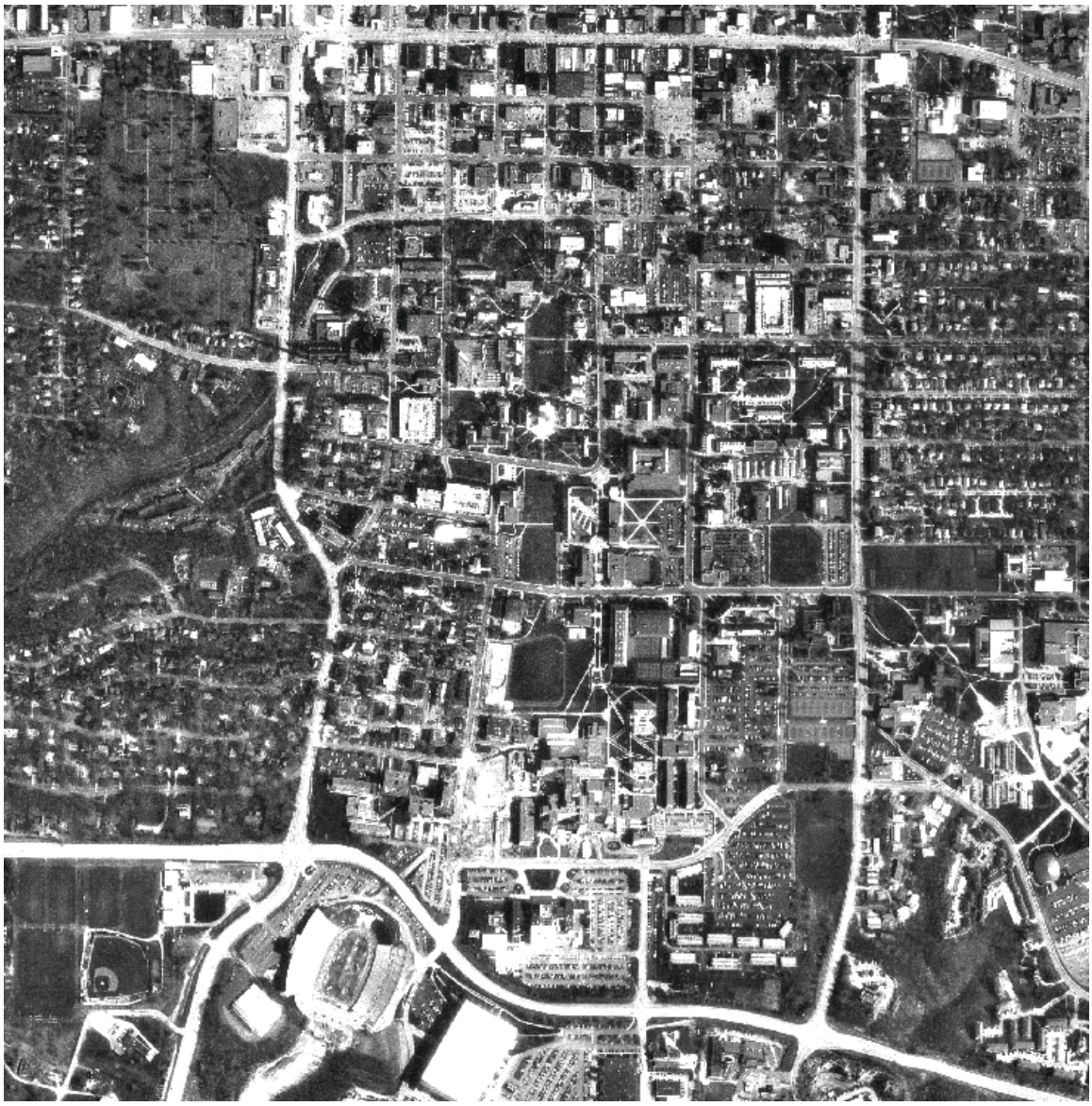


Figure 1.9 Black & white 1 m resolution USGS digital orthophoto quadrangle (DOQ) of MU campus, MO

USDA National Agriculture Imagery Program NAIP Imagery

Since the 1930's, the USDA Farm Service Agency (FSA) has relied on aerial photography enlargement (photomaps) to accomplish their farm program management requirements. The purpose of the photomap was to provide an accurate geospatial record of farm tract and field boundaries and acreages. The tract and field boundary information was delineated on the photomaps with ink lines and notes, and measured with planimeters, which were linked to tabular data regarding producer information [13].

Beginning in 2003, USDA FSA started the National Agriculture Imagery Program (NAIP) to acquire aerial imagery during the agricultural growing seasons in the continental U.S. on a 5-year cycle, and a three-year cycle began in 2009. NAIP imagery is acquired at 1 meter or 2 meters ground sample distance (GSD) with a horizontal accuracy that matches within six meters of photo-identifiable ground control points. The default spectral resolution is natural color (Red, Green and Blue, or RGB) but beginning in 2007, some states have been delivered with four bands of data: RGB and Near Infrared. This "leaf-on" imagery is used as a base layer for GIS programs in FSA's County Service Centers, and is used to maintain the Common Land Unit (CLU) boundaries. A primary goal of the NAIP program is to make digital ortho photography available to governmental agencies and to the public. NAIP imagery is available for free download through the USDA Geospatial Data Gateway. NAIP is used by many non-FSA public and private sector customers for a wide variety of projects [14]. Figure 1.10 shows the infrared NAIP imagery of Memorial Stadium of University of Missouri in Columbia, MO.

Figure 1.10 Multispectral 1 m Resolution NAIP imagery of Memorial Stadium, Columbia, MO

While the best spatial resolution is one meter for the national aerial photography programs, many state and local governments have acquired even higher resolution aerial imagery for their own communities. For example, The Missouri's Imagery for the State project acquired 2ft spatial resolution leaf-off digital orthophtography for the entire State of Missouri in 2007-2008. A sample 2ft imagery of Memorial Stadium is shown in figure 1.11. City of Columbia, MO acquired 6 inch resolution imagery for the Natural Resources Inventories project to create land cover database to support local planning&

management activities such as natural resource assessment, vegetation mapping, land use analyses, and energy conservation. Again the imagery of Memorial Stadium is shown in figure 1.12. The letter “MISSOURI” and numbers marked on the field are easy to see and the individual football player can also be identified from the field.

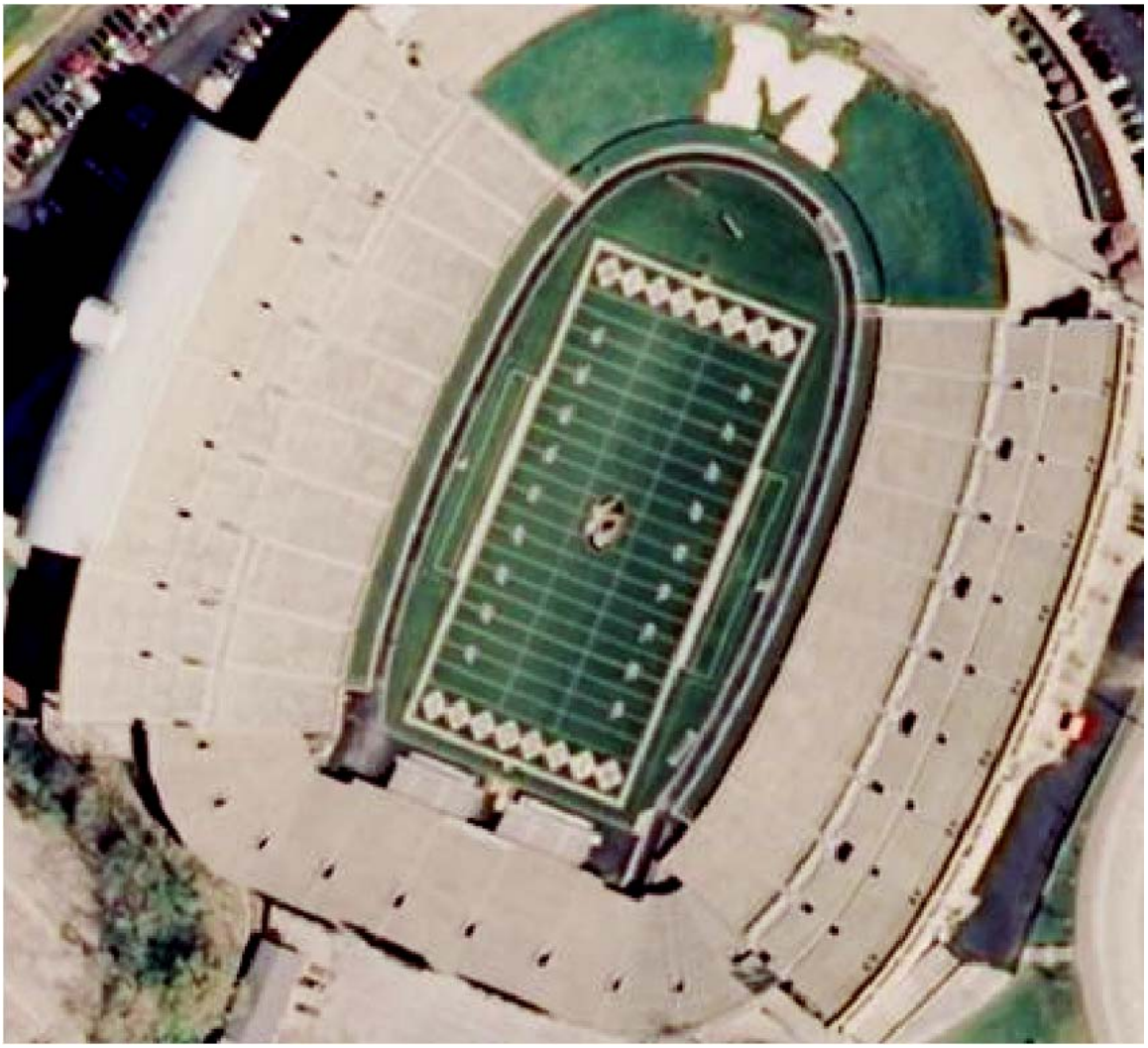


Figure 1.11 2ft resolution multispectral Imagery of Memorial Stadium, Columbia, MO

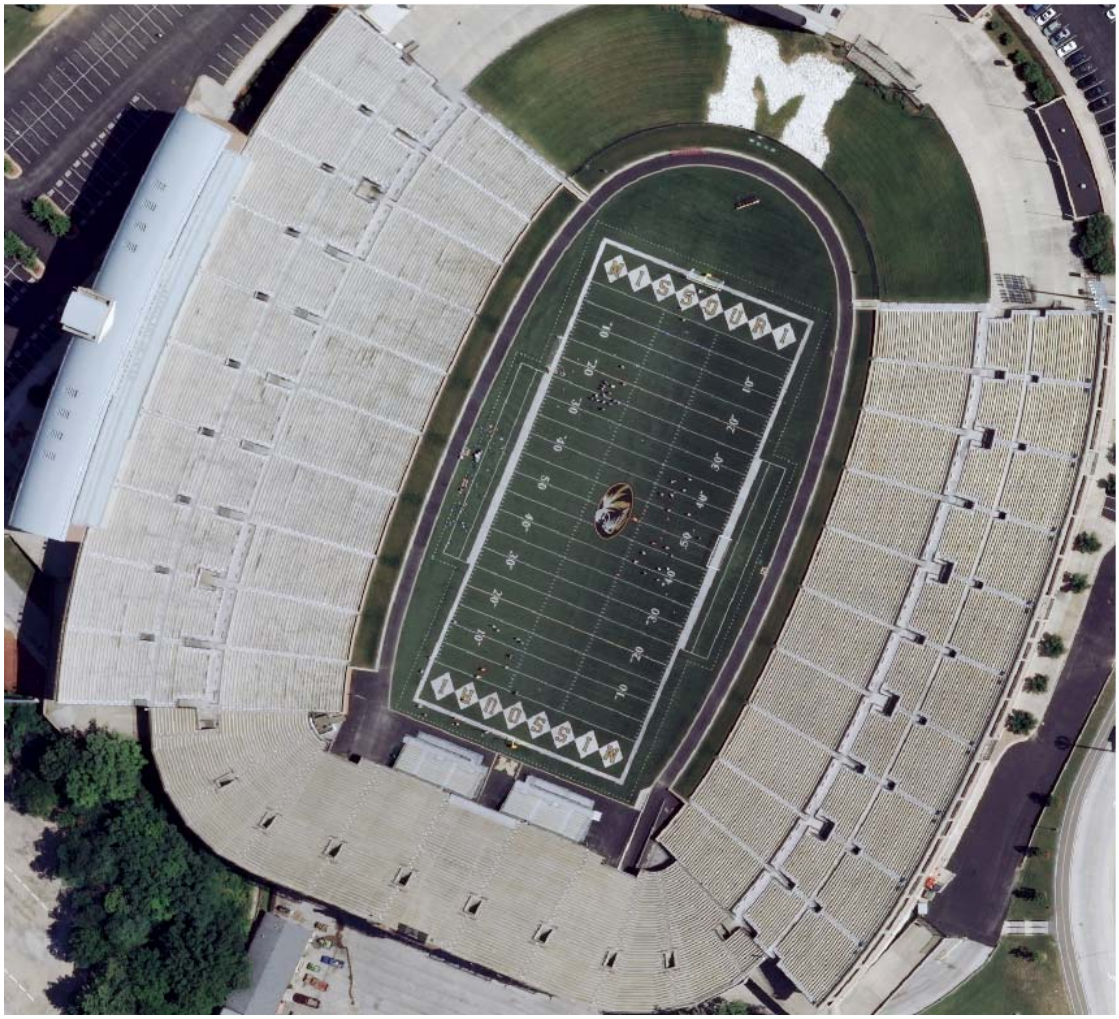


Figure 1.12 Multispectral 6 inch resolution imagery of Memorial Stadium, Columbia, MO

1.5 Need for Conflation

With the increasing availability of large volume of geospatial datasets such as GIS vector layers and remote sensing imagery, there is an urgent need to integrate these multiple datasets together. By fusing spatial datasets from different sources, one can support more comprehensive geospatial analysis that could not have been achieved by the

use of any single dataset in isolation. Also this integration would result in cost savings for many applications.

The National Technology Alliance (NTA) conducted a detailed examination of the needs of US government for conflation technologies [15]. A few key US government activities involving conflation technology are described in the following.

The Engineer Research and Development Center's (ERDC) Topographic Engineering Center (TEC)

ERDC TEC has been overseeing research and development of conflation technology for over ten years. Many of the conflation software packages such as ESEA's MapMerger, Intergraphs' Geomedia Fusion, Northrop Grumman's automated conflation service (ACS), and Digitial Engineering's Conflex were developed in part with TEC funding or under TEC leadership. TEC funded them through small business innovation research (SBIR) contract, DARPA Terrain and Feature Generator project, Advanced Geospatial Management for Information, Integration, and Dissemination (AGMIID) program, US Army's Advanced Concept Technology Automated Feature Extraction contract, Geospatial Information Integration and Generation Tools (GIIGT) Science and Technology Objective [15].

National Geospatial-Intelligence Agency (NGA)

U.S. geospatial intelligence community has moved over the past 10+ years from a "mono- to multi-" environment. This "multi-"environment includes multi-sensors, multi-sources, multi-producers, multi-representation, and multi-answer. However, tools to

support this transition have been lagging behind in development. There are few robust automated tools to integrate and validate Features, Imagery, and Elevation, and other spatial-temporal referenced data sources [15].

Conflation technology is part of NGA's future architecture plans. It is mentioned in both the Enterprise Engineering and the Geoscout documentation. Conflation technology also is stated as a requirement in the Multi-National Geospatial-Intelligence Community Partnership (MGCP), and in a Foundation Based Operations (FBO) environment that implements a Mission Specific Data (MSD) standard derived from the new DGIWIG Feature Data Dictionary (DFDD) [15].

One conflation challenge facing NGA was the MC&G database, which uses obsolete technology that is difficult and expensive to maintain. The MC&G feature database is isolated from newer databases. To "sunset" the MC&G database would be to risk losing data that are not included in the newer databases. The solution was to use conflation technology in a salvage operation. The GIDI data and MC&G data was integrated to derive a "change" dataset, and then ingest the change dataset back into GIDI as an update. ESEA, Intergraph, and Northrop Grumman are working with NGA to add automation to the process. Maintaining the MC&G database was a substantial annual cost. The use of conflation technologies to salvage the MC&G data and retire the system offers a satisfactory solution to a corporate dilemma [15].

U.S. Army

The Department of Defense's current Geospatial Information and Services (GI&S) are inadequate to support current operations or contingencies that may arise in the near

future. Current geospatial data is unavailable or grossly outdated for many key geographic areas of the world where future conflicts may arise. In many cases, current geospatial data is more than 20 years old and not compatible with current models, simulations, systems, or training devices. For example, at the beginning of Operation Enduring Freedom in Afghanistan, the United States Department of Defense (DOD) did not have an existing inventory of current geospatial data available for immediate use. Since traditional data sources cannot provide sufficient geographic coverage, complete content, or sufficient accuracy, a number of data sources (NGA Geospatial Database, digital map, Imagery, enhanced feature data, field data etc.) may need to be fused and/or conflated. Therefore, in the Army Geospatial Data Integrated Master Plan, data fusion/integration/conflation is one of the key recommendations for integrating geospatial operations into the Battle Command network [16].

U.S Geological Survey (USGS)

As one of the cornerstones of the U.S. Geological Survey's National Geospatial Program, the National Map is a collaborative effort among the USGS and other federal, state, and local partners to improve and deliver topographic information for the Nation. The geographic information available from the National Map includes orthoimagery, elevation, geographic names, hydrology, boundaries, transportation, structures, and land cover. The National Map will serve as a foundation for integrating, sharing, and using spatial data easily and consistently and will provide a new approach to provide more current information while retaining and improving other valued characteristics, such as positional accuracy and content completeness [17].

In the past, USGS played the primary role in the United States for the collection of basic geospatial data through the USGS topographic map. Today current, high-resolution geospatial data often reside with state, local, and tribal governments, the private sector, and other federal agencies. These organizations collect, manage, and store geospatial data in response to their specific needs. In general, they have no mission requirement to make these data available for secondary use or integrate the data on a national scale.

Working with federal, state, local partners, USGS is building the National Map as a framework for geographic knowledge that will provide current, accurate, seamless, and nationally consistent digital geospatial data. Data will be comprised of the best available sources and the USGS will depend on state, local, tribal, and other governments and private industry to supply data. The USGS will become a data producer only in cases where no other data are available [18].

However, one primary complexity is the integration of the various resolutions and accuracies of data in both horizontal and vertical directions, which is one of massive proportion when considering national implementation. Combining spatial data from various, disparate sources present many potential data integrity problems [18]. Spatial data are not represented consistently among organizations with regard to schema, attribution, or geometry. For transportation networks, there may be significant overlap in geographic extent between neighboring partners with many of the same road segment features represented differently. Geometric inconsistencies across datasets also lead to topological accuracy issues when multiple data themes from multiple sources form a composite map for a particular geography. Data integration is a significant problem for

The National Map. Therefore, developing conflation technology for successful integration in the midst of such heterogeneity is the challenge of the National Map.

U.S. Census Bureau

The Bureau of the Census developed the Topologically Integrated Geographic Encoding and Referencing (TIGER) database to provide geographic services and products in support of the 1990 decennial and other Census Bureau statistical programs. TIGER was built and has been continuously updated using a wide variety of source materials and techniques, including the GBF/DIME files, USGS 1:100,000-scale topographic maps, local and tribal maps, and enumerator updates of differing positional accuracy [19].

The U.S. Census Bureau's mission to count and profile the Nation's people and institutions does not require very high levels of positional accuracy in its geographic products. It was not necessary to produce positional accurate files. The main concern was the location of geographic features in relation to other geographic features. Its files and maps are designed to show only the relative positions of elements.

Coordinates in the TIGER/Line files are in decimal degrees and have six implied decimal places. The positional accuracy of these coordinates is not as great as the six decimal places suggest. The positional accuracy varies with the source materials used, but at best meets the established National Map Accuracy standards (approximately +/– 167 feet) where 1:100,000-scale maps from the USGS are the source. The U.S. Census Bureau cannot specify the accuracy of feature updates added by its field staff or of features derived from the GBF/DIME-Files or other map or digital sources. Thus, the

level of positional accuracy in the TIGER/Line files is not suitable for high-precision measurement applications such as engineering problems, property transfers, or other uses that might require highly accurate measurements of the earth's surface [19].

Whereas relational accuracy once was adequate for census activities, changing goals and technologies require that the Census Bureau improve the positional accuracy of TIGER. One of the Bureau's goals for the next decade is to capture the latitude and longitude coordinates for living quarters and to equip each field interviewer with portable computers equipped with Global Positioning System (GPS) technology to accurately locate living quarters requiring a visit. However, to integrate the more accurate coordinates that GPS can provide for living quarters (and the streets they are along) with the existing MAF (master address file) /TIGER System, current TIGER features must have an equivalent level of positional accuracy. Additionally, the Census Bureau has found that many local GIS files have a greater positional accuracy; hence, the current positional accuracy of TIGER is a limiting factor. It precludes more effective address lists and geographic information partnerships with those state, local, and tribal governments that have high quality address, street, boundary and related geographic information. The successful partnerships with state, local, and tribal agencies rely upon improving the positional accuracy of TIGER [20].

Federal Emergency Management Agency (FEMA)

Floods are among the most frequent and costly of all natural disasters and have great impact in terms of economic and human losses each year. Communities with effective and accurate flood maps are prepared to prevent not only economic devastation,

but to preserve life as well. FEMA manages the National Flood Insurance Program, which was created in 1968 to reduce flood damage by identifying flood risks, encouraging sound floodplain management practices, and providing a mechanism through which citizens can insure their investments. To help determine insurance requirements, FEMA maintains an inventory of over 90,000 flood map panels that detail areas at risk; identify where flood insurance is needed; and help limit construction within flood zones [21].

These flood maps were initially intended to be used principally by insurance agents, floodplain managers, and others charged with implementing the National Flood Insurance Program. However, over the years the flood maps have come to be much more widely used for many purposes, including local planning, emergency preparedness and response, and natural resource management. It has become essential tools for a much wider range of users, from builders and developers to real estate agents and lenders to local planners and citizens attempting to make informed decisions about the degree of flood risk faced by particular pieces of property [21].

Natural processes and human interventions lead to changes in flooding and floodplains. The addition of pavement, impervious roofs, and other forms of development alters the natural drainage patterns and the timing and volume of runoff to floodplains, leading to changes in the size and extent of floodplains. Therefore, flood maps need to be updated accordingly. However, the majority of FEMA's flood maps are outdated and in unalterable paper format that are difficult to update.

In response to demands for more accurate mapping products, FEMA has embarked on flood map modernization program to update and digitize the nation's flood

maps. The objective of flood modernization program is to convert 80% of existing paper map panels to digital format with a high-quality base map, update 20% of existing panels with new flood risk information while converting them to digital format, add 13700 completely new panels to cover previously unmapped communities [21]. In addition, digital flood insurance rate maps require a digital base map that reflects reference features (i.e., roads, street, hydrographic features, political boundaries) needed by users to locate properties on maps.

The map modernization plan includes strategies for working with other federal, state, local government entities. FEMA has established and implemented initiatives to build federal, state, regional, and local partnerships to formalize cooperation in the flood hazard identification and mapping processes. FEMA relies on the combined contributions of FEMA staff and flood mapping partners to obtain and maintain accurate, up-to-date flood hazard information. FEMA will maximize the use of such partner's contributions as a mean of leveraging limited public funds to the fullest extent possible. For example, where accurate local community vector data is available, community vector data will be utilized to show base map data as backdrops to flood data layers. If accurate community data is not available, digital aerial photography (DOQs) will be utilized as the base map layer [21].

However, the data are from different sources and are created for different purposes by using different specifications. Therefore, integrating the multiple datasets from different sources to update flood maps is a challenging task.

Geospatial Integration/conflation

Geospatial technology has provided new ways to collect, maintain, access, and use basic spatial data. More and more organizations adopt GIS as a spatial data management tool. Rapid adoption of GIS technology has not come without cost. Many organizations employing GIS have developed and maintained spatial databases mostly independent of other organizations using GIS. This is apparent across all levels of government. One example of this problem is the existence of multiple transportation databases among government agencies - each one designed to fulfill a particular agency mission. The Topologically Integrated Geographic Encoding and Referencing system (TIGER) data support the Bureau of the Census mission; Digital Line Graphs support the basic topographic mapping mission of the United States Geological Survey (USGS), while the Bureau of Transportation Statistics maintains the National Transportation Atlas Database. Further, many states and counties have transportation networks that they maintain autonomously. These datasets cover many of the same geographic extents and real-world features, but have been designed and maintained disparately within distinct fiscal budgets. The result is not only redundancy but also diminished data quality. Moreover, with budget constraints commonplace, it is very difficult for any one agency to maintain a current spatial dataset for geographies of significant extent [22].

The availability of geospatial data from multiple sources requires the integration of this information before using it effectively. However, accurately integrating geospatial data from different sources is a challenging task. Current data integration strategies and methodologies have not kept pace with advances in data collection. It remains difficult to analyze even two spatial data sets acquired at different times, for different purposes,

using different datums, positional accuracy (x,y,z), classification schemes, and levels of *in situ* sampling or enumeration precision [23]. This is because such geospatial data may have different levels of accuracy and precision in their attributes as well as their spatial and temporal dimensions. Furthermore, they may have been created for different purposes. All of this leads to discrepancies in the representation of the same features from different sources. In addition, the spatial displacement is often non-systematic such that a simple global transformation will not solve the problem.

Our Conflation Research

To attack the conflation problem, we developed an innovative conflation approach consisting of several algorithms. Each algorithm is designed to solve a sub-problem. Appropriate combination of the algorithms has been applied to vector-to-vector and vector-to-imagery conflation. Real world datasets such as TIGER roads from U.S. Census Bureau, MODOT roads from Missouri Department of Transportation, Vector parcel data from county assessor's office, and high-resolution aerial photograph and Satellite imagery are used in our experiments.

In chapter 2, we proposed a hybrid approach based on the combination of the traditional conflation and the snake algorithm. The feature matching and rubber-sheeting map alignment processes remain the same as traditional conflation and the attribute transfer process is replaced by snake alignments. The snake algorithm moves the TIGER roads towards the high accuracy roads. This has advantages over traditional conflation approaches.

While a simple proximity based feature matching was used in chapter 2, a detail study of point feature matching is proposed in chapter 3. A new relaxation labeling based point feature matching approach is developed. Its performance is compared with proximity based matching method.

Since imagery is acquired more frequently and more current, conflating vector data to imagery is needed. An innovative vector-to-imagery conflation approached is presented in chapter 4. This approach uses road intersection detection, relaxation based feature matching, rubber-sheeting alignment and snake algorithm to solve the vector-to-imagery conflation problem.

In chapters 2-4, we use the road centerlines for conflation. We extend our approach to solve parcel conflation problem in chapter 5. The same set of road intersection detection, feature matching, and rubber-sheeting algorithm is applied to migrate the digital vector parcel data to high resolution imagery.

In summary, chapter 3 focus on the critical sub-problem of point feature matching. The vector-to-vector and vector-to-imagery road conflation approaches are presented in chapter 2 and chapter 4 respectively. The scale-up experiment is presented in chapter 5. The vector-to-imagery road conflation approach is extended to conflate parcel polygon data to imagery in chapter 6. All the problems and solutions are interconnected in those chapters.

Chapter 2

A Snake Based Approach for TIGER Road Data Conflation

The TIGER (Topologically Integrated Geographic Encoding and Referencing) system has served the U.S. Census Bureau and other agencies' geographic needs successfully for two decades; however poor positional accuracy has made it extremely difficult to integrate TIGER with advanced technologies and data sources such as GPS, high resolution imagery, and state/local GIS data. In this chapter an alternative solution to conflate TIGER road centerline data with other geospatial data is presented. The first two steps of the approach (feature matching and map alignment) remain the same as traditional conflation. Following these steps, a third is added in which active contour models (snakes) are used to automatically move the vertices of TIGER roads to high accuracy roads, rather than transferring attributes between the two datasets. This approach has benefits over traditional conflation methodology. It overcomes the problem of splitting vector road line segments and can be extended for vector-imagery conflation as well. Thus, a variety of data sources (GIS, GPS, and Remote Sensing) could be used for the TIGER improvement. Preliminary test results indicate that the approach performs

very well. The positional accuracy of TIGER road centerline can be improved from an original 100 plus meters' RMS error to only 3 meters. Such improvement can make TIGER data more useful for much broader applications.

2.1 Introduction

The U.S. Census Bureau's TIGER database is primarily used to support the mapping, geographic analysis, and other GIS activities that serve the various censuses and surveys conducted by the Bureau. Initial sources used to create the database were the U.S. Geological Survey (USGS) 1:100,000-scale Digital Line Graph (DLG), the U.S. Census Bureau's 1980 geographic base files (GBF/DIME-Files), and a variety of miscellaneous maps. In order to maintain a current geographic database, TIGER has been continuously updated using a wide variety of source materials and techniques. Some updates have come from map annotations made by enumerators as they attempted to locate living quarters by traversing every street feature in their assignment area. The Census Bureau digitized these updates directly into the TIGER database without geodetic controls or the use of aerial photography to confirm the features' positional accuracy. Other corrections and updates were supplied by local participants in various Census Bureau programs. Maps were sent to participants for use in various programs, and some were returned with updated annotations and corrections. The Census Bureau generally added the updates to the TIGER database without extensive checks. Changes made by local officials did not have geodetic control [19].

Whereas relational accuracy once was adequate for census activities, changing goals and technologies require that the Census Bureau improve the positional accuracy of

TIGER. One of the Bureau's goals for the next decade is to capture the latitude and longitude coordinates for living quarters and to equip each field interviewer with portable computers equipped with Global Positioning System (GPS) technology to accurately locate living quarters requiring a visit. However, to integrate the more accurate coordinates that GPS can provide for living quarters (and the streets they are along) with the existing MAF (master address file) /TIGER System, current TIGER features must have an equivalent level of positional accuracy. Additionally, the Census Bureau has found that many local GIS files have a greater positional accuracy; hence, the current positional accuracy of TIGER is a limiting factor. It precludes more effective address lists and geographic information partnerships with those state, local, and tribal governments that have high quality address, street, boundary and related geographic information. The successful partnerships with state, local, and tribal agencies rely upon improving the positional accuracy of TIGER [20].

The Census Bureau launched the MAF/TIGER enhancement program in 2000. The first strategic objective was to improve address/street location accuracy and implement automated change detection. The Census Bureau sought to achieve a high level of map coordinate accuracy in TIGER by acquiring and using, as a first priority among data sources, digital files prepared and provided by state, local and tribal governments [24].

Conflation approaches are often used to improve the TIGER road data. However, the traditional approach requires two vector datasets. It could not make use of the high resolution imagery that is often readily available and more current. In addition, to deal

with non one-to-one feature matching, splitting road line segments or attributes are often performed manually by human operator which is labor intensive and time-consuming.

In this chapter we propose a new hybrid conflation approach that combines traditional conflation with active contour models (snakes). Snakes are deformable contours that can move iteratively under the influence of internal forces and external image forces. The basic processes are feature matching, map alignment, and position correction by snakes. The next section presents an overview of traditional conflation methodology. This is followed by our snake based conflation approach. Experimental results are then presented, followed by discussions and conclusions.

2.2 Traditional Conflation Methodology

Although local GIS data may have better positional accuracy than TIGER data, most do not have richer attributes. To get the best spatial and attribute information from both sources, conflation technology was developed. Conflation is the process that combines two spatial datasets of the same region to produce a superior dataset that is better than either source in spatial and attribute aspects. Through the conflation process individual strengths of the source datasets can be combined. A dataset with excellent spatial accuracy but little attribute information can be merged with one with rich attribute information but poor spatial accuracy to produce a new dataset that is both spatially accurate and attribute rich.

The history of conflation goes back to the early 1980's, where the first development and application of an automated conflation process occurred during a joint U. S. Geological Survey - U.S. Census Bureau project designed to integrate the agencies'

respective digital map files of U.S. metropolitan areas [25]. The implementation of a computerized system for this task provided an essential foundation for much of the theory and many of the techniques used today. Since then, much progress has been made in both research and application. Existing local GIS data were utilized to update and enhance the TIGER database [26-27]. Cobb et al. [28] proposed a rule-based conflation approach to the National Imagery and Mapping Agency's (NIMA) Vector Product Format (VPF) datasets. Walter and Fritsch [29] developed a statistical approach to match road network data of the European standard Geographic Data File (GDF) and the German topographic cartographic spatial database. Samal et al. [30] utilized similarity measures for attribute matching and proximity graphs to model geographical context in conflating features manually extracted from topographic maps and tourist maps. Rahimi et al. [31] presented a system design for a multi-agent-based infrastructure to perform distributed conflation. Filin and Doytsher [32] developed a local transformation algorithm based on counterpart linear features instead of point features.

However, the basic idea and procedures of conflation remain the same. Generally the conflation process has the following steps:

2.2.1 Feature Matching

Feature matching is the core strategy to identify the correspondence of features from two different datasets as representations of the same geographic object. There are three basic types of map features: point, line and polygon. Different matching criteria or methods could be applied to these feature types, with the most commonly used criteria being geometric, topological and attributes [33].

Geometric criteria include length, perimeter, area, shape, angle, distance etc. Distance (proximity or nearness) between features is the most straightforward way to match common objects. A feature within a certain range of another feature in the second dataset has a better chance of representing the same object. Euclidean distance is often used to calculate the proximity of point features. Hausdorff distance is more suitable for line to line matching. Other geometric criteria such as angular information of linear objects, or shape similarity of lines or polygons (length, perimeter, area etc) are also useful matching criteria [33].

Topology explicitly defines spatial relationships. The principle in practice is quite simple; spatial relationships are expressed as lists (e.g., a polygon is defined by the list of arcs comprising its border). In many GIS systems the vector data is stored using arc-node data structure. The arc-node structure stores and references data so that nodes construct arcs and arcs construct polygons. Nodes define the two endpoints of an arc. An arc is composed of its two nodes and an ordered series of points called vertices that define its shape. Nodes and vertices are represented as x, y coordinates. The arc-node data structure supports three major topological concepts [34]: 1) Connectivity: arcs connect to each other if they share a common node. 2) Area definition: arcs that connect to surround an area define a polygon. 3) Contiguity: arcs have direction and left and right sides. Two geographic features which share a boundary are called adjacent. Polygons are contiguous if they share a common arc. Contiguity is the topological concept which allows the vector data model to determine adjacency. The topology should be consistent in order to match features.

Attribute criteria. If two datasets have common attribute fields and the semantics of both datasets are known, the attribute item can be used to find the corresponding match. If there is conflict or inconsistency, human operators have to deal with it interactively.

2.2.2 Map Alignment

For a linear feature dataset such as the TIGER road centerline, node (road intersections, dead-ends) matching is the easiest strategy to define and implement in a manner that permits topological consistency.

After identifying the matched node pairs, a rubber-sheeting transformation is often used to align the two datasets. A rubber-sheet transformation is local affine transformation which subdivides the map areas into triangular-shaped pieces and applies local adjustments on each single piece. The Delaunay triangulation is one of the most frequently used triangulation methods for rubber-sheeting. A Delaunay triangulation partitions the control point set into a group of triangular regions with the property that no point falls in the interior of the circumcircle of any triangle (the circle passing through the three triangle vertices). The Delaunay triangulation maximizes the minimum angle of all the angles in the triangulation, thus avoiding triangles with extremely small angles [35].

Matched nodes serve as vertices for the triangles of the space. After rubber-sheeting, these nodes are moved into perfect alignment and the displacements of other intermediate points of road segments are interpolated according to the movements of the vertices of the triangles which contain them.

2.2.3 Attributes Transfer

After feature matching and rubber sheeting, the selected attributes of features in the source dataset (low positional accuracy, rich attributes) may be copied and transferred to the matched features in the destination dataset (high accuracy, poor attributes). As a result the conflated dataset will have improved spatial accuracy and richer attributes.

While attribute transfer is a relatively trivial task if a one-to-one correspondence exists between the source and destination datasets, such ideal cases are rare. In the real world, many scenarios arise. For example, one feature in the source may have many corresponding features in the destination dataset, or many features in the original can have a single corresponding feature in the destination. Figure 2.1 shows a very simple case of different matching scenarios.

For the traditional conflation approach, to deal with a many-to-one match, the feature in the destination needs to be split into smaller segments to create several one-to-one matches, allowing the desired attributes of the source to be transferred. For the one-to-many match, some attributes, such as address range in the source dataset, may need to be split, and then transferred to target features. Both spatial and attribute splitting have to be done in many-to-many matching cases. The manual splitting operation is time-consuming. In addition, many road datasets may have a route system defined in terms of its existing segments. The splitting of segments would likely corrupt any defined route system. Manually regenerating route systems after conflation is undesirable [36]. The snake-based conflation approach alleviates the problem of segment splitting. It iteratively moves the vertices of line segments in source dataset toward the target position, so the positional accuracy is improved.

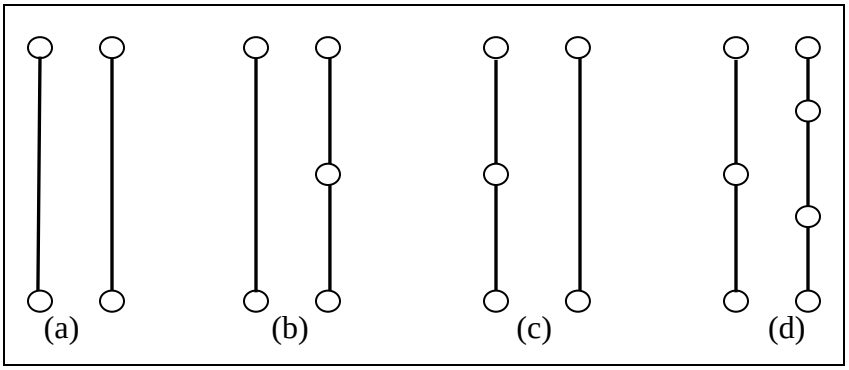


Figure 2.1 The different matching scenarios. a) one-to-one match b) one-to-many match c) many-to-one match d) many-to-many match

2.3 Snake Based Conflation Approach

The concept of snake was introduced by Kass et al. [37]. They defined a snake as an energy-minimizing spline, guided by external constraints and influenced by image forces that pull it toward features such as edges and lines. Snakes are active contour models; they lock onto nearby edges, localizing them accurately.

Snakes have been used extensively in many image processing and computer vision applications [38-43]. In the geospatial community, snakes have been used as a tool for road extraction or road database updating from imagery.

Gruen and Li [44] utilized LSB-Snakes (Least Squares B-Spline Snakes) to extract road features from digital images. A human operator identifies the roads from the image and provides a few coarsely distributed seed points along the roads. With these seed points as an approximation of the roads' position and shape, the roads are extracted automatically by LSB-Snake. Baumgartner et al. [45] and Laptev et al. [46] proposed an approach for automatic road extraction in aerial imagery that exploits the scale-space

behavior of roads in combination with geometrically constrained snake-based edge extraction, using snake approach to bridge shadows and partially occluded areas between two road segments.

Klang [47] developed a method for detecting changes between an existing road database and a co-registered satellite image. First he used the road database to provide approximation for subsequent road detection using a snake algorithm to correct road location. Then he ran a statistical line-following process to detect a new road starting from the existing network. Fortier et al. [48] extended the above approach by using road intersections. Road intersections improve matching between the road database and the lines on the image, with hypotheses for new road segments generated from these line junctions. Agouris [49] expanded the model of snakes to function in a differential mode by introducing an additional energy term that describes the discrepancy between the current snake solution and the pre-existing road shape information. This method combines object extraction and change detection to update and improve the pre-existing GIS information for that object. Bentabet et al. [50] presented an approach for road detection from synthetic aperture radar (SAR) images and road databases. The vectors provided by the database give an initial estimation of the road location which is refined through snakes. Neuenschwander et al. [51] presented the ziplock snake model. The optimization process for a ziplock snake starts from the two endpoints and processes toward the center of the snake. During the process the image potential is progressively turned on to clamp the two ends of the snake onto an image contour. The ziplock snake model reduces the initialization requirement, one of the main problems for any snake algorithm.

In the following section, we propose a new conflation approach that combines traditional conflation with snakes. The basic processes are feature matching, map alignment, and position correction by snakes. The first two steps are the same as in traditional conflation. In the third step we use snakes to correct the road position instead of attribute transfer.

2.3.1 Feature matching

Geometric and topological criteria consisting of node matching and line matching are used for feature matching. The source data are TIGER roads and the target data are MODOT (Missouri Department of Transportation) roads, derived from a high accuracy GPS survey with 3 m root mean square (RMS) error. Both datasets are stored with arc-node data structure and the topology is created in Arc/Info. First, node matching is conducted using all the nodes (road intersections and end points) in MODOT roads as the reference and finding their corresponding nodes in the TIGER dataset. The matching criteria are nearness between the node pairs and the number of emanating lines from the nodes. Second, the matched nodes are used for line segment matching. Several simple rules are proposed. If a dangling line segment has an unmatched dangling node, this segment cannot be matched. If both nodes of a segment are matched, then this segment is a matched line. If a segment is on the shortest path which connects two matched nodes, this segment is a matching candidate.

Figure 2.2 illustrates our matching method. For each node in the MODOT data, we find all nodes in TIGER roads within a specified search radius and compute their distances. In this example, the search radius is set to 400 m. For MODOT node 4, two

nodes, 11 and 12 are found from TIGER data with distance of 330 m and 198 m respectively. Next we examine the node types i.e. end point (dangle node), three-way or four-way intersections etc. by counting the number of emanating arcs from each node. This information can be easily obtained from topology. Node 4 and 12 have one connecting arc, and node 11 has three emanating arcs. Therefore, node 11 is out, and node 4 and 12 are correspondence. If more than two nodes of the same type are found, then the closest node has preference, but the general arc direction is considered also.

Suppose after node matching, node 12 is matched to node 4, and nodes 8, 9, 10, 11 are unmatched. Then, the dangling line segments connecting nodes 8-9, 10-11 can be marked as unmatched lines. The remaining two longer line segments starting with nodes 4 and 12 are the matched road lines.

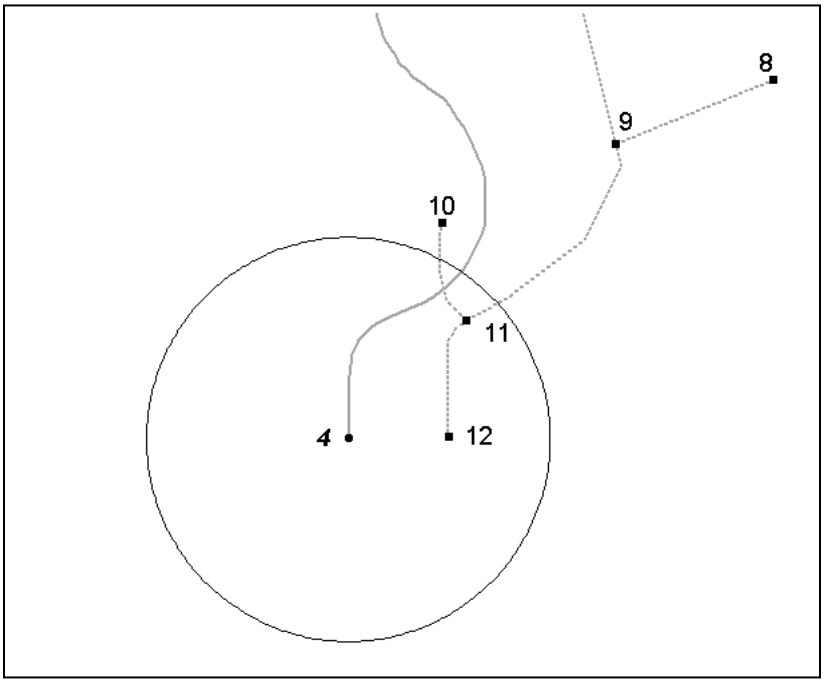


Figure 2.2 Feature matching by geometric and topology criteria. For each node in the MODOT road (solid line), all nodes in TIGER data (dashed line) within 400 m radius are found. The node matching is based on nearness and node types.

2.3.2 Map alignment

The matched nodes are used as control points to perform the rubber-sheeting transformation, bringing these two datasets closer to each other. More importantly, the matched nodes in TIGER roads are moved to the exact positions of their corresponding nodes in MODOT roads. This facilitates the following snake operation.

2.3.3 Position correction using snakes

The snake is an active contour model under the influence of internal and external forces. The internal force imposes a piecewise smoothness constraint. The external image force pushes the snake toward salient image features like lines and edges [37]. The snake can be represented as a parametric curve by

$$v(s) = (x(s), y(s)) \tag{2.1}$$

where s is proportional to the curve length, and x and y are the curves coordinates.

The snake's total energy function is composed of internal and external components and is given by

$$E_{Snake} = \int_0^1 E_{Snake}(v(s))ds = \int_0^1 E_{Internal}(v(s)) + E_{External}(v(s))ds \tag{2.2}$$

and in discrete form:

$$E_{Snake} = \sum_{i=1}^{n} E_{Internal}(i) + E_{External}(i) \tag{2.3}$$

Here, *i* represent a vertex point of the vector road segment and n is the total number of vertices of each snake (vector road polyline).

The internal energy is usually based on the first and second derivative of the curve, constraining the snake to be smooth. In [37], the internal energy is represented as:

$$E_{Internal} = (\alpha(s) \mid v_s(s) \mid^2 + \beta(s) \mid v_{ss}(s) \mid^2) / 2 \tag{2.4}$$

Where $v_s(s)$ and $v_{ss}(s)$ are first and second derivatives. In discrete form (4) becomes:

$$E_{Internal}(i) = (\alpha_i \mid v_i - v_{i-1} \mid^2 + \beta_i \mid v_{i-1} - 2v_i + v_{i+1} \mid^2) / 2 \tag{2.5}$$

The internal energy is composed of a first-order term and a second-order term. The first term keeps the snake from stretching or contracting along its length (elasticity) and the second term curvature keeps the snake from bending too much. $\alpha(s)$ and $\beta(s)$ are functions of the arc length along the snake and are used to weight the relative importance of the elasticity and bending. Usually, $\alpha(s)$ and $\beta(s)$ can be replaced by constants α and β to simplify computations.

The external energy represents the image force attracting the snake. It depends on image structure, usually the features of interest. The image energy can be defined as

$$E_{External} = -E_{\mathrm{Image}}(v(s)) \tag{2.6}$$

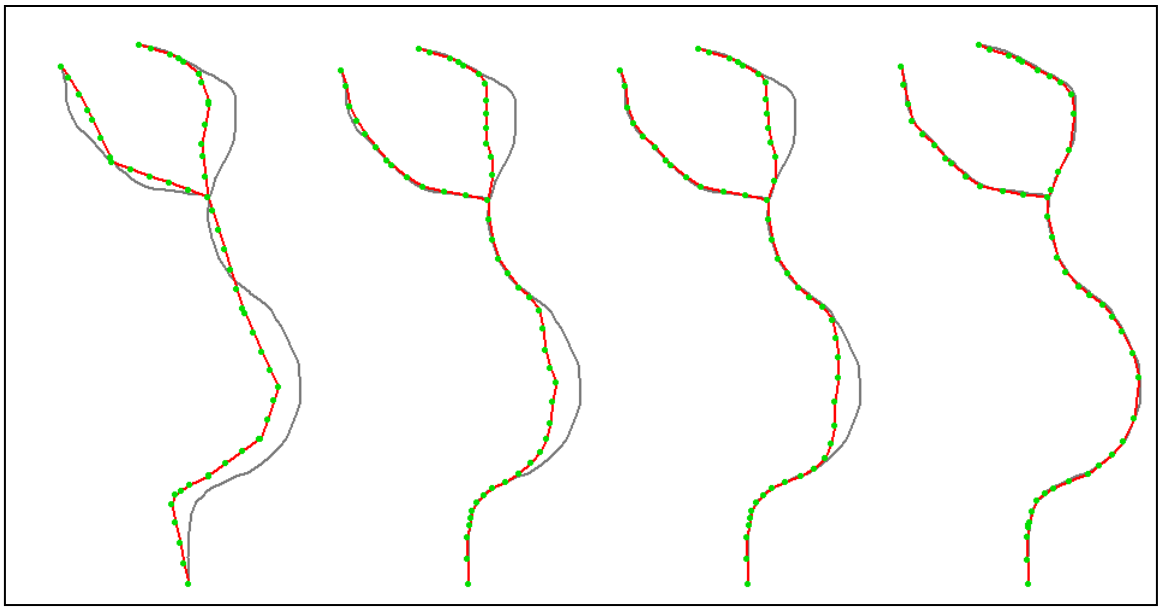
Where $E_{Image}(v(s))$ is a function where high values correspond to the feature of interest. Usually, $E_{Image}(v(s))$ can be the image intensity itself, I(x, y), or the magnitude of the image gradient, $|\nabla I(x, y)|^2$. Then the snake is attracted to edges with large image gradients. In general, the external energy is calculated from some feature image.

To compute the minimum of the snake energy function, we used the approach shown in [37]. The energy minimization is solved using the Euler-Lagrange differential equation of motion with a discrete representation of the energies. The minimization is solved iteratively until convergence is reached. Detailed equations can be found in the appendix of Kass et al. [37].

We have two vector road datasets: one is the TIGER road centerline, the other being the high positional accuracy MODOT road centerline data, which are derived from

GPS surveys by Missouri Department of Transportation. Since the MODOT roads have an estimated Root mean square (RMS) error of about 3 meters, we first create a 3 meter's buffer area to each MODOT road, then translate the buffers into image (raster format) so that road pixels have value of 1 and background (non-road) pixels have value of 0. The grid cell size is set as 1 meter. Then we simply use this MODOT road buffer image as external energy in (2.6).

Each TIGER road segment (a polyline) has two nodes (endpoints) and may have multiple intermediate vertices. After the rubber-sheeting transformation, these two nodes have been aligned correctly with the reference. But many of the intermediate vertices are not (figure 2.3a). Each vector road segment is treated as a snake. If a vertex falls within the MODOT road buffer image, it won't move. But if a vertex falls outside the MODOT road buffer image (in background) where the external image energy is 'high', it will move toward the MODOT road image where the external energy is 'low'. The road image attracts the vertices toward it. Therefore the road vertices will iteratively move into better alignment.



(a) (b) (c) (d)

Figure 2.3 Illustration of the movement of snake. a) After the rubber-sheeting transformation, the nodes of each TIGER line segment (red) are in perfect alignment with the MODOT road image (grey), but most of the intermediate vertices (green) are not. b) The vertices are moved closer to road buffer image by snake after two iterations. c) Similar results after four iterations. d) Final vector roads are in very good alignment with image.

2.4 Experimental Results

The snake-based conflation approach described above has been successfully implemented in a PC windows environment. The snake algorithm was written in Visual C++ and the remaining operations were performed in Arc/Info.

We obtained TIGER road centerline data from U.S. Census Bureau and GPS-derived road data from the Missouri Department of Transportation. Also, we obtained U. S. Department of Agriculture (USDA)'s NAIP (National Agricultural Imagery Program)

imagery with 1 meter resolution. All three datasets are georeferenced in the same coordinate system and can be superimposed together. The MODOT road data has an estimated RMS error of about 3 m and the NAIP image has less than 5 m accuracy. The TIGER road data has a varied error distribution across the test area.

Figure 2.4(a) shows one section of test area in which the TIGER roads and the MODOT roads are displayed together. The discrepancy between the two datasets is obvious, with TIGER roads having a few more road segments. We conducted node matching using all the nodes (road intersections and end points) in the MODOT roads as the reference, determining their corresponding nodes in the TIGER dataset. The matching criteria are nearness between the node pairs and the number of emanating lines from the nodes. The matched nodes are used to check line segment matching using the rules described above. Figure 2.4(b) shows the matched road segments.

Using the matched nodes as control points, a piecewise rubber sheeting transformation is performed to bring the TIGER roads closer to the MODOT roads forcing the matched TIGER nodes to the exact position of their corresponding nodes in the MODOT layer. The results of rubber sheeting are illustrated in figure 2.4(c).

The vector MODOT road buffers are converted into raster format and used as an image with a pixel size of 1 meter. The snake algorithm moves the vertices of TIGER roads toward MODOT road image. Figure 2.4(d) shows the results of snake operation.

Another section of the test area is shown in figure 2.5. The discrepancy between the two datasets ranges from 2 meters to 380 meters. The rubber-sheeting transformation drops the discrepancy to within 100 meters. Our snake algorithm corrects the position of

TIGER roads and produces a product with high positional accuracy, close to the accuracy of MODOT roads.

From figure 2.4 and figure 2.5, it can be seen that the positional accuracy of the TIGER data has been improved greatly. To show this improvement quantitatively we used all the vertices of TIGER roads as the check points and the MODOT roads as reference to estimate the RMS errors of original TIGER roads (only matched road segments), TIGER roads after rubber sheeting transform, and final TIGER roads after snake processing. The improvements are dramatic. The original RMSE for the two test sections are 26 m and 124 m respectively. The rubber-sheeting transformation dropped the RMSE to 12 m and 26 m. Finally the snake correction further improved the accuracy, with RMSE of about 3 m, an incredible accuracy improvement.

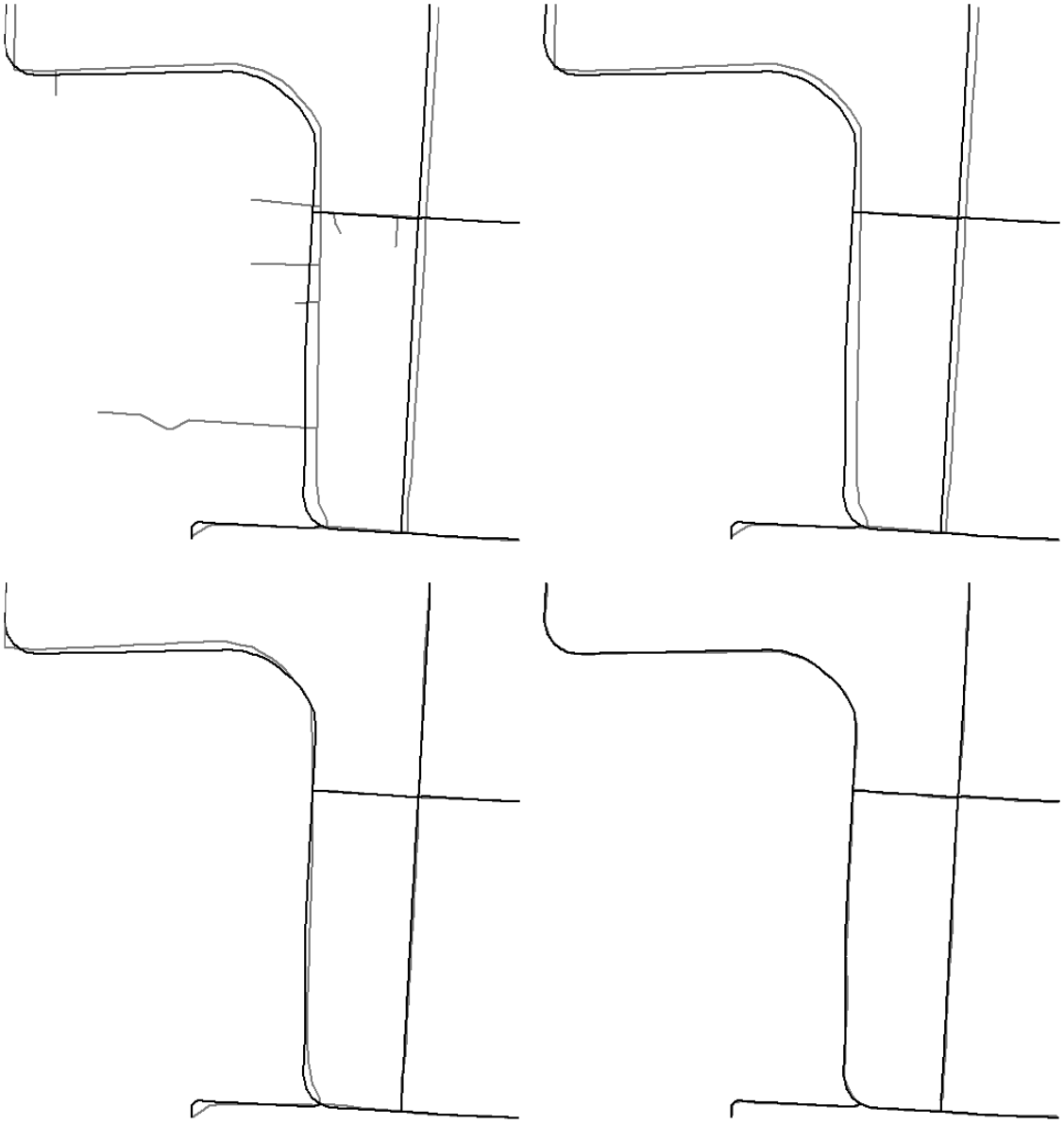
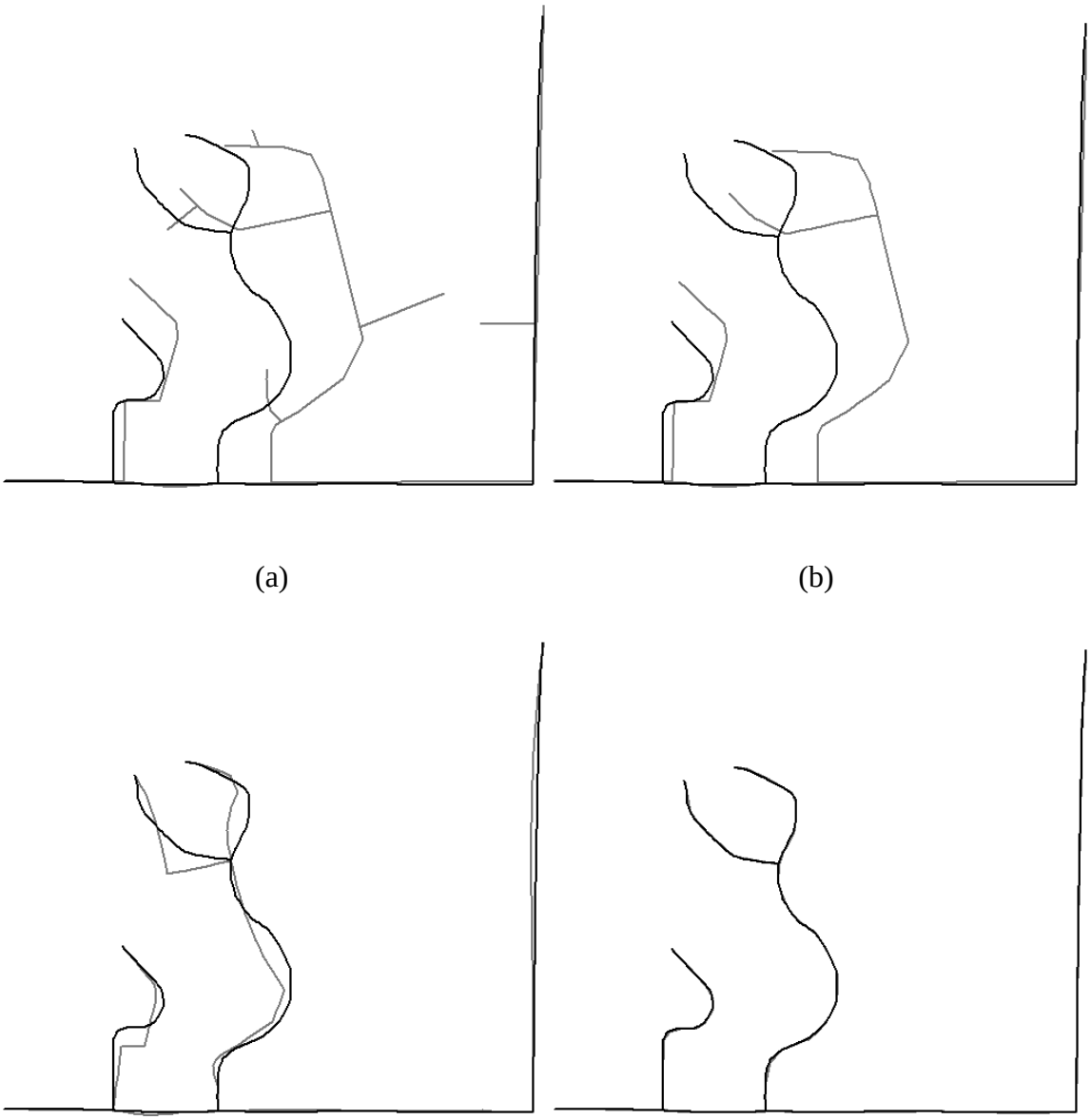


Figure 2.4 One section of the test area (TIGER roads are showed as grey, MODOT roads are black). a) Original TIGER and MODOT roads displayed together. b) Matched road segments. c) Results after the rubber-sheeting transform. d) Final results after the snake alignment.



(a) (b)

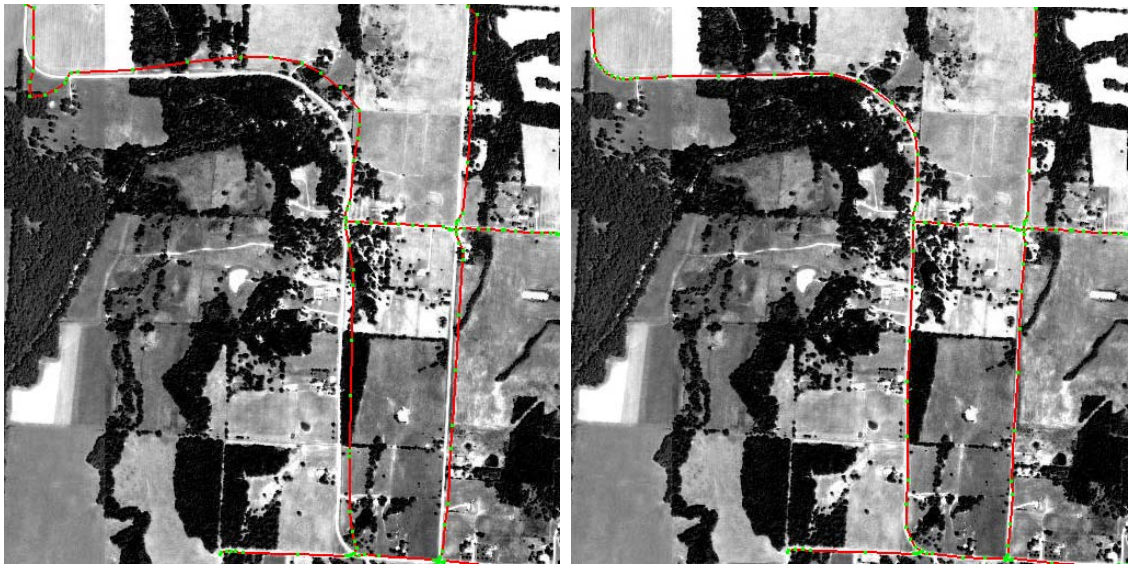
(c) (d)

Figure 2.5 Another section of the test area (TIGER roads are showed as grey, MODOT roads are black). a) Original TIGER and MODOT roads displayed together. b) Matched road segments. c) Results after the rubber-sheeting transform. d) Final results after the snake alignment.

2.5 Discussions and Conclusions

Conflating vector datasets together has been a subject of research for over two decades and strides are being made. However, much less research has been done with conflation between vector and image data. Traditional conflation requires at least two existing vector datasets. In many cases a second vector GIS dataset is not available and acquiring and collecting such vector GIS data is time-consuming and cost-prohibitive. Remote sensing provides an alternative solution. Large amounts of high-resolution satellite imagery and aerial photographs are available today, and many of these are free or have minimal cost to the public, such as USGS's DOQ (Digital Orthophoto Quadrangles) or USDA's NAIP imagery. Many agencies may have already acquired some high resolution images for other purposes. Making good use of widely available free imagery is cost effective for conflation. Our innovative snake based approach opens the door to that direction.

Using the TIGER road centerline as source and georectified satellite imagery or aerial photographs as destination, our approach can be extended from vector-vector conflation to vector-imagery conflation. Figure 2.6 demonstrates the vector-imagery conflation idea where 2.6(a) shows the original TIGER roads superimposed on the NAIP imagery and 2.6(b) indicates the results of conflation for the same section as in figure 2.4. The preliminary test is very encouraging. Here we manually identified and extracted the road intersections from the NAIP image and matched them to TIGER road intersections.



(a) (b)

Figure 2.6 Vector-Imagery conflation for the same section as figure 2.4 a) Matched TIGER road segments superimposed on NAIP imagery. b) Final results after snake-based conflation.

In this chapter we presented a hybrid conflation approach to improve the positional accuracy of TIGER road centerlines. This approach is based on the combination of the traditional conflation and the snake algorithm. The processes are similar to the traditional conflation approach except that the attribute transfer is replaced by snake alignments. After feature matching and rubber sheeting processes, the snake algorithm moves the TIGER road towards the high accuracy road images. This has advantages over traditional conflation approaches. No manual line segment splitting operation is needed and either vector road data or high-resolution satellite imagery/aerial photographs can be used as the destination dataset. Therefore, snake-based conflation is

not limited to vector-vector conflation and has much broader applications for integration of GIS, GPS and remote sensing data.

The test results show that this approach can dramatically improve the positional accuracy of TIGER road centerlines from originally over 100 m RMS error to a RMS error of only 3 m. Such significantly improved TIGER data with rich attributes could lead to many new applications for federal, state, and local governments and private industries.

Feature matching is critical to conflation. Here we applied a simple method based on distance and topology. Advanced matching approaches will be exploited in our future study. Also we will develop an algorithm to automatically extract the road intersections from imagery, thus making our approach fully automatic for vector-to-imagery conflation.

Chapter 3

Relaxation-Based Point Feature Matching for Vector Map Conflation

With the rapid advance of geospatial technologies, the availability of geospatial data from a wide variety of sources has increased dramatically. It is beneficial to integrate / conflate these multi-source geospatial datasets since the integration of multi-source geospatial data can provide insights and capabilities not possible with individual datasets. However, multi-source datasets over the same geographical area are often disparate. Accurately integrating geospatial data from different sources is a challenging task. Among the subtasks of integration/conflation, the most crucial one is feature matching, which identifies the features from different datasets as presentations of the same real-world geographic entity. In this chapter we present a new relaxation labeling-based point feature matching approach to match the road intersections from two GIS vector road datasets. The relaxation labeling algorithm utilizes iterated local context updates to achieve a globally consistent result. The contextual constraints (relative distances between points) are incorporated into the compatibility function that is employed in each iteration's updates. The point-to-point matching confidence matrix is initialized using the road connectivity information at each point. Both traditional proximity-based approach

and our relaxation labeling-based point matching approach are implemented and experiments are conducted over 18 test sites in rural and suburban areas of Columbia, MO. The test results show that our relaxation labeling approach has much better performance over the proximity matching approach in both simple and complex situations.

3.1 Introduction

Traditional geospatial mapmaking is a manual and expensive process that historically has been the responsibility of large government agencies such as U.S Geological Survey (USGS) or National Geospatial-Intelligence Agency (NGA). With the rapid advancement of Geographic Information System (GIS), Remote Sensing (RS) and Global Positioning System (GPS) technologies, it's much easier to produce, distribute and utilize digital geospatial information. It is now commonplace for government, academia, business and individuals to use geospatial information for many applications. Today, large amount geospatial information may be produced by and easily obtained from federal, state, local governments, commercial geospatial industries and even private citizens.

However, the exploitation of spatial map information from multiple sources often results in a disturbing reality: multi-source data over the same geographical area are disparate, both spatially and thematically. The disparity can be due to scale, resolution, time, compilation standards, source accuracy, registration, sensor characteristics, or errors. [52]. In addition, the spatial displacement is often non-systematic. Thus, a simple global registration or transformation of the data will not solve the problem. Figure 3.1 shows a

real-world example where U.S. Census Bureau's Topological Integrated Encoding and Referencing (TIGER) roads and Missouri Department of Transportation (MODOT)'s road centerlines are superimposed together on orthorectified high-resolution aerial photography. The problem of nonsystematic displacement is obvious. Accurately integrating geospatial data from different sources is a challenging task. Current integration strategies and methodologies have not kept pace with advances in data collection. It remains difficult to analyze even two spatial datasets acquired at different times, for different purposes, using different datums, positional accuracy (x, y, and z), classification schemes, and levels of *in situ* sampling or enumeration precision [23].

Map conflation is the process that combines two or more spatial representations of the same region to produce a superior dataset that is better than any of the original inputs in both spatial and attribute aspects. Through the conflation process, individual strengths of each source can be aggregated. For example**,** a dataset with excellent spatial accuracy but little attribute information can be integrated with one that has many attributes but of poor spatial accuracy to produce a new representation that has the best spatial and attributes from both sources.

The history of map conflation goes back to the early 1980's, where the first development and application of an automated conflation process occurred during a joint U. S. Geological Survey - U.S. Census Bureau project designed to integrate the agencies' respective digital map files of U.S. metropolitan areas [25]. The implementation of a computerized system for this task provided an essential foundation for much of the theory and many of the techniques used today. Since then much progress has been made in both research and application. However, the basic idea and procedures of conflation remain

the same. Generally, it has the following steps: feature matching, map alignment, and attributes transfer. In this chapter, we focus on feature matching which is the most important step for conflation.

Figure 3.1 TIGER roads (blue) and MODOT roads (red) are superimposed on an aerial photograph. The nonsystematic discrepancy problem between the two datasets is easy to see.

3.2 Previous Work on Point Feature Matching

Feature matching involves the identification of features from different geospatial datasets as being representations of the same geographic entity. This is the most critical and difficult step of conflation. Once the corresponding features from two spatial datasets are identified, the following map alignment and attributes transfer are easier tasks. Many *ad hoc* feature matching approaches in the field of geography have been developed for specific dataset pairs. One important approach is the plane-graph node-matching strategy used to conflate the Census TIGER files and USGS DLG files [25]. This iterative matching approach is based on nearness or proximity of features. The matched points serve as vertices for triangulations for rubber-sheeting to align the maps. Cobb et al., [28] applied a rule-based feature matching strategy to National Imagery and Mapping Agency (NIMA) Vector Product Format (VPF) products. Their feature matching is based on semantic similarities of attribute values and shape similarity of linear features. A statistical approach was developed to match road network data of the European standard Geographic Data File (GDF) and the German topographic cartographic spatial database (ATKIS) [29]. The matching problem is mapped onto a communication system, and measures derived from information theory are used to find an optimal solution. In [30], similarity measures were used for attribute matching and proximity graphs were used to model geographical context. A graph representing all possible matches is created and searched for groups that most likely represent the same real-world object. A semi-automated matching was proposed for network database integration [27]. The automated algorithm established robust correspondences for nodes, edges, and segments between two networks using a cluster-based matching mechanism. The interactive procedure

allowed a user to visually check and correct correspondences that were mismatched by the automated algorithm.

Feature matching has also been the topic of research in the fields of remote sensing, computer vision and pattern recognition. For example, feature matching is a critical step of feature-based image registration. Many feature matching methods have been presented such as methods using spatial relations, methods using invariant descriptors, graph-based methods, relaxation labeling, pyramids and wavelets [53].

While the features may be of different types such as point, line, or region etc., the point is the simplest form. It often serves as the basis upon which other more sophisticated representations can be built. It can be regarded as the most fundamental of all features. Feature-based methods for object recognition, motion analysis and image registration often rely on point pattern analysis to establish a correspondence within two related point-sets. In this chapter, we focus on the point feature matching methods.

Numerous point matching algorithms have been proposed in the fields of computer vision and pattern recognition. These algorithms include cluster method, graph method, spectral method, relaxation-based method etc. Cox & Jager compared several point pattern matching methods [54]. A review of selected methods is provided in the following.

Clustering method: The transformation parameters are calculated for all combinations of point pairs from both patterns. Correct matches tend to make a cluster. The parameter values correspond to the center of densest cluster. The strongest clusters in the parameter space then represent the most likely transformation parameters for the best match. Stockman [55] used a clustering approach to estimate the transformation

parameters and then determine the corresponding points between two point patterns. This technique was applied to aerial photographs in which automatically and manually detected features were used to register aerial photographs to maps for automatic map revision. Goshtasby & stockman [56] used the convex hull property to choose the subsets of points for matching in order to reduce computational time. A least-squares estimation approach to determining the optimum parameters that will match two point patterns was described by Umeyama [57]. Chang et. al. [58] developed a fast 2-D clustering algorithm which reduced the parameter space to two-dimensions (scaling, rotation) from four-dimensions (scaling, rotation, x translation, y translation).

Graph matching method is the process of finding a correspondence between the nodes and the edges of two graphs that satisfies some constraints ensuring that similar substructures in one graph are mapped to similar substructures in the other. Conte [59] presented a literature review on graph matching algorithm and the applications of graph-based techniques over the past 30 years. Caetano provided an approach which model the point matching task as a weighed graph matching problem, where weights correspond to Euclidean distances between nodes. He then formulated graph matching as a problem of finding a maximum probability configuration in a graphical model [60]. Ullman [61] proposed a minimal mapping approach, where the probabilistic cost function was based on the distance between the points in consecutive frames. In Kumar [62], point pattern matching problem was viewed in terms of weighted bipartite graph matching and a greedy algorithm was used to solve the correspondence.

One class of solutions to graph matching problems for point patterns is spectral methods which are based on the following observation: the eigenvalues and the

eigenvectors of the adjacency matrix of a graph are invariant with respect to node permutations. Hence, if two graphs are isomorphic, their adjacency matrices will have the same eigenvalues and eigenvectors. The basic idea common to most methods consists in performing spectral analysis in each of the adjacency matrices and comparing the eigenvalues /eigenvectors using some matching criteria [59]. There have been a number of attempts to use spectral methods for point-set matching. Scott and Longuet-Higgins [63] aligned point-sets by performing singular value decomposition on a point association weight matrix. This method was extended by Pilu [64] who included neighborhood intensity correlation information into the association weight calculation. Shapiro and Brady [65] reported a correspondence method which relies on measuring the similarity of the eigenvectors of a Gaussian point-proximity matrix. Carcassoni and Hancock [66] proposed a spectral method that is based on the use of spectral features to define clusters of nodes. This method uses hierarchical matching by first finding a correspondence between clusters and then between the nodes in the clusters.

The iterative closest point (ICP) algorithm [67] utilizes the nearest-neighbor relationship to assign a binary correspondence at each step. This estimate of the correspondence is then used to refine the transformation. It is a very simple and fast algorithm which is guaranteed to converge to a local minimum. Chui and Rangarajan [68] described a feature-matching method that is an extension of the iterative closest point (ICP) method, determining the transformation function and the feature correspondences at the same time while minimizing an energy function. Paragios et al. [69] achieved the same by first aligning the feature sets globally through chamfer matching and then estimating the local deformations by local searching.

Relaxation labeling techniques iteratively assign values to mutually constrained objects in such a way to ensure that the values remain consistent. A solution is found when these values converge. Ranade and Rosenfeld [70] proposed relaxation algorithm based on the relative distance information between points. Their method can handle translational differences. Wang *et al.* [71] extended Ranade and Rosenfeld's method to use feature information associated with each point. This algorithm allows translations and rotations of the point patterns, but its time complexity is O (n^4)**,** where *n* is the number of points. In [72], the megersort concept was used to speed up this point-matching algorithm so that its complexity is reduced from O (n^4) to O (n^3). Ogawa [73] described a fuzzy relaxation labeling technique for point patterns.

Other methods: Starink & Backer [74] presented a point matching method that iteratively changes an arbitrary initial mapping towards the minimal cost mapping based on a simulated annealing minimization scheme. Zhang [75] used partial Hausdorff distance and presented a genetic algorithm method to solve the incomplete unlabeled matching problem under general affine transformation. Yin [76] presented a new point pattern matching algorithm based on particle swarm optimization (PSO). The set of transformation parameters is encoded as a real-valued vector called a particle. A swarm of particles are initiated at random and fly through the transformation space influencing each other for targeting the optimal transformation.

In our conflation case, the spatial displacement between the two road datasets is often non-systematic. A simple global transformation will not solve the problem. Therefore, the clustering method is not suitable here. In addition, an exact one-to-one correspondence between the two datasets does not often exist. There may be extra or

missing points in one point pattern that have no corresponding points in the other set. Since spectral methods have limitations to size difference and structural corruption, spectral methods are not appropriate in our case. However, a relaxation labeling approach can tolerate local geometric distortion, and we propose it here. It was applied to our vector-to-imager conflation application (chapter 4) and is described in the following section.

3.3 Methodology

One of the most commonly used matching criteria is geometric. Geometric criteria include length, perimeter, area, shape, angle, distance etc. [33]. Distance (proximity or nearness) between features is the most straightforward way to match common objects. A feature in one map that is close to a given feature in the second dataset has a better chance of representing the same object. Euclidean distance is often used to calculate the proximity of point features. Therefore, we implemented the traditional proximity algorithm for comparison purpose.

In most conflation situations, many local distortions and deformations exist due to the non-systematic differences between the two modalities. Thus, this is a more complicated non-rigid point matching problem. Most approaches to non-rigid shape matching use an iterated estimation framework. In our case, the local relationship among neighboring points is stronger and more stable than the global one. In addition some outliers may exist. Therefore, we chose relaxation labeling because it can tolerate local geometric distortions [77]. The basic idea is to use iterated local context updates to achieve a globally consistent result. The contextual constraints are expressed in the form

of a compatibility function. It is a bottom-up search strategy that involves local rating of similarity which depends on the confidence of label assignments of their neighbors. These ratings are updated iteratively until the ratings converge or until a sufficiently high confidence assignment is found. In fact, Rosenfeld and Kak (1982) recommend only running a relation labeling for a few iterations since that is normally sufficient to "identify" real correspondences. The local similarity measures are used to assign heuristic, fuzzy, or probabilistic ratings for each location. These ratings are then iteratively strengthened or weakened in accordance with the ratings of the neighboring measures [77]. An iterative relaxation algorithm was developed for point matching based on the relative distance information between points [70]. Here we apply and extend the approach as found in [78].

Let $A = \{A_1, A_2, \ldots, A_n\}$ be a set of road intersections and terminations from one vector road dataset, and $B = \{B_1, B_2, \ldots, B_m\}$ be a corresponding set from another vector road dataset. Suppose A_i and B_j are postulated as corresponding points. For any other point pair (A_h, B_k), their joint compatibility C(i, j; h, k) is defined as a function of how much the actual position of A_h relative to A_i differs from the corresponding position of B_k relative to B_j [78]. Suppose B_j is shifted to A_i and becomes $B_{j'}$, and B_k is shifted the same amount to $B_{k'}$. Let $D_{hk'}$ be the distance between A_h and $B_{k'}$. In our application we divide $D_{hk'}$ by the distance from A_i to A_h (D_{ih}) to make it relative rather than absolute difference. Define δ and the compatibility between this pair of assignments as:

$$\delta = D_{hk'} / D_{ih} \tag{3.1}$$

$$C(i, j; h, k) = 1/(1+\delta^2) \tag{3.2}$$

Note that $C(i, j; h, k) = 1$ for $\delta = 0$ and goes to zero as δ becomes large.

This matrix of point pair compatibilities can be computed once per map pair and is used to modify confidence in point-pair assignment confidence between the two maps. Suppose P_{ij} represents the confidence of a match between road intersection or termination points A_i and B_j. Then, at a given iteration, the set of numbers $C(i, j; h, k) \cdot P_{hk}$ for all "neighbors" of A_i and B_j is used to obtain a new estimate of P_{ij}. The new estimate is commonly defined as an average of the previous estimate and the other points' assignment confidences. In defining the net contribution of A_h to P_{ij}, it is reasonable to use the max, rather than the average of terms $C(i, j; h, k) \cdot P_{hk}$, since if any one of these terms is large there is strong support for P_{ij} from A_h, even if all the other terms are small. Thus a plausible relaxation formula in this situation is:

$$P_{ij}^{(r+1)} = \frac{1}{n} \sum_{h=1}^{n} \max_{k=1}^{m} \{ C(i, j; h, k) P_{hk}^{(r)} \} \qquad (3.3)$$

where r = 0, 1, 2, … is the iteration number. We define C (i, j; h, k) = 1 if h = i; thus the self-support term h = i in the average is just P_{ij}.

The initial estimates of the P's can be made in various ways. $P_{ij}^{(0)}$ can be defined with some measure of similarity between A_i and B_j. If the patterns match exactly, the P's do not change under the iteration process, but otherwise they decrease after each iteration. However, if a good correspondence exists between some of the A's and some of the B's, those P_{ij}'s for which A_i corresponds to B_j decrease slowly, since they have substantial support, while the other P_{ij}'s decrease much more rapidly [78]. Hence, the goal of relaxation labeling is not to iterate until convergence, but instead is to drive the assignment confidence matrix toward an easily thresholded version. The resultant binary matrix then produces a one-to-one mapping between subsets of points from the two maps.

To successfully implement the relaxation labeling algorithm in our application, making use of relevant available information is necessary. Chen et al [35] exploited auxiliary information, such as the degree of an intersection (i.e., the number of intersected road segments), the density of these intersections, the orientation of roads to improve the brute-force point pattern matching algorithm. This efficiently pruned the search space. In our algorithm, we only used the road connectivity information (degree of intersection) since it is invariant to rotation, and many of the map pairs we process have sections that exhibit local rotation. The road intersections and road terminations (i.e., road dead ends along the map borders or cul-de-sacs) for both vector datasets have one common attribute that indicates the number of road arms emanated from that point. Road terminations, 3-way and 4-way road intersections have values of 1, 3 and 4 respectively. We use this attribute to initialize the $P_{ij}^{(0)}$ matrix: $P_{ij} = 1$ if and only if point A_i and B_j have the same number of emanated road arms, i.e. the same type of road intersection or road termination, otherwise $P_{ij} = 0$. We do not allow a match between different road intersection types. Hence, our original confidence matrix is a many-to-many assignment.

In our particular application, the road vector datasets are already georeferenced into the same coordinate system but have substantial local distortion. The points in a neighborhood of a given point have greater influence than points far away. The discrepancy within a neighborhood is relatively similar although the global discrepancy is non-systematic. Therefore, we further refine the initial confidence matrix: $P_{ij} = 0$ if the distance between A_i and B_j is larger than a given threshold. A point can only be matched to another point within a certain distance. This threshold is determined by checking the displacement between the two datasets. The main purpose of setting such a threshold is to

eliminate useless computation since we only need to update a point pair assignment confidence with assignments and compatibilities of pairs that are relatively close to the given pair in the two maps. The actual value of this threshold is not critical as long as it's larger than the displacement between the two datasets. In our experiment, we set it to 300 meters.

Similarly, to calculate the new P_{ij} for point A_i during each iteration, we only consider the points A_h within a given distance to A_i. If the distance between A_h and A_i is larger than a threshold, it has no influence on P_{ij}.

This initialization and relaxation process works well. Our experiments showed that good matches are found after a few iterations, so we only run two iterations. From the final P matrix, we find the maximum value for each row i and set the rest to zero. If that value is also the maximum in its column j, then we say point pair (A_i, B_j) is a match. The pseudo code of relaxation labeling algorithm is listed below.

```
Function Relaxation (A, B)
    Initialize P

    For iteration 1 to n
        For each pair Ai, Bj
            For each pair Ah, Bk satisfying the "closeness" constraints
                Compute C(i,j;h,k) based on formula (3.2)
            End for
        Update P[i,j] using formula (3.3)
        End for
```

End iteration

Find the maximum in each row of P matrix

Create the Match matrix

Return

In the following we use a simple example (figure 3.2) to illustrate the entire processes.

Suppose we have TIGER point set A={A_1,A_2,A_3,A_4,A_5,A_6,A_7,A_8} and MODOT point set B = {B_1,B_2,B_3,B_4,B_5,B_6,B_7,B_8}. Let's use point A_6 to illustrate the process. Since A_6 is a 3-way intersection, it can only to match to one of the 3-way intersections in set B (B_3, B_6, B_7). Therefore, only P_{63}, P_{66}, P_{67} = 1 in row 6 of the initial P matrix.

After two iterations, we find the maximum value in each row, and set the other values to zero. The value P_{63} = 0.202, is the maximum value in 6^{th} row of the final P matrix. Since it is also the maximum in 6th column, we set M_{66} = 1 in the match matrix. We know A_i matches to B_j if A_{ij} = 1 in the match matrix. In this example, A_1->B_1, A_2->B_4, A_3->B_3, A_4->B_2, A_5->B_5, A_6-> B_6, A_7->B_7, A_8->B_8. The black arrows in figure 3.2 indicate the matches.

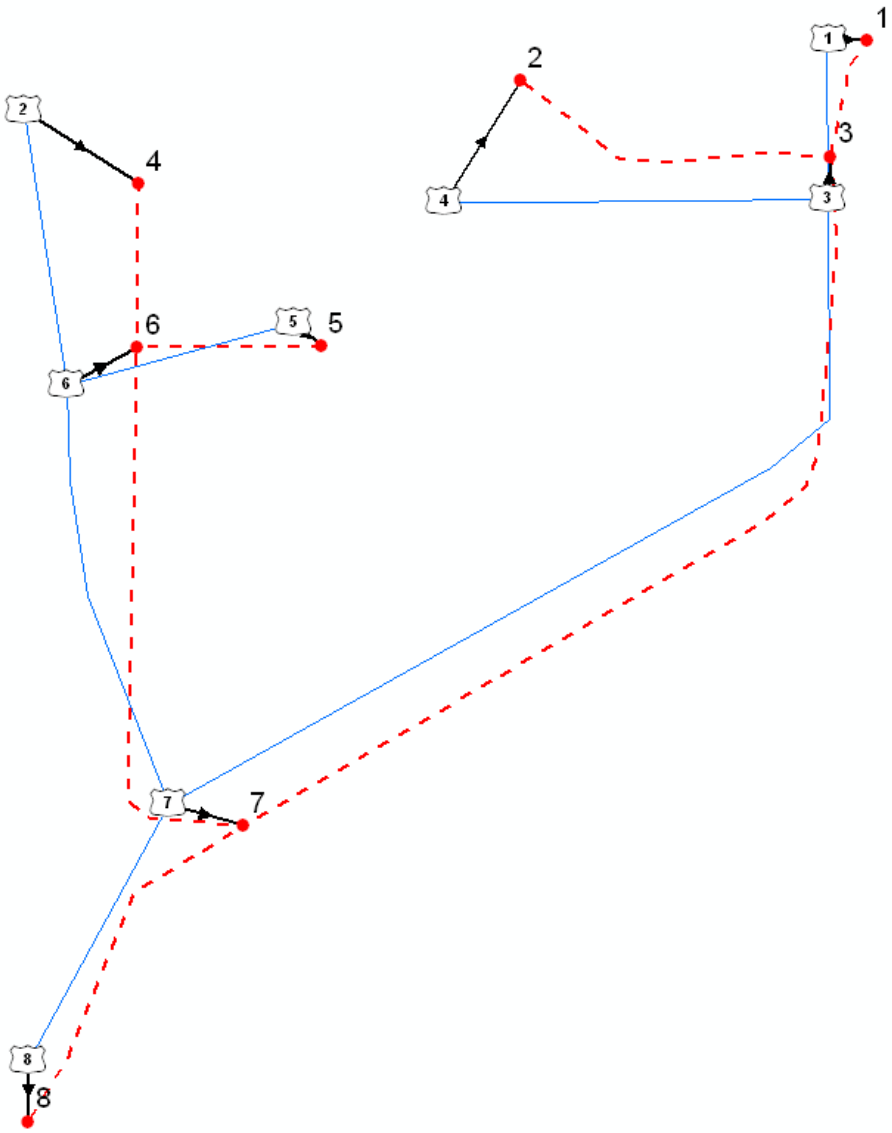


Figure 3.2 The example is used to illustrate the relaxation process. Point set A (white polygon shaped symbol) are road intersections and terminations from TIGER road (solid blue), and point set B (red dot) are from MODOT road (dashed red). Black lines with arrow represent the matched point pairs. The point # of set A is shown inside the polygon and the point # of set B is labeled on its upper-right side.

Initial P matrix

1, 1, 0, 1, 1, 0, 0, 1,

1, 1, 0, 1, 1, 0, 0, 1,

0, 0, 1, 0, 0, 1, 1, 0,

1, 1, 0, 1, 1, 0, 0, 1,

1, 1, 0, 1, 1, 0, 0, 1,

0, 0, 1, 0, 0, 1, 1, 0,

0, 0, 1, 0, 0, 1, 1, 0,

1, 1, 0, 1, 1, 0, 0, 1,

P matrix after first iteration

0.351, 0.272, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000,

0.000, 0.000, 0.000, 0.448, 0.300, 0.000, 0.000, 0.000,

0.000, 0.000, 0.352, 0.000, 0.000, 0.000, 0.000, 0.000,

0.438, 0.656, 0.000, 0.517, 0.562, 0.000, 0.000, 0.000,

0.000, 0.308, 0.000, 0.310, 0.431, 0.000, 0.000, 0.000,

0.000, 0.000, 0.000, 0.000, 0.000, 0.587, 0.000, 0.000,

0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.363, 0.000,

0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.241,

P matrix after second iteration

0.084, 0.054, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000,

0.000, 0.000, 0.000, **0.136**, 0.066, 0.000, 0.000, 0.000,

0.000, 0.000, **0.085**, 0.000, 0.000, 0.000, 0.000, 0.000,

0.112, **0.234**, 0.000, 0.141, 0.176, 0.000, 0.000, 0.000,

0.000, 0.070, 0.000, 0.069, **0.127**, 0.000, 0.000, 0.000,

0.000, 0.000, 0.000, 0.000, 0.000, **0.202**, 0.000, 0.000,

0.000, 0.000, 0.000, 0.000, 0.000, 0.000, **0.082**, 0.000,

0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, **0.040**,

Find the maximum value in each row, and set the rest to zero

0.084, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000,

0.000, 0.000, 0.000, 0.136, 0.000, 0.000, 0.000, 0.000,

0.000, 0.000, 0.085, 0.000, 0.000, 0.000, 0.000, 0.000,

0.000, 0.234, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000,

0.000, 0.000, 0.000, 0.000, 0.127, 0.000, 0.000, 0.000,

0.000, 0.000, 0.000, 0.000, 0.000, **0.202**, 0.000, 0.000,

0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.082, 0.000,

0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.040,

The final match matrix

1, 0, 0, 0, 0, 0, 0, 0,

0, 0, 0, 1, 0, 0, 0, 0,

0, 0, 1**,** 0, 0, 0, 0, 0,

0, 1, 0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 1, 0, 0, 0,

0, 0, 0, 0, 0, **1**, 0, 0,

0, 0, 0, 0, 0, 0, 1, 0,

0, 0, 0, 0, 0, 0, 0, 1,

The relaxation labeling algorithm was coded in C# and implemented as plug-in tool within ArcMAP 9.3, ESRI's Desktop ArcGIS product.

3.4 Experiments

To evaluate our relaxation based feature matching algorithm, we obtained both U.S Census Bureau's TIGER road centerline data and Missouri Department of Transportation (MODOT)'s road data from Missouri Spatial Data Information Service. The two vector datasets are already georeferenced in the same coordinate system and can be superimposed together. The TIGER road data has poor positional accuracy with varied error distribution across the area whereas the MODOT road map has good positional accuracy. The feature matching algorithms were run on 18 small test areas. Each area has the same size of 2 km by 2 km as seen for example in figures 3.3 and 3.4. Those 18 areas represent typical rural and suburban scenes in Columbia, MO. From figure 3.3 and figure 3.4, the non-systematic discrepancy between the two datasets is easy to see.

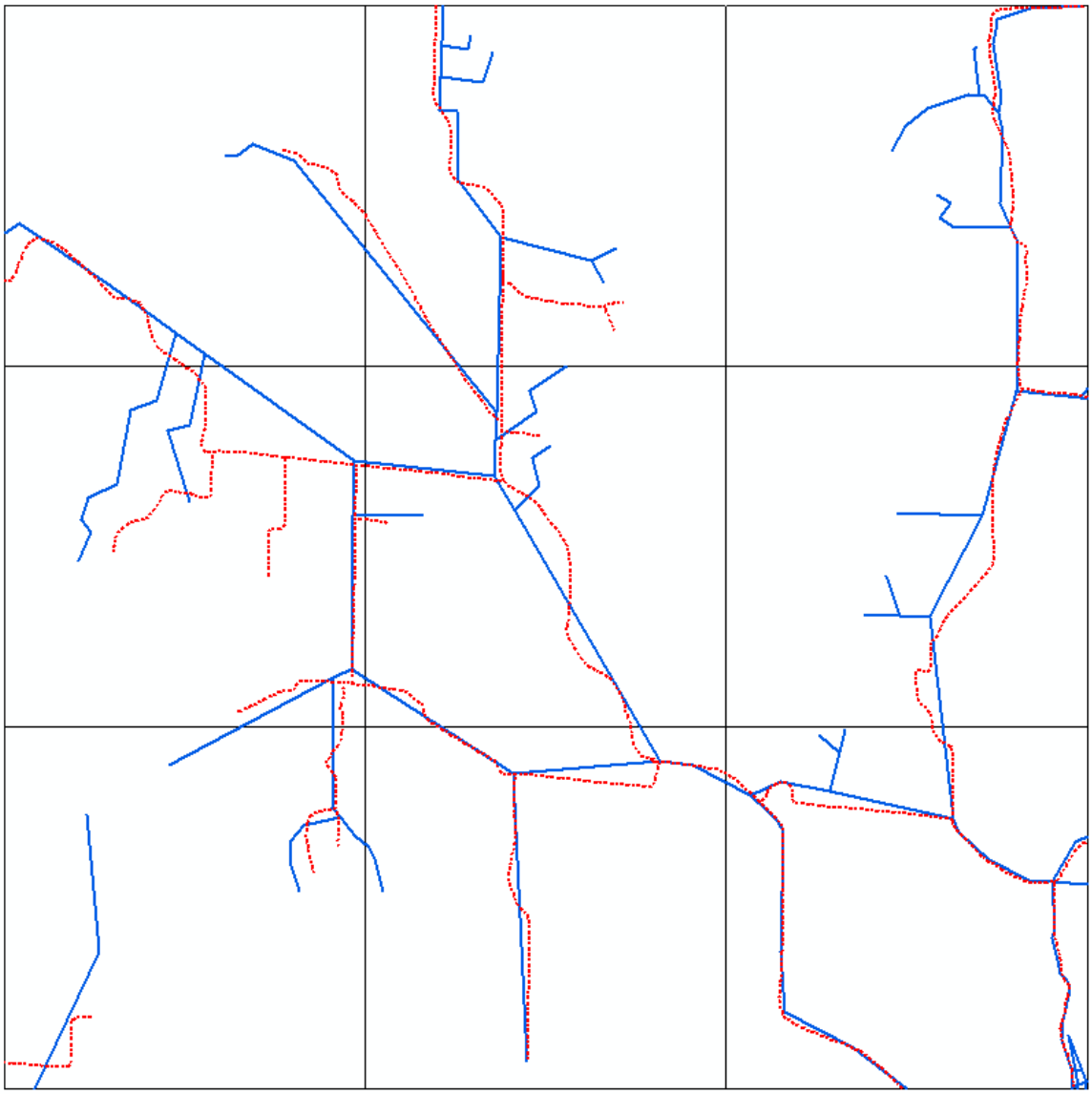


Figure 3.3 TIGER roads (solid blue) and MODOT roads (dashed red) of 9 rural test areas in city of Columbia, MO

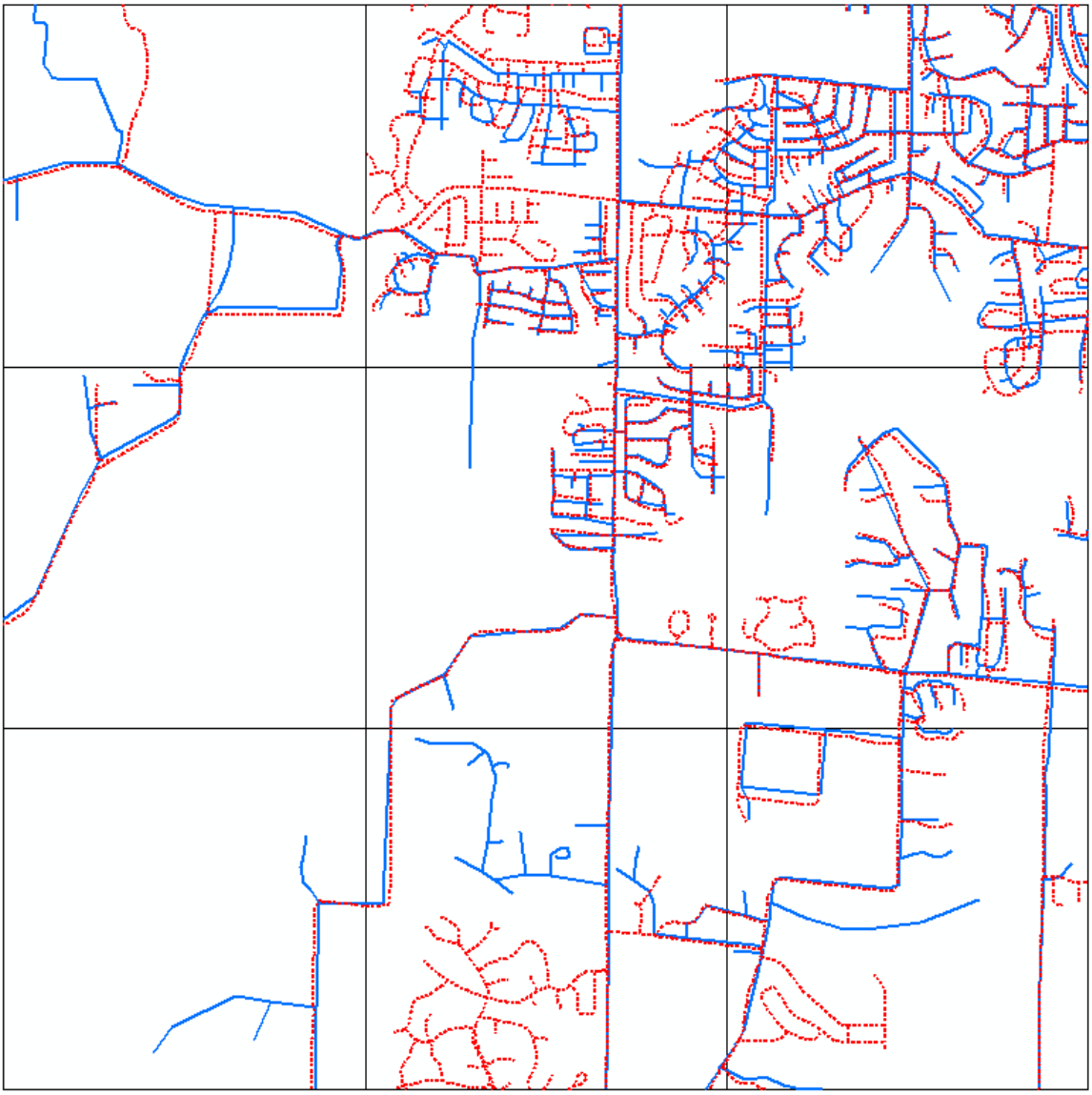


Figure 3.4 TIGER roads (solid blue) and MODOT roads (dashed red) of another 9 test areas in City of Columbia, MO

We first extract the road intersections and road terminations from the road centerline files. These point sets are used by point matching algorithms. For comparison, two point matching methods are implemented. One approach uses the traditional proximity matching, where, for each point in set A, the algorithm finds its nearest point in

Set B that has the same type (i.e., road termination, 3-way or 4-way intersection). The second approach is our relaxation labeling approach described in section 3. The two point matching approaches are performed on each of the 18 test areas separately.

To evaluate the performance of matching, two criteria correctness and completeness are used, defined as follows. Correct match (CM) is the number of correctly detected matches. Incorrect match (IM) is the number of incorrectly detected matches. Undetected match (UM) is the number of actual matches which are not detected by the algorithm. Correctness and completeness are then defined as following:

$$Correctness = \frac{CM}{CM+IM} \tag{3.4}$$

$$Completeness = \frac{CM}{CM+UM} \tag{3.5}$$

Hence, correctness is the number of correct matches (CM) divided by the total number of matches found by algorithm (CM + IM). Completeness is the number of correct matches divided by the total number of actual true matches (CM + UM). The correctness and completeness provide two complementary measures for evaluating the point matching.

In figure 3.4, we notice the roads in the left side of the test areas are quite sparse, so we group these left three test areas into rural. Therefore, there are 12 rural test areas and 6 suburban teat areas.

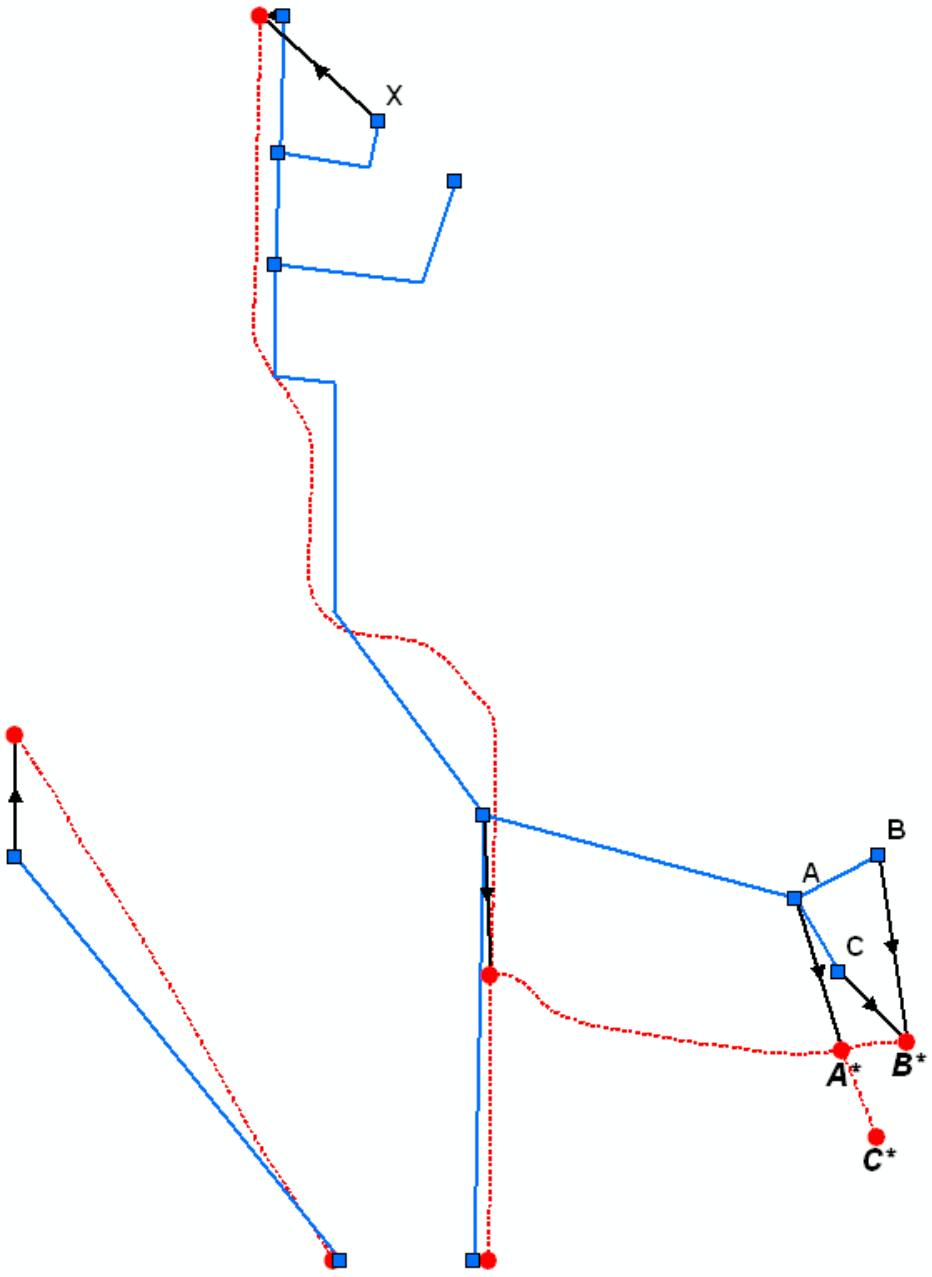
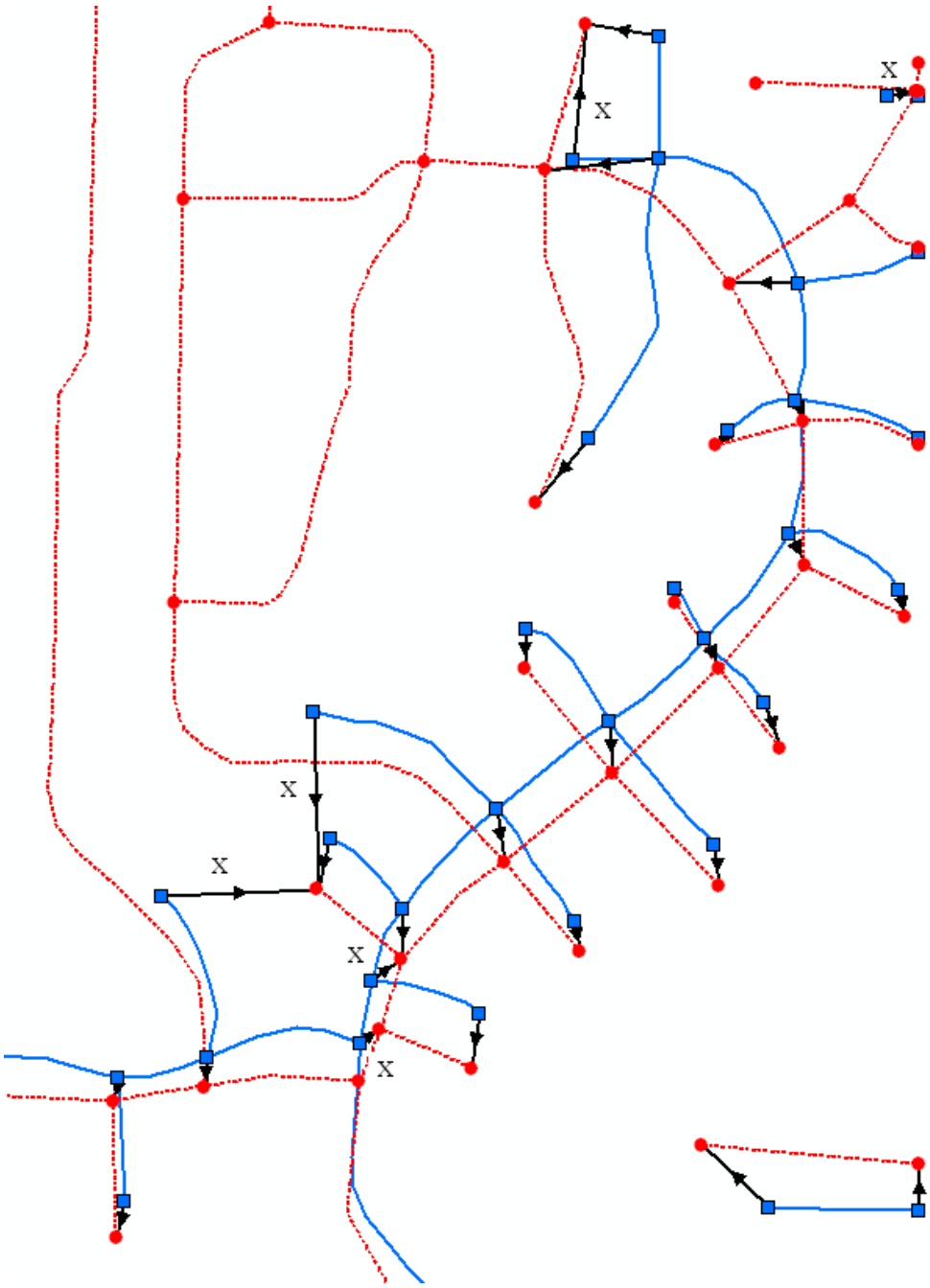
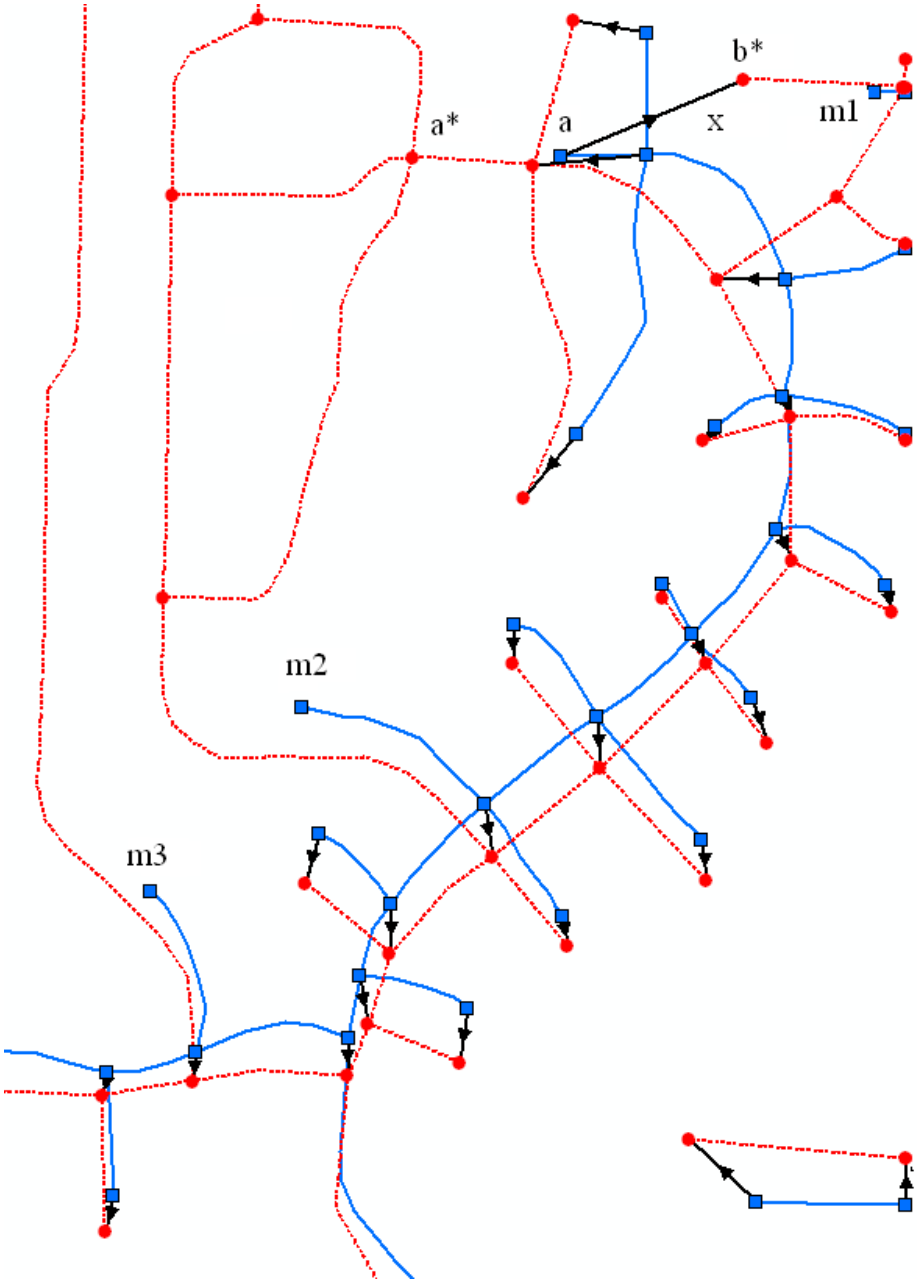


Figure 3.5 The matching result produced by proximity algorithm for one rural test area. The black lines with arrow are links from TIGER (solid blue) to MODOT (dashed red) road that represent the matched point pairs.



3.6a)



3.6b)

Figure 3.6 Comparison of the matching results for one portion of suburban test area. The black lines with arrow are links from TIGER (solid blue) to MODOT (dashed red) road that represent the matched point pairs. Figure 3.6a shows the result produced by proximity algorithm where incorrect matches are marked with symbol ‘X’. Figure 3.6b shows the results of relaxation labeling algorithm.

In rural areas, the matching situation is quite simple. There are at most 1 or 2 incorrect or missed matches in each rural test area. However, the correctness and completeness seem low for some areas due to the very small total number of matches. For example, in figure 3.5, only 2 out of the 9 matches by the proximity algorithm are incorrect, so the correctness is 7/9=78%. In figure 3.5, for point C in the TIGER road, the proximity algorithm found the nearest matching point B^* in the MODOT road, which is wrong. Our relaxation labeling approach considers the relationship among a set of nearby points. Therefore, such errors are avoided.

Table 3.1 shows the performance of two algorithms over the 18 test areas. For the 12 rural test areas, the average correctness and completeness are 90% and 92% for proximity algorithm, and the average correctness and completeness are 94% and 96% for our relaxation labeling algorithm. Both the measures for relaxation approach are 4% higher than proximity approach. Our relaxation labeling algorithm has better performance than the traditional proximity approach even for simple road situations in rural areas.

In suburban areas, the road network is more dense and complex. It is quite common that the nearest point is not the correct match. Figure 3.6 shows one portion of a suburban test area. It can be seen that several matches detected by proximity algorithm are incorrect (marked by ‘x’ in figure 3.6a). However, the relaxation labeling algorithm still did good job to find the right matches (figure 3.6b). Since TIGER roads are older than MODOT roads, there are some updates in MODOT roads. Point a is a road termination in TIGER, it could not match to the real counterpart (point a*) in MODOT road, which is a 4-way intersection. It matched to the wrong point b* and that in turn

caused point m1 to miss the corresponding true match. In some cases, missed matches such as m2 and m3 are proper because they don't have counterparts in MODOT roads.

Table 3.1 Performance evaluation of two point matching approaches

	Proximity Approach		Relaxation Approach	
	Correctness	**Completeness**	**Correctness**	**Completeness**
Rural01	100%	83%	100%	83%
Rural02	100%	100%	100%	100%
Rural03	88%	100%	100%	100%
Rural04	85%	92%	91%	83%
Rural05	85%	92%	92%	92%
Rural06	100%	100%	100%	100%
Rural07	60%	75%	67%	100%
Rural08	78%	75%	89%	100%
Rural09	100%	100%	100%	100%
Suburban01	100%	100%	100%	100%
Suburban04	100%	100%	100%	100%
Suburban07	88%	88%	88%	88%
Rural Average	**90%**	**92%**	**94%**	**96%**
Suburban02	78%	100%	82%	100%
Suburban03	76%	85%	96%	96%
Suburban05	78%	85%	93%	93%
Suburban06	84%	91%	96%	100%
Suburban08	72%	81%	93%	92%
Suburban09	77%	82%	95%	91%
Suburban Average	**78%**	**87%**	**93%**	**95%**

The accuracy measures in table 3.1 confirm this observation. For the 6 suburban test areas, the average correctness and completeness are 93% and 95% for relaxation labeling algorithm. Its performance in suburban areas only slightly decreased (1%) over simple rural areas. The performance of the proximity algorithm in suburban areas is much worse. The average correctness and completeness dropped to78% and 87% respectively. They are 15% and 8% lower than the relaxation labeling algorithm on the same suburban areas.

3.5 Summary of Point Feature Matching

Feature matching is the most crucial element of conflation. The quality of feature matching determines the success of conflation. In this chapter we present a relaxation labeling point feature matching approach to match road intersections from two vector road datasets. The relaxation labeling algorithm considers local context and calculates the compatibility of nearby points. It is more robust and consistent both in simple rural areas and complex suburban areas. The test results clearly showed that our relaxation labeling approach has better performance over the proximity matching method commonly used in traditional conflation.

Once the features are correctly matched, we can either apply the traditional conflation approach to transfer the attributes from one dataset to another, or use our snake-based conflation approach (Song et al. 2006) to move the road networks from one dataset to correct locations in another vector dataset or imagery. Therefore, the final result has good spatial accuracy and rich attributes producing a product better than the original datasets.

Chapter 4

Automated Geospatial Conflation of Vector Road Maps to High Resolution Imagery

As the availability of various geospatial data increases, there is an urgent need to integrate multiple datasets to improve spatial analysis. However, since these datasets often originate from different sources and vary in spatial accuracy, they often do not match well to each other. In addition, the spatial discrepancy is often non-systematic such that a simple global transformation will not solve the problem. Manual correction is labor-intensive and time-consuming and often not practical. In this chapter, we present an innovative solution for a vector-to-imagery conflation problem by integrating several vector-based and image-based algorithms. We only extract the different types of road intersections and terminations from imagery based on spatial contextual measures. We eliminate the process of line segment detection which is often troublesome. The vector road intersections are matched to these detected points by a relaxation labeling algorithm. The matched point pairs are then used as control points to perform a piecewise rubber-sheeting transformation. With the end points of each road segment in correct positions, a modified snake algorithm maneuvers intermediate vector road vertices toward a candidate road image. Finally a refinement algorithm moves the points to center each

road and obtain better cartographic quality. To test the efficacy of the automated conflation algorithm, we used U.S. Census Bureau's TIGER vector road data and U.S. Department of Agriculture's 1-meter multispectral near infrared aerial photography in our study. Experiments were conducted over a variety of rural, suburban, and urban environments. The results demonstrated excellent performance. The average correctness measure increased from 20.6% to 95.5% and the average root-mean-square error decreased from 51.2 meters to 3.4 meters.

4.1 Introduction

With the rapid advancement of Geographic Information Systems (GIS), Remote Sensing (RS), and Global Positioning System (GPS) technologies, the application of geospatial information for a wide variety of problems has grown dramatically. A wealth of geospatial data can be easily accessed and obtained through internet mapping services or geospatial data clearinghouses provided by government agencies (U. S. Geological Survey, U.S. Census Bureau, GIS departments of local governments, etc.), public organizations or the private mapping industry.

As the availability of various geospatial datasets increases, there is an urgent need to integrate multiple datasets together. By fusing spatial datasets from different sources, one can support more comprehensive geospatial analysis that could not have been achieved by the use of any single dataset in isolation. Also this integration would result in cost savings for many applications.

Figure 4.1 The TIGER vector roads (red) are superimposed on an aerial photograph. The non-systematic discrepancy ranges from a few meters to over 100 meters, and has diverse directions.

However, accurately integrating geospatial data from different sources is a challenging task. This is because such geospatial data may have different levels of accuracy and precision in their attributes as well as their spatial and temporal dimensions [79]. Furthermore, they may have been created for different purposes. All of this leads to discrepancies in the representation of the same features from different sources. For example, one widespread problem occurs when vectors representing road segments do not line up with roads in background imagery. In addition, the spatial displacement is often non-systematic. Figure 4.1 shows a real-world example where vector road centerlines extracted from U.S. Census Bureau Topological Integrated Encoding and

Referencing (TIGER) database are superimposed on orthorectified high-resolution (1 meter) aerial photography. Here and in following figures we display the near infrared imagery in grayscale for a better view of the road vectors. The non-systematic displacement is obvious where the spatial displacements range from a few meters to over 100 meters, and the displacement is in different directions across the image. Thus, a simple overlay or global transformation of the data will not solve the problem. Manual correction is labor-intensive and time-consuming. Automated or semi-automated algorithms are needed to attack this vector-to-imagery integration problem.

Current data integration strategies and methodologies have not kept pace with advances in data collection. It remains difficult to analyze even two spatial data sets acquired at different times, for different purposes, using different datums, positional accuracy (x,y,z), classification schemes, and levels of *in situ* sampling or enumeration precision [23]. The University Consortium for Geographic Information Science (UCGIS) has identified data integration as a long term (5-15 year) research challenge entitled Spatial Data Acquisition and Integration and particular aspects of the problem, Geospatial Data Fusion, as a short-term (3-5 years) research priority [23].

The word conflation is often used as a synonym for integration of multiple sets of spatial data from different sources. Conflation is the process that combines two or more spatial representations of the same region to produce a superior dataset that is better than any of the original inputs in both spatial and attribute aspects. Through the conflation process, individual strengths of each source can be aggregated. For example, a dataset with excellent spatial accuracy but little attribute information can be integrated with one

rich in attribute information but of poor spatial accuracy to produce a new representation that is both spatially accurate and attribute rich.

The history of map conflation goes back to the early 1980's, where the first development and application of an automated conflation process occurred during a joint U. S. Geological Survey - U.S. Census Bureau project designed to integrate the agencies' respective digital map files of U.S. metropolitan areas [25]. The implementation of a computerized system for this task provided an essential foundation for much of the theory and many of the techniques used today. Since then much progress has been made in both research and application. Existing local GIS data were utilized to update and enhance the U.S. Census Bureau's TIGER database [26, 27, 80]. A rule-based conflation approach was applied to the National Imagery and Mapping Agency's (NIMA) Vector Product Format (VPF) datasets [28]. A statistical approach was developed to match road network data of the European standard Geographic Data File (GDF) and the German topographic cartographic spatial database (ATKIS) [29]. In [30], similarity measures were used for attribute matching and proximity graphs were used to model geographical context. This approach was utilized to conflate features manually extracted from topographic maps and tourist maps. Reference [31] presented a system design for a multi-agent-based infrastructure to perform distributed conflation. A semi-automated matching was proposed for network database integration [27]. The automated algorithm established robust correspondences for nodes, edges, and segments between two networks using a cluster-based matching mechanism. The interactive procedure allowed a user to visually check and correct correspondences that were mismatched by the automated algorithm. A local transformation algorithm based on counterpart linear features instead of point

features was developed in [32, 81]. Kovalerchuk [82] developed an approach to conflation/registration of images that utilized invariant algebraic properties of linear features identified in each image. In [83], a distance measure between two polylines and an algorithm for conflating or optimally matching two polylines, i.e., determining the rotation and translation which minimizes the distance between two polylines was proposed.

Matching and integrating vector datasets together has been a subject of research for over two decades and strides are being made. Several specialized commercial conflation software packages are available for vector-to-vector conflation. However, much less research has been done with matching and fusing vector and image data.

On the other hand, automated feature extraction (road, building etc.) from remotely sensed imagery has been an active research topic for many years. Research on automated road extraction from imagery can date back to late 1970's and early 1980's. A procedure for tracking road segments in aerial imagery was described in [84]. Local road evidence obtained from multiple line detectors was combined and the road was tracked by finding a best path using the F* algorithm [85]. Reference [86] described a road tracking system that uses multiple cooperative methods. An intermediate process was developed to connect road-like fragments [87]. Antiparallel edge pairs were utilized to construct road hypotheses followed by linking [88]. A road tracing algorithm based on least squares profile matching and the Kalman filter was presented in [89]. An active testing model for tracking roads from satellite images was proposed in [90]. In [91], geometric-probabilistic models and MAP (maximum a posteriori probability) estimation were used to find main roads in aerial images. Steger [92] developed a line detector based

on differential geometry. A multi-resolution approach was presented in [45]. Lines are extracted using the Steger's algorithm [92] in a reduced-resolution image and edges are extracted from the original high-resolution aerial photograph. Then the results were fused and grouped. An approach based on road network topology was proposed in [93]. Reference [94] used a feature-based hypothesis and verify paradigm to extract a street grid in an urban environment. A model-based method for linear feature extraction from aerial images was described [95]. A self-organizing road map algorithm for road extraction in classified high-resolution image was proposed in [96]. In [97], an approach of urban road network extraction by integrating knowledge about roads and their context using scale-dependent models was presented. Here we only mentioned a few citations. Several collections of road extraction research can be found in [98-101], and an extensive bibliography is provided in [102].

However, most of these approaches still have difficulty in practice due to the complexity of earth imagery. Normally, automatically extracted cartographic features need considerable post editing. In addition, the extracted features are of very limited value without attributes such as street names, address ranges, speed limits, road conditions, etc. Those attributes cannot be obtained from imagery. Therefore, the extracted features need to be integrated with other existing GIS data sets containing such attributes.

Existing vector datasets can also provide knowledge to assist feature extraction from imagery. Some research has explored the integration of vector geospatial data and imagery for update and refinement of spatial databases. Maps, GIS data or other ancillary data can be used to facilitate road extraction. An object-oriented expert system was used

to extract roads from SPOT imagery [103]. Reference [104] constructed knowledge-based semantic models for interpretation of road networks in aerial images. In [105], a map-guided procedure was combined with a back-propagation neural network. A map-based road detection algorithm from space borne SAR images was presented in [106]. Markov random fields were used to locally register cartographic road networks on SPOT satellite images [107]. A graph-based approach for road verification from aerial imagery was presented in [108]. Reference [109] presented a system for 3-D road network reconstruction from aerial images using knowledge-based image analysis, while reference [110] surveyed the state of the art in automatic road extraction for GIS update from aerial and satellite imagery. One such approach is the active contour model (snake) which has been used for semi-automatic road extraction or road database update from imagery. LSB-Snakes (Least Squares B-Spline Snakes) were utilized to extract road features from digital images in [44]. A human operator identifies roads from an image and provides a few coarsely distributed seed points along the roads. With these seed points as an approximation of the roads' position and shape, the roads are identified automatically by the LSB-Snake. Another approach proposed for automatic road extraction in aerial imagery exploits the scale-space behavior of roads in combination with geometrically constrained snake-based edge extraction. Here the snake approach is used to bridge shadows and partially occluded areas between two road segments [46].

A method was developed for detecting changes between an existing road database and a co-registered satellite image in [47]. First the road database was used to provide an approximation for subsequent road detection using a snake approach to correct road location. Then a statistical line-following process was run to detect a new road starting

from the existing network. Reference [48] extended the above system by using road intersections. Road intersections improve matching between the road database and the lines on the image, with hypotheses for new road segments generated from these line junctions. In [49], the model of snakes was expanded to function in a differential mode by introducing an additional energy term that describes the discrepancy between the current snake solution and the pre-existing road shape information. This method combines object extraction and change detection to update and improve pre-existing GIS information for that object. An algorithm to match synthetic aperture radar (SAR) images and road databases was presented in [50]. Vectors provided by the database give an initial estimation of the road location that is refined through snakes. The ziplock snake model was developed in [51]. The user-supplied end points and the automatically computed edge gradients in their vicinities served as anchors. The image term is then turned on progressively from the snake's extremities toward its middle section. As a result, the snake is progressively clamped onto an image contour.

However, a well known problem of active contour models is that they require very good initialization and are very sensitive to noise in imagery. In most of the snake-based applications, it is assumed that the initial position of the snake is relatively close to the desired solution. When the snake's initial position is far away from the desired result or the image contains considerable noise, the snake often gets stuck in an undesirable local minimum.

In our application, the initial positions of vector roads are often far away from their counterpart in the imagery (figure 4.1). The results are undesirable if we apply a traditional snake algorithm directly without any preprocessing as will be seen in section

II-E. To overcome this problem, here we develop an innovative solution to this vector-to-imagery conflation problem by integrating and coordinating several vector-based and image-based algorithms. The heart of our approach is to automatically detect tie points between the road segment vectors and the image so that an active contour model can successfully conflate the two sources. The principal features we exploit are roadway intersections and terminal ends. Note that we do not perform road line extraction from imagery. Our work takes attribute-rich vector road maps with large non-systematic spatial discrepancies and improves the spatial accuracy by conflating it to image data using evidence of roads derived from the imagery. Details of these processes will be described in the next section, followed by experiments and accuracy assessment. Some discussions and conclusions are given in the final section.

With this approach we are addressing the errors of misalignment, attribution differences, or simply elements that have been generalized in their representation. These types of error are common place and exist in geospatial data as they are moved between scales of representation as well as through time. For example, a parcel map may have been created over 50 years ago and maintained on mylar separates until recently. These files are then digitized or scanned to create a digital version of these original files. This process typically faithfully reproduces any and all error inherent in these source files. When combined with the higher accuracy imagery files now prevalent in the industry these map files do not line up and show highly variable patterns of error across the mapped area. The ability to use higher accuracy imagery to help guide the conflation process allows local governments or other organizations to leverage their data investment in these systems as they increase positional accuracy. This function is of great benefit,

both fiscally and politically, to the jurisdiction. Fundamentally, the attributes on the poorer quality line work can be migrated to a new positional accuracy (that of the imagery) while maintaining all the attributes and relational linkages associated with those attributes.

4.2 Methodology

Conflation requires that associations can be determined to move the geometry and/or attributes of one set of feature representations to another more spatially accurate representation. The determination of these associations is based on the ability to extract and recognize tie-points between files. When matching vector-to-vector, the nodes, valences, lengths, and pattern of these relationships are well described within their inherent topologies. However, within an image, these point- and line-based relationships must be built from extracted information created through a series of processes outlined in figure 4.2. The major steps include spatial contextual information extraction, automated road intersection and road termination extraction from imagery, a novel point pattern matching approach by relaxation labeling, piecewise local affine transformation, position correction by snakes, and position refinement. As with all automated systems, parameter choices must be made or learned. We set those parameters (mostly thresholds) based on some preliminary experiments and then held them fixed for the results shown later. We underlined our choices of the values needed for the various steps. While we do not show a sensitivity analysis, we have found subjectively that results are not overly dependent on exact choices of the parameters.

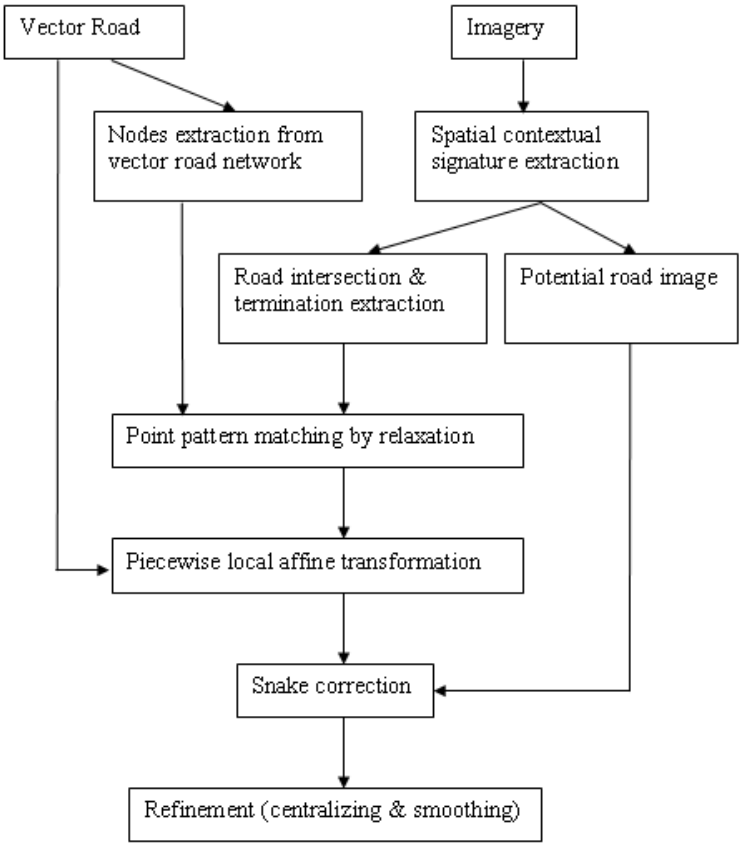


Figure 4.2 The work flow of vector-to-imagery conflation process

4.2.1 Spatial Contextual Information Extraction

Generally roads are long narrow linear features and buildings are generally rectangular shapes. In [111] a spatial length-width contextual measure was developed to increase discrimination between the road and building classes for urban land cover classification. The context of each pixel was examined by measuring the spatial dimensions of groups of spectrally similar connected pixels. Here we applied and extended this approach for road intersection detection and candidate road image creation. We emphasize that we are not performing road extraction from an image as described above. Our approach only needs a rough “optimistic” binary mask of the image that contains most of the road areas.

Vegetation and road surfaces have significant difference in spectral reflectance characteristics. Green vegetation generally reflects 40%-50% of the incident near-infrared energy, with the chlorophyll in the plants absorbing 80%-90% of the incident energy in the visible part of the spectrum [112]. Road surfaces generally have higher reflectance in the visible region and lower reflectance in near-infrared than vegetation. The Normalized Difference Vegetation Index (NDVI) is commonly used to measure vegetation amount. As the NDVI value grows, so does the amount of photosynthesizing vegetation present. NDVI is calculated as:

$$NDVI = (NIR - Red)/(NIR + Red) \tag{4.1}$$

Where *NIR* is the reflectance value of the near-infrared band and *Red* is the reflectance value of the red band.

Therefore, we first mask out vegetation by applying a threshold (0.2 in our experiments) to NDVI to reduce the computational load, leaving roads and other non-vegetated regions for further processing. For a grey scale image, a brightness threshold separates the image into two parts (bright and dark). Then the spatial contextual extraction algorithm runs only on the bright or dark part depending on whether roads are brighter or darker than the background.

A median filter is applied to the image to remove noise because of its inherent properties of reducing tonal variations while retaining edges. It was found that if the data were median filtered before the length-width algorithm was applied, then the length-width measurements were more accurate representations of the scene content [111].

Next the spatial contextual measure extraction algorithm is run on the processed image. For each pixel in the image, we search its surrounding pixels along each of the

360 directions (0-359 degrees) radiating from the central pixel. If the spectral similarity between the new pixel and the central pixel is less than a given threshold, the search continues outward. The Euclidian spectral distance is used as the similarity measure. For mulitspectral image, the dynamic ranges between different channels may have significant difference. Each channel is normalized or stretched to the same range (0-255) based on its statistical properties. Let two pixels *i* and *j* be characterized by their B-dimensional normalized feature vectors $V_i = (v_{i1}, v_{i2},\ldots, v_{iB})$ and $V_j = (v_{j1}, v_{j2}, \ldots, v_{jB})$, the Euclidian spectral distance is defined as:

$$D_{ij} = \sqrt{\sum_{k=1}^{B} (v_{ik} - v_{jk})^2} \tag{4.2}$$

A similarity threshold is experimentally set based on simple examination of the image characteristics. A value of 50 works well in our application. Generally the roads consist of groups of spectrally similar connected pixels and have a significant spectral difference with their background. Therefore, the algorithm is not too sensitive to the similarity threshold.

The maximum of the sum of traced lengths along two opposing directions is used as the length measure and the minimum is used as the width measure. This is called the spatial length-width contextual measure. Generally road pixels have a bigger value for the length measure and smaller one for the width measure since roads are linear features. By combining those two measures, a candidate road image can be created by setting a length threshold and a width threshold. For a pixel, if its length measure is larger than the length threshold and its width measure is less than the width threshold, then it is assigned as a candidate road pixel. Here, based on the image characteristics, the length threshold is set to 50 meters and width threshold is set to 30 meters. Finally a morphological opening

is applied to the candidate road image to remove noise. All candidate road pixels have a unique value of one and other pixels are set to zero. This image is used later in the external image energy function in our snake algorithm. Note that this is a simple process that does not extract roads, but only provides areas where roads are likely.

4.2.2 Road Intersection and Road Termination Extraction

Road intersections and road terminations (endpoints) are generally very reliable and stable across different datasets. They represent robust information that makes them very useful for feature matching. However, few approaches make use of such pertinent information in the large amount literature on automated road extraction from remotely sensed imagery.

A line junction and line termination detection method was proposed in [113]. Given the lines already extracted from the original image, the local line curvature was measured. Two measures of curvature were presented: rate of change of the orientation vector's direction along the line and the mean of the dot products of orientation vectors within a given neighborhood. Since junction and termination locations do not necessarily correspond to local maxima of curvature, additional processing based on low curvature endpoints was done to localize junctions and terminations. Reference [114] classified the image into road class and background class. Then they utilized intersection shape models (crossroads, T-junctions, Y-junction) that consisted of elongated rectangles as templates. The intersections in the image were detected by model-based template matching. Similarly, in [115], image pixels were classified into on-road/off-road categories. A template around each intersection from the vector road network was generated using

information such as road widths and road directions. The localized template matching found the best corresponding point from the image in a rectangular area centered at the road intersection of vector data. Barsi and Heipke presented a feed-forward neural network road junction operator applied in a running window to decide whether or not it contained a 3 or 4-arm road junctions [116]. In [117], a wavelet transform was applied to the image to obtain wavelet coefficients at different levels. The road junctions and centerline pixels were detected based on the analysis of the characteristics of road pixels in the wavelet domain.

The approach in [113] requires lines extracted before junction detection. If the lines were converted into vector format and topology was created, it would be trivial to get the correct line junctions. In addition, the simple line detection algorithm they used often produced fragmented results that lead to many false intersections. For the template matching methods [114-115], only limited shape models can be provided. The angle and road branch width may vary significantly even for one type of intersection in a real image, and this causes matching issues. Using TIGER roads for the localized template matching, only 62% precision and 39.5% recall were obtained in [115].

Here we present a novel algorithm for road intersection and road termination extraction based on spatial contextual information. In the previous step, we calculated the spatial contextual information along each of the 360 directions for a given pixel. Therefore, we can construct a histogram to represent each pixel's spatial context. The x-value of the histogram is the direction range from 0 to 359 and the y-value is the distance between current pixel and the furthermost connected pixel within a certain similarity range along that direction.

Intuitively, we know that a point in the middle of a road will have two peaks with opposite directions in the histogram, while a point at a 3-way road intersection will have three peaks corresponding to the three radiating road arms. Similarly, four-way intersections have four peaks and road terminations (end points) have one peak in the histogram. Based on automatic identification of the number of peaks from the histogram, we can classify each pixel into different categories: 1 peak => road termination; 2 peaks => middle of road; 3 peaks => 3-way road intersection; 4 peaks => 4-way road intersection, etc.

First, we smooth the histogram to remove small bumps and dips that would hinder peak detection. We replace each value with the average value in a small neighborhood. Here, five neighboring cells are chosen. The new value is the mean of the histogram value of the current, its two left neighbors, and its two right neighbors.

For a point x in the histogram to represent a peak location, the y-value should gradually increase from the left side and decrease as we move to the right side within a neighborhood. The peaks are found by making use of the first derivative information. Also, the y-value at the peak must be larger than a given length threshold 50 meters is chosen. Based on general knowledge about the angles between two nearby road arms at an intersection, we can merge two adjacent peaks if the angle between them is too small, say less than 30^0.

Since roads in 1-meter resolution imagery are about 15 pixels/meters wide in Fig. 1, a cluster of pixels is detected at each road intersection and termination. A morphological opening is performed to remove noise, and small clusters are deleted also. For each remaining cluster, its centroid is chosen as a representative location for that road

intersection. For the test area in Fig. 1, all road intersections and road terminations corresponding to the vector road dataset, except for three, are successfully identified and correctly labeled. The three missed road intersections and terminations are extracted later by lowering the length threshold during the peak detection process. There are also a few incorrect road intersections and terminations, and these will be identified during the point matching step.

4.2.3 Point Matching by Relaxation Labeling

Feature matching is used to identify the correspondence from two different datasets as representations of the same geographic object. This is the most critical step of conflation. There are three basic types of map features: points, lines and polygons. Different matching criteria or methods can be applied to these feature types, with the most commonly used criteria being geometric, topological, and attributes [33].

Geometric criteria include length, perimeter, area, shape, angle, distance etc. Distance (proximity or nearness) between features is the most straightforward way to match common objects. A feature within a certain range of a given feature in the second dataset has a better chance of representing the same object. Euclidean distance is often used to calculate the proximity of point features. Hausdorff distance is more suitable for line to line matching. Other geometric criteria such as angular information of linear objects, or shape similarity of lines or polygons (length, perimeter, area etc) are also useful matching criteria.

Topology explicitly defines spatial relationships. The principle in practice is quite simple; spatial relationships are expressed as lists (e.g., a polygon is defined by the list of

arcs comprising its border). In many GIS systems, the vector data is stored using an arc-node data structure. The arc-node structure stores and references data so that nodes construct arcs and arcs construct polygons. Nodes define the two endpoints of an arc. An arc is composed of its two nodes and an ordered series of points which define its shape, called vertices. Nodes and vertices are represented as (x, y) coordinates. The arc-node data structure supports three major topological concepts [34]: 1) Connectivity; arcs connect to each other if they share a common node. 2) Area definition; arcs that connect to surround an area define a polygon. 3) Contiguity; arcs have direction and left and right sides. Two geographic features which share a boundary are called adjacent. Polygons are contiguous if they share a common arc. Contiguity is the topological concept that allows the vector data model to determine adjacency. The topology should be consistent in order to match features.

For the attribute criteria, if two datasets have some common attribute fields and the semantics of both sets are known, the attribute can be used to find the corresponding match. For example, the street name is frequently used for matching two vector road datasets.

In our road vector to imagery conflation application, there are no street names available from imagery and we have no intention to extract road centerlines from the imagery. Only the road intersections and terminations are readily computable from both the vector and image datasets. Therefore, a robust point matching approach is needed to find the feature correspondence.

Many point matching algorithms including clustering techniques, inter-point distance, relaxation etc. have been developed for feature-based image registration [77,

53]. Generally a global matching and transformation is sufficient for image registration. In our case, many local distortions and deformations exist due to the non-systematic differences between the two modalities. Thus, this situation is a more complicated non-rigid point matching problem.

The correspondence and the transformation are unknown in the point matching problem. Since solving for either without information regarding the other is quite difficult, most approaches to non-rigid shape matching use an iterated estimation framework. Given an estimate of the correspondence, the affine transformation may be estimated and used to update the correspondence [118]. The iterated closest point (ICP) algorithm assumes two point patterns are roughly aligned, for each point in one dataset, the closest point in the other data set is taken as the current estimate of the correspondence [67]. A non-rigid point matching TPS-RPM algorithm was proposed in [68], with the thin-plate spline (TPS) as the parameterization of the non-rigid transformation and the soft assign for the correspondence. In [119], another method for non-rigid point matching was proposed. The shape context is assigned to each point, which describes the distribution of the remaining points relative to this point. The solution that minimizes the overall shape context distances is the optimal match. In [118], point matching was formulated as an optimization problem to preserve local neighborhood structures during matching. Shape context distance was used to initialize graph matching, followed by a relaxation labeling process to refine the match. However, their algorithm was tested on the synthesized data only.

In our case, the local relationship among neighboring points is stronger and more stable than the global one. In addition some outliers may be generated from the image by

the road intersection detection algorithm, and some road intersections may not be extracted due to shadows, etc. Therefore, we chose the relaxation approach because it can tolerate local geometric distortions [77]. The basic idea is to use iterated local context updates to achieve a globally consistent result. The contextual constraints are expressed in the compatibility function. It is a bottom-up search strategy that involves local rating of similarity which depends on the ratings of their neighbors. These ratings are updated iteratively until the ratings converge or until a sufficiently good match is found. The local similarity measures are used to assign heuristic, fuzzy, or probabilistic ratings for each location. These ratings are then iteratively strengthened or weakened in accordance with the ratings of the neighboring measures [77]. An iterative relaxation algorithm was developed for point matching based on the relative distance information between points [70]. Here we apply and extend the approach as found in [78].

Let $A = \{A_1, A_2, \ldots, A_n\}$ be a set of road intersections and terminations from the vector road map, and $B = \{B_1, B_2, \ldots, B_m\}$ be a corresponding set from imagery. Suppose A_i and B_j are corresponding points. For any other point pair (A_h, B_k), their compatibility $C(i, j; h, k)$ was defined as a function of how much the actual position of A_h relative to A_i differs from the desired position of B_k relative to B_j [78]. Suppose B_j is shifted to A_i and becomes $B_{j'}$, and B_k is shifted the same amount to $B_{k'}$. The magnitude of the relative difference δ is the distance $D_{hk'}$ between A_h and $B_{k'}$. Then the compatibility is defined as $1 / (1 + \delta^2)$, that is, 1 for $\delta = 0$ and going to zero as δ becomes very large. In our application we divide δ by the distance from A_i to A_h (D_{ih}) to make it relative rather than absolute difference. The compatibility values are always nonnegative.

$$\delta = D_{hk'} / D_{ih} \tag{4.3}$$

$$C(i, j; h, k) = 1/(1+\delta^2) \tag{4.4}$$

Suppose P_{ij} represents the probability of a match between A_i and B_j. Then C (i, j; h, k) P_{hk} is used as a contribution to a new estimate of P_{ij}. The new estimate is commonly defined as an average of the previous estimate and the other points' contributions. In defining the net contribution of A_h to P_{ij}, it is reasonable to use the max, rather than the average of terms C (i, j; h, k) P_{hk}, since if any one of these terms is large there is strong support for P_{ij} from A_h, even if all the other terms are small. Thus a plausible relaxation formula in this situation is:

$$P_{ij}^{(r+1)} = \frac{1}{n}\sum_{h=1}^{n} \max_{k=1}^{m} \{ C(i, j; h, k) P_{hk}^{(r)} \} \tag{4.5}$$

Where r = 0, 1, 2, … is the iteration number. We define C (i, j; h, k) = 1 if h = i; thus the self-support term h = i in the average is just P_{ij}.

The initial estimates of the P's can be made in various ways. $P_{ij}^{(0)}$ can be defined with some measure of similarity between A_i and B_j. If the patterns match exactly, the P's do not change under the iteration process, but otherwise they decrease after each iteration. However, if a good correspondence exists between some of the A's and some of the B's, those P_{ij}'s for which A_i corresponds to B_j decrease slowly, since they have substantial support, while the other P_{ij}'s decrease much more rapidly [78].

To successfully implement the relaxation algorithm in our application, making use of all available information is necessary. The road intersections and road terminations for both vector and image representations have one common attribute that indicates the number of road arms emanated from that point. Road termination have a value of one, 3-way and 4-way road intersections have values of 3 and 4 respectively, etc. We use this attribute to initialize the $P_{ij}^{(0)}$ matrix: P_{ij} = 1 only if point A_i and B_j have the same

number of emanated road arms, i.e. the same type of road intersection or road termination, otherwise $P_{ij} = 0$. We do not allow a match between different road intersection types.

In our particular application, the road vector dataset is already roughly registered but has substantial local distortion. The points in a neighborhood have greater influence than points far away. The discrepancy within a neighborhood is relatively similar although the global discrepancy is non-systematic. Therefore, we further refine the initial probability matrix: $P_{ij} = 0$ if the distance between A_i and B_j is larger than a given threshold; 200 meters works well. A point can only be matched to another point within a certain distance.

Similarly, to calculate the new P_{ij} for point A_i during each iteration, we only consider the points A_h within a given distance to A_i. If the distance between A_h and A_i is larger than a threshold of 300 meters, it has no influence to P_{ij}.

This initialization and relaxation process works well for our application. Generally one iteration is enough to produce good results. We are only interested in finding the best matches. From the final P matrix, we find the maximum value for each row i, and if that value is also the maximum in its column j, then we say point pair (A_i, B_j) is a match.

In the test area shown in Fig. 1, there are 26 points from the vector road dataset. Among these points, 8 points are road terminations; 16 points are 3-way road intersections; and 2 points are 4-way road intersections. For the 31 points extracted from imagery, 13 points are classified as road terminations, 16 points are classified as 3-way intersections and 2 being called 4-way road intersections. The relaxation algorithm found 23 matches. Only 1 match was wrong and the other 22 matches were correct matches. For

the 26 points from vector dataset, 3 points did not find their correct correspondence because those road intersections / terminations were not detected from imagery.

To extract these missed road intersections and terminations, we run the road intersection detection algorithm again but lower the length threshold from 50 meters to 40 meters. Other parameters are kept unchanged. Therefore, more points are extracted, but there is more noise also. Since we are only interested in finding the missed intersections, we select those points close to the three unmatched road intersections of vector road, and add them to the originally extracted image point set from the first run. We perform the relaxation algorithm again. This time, all of the 26 matches are correctly found in Fig. 1. The 26 point pairs are used as control point in the next rubber-sheeting transformation. The remaining unmatched points are discarded.

In theory, this process could be iterated, but we have found that the two-stage intersection match was sufficient.

4.2.4 Piecewise Local Affine Transformation (Rubber-Sheeting)

A rubber-sheeting transformation is a local affine transformation which subdivides the map areas into pieces and applies local adjustments on each single piece, preserving topology in the process. This technique derives its name from the logical analogy of that of stretching a piece of rubber to fit over some object. Rubber-sheeting techniques typically subdivide the map areas into triangular-shaped regions. One such triangulation method is the Delaunay triangulation, considered by some to be the “best” triangulation for rubber-sheeting. Two characteristics distinguish the Delaunay triangulation: (1) no vertex other than the three forming a triangle is contained within the

circumscribing circle of that triangle, and (2) the minimum angles of the triangles formed are maximized [28]. The Delaunay triangulation maximizes the minimum angle of all the angles in the triangulation, thus avoiding triangles with extremely small angles.

Since the TIGER roads have non-systematic spatial errors, localized rubber-sheeting is especially suited for the vector-to-image conflation purpose. We use the matched road intersections and road terminations as control points to generate the Delaunay triangulation and perform the local piecewise transformation. The control points serve as vertices for triangulations to subdivide the map space. After rubber-sheeting, these control points (matched road intersections and road terminations) are moved into exact alignment and the displacements of other road points are interpolated according to the movements of the vertices of the triangles which contain them. The rubber-sheeting transform brings the vector roads much closer to true road locations in the image. Thus, the positions of the vector roads have been greatly improved.

4.2.5 Snake-based Position Correction

While the rubber-sheeting transform improves the positional accuracy significantly, there still are quite a few road segments that are not in correct alignment. An active contour model (snake) is used to deal with the residual displacement errors. In our earlier work [120], we applied the snake algorithm for conflating two vector road layers. Here we use the extracted length-width contextual information to create a candidate road image based on length/wide ratio. Then a mathematical morphology opening is used to clean the candidate road image. The cleaned candidate road image is used to generate the image energy function for a snake. Each vector road segment

consists of a sequence of vertices. We further densify the vertices at 30 meters' interval to better fit the curved roads. The road vertices are moved iteratively toward the candidate road image by the snake algorithm.

The snake is an active contour model under the influence of internal and external forces. The internal force imposes a piecewise smoothness constraint. The external image force pushes the snake toward salient image features like lines and edges [37]. The snake can be represented as parametric curve by

$$v(s) = (x(s), y(s)) \tag{4.6}$$

Where s is proportional to the curve length, and x and y are the curves coordinates.

The snake's total energy function is composed of internal and external components, given by

$$E_{Snake} = \int_0^1 E_{Snake}(v(s))ds = \int_0^1 E_{Internal}(v(s)) + E_{External}(v(s))ds \tag{4.7}$$

and in discrete form:

$$E_{Snake} = \sum_{i=1}^{n} E_{Internal}(i) + E_{External}(i) \tag{4.8}$$

Here, *i* represent a vertex point of the vector road segment.

The internal energy is usually based on the first and second derivative of the curve, constraining the snake to be smooth. In [37], the internal energy is represented as:

$$E_{Internal} = (\alpha(s)|v_s(s)|^2 + \beta(s)|v_{ss}(s)|^2)/2 \tag{4.9}$$

Where $v_s(s)$ and $v_{ss}(s)$ are first and second derivatives. In discrete form (9) becomes:

$$E_{Internal}(i) = (\alpha_i|v_i - v_{i-1}|^2 + \beta_i|v_{i-1} - 2v_i + v_{i+1}|^2)/2 \tag{4.10}$$

The internal energy is composed of a first-order term and a second-order term. The first term keeps the snake from stretching or contracting along its length (elasticity)

and the second term (curvature) keeps the snake from bending too much. Also $\alpha(s)$ and $\beta(s)$ are functions of the arc length along the snake and are used to weight the relative importance of the elasticity and bending. Usually, $\alpha(s)$ and $\beta(s)$ can be replaced by constants α and β to simplify computations.

The external energy represents the image force attracting the snake. It depends on image structure, usually the features of interest. The image energy can be defined as

$$E_{External} = -E_{\text{Image}}(v(s)) \tag{4.11}$$

Where $E_{Image}(v(s))$ is a function where high values correspond to the feature of interest. Normally, $E_{Image}(v(s))$ is the image intensity itself, I(x, y), or the magnitude of the image gradient, $|\nabla I(x, y)|^2$. For the latter case, the snake is attracted to positions with large image gradients. In general, the external energy is calculated from some feature image, called *f (i)* in what follows.

The image gradient image or line plausibility map generated using the line detector algorithm in [92] is too noisy in our case. The snakes will easily get stuck in local minimum and it is unlikely that a good solution can be found. Therefore, we use the binary candidate road image that was already created from length-width contextual measures. Candidate road pixels have value one and background pixels have value zero. This image provides a much 'cleaner' snake external image force.

To compute the minimum of the snake energy function, we used the approach shown in [37]. The energy minimization is solved using the Euler-Lagrange differential equation of motion with a discrete representation of the energies. The minimization is solved iteratively until convergence is reached. Detailed equations can be found in the appendix of [37].

Assuming that $\alpha(s) = \alpha$, and $\beta(s) = \beta$ are constants, minimizing the energy functional of equation (7) gives rise to the following two independent Euler equations:

$$\alpha x_{ss} + \beta x_{ssss} + f_x(i) = 0 \tag{4.12}$$

$$\alpha y_{ss} + \beta y_{ssss} + f_y(i) = 0 \tag{4.13}$$

Here, x_{ss}, y_{ss} are second derivatives, x_{ssss}, y_{ssss} are fourth derivatives, $f_x(i) = \frac{\partial E_{external}}{\partial x}$ and $f_y(i) = \frac{\partial E_{external}}{\partial y}$, where the derivatives are approximated by the finite difference in the x and y direction, respectively.

In our application, if a snake point (a vertex of the vector road) is located in the middle of background with value of zero, it won't feel any external force. To make the snake point move toward roads, we modify f_x(i) and f_y(i) according the following algorithm.

Suppose a snake has a sequence of points (v_1, v_2, …, v_n). We assume that v_1 and v_n are end points that have been moved the correct positions by the rubber-sheeting transformation. For any intermediate snake point v_i, we draw a line through v_i which is perpendicular to line $v_{i-1}v_{i+1}$. Given a standard coordinate axes, label that direction θ_i. We count the number of road pixels on each side of v_i along the line over a range of 50 meters. If the number on one side falls into a certain range, determined by information about road width, we move that point in the direction θ_i. The f_x and f_y are defined by the direction θ_i.

$$f_x(i) = 1, \quad f_y(i) = 1 \qquad if \quad 0 < \theta_i \le \frac{\pi}{2} \tag{4.14}$$

$$f_x(i) = -1, \quad f_y(i) = 1 \qquad if \quad \frac{\pi}{2} < \theta_i \le \pi \tag{4.15}$$

$$f_x(i) = -1, \quad f_y(i) = -1 \qquad if \quad \pi < \theta_i \le \frac{3\pi}{2} \tag{4.16}$$

$$f_x(i) = 1, \quad f_y(i) = -1 \qquad if \quad \frac{3\pi}{2} < \theta_i \le 2\pi \tag{4.17}$$

This assignment pushes the point in the correct direction. Once a snake point falls on a road pixel, it will stop moving. Only the points in the background will move. After a few iterations, most snake points will move to road pixels. Since the candidate road image often still contains noise, some points may get stuck in local minima. We implemented a simple perturbing algorithm to force the point move out of a local minimum.

In our experiments, the initial TIGER roads have bad positional accuracy. We do not care to keep the initial vector road shape. We are interested to move the vector as close as possible to candidate road image. Therefore, we set the snake parameters $\alpha = 0.1$ and $\beta = 0$. Thus, the snakes have considerable freedom of stretching and bending.

4.2.6 Refinement

Generally at this stage, the vector points fall on roads, but not necessarily in the center of roads because the snake algorithm moves the vector points to the edge of road image due to nature of the energy minimization. To make the result better, we developed an algorithm to move the vector points to the center of each road by using directional context information.

The centralizing algorithm works similar to the previous approach in finding a good moving direction for snake points. Suppose the point v_i is on a road pixel, we draw a line passing through v_i with the direction perpendicular to line $v_{i-1}v_{i+1}$. Then we find the

two points intersecting with the edges of the road. The middle point of the two intersecting edge points is used as the estimation of road central point.

Finally, we use some standard GIS operations to generalize and smooth the road centerlines. The refinement operation further improved the quality of vector roads. The vector roads are in very good alignment with imagery.

4.3 Experiments and Accuracy Assessment

To evaluate the performance of our automatic vector-to-imagery conflation approach, we ran the conflation system on several test areas with various global contexts such as rural, suburban, and urban. We manually extracted the corresponding roads from each test image as a reference for assessment. Then we compared the original TIGER road, rubber-sheeting transformed road, and final conflated result to the reference to illustrate how positional accuracy is improved.

We used two measures for the accuracy assessment: correctness and root-mean-square error (RMSE). First, buffer zones were generated around the reference roads. The chosen buffer width was approximately half of the actual road width. Then, by using GIS overlay analysis, road sections within the buffer zone of reference were found. The correctness is defined as the percentage of length of road segments that fall within the buffer over the total length of roads (18). For each vertex of the vector road, we calculated its shortest distance to the reference road. The RMSE can be estimated by (19).

$$Correctness = \frac{Length\ of\ \ Road\ \ within\ Reference\ Buffer}{Total\ \ \ Road\ \ Length} \tag{4.18}$$

$$RMSE = \sqrt{\sum \frac{(x_{target} - x_{reference})^2 + (y_{target} - y_{reference})^2}{n}} \tag{4.19}$$

We tested our vector-to-image approach using data covering Boone County, Missouri. The vector road centerline data are from the U.S. Census Bureau TIGER database. The orthorecitified multispectral near infrared aerial photographs are from the National Agriculture Imagery Program (NAIP). The USDA acquires NAIP imagery in the continental U.S. The NAIP imagery used for this conflation study has a one meter ground resolution.

We chose 21 tiles of size 700 x 700 m in rural, suburban, and urban areas. Figure 4.4 shows a rural test area, while figure 4.3 depicts a typical suburban scene. Table 4.1 illustrates the quality measures (correctness and RMSE) for all test areas. Note that the RMSE does not represent the true positional accuracy. It is relative accuracy using the NAIP image as a reference. NAIP imagery typically has a horizontal accuracy on the order of 5 m or less.

Figure 4.3 Comparison of original TIGER roads (green) and final conflation result (red) for test area shown in Fig. 4.1

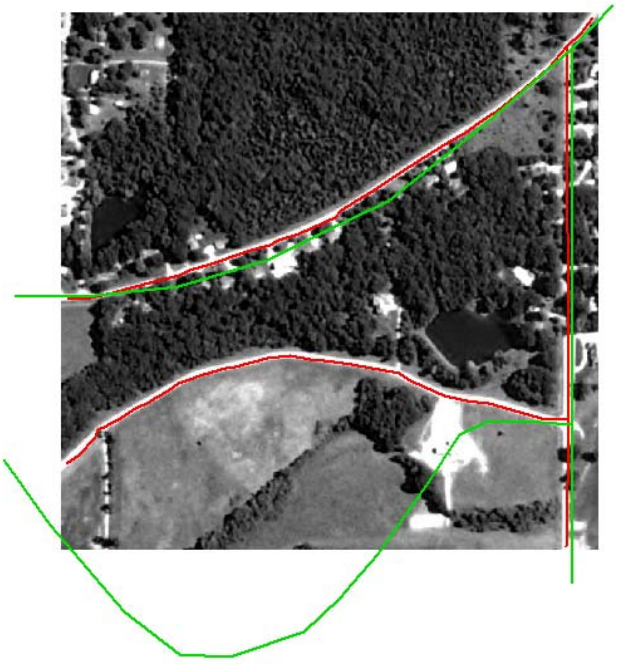


Figure 4.4 Comparison of original TIGER roads (green) and final conflation result (red) for a rural area. Our modified snake algorithm works well even some of the initial

vectors are far away (>300 pixels) from the correct positions. The gap caused by shadow in the middle of the road is not a problem.

Table 4.1 Accuracy assessment for 21 test areas

	Original TIGER road		**Rubber-sheeting transformed**		**Final results**	
	Correctness	RMSE (meter)	Correctness	RMSE (meter)	Correctness	RMSE (meter)
Test Image 1	19.6%	35.0	67.2%	13.2	99.5%	2.6
Test Image 2	17.7%	34.5	58.3%	20.8	88.7%	7.3
Test Image 3	25.3%	27.1	91.4%	4.8	99.6%	2.3
Test Image 4	7.7%	65.9	67.0%	8.4	96.8%	3.2
Test Image 5	36.4%	16.3	80.7%	6.7	96.0%	3.7
Test Image 6	1.7%	63.6	41.2%	22.8	95.8%	2.0
Test Image 7	3.3%	19.4	44.8%	7.9	100.0%	1.5
Test Image 8	4.6%	30.7	47.1%	9.8	86.6%	3.8
Test Image 9	3.6%	21.2	66.1%	8.4	98.2%	1.9
Test Image 10	0.6%	82.6	54.9%	6.8	93.6%	1.9
Test Image 11	41.2%	46.7	66.9%	11.3	98.3%	2.9
Test Image 12	59.2%	8.4	91.7%	5.2	91.7%	5.2
Test Image 13	9.4%	150.8	32.0%	125.3	100.0%	1.9
Test Image 14	58.6%	14.7	62.3%	9.5	97.0%	3.0
Test Image 15	1.1%	90.4	21.8%	60.9	97.4%	2.1
Test Image 16	1.0%	154.4	58.2%	10.7	92.9%	6.2
Test Image 17	0.6%	132.6	23.5%	25.4	95.4%	2.1
Test Image 18	24.1%	25.4	80.7%	8.9	97.4%	4.9
Test Image 19	14.4%	29.5	28.4%	18.5	88.4%	5.7
Test Image 20	53.6%	14.3	81.9%	7.9	98.9%	3.1
Test Image 21	48.3%	11.2	84.0%	4.8	93.6%	3.8
Average	20.6%	51.2	59.5%	18.9	95.5%	3.4

From table 4.1, we can see that original TIGER roads have poor positional accuracy with an average RMSE of 51.2 meters and correctness of only 20.6%. Several test areas have an over 100m's RMSE and many areas have correctness values of less than 10%. Our conflation approach improves the accuracy significantly. In fact, the average correctness reaches 95.5% and the average RMSE is only 3.4 m. This is an

incredible accuracy improvement. Note that in test area #12, no snake operation was performed since the road network is in a grid pattern with a line segment consisting of two end points. A human operator can iteratively correct a few imprecise road intersections for even further improvement.

4.4 Discussions and Conclusion

In this chapter, we presented a novel vector-to-image conflation approach. This technique uses a variety of algorithms to automatically detect and classify different types of road intersections and terminations based on spatial contextual measures. The vector road intersections are matched to extracted point sets by a relaxation labeling algorithm. The matched point pairs are used as control points to perform a piecewise rubber-sheeting transformation. With the end points of each road in correct position, a modified snake algorithm moves intermediate road points toward the road image. Finally, a refinement algorithm moves the points to center of each road to obtain the best cartographic quality.

Other snake-based road extraction approaches use simple filtering to detect lines from imagery and create line plausibility image. Often the simple line detector will create many false alarms and the line plausibility image is noisy. It requires that the initial vectors must be very close (within a few pixels) to solution. This definitely won't work in our application in which many of the initial vector roads are many tens or even hundreds of pixels away from desired positions. Additionally, in our conflation approach, there is no need to detect lines from imagery, only the points of intersection and termination are extracted. Although the spatial contextual information extraction process takes

considerable computational effort, the generated candidate road image is ‘cleaner’ and much better suited for snake application. So the far away vector roads still can be corrected and it is not sensitive to snake parameters.

Additionally, spatial contextual information is normally calculated in the Center for Geospatial Intelligence (CGI) at the University of Missouri for other purposes, such as feature extraction. Hence, in a normal operation mode, we are making use of already computed data.

Experiments show our vector-to-image conflation approach has excellent performance. The positional accuracy was improved significantly from 51.2 m RMSE to 3.4 m, and the average correctness increased from 20.6% to 95.5%. The improved TIGER roads with rich attributes could lead to many new applications for federal, state, and local governments as well as for private mapping industries.

However, some issues remain to be solved. To make the snake algorithm work, a good candidate road image created by spatial contextual information is necessary. In some areas, trees overshadow the roads or black bridges overpass bright roads, and length-width measure ratio thresholding may also cause some fragments in the candidate road image. If the gap is in the middle of road, the snake still can close the gap successfully. But if the gap is near intersection, that road intersection cannot be correctly identified. Snake algorithms may have trouble in such areas. In further research, we will develop a gap closing algorithm in the spatial contextual information processing. A small jump will be allowed. This should solve some of the problems. In some old residential areas, if the roads are almost completely overshadowed by trees, there is no way for the algorithm to succeed. Other “leaf-off” images need to be acquired or GPS technology

must be used in such areas.

Currently, we use a simple algorithm to find the moving direction for snake points. The direction of motion of each point is calculated independently. We intend to improve the algorithm by considering the movement of other points in a neighborhood. Each point will have repelling force to keep others away.

In this study, we integrated and conflated existing vector roads to imagery. A better road dataset is created by improving the positional accuracy while keeping the existing attributes. Generally the imagery is more current than the vector data because traditional map compilation and production are time consuming processes. Newly constructed roads are often not shown in vector datasets. The unmatched road intersections from imagery may provide useful clues for automatic extraction of new roads from imagery, which will facilitate map updating. In our next step, we will combine this vector-to-imagery conflation approach and an automatic road extraction approach to address the digital map revision and updating issue.

Chapter 5

Scale-up Experiment of Vector-to-Imagery Conflation

5.1 The need for scale-up experiment

In chapter 4, we already selected 21 tiles of size 700 m by 700 m in rural, suburban, and urban areas (figure 5.1) to test our vector-to-imagery conflation algorithm. The selection criteria are following: there exist obvious displacements between vector roads and imagery; majority of the roads are easy to see in imagery to facilitate image processing; and the vector roads have good one-to-one correspondences to imagery.

Although our algorithm works well on those selected small test areas as shown in table 4.1, a scale-up experiment is needed to find out any potential issues and make improvement to the algorithms. Thus to make our algorithms useful to solve a real world conflation problem, two big test areas, each with 6000 m by 6000 m dimension are selected (figure 5.2). They represent typical rural and suburban scenes respectively in Columbia, MO. Each of the two big areas is subdivided into 3 by 3 tiles of size 2000 m x 2000 m (figure 5.3 & 5.4). The vector-to-imagery conflation algorithm run on those 18 tiles separately.

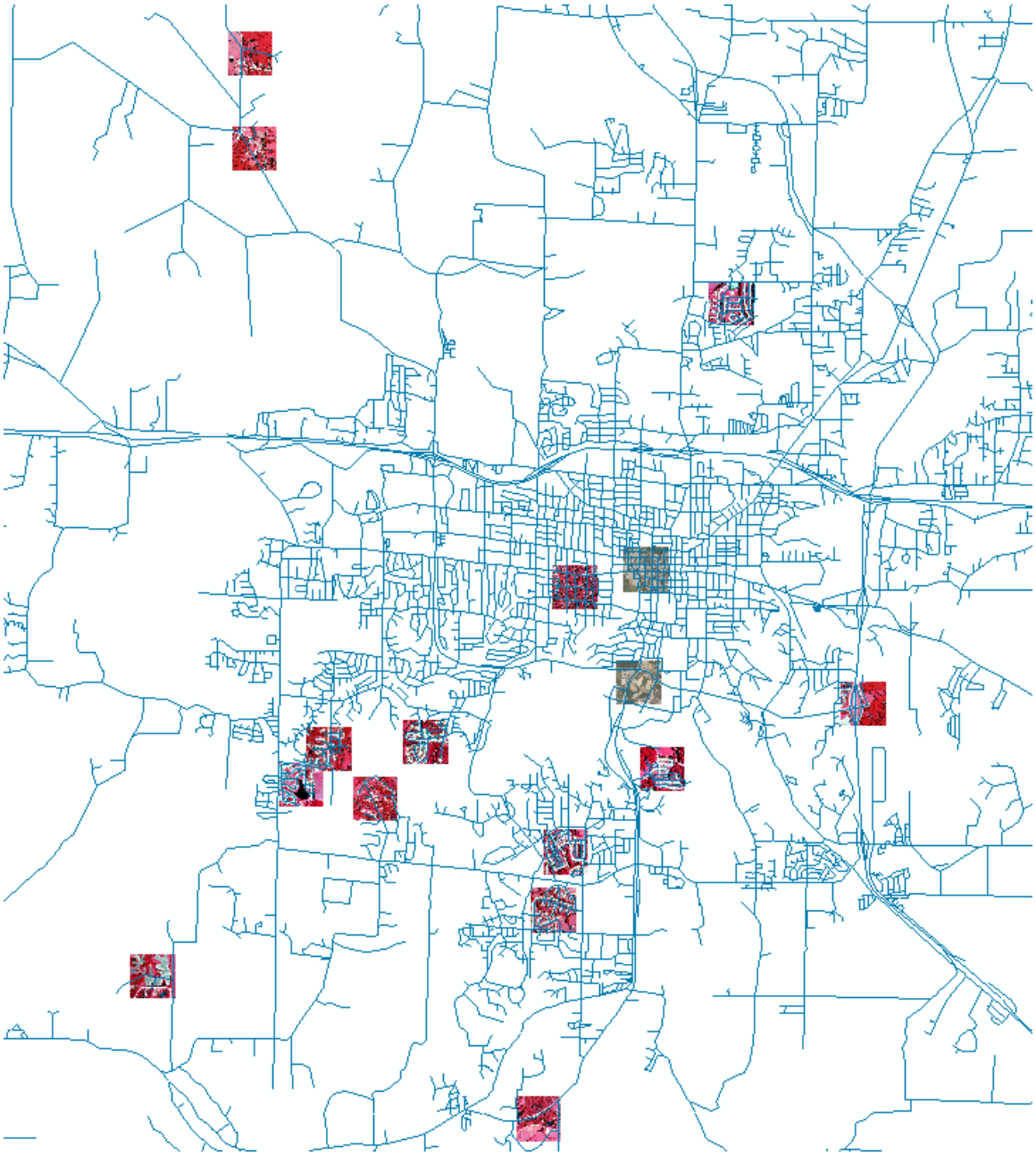


Figure 5.1 Selected test tiles used in table 4.1

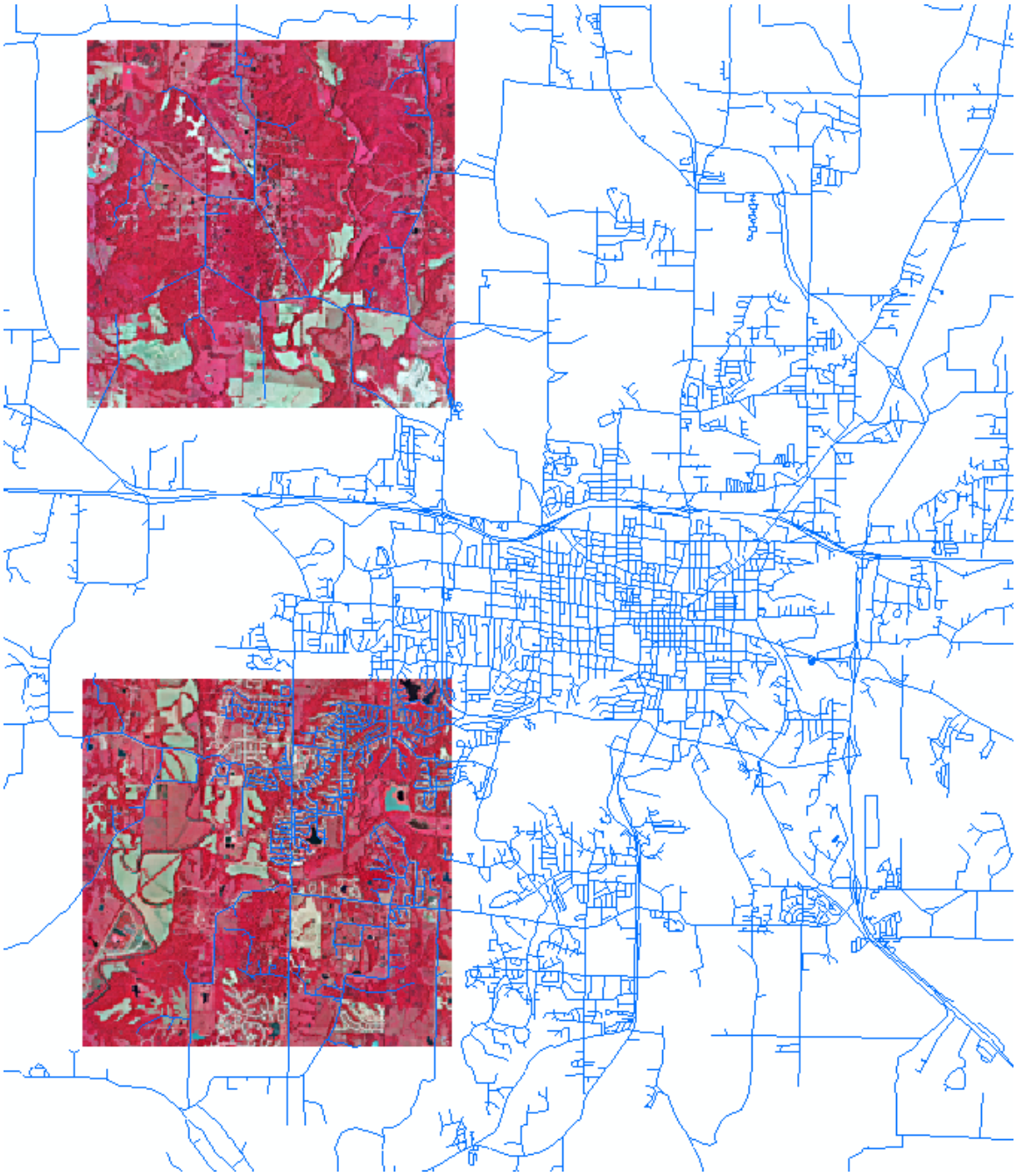


Figure 5.2 Two big scale-up test areas in Columbia, MO

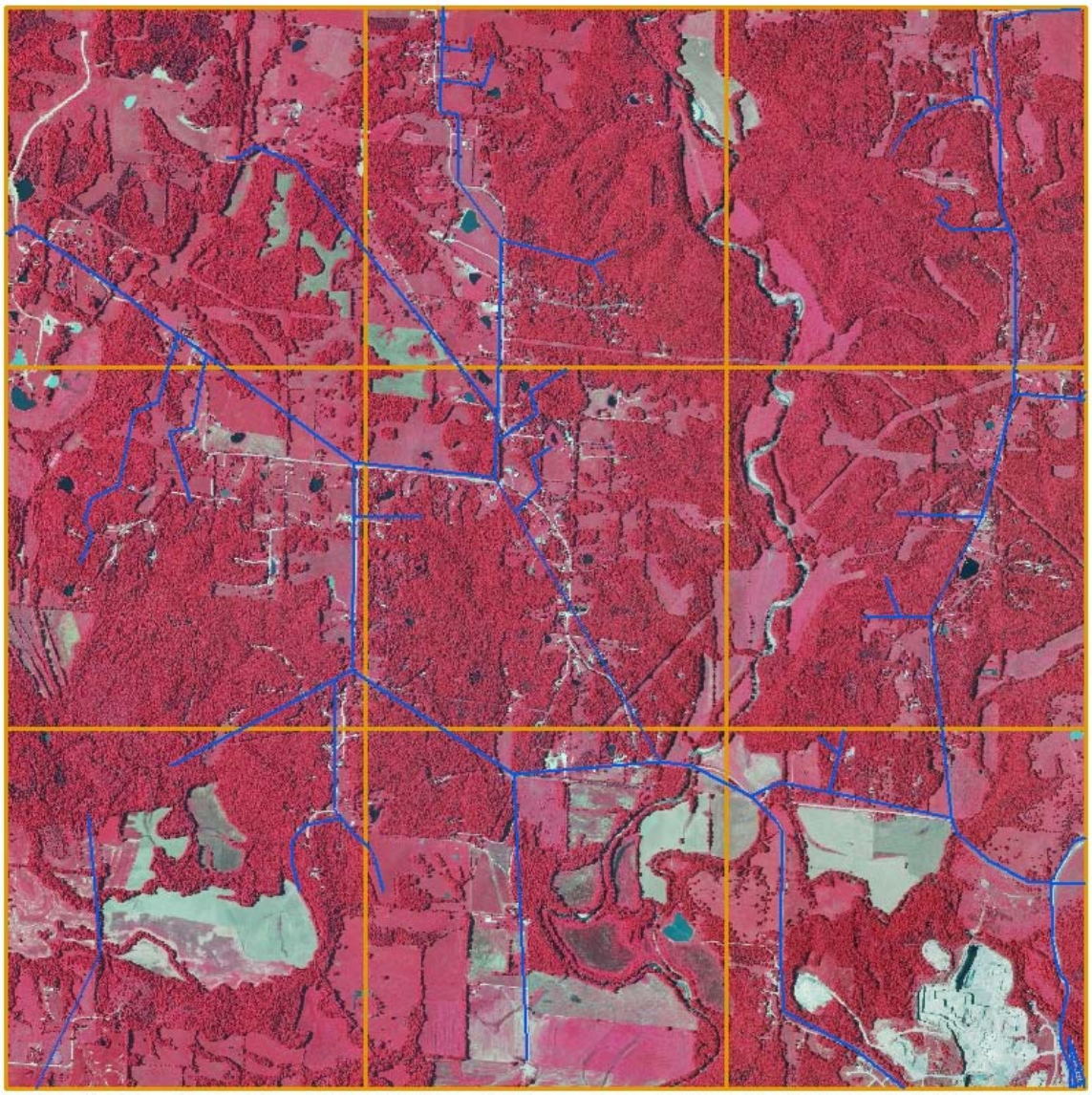


Figure 5.3 TIGER roads (blue) of nine test sections in rural area of Columbia, MO, each section has 2000 m * 2000 m dimension.

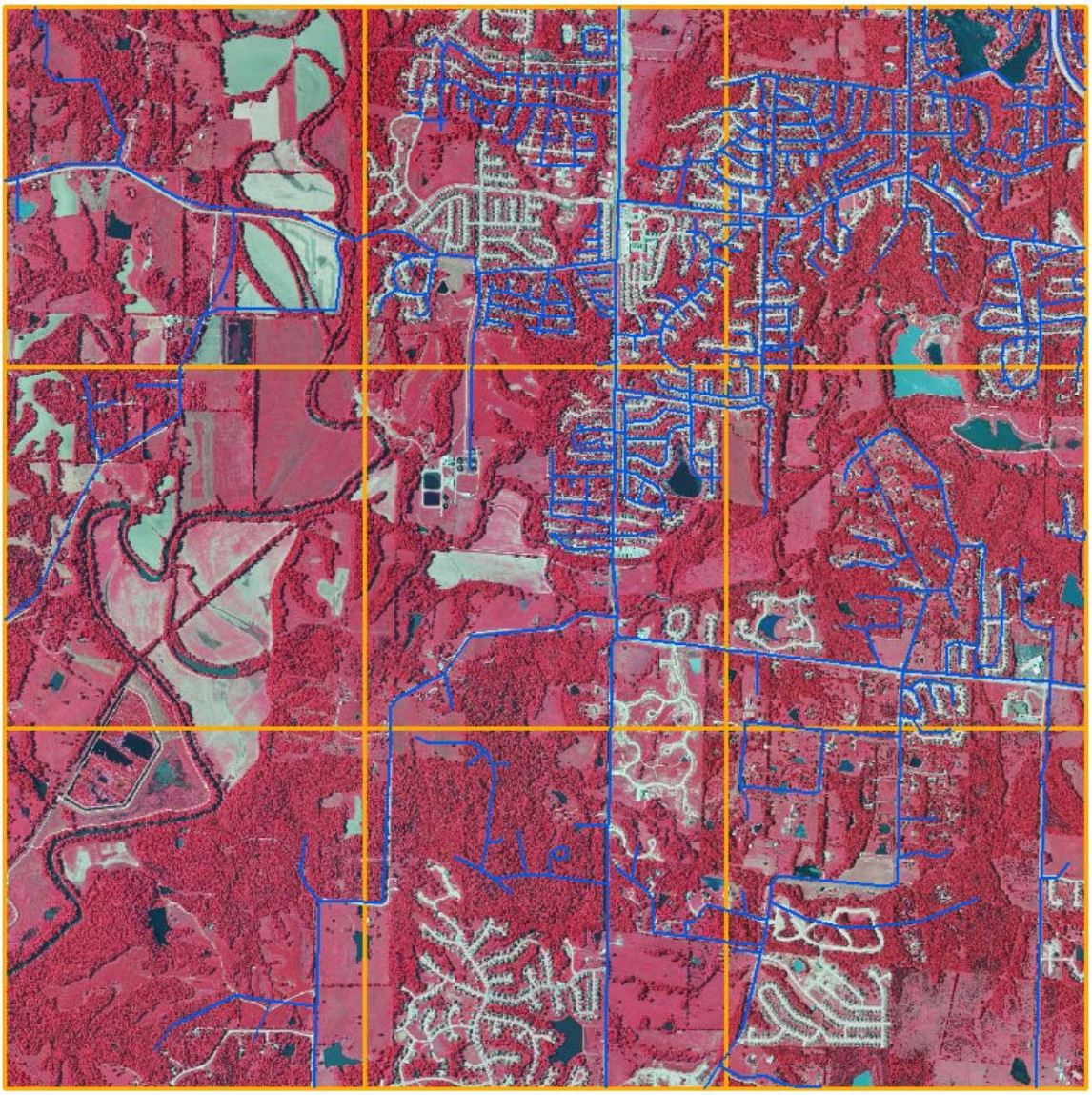


Figure 5.4 TIGER roads (blue) of nine test sections in suburban area of Columbia, MO, each section has 2000 m * 2000 m dimension.

5.2 Use classified road imagery to speed up the process

In chapter 4, to calculate the spatial contextual measures and extract the road intersections, the algorithms need to examine each pixel and search its surrounding pixels along given directions radiating from the central pixel. Then the algorithm calculates its

similarity with surrounding pixels. This task is very time-consuming. It may take hours for larger areas since this is done on the original imagery. Much of computing is useless for non-road regions in imagery such as forest, grass, water etc.

To improve this process, a simple idea is to classify the imagery into two classes: road and non-road. Here we use the standard ISODATA unsupervised learning algorithm to generate clusters and interactively label each cluster as road or non-road class. Then we run our algorithm only on the classified road regions. Figure 5.5 shows a suburban test area and the classified results are shown in figure 5.6. However, since buildings, parking lots have similar spectral attribute with roads, they are connected to roads to form bigger regions. This caused a problem during the thresholding process (figure 5.7). If we choose a bigger threshold, we can get clean major roads, but many shorter road segments were removed (figure 5.8). If a smaller threshold is used, the result is quite noisy and includes driveways and buildings (figure 5.9). The snake operation may get stuck on the noisy road image.

In this scale-up experiment, we used the 4-bands color infrared aerial photography obtained from city of Columbia. It has very high resolution (0.15m). Obviously many small details can be captured in such a high-resolution imagery. Since we are dealing with road centerlines in our conflation application, it's not harmful to resample the imagery to lower resolution. We decide to resample the 0.15 m imagery to 1 m resolution. This will help to remove small details and speed up the process.

When we closely examine the imagery, we found that the majority of roads are brighter and buildings are relatively darker, but some roads are dark colored in selected

areas. We reassign the previous established clusters into three classes: brighter road, darker road & building, and non-road class.

Figure 5.5 Color infrared image of one suburban test area.

Figure 5.6 The road/building regions for figure 5.5

Figure 5.7 The grayscale image of length measure.

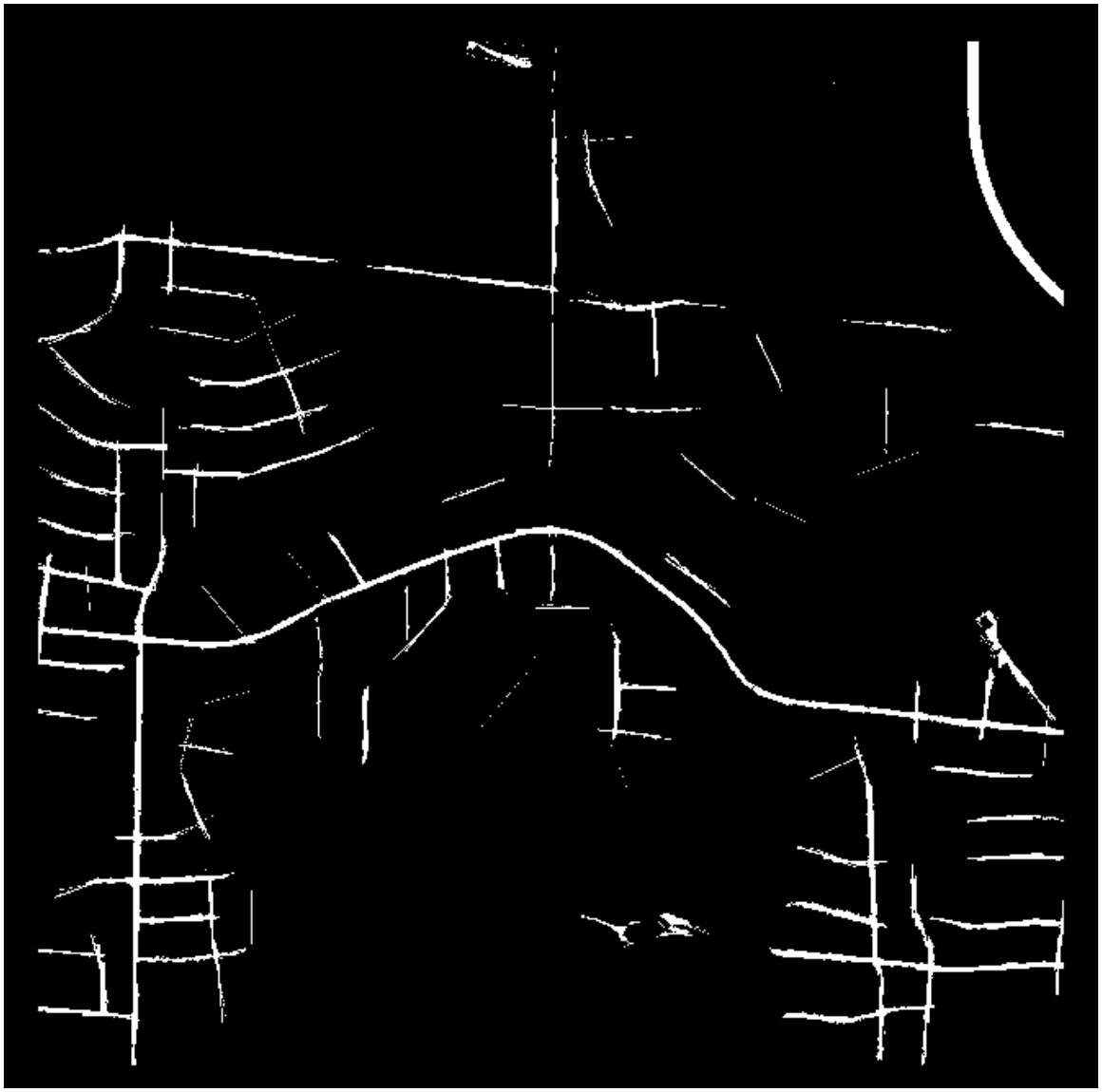


Figure 5.8 Result of road image using bigger threshold (Length > 100 m)

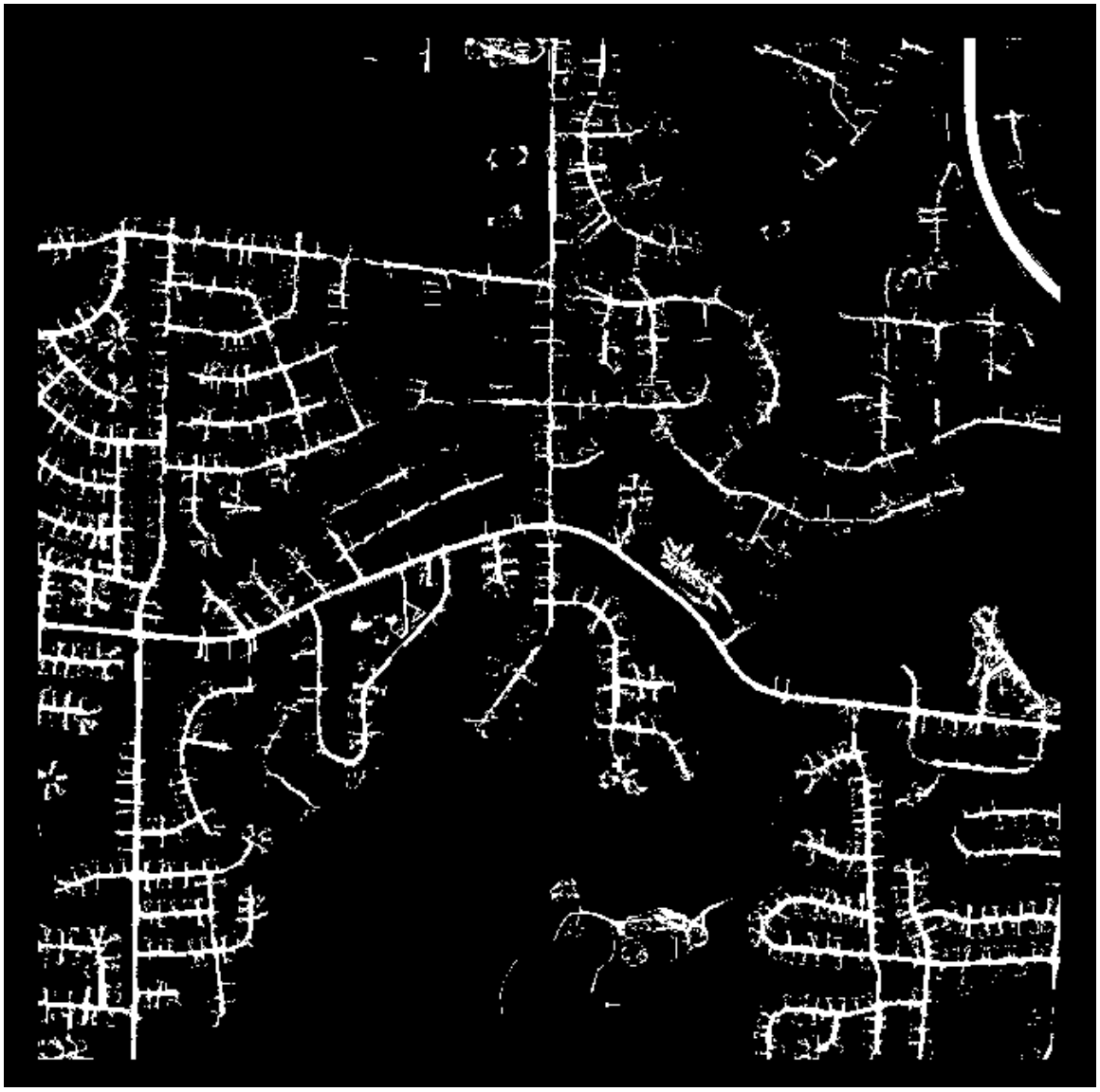


Figure 5.9 Result of road image using smaller threshold (Length > 30 m)

Figure 5.10 The classification results. Bright roads are red. Darker roads and buildings are green. Black are non-road class.

Figure 5.11 The final result after removing smaller regions

Figure 5.10 showed the 3-class classification results. Since buildings are darker, they can be separated from bright roads (red). Fortunately the driveways are bright (red) in darker road regions, thus make the buildings disconnect to dark roads. Therefore, we process the dark class to remove the buildings using an area threshold since they are smaller rectangular shaped regions, while the roads are bigger connected linear regions. Finally, we add the processed dark class back into the bright class to create the final road image class (figure 5.11). It can be seen that most of buildings are removed successfully. This will facilitate the next spatial context measure extraction. A single threshold will generate very clean results (figure 5.12).

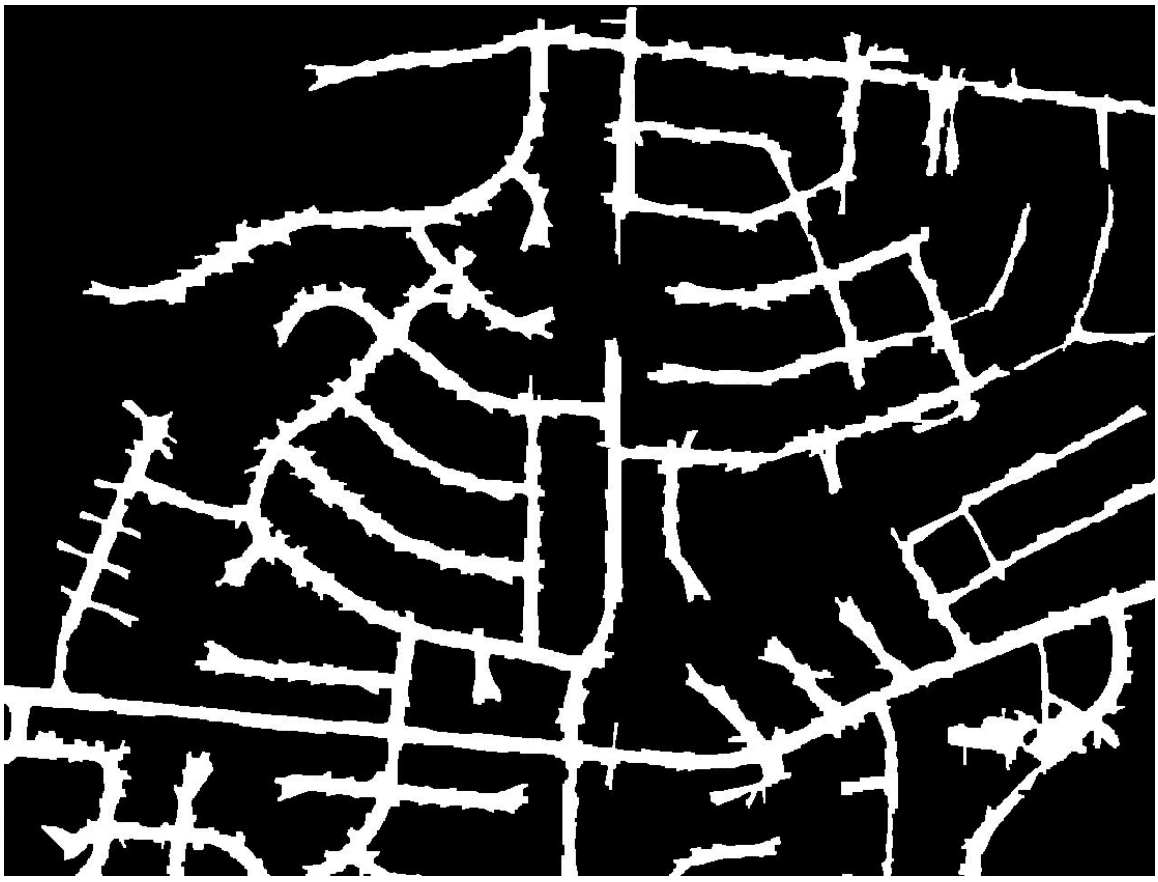


Figure 5.12 The road image is generated by thresholding the length measure.

5.3 Accuracy assessment

Our vector-to-imagery conflation algorithms were run on the 18 test areas. All of the images used in experiment and results are listed appendix A. To evaluate the result, we use the high accuracy MODOT roads from Missouri Department of Transportation (MODOT) as the reference. The original TIGER roads and final conflated results generated by our vector-to-imagery conflation algorithm are assessed to show the positional accuracy improvement.

The same accuracy assessment measures (correctness and root-mean-square error RMSE) used in chapter 4 are used again. In addition, several criteria are added: total

length of roads, length of road in correctness zone, number of different node types (intersections & terminations), and the maximum offset distance. The accuracy assessment is performed in ArcGIS and results for all the test areas are listed in table 5.1.

Table 5.1 Accuracy assessment for 18 scale-up test areas

	Original TIGER Roads						Conflated Results				
	Total Length	Correct Length	No of Nodes types (1,3,4)	Correctness	RMSE	Max	Total Length	Correct Length	Correctness	RMSE	MAX
Rural01	3364	88	7, 1,0=8	2.6%	450	1126	3474	1578	45.4%	445	1165
Rural02	3780	522	6,2,0=8	13.8%	220	772	3999	2658	66.5%	215	788
Rural03	6789	1144	8,4,1=13	16.9%	90	429	6918	4170	60.3%	55	386
Rural04	5438	68	10,4,0=14	1.3%	159	418	5924	2622	44.3%	32	110
Rural05	5204	178	10,4,0=14	3.4%	94	360	5528	3240	58.6%	70	358
Rural06	3633	109	7,5,0=12	3.0%	176	533	3785	1668	44.1%	140	495
Rural07	2792	136	6,2,0=8	4.9%	115	326	2777	1006	36.2%	117	307
Rural08	4447	125	8,4,0=12	2.8%	131	321	4072	2914	71.6%	37	238
Rural09	4077	331	5,3,0 = 8	8.0%	198	595	4257	1802	42.3%	150	570
Average				6%	181				52%	140	
Urban01	2941	337	5,3,0=8	11.5%	115	269	2752	2540	92.3%	20	116
Urban02	4731	1315	11,3,0=14	27.8%	29	116	4393	3826	87.1%	22	100
Urban03	9654	2203	18,10,0=28	22.8%	97	405	10356	8544	82.5%	68	399
Urban04	2699	366	5,3,0=8	13.6%	32	98	2684	2572	95.8%	7	39
Urban05	13610	5960	24,34,5=63	43.8%	21	192	14715	12763	86.7%	15	180
Urban06	15104	3970	47,31,5=83	26.3%	38	289	15171	12248	80.7%	32	302
Urban07	5718	198	5,5,0=10	3.5%	179	512	5284	5029	95.2%	9	50
Urban08	20921	5080	77,65,15=157	24.3%	31	146	22628	19516	86.2%	12	158
Urban09	31973	8443	92,98,23=213	26.4%	255	196	34613	29283	84.6%	12	168
Average				22%	89				88%	22	
Downtown	20600	16188	18,50,49=117	78%	14	43	20531	19262	94%	10	39

From table 5.1, we know that the original TIGER roads have really bad positional accuracy. The average correctness is only 6% for rural area and 22% for suburban area. The root-mean-square error is 181 m and 89 m for rural and suburban area respectively.

Our vector-to-imagery conflation approach greatly improved the TIGER roads. The correctness increased to 52% and 88% for rural and suburban area. The RMSE decreased to 140 m and 22 m for rural and suburban area respectively. That is a very significant improvement.

Although the correctness increase from 6% - 52%, 22% - 88% are significant, the absolute value of 52% is not satisfying. We further check and compare the TIGER roads, the conflated roads, MODOT reference road and imagery. There are a few reasons for this low correctness and high RMSE.

First, the original TIGER roads have errors. The errors have no corresponding parts in reference roads. In figure 5.12, the TIGER road segments within the black oval are errors. They have no counterparts in the reference and proved to be nonexistence in reality when they are overlaid on imagery (figure 5.13).

Secondly, the TIGER roads were created in 1999 by the U.S. Census Bureau for census 2000. MODOT roads are maintained and frequently updated by Missouri Department of Transportation and the most current version was created in 2009. The two agencies have different responsibilities and may use different methods and standards to create the data. For example, in figure 5.15, MODOT only cares about the major roads, but the U.S. Census Bureau needs to present each road as detail as possible. That caused some mismatch between the two datasets.

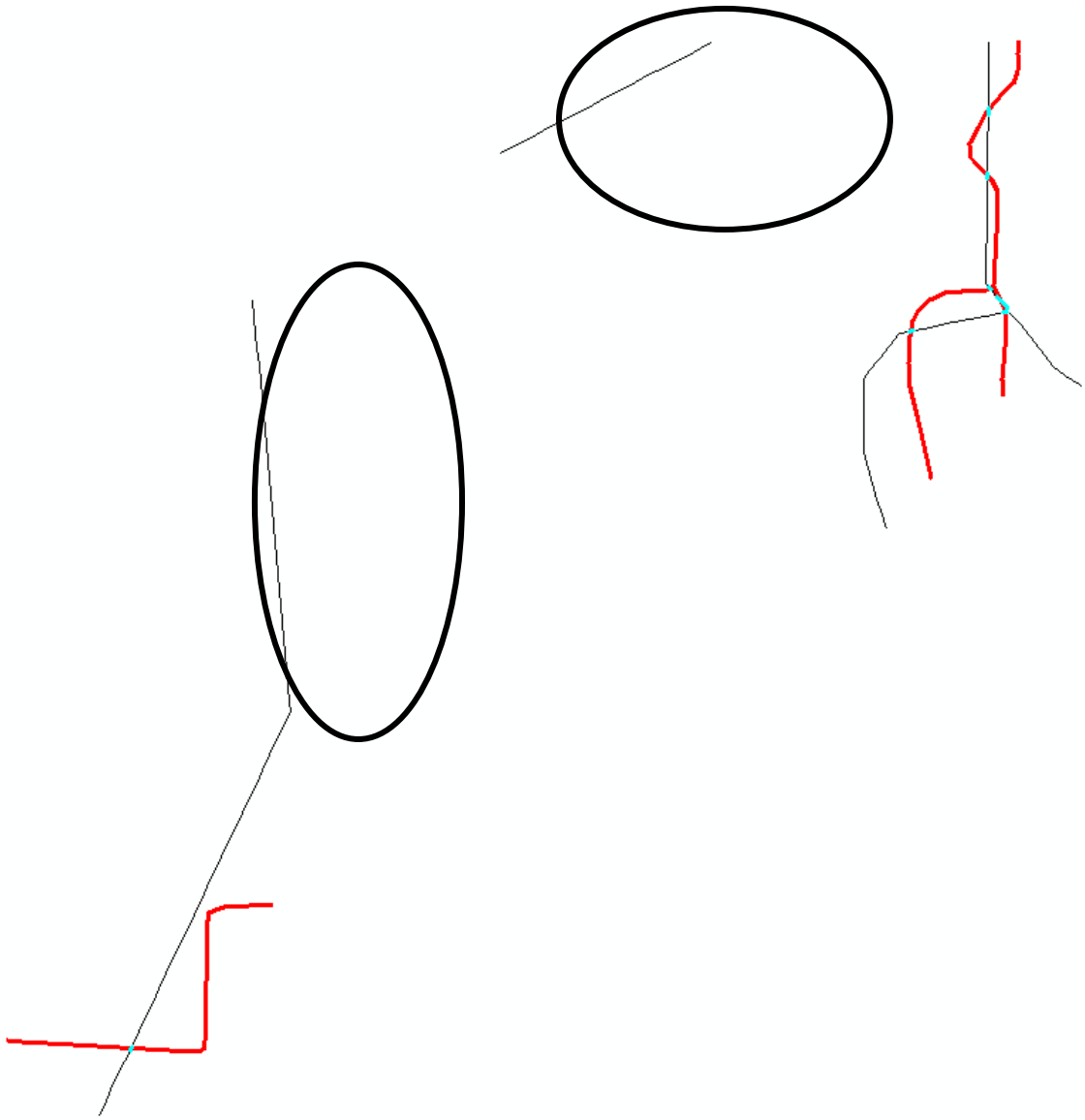


Figure 5.13 The TIGER Roads are overlaid on top of MODOT road buffer zone (red).

Figure 5.14 The TIGER roads are overlaid on imagery. The road segments within the black oval are errors. Actually they do not exist.

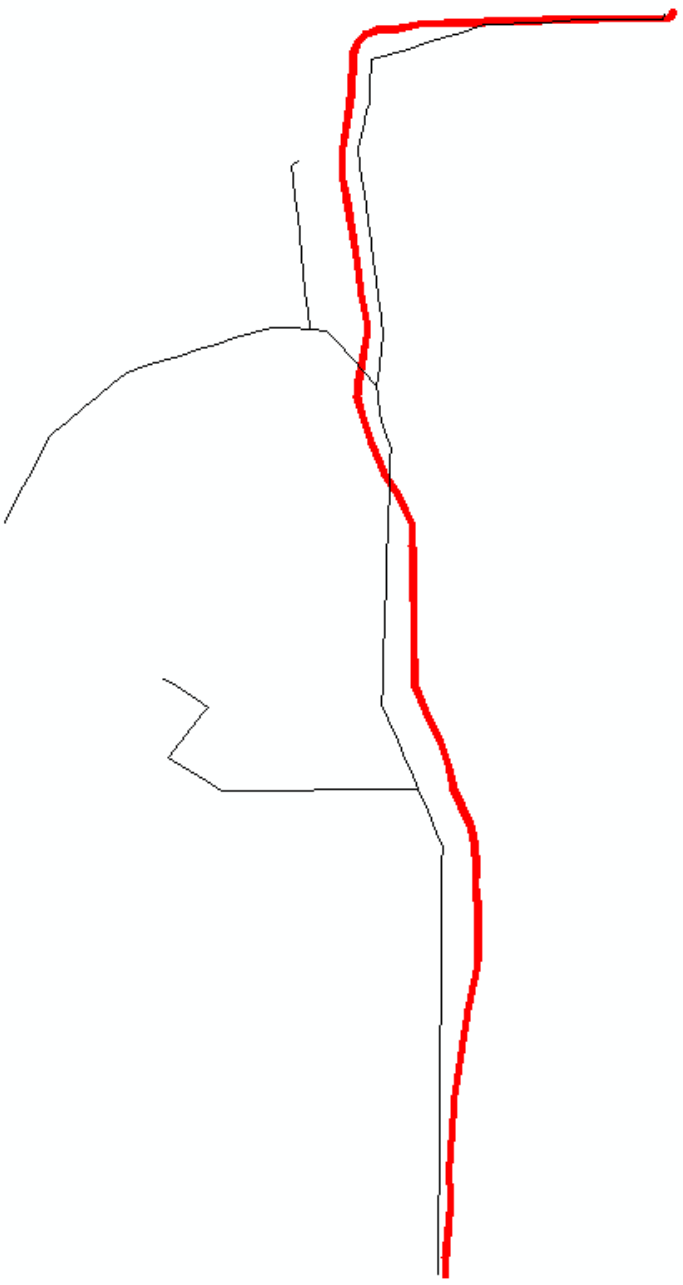


Figure 5.15 TIGER roads are overlaid with MODOT road buffer zone (red). TIGER and MODOT Roads are generated by different agencies and using different standard. In case like this, MODOT picks up only the major roads and TIGER has more detailed minor roads.

Figure 5.16 The conflated TIGER roads are overlaid on buffer zone generated from MODOT roads.

Figure 5.17 The conflated TIGER roads are overlaid on road image. Tree shadows cause gaps in classified road image and short road segments are removed during our conflation process. Therefore, some roads could not find their correspondences in road imagery.

Thirdly, the high resolution aerial image was acquired in summer season. The shadows cast by trees caused some gaps in road image. The following image processing, such as clean-up and thresholding further removed those short road segments. As shown in figure 5.16, some vector roads could not find their correspondences in road imagery.

Therefore, the situations mentioned above make the assessment results unreasonably low when all the data are used. The missing/unmatched road segments need to be eliminated to do a true evaluation. Thus, we removed the major parts of missing/unmatched road segments from original TIGER roads and conflated roads. Then run the accuracy assessment again. The results are listed in table 5.2.

By comparing tables 5.1 and 5.2, we noticed that the correctness measures are only slightly better for TIGER roads: 6% - 8% for rural and 22% - 23% for suburban area. The RMSE decreased a lot from 181 m to 79 m for rural, 89 m to 56 m for suburban area. Generally the original TIGER roads still have bad positional accuracy.

In contrast, after eliminating the major part of missing/unmatched roads, the conflated roads have very good accuracy. The correctness is 83% and RMSE is 5 m for rural area, and the correctness is 93% and RMSE is 9 m for suburban area. The statistic numbers as well as the graphics shown in appendix A prove that our vector-to-imagery conflation approach greatly improved the positional accuracy of TIGER road data.

Table 5.2 Accuracy assessment for 18 scale-up test areas after removing major unmatched roads

	TIGER Roads					Conflated Roads				
	Total Length	Correct Length	Correctness	RMSE	Max	Total Length	Correct Length	Correctness	RMSE	MAX
Rural01	1749	88	5.0%	68	147	1717	1477	86.0%	5	32
Rural02	2696	455	16.9%	40	113	2602	2548	97.9%	3	16
Rural03	5606	763	13.6%	27	91	5535	4054	73.2%	6	34
Rural04	3435	27	0.8%	200	418	2661	2304	86.6%	3	8
Rural05	4087	169	4.1%	96	360	3608	3016	83.6%	3	6
Rural06	2357	100	4.2%	69	164	2416	1767	73.1%	10	45
Rural07	1152	128	11.1%	62	147	973	909	93.4%	2	7
Rural08	3813	125	3.3%	114	302	3637	2900	79.7%	4	22
Rural09	2436	316	13.0%	35	63	2499	1853	74.1%	8	35
Average			8%	79				83%	5	
Urban01	2941	337	11.5%	115	268	2650	2583	97.5%	4	19
Urban02	4731	1315	27.8%	29	116	4393	3826	87.1%	22	100
Urban03	8104	2199	27.1%	51	265	8586	8446	98.4%	4	25
Urban04	2699	366	13.6%	33	98	2684	2572	95.8%	7	39
Urban05	12332	5647	45.8%	16	76	13146	12131	92.3%	8	49
Urban06	12940	3543	27.4%	23	84	12943	12025	92.9%	7	53
Urban07	5718	198	3.5%	179	511	5284	5029	95.2%	9	50
Urban08	20506	4830	23.6%	31	146	21180	19084	90.1%	7	63
Urban09	31973	8483	26.5%	25	196	34613	29283	84.6%	12	168
Average			23%	56				93%	9	
Downtown	20600	16188	78%	14	43	20531	19262	94%	10	39

Over the entire conflation process, the contextual signature extraction algorithm is the most time-consuming. Since we have the same 2000 m by 2000 m image size for each tile, we calculate the total number of classified road pixels in each tile and record the execution time of contextual signature extraction algorithm. The result is listed in table 5.3. It is obvious that it takes more time to process denser road regions. The algorithm is run on a DELL Optiplex 755 with Intel Core 2 Duo E4500 2.2 GHz CPU and 2GB RAM.

Table 5.3 Execution time for spatial contextual signature extraction algorithm

	Number of Classified Road Pixels	Execution Time (Minute)
Rural01	20092	3.4
Rural02	31673	3.5
Rural03	94110	4.4
Rural04	93382	4.3
Rural05	35335	3.6
Rural06	24568	3.4
Rural07	27896	3.5
Rural08	30020	3.5
Rural09	76698	4.0
Urban01	133812	4.7
Urban02	227745	5.6
Urban03	216951	5.8
Urban04	93474	4.2
Urban05	197356	5.3
Urban06	320800	7.1
Urban07	78541	4.1
Urban08	574985	9.5
Urban09	509017	9.5
Downtown	612890	8.7

5.4 Edge Matching

Since we run the vector-to-imagery conflation on each 2000 m by 2000 m chip individually, there are some scale-up issues when the results from individual chips are put together. Figure 5.18 shows all the conflated roads in the 9 chips of the rural test area. We can see some gaps near the edges of chips. This is a common edge matching problem in GIS world. ArcGIS has existing tools to deal with the edge matching issues. In the edge matching process, the features along the boundaries of adjoining tiles are spatially aligned to ensure the features are continuous and will meet at the join.

First we append all the conflated roads from the 9 chips together into one big new data layer. Since the chips have fixed size and locations, we run a script to automatically generate several buffer zones along the edges of the adjoin chips with a given buffer size. Figure 5.19 shows an example of the generated buffer zones. Here the buffer distance is set as 50 m. Then the script selects all the road endpoints within the buffer zones. Only those selected endpoints will be further processed and leave other road features unchanged.

Snapping is one of the edge matching tools that can be used to make sure the features properly connect together, including avoiding gaps and overlaps. Snapping is the easiest ways to more accurately position features and establish exact locations in relation to other features. There are two parameters need to be specified for the snapping environment: snapping tolerance and search method.

The snapping tolerance is the distance within which a feature is snapped to another location. Two points are automatically snapped together if they fall within a

given distance of each other specified by the snapping tolerance. If a feature A such as a vertex or endpoint is within the distance to another feature B, it will be automatically snapped to feature B.

Snapping search method defines how to search the snapping points. There are two options: first and closest. When the search method is set to first, the selected point will be snapped to the fist point found within the snapping tolerance. The closest method will search the entire area inside the snapping tolerance from the point under consideration and snap to the closest point found. It requires more processing time and may be slightly slower than using first method, but the result is better and more predictable since if there is more than one point within the search distance, the first point found may not be the closest.

In our study, we set the snapping tolerance to 50 m and choose the closest search method. Therefore, the snapping tool automatically moves the selected endpoints along the edges to their closest endpoints found within the snapping tolerance. Figure 5.20 give an example to illustrate the edge matching result. The final edge matching results for rural and suburban test areas are shown in figure 5.21 and figure 5.22. Almost all of the gaps along the edges are closed successfully.

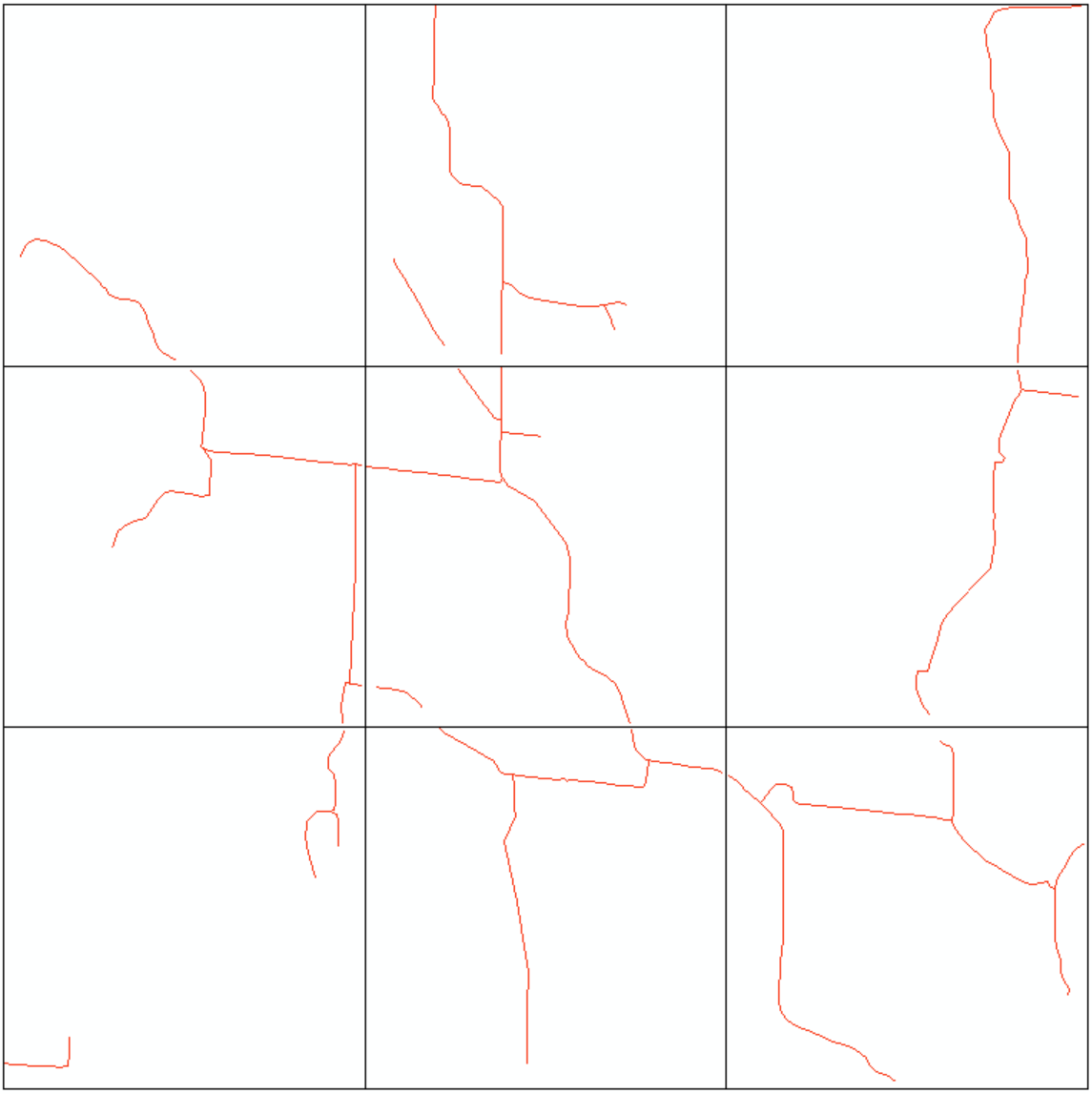


Figure 5.18 the conflated roads in rural areas before edge matching. Some gaps along the edge of chips are obvious.

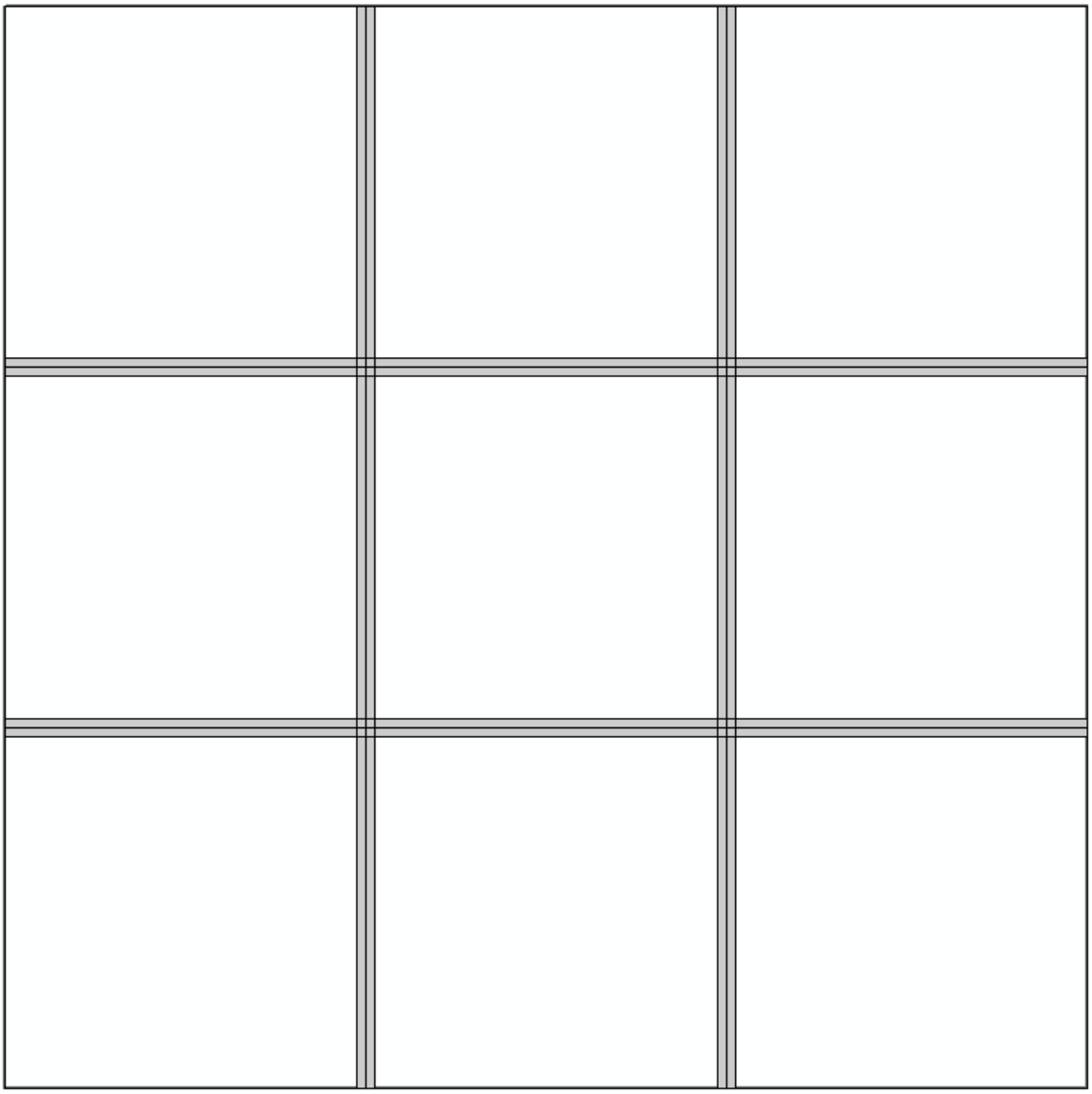
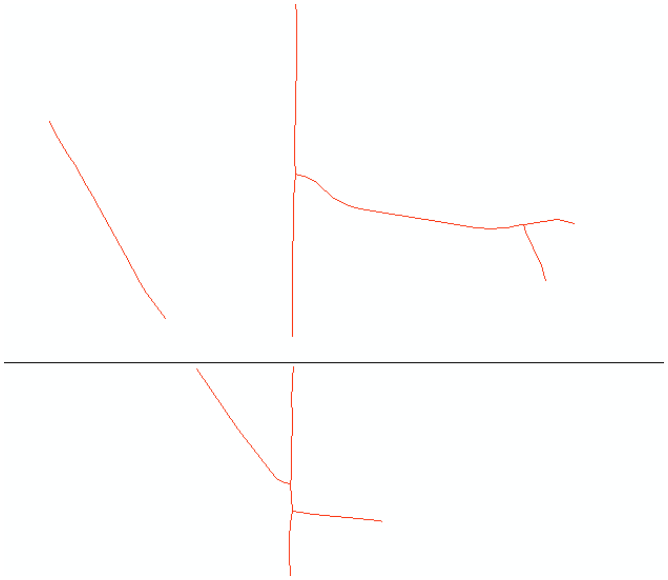
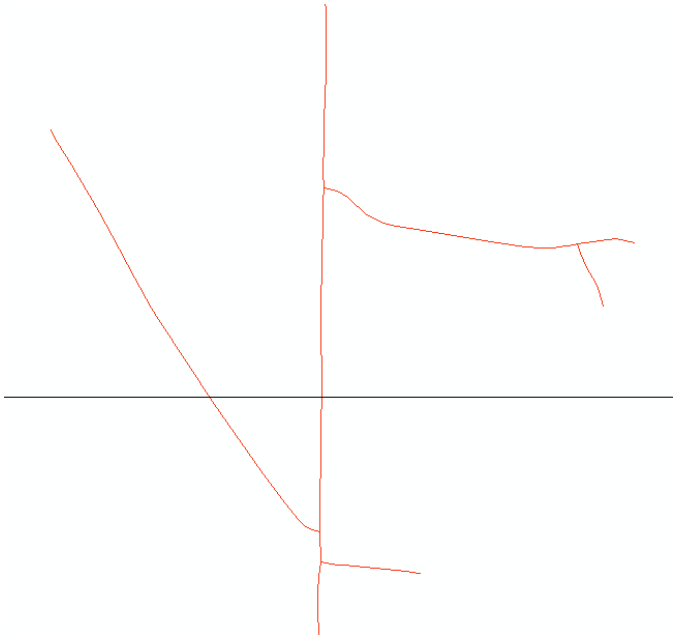


Figure 5.19 The generated buffer zones (gray areas) along the edges of adjoin tiles.



a) Roads before edge matching



b) roads after edge matching

Figure 5.20 a simple example of edge matching. After edge matching, the gaps are closed and the roads are connected.

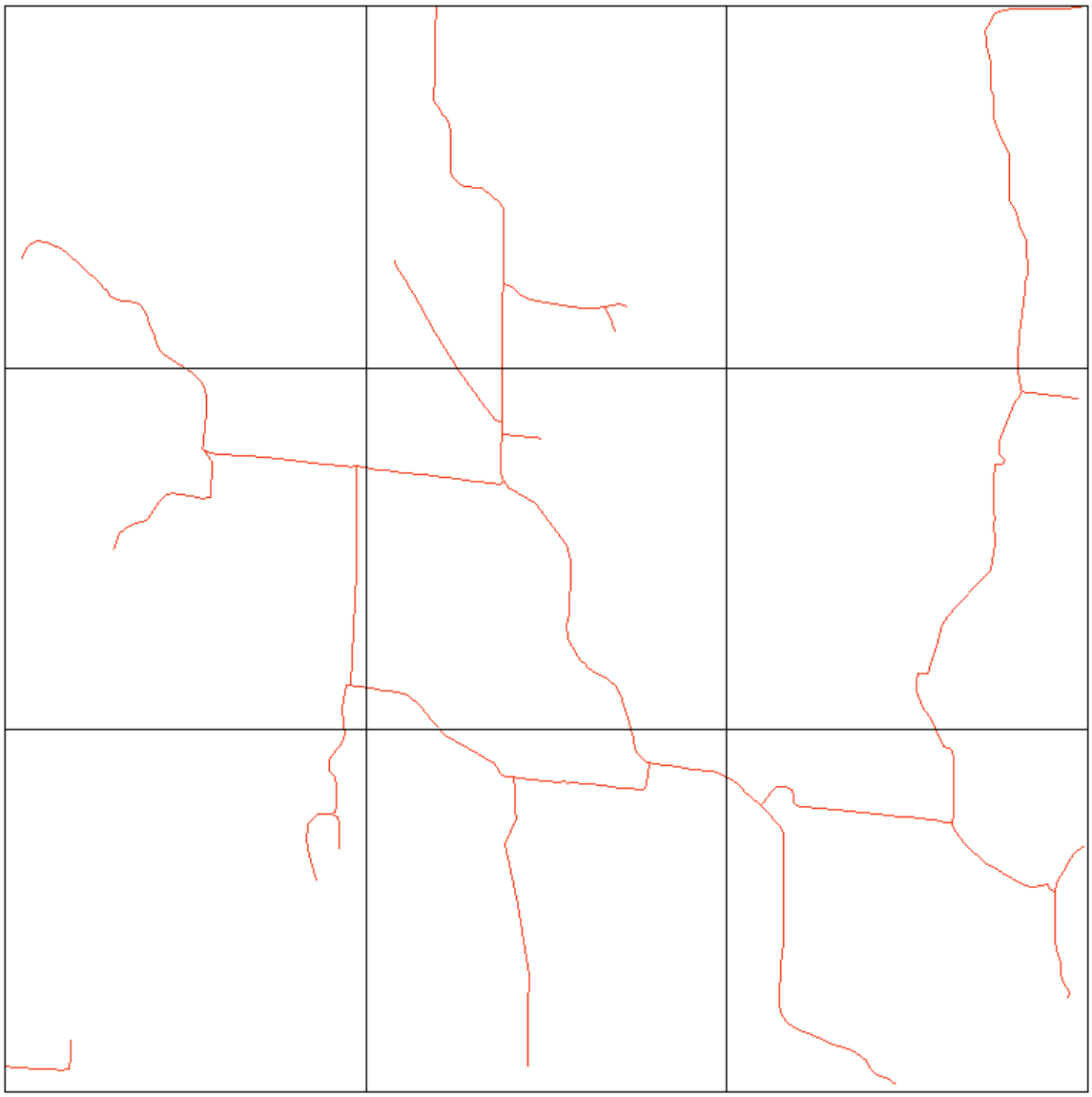


Figure 5.21 The final result in rural area after edge matching

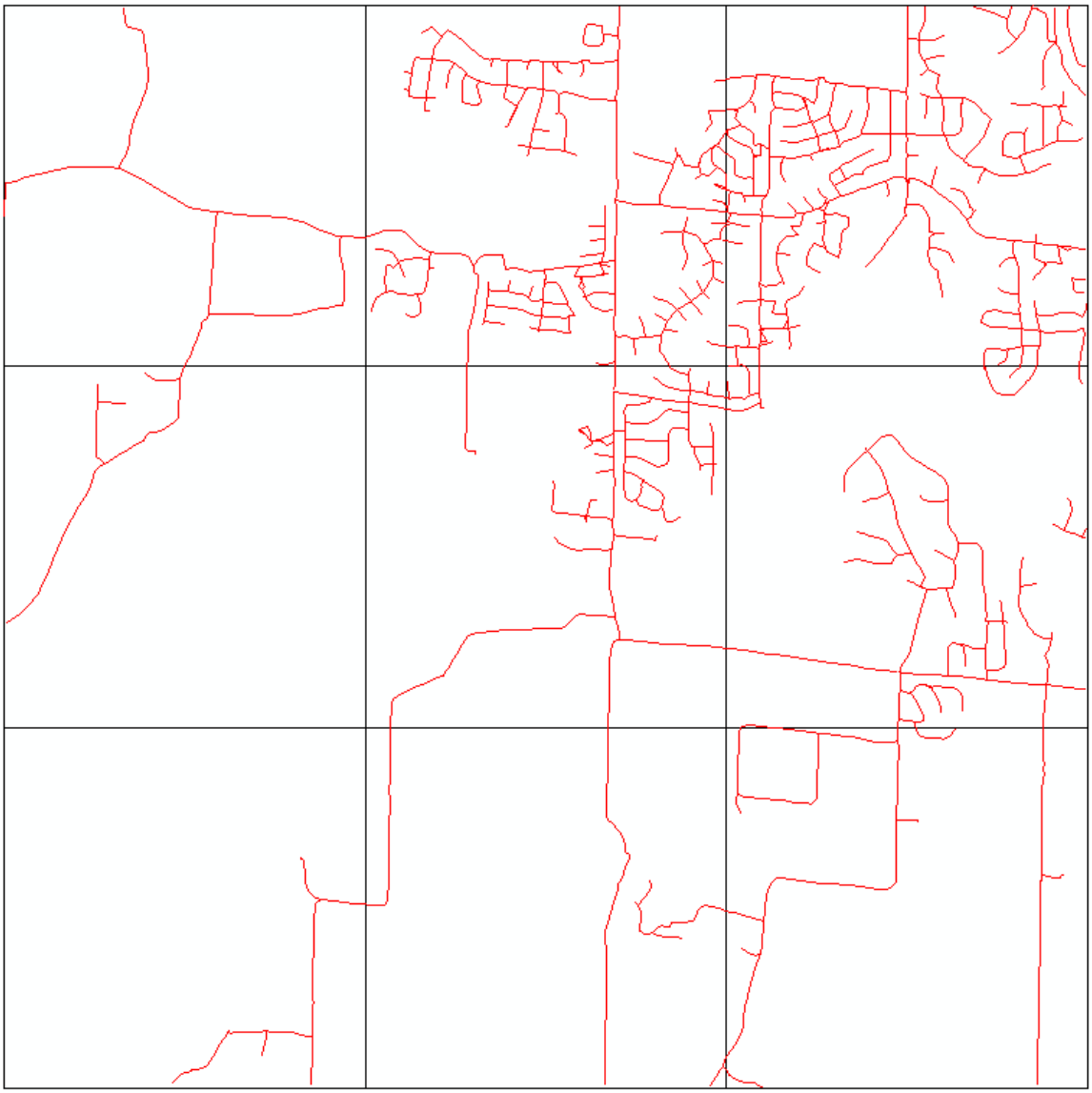


Figure 5.22 the final result in suburban area after edge matching

5.5 Performance Issues

Since we use the real world vector data and imagery to run our vector-to-imagery conflation algorithm, it is unrealistic to have 100% correct matching. The matched road intersections found by the relaxation algorithm do contains errors, especially for some

complex situations such as figure 5.23. The incorrectly matched road intersections caused the snake to snap the road to the wrong location.

Figure 5.23 some road pattern is very complex, the incorrectly matched road intersections cause the roads snapped to wrong location in imagery.

Similarly, suppose we add a new parallel road to test area Urban07. This might happen if the image chip and the vector road chip are significantly shift relative to one another. In this example, the two end points of one vector road are matched to two different roads in the image (figure 5.24). The incorrect initialization caused the snake convergence to be split across both roads (figure 5.25).

Figure 5.24 Original TIGER roads (red) in test area Urban07 are overlaid on the road mask with an additional road edited in to demonstrate a limitation of a fully automated approach. The green dashed lines indicate the matched road intersections found by relaxation labeling algorithm.

Figure 5.25 The final conflated TIGER roads (green) with vertices (red dot) are overlaid on road imagery.

Currently we do not correct the wrong matches on purpose to see how our automatic conflation algorithm works. If a perfect map is required, a human GIS analyst may need to examine and edit the road intersection matching result and final conflated roads to make sure the final map is 100% correct.

5.6 Additional Snake Experiment

It is supposed that both ends of a snake are in correct positions after piecewise transformation in our snake application. Therefore, we fix the end points and only move the intermediate vertices in previous snake algorithm. However, not all of the end points are in perfect places in the real world situation. To solve such problem, we modify and add a new snake function to relax the end points of a snake. In general, we run the normal snake function first to move the intermediate vertices, then run the relax version of snake to move the end points if needed.

To test the robustness of our snake algorithm, we run additional experiments using an interesting real world example (figure 5.26). We move a few points (end points or intermediate vertices) to different extremely bad locations to examine the movements of the snake. As shown in appendix B, even when the vector road vertices are far away from imagery, or both of the end points are not in correct locations, our snake algorithm still can move the vector roads to correct imagery location as long as a clean road image is provided.

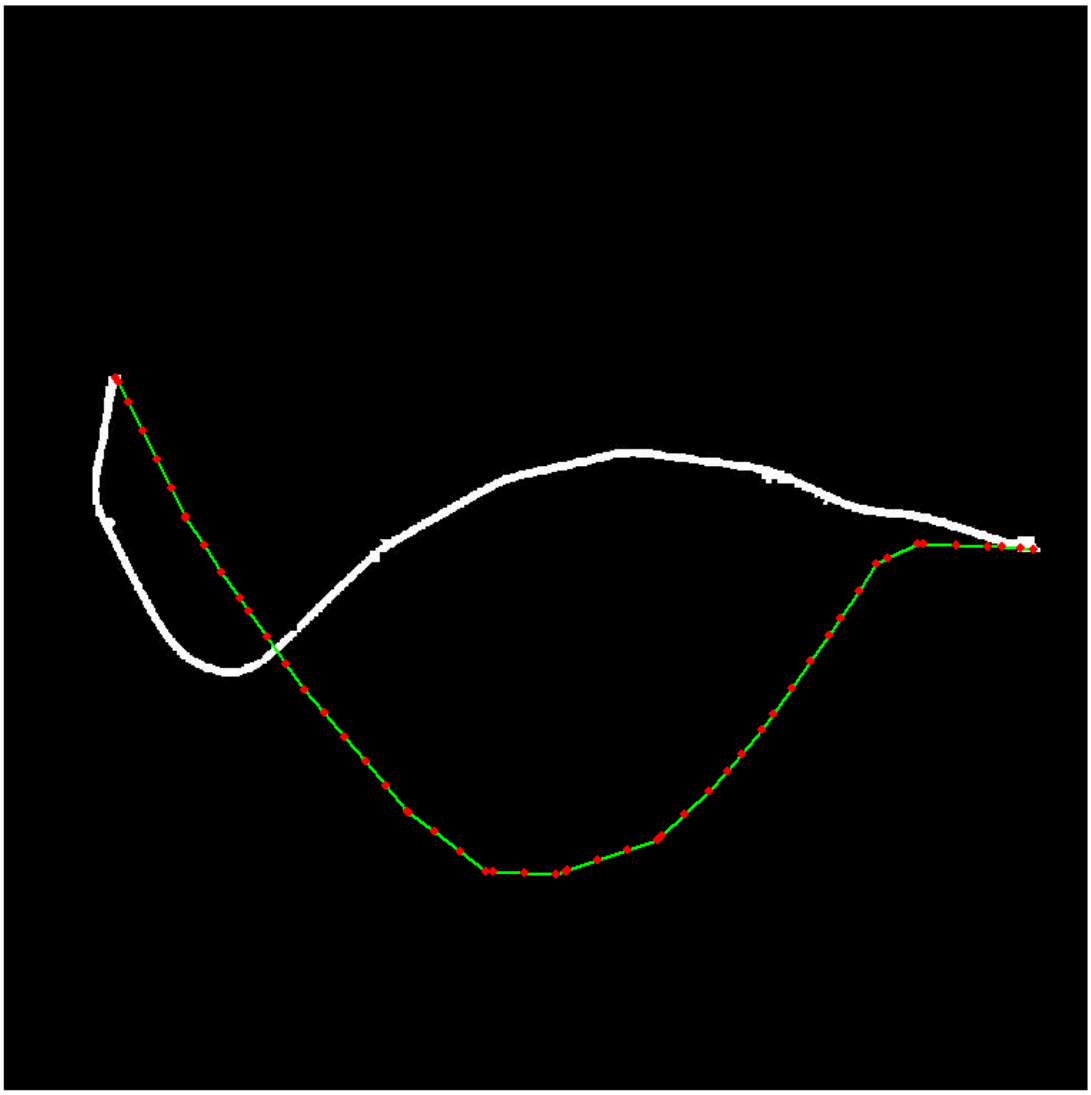


Figure 5.26 An interesting real world case. After point matching and rubber-sheeting transformation, the adjusted TIGER road (green) with vertices (red dot) is overlaid on road imagery. Note only the end points of road are in correct locations.

5.7 Experiment in downtown area

Although the experiment results in rural and suburban areas are excellent, it is interesting to see how our conflation algorithms perform in urban area. We run the same processes on the downtown area of Columbia, MO (figure 5.27). There is less vegetation, and the roads, buildings and parking lots have quite similar spectral characteristics. It is really difficult to distinguish roads from other regions in the classified image (figure 5.28) based on spectral information only.

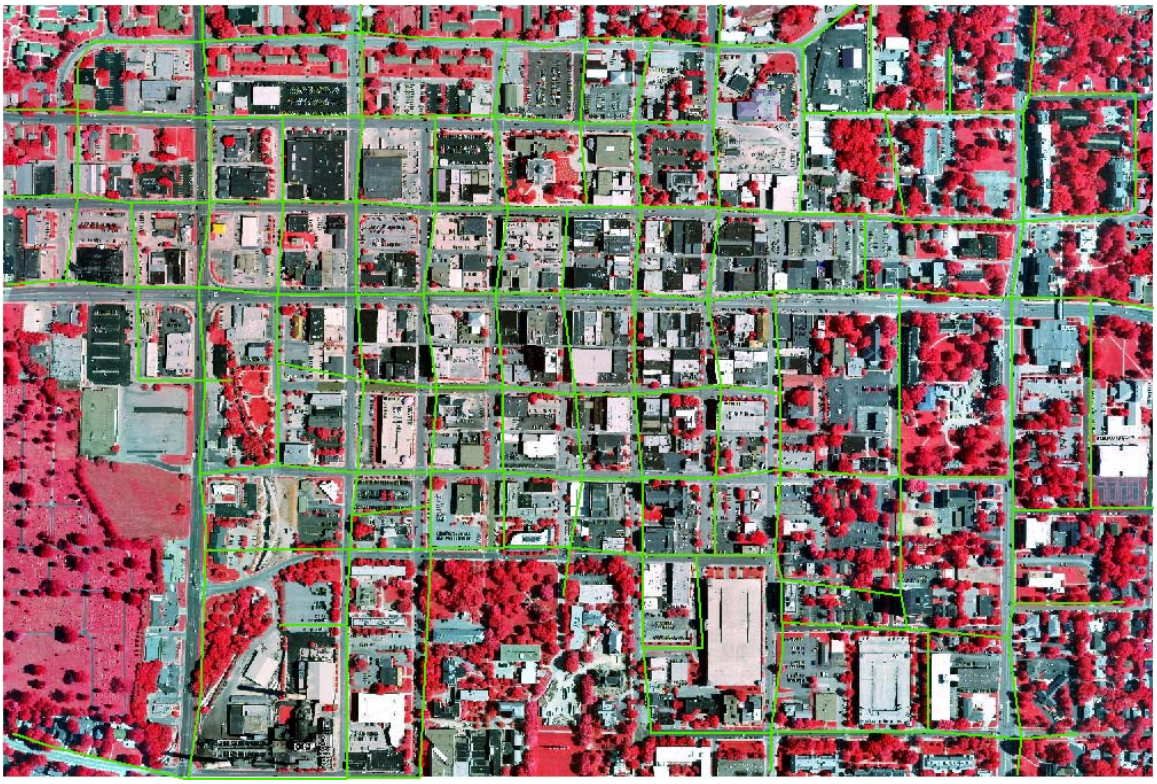


Figure 5.27 TIGER roads (green) are overlaid on multi-spectral aerial photo in the downtown area of Columbia, MO.

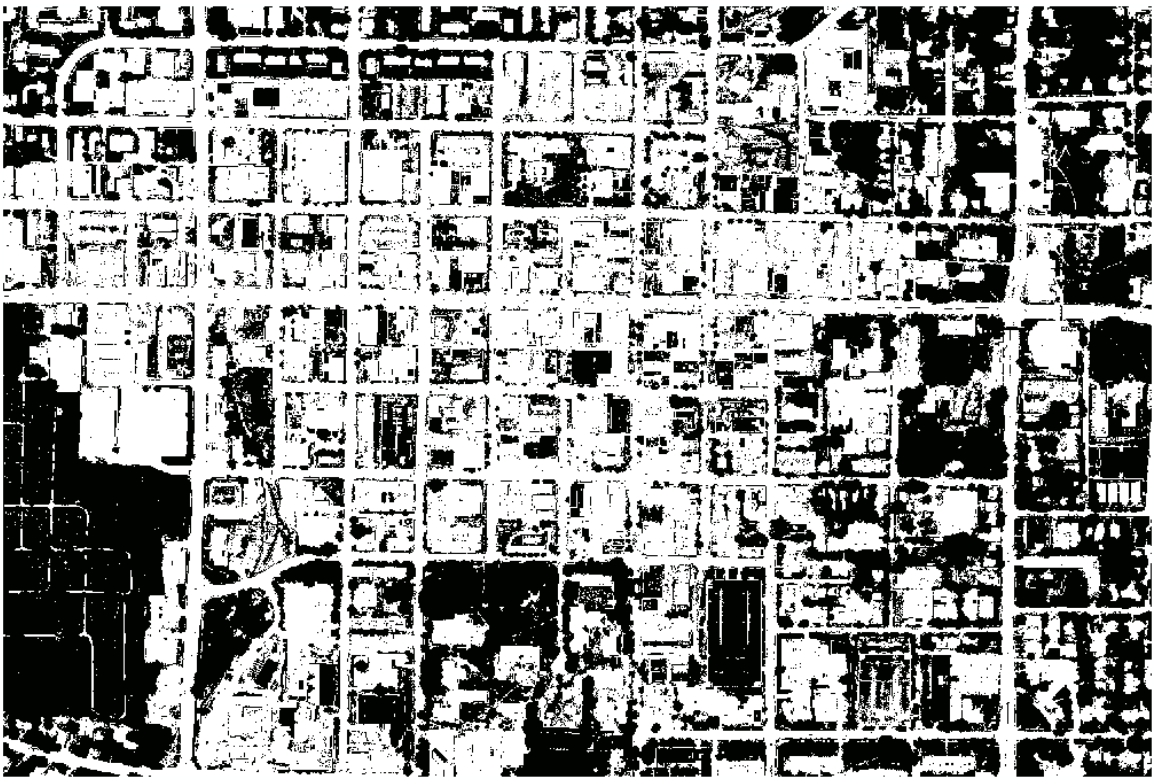


Figure 5.28 The classified road image in downtown area of Columbia, MO. The roads, buildings, and parking lots have similar spectral characteristics. It is difficult to separate them based on imagery information only.

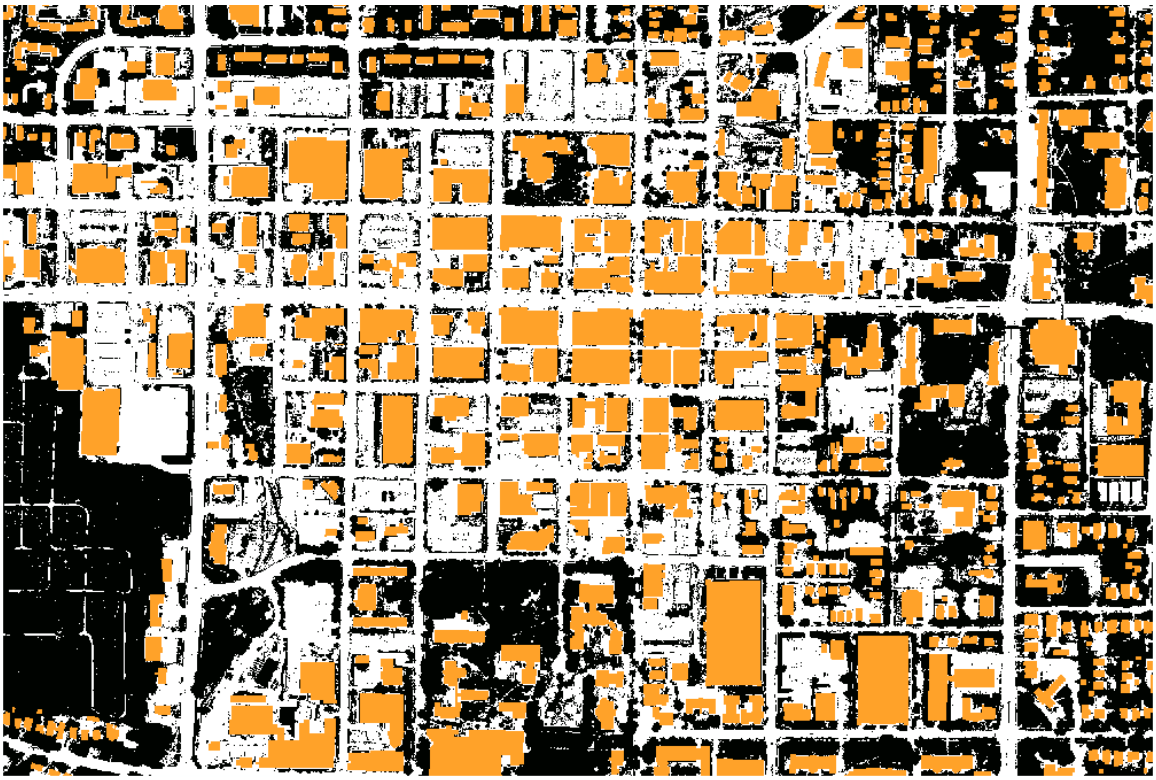


Figure 5.29 The vector building footprints (yellow polygons) are used as mask to clip out buildings from road image.

To solve this problem, we use existing vector building footprints (figure 5.29) obtained from city of Columbia to remove the building regions from the image, and then we run our algorithms as usual. Since most roads in the downtown area are connected with two intersections, no snake operation is needed. The potential road image is shown in figure 5.30 and the final conflated results are shown in figure 5.31. Again we conducted the acuucary assessment using MODOT roads as reference. We improved the correctness from original 78% in TIGER roads to 94% and reduced the RMSE from 14 m to 10 m. The result has been added into tables 5.1-5.3. Although there are a few errors

caused by unmatched/mismatched road intersections, our conflation algorithm still performed well in a dense downtown area.

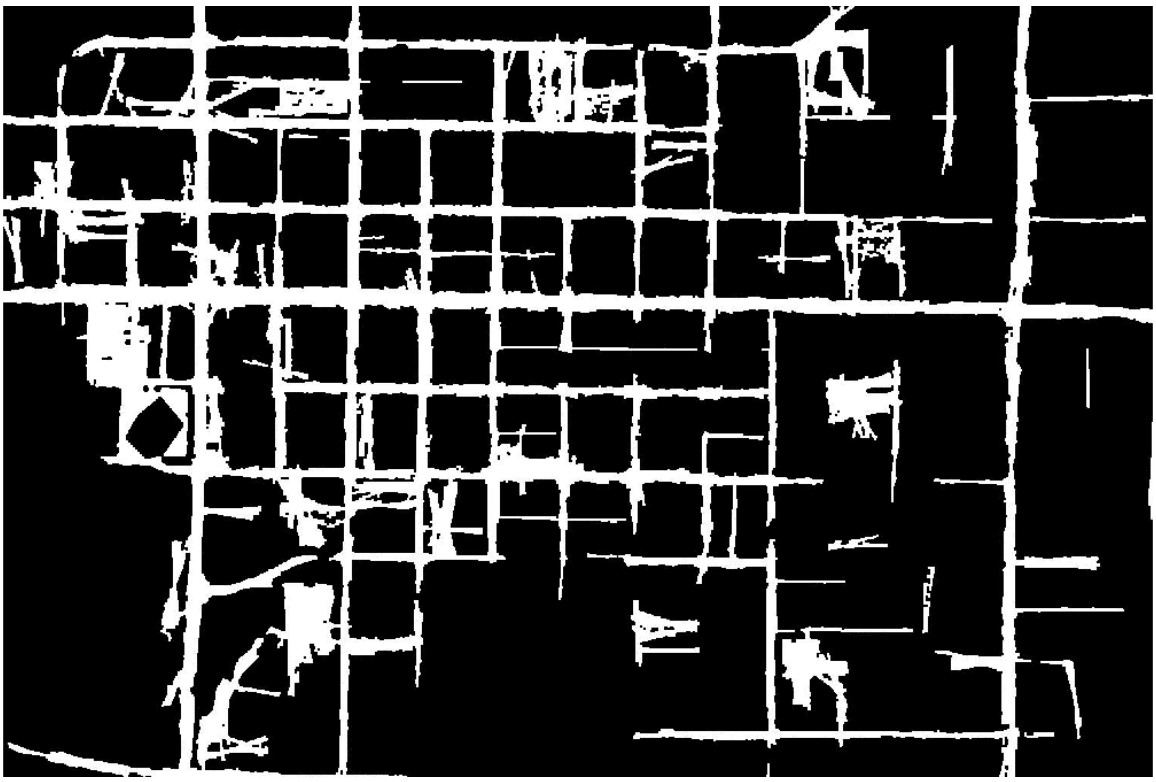


Figure 5.30 The potential road image generated by spatial contextual algorithm

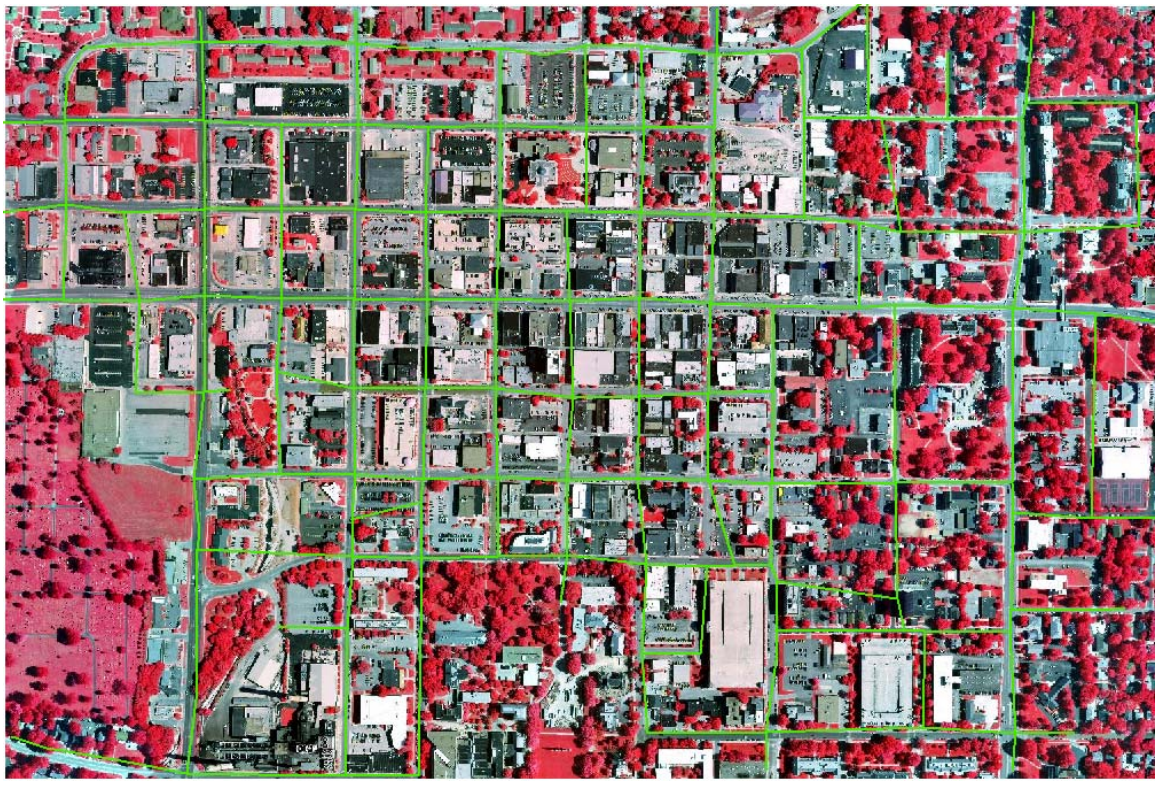


Figure 5.31 The final conflated TIGER roads (green) are overlaid on imagery.

Chapter 6

Improving the Positional Accuracy of Digital Parcel Map through Vector-to-Imagery Conflation

Local governments frequently use maps or spatial data for decision-making, but many of these data are often inaccurate and outdated which may have a negative influence on the expected outcome. Remote sensing can provide accurate and current data. However, one major bottleneck to the integration of remotely sensed imagery into existing geographic information systems is the issue of positional accuracy of the existing line-work within the vector database, making it difficult to match the imagery. This chapter presents a vector-to-imagery conflation approach to improve the positional accuracy of digital parcel map by moving the vector parcel map to make it consistent with high-resolution imagery. The road intersections are automatically extracted from imagery and used as control points. Our relaxation labeling algorithm is used to find the matches between road intersections from vector database and road intersections extracted from imagery. The links are created from the matched road intersection point pairs and

are used to perform the piecewise rubber sheeting transformation. The test results show that this approach can improve the accuracy of vector parcel map significantly. It's a very cost-effective method and has great potential to save considerable time and money for local governments to upgrade their inaccurate vector datasets.

6.1 Introduction

Parcel maps indicate the location, size and shape of each property, and are indispensable reference tools for tax assessment, zoning, utility construction, etc. Many city or town officials (assessors, planners, engineers, and others) use parcel maps on a daily basis [121].

Due to the rapid advances in information technology, many local governments have begun to use the geographic information systems (GIS) and millions of dollars have been invested to transfer paper parcel maps and other data into digital format. Unfortunately, many parcel maps lack positional accuracy. Unaware of accuracy requirements, local governments continue to digitize parcel maps which are often used as a base to compile other data such as zoning, utilities, etc. Therefore, while positionally inaccurate, the maps can keep the relative relationship between features.

However, when these vector data are overlaid to remotely sensed imagery with higher accuracy, the problem of positional accuracy is obvious. For example, many parcel lines cut through houses. The inaccurate vector data have to be adjusted, but traditional transformation methods are believed to be ineffective because the process of parcel mapping often pieces together data from multiple sources, each with their own distortion characteristics. Therefore, no one function has shown the ability to transform

the entire map space effectively [122]. One solution is extensive manual editing. This is labor intensive and time-consuming. Another possible solution is re-mapping the entire area. This also is too expensive for most agencies.

This chapter presents a conflation approach to improve the positional accuracy of existing digital vector parcel data. It utilizes the available remotely sensed imagery such as aerial photographs, or high-resolution satellite imagery. All processes are automatic, making it very cost-effective. The next section briefly reviews some parcel mapping methods and transformation techniques. The following section describes the vector-to-imagery conflation methodology. Finally the test results and conclusions are presented.

6.2 Parcel Mapping Methods and Transformation Techniques

6.2.1 Parcel Mapping Methods

There are several ways to create digital parcel maps: digitizing from analog maps, coordinate geometry (COGO) conversion, positioning property corners with GPS (global positioning system), etc [121].

COGO uses land surveys or deeds to produce information about each parcel in terms of geometric distances and angles from control points (benchmarks) that are converted to create a geometrically accurate parcel polygon. Unfortunately, not all parcels have property boundary descriptions and the information could be inaccurate for those that do. Hence, not all of the parcel polygons can be created. Field work is often required, which is time-consuming and costly.

With the rapid development of GPS technology, it has become relatively easy to get coordinates for the property corners. Digital parcel maps can be obtained by

connecting these corners. However, parcel corners may be invisible on the ground and this field method is labor intensive.

Digitizing paper maps is a basic method commonly used by many local governments and mapping companies. This can be done either using a digitizing board or scanning and then digitizing on a screen. This process is also labor intensive and time-consuming. If the map is in very good condition, automatic vectorization may be used. However most of the analog map inaccuracies will be very accurately reproduced on digital maps.

Due to the positional accuracy issue, one process often used during digitizing is called occupational adjustment [123]. There are two major approaches: shift-and-digitize & cut-and-paste. The shift-and-digitize method first registers a planimetric map on a digitizer table. The corresponding parcel map, at the same scale and on transparent medium, is moved around until one or more city blocks fit as well as possible inside the road-casing polygons in the planimetric map. Next, the parcel map is lightly taped to the table. The block or blocks of parcels are then digitized. When complete, the parcel map is shifted until the best concordance is reached for the next set of blocks. For the cut-and-paste method, a copy of a parcel map is cut into many sections, each of them including a number of city blocks. Each piece is then placed on a planimetric map and shifted and rotated until the best fit is achieved. Then, adjoining pieces are taped together. The whole ensemble is digitized or scanned into digital database. These approaches highlight the same issue was found in TIGER roads: no global transformation can be found to register large areas – local geometry needs to be exploited.

6.2.2 Transformation Techniques

Many transformation techniques have been used in GIS. The traditional techniques based on polynomials are the conformal transformation, the affine transformations, and high order polynomial transformation. The advanced techniques include least squares adjustment using indirect observations, least squares adjustment using indirect observations with geometric constraints, and piecewise transformation [122].

The traditional methods use one model with limited parameters for the entire space. Therefore, they are not suitable for parcel maps where distortions vary over space.

Adjustment with indirect observations attempts to maintain the topological consistency between all lines by distributing errors along all the lines. The observation equations represent each line length by the difference between the *x, y* coordinates of the endpoints. The least square adjustment is used to minimize residuals of the differences between line vertices.

Piecewise transformation based on Delaunay triangulation uses localized transformations whereby the vertices of the Delaunay triangles are the control point locations. Distinct transformation parameters are used on each triangle facet. The piecewise nature of this method is able to account for variations of error throughout the map space.

Lembo and Hopkins' research [122] showed that the advanced techniques are superior to the traditional approach in terms of improving the accuracy of parcel maps in a GIS. Piecewise transformation with adequate control points provides the best accuracy.

However, automatically acquiring adequate control points is still a critical problem and remains a challenging task for GIS research. Normally these control points are collected manually by human analysts from high accurate maps, GPS or imagery. Then the human analysts interactively identify and match the control point pairs on maps for performing the piecewise transformation. The control point collecting and matching processes may become quite labor intensive if the mapping area is large. Therefore, automated control point collecting and matching approaches need to be exploited. Our automated road intersection extraction and matching approach presents a promising solution.

6.3 Vector-Imagery Conflation Methodology

To conflate/migrate vector parcel map to high-resolution imagery, an adequate set of control point pairs is needed. The control points should be clearly visible and easily identifiable from both vector map and imagery. One such example is road intersections which are frequently used as control points to perform map/imagery registration and transformation.

In previous chapters, we already developed a set of conflation algorithms to automatically extract road intersections from high-resolution imagery, match the road intersection pairs from vector map and imagery, and conflate vector roads to imagery. As an added benefit, we can easily apply the same approach to parcel map conflation. Here we only need to add one process which is to generate vector road centerline from a parcel map if no road centerline layer is available. The rest of processes (road intersection

extraction from vector and imagery, point feature matching and rubber-sheeting adjustment) are exactly the same as what we described chapter 4.

6.3.1 Generate Vector Road Centerline from Parcel Map

If a GIS database does not have a road centerline layer, it can be automatically created from dual-line features such as road casings or parcel data [124]. From the original parcel map, we create a new block layer which only contains the parcel polygon outlines of each street block. Then we create the road centerline layer from the block layer. Creating single centerline from dual-line features is a typical generalization operation in GIS. Since the road centerlines are created from the parcel map, their relative relationship is consistent with parcel map. Therefore, later when we apply the same rubber-sheeting transformation to both road centerlines and parcel map, the results are still consistent.

6.3.2 Extract Road Intersections from Vector Road Centerline

Since the vector road centerlines are digital files, we can create the topology to describe the spatial relationship between features. Then one simple node-to-point operation can generate a new intersection point layer from the road centerlines. We use the topology information to assign one attribute to each intersection. The value is the number of roads connected at each road intersection. For example, the value is 3 for 3-way intersection and 4 for 4-way intersection.

6.3.3 Extract Road Intersections from Imagery

The same road intersection extraction algorithm described in section 4.2.2 is used. Again, the extracted intersections have one attribute: number of road arms connected at that point.

6.3.4 Point Feature Matching

Similarly, the same relaxation labeling-based point matching algorithm described in section 4.2.3 is used here. The relaxation labeling algorithm finds the matches between the two road intersection point sets, and the links are automatically created by using the coordinates of each corresponding matching point pair. Each link consists of one from-point (vector road intersection) and one to-point (intersection extracted from imagery).

6.3.5 Piecewise Transformation

Piecewise transformations have been developed and used by computer cartographers over the last 30 years [123]. They are characterized by the subdivision of the maps in pieces and by the use of different transformation parameters in each piece. The aim is to adjust maps showing deformations that do not follow a clear trend. Such deformations are easily found in parcel maps compiled piecemeal from non-contemporary survey documents of diverse origins and scales. Our vector-to-imagery conflation approach utilizes existing ArcGIS piecewise transformations tool.

This piecewise transformation tool requires a set of deformation vectors, also called links. These links show where coordinates are to be moved. In our experiment, the links are generated from matched road intersection pairs. Each link has two points, a

from-point and a to-point. The from-points are road intersections from vector map and to-points are road intersections extracted from imagery. The piecewise transformation uses a triangulation method to divide the entire area into local transformation pieces. Here we use the Delaunay triangulation which is a triangulation of the point set with the property that no point falls in the interior of the circumcircle of any triangle. The triangle vertices are represented by from-end of links. After the transformation, the matched road intersections are moved into exact alignment with imagery and the displacements of other road vertices are interpolated according to the movements of the vertices of the triangles which contain them.

We use the same piecewise transformation parameters to adjust road centerlines and parcel map. Therefore, the adjusted road centerlines are still consistent with adjusted parcel map.

6.4 Experiment Results

We tested the vector-to-imagery conflation approach using real world data. We obtained digital parcel maps from city of Springfield, Missouri and high resolution satellite IKONOS imagery (1 meter panchromatic band and 4 meter multispectral bands) from previous NASA research project. The geo-referenced IKONOS imagery has much better positional accuracy than do the parcel maps.

We selected one 1000 m by 700 m test area in Springfield. The parcel data is shown in figure 6.1 and IKONOS imagery is shown in figure 6.2. When the parcel map is overlaid on the IKONOS frame (figure 6.3), it is clear that it does not line up. As can be

observed, many parcel lines cut through houses, and some parcel lines even fall into another street block.

Using ArcGIS software package, we generate the new street block layer (figure 6.4) automatically from the parcel map (figure 6.1). Then we create the road centerlines (figure 6.5) from the street blocks. Therefore, the road centerlines are consistent with the parcel map.

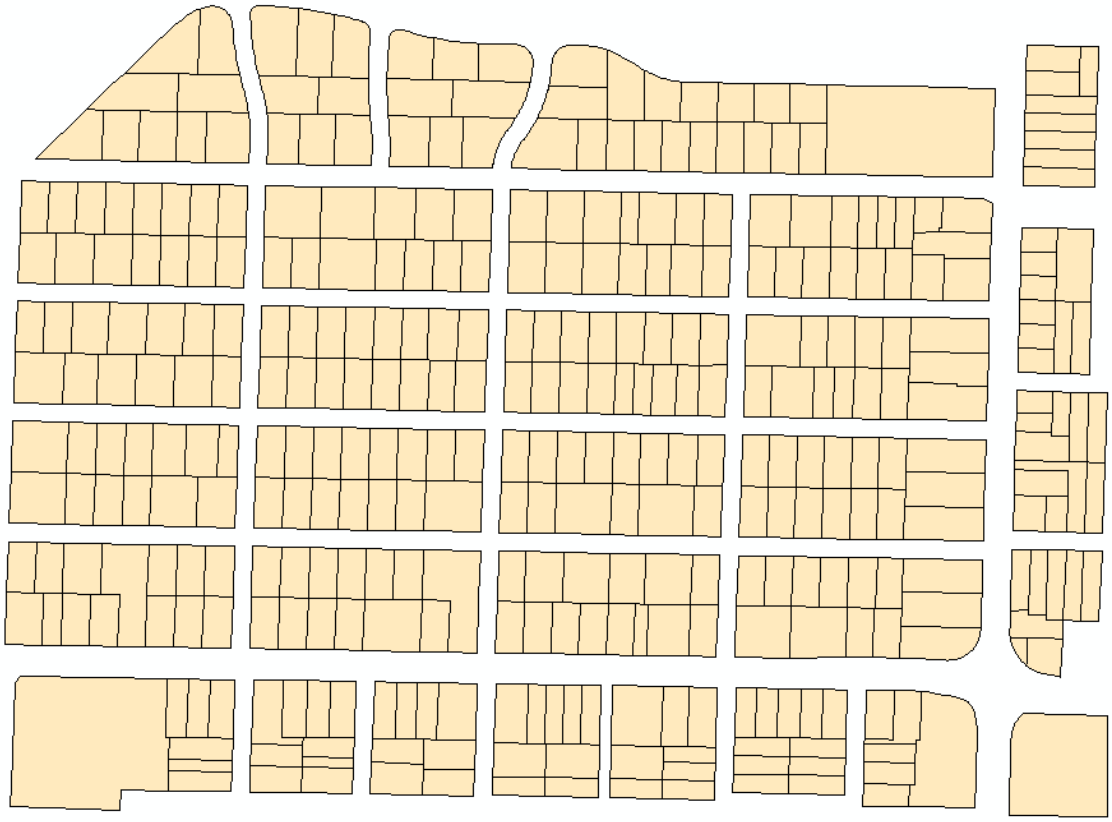


Figure 6.1 Digital parcel map of the test area in city of Springfield, MO

Figure 6.2 Multispectral IKONOS imagery of the test area in city of Springfield, MO

Figure 6.3 Original parcel lines (yellow) overlaid on the high-resolution IKONOS imagery. It is obvious that many parcel lines cut through houses.

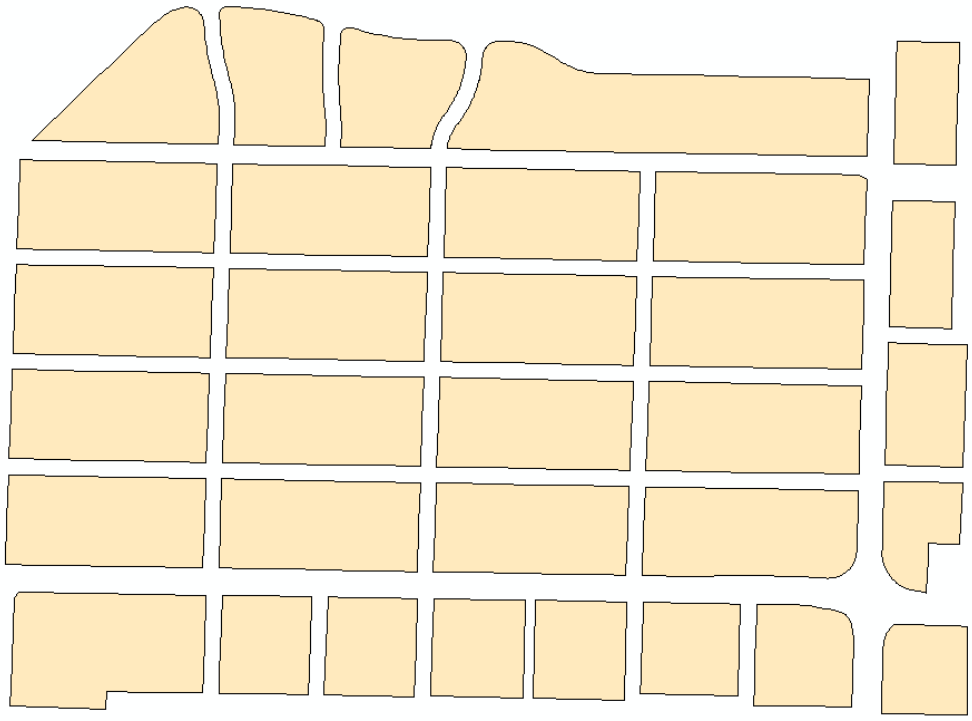


Figure 6.4 Street block layer generated from parcel map

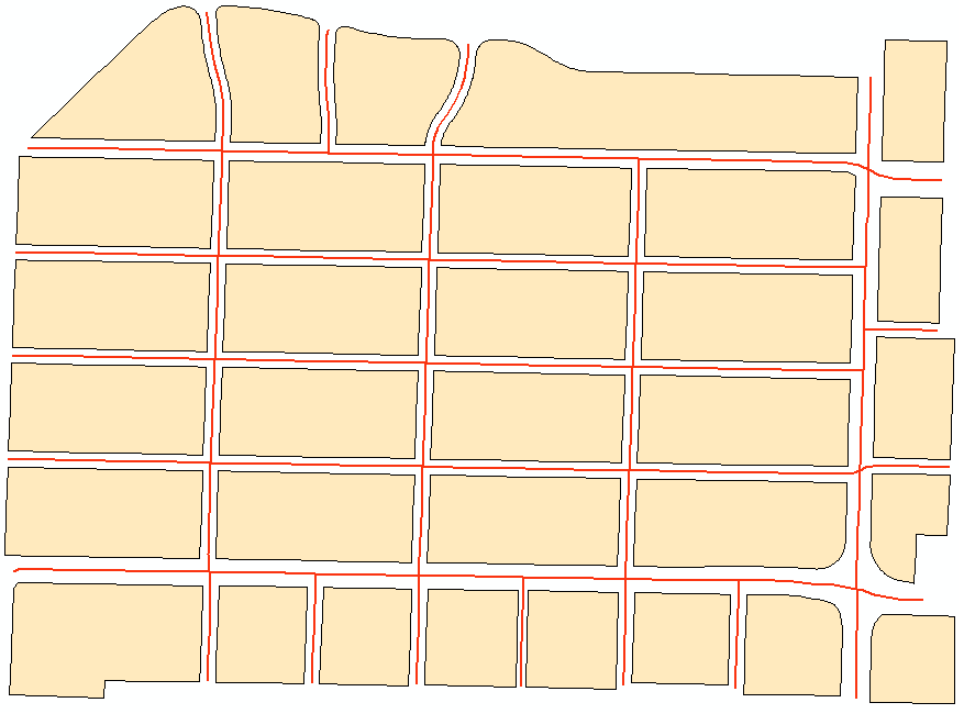


Figure 6.5 Road centerlines (red) generated from street blocks

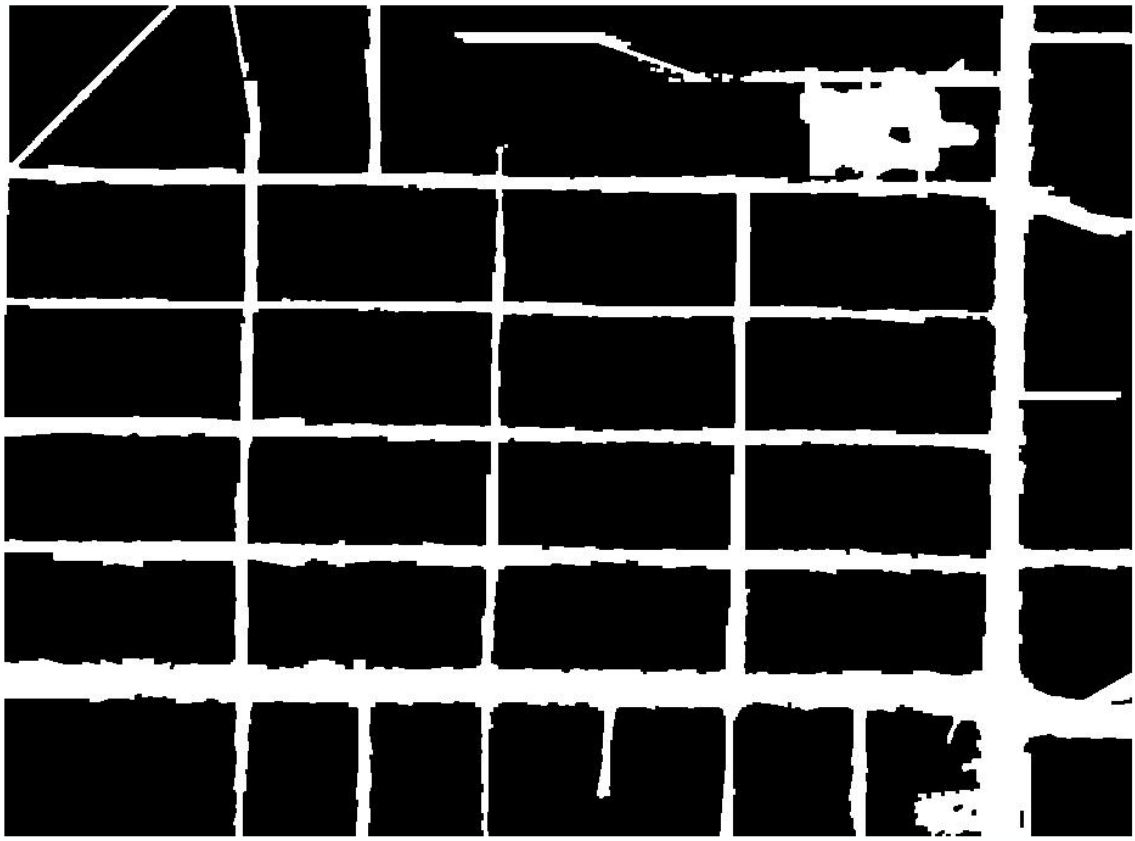


Figure 6.6 Road image created by spatial contextual signatures using the methods described in chapter 4.

Figure 6.7 after running vector-to-imagery conflation, the adjusted parcels (green) and road centerlines (red) are in line with imagery.

Figure 6.8 Zoom-in section of the parcel lines (green) and road centerlines (red) after vector-to-imagery conflation

We run the ISODATA unsupervised learning algorithm on IKONOS imagery and assign the clusters into two classes: road and others. Then the spatial contextual signatures are extracted to create the road image (figure 6.6). Next we automatically extract the road intersections from the road image. The relaxation labeling algorithm finds the matches between road intersections from vector road centerlines and intersections extracted from image. Finally the rubber-sheeting transformation adjusts the

parcel map and road centerlines using the links created by the matched pairs. The final results for entire test area are shown in figure 6.7 and figure 6.8 shows a zoomed-in section. The adjusted parcel polygons are within the correct block and line up with imagery very well. There is no clear problem of parcel lines transecting houses. The test results are visually satisfactory.

The same piecewise transformation can be used to conflate other vector layers that have already aligned with the original parcel layer. Therefore, the relative relationship between features still does not change after transformation.

6.5 Discussions and Conclusion

In our previous conflation work, we developed a set of automatic algorithms to extract road intersections from imagery, match the point pairs from vector and imagery, and conflate the road centerlines to high-resolution imagery. As an added benefit, the same conflation approach was extended to parcel map conflation in this chapter.

The experiment showed that the presented vector-to-imagery conflation methodology has been proved to be valuable for improving the positional accuracy of an existing digital parcel map. This approach is very cost-effective by using available remotely sensed imagery. This method can easily be performed and used by many local governments and mapping companies.

The key to success for the piecewise transformation is to have sufficient and well-distributed control points. Road intersections can be clearly identified from vector and imagery and are frequently used as control points. Normally human operators interactively collect and match the control points. We developed algorithms to

automatically extract and match these control points. Our automated approach will greatly improve the efficiency and can also be applied to other applications such as map registration etc.

Local governments often use the parcel map as a baseline with which many other data layers are aligned. Once the transformation parameters are determined, the same transformation can be used to adjust those layers as well. Therefore, all the adjusted data layers are still aligned well with each other.

Currently we visually evaluated the results of the transformation and did not conduct a statistical accuracy assessment due to lack of 'true' parcel data. If the high accuracy parcel data is available, further assessments can be conducted and the statistical results can be obtained to show how much this approach may improve the positional accuracy of digital vector parcel map.

Users must consider the quality of data used to produce a GIS dataset. The use of existing data, without careful review of the accuracy, may mean that the quality of the final GIS result is poor, out-of-date, or incomplete [125]. Remotely sensed imagery provides a very good media to upgrade and update digital vector data. The proposed vector-to-imagery conflation approach utilizes GIS and Remote Sensing technologies, and should be very suitable for improving the positional accuracy of digital parcel map.

Chapter 7

Conclusion and Future Work

7.1 Summary of Completed Research

The availability of geospatial data from multiple sources requires integration/fusion of multi-source datasets. However, accurately integrating geospatial data from different sources is a challenging task. In this dissertation research, we developed an innovative geospatial conflation approach consisting of several algorithms to attack the multi-source geospatial integration/conflation problem. Appropriate combinations of the algorithms are applied to vector-to-vector and vector-to-imagery conflation.

Conflation is real world problem in GIS field. To make our approach suitable to solve the real world problem, we obtained real world datasets such as TIGER roads from U.S. Census Bureau, MODOT roads from Missouri Department of Transportation, vector parcel maps from Greene county assessor's office, high-resolution aerial photographs from city of Columbia and multi-spectral IKONOS satellite imagery. All of our algorithms are tested on those real world datasets.

7.1.1 Vector-to-vector Conflation

In chapter 2, we proposed a hybrid approach based on the combination of the traditional conflation and a snake algorithm. The feature matching and rubber-sheeting map alignment processes remain the same as traditional conflation and the attribute transfer process is replaced by snake alignments. The snake algorithm moves the TIGER roads towards the high accuracy roads. Our approach has benefits over traditional conflation methodology in that it overcomes the problem of splitting vector road line segments and can be extended for vector-imagery conflation as well.

Traditional conflation requires at least two existing vector datasets. In many cases a second vector GIS dataset is not available. Acquiring and collecting such vector GIS data is time-consuming and cost-prohibitive. On the other hand, large amounts of high-resolution satellite imagery and aerial photographs are available today, and many of these are free or have minimal cost to the public. Many agencies may have already acquired some high resolution images for other purposes. Making good use of widely available free imagery is cost effective for conflation. Our snake-based conflation can make use of both vector and imagery data to conduct conflation. Therefore, it has much broader applications for integration of GIS, GPS and remote sensing data.

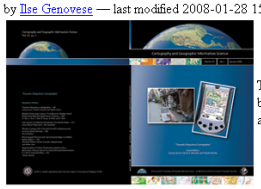
Our unique contribution is to present a novel snake-based approach which is a brand new idea for vector-to-vector conflation, with this same approach applied to both vector-to-vector and vector-imagery conflation.

This work was published and featured as cover page in ***Cartography and Geographic Information Science***. Our paper is of interest to a broad section of readers in

GIS field. In fact, it made the list of top contributions to CaGIS as one of the most downloaded papers over a certain time period.

Top contributions to CaGIS

by Ilse Genovese — last modified 2008-01-28 15:06



The top 10 articles downloaded from http://www.ingentaconnect.com/content/acsm/cagis/
between December 2005 and November 2007
are, in order of full-text downloads:

1. *Visualizing Geospatial Information Uncertainty: What We Know and What We Need to Know*. By Alan M. MacEachren, Anthony Robinson, Susan Hopper, Steven Gardner, Robert Murray, Mark Gahegan, and Elizabeth Hetzler. 2005. *CaGIS* 32(3): 139-160.
2. *Geovisualization and GIScience*. By Menno-Jan Kraak and Alan M. MacEachren. 2005. *CaGIS* 32(2): 67-68.
3. *Automated Cartography in a Bush of Ghosts*. By William A. Mackaness. 2006. *CaGIS* 33(4): 245-256.
4. *Visualization of Qualitative Locations in Geographic Information Systems*. By Xiaobai Yao and Bin Jiang. 2005. *CaGIS* 32(4): 219-229.
5. *Evaluating the Usability of a Tool for Visualizing the Uncertainty of the Future Global Water Balance*. By Terry A. Slocum, Daniel C. Cliburn, Johannes J. Feddema, and James R. Miller. 2003. CaGIS 30(4): 299-317.
6. *Mobile Mapping and Geographic Information Systems*. By Keith C. Clarke. 2004. *CaGIS* 31(3): 131-136.
7. *Designing the Next Generation of Distributed, Geocollaborative Tools*. By Wendy A. Schafer, Craig H. Ganoe, Lu Xiao, Gabriel Coch, and John M. Carroll. 2005. *CaGIS* 32(2): 81-100.
8. *ColorBrewer in Print: A Catalog of Color Schemes for Maps*. By Cynthia A. Brewer, Geoggrey W. Hatchard, and Mark A. Harrower. 2003. *CaGIS* 30(1): 5-32.
9. *A Prototype Temporal GIS for Multiple Spatio-Temporal Representations*. By Yanfen Le. 2005. *CaGIS* 32(4). 315-329.
10. *A GIS Analysis of the Relationship between Criminal Offenses and Parks in Kansas City, Kansas*. By Nicole DeMotto and Caroline P. Davies. 2006. *CaGIS 33*(2): 141-157.

Articles most downloaded in November 2007, after an addition of part of the *CaGIS* backfile:

1. *Rethinking Levels of Measurement for Cartography*. By Nicholas R. Chrisman. 1998. *CaGIS* 25(4): 231-242.
2. *Cognitive Map-Design Research in the Twentieth Century: Theoretical and Empirical Approaches*. By Daniel R. Montello. 2002. *CaGIS* 29(3): 283-304.
3. *Dasymetric Mapping and Areal Interpolation: Implementation and Evaluation*. By Cory L. Eicher and Cynthia A. Brewer. 2001. *CaGIS* 28(2): 125-138.
4. *Perception of Spatial Dispersion in Point Distributions*. By Yukio Sadahiro. 2000. *CaGIS* 27(1): 51-64.
5. *ColorBrewer in Print: A Catalog of Color Schemes for Maps*. By Cynthia A. Brewer, Geoffrey W. Hatchard, and Mark A. Harrower. 2003. *CaGIS* 30(1): 5-32.
6. *Toward a Participatory GIS: Evaluation Case Studies of Participatory Rural Appraisal and GIS in the Developing World*. By Brian H. King. 2002. *CaGIS* 29(1): 43-52.
7. *Choosing Geographic Units for Choropleth Rate Maps, with an Emphasis on Public Health Applications*. By Francis P. Boscoe and Linda W. Pickle. 2003. *CaGIS* 30(3): 237-248.
8. *Individual Differences in Map Reading Spatial Abilities Using Perceptual Memory Processes*. By Robert earl Lloyd and Rick L. Bunch. 2005. *CaGIS* 32(1): 33-46.
9. *A Snake-based Approach for TIGER Road Data Conflation*. By Wembo Song, Timothy L. Haithcoat, and James M. Keller. 2006. *CaGIS* 33(4): 287-298.
10. *Daymentric Estimation of Population Density and Areal Interpolation of Census Data*. By James B. Holt, C.P. Lo, and Thomas W. Hodler. 2004. *CaGIS* 31(4): 103-121.

Song, W., Haithcoat, T., Keller, J., "A Snake-based Approach for TIGER Road Data Conflation", ***Cartography and Geographic Information Science***, Vol. 33, No. 4, 2006, pp. 287-298.

7.1.2 Point Feature Matching by Relaxation Labeling

Feature matching is the most crucial element of conflation. The quality of feature matching determines the success of conflation. In chapter 3 we presented a new relaxation labeling-based point feature matching approach to match the road intersections from two GIS vector road datasets. The relaxation labeling algorithm utilizes iterated

local context updates to achieve a globally consistent result. The contextual constraints (relative distances between points) are incorporated into the compatibility function that is employed in each iteration's updates. The point-to-point matching confidence matrix is initialized using road connectivity information at each point. Both the traditional proximity-based approach and our relaxation-based point matching approach are implemented and experiments are conducted over 18 test sites in rural and suburban areas of Columbia, MO. The test results show that our relaxation labeling approach is more robust and has much better performance over the proximity matching approach in both simple and complex situations.

Once the features are correctly matched, we can either apply the traditional conflation approach to transfer the attributes from one dataset to another, or use our snake-based conflation approach presented in chapter 4 to move the road networks from one dataset to correct locations in another vector dataset or imagery. Therefore, the final result has good spatial accuracy and rich attributes, which is better than both of the original datasets.

Our contribution is to propose a new relaxation labeling-based point matching approach for vector map conflation. We believe that we are the first to present this approach to the conflation field. Our approach provides an elegant and well-motivated solution to the problem. This new point matching algorithm produces much better results than previous techniques and greatly improved the accuracy of point matching so that the manual workload can be significantly reduced.

The work is accepted and will be published in ***Transactions in GIS.***

Song, W., Keller, J., Haithcoat, T., Davis, C., “Relaxation-Based Point Feature Matching for Vector Map Conflation”, in Press ***Transactions in GIS***

7.1.3 Vector-to-imagery Conflation

Since imagery is acquired more frequently and is more current, conflating vector data to imagery has higher demands. An innovative vector-to-imagery conflation approached was presented in chapter 4 by integrating several vector-based and image-based algorithms. We only extract the different types of road intersections and terminations from the image frames based on spatial contextual measures. We eliminate the process of line segment detection which is often troublesome. The vector road intersections are matched to these detected points by a relaxation labeling algorithm. The matched point pairs are then used as control points to perform a piecewise rubber-sheeting transformation. A modified snake algorithm maneuvers vector road vertices toward a candidate road image. Finally a refinement algorithm moves the points to center each road and obtain better cartographic quality. Experiments were conducted over a variety of rural, suburban, and urban environments. The results demonstrated excellent performance. The average correctness measure increased from 20.6% in original TIGER road to 95.5% in the conflated results and the average root-mean-square error decreased from original 51.2 meters to 3.4 meters in our final results.

There are several related vector-to-imagery papers. Some require the vector roads to be very close to the corresponding roads in an image, and other methods stop at rubber-sheeting without further refinement. We believe our work is the most comprehensive vector-to-imagery conflation research. We use the TIGER roads with

very bad positional accuracy, where some road segment may be over 100 meters away from the truth. We applied spatial contextual signature extraction algorithm [111] to generate the potential road image. This road image is much cleaner and facilitates the use of a snake algorithm. We made several significant improvements to the snake operations in this research. We developed a new road intersection extraction algorithm by using the spatial contextual information. It is more reliable than other approaches. As one reviewer said, "*The authors deserve a compliment for producing a nice work in this paper."* This vector-to-imagery conflation research was published in ***IEEE transactions on Image processing***.

Song, W., Keller, J., Haithcoat, T., Davis, C., "Automated Geospatial Conflation of Vector Road Maps to High Resolution Imagery", ***IEEE Transactions on Image Processing***, Vol. 18, No. 2, 2009, pp. 388-400.

7.1.4 Vector-to-imagery Scale-up Experiment

In chapter 4, we tested the vector-to-imagery conflation approach on selected small areas. Those selected areas are ideal real world data where vector roads have one-to-one matches and roads are relatively easy to see in general.

To make our approach applicable in more general situations, we made several improvements and tested it on big rural, suburban, and downtown areas where the situation may not be ideal for conflation. Originally, we ran spatial contextual signature extraction and road intersection extraction algorithm on the raw image. This is time-consuming for large image data. In chapter 5, we first classified the image into road and

non-road classes, and ran the spatial contextual signature extraction algorithm on road region only. Then we ran the road intersection extraction algorithm on the thresholded road image. This greatly reduced the computation time. It also overcame the problem at locations where roads with significant different spectral characteristics meet together.

In our previous snake application, we supposed that both ends of a snake are in correct positions. We fixed the end points and only move the intermediate vertices. We added a new function to relax the end points after several normal snake movements. As shown in appendix B, even when the vector road vertices are far away from imagery, or both of the end points are not in correct locations, our snake algorithm still can move the vector roads to correct imagery location as long as a clean road image is provided.

7.1.5 Parcel Map Migration through Vector-to-imagery Conflation

In previous chapters, we used the road centerlines for conflation. We extended our approach to solve parcel migration problem in chapter 6. We developed a vector-to-imagery migration approach to improve the positional accuracy of a digital parcel map by moving it to make it consistent with high-resolution imagery. The road intersections are automatically extracted from imagery and used as control points. A relaxation labeling algorithm is used to find the matches between road intersections from the vector database and the road intersections extracted from imagery. The links are created from the matched road intersection point pairs and are used to perform a piecewise rubber sheeting transformation. The test results show that this approach can improve the accuracy of vector parcel map significantly. It is a very cost-effective method and has great potential

to save considerable time and money for local governments to upgrade their inaccurate vector datasets.

In addition, we conducted another conflation project to integrate vector buildings with imagery. The building outlines are extracted from high-resolution imagery via a shape-driven level set scheme. Shape and relative position features are then computed for the image-extracted buildings and for vector buildings from GIS database. These two features are used by a graph-matching procedure that finds the correspondences between the two sets of buildings. This work was published in ***IEEE Geoscience and Remote Sensing Letters***.

Sledge, I., Keller, J., Song, W. and Davis, C. "Conflation of Vector Buildings with Imagery", ***IEEE Geoscience and Remote Sensing Letters***, Vol. 8, No. 1, 2011, pp. 83-87.

7.2 Future Work

After reviewing the whole conflation process (image classification, spatial contextual signature exaction, potential road image creation, road intersection extraction, matching by relaxation labeling, rubber-sheeting transformation, and snake-based alignment), we think that creating a very good road image is the most important step. The following road intersection extraction and snake movement are all dependent upon the road image.

In the scale-up experiment, we only use the standard ISODATA unsupervised learning algorithm to generate clusters, and assign the clusters to road and non-road classes. Some specialized classifiers can be explored to classify the object-specific

geographic features. For example, machine learning algorithms can exploit the attributes of size, shape, color, texture, pattern, shadow, and spatial association to efficiently extract user-defined features such as roads, buildings etc. Currently we utilize the spectral information and spatial contextual separately in two sequential steps. We could explore ways to integrate both spectral and spatial information in a road-specific classification. This should create better classification results, and hence, better overall conflation results.

Having a clean road image is critical to the success of our conflation approach. We will further study the image clutter (noise) removing techniques. All kinds of geometric attributes such as size, length, and shape could be explored to remove non-linear clutter. Smoothing operators can be developed to remove the small spurs on the linear objects.

For the relaxation labeling algorithm, we just simply implemented Rosenfeld's original point matching algorithm. The computational complexity is O (n^4). Even for a very big dataset, the number 'n' in relaxation loop is a relatively small constant since we limit the computation in a small neighborhood. The constant is determined by the radius of the neighborhood and the density of points. The true computational complexity is O (n^2) since we need to pre-compute the distance between each point pair. Since both input datasets are already georeferenced in the same coordinate system, we could create a spatial index for the data. Therefore we limit the distance computation to a small neighborhood also and improve the computation efficacy.

Our conflation approach did not extract road centerlines from imagery although we did generate a potential road image. Since the imagery is more current than vector data, some road extraction and update work is needed in newly developed areas. The

unmatched information between vector road intersections and imagery intersections could provide useful hints to identify the areas where new roads can be found. Further clean-up and vectorization of the potential road image in identified areas will create new vector road centerlines.

7.3 List of Publications

Currently there are four different journal articles have been published. They are listed below. In the future, we plan to write two more papers to address the scale-up issue and parcel map conflation.

Song, W., Haithcoat, T., Keller, J., "A Snake-based Approach for TIGER Road Data Conflation", ***Cartography and Geographic Information Science***, Vol. 33, No. 4, 2006, pp. 287-298.

Song, W., Keller, J., Haithcoat, T., Davis, C., "Automated Geospatial Conflation of Vector Road Maps to High Resolution Imagery", ***IEEE Transactions on Image Processing***, Vol. 18, No. 2, 2009, pp. 388-400.

Song, W., Keller, J., Haithcoat, T., Davis, C., "Relaxation-Based Point Feature Matching for Vector Map Conflation", in Press ***Transactions in GIS***

Sledge, I., Keller, J., Song, W. and Davis, C. "Conflation of Vector Buildings with Imagery", ***IEEE Geoscience and Remote Sensing Letters***, Vol. 8, No. 1, 2011, pp. 83-87.

Reference List

[1] A. J. Kimerling, A. R. Buckley, P. C. Muehrchke, and J. O. Muehrcke, *Map Use: Reading and Analysis*. 6th ed., Redlands, CA: ESRI Press, 2009.

[2] K. C. Clark, *Getting started with Geographic Information Systems*, 4th ed., Upper Saddle River, NJ: Pearson Education, 2003.

[3] Wikipedia.org. Topographic Map. [Online]. Available: http://en.wikipedia.org/wiki/Topographic_map

[4] Canada Centre for Topographic Information, Topo Maps: Frequently Asked Questions. [Online]. Available: http://maps.nrcan.gc.ca/topo101/faq_e.php

[5] USGS, Topographic Mapping. [Online]. Available: http://egsc.usgs.gov/isb/pubs/booklets/topo/topo.html

[6] C. McGlone, *Manual of Photogrammetry*, 4th ed., Falls Church, VA: ASPRS, 1980.

[7] Advameg. Inc., Topographic Map. [Online]. Available: http://www.madehow.com/Volume-4/Topographic-Map.html.

[8] L. Moore, The U.S. Geological Survey's Revision Program for 7.5-Minute Topographic Maps. [Online]. Available: http://thor-f5.er.usgs.gov/topomaps/revision_overview.pdf

[9] H. Kennedy, *Dictionary of GIS terminology*, Redlands, CA: ESRI Press, 2001.

[10] D. Dibiase, Integrating Geographic Data. [Online]. Available: https://www.e-education.psu.edu/natureofgeoinfo/book/export/html/1604

[11] USGS, Landsat: A global land-observing Program. [Online]. Available: http://egsc.usgs.gov/isb/pubs/factsheets/fs02303.html

[12] USGS, Standards for Digital Orthophotos, Part1: General. [Online]. Available: http://rockyweb.cr.usgs.gov/nmpstds/acrodocs/doq/1DOQ1296.PDF

[13] G. Gabbott, The National Agriculture Imagery Program. [Online]. Available: http://msdis.missouri.edu/data/naip/naip_faq_2003.pdf

[14] USDA, National Agriculture Imagery Program (NAIP) Information Sheet. [Online].
Available: http://www.fsa.usda.gov/Internet/FSA_File/naip_2009_info_final.pdf

[15] National Technology Alliance, "Accelerating Conflation Capability for the US Government," TR-001-120804-128, Jan. 2005.

[16] U.S. Army, U.S. Army Geospatial Data Integrated Master Plan. [Online]. Available: http://www.ms.army.mil/current/AGDIMP%20Fact%20Sheet.pdf

[17] M. R. Newell, "The national map product and services directory," U.S. Geological Survey Fact sheet 2008-3065, 2008.

[18] E. L. Usery, M. P. Finn, and M. Starbuck, Integrating data layers to support the national map of the United States. [Online]. Available: http://carto-research.er.usgs.gov/data_integration/pdf/data-integration-icc.pdf

[19] K. Grady and L. Godwin, The positional accuracy of MAF/TIGER. [Online]. Available: http://www.census.gov/geo/mod/positional_accuracy.pdf

[20] R. W. Marx, A vision for a 21st century MAF/TIGER. [Online]. Available: http://www.census.gov/geo/mod/vision.pdf

[21] FEMA, "Challenges in FEMA's Flood Map Modernization Program," OIG-05-44, Sep. 2005.

[22] B. Weaver, "Implementation of the National Map Road Database," in *Proc. American Congress on Surveying and Mapping*, 2004.

[23] J. R. Jensen, A. Saalfeld, F. Broome, D. Cowen, K. Price, D. Ramsey, and L. Lapine, "Spatial data acquisition and integration," in *A Research Agenda for Geographic Information Science*, R. McMaster and E. L. Usery, Ed. Boca Raton: CRC Press, 2004.

[24] F. R. Broome and L. S. Godwin, "Partnering for the people: improving the U.S. Census Bureau's MAF/TIGER Database," *Photogrammetric Engineering & Remote Sensing*, vol. 69, no. 10, pp. 1119-1123, Oct. 2003.

[25] A. Saalfeld, "Conflation: automated map compilation," *International Journal of Geographic Information Systems*, vol. 2, no. 3, pp. 217-228, July-Sep. 1988.

[26] J. Sperling, and S. Sharp, "A prototype cooperative effort to enhance TIGER," *URISA Journal*, vol. 41, no. 9, pp. 35-42, summer 1999.

[27] D. Xiong and J. Sperling, "Semiautomatic matching for network database integration," *ISPRS Journal of Photogrammetry & Remote Sensing*, vol. 59, no. 1-2, pp. 35-46, Aug. 2004.

[28] M. A. Cobb, M. J. Chung, H. Foley, F. E. Petry, and K. B. Shaw, "A rule-based approach for the conflation of attributed vector data," *Geoinformatica*, vol. 2, no. 1, pp. 7-35, Mar. 1998.

[29] V. Walter and D. Fritsch, "Matching spatial data sets: a statistical approach," *International Journal of Geographic Information Science*, vol. 13, no. 5, pp. 445-473, July 1999.

[30] A. Samal, S. Seth, and K. Cueto, "A feature-based approach to conflation of geospatial sources," *International Journal of Geographic Information Science*, vol. 18, no. 5, pp. 459-489, July-Aug. 2004.

[31] S. Rahimi, M. Cobb, D. Ali, M. Paprzycki, and F. E. Petry, "A knowledge-based multi-agent system for geospatial data conflation," *Journal of Geographic Information and Decision Analysis*, vol. 6, no. 2, pp. 67-81, 2002.

[32] S. Filin, and Y. Doytsher, "A linear approach to map conflation: matching of polylines," *Surveying and Land Information Systems*, vol. 59, no. 2, pp. 107-114, June 1999.

[33] S. Yuan and C. Tao, "Development of conflation components," in *Proc. The International Conference on Geoinformatics and Socioinformatics*, Ann Arbor, 1999, pp. 1-12.

[34] ESRI, *Arc/Info User Manual*. Redlands, CA: ESRI Press, 2004.

[35] C. C. Chen, C. A. Knoblock, C. Shahabi, Y. Chiang, and S. Thakkar, 2004. "Automatically and accurately conflating orthoimagery and street maps," in *Proc. ACM-GIS'04*, Washington, DC, 2004.

[36] GIS/Trans, ltd, 1995, Conflation background information. [Online]. Available: http://www.gistrans.com/products/cf_info.html

[37] M. Kass, A. Witkin and D. Terzopoulos, "Snakes: active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321-331, Jan. 1988.

[38] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *International Journal of Computer Vision*, vol. 22, no. 1, pp. 61-79, Feb. 1997.

[39] T. F. Chan and L. A. Vese, "Active contours without edges," *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 266-277, Feb. 2001.

[40] L. D. Cohen and I. Cohen, "Finite-element methods for active contour models and balloons for 2-D and 3-D images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 1131-1147, Nov. 1993.

[41] T. McInerney and D. Terzopoulos, "Deformable models in medical image analysis: a survey," *Medical Image Analysis*, vol. 1, no. 2, pp. 91-108, June 1996.

[42] C. Xu and J. L. Prince, "Snakes, shapes and gradient vector flow," *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 359-369, Mar. 1998.

[43] A. L. Yuille, P. W. Hallinan, and D. S. Cohen, "Feature extraction from faces using deformable templates," *International Journal of Computer Vision*, vol. 8, no. 2, pp. 99-111, Aug. 1992.

[44] A. Gruen, and H. Li, "Semi-automatic linear feature extraction by dynamic programming and LSB-snakes," *Photogrammtric Engineering & Remote Sensing*, vol. 63, no. 8, pp. 985-995, Aug. 1997.

[45] A. Baumgartner, C. Steger, H. Mayer, W. Eckstein, and H. Ebner, "Automatic road extraction based on multi-scale, grouping and context," *Photogrammetric Engineering & Remote Sensing*, vol. 65, no. 7, pp. 777-785, July 1999.

[46] I. Laptev, H. Mayer, T. Lindeberg, W. Eckstein, C. Steger, and A. Baumgartner, "Automatic extraction of roads from aerial images based on scale space and snakes," *Machine Vision and Applications*, vol. 12, no. 1, pp. 23-31, July 2000.

[47] D. Klang, "Automatic detection of changes in road databases using satellite imagery," *International Archives of Photogrammetry and Remote Sensing*, vol. 32, no. 4, pp. 293-298, 1998.

[48] M. F. A. Fortier, D. Ziou, C. Armenakis, and S. Wang, "Automated correction and updating of road databases from high-resolution imagery," *Canadian Journal of Remote Sensing*, vol. 27, no. 1, pp. 76-89, Feb. 2001.

[49] P. Agouris, A. Stefanidis, and S. Gyftakis, "Differential snakes for change detection in road segments," *Photogrammetric Engineering & Remote Sensing*, vol. 67, no. 12, pp. 1391-1399, Dec. 2001.

[50] L. Bentabet, S. Jodouin, D. Ziou, and J. Vaillancourt, "Road vectors update using SAR imagery: a snake-based method," *IEEE Trans. Geoscience and Remote Sensing*, vol. 41, no. 8, pp. 1785-1803, Aug. 2003.

[51] W. M. Neuenschwander, P. Fua, L. Iverson, G. Szekely, and O. Kubler, "Ziplock snakes," *International Journal of Computer Vision*, vol. 25, no. 3, pp. 191-201, Dec. 1997.

[52] U.S. Army Corps of Engineers, Geographic Information Integration and Generation tools, [Online]. Available: http://www.erdc.usace.army.mil/pls/erdcpub/docs/erdc/images/ERDC_FS_Research_GIIGT.pdf

[53] B. Zitova and J. Flusser, "Image registration methods: a survey," *Image and Vision Computing*, vol. 21, no. 11, pp. 977-1000, Oct. 2003.

[54] G. S. Cox and G. D. Jager, "A survey of point pattern matching techniques and a new approach to point pattern recognition," in *Proc. South African Symposium on Communications and Signal Processing*, 1992, pp. 243-248.

[55] G. Stockman, S. Kopstein, and S. Benett, "Matching images to models for registration and object detection via clustering," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. PAMI-4, no. 3, pp. 229-, May 1982.

[56] A. Goshtasby and G. C. Stockman, "Point pattern matching using convex hull edges," *IEEE Trans. System, Man and Cybernetics*, vol. SMC-15, no. 5, pp. 631-637. Sep.-Oct. 1985.

[57] S. Umeyama, "Least-squares estimation of transformation parameters between two point pattern," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, no. 4, pp. 376-380, Apr. 1991.

[58] S. H. Chang, F. H. Cheng, W. H. Hsu, and G. Z. Wu, "Fast algorithm for point pattern matching: invariant to translations, rotations, and scale changes," *Pattern Recognition*, vol. 30, no. 2, pp. 311-320, Feb. 1997.

[59] D. Conte, P. Foggia, C. Sansone, and M. Vento, "Thirty years of graph matching in pattern recognition," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 18, no. 3, pp. 265-298, May 2004.

[60] T. S. Caetano, T. Caelli, D. Schuurmans, and D. A.C. Barone, "Graphical models and point pattern matching," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, no. 10, pp. 1646-1663, Oct. 2006.

[61] S. Ullman, *The Interpretation of Visual Motion*. Cambridge, MA: MIT Press, 1979.

[62] S. Kumar, M. Sallam, and D. Goldgof, "Matching point features under small nonrigid motion," *Pattern Recognition*, vol. 34, no. 12, pp. 2353-2365, Dec. 2001.

[63] G. Scott and H. Longuet-Higgins, "An algorithm for associating the features of two patterns," in *Proc. Royal Society London*, 1991, pp. 21-26.

[64] M. Pilu, "A direct method for stereo correspondence based on singular value decomposition," in *Proc. IEEE Computer Vision and Pattern Recognition Conference*, 1997, pp. 261-266.

[65] L. S. Shapiro and J. M. Brady, "Feature-based correspondence: an eigenvector approach," *Image and Vision Computing*, vol. 10, no. 5, pp. 283-286, June 1992.

[66] M. Carcassoni and E. R. Hancock, "Weighted graph-matching using modal clusters," in *Proc. 3rd IAPR-TC15 Workshop Graph-Based Representations in Pattern Recognition*, 2001, pp. 260-269.

[67] P. J. Besl and N. D. Mckay, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239-256, Feb. 1992.

[68] H. Chui and A. Rangarajan, "A new point matching algorithm for non-rigid registration," *Computer Vision and Image Understanding*, vol. 89, no. 2-3, pp. 114-141, Feb.-Mar. 2003.

[69] N. Paragios, M. Rousson, and V. Ramesh, "Non-rigid registration using distance functions," *Computer Vision and Image Understanding*, vol. 89, no. 2-3, pp. 142-165, Feb.-Mar. 2003.

[70] S. Ranade and A. Rosenfeld, "Point pattern matching by relaxation," *Pattern Recognition*, vol. 12, no. 4, pp. 269-275, 1980.

[71] C. Wang, H. Sun, S. Yada, and A. Rosenfeld, "Some experiments in relaxation image matching using corner features," *Pattern Recognition*, vol. 16, no. 2, pp. 167-182, 1983.

[72] J. Ton and A. K. Jain, "Registering Landsat images by point matching," *IEEE Tran. Geoscience and Remote Sensing*, vol. 27, no. 5, pp. 642-651, Sep. 1989.

[73] H. Ogawa, "Labeled point pattern matching by fuzzy relaxation," *Pattern Recognition*, vol. 17, no. 5, pp. 569-573, 1984.

[74] J. P. Starink and E. Backer, "Finding point correspondences using simulated annealing," *Pattern Recognition*, vol. 28, no. 2, pp. 231-240, Feb. 1995.

[75] L. Zhang, W. Xu and C. Chang, "Genetic algorithm for affine point pattern matching," *Pattern Recognition Letters*, vol. 24, no. 1-3 , pp. 9-19, Jan. 2003.

[76] P. Yin, "Particle swarm optimization for point pattern matching," Journal of Visual Communication and Image Representation, vol. 17, no. 1, pp. 143-162, Feb. 2006.

[77] L. G. Brown, "A survey of image registration techniques," *ACM Computing Surveys*, vol. 24, no. 4, pp. 325-376, Dec. 1992.

[78] A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, Orlando: Academic Press, 1982.

[79] M. F. Goodchild, "Attribute accuracy," in *Elements of Spatial Data Quality*, 1st ed. S. Guptill and J. Morrison Ed. Oxford: Elsevier Science, 1995.

[80] H. Kang, "Spatial data integration: a case study of map conflation with census bureau and local government data," in *2001 University Consortium for Geographic Information Science Summer Assembly*.

[81] Y. Doytsher, S. Filin and E. Ezra, "Transformation of datasets in a linear-based map conflation framework," *Surveying and Land Information Systems*, vol. 61, no. 3, pp. 159-169, Sep. 2001.

[82] B. Kovalerchuk, W. Sumner, and J. Schwing, "Image registration and conflation based on structural characteristics," in *Proc. International Conference on Imaging Science, Systems, and Technology*, Las Vegas, 2003, pp. 239-245.

[83] S. Kundu, "Conflating two polygonal lines," *Pattern Recognition*, vol. 39, no. 3, pp. 363-372. March 2006.

[84] L.H. Quam, "Road tracking and anomaly detection in aerial imagery," in *Image Understanding Workshop*, Pittsburgh, 1978, pp. 51-55.

[85] M.A. Fischler, J.M. Tenenbaum, and H.C. Wolf, "Detection of roads and linear structures in low-resolution aerial imagery using a multisource knowledge integration technique," *Computer Graphics and Image Processing*, vol. 15, no. 3, pp. 201-223, March 1981.

[86] D. M. Mckeown, and J.L. Denlinger, "Cooperative methods for road tracking in aerial imagery," *Proceedings Computer Vision and Pattern Recognition 88*, Ann Arbor, 1988, pp. 662-672.

[87] Vasudevan, S., Cannon, R.L., Bezdek, J.C., and Cameron, W.L., Heuristics for Intermediate-Level Road Finding Algorithms, *Computer Vision, Graphics, and Image Processing*, vol. 44, no. 1, pp. 175-190, November 1988.

[88] A. Zlotnick, and P.D. Carnine, "Finding road seeds in aerial images," *CVGIP: Image Understanding*, vol. 57, no. 2, pp. 243-260, March 1993.

[89] G. Vosselman, and J. Knecht, "Road tracing by profile matching and Kalman filtering," in *Automatic Extraction of Man-Made Objects from Aerial and Space*

Images (I). Gruen, Kuebler, and Agouris, Ed. Basel: Birkhauser Verlag, 1995, pp. 265-274.

[90] D. Geman, and B. Jedynak, "An active testing model for tracking roads in satellite images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 1, pp. 1-14, January 1996.

[91] M. Barzohar, and D.B. Cooper, "Automatic finding of main roads in aerial images by using geometric-stochastic models and estimation," *IEEE transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 7, pp. 707-721, July 1996.

[92] C. Steger, "An unbiased detector of curvilinear structures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 2, pp. 113-125, February 1998.

[93] Y. Wang, and J. Trinder, "Use of topology in automatic road extraction," *International Archives of Photogrammetry and Remote Sensing*, Columbus, 1998.

[94] K. Price, "Urban street grid description and verification," *Fifth IEEE Workshop on Applications of Computer Vision*, Palm Springs, 2000, pp. 148-154.

[95] A. Kartartzis, H. Sahli, V. Pizurica, and J. Cornelis, "A model-based approach to the automatic extraction of linear features," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, no. 9, pp. 2073-2079, Sept. 2001.

[96] P. Doucette, P. Agouris, A. Stefanidis, and M. Musavi, "Self-organised clustering for road extraction in classified imagery," *ISPRS Journal of Photogrammetry & Remote Sensing*, vol. 55, no. 5-6, pp. 347-358, March 2001.

[97] S. Hinz and A. Baumgartner, "Automatic extraction of urban road networks from multi-view aerial imagery," *ISPRS Journal of Photogrammetry & Remote Sensin*g, vol. 58, no. 1-2, pp. 83-98, June 2003.

[98] A. Gruen, O. Kuebler, and P. Agouris, *Automatic Extraction of Man-Made Objects from Aerial and Space Images (I)*. Basel: Birkhauser Verlag, 1995.

[99] A. Gruen, E. P. Baltsavias, and O. Henricsson, *Automatic Extraction of Man-Made Objects from Aerial and Space Images (II)*. Basel: Birkhauser Verlag, 1997.

[100] E. P. Baltsavias, A. Gruen and L. V. Gool, *Automatic Extraction of Man-Made Objects from Aerial and Space Images (III)*. Lisse: A.A. Balkema, 2002.

[101] R. A. Saleh, "Special issue on linear feature extraction from remote sensing data for road network delineation and revision," *Photogrammetric Engineering & Remote Sensing*, vol. 70, no. 12, pp. 1351-1352, Dec. 2004.

[102] K. Price, Annotated Computer Vision Bibliography, Available: http://iris.usc.edu/Vision-Notes/bibliography/cartog930.html

[103] V. Chalasani, and P. Beling, "Optimization based classifiers for road extraction," *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, Piscataway, 1998, pp. 2938-2943.

[104] M. D. Gunst, and G. Vosselman, 1997, "A semantic road model for aerial image interpretation," in *Semantic Modeling for the Acquisition of Topographic Information from Images and Maps*, Forstner and Plumer, Ed. Birkhauser Verlag, Basel, pp.107-122.

[105] R. Fiset and F. Cavayas, "Automatic comparison of a topographic map with remotely sensed images in a map updating perspective: the road network case," *International Journal of Remote Sensing*, vol. 18, no. 4, pp. 991-1006, March 1997.

[106] B. K. Jeon, J. H. Jang, and K. S. Hong, "Road detection in spaceborne SAR images based on ridge extraction," *IEEE International Conference on Image Processing*, Kobe, 1999, pp.735-739.

[107] G. Rellier, X. Descombes, and J. Zerubia, "Local registration and deformation of a road cartographic database on a SPOT satellite image," *Pattern Recognition*, vol. 35, no. 10, pp. 2213-2221, October 2002.

[108] M. Gerke, M. Butenuth, C. Heipke, and F. Willrich, "Graph-supported verification of road databases," *ISPRS Journal of Photogrammetry & Remote Sensing*, vol. 58, no. 3-4, pp. 152-165, January 2004.

[109] C. Zhang, "Towards an operational system for automated updating of road databases by integration of imagery and geodata," *ISPRS Journal of Photogrammetry & Remote Sensing*, vol. 58, no. 3-4, pp. 166-186, January 2004.

[110] J. B. Mena, "State of the art on automatic road extraction for GIS update: a novel classification," *Pattern Recognition Letters*, vol. 24, no. 16, pp. 3037-3058, Dec. 2003.

[111] A. K. Shackelford and C. H. Davis, "A hierarchical fuzzy classification approach for high-resolution multispectral data over urban areas," *IEEE Trans. Geoscience and Remote Sensing*, vol. 41, no. 9, pp. 1920-1932, Sep. 2003.

[112] J. R. Jensen, "Biophysical remote sensing," *Annals of the Association of American Geographers*, vol. 73, no. 1, pp. 111-132, Mar. 1983.

[113] F. Deschenes and D. Ziou, "Detection of line junctions and line terminations using curvilinear features," *Pattern Recognition Letters*, vol. 21, no. 6-7, pp. 637-649, June 2000.

[114] G. Koutaki and K. Uchimura, "Automatic road extraction based on cross detection in suburb," in *Proc. SPIE Electronic Imaging*, San Jose, 2004, pp.337-344.

[115] C. C. Chen, C. Shahabi, and C. A. Knoblock, "Utilizing road network data for automatic identification of road intersections from high resolution color orthoimagery," in *Proc. Second Workshop on Spatio-Temporal Database Management*, Toronto, 2004, pp. 17-24.

[116] A. Barsi and C. Heipke, "Detection road junctions by artificial neural networks," in *Proc. 2nd GRSS/ISPRS Joint Workshop on Data Fusion and Remote Sensing over Urban Areas*, Piscataway, 2003, pp. 129-132.

[117] Q. Zhang and I. Couloigner, "A wavelet approach to road extraction from high spatial resolution remotely-sensed imagery," *Geomatica*, vol. 58, no. 1, pp. 33-39, Mar. 2004.

[118] Y. Zheng and D. Doermann, "Robust point matching for nonrigid shapes by preserving local neighborhood structures," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 643-649, Apr. 2006.

[119] S. Belongie, J. Malik and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 509-522, Apr. 2002.

[120] W. Song, T. Haithcoat, and J. Keller, "A snake based approach for TIGER road data conflation," *Cartography and Geographic Information Science*, vol. 33, no. 4, pp.287-298, Oct. 2006.

[121] MASSGIS, Parcel Mapping Using GIS--A Guide to Digital Parcel Map Development for Local Governments in Massachusetts. [Online]. Available: http://www.umass.edu/tei/ogia/parcelguide/

[122] A. J. Lembo and P. F. Hopkins, "The use of adjustment computations in Geographic Information Systems for improving the positional accuracy of vector data," *Surveying and Land Information Systems*, vol. 58, no. 4, pp. 195-204. 1998.

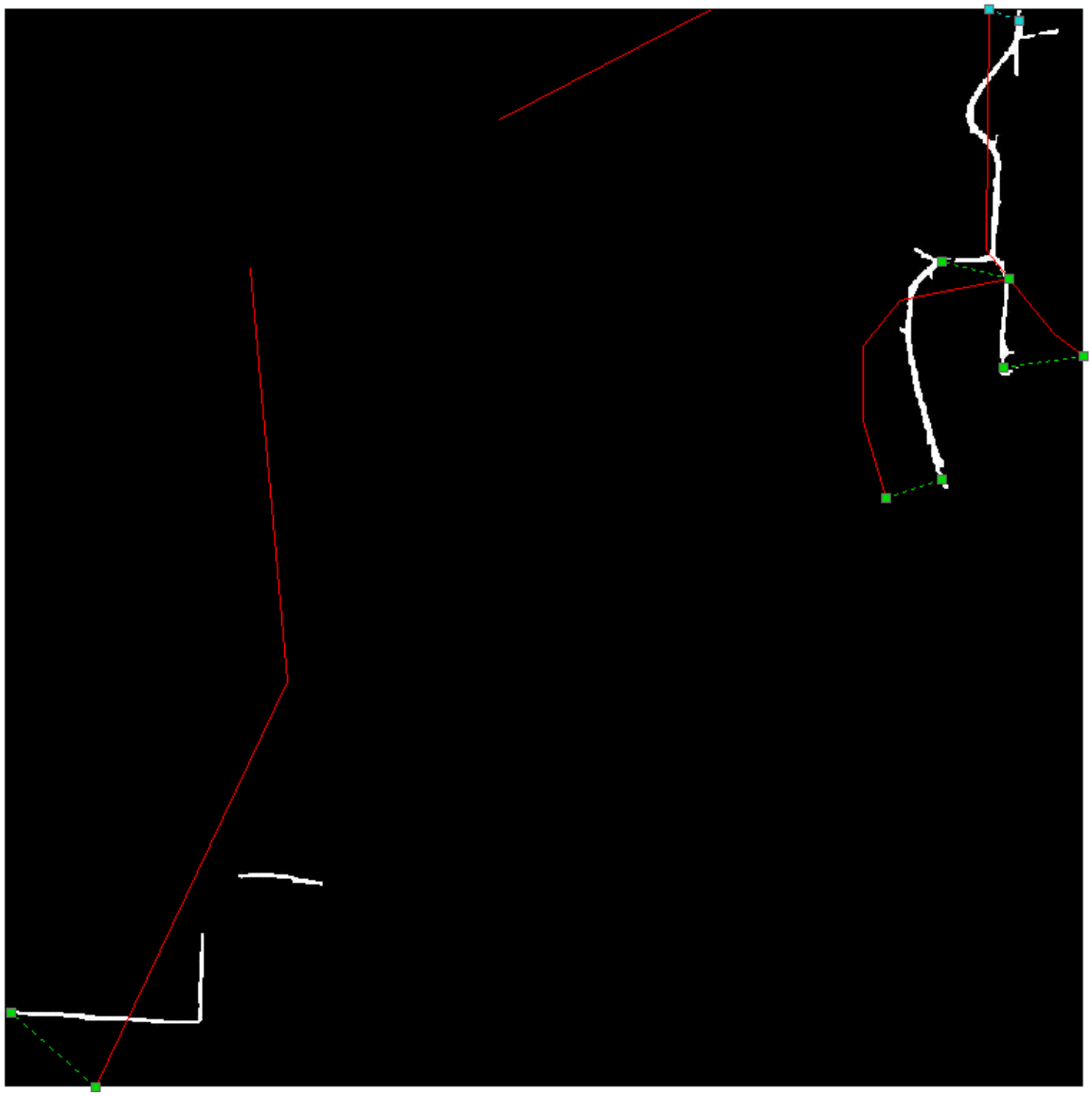
[123] A. H. Christensen, "The Practice of Piecewise Fits with Particular Reference to Property Mapping," *Surveying and Land Information Systems*, vol. 56, no. 3, pp. 165-183, Sep. 1996.

[124] ESRI, (2000, July). Map Generalization in GIS: Practical solutions with workstation Arcinfo software. [Online]. Available: http://downloads2.esri.com/support/whitepapers/ao_/Map_Generalization.pdf

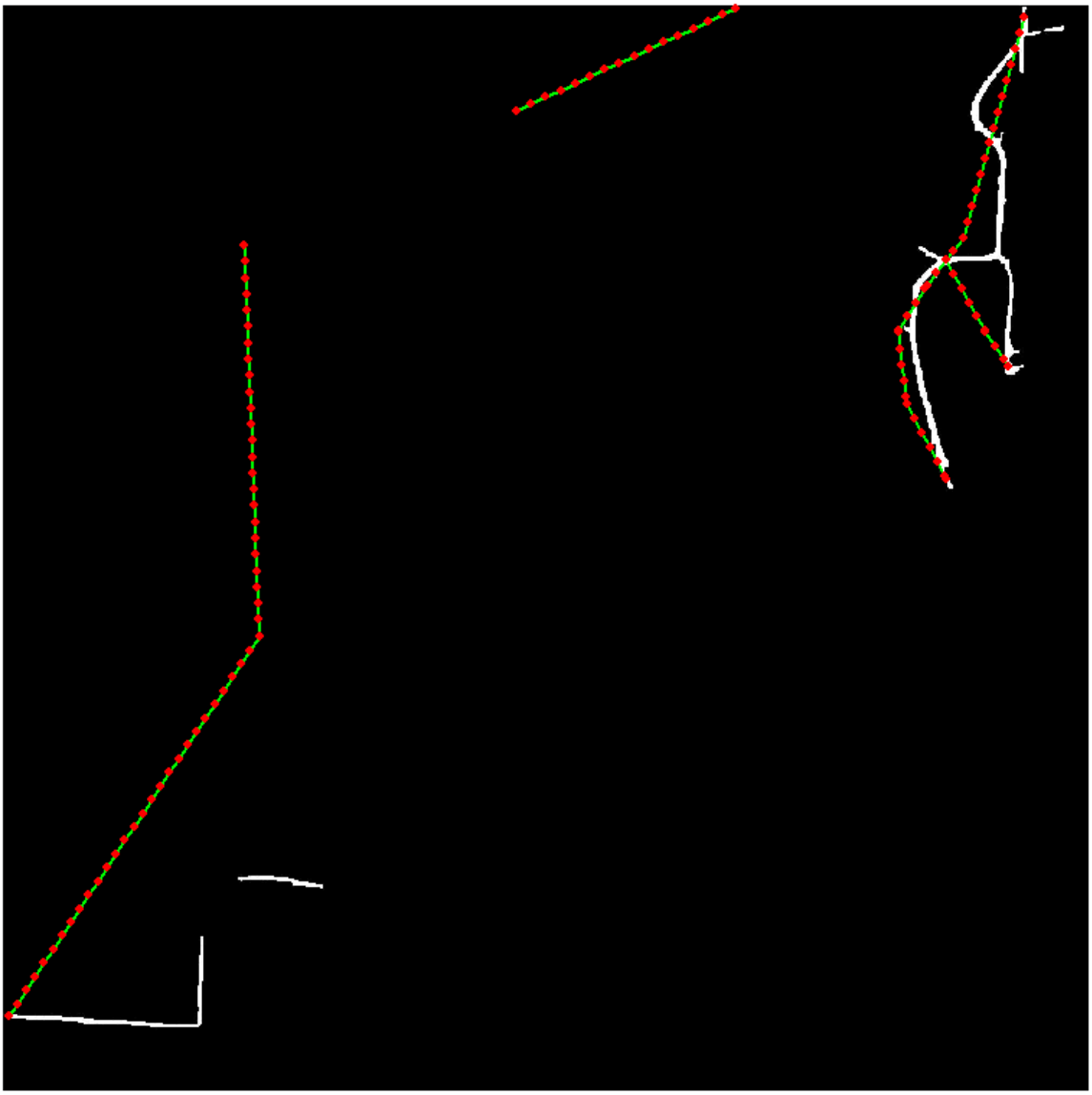
[125] T. Wright (1997, April). Geographic Information Systems. [Online]. Available: http://www.ipc.on.ca/images/Resources/gis.pdf

Appendix A Scale-up Experiment Results

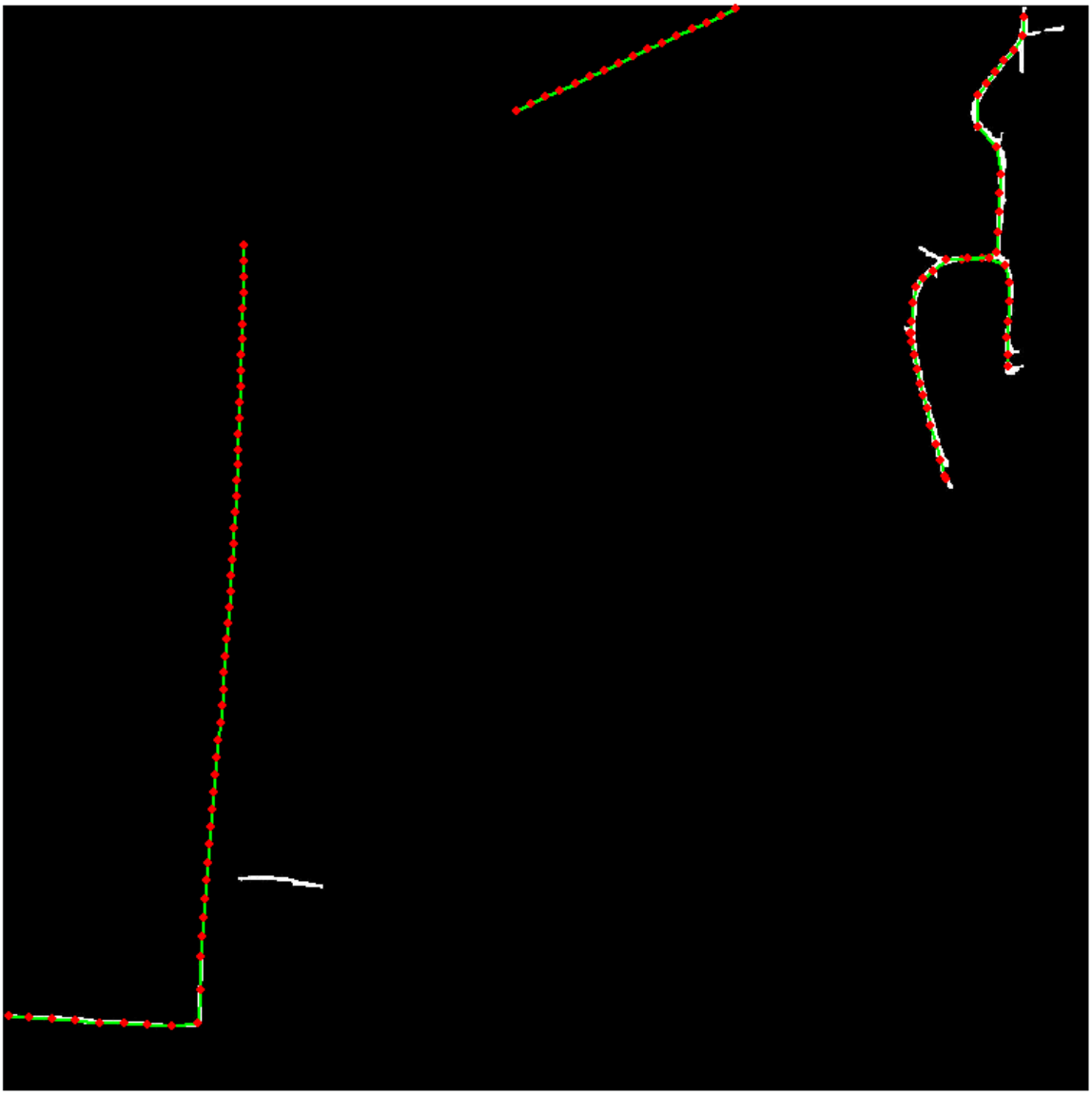
Appendix A lists all the 18 test areas used in scale-up experiment. For each test area, three figures are included (Original TIGER roads, rubber-sheeting adjusted TIGER roads, and final conflated TIGER roads).



Test area Rural_01. Original TIGER roads (red) are overlaid on road imagery. The green dashed lines indicate the matched road intersections found by relaxation labeling algorithm.

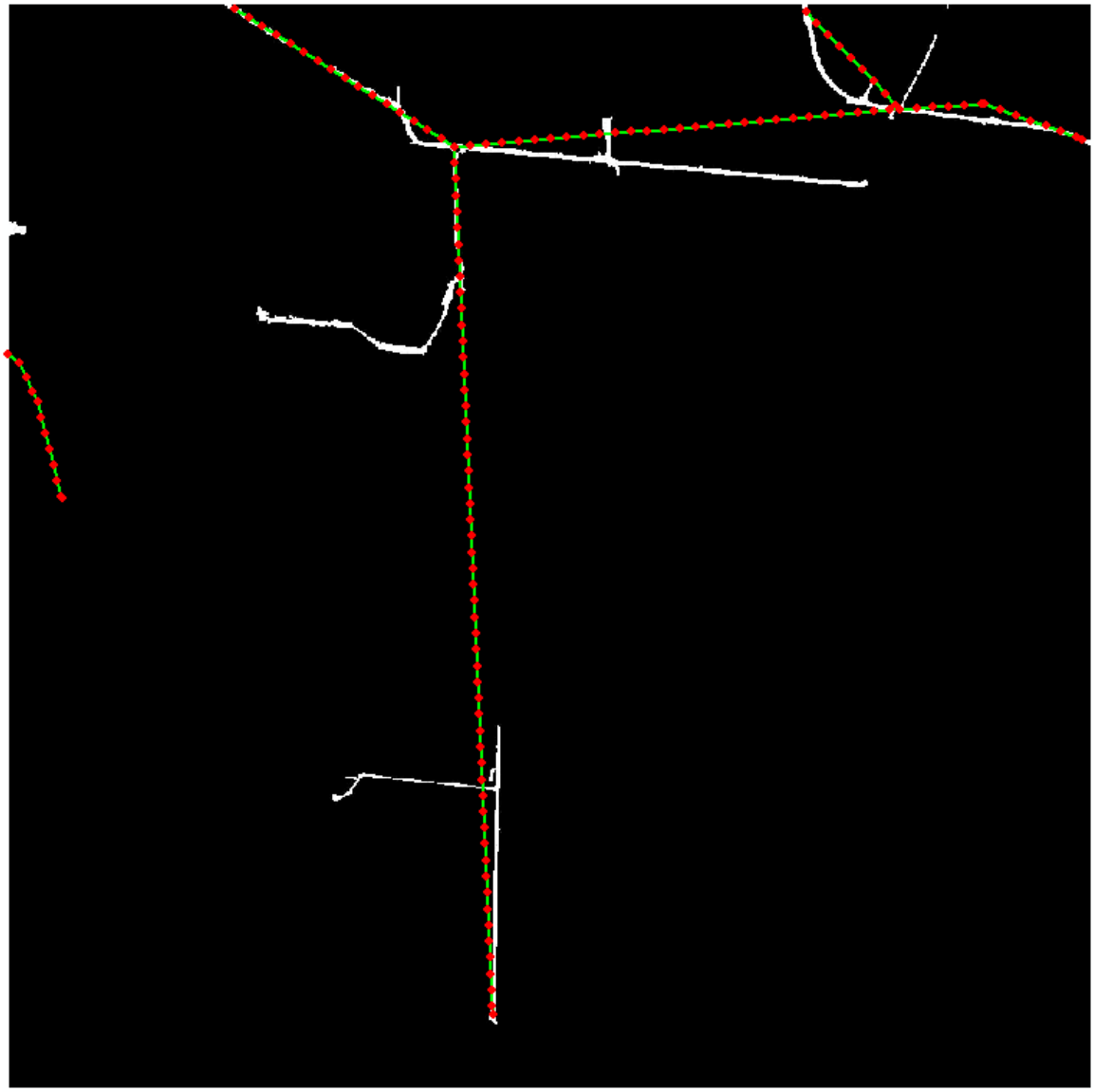


Test area Rural_01. The rubber-sheeting adjusted TIGER roads (green) with vertices (red dot) are overlaid on road imagery.

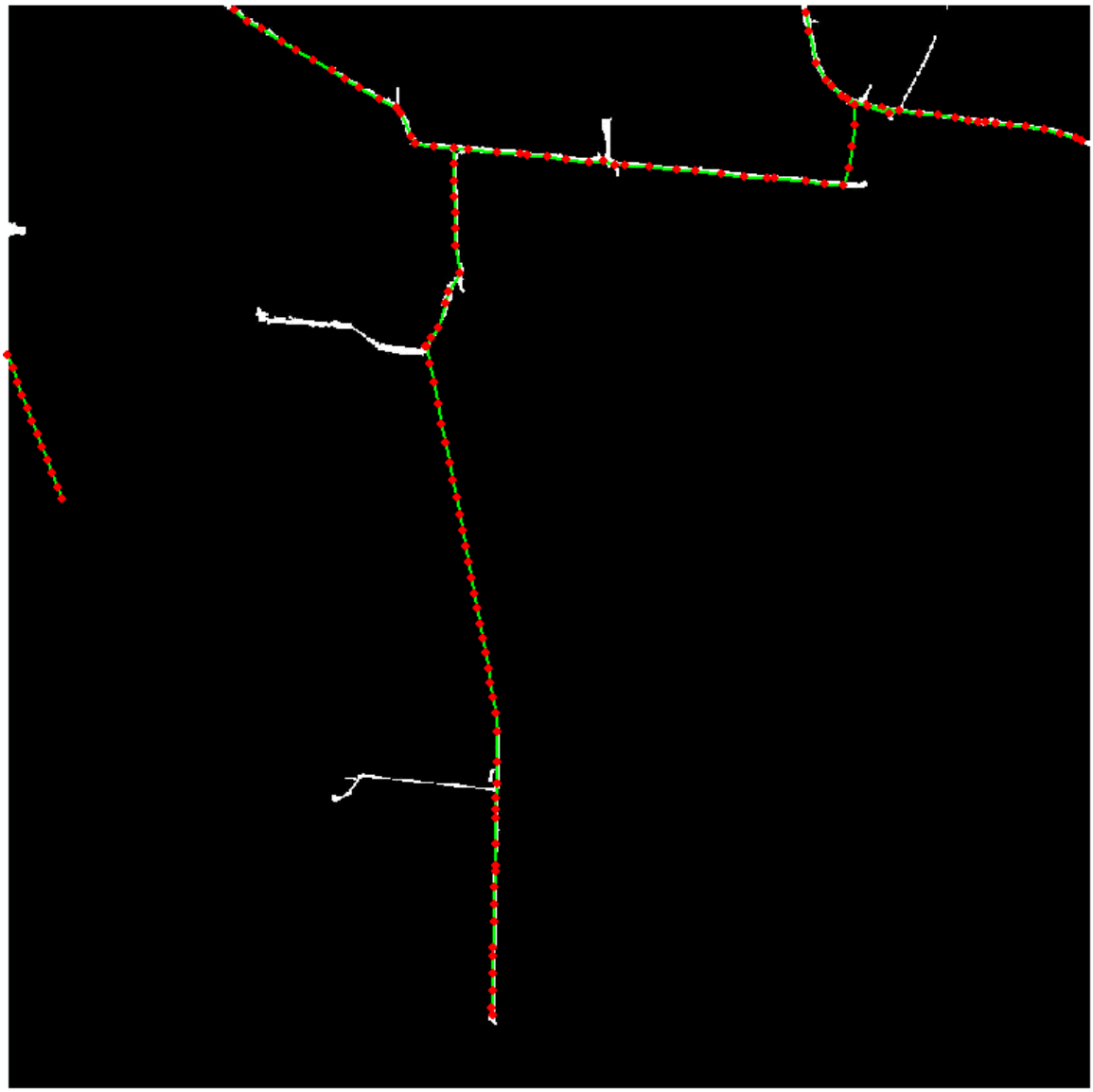


Test area Rural_01. The final conflated TIGER roads (green) with vertices (red dot) are overlaid on road imagery.

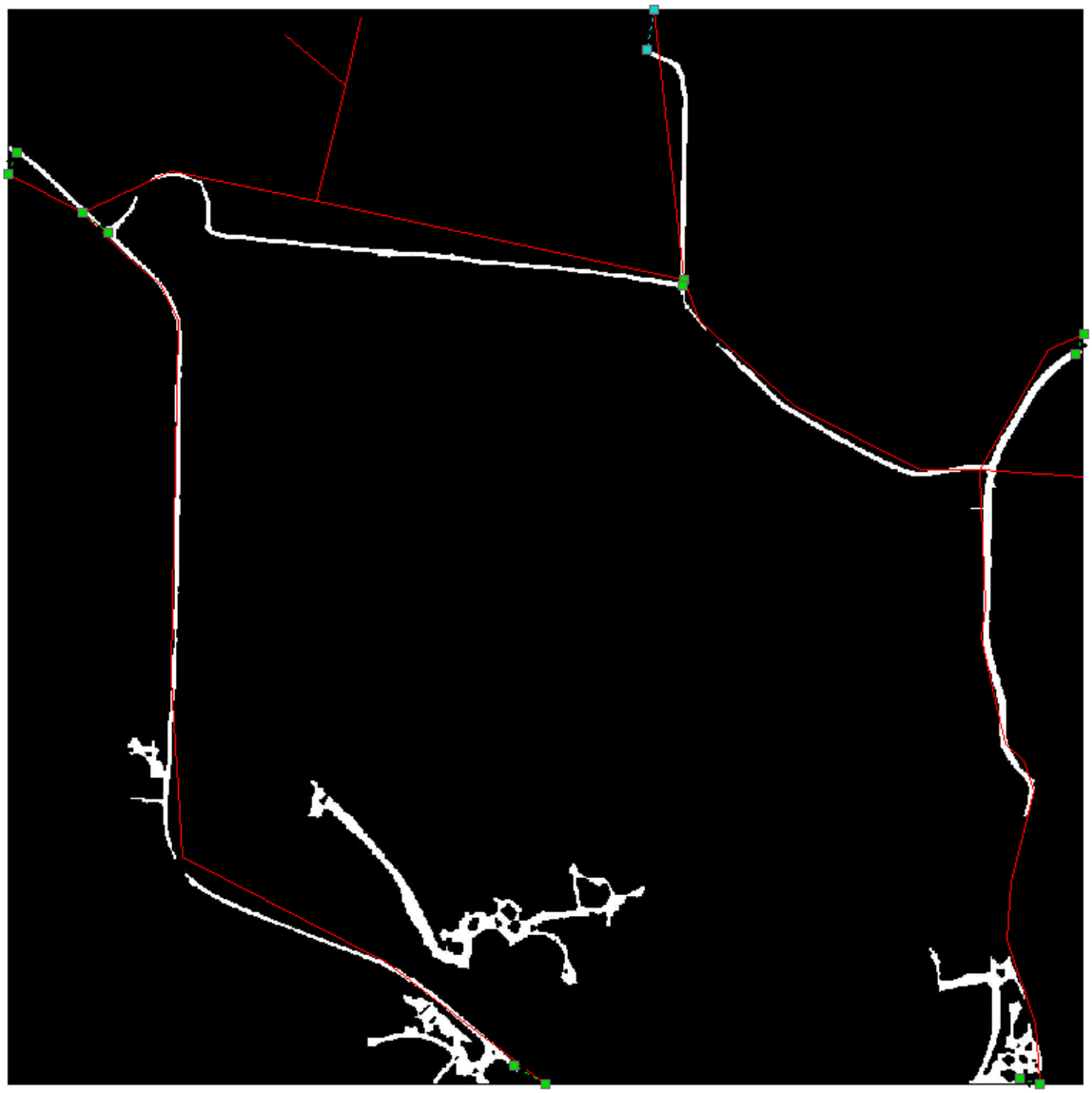
Test area Rural_02. Original TIGER roads (red) are overlaid on road imagery. The green dashed lines indicate the matched road intersections found by relaxation labeling algorithm.



Test area Rural_02. The rubber-sheeting adjusted TIGER roads (green) with vertices (red dot) are overlaid on road imagery.



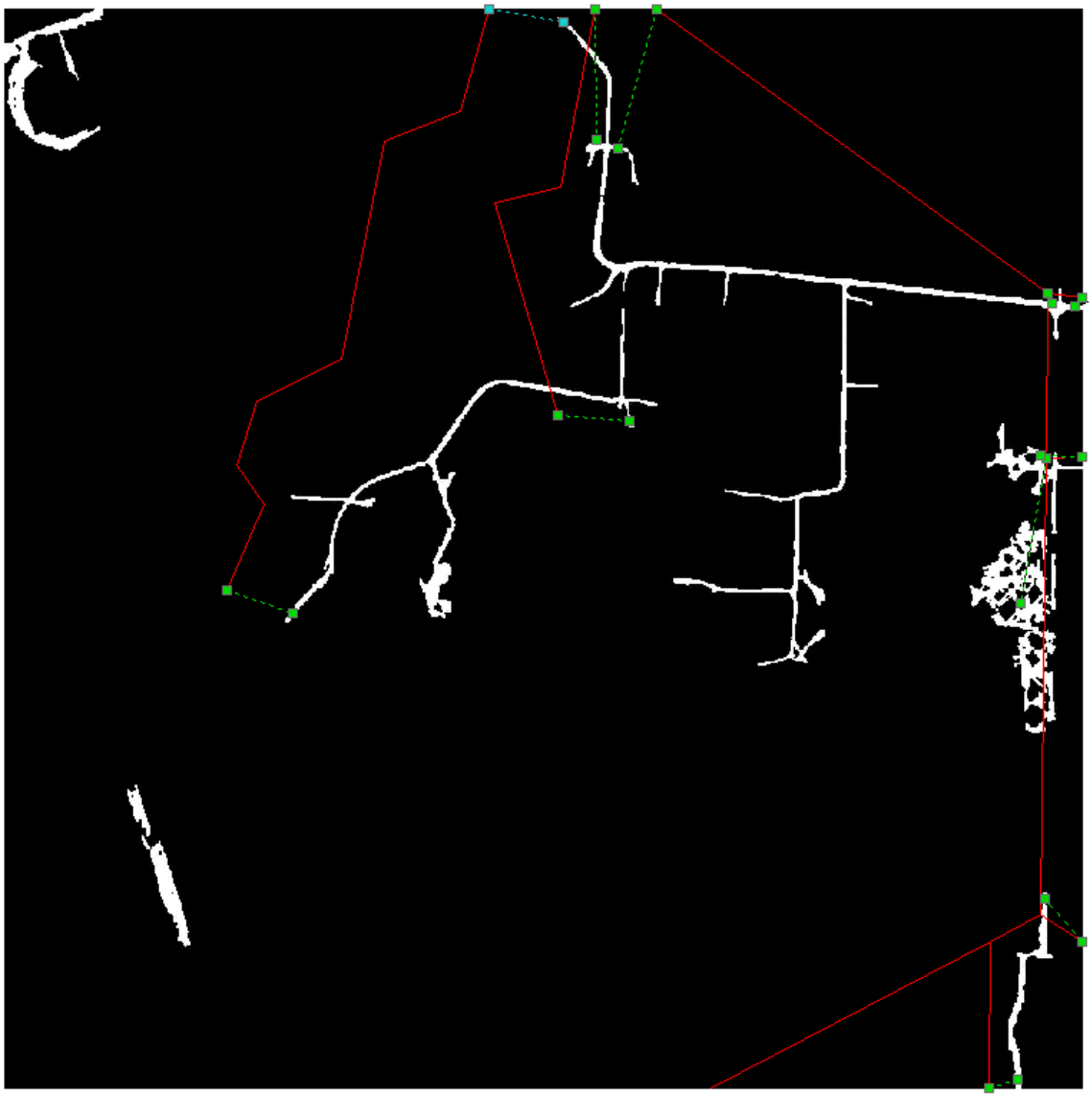
Test area Rural_02. The final conflated TIGER roads (green) with vertices (red dot) are overlaid on road imagery.



Test area Rural_03. Original TIGER roads (red) are overlaid on road imagery. The green dashed lines indicate the matched road intersections found by relaxation labeling algorithm.

Test area Rural_03. The rubber-sheeting adjusted TIGER roads (green) with vertices (red dot) are overlaid on road imagery.

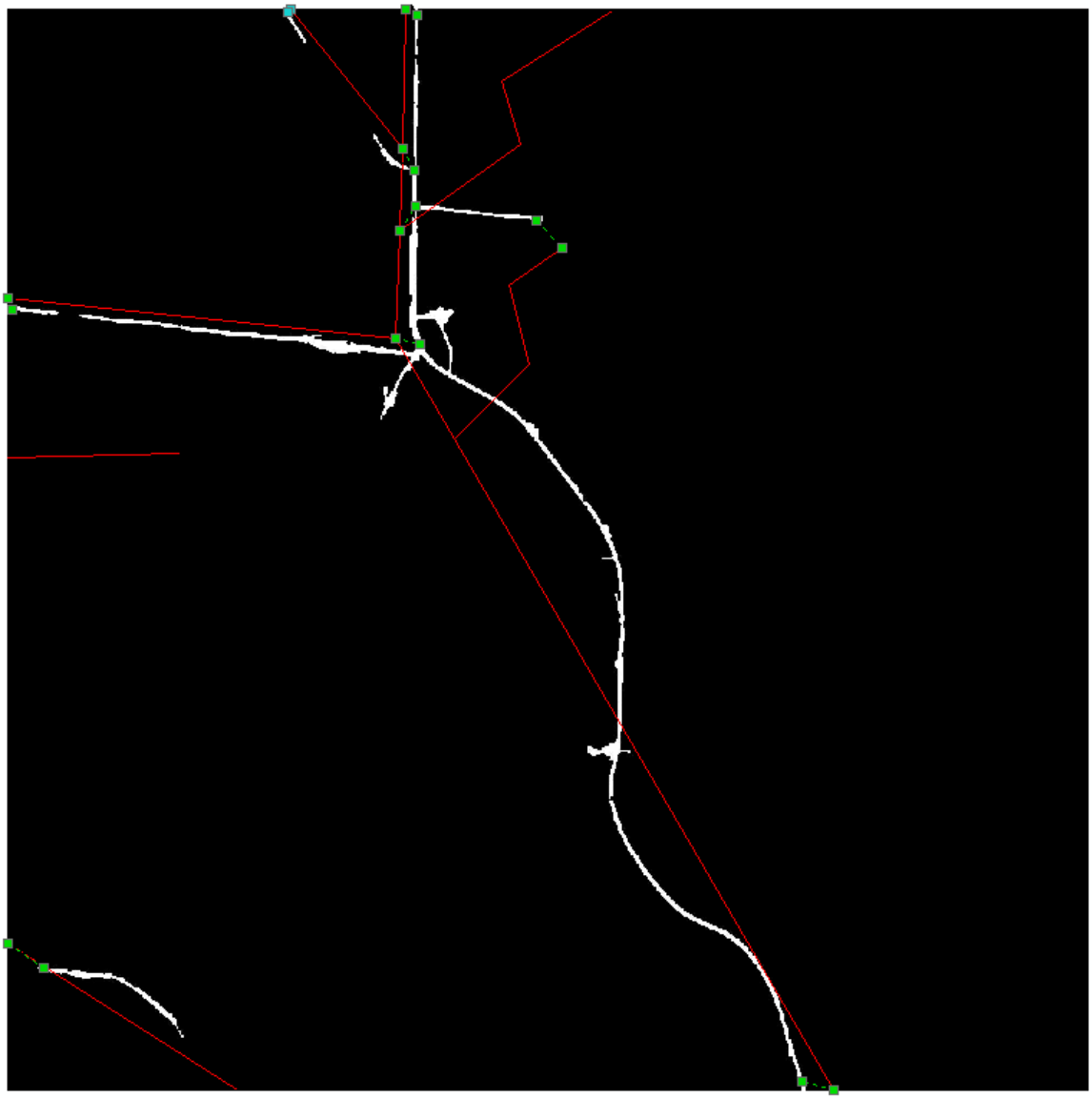
Test area Rural_03. The final conflated TIGER roads (green) with vertices (red dot) are overlaid on road imagery.



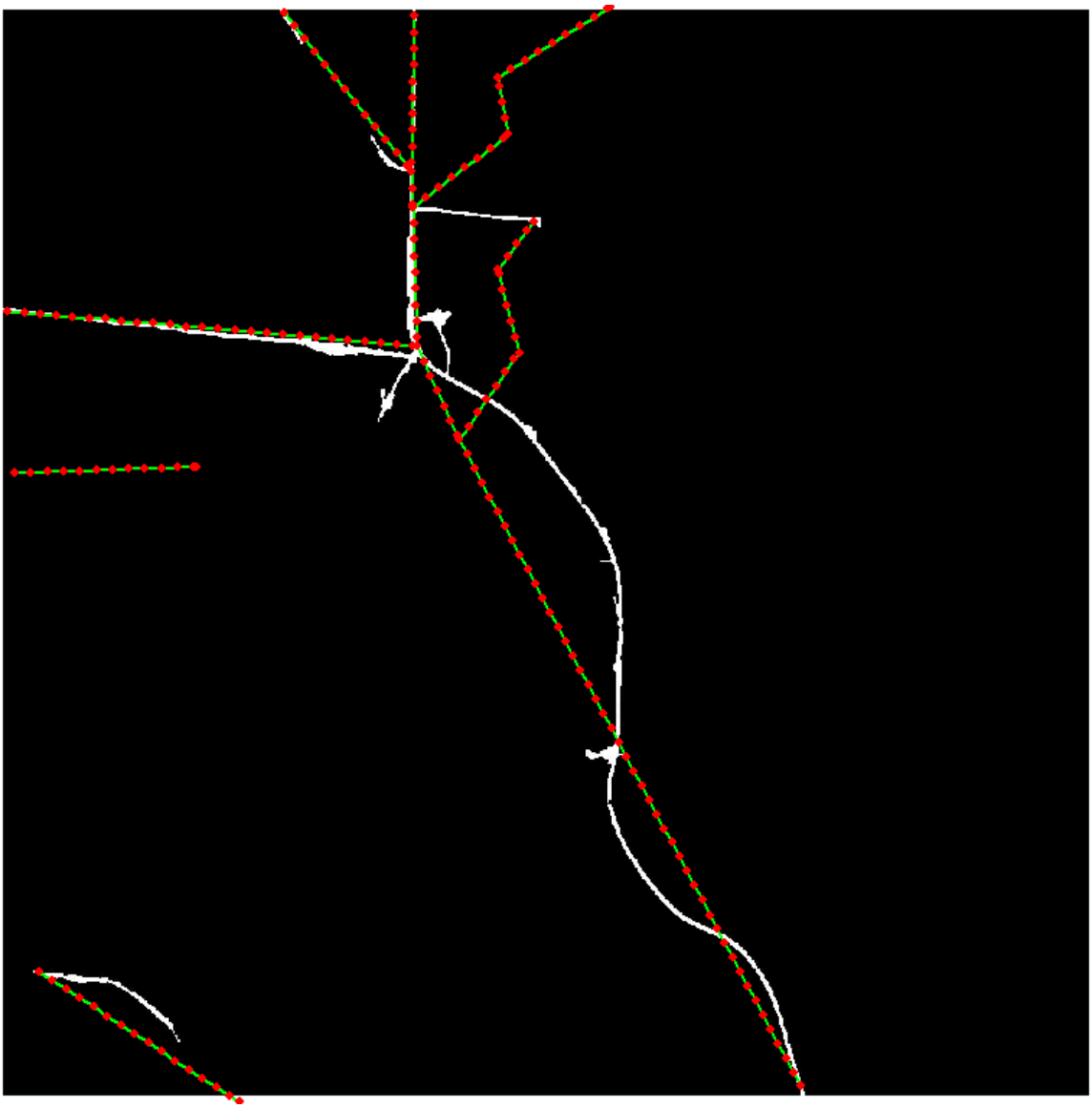
Test area Rural_04. Original TIGER roads (red) are overlaid on road imagery. The green dashed lines indicate the matched road intersections found by relaxation labeling algorithm.

Test area Rural_04. The rubber-sheeting adjusted TIGER roads (green) with vertices (red dot) are overlaid on road imagery.

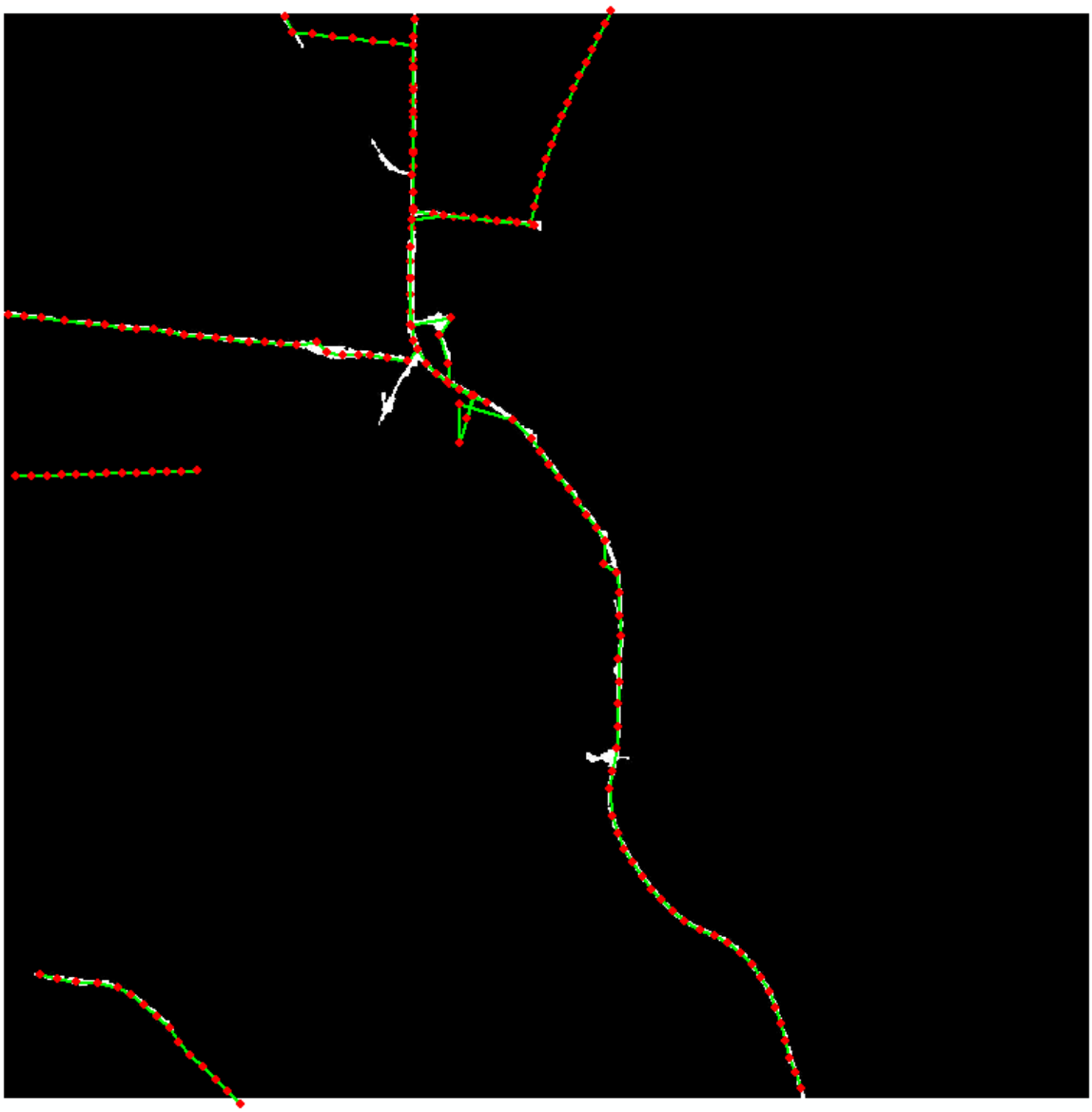
Test area Rural_04. The final conflated TIGER roads (green) with vertices (red dot) are overlaid on road imagery.



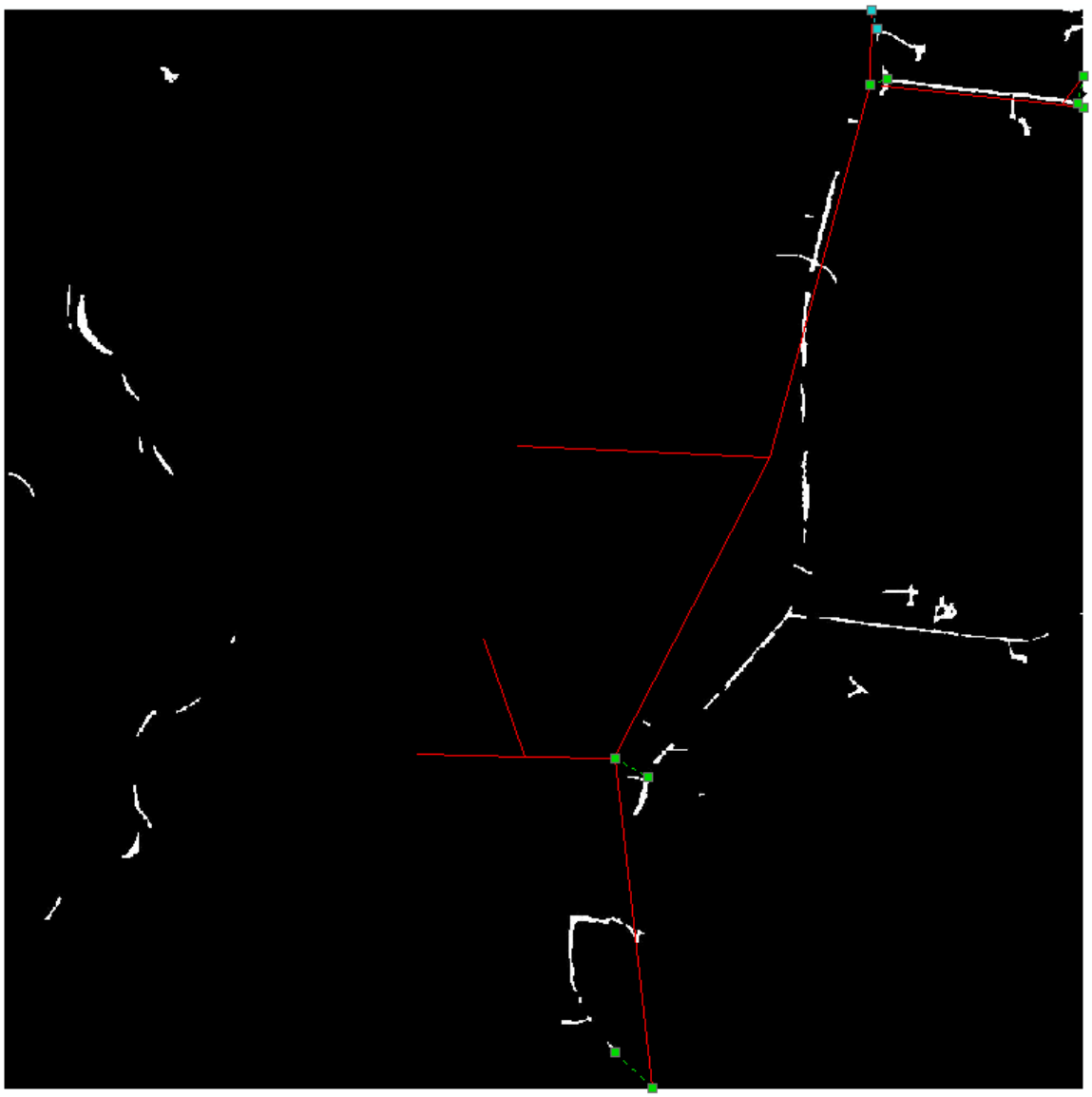
Test area Rural_05. Original TIGER roads (red) are overlaid on road imagery. The green dashed lines indicate the matched road intersections found by relaxation labeling algorithm.



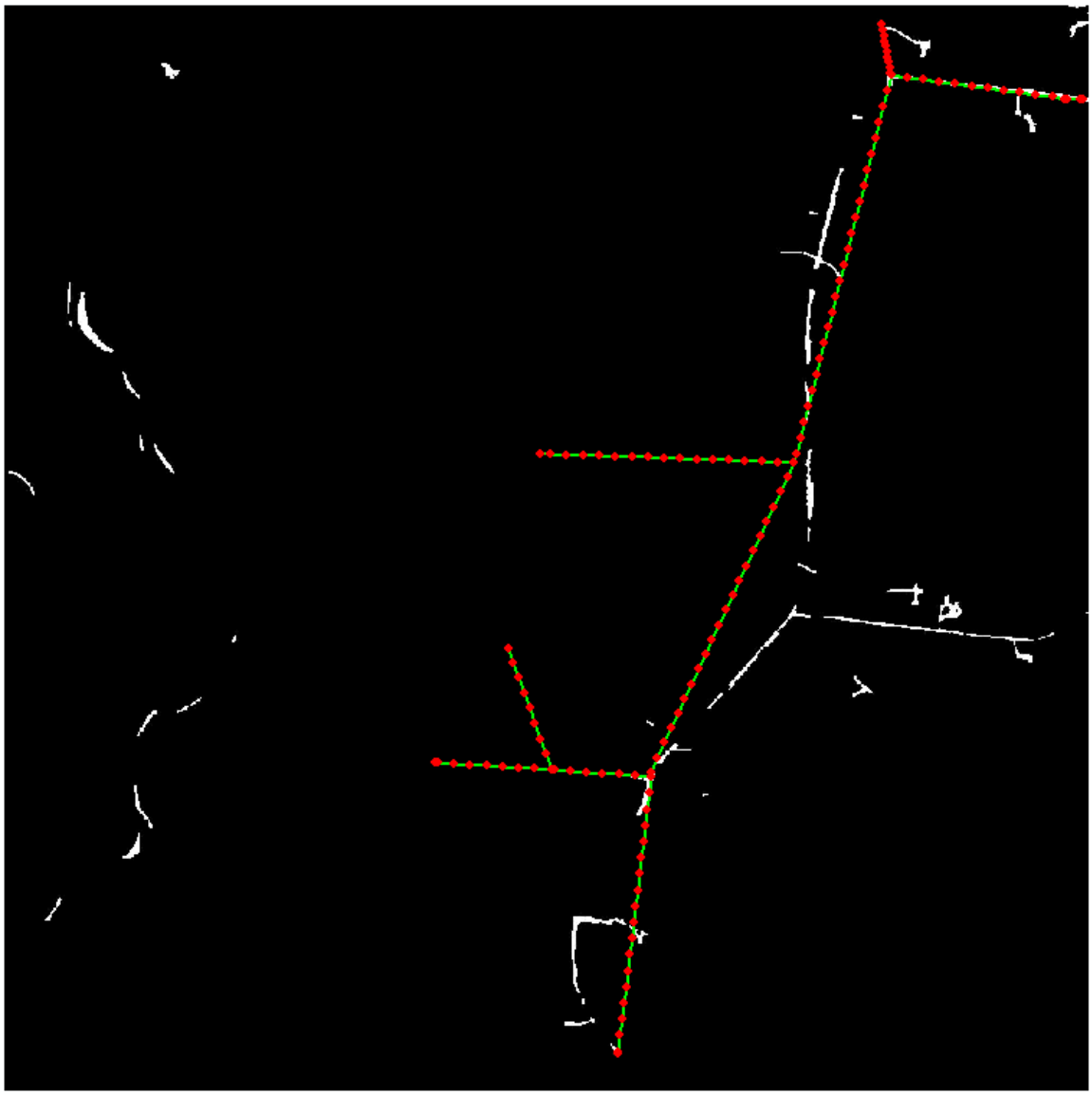
Test area Rural_05. The rubber-sheeting adjusted TIGER roads (green) with vertices (red dot) are overlaid on road imagery.



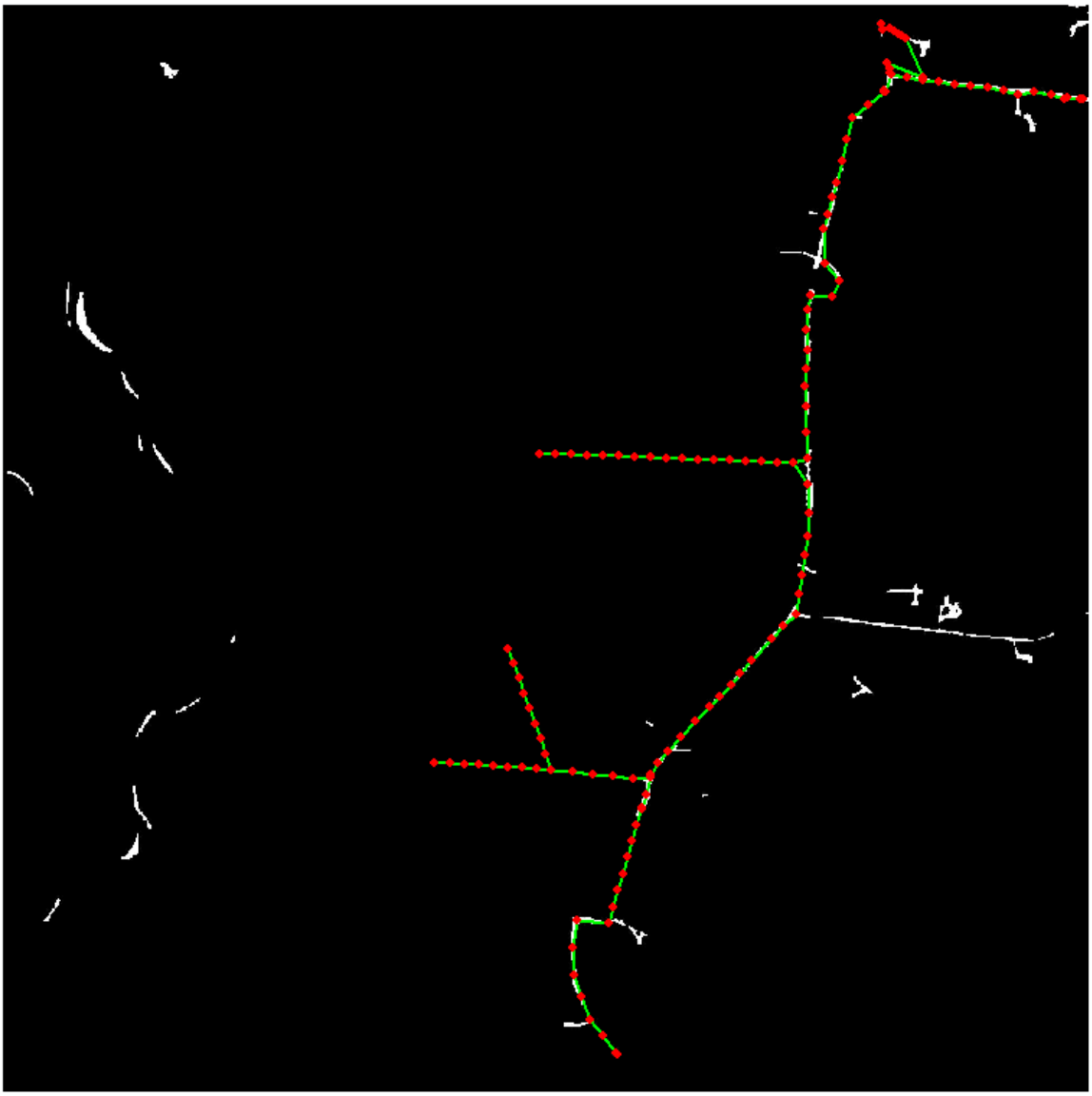
Test area Rural_05. The final conflated TIGER roads (green) with vertices (red dot) are overlaid on road imagery.



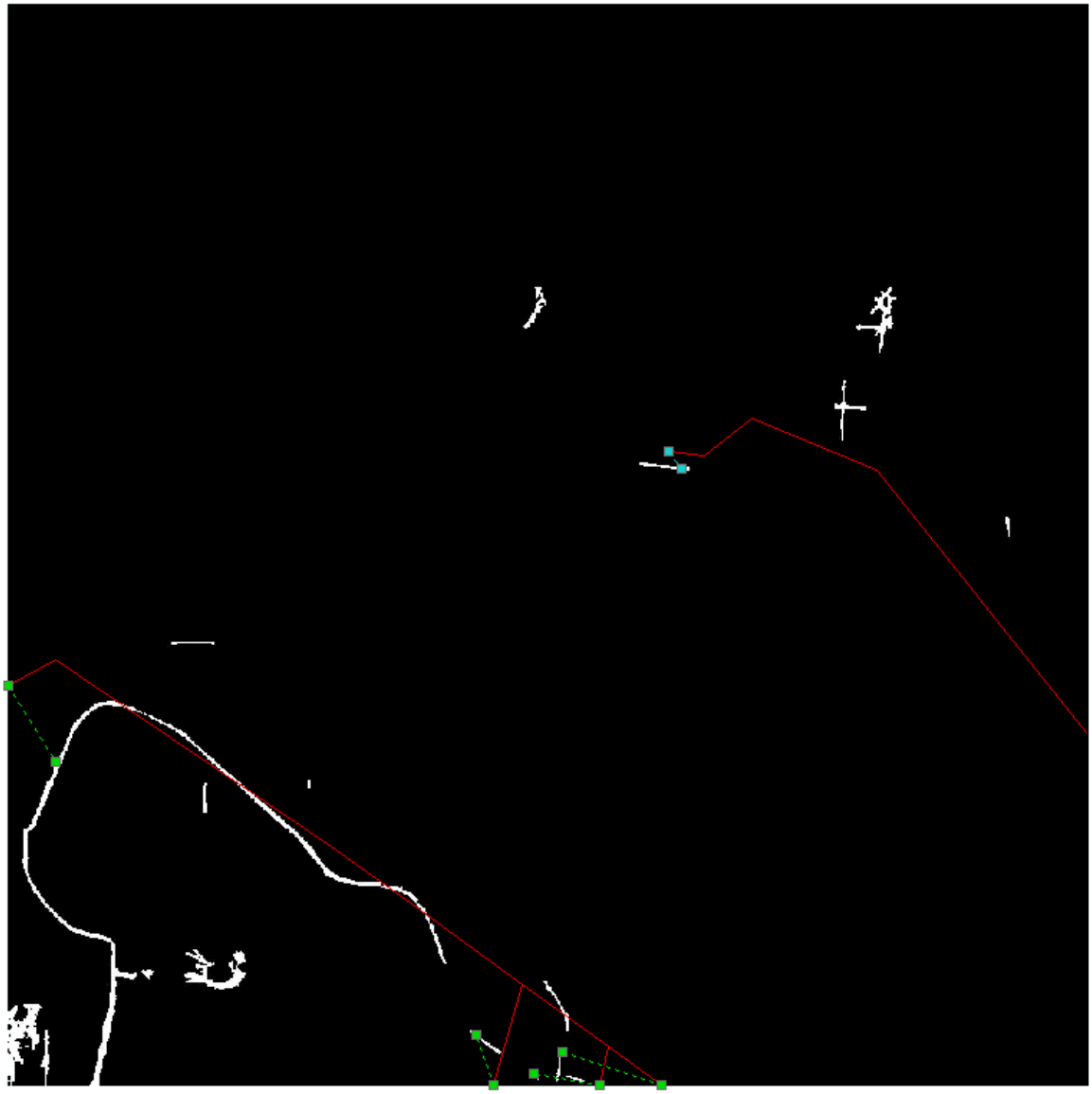
Test area Rural_06. Original TIGER roads (red) are overlaid on road imagery. The green dashed lines indicate the matched road intersections found by relaxation labeling algorithm.



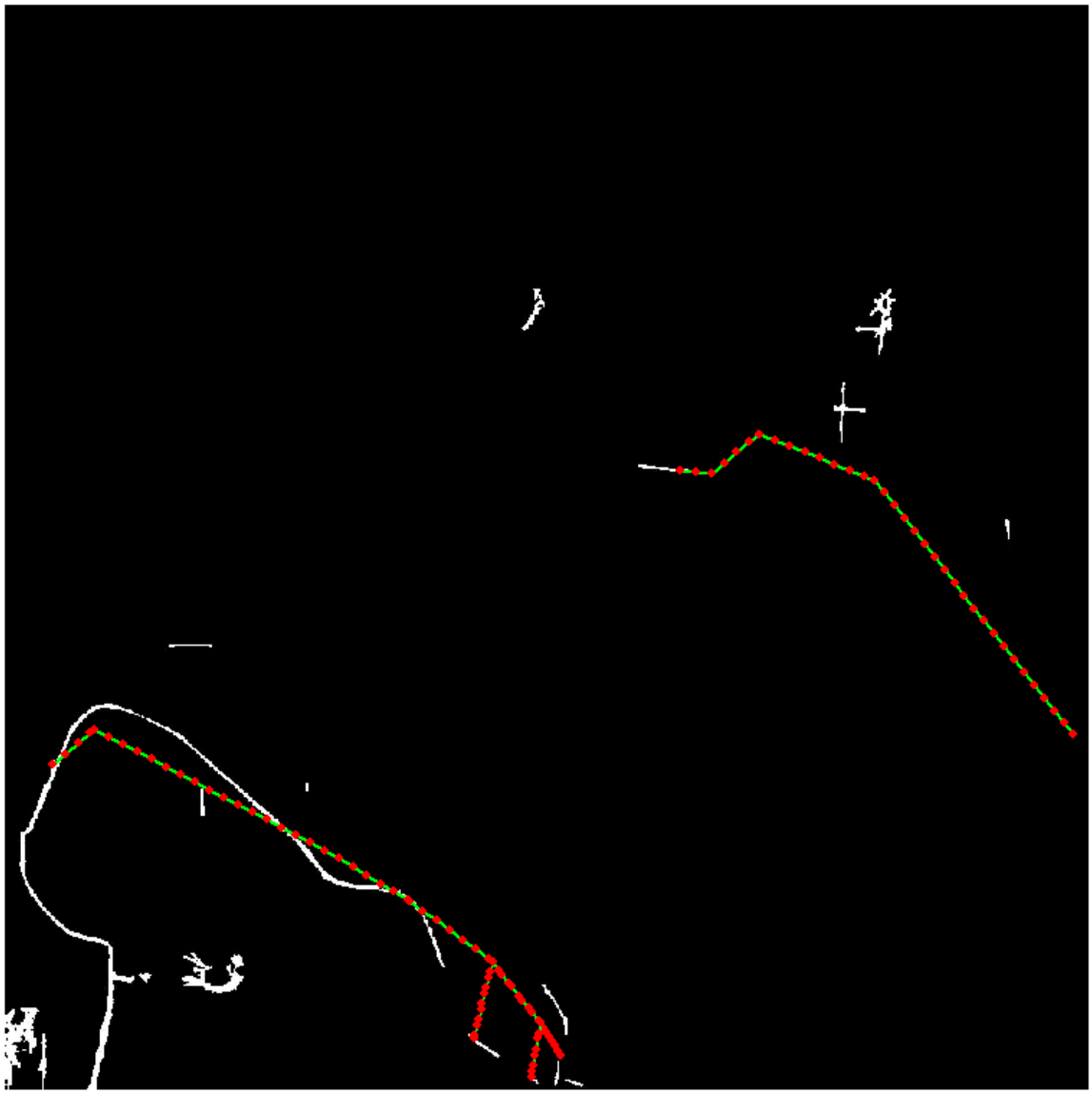
Test area Rural_06. The rubber-sheeting adjusted TIGER roads (green) with vertices (red dot) are overlaid on road imagery.



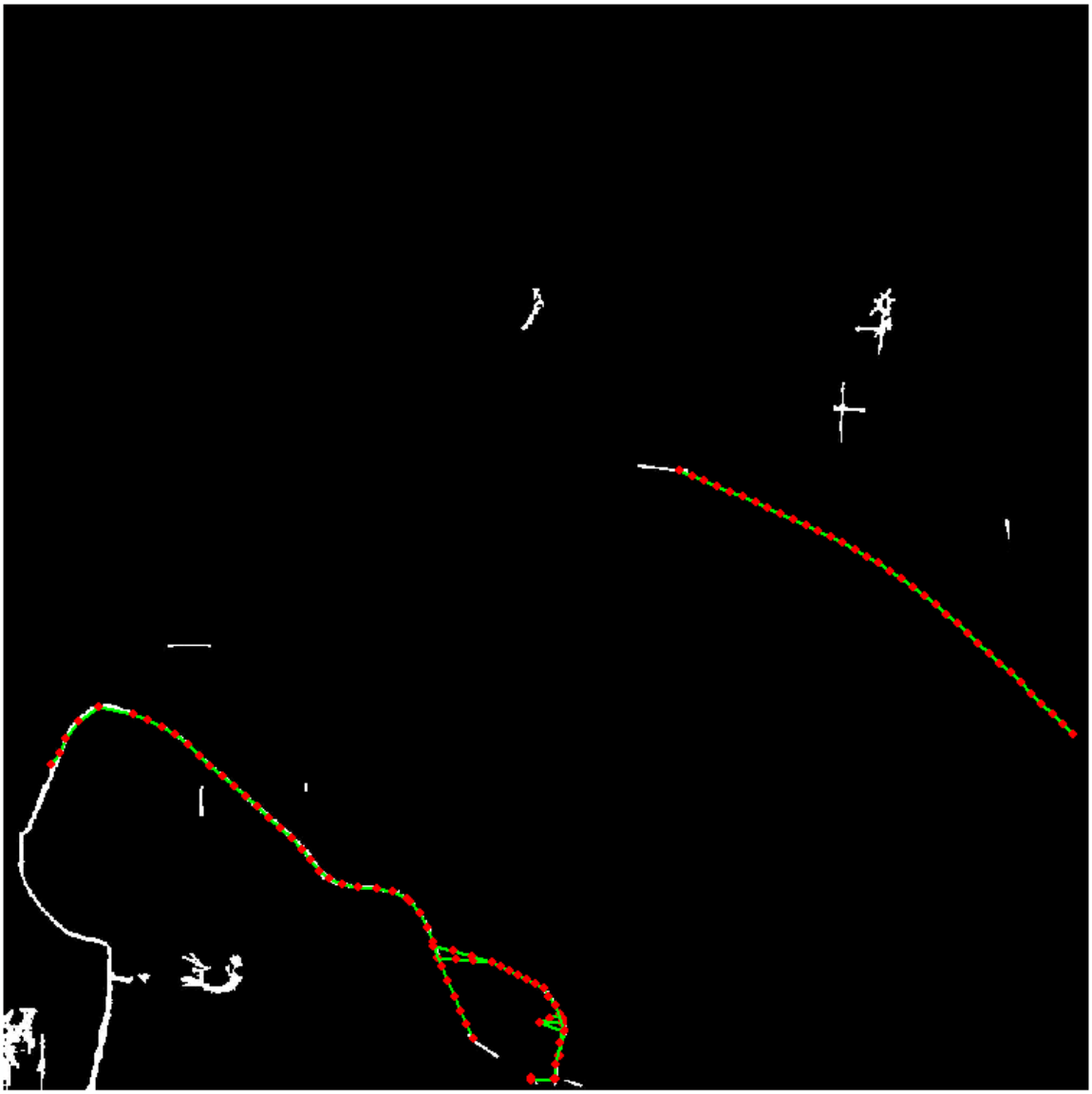
Test area Rural_06. The final conflated TIGER roads (green) with vertices (red dot) are overlaid on road imagery.



Test area Rural_07. Original TIGER roads (red) are overlaid on road imagery. The green dashed lines indicate the matched road intersections found by relaxation labeling algorithm.



Test area Rural_07. The rubber-sheeting adjusted TIGER roads (green) with vertices (red dot) are overlaid on road imagery.

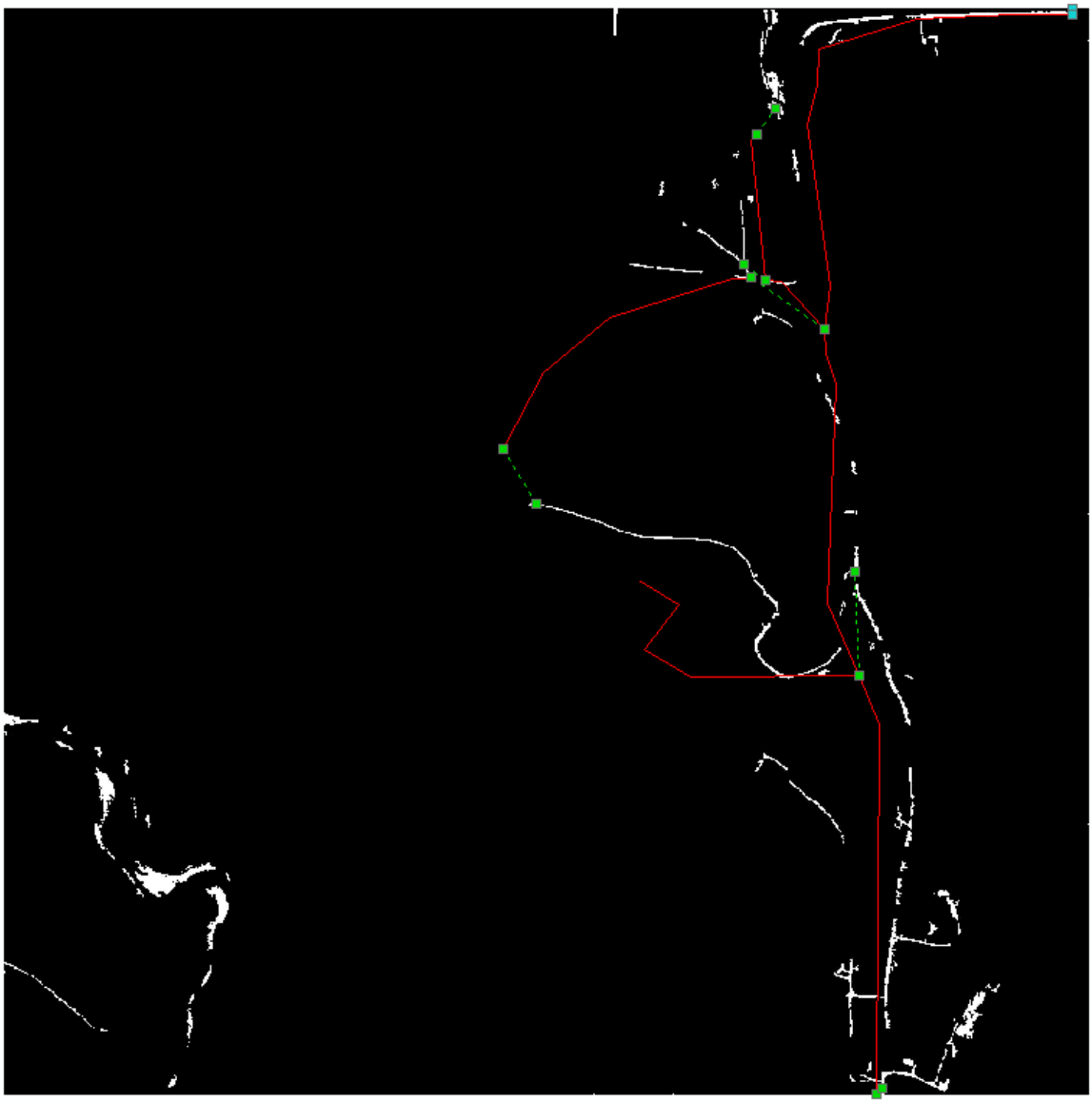


Test area Rural_07. The final conflated TIGER roads (green) with vertices (red dot) are overlaid on road imagery.

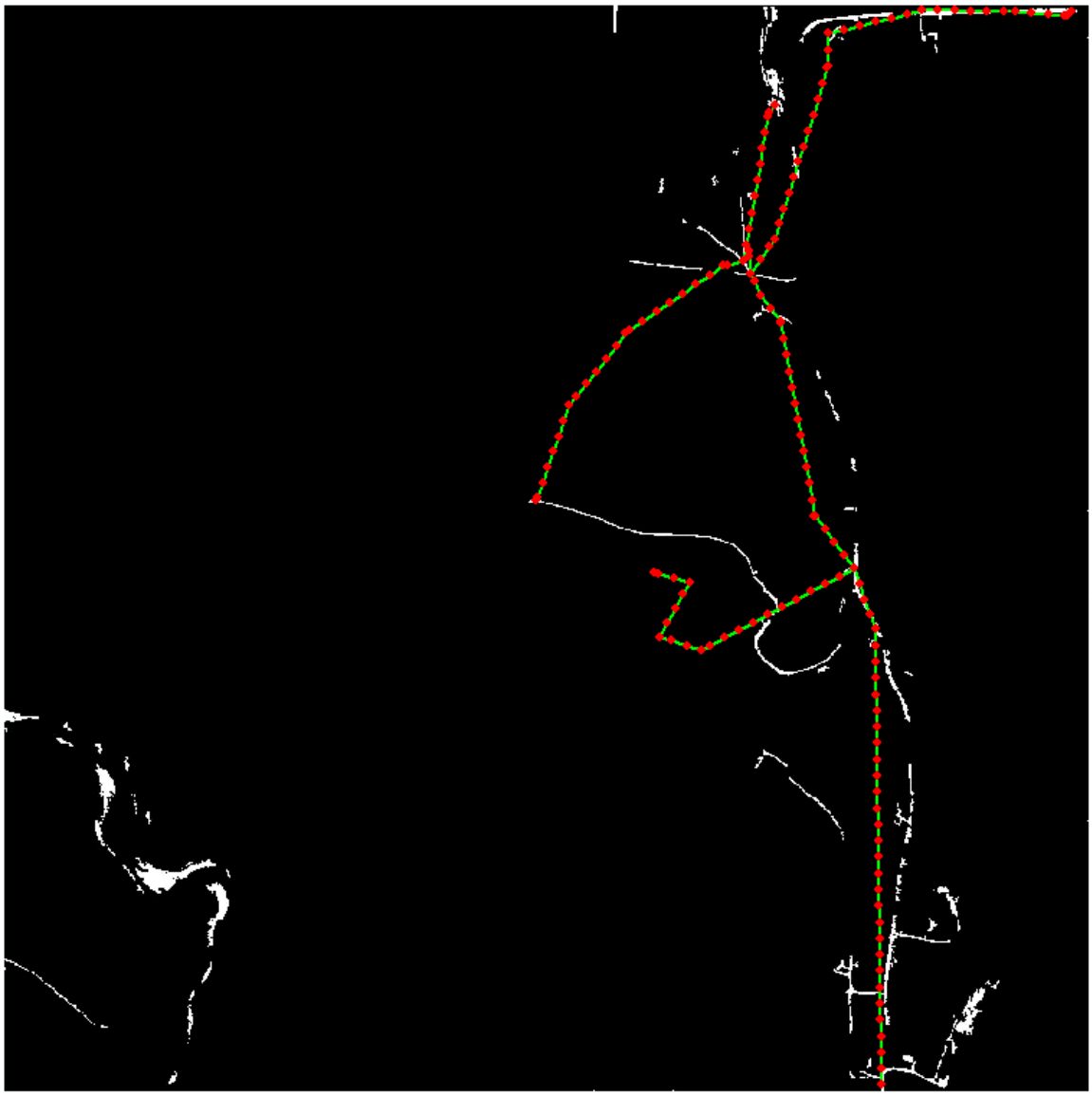
Test area Rural_08. Original TIGER roads (red) are overlaid on road imagery. The green dashed lines indicate the matched road intersections found by relaxation labeling algorithm.

Test area Rural_08. The rubber-sheeting adjusted TIGER roads (green) with vertices (red dot) are overlaid on road imagery.

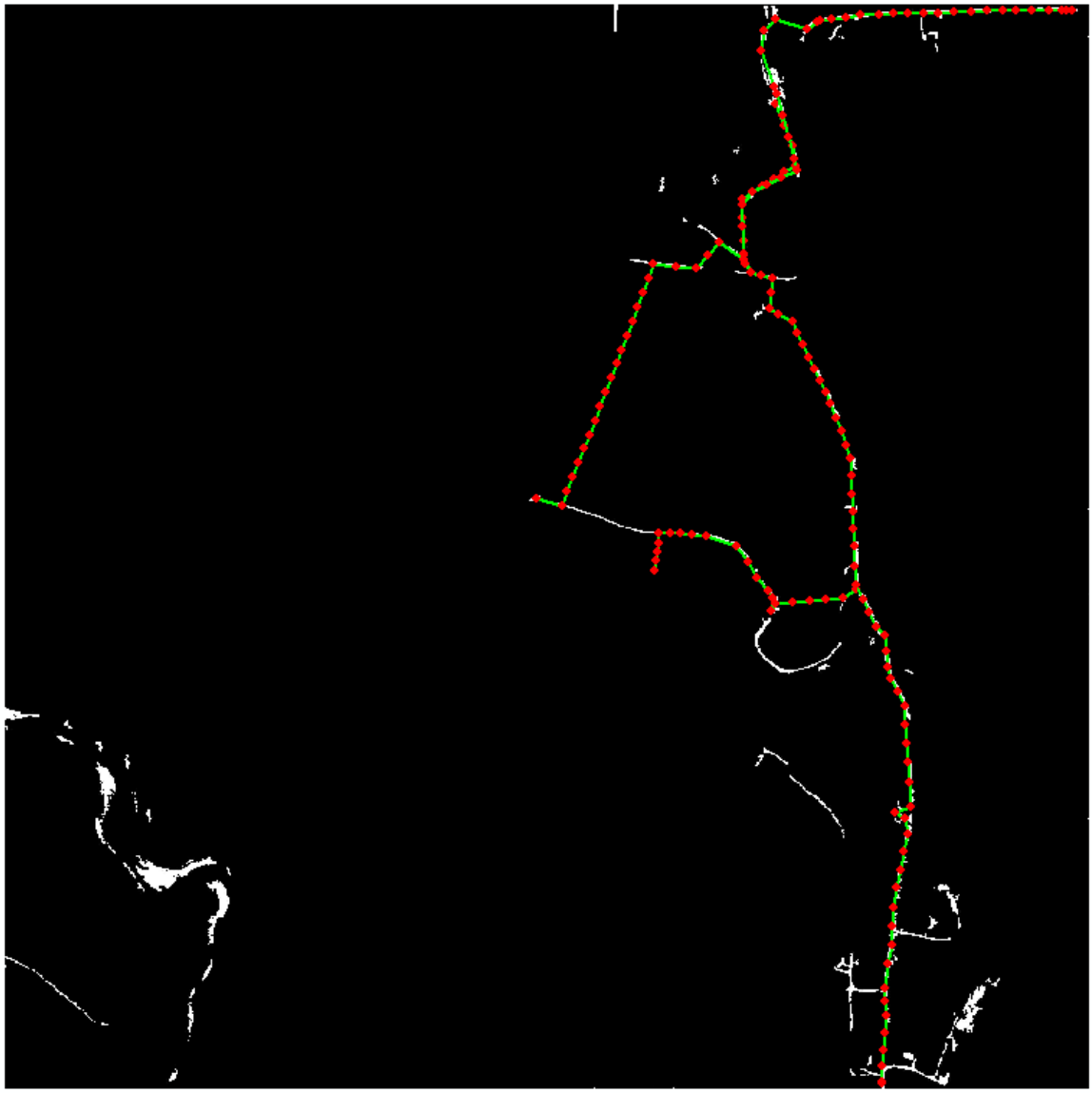
Test area Rural_08. The final conflated TIGER roads (green) with vertices (red dot) are overlaid on road imagery.



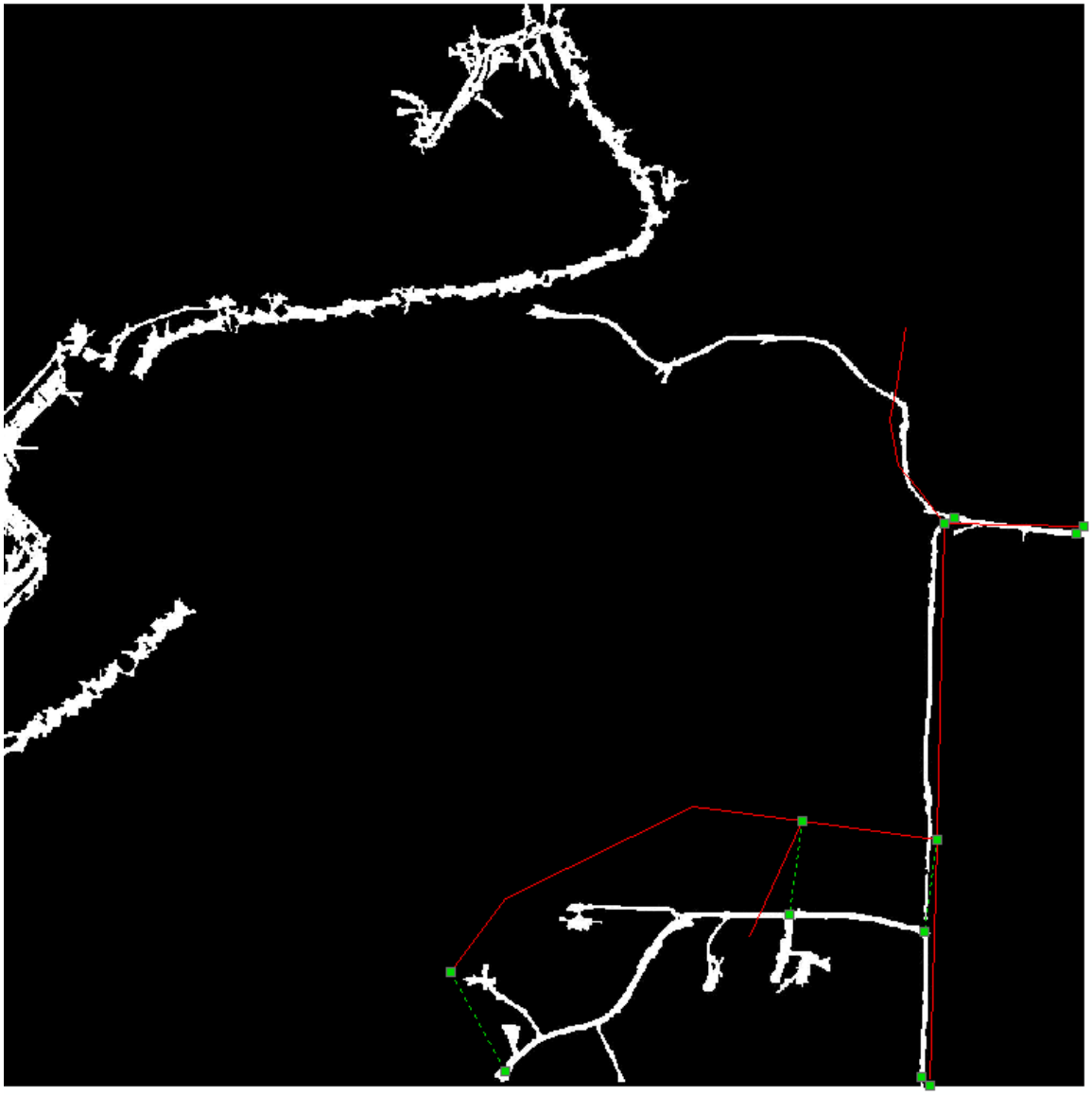
Test area Rural_09. Original TIGER roads (red) are overlaid on road imagery. The green dashed lines indicate the matched road intersections found by relaxation labeling algorithm.



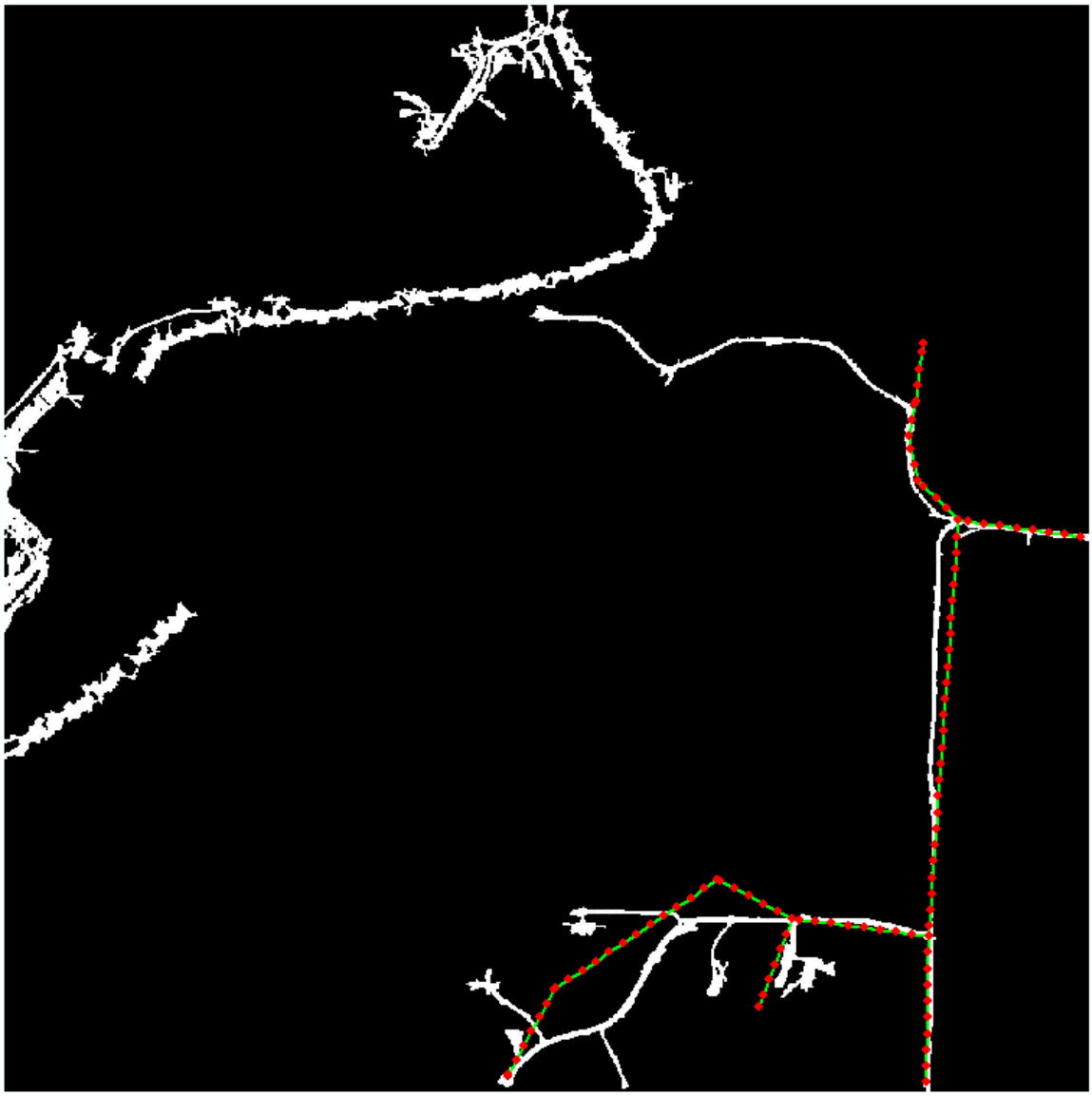
Test area Rural_09. The rubber-sheeting adjusted TIGER roads (green) with vertices (red dot) are overlaid on road imagery.



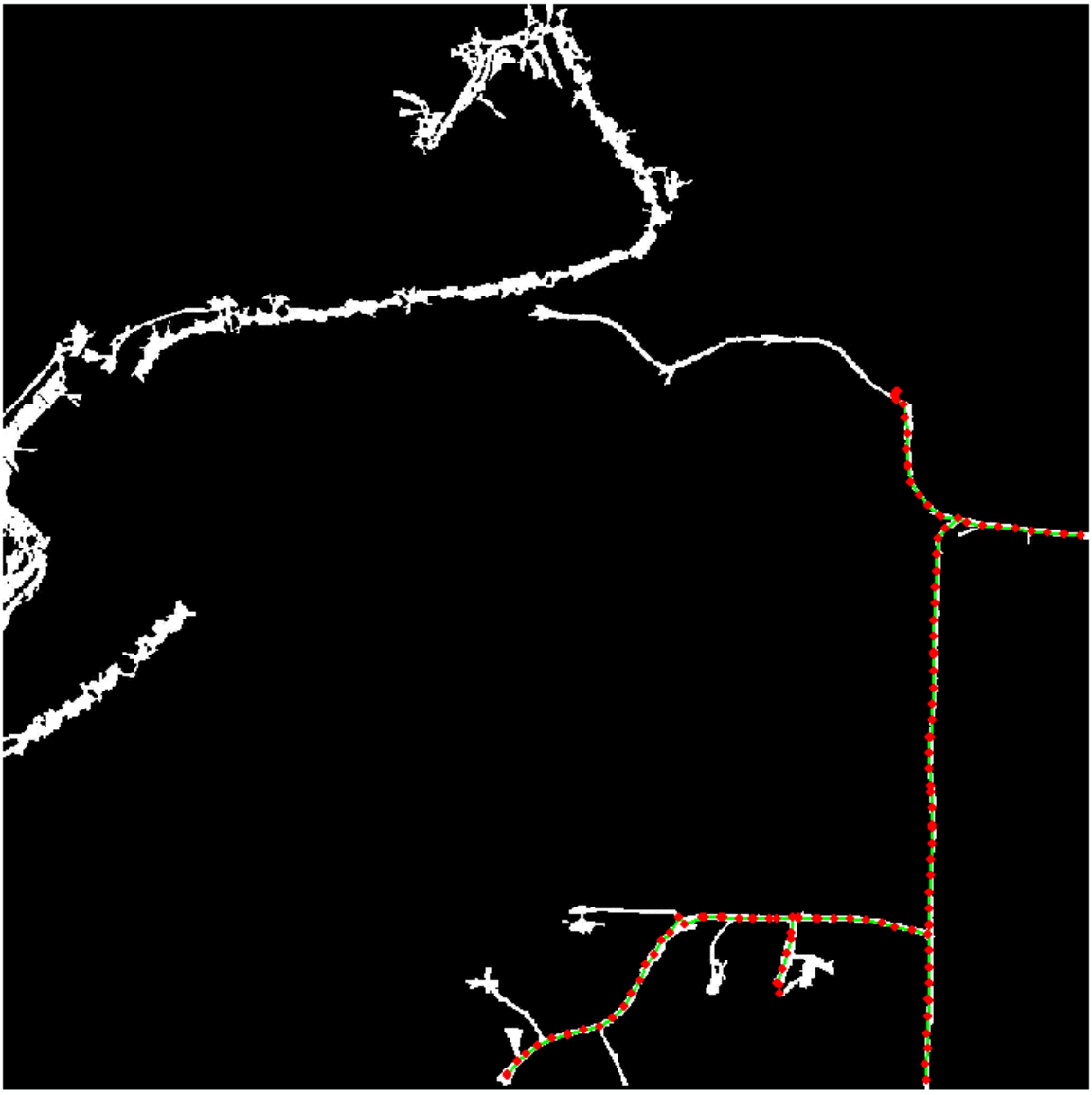
Test area Rural_09. The final conflated TIGER roads (green) with vertices (red dot) are overlaid on road imagery.



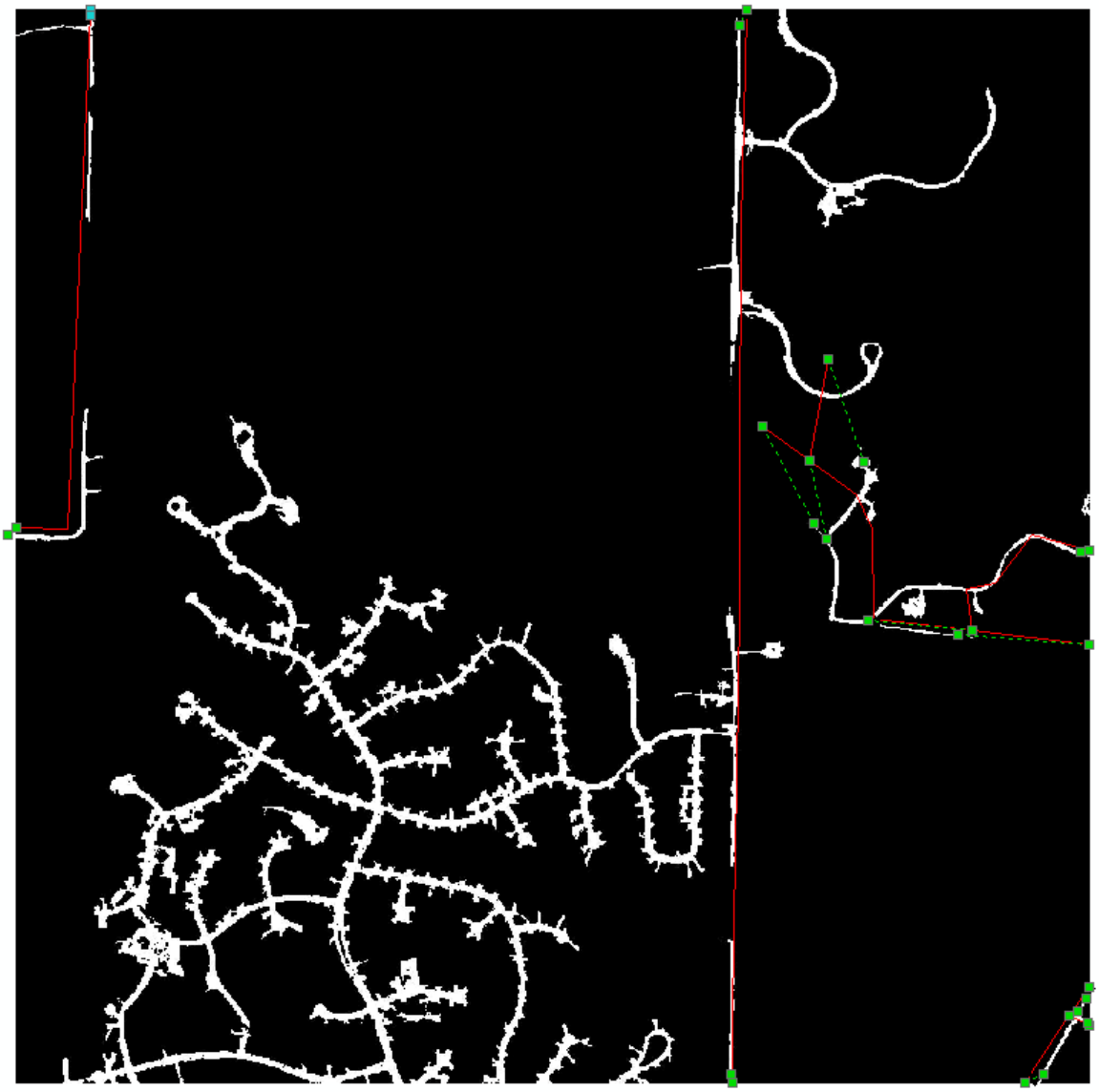
Test area Urban_01. Original TIGER roads (red) are overlaid on road imagery. The green dashed lines indicate the matched road intersections found by relaxation labeling algorithm.



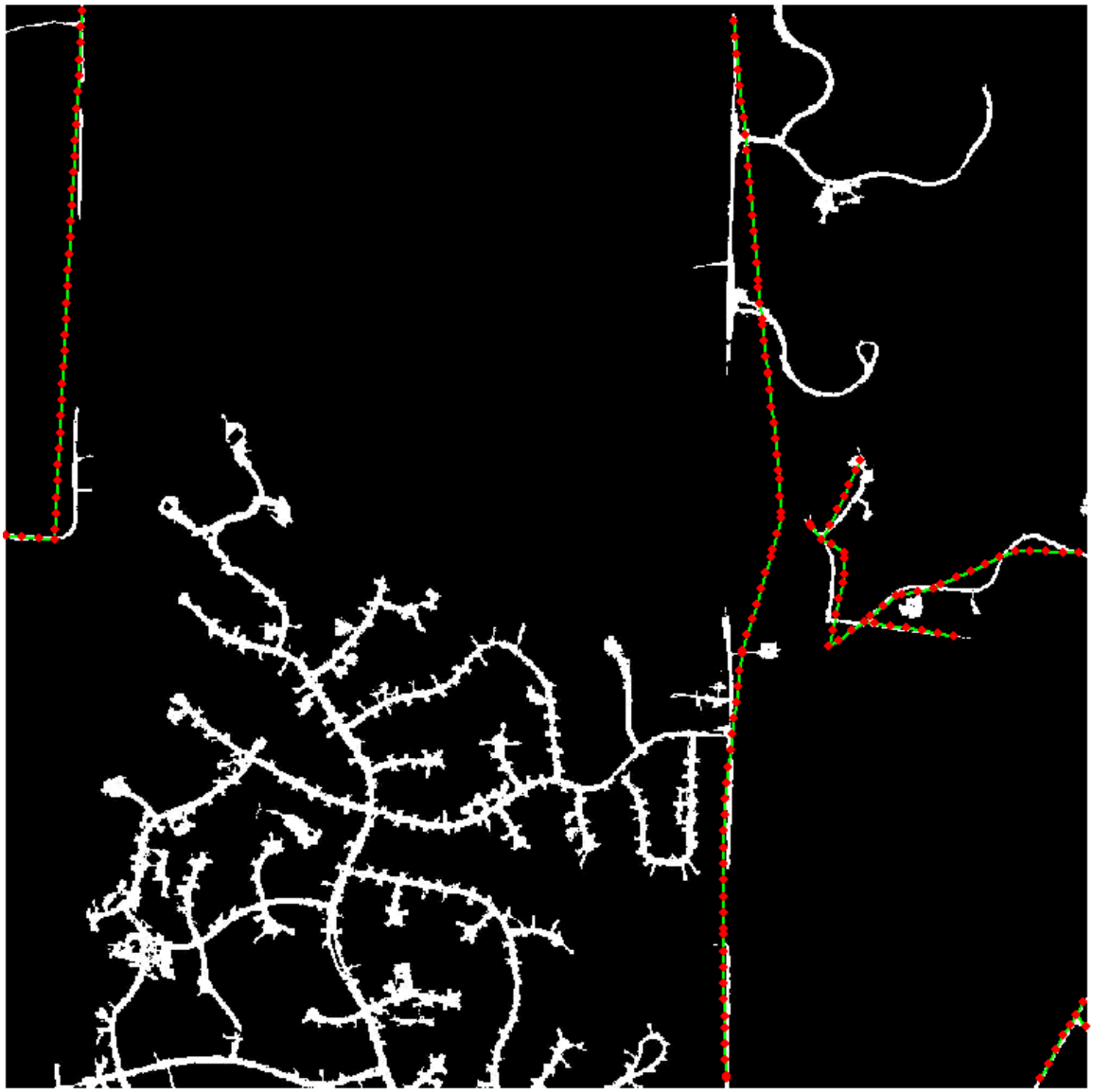
Test area Urban_01. The rubber-sheeting adjusted TIGER roads (green) with vertices (red dot) are overlaid on road imagery.



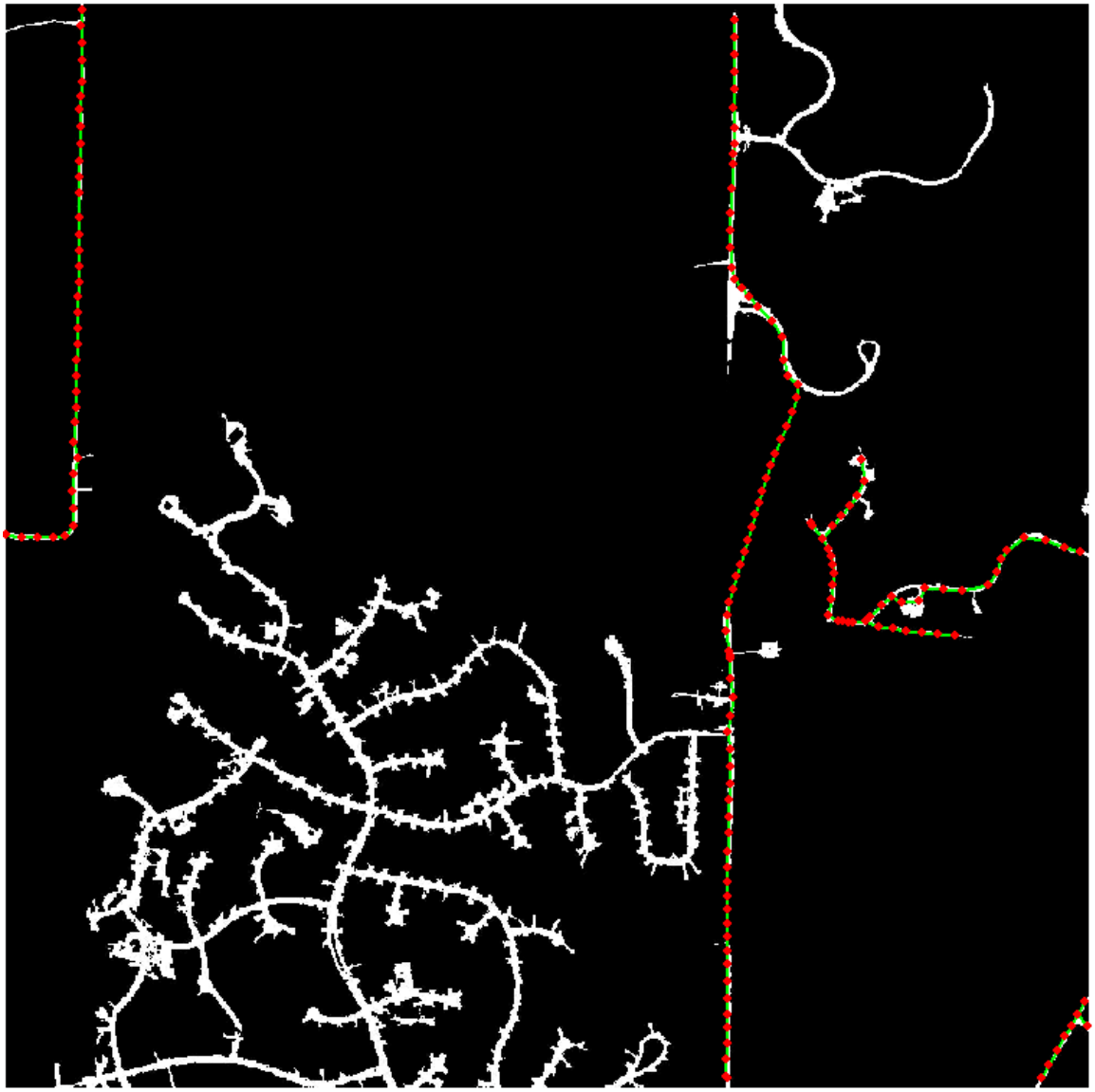
Test area Urban_01. The final conflated TIGER roads (green) with vertices (red dot) are overlaid on road imagery.



Test area Urban_02. Original TIGER roads (red) are overlaid on road imagery. The green dashed lines indicate the matched road intersections found by relaxation labeling algorithm.



Test area Urban_02. The rubber-sheeting adjusted TIGER roads (green) with vertices (red dot) are overlaid on road imagery.

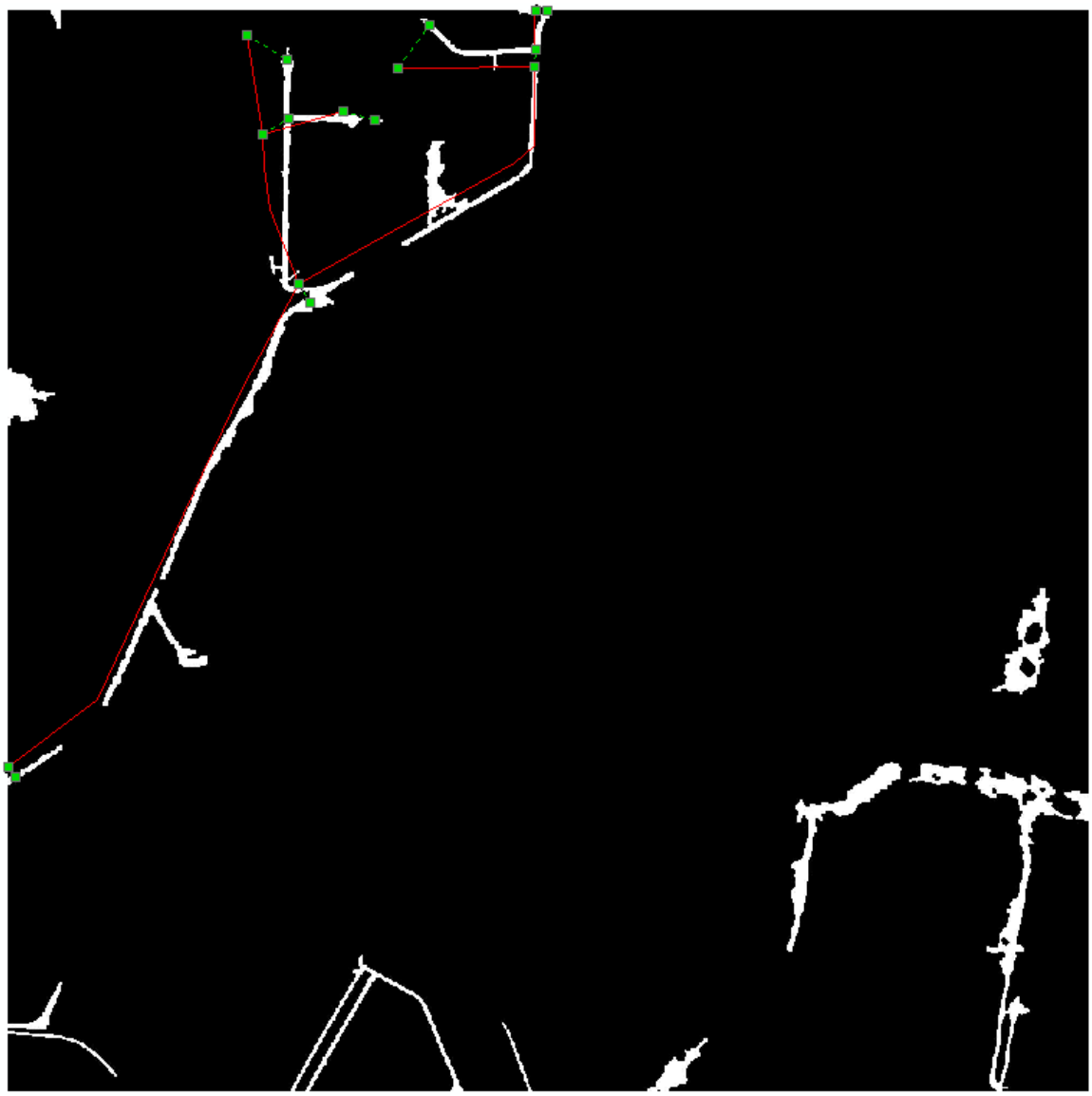


Test area Urban_02. The final conflated TIGER roads (green) with vertices (red dot) are overlaid on road imagery.

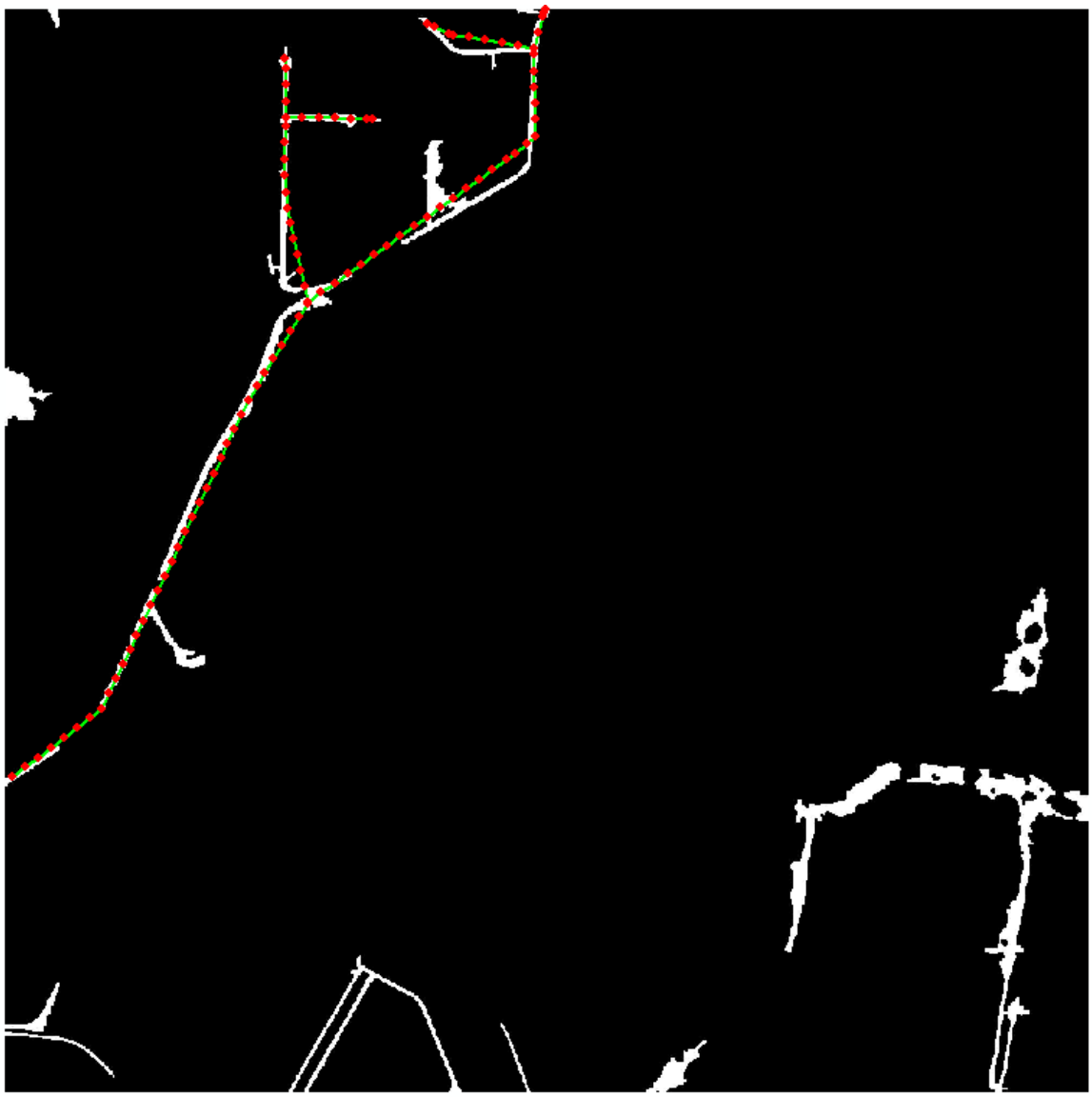
Test area Urban_03. Original TIGER roads (red) are overlaid on road imagery. The green dashed lines indicate the matched road intersections found by relaxation labeling algorithm.

Test area Urban_03. The rubber-sheeting adjusted TIGER roads (green) with vertices (red dot) are overlaid on road imagery.

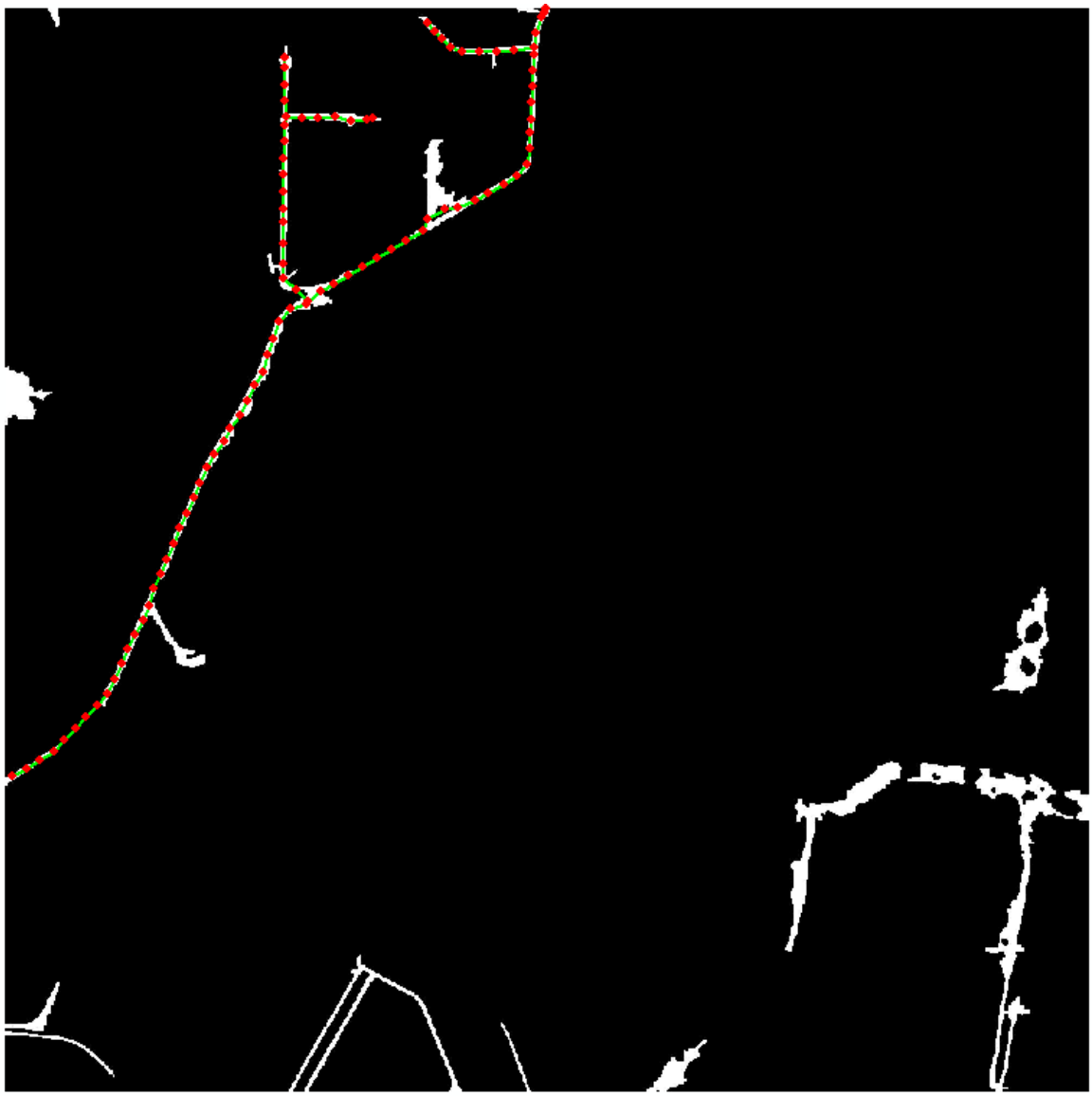
Test area Urban_03. The final conflated TIGER roads (green) with vertices (red dot) are overlaid on road imagery.



Test area Urban_04. Original TIGER roads (red) are overlaid on road imagery. The green dashed lines indicate the matched road intersections found by relaxation labeling algorithm.

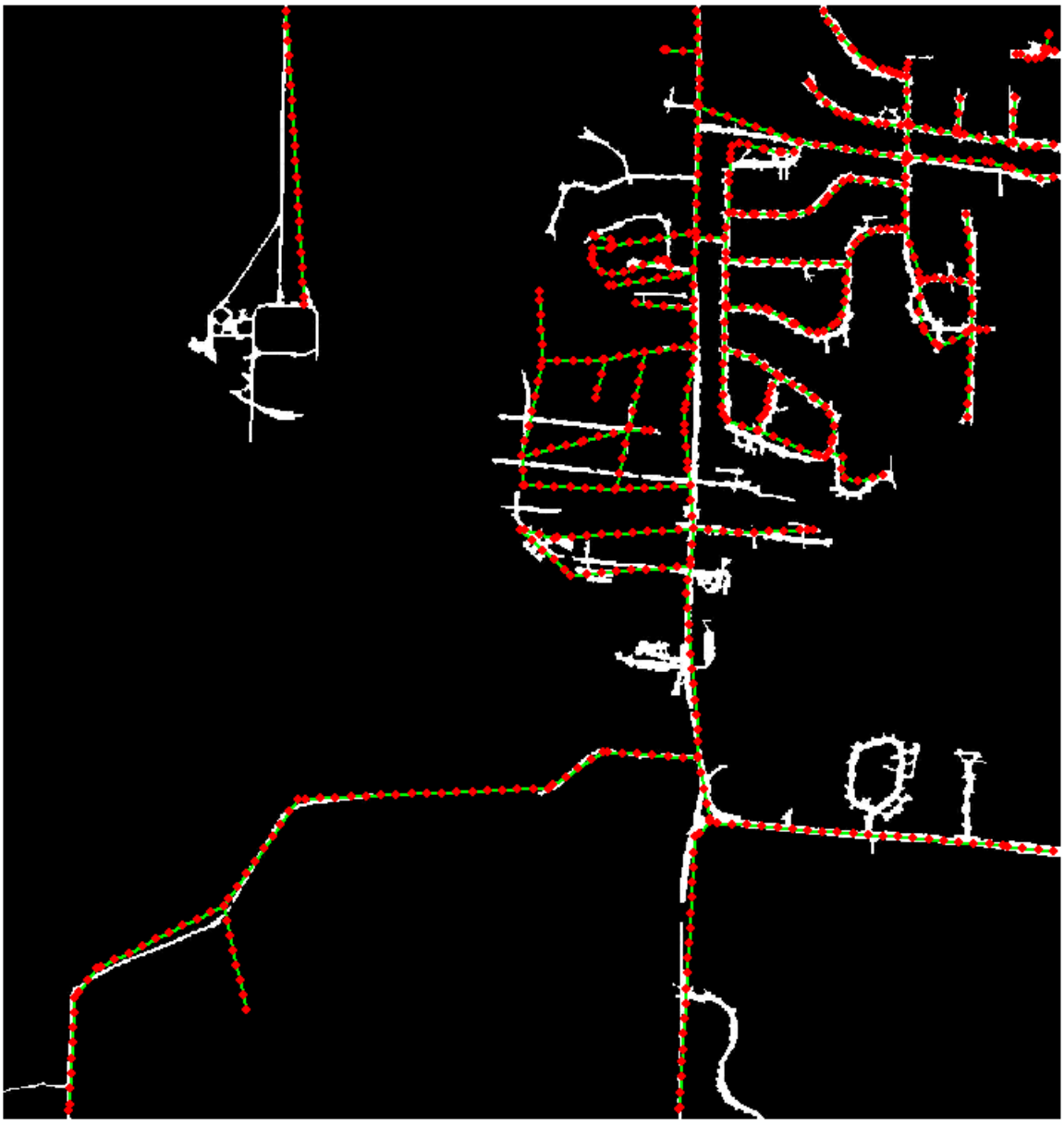


Test area Urban_04. The rubber-sheeting adjusted TIGER roads (green) with vertices (red dot) are overlaid on road imagery.



Test area Urban_04. The final conflated TIGER roads (green) with vertices (red dot) are overlaid on road imagery.

Test area Urban_05. Original TIGER roads (red) are overlaid on road imagery. The green dashed lines indicate the matched road intersections found by relaxation labeling algorithm.



Test area Urban_05. The rubber-sheeting adjusted TIGER roads (green) with vertices (red dot) are overlaid on road imagery.

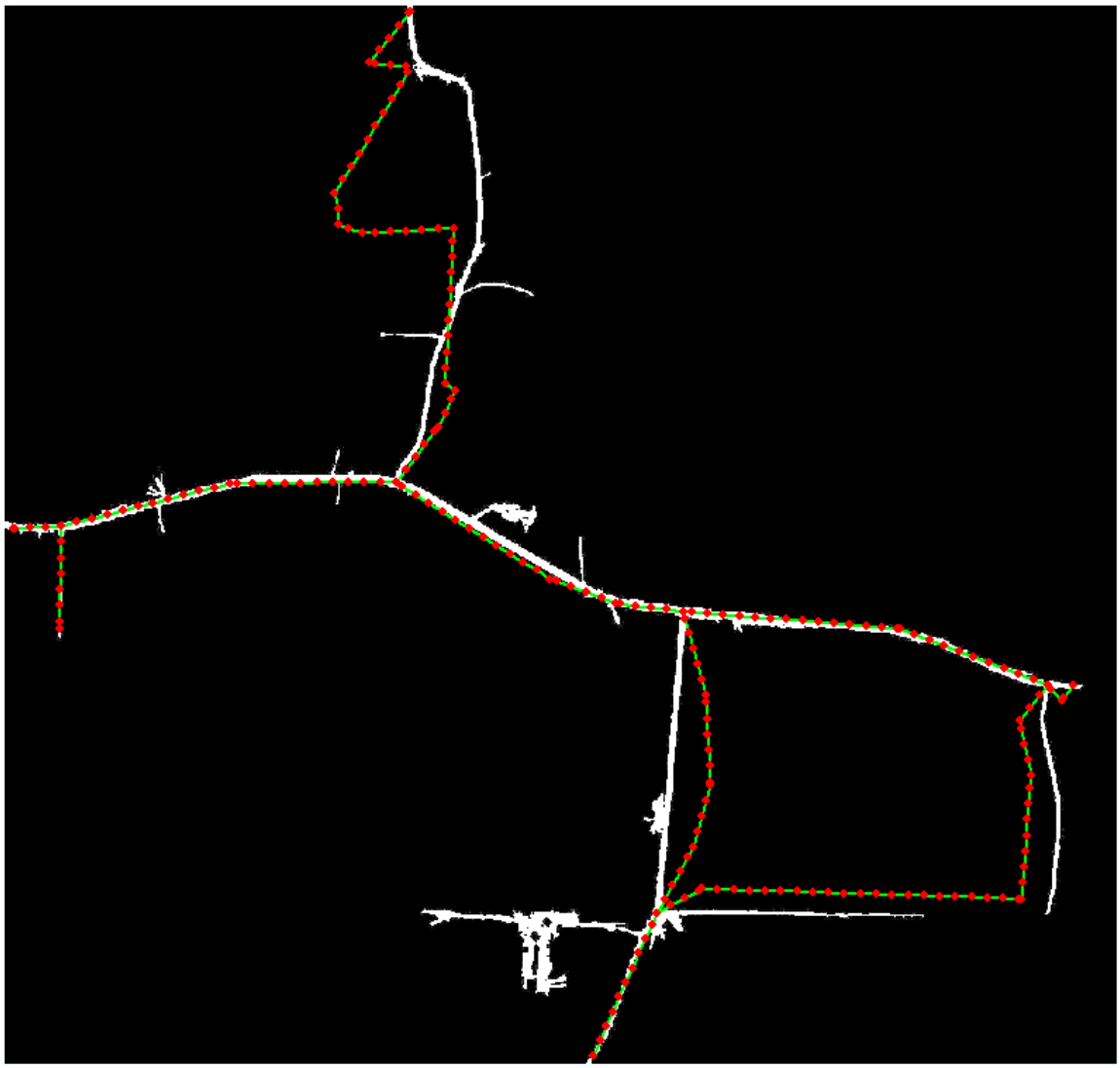
Test area Urban_05. The final conflated TIGER roads (green) with vertices (red dot) are overlaid on road imagery.

Test area Urban_06. Original TIGER roads (red) are overlaid on road imagery. The green dashed lines indicate the matched road intersections found by relaxation labeling algorithm.

Test area Urban_06. The rubber-sheeting adjusted TIGER roads (green) with vertices (red dot) are overlaid on road imagery.

Test area Urban_06. The final conflated TIGER roads (green) with vertices (red dot) are overlaid on road imagery.

Test area Urban_07. Original TIGER roads (red) are overlaid on road imagery. The green dashed lines indicate the matched road intersections found by relaxation labeling algorithm.



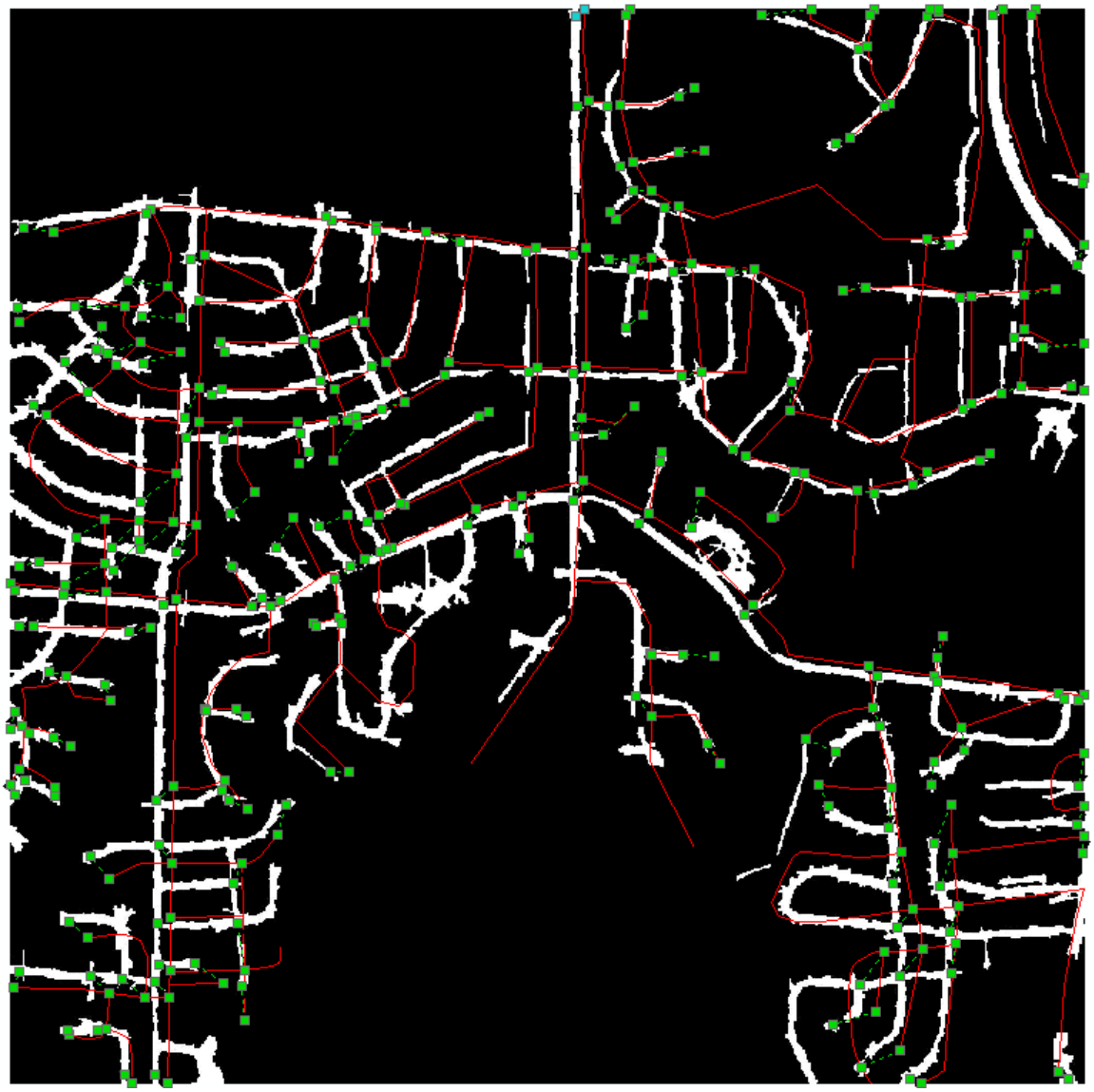
Test area Urban_07. The rubber-sheeting adjusted TIGER roads (green) with vertices (red dot) are overlaid on road imagery.

Test area Urban_07. The final conflated TIGER roads (green) with vertices (red dot) are overlaid on road imagery.

Test area Urban_08. Original TIGER roads (red) are overlaid on road imagery. The green dashed lines indicate the matched road intersections found by relaxation labeling algorithm.

Test area Urban_08. The rubber-sheeting adjusted TIGER roads (green) with vertices (red dot) are overlaid on road imagery.

Test area Urban_08. The final conflated TIGER roads (green) with vertices (red dot) are overlaid on road imagery.



Test area Urban_09. Original TIGER roads (red) are overlaid on road imagery. The green dashed lines indicate the matched road intersections found by relaxation labeling algorithm.

Test area Urban_09. The rubber-sheeting adjusted TIGER roads (green) with vertices (red dot) are overlaid on road imagery.

Test area Urban_09. The final conflated TIGER roads (green) with vertices (red dot) are overlaid on road imagery.

Appendix B Snake Experiment Results

Appendix B shows an interesting real world example used in chapter 4. Suppose we have generated clean road image, we manipulate a few key vector vertices to different extreme locations to see how the snake perform.

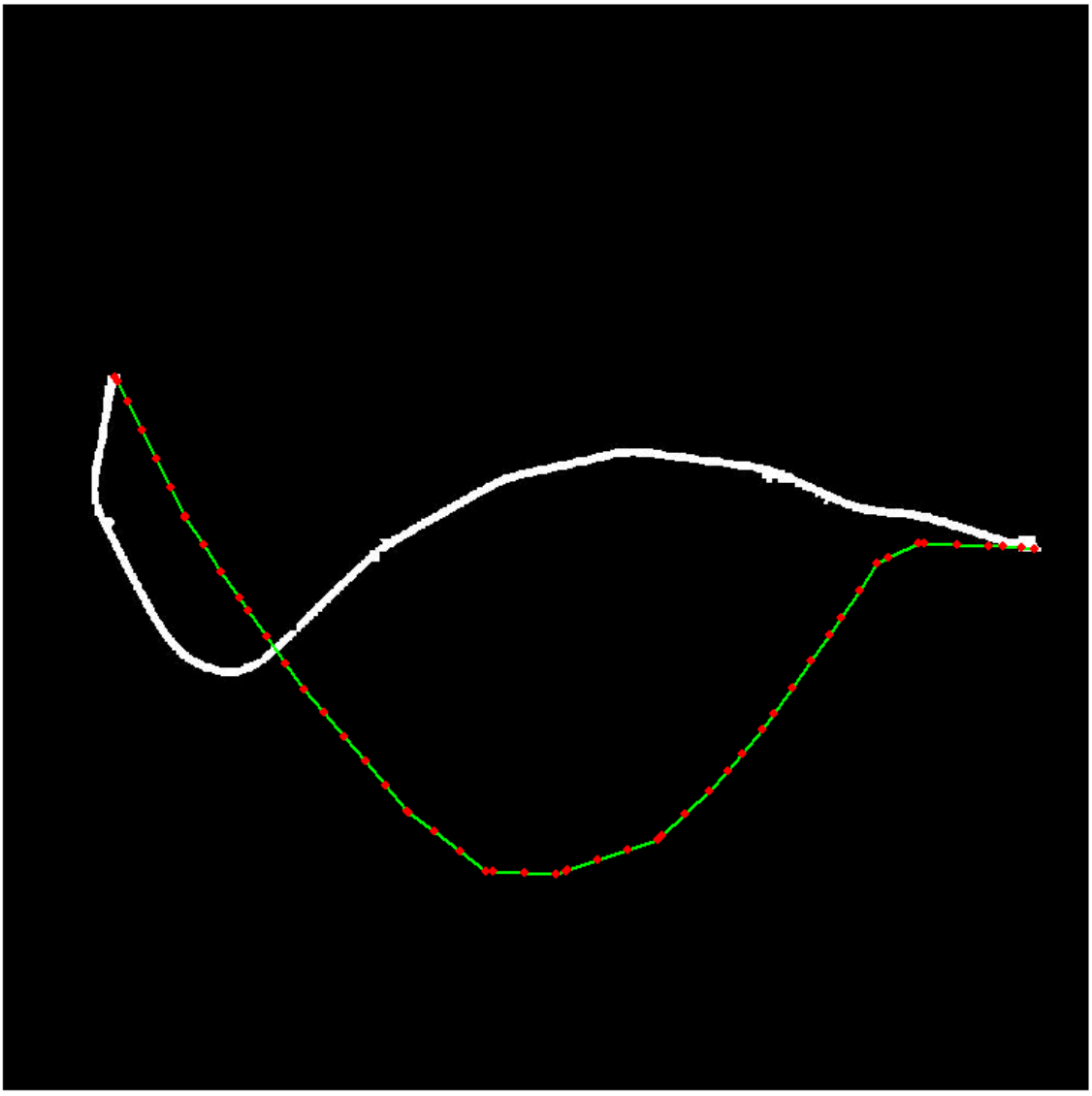


Figure 1. After point matching and rubber-sheeting transformation, the adjusted TIGER road (green) with vertices (red dot) is overlaid on road imagery. Note only the end points of road are in correct locations.

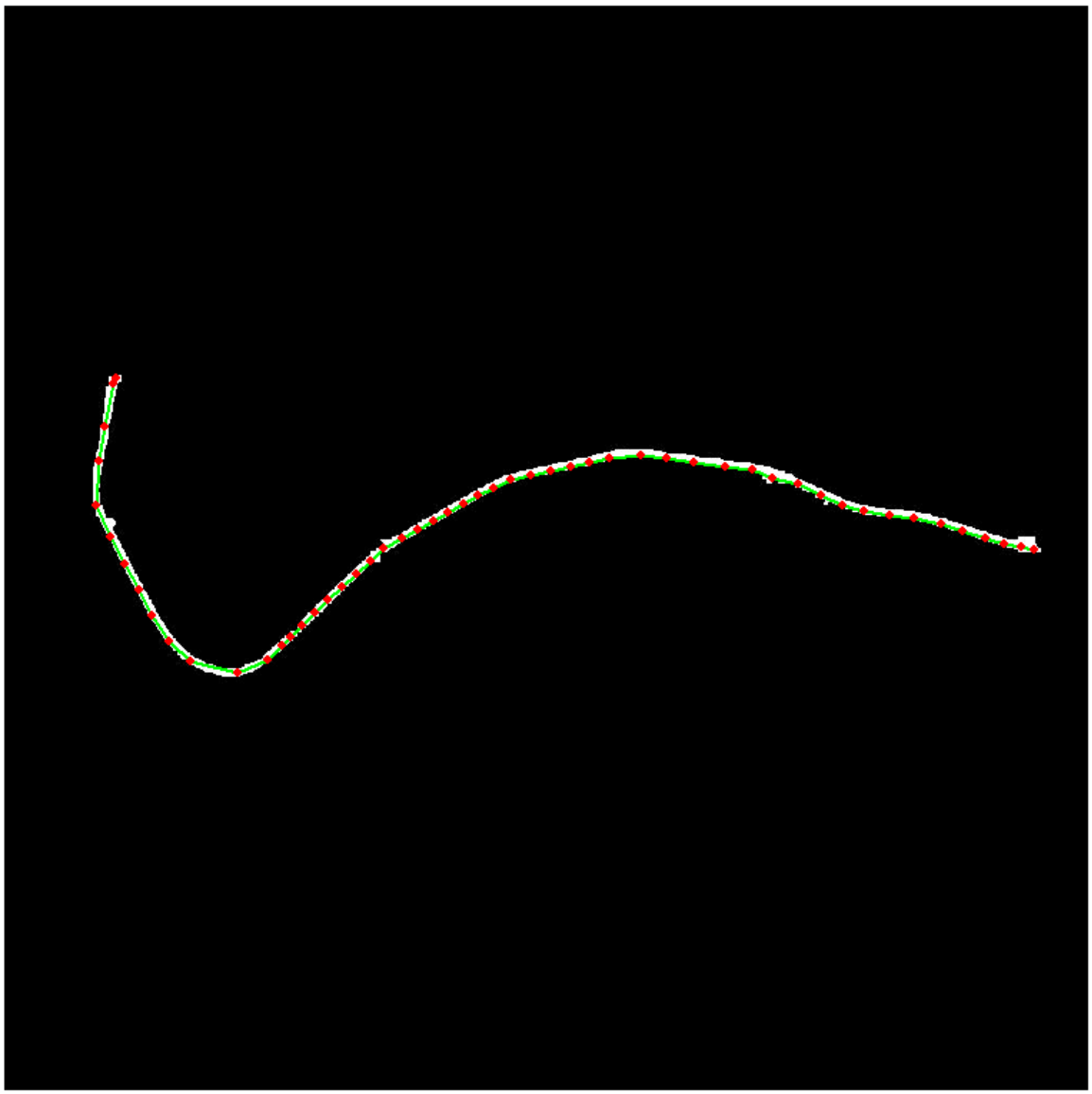


Figure 2. The snake algorithm moved all the vertices (red dot) to correct locations on road imagery.

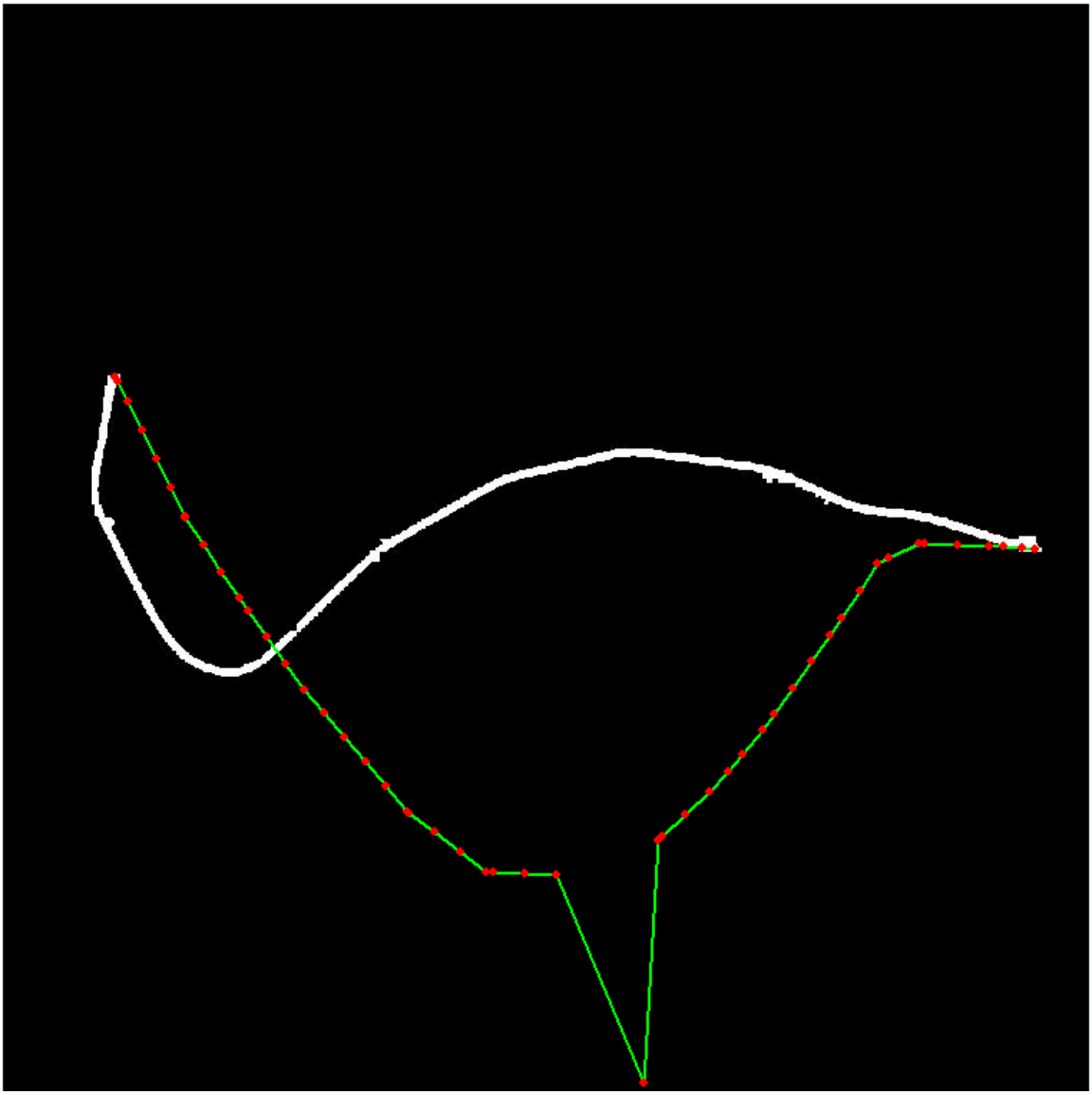


Figure 3. We manually move one intermediate vertex to extreme location near bottom of image border.

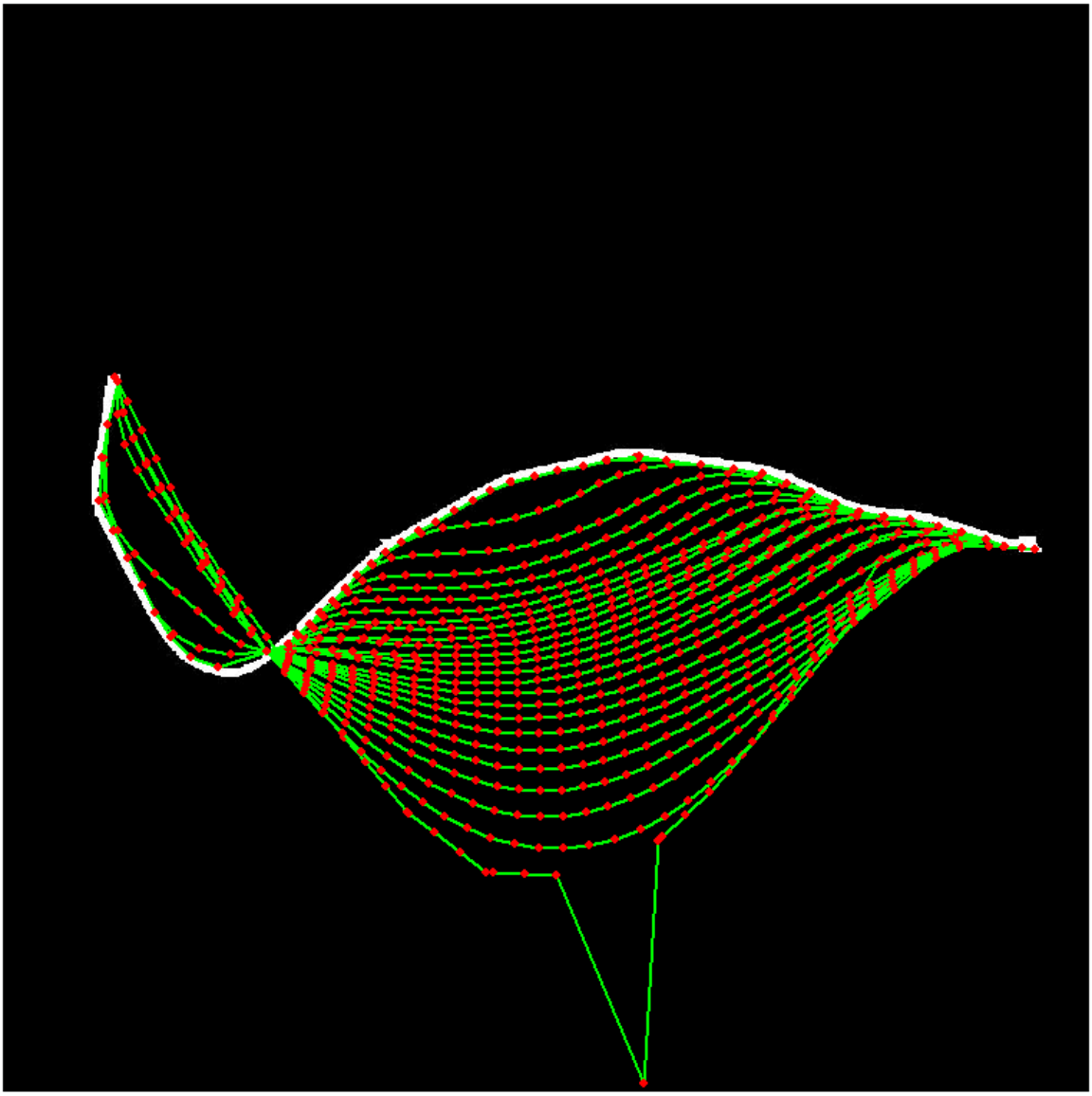


Figure 4. The movements of each snake iteration are shown and the algorithm has no problem to move the vertices to correct locations.

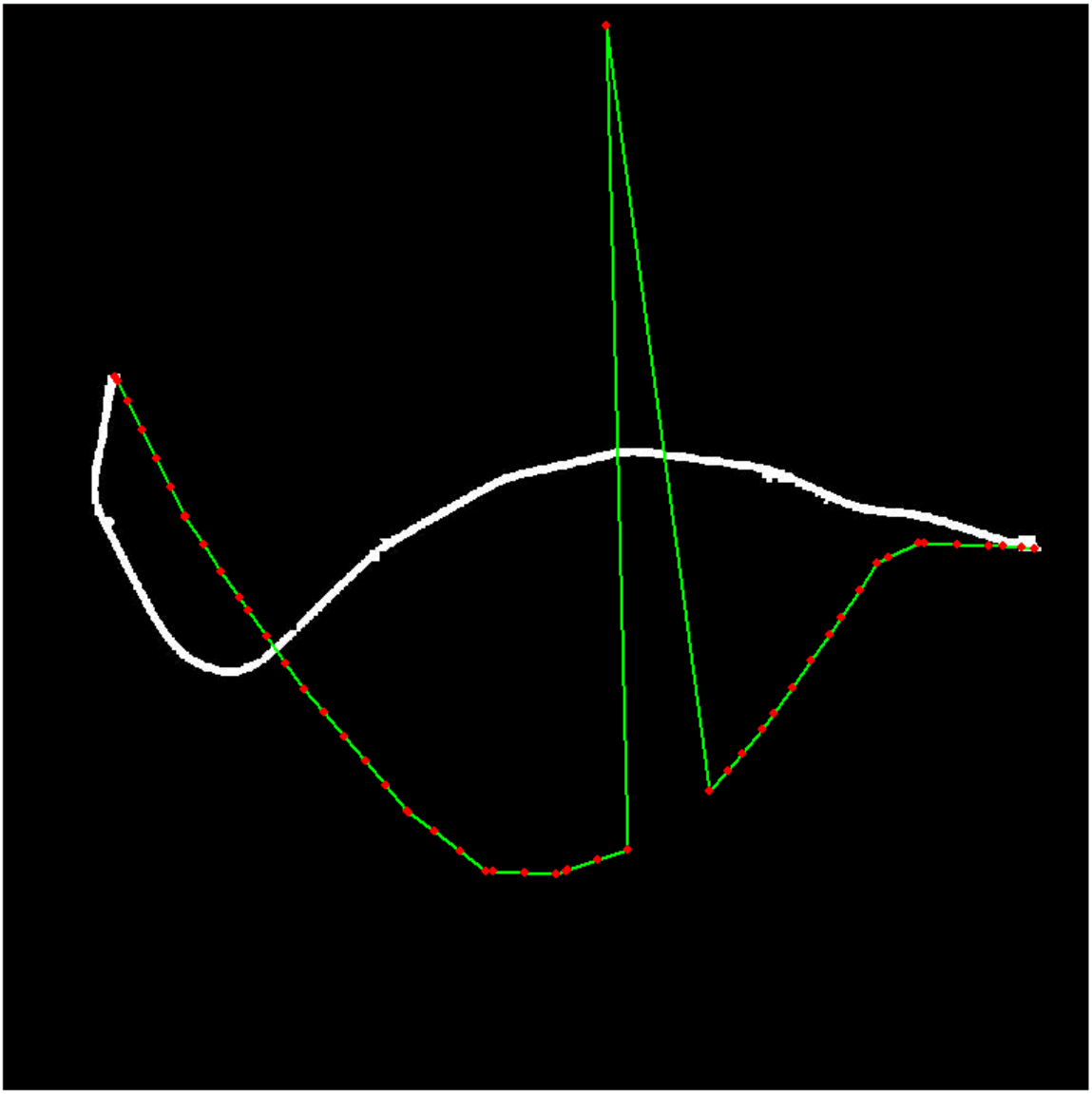


Figure 5. Similarly we manually move one intermediate vertex to extreme location near top of image border.

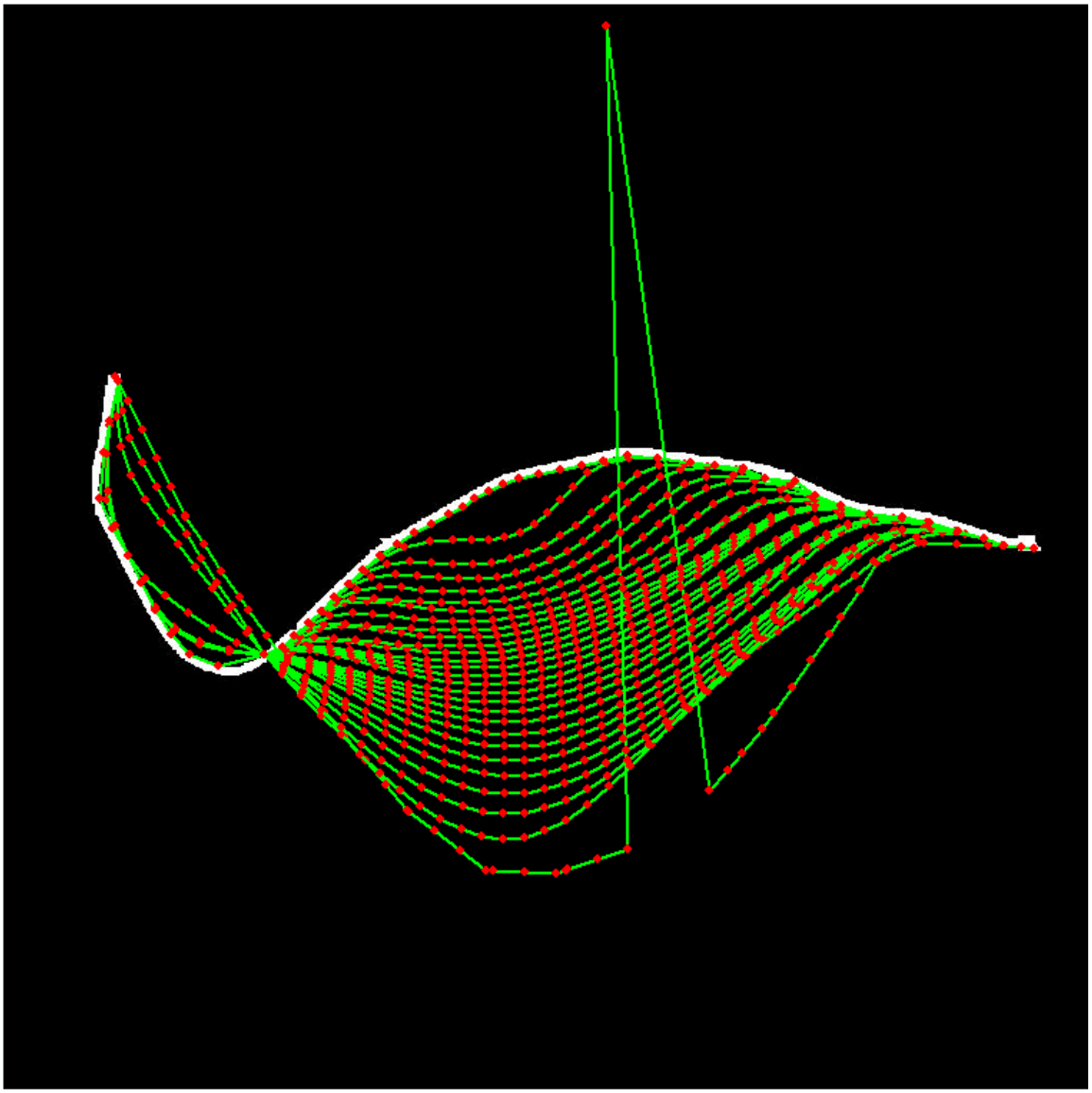


Figure 6. From the movements of each snake iteration, we can see that the algorithm still has no problem to move the vertices to correct locations.

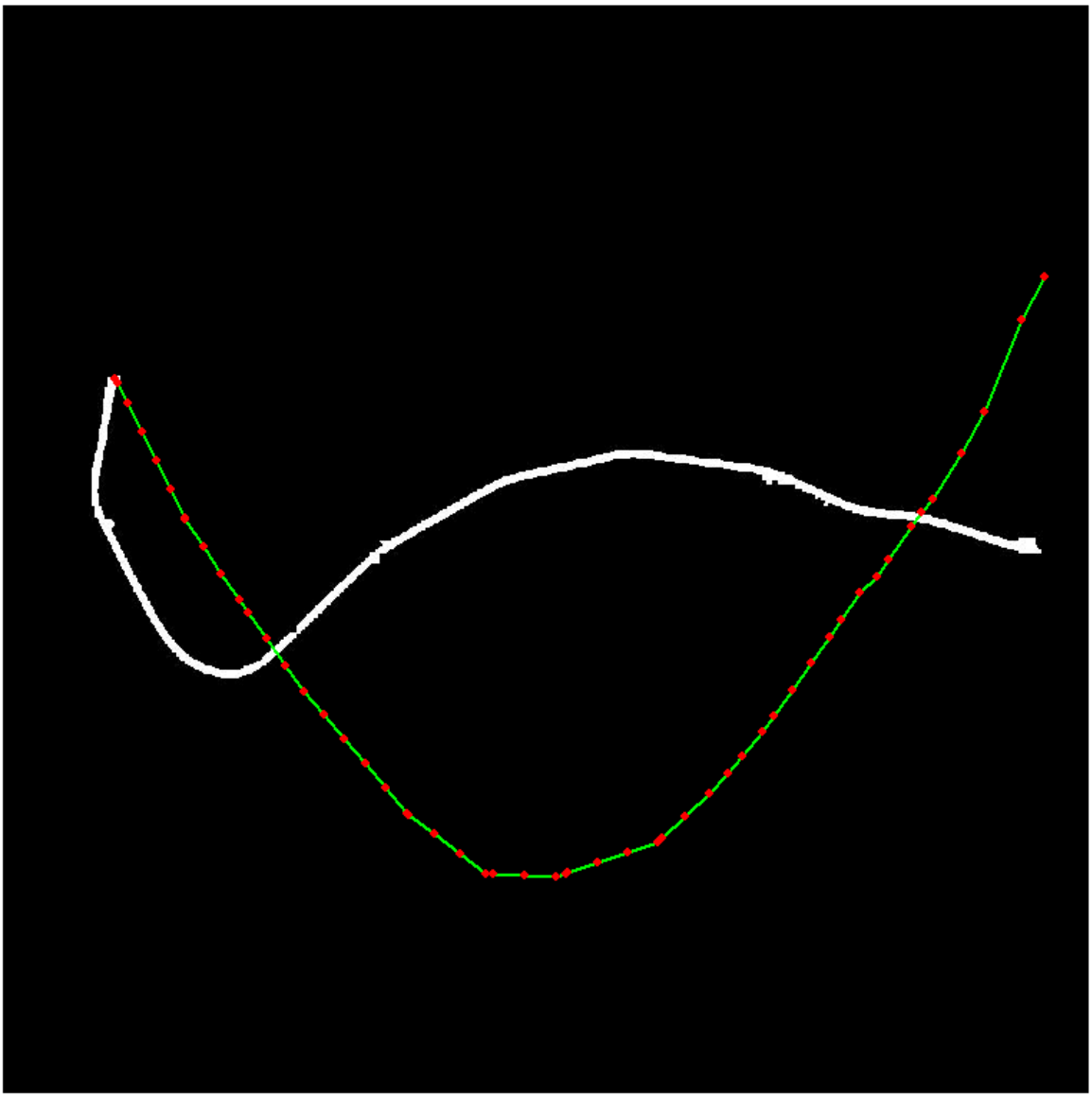


Figure 7. In previous figures, both of the end points of road are in correct locations. Now we move one end point to wrong location.

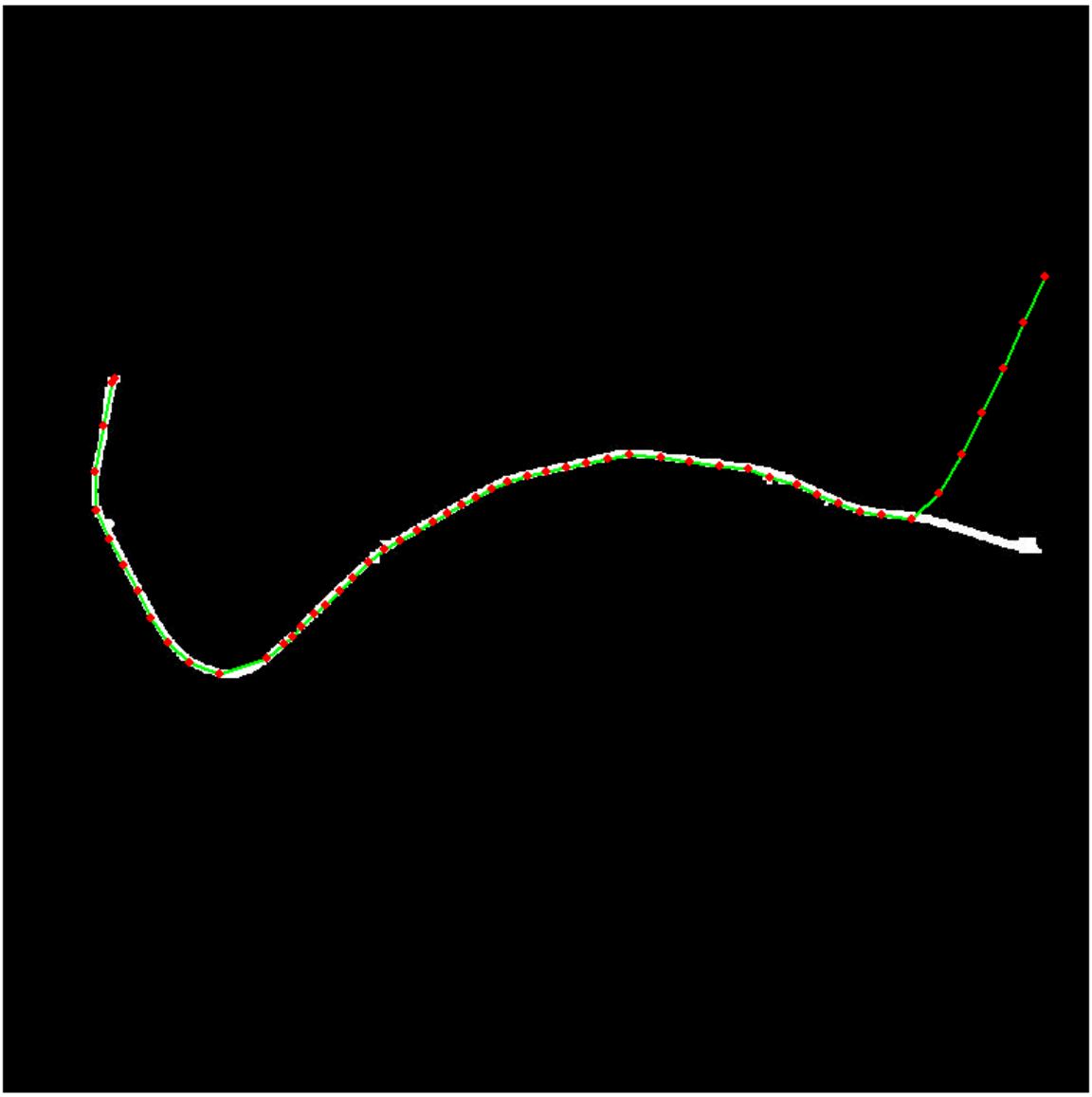


Figure 8. In our normal snake operation, we fix the end points and only move the intermediate vertices. Therefore, a few vertices near the right end are not corrected.

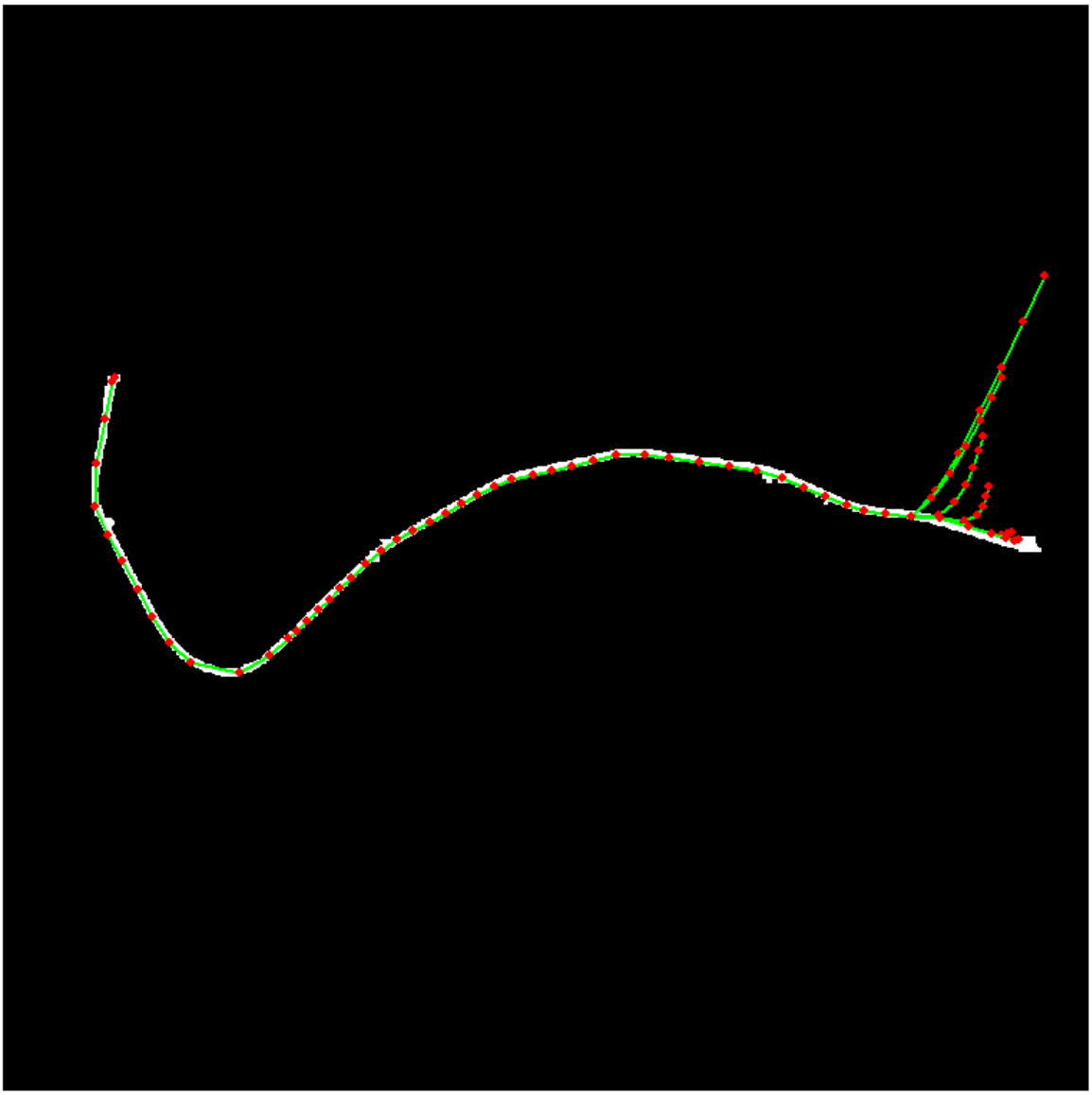


Figure 9. Another relaxed snake function is run to relax the end points and the movements are shown.

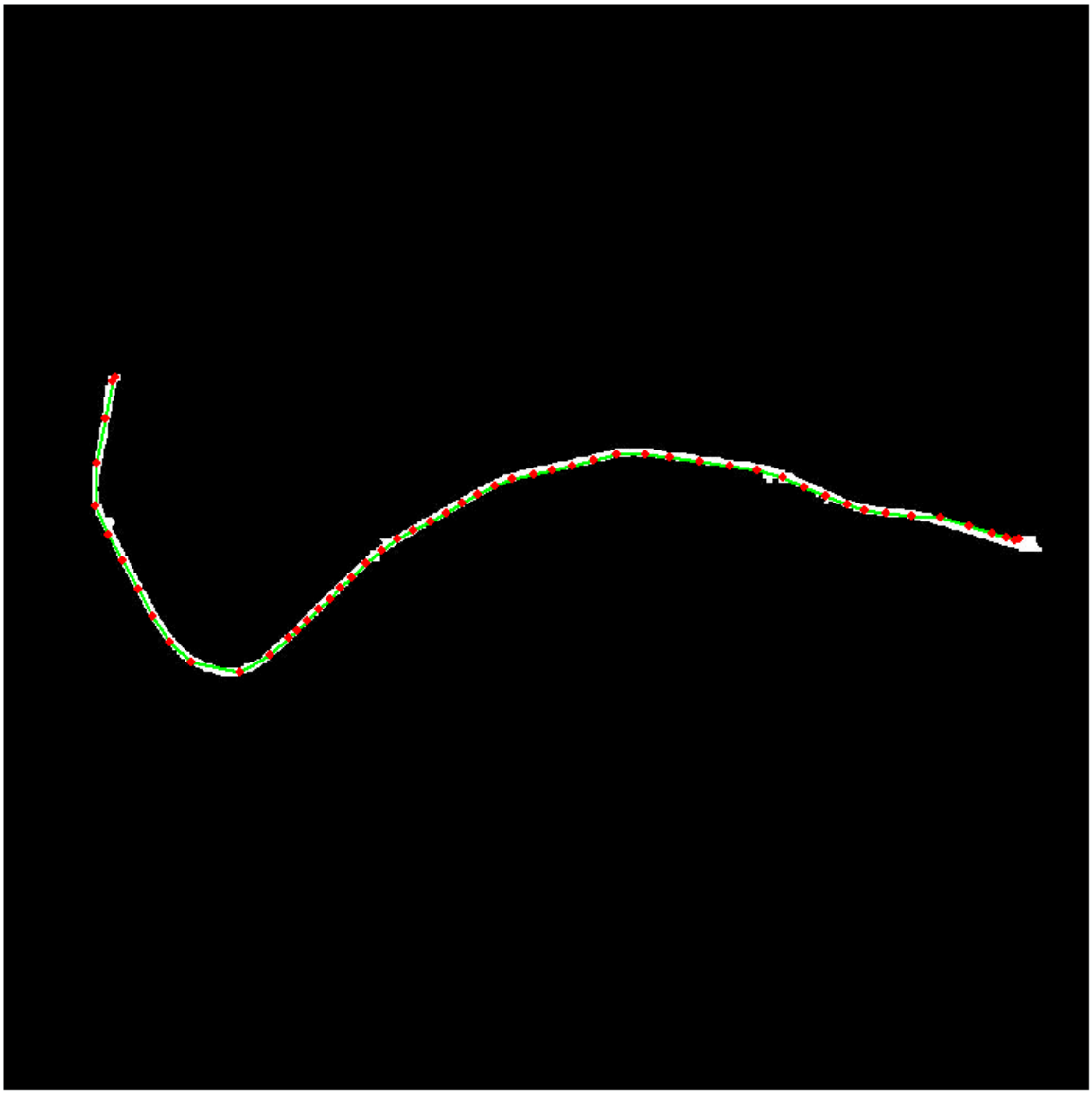


Figure 10. After normal and relaxed snake operations, all vertices are moved to correct locations.

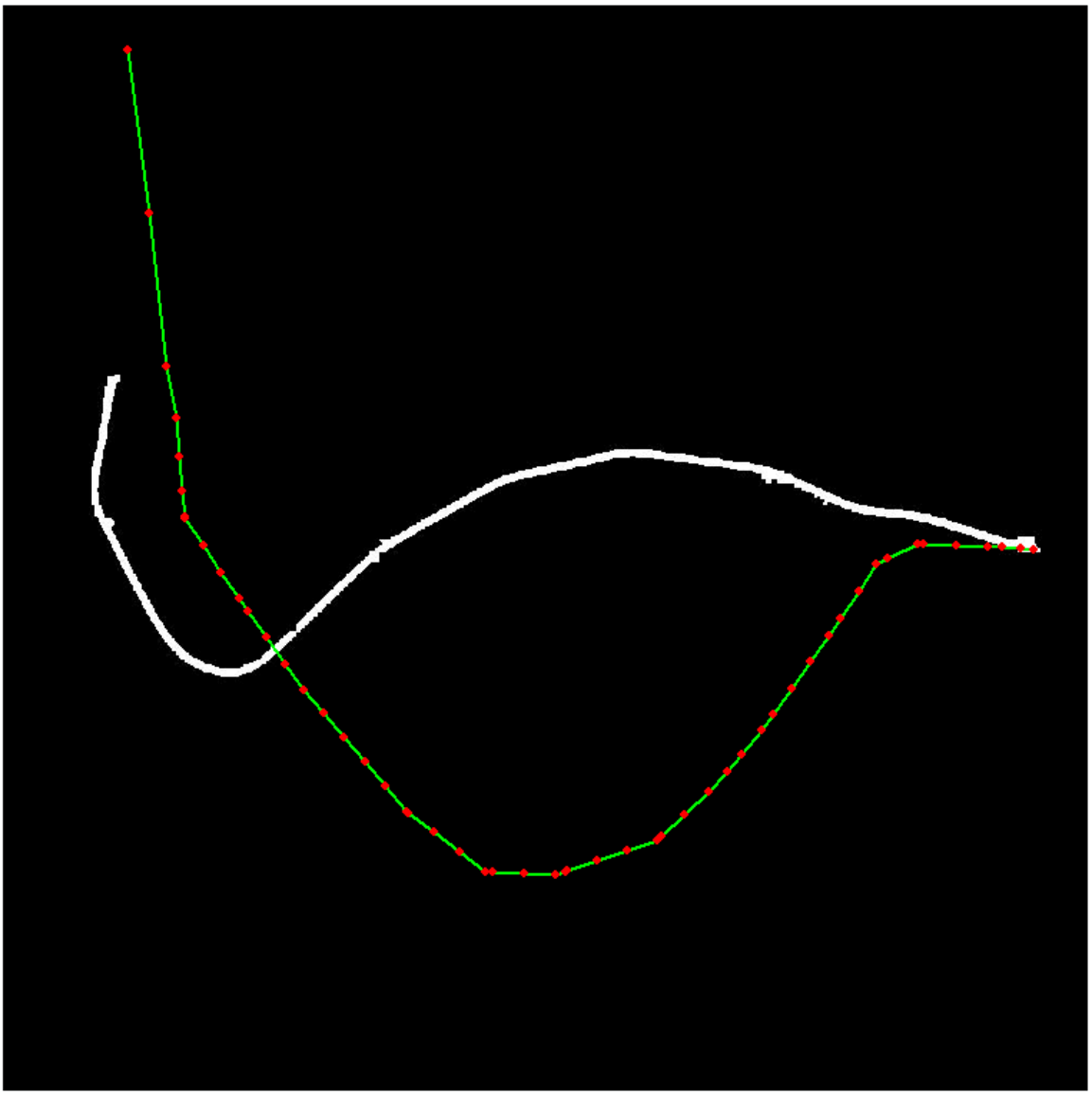


Figure 11. We move the left end point away to wrong location.

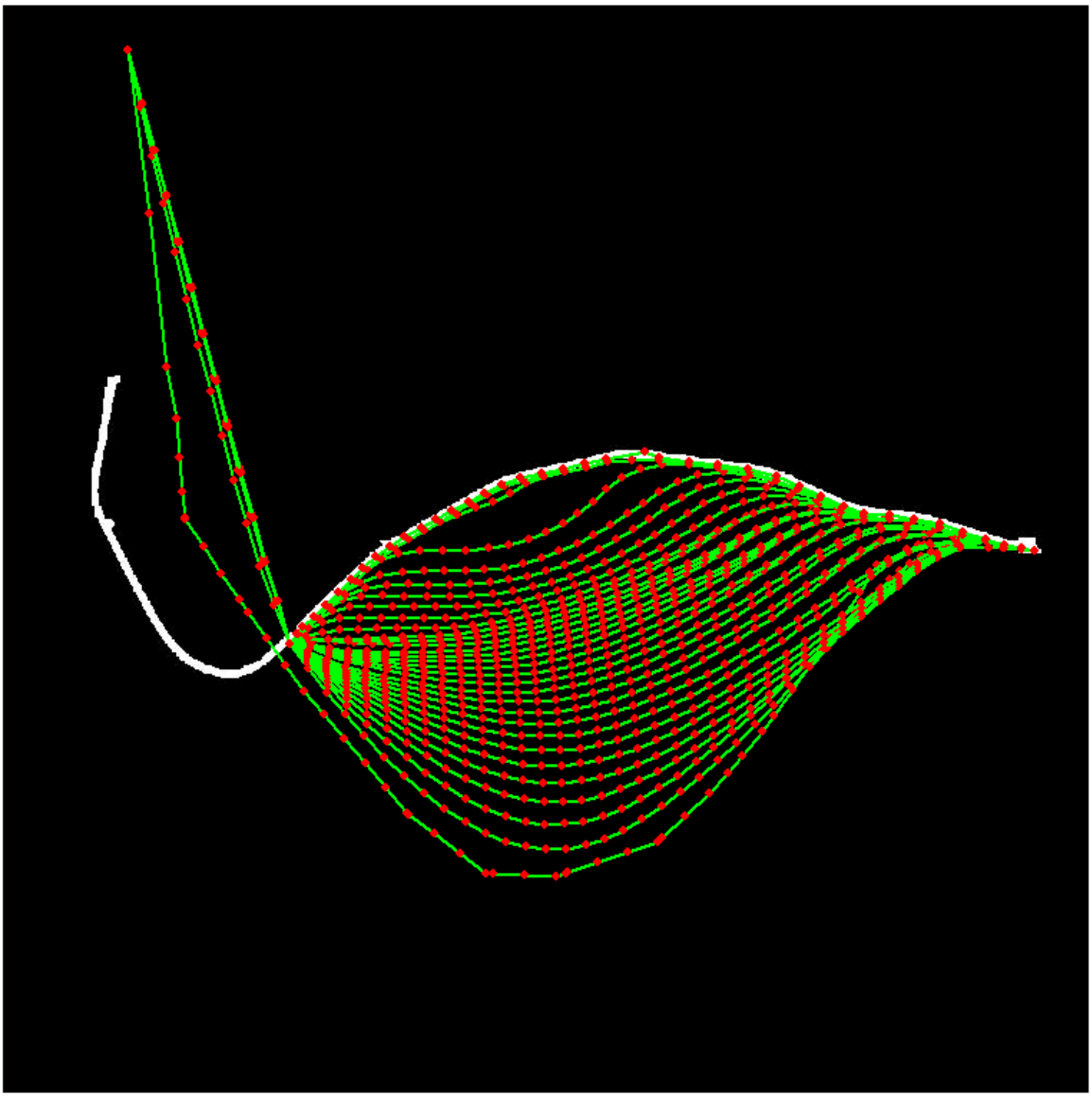


Figure 12. The movements of normal snake operations are shown.

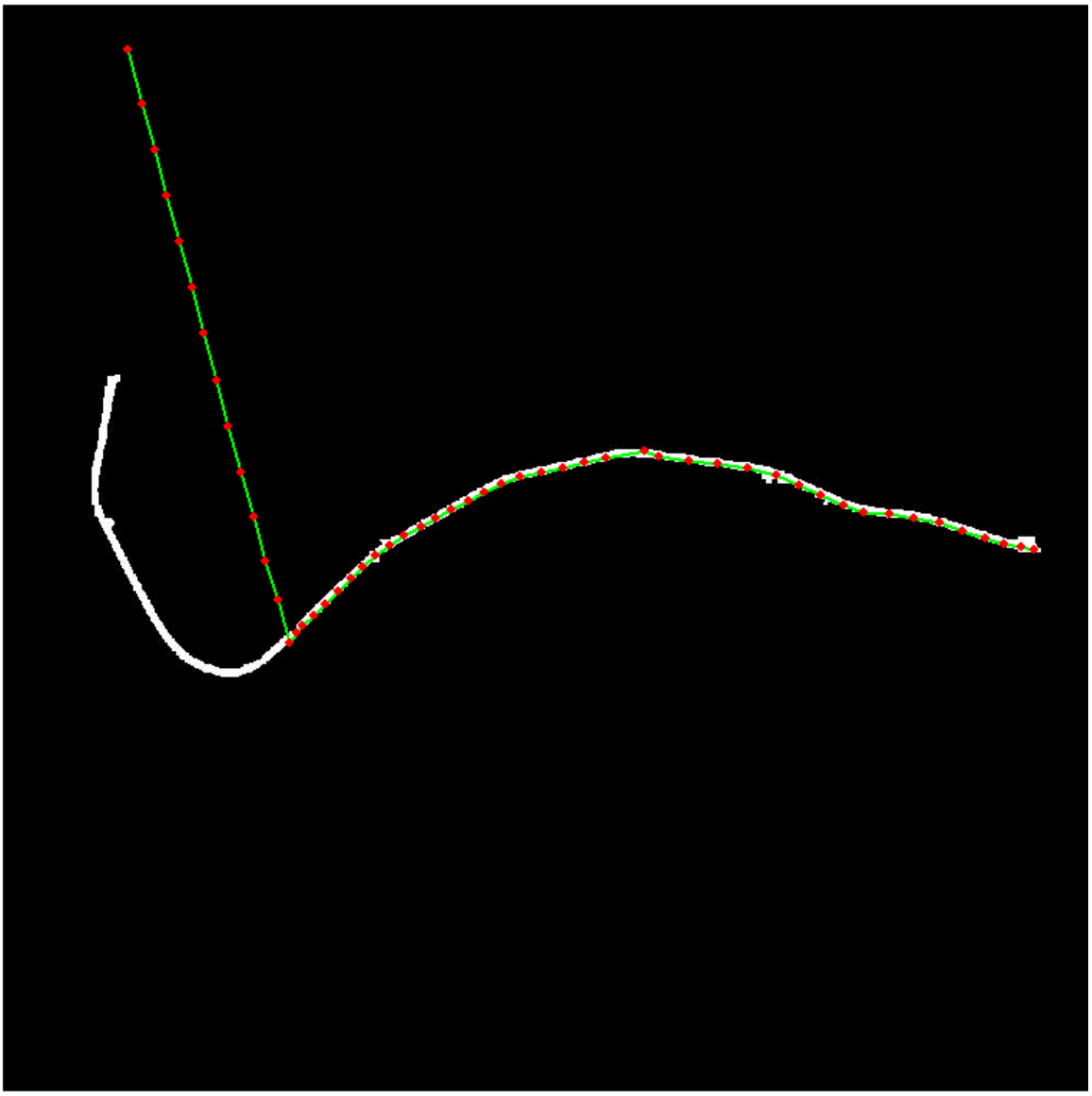


Figure 13. The left part of road segment could not find any external energy since we use smaller search distance, it becomes straight.

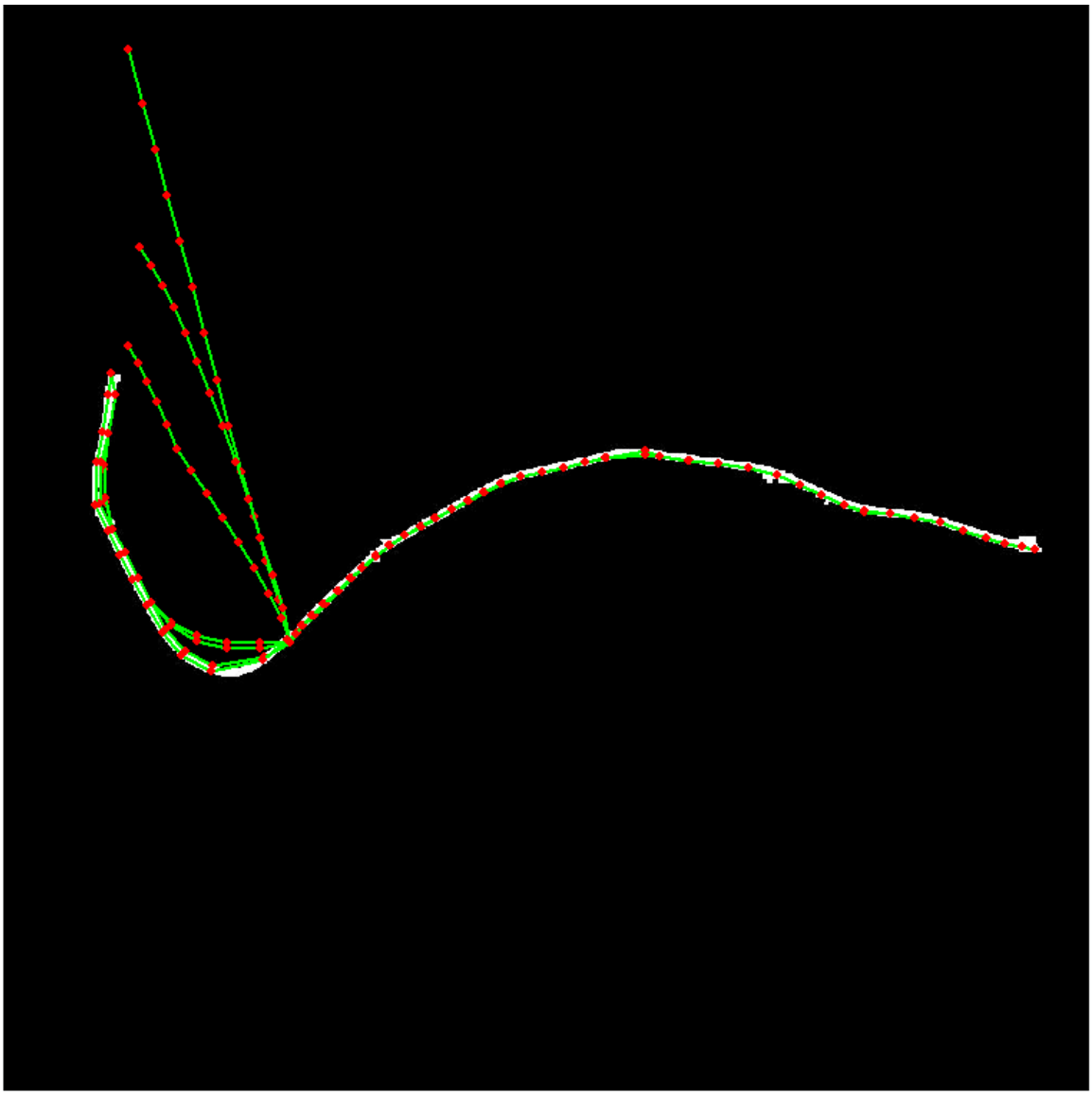


Figure 14. We increase the search distance and run the relaxed version of snake algorithm to move all the vertices to correct locations.

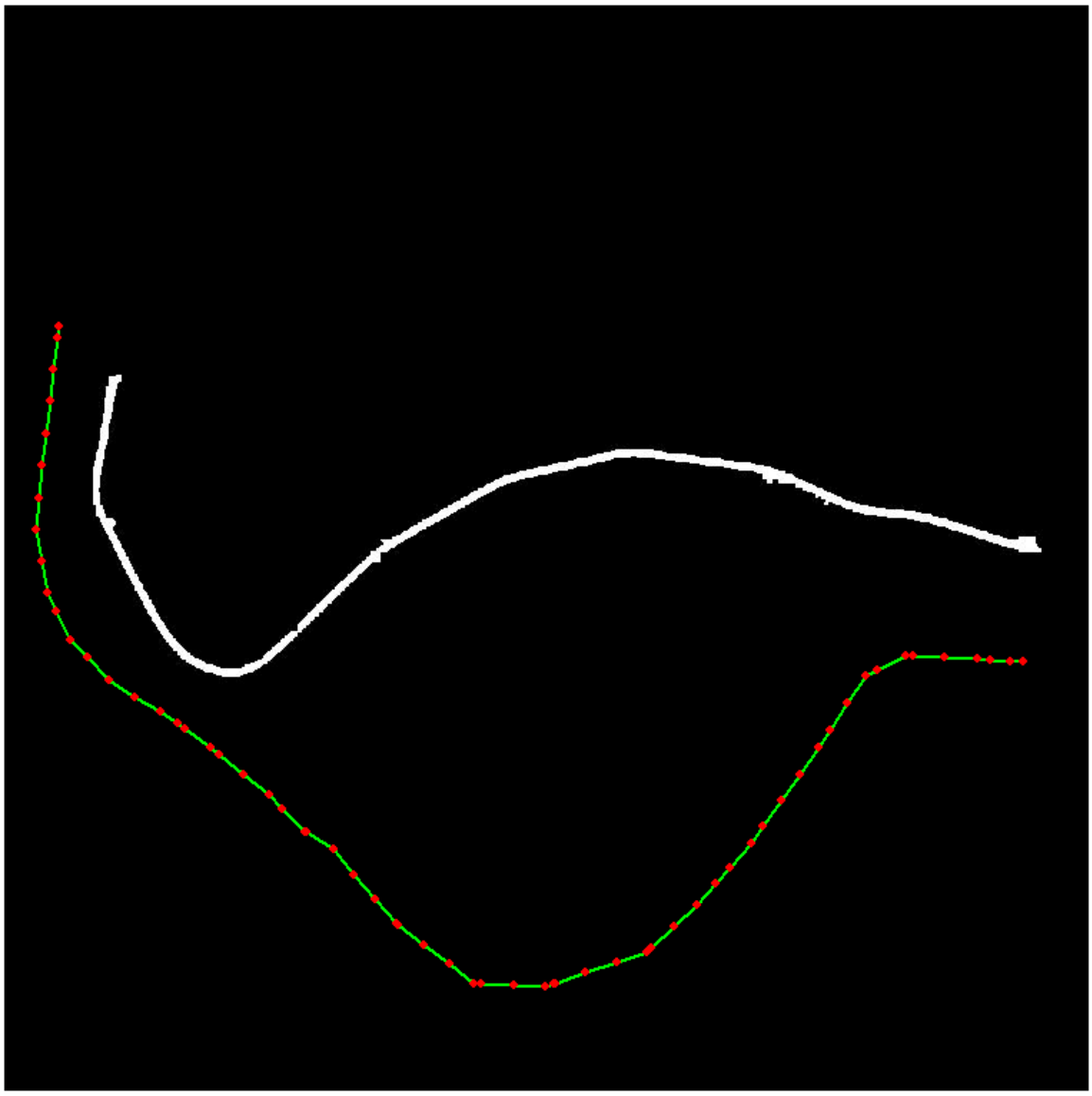


Figure 15. We move both ends of road to wrong location.

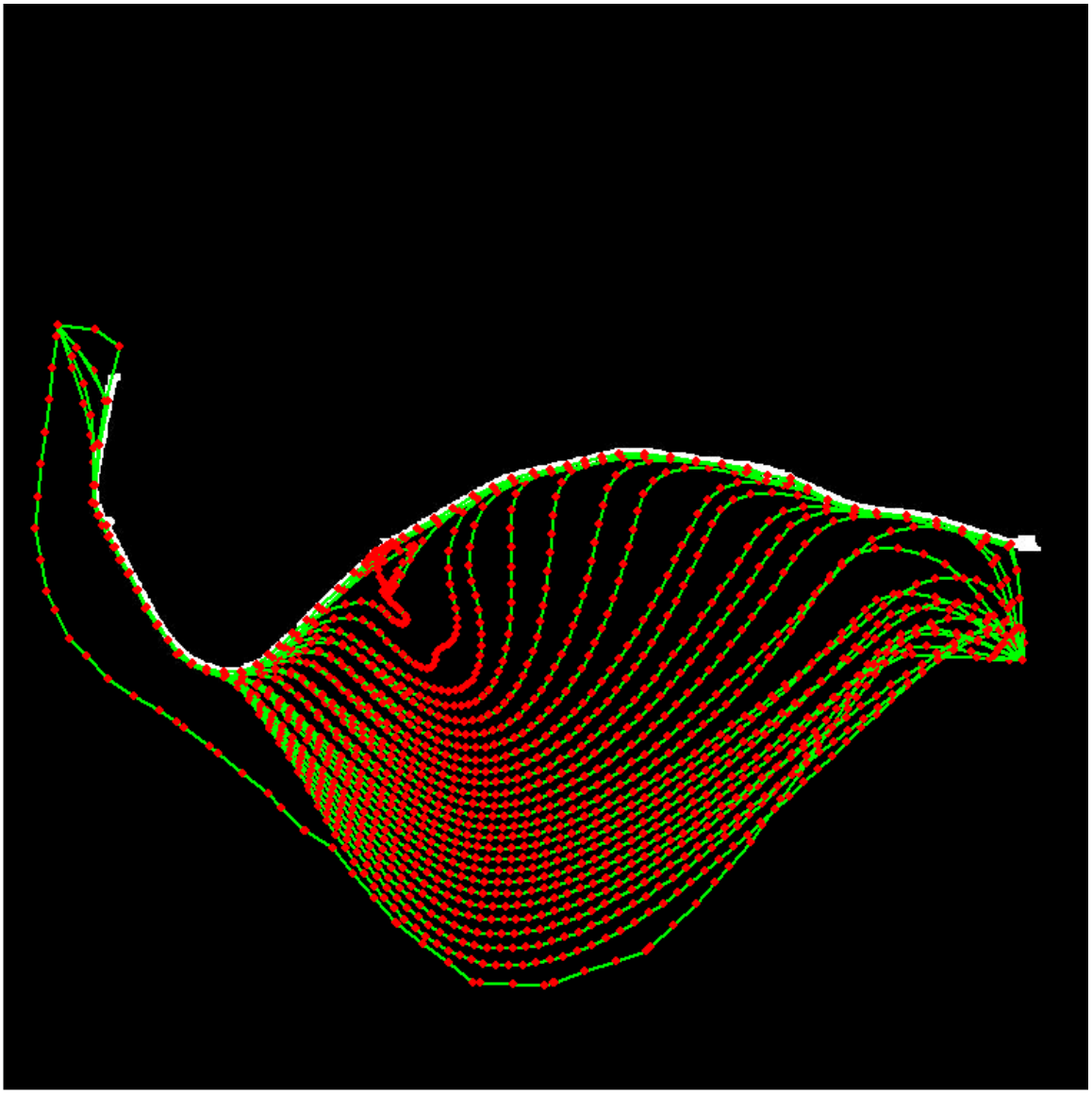


Figure 16. The movements of normal snake iterations are shown.

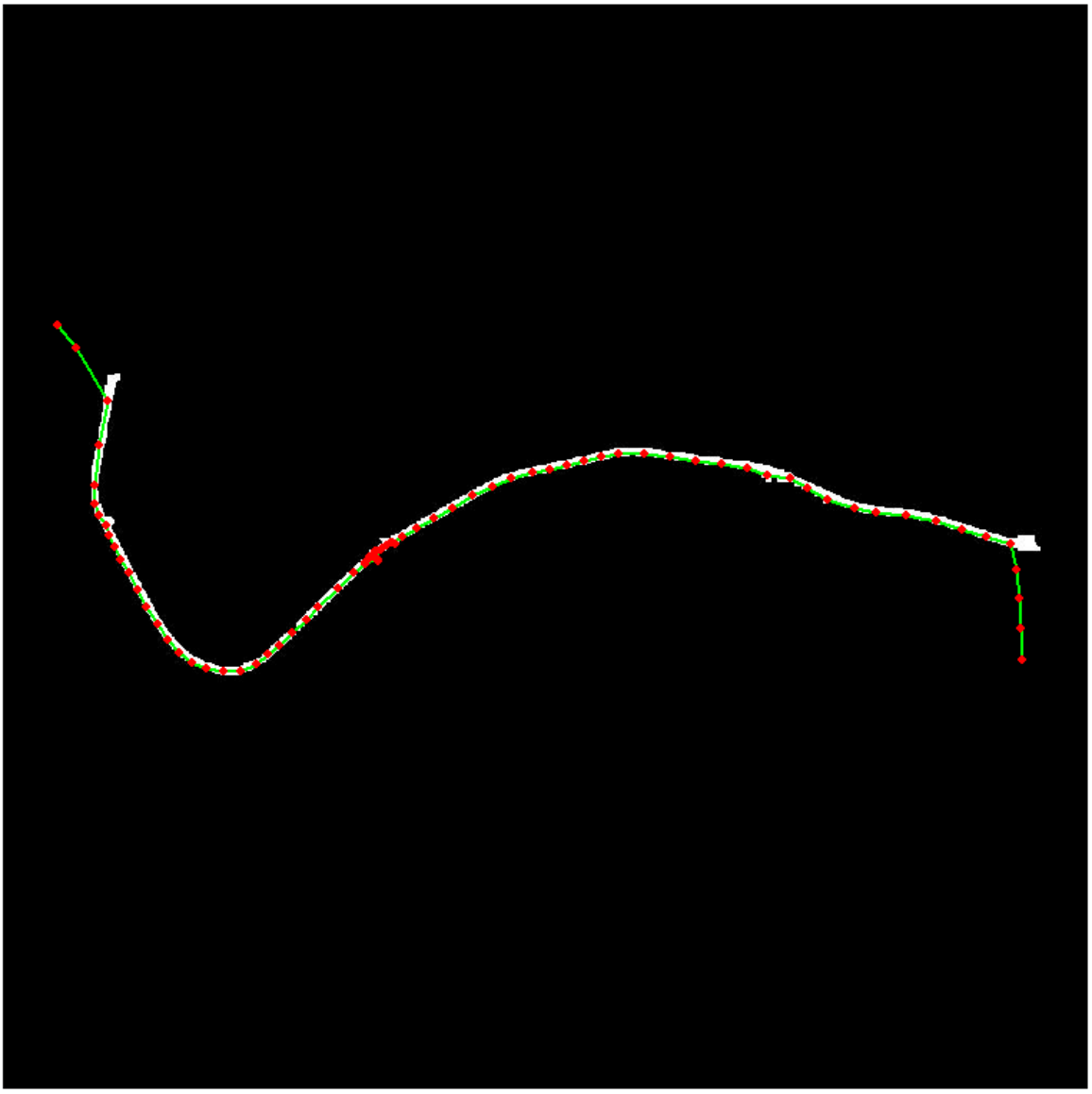


Figure 17. After normal snake operation, most vertices are in correct locations. The vertices near both ends need further correction.

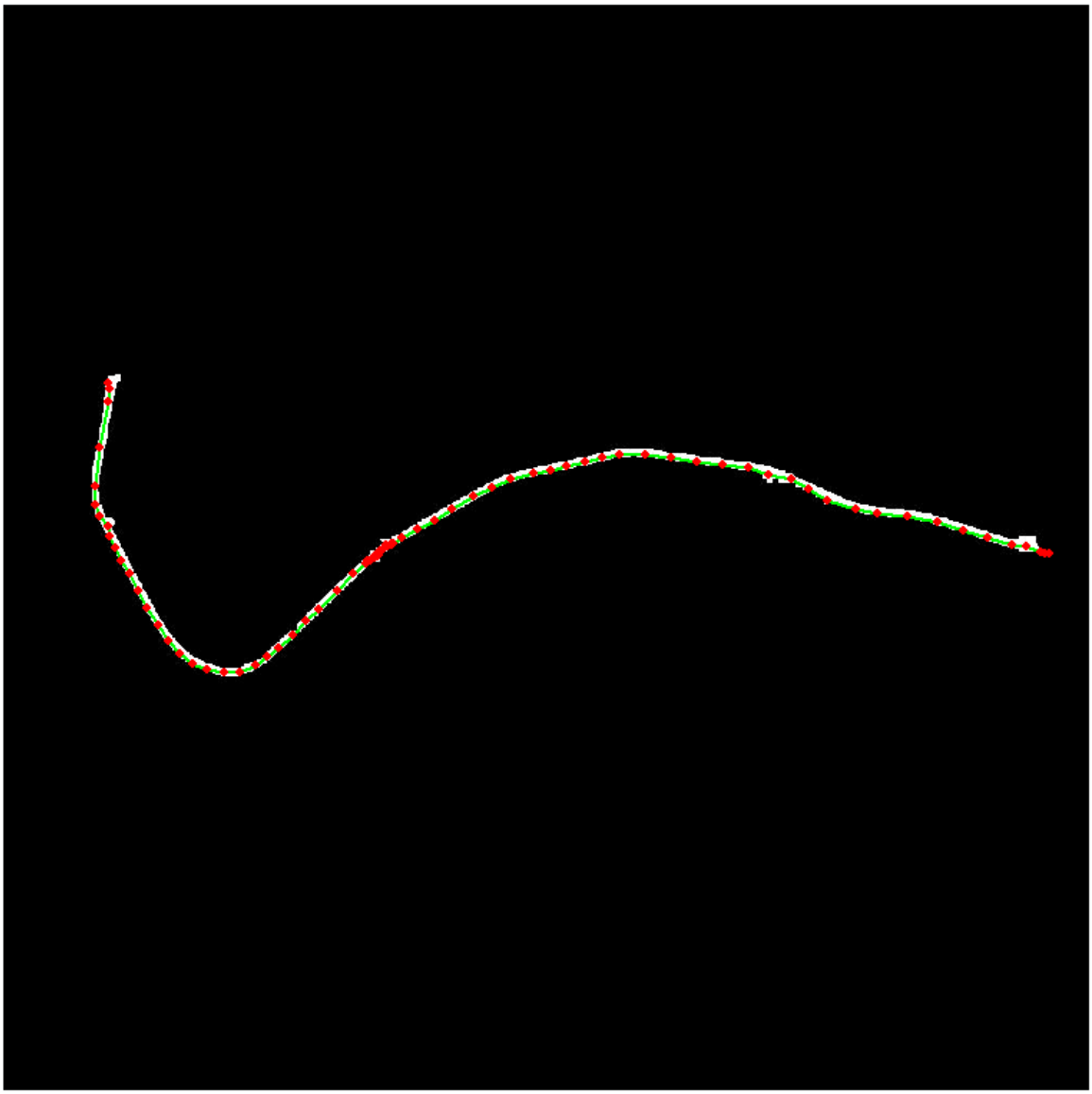


Figure 18. The relaxed snake operation move the vertices near road ends to correct locations.

VITA

Wenbo Song was born in Hubei, China. He received the B.S. degree in Urban Planning in 1990 from Wuhan University in China. He got the M.A. degree in Geography from University of Missouri in 2001. He received his Ph. D. degree in Computer Science from University of Missouri in May, 2011. He works as a research specialist at Department of Geography, University of Missouri since 2001. He married Jie Ning and had two children David and Jenna.