GRID BASED RASTER SELECTION AND VECTOR EXTRACTION

By

AMAR GUPTA

Bachelor of Engineering

D.A.V.V.

Indore, MP, India

2002

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
December, 2006

GRID BASED RASTER SELECTION AND

VECTOR EXTRACTION

Thesis Approved:

Dr. Johnson P Thomas

Thesis Advisor

Dr. G.E. Hedrick

Dr. Allen Finchum

Dr. A. Gordon Emslie

Dean of the Graduate College

ACKNOWLEDGEMENTS

I would like to express my deep and sincere gratitude to my advisor Dr. Johnson P Thomas. His support and ideas have been great value to me. I am grateful for his valuable time and patience in guiding me to complete my thesis. I wish warm and heartfelt thanks to Dr. Allen Finchum for providing me his expertise in the subject and keeping me in right direction. My thanks to Dr. G.E. Hedrick for serving on my graduate committee and supporting me. I am grateful for his comments and suggestions about my thesis.

I owe my loving thanks to my wife Garima, my parents and all family members and friends who have been constant supporter and encourager to me. Without their encouragement and understanding it would have been impossible for me to finish this work.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I

INTRODUCTION

In the new millennium, the Geographic Information System (GIS) industry is seeing a number of new developments and trends that are changing the industry. Among the key trends observed are the explosion of digital imagery and the increasing application of digital imagery with spatial and non-spatial data. The beauty of GIS is bringing together data from many sources by means of a common referencing system. This applies to information that is created with GIS, but also to information that comes to us in the form of paper maps and aerial photographs. This research discusses the process of applying a real-world coordinate system to images and proposes a novel method of extracting information related with maps and aerial photos that we may want to view in juxtaposition with other GIS data. A completely new concept of retrieving vector data by selecting an Area of Interest (AoI) on raster imagery, based on well defined grid structure is proposed.

1.1 Geographical Information System

Geographical Information System (GIS) is an information system that is specifically designed for handling spatial (or geographical) data. It combines a set of interrelated software components that create, edit, manipulate, analyze and display data both in text and graphic forms. GIS supports spatial analysis and modeling within the discipline of geography (e.g. location, proximity and spatial distribution), so that it becomes a vital tool for modern geography. A Geographic Information System (GIS), simply, combines layers of information about a place to give one a better understanding of that place [18]. What types of layers a user combines depends upon the user's perspective and purposes as mentioned above. The combined group of layers is called a Map. One can think of geographic data as layers of information underneath the computer screen. Each layer represents a particular theme or particular feature of a map. One theme could be made up of all the roads in an area. Another theme may represent all the lakes in the same area. Yet another theme underneath of all the themes could be the imagery of that particular area. These themes can be laid on top of each other creating a stack of information about an area that one wants to see, at any time on any specific map.

1.2 Raster Imagery in GIS

Advancements in sensor/scanner technology have resulted in the availability of constantly increasing volumes of digital imagery [4]. Satellite remote sensing development has generated a huge amount of image data during the last two decades. In Parallel with the availability of high-resolution satellite imagery is the development of digital orthophotography. It is a nationwide effort to make aerial photographs available to the public in a high resolution and digital format [3]. Remote sensing, using aerial photography and satellite imagery, coupled with digitized map data offers large-scale environmental data. The integration of these large-scale data sets in GIS allows correlation between a location and environmental, demographical etc variables such as temperature, population etc. Overlaying a GIS map on a digital air photo background will certainly be a more common way of presenting GIS results in the future. For many years before computer-assisted GIS and image processing, it was a familiar presentation form for users. New tools, such as orthophoto quads, along with more options in GIS and image processing software, will make this even easier to do and more commonplace in the future. Thus, there is a great demand for effective GIS data retrieval mechanisms from GIS images. A challenging problem arises with many GIS applications, where queries are posed via visual examples. A typical visual query might entail the location of all images in a database that contain some vector data correlated with a given query image. Such a query could, for example, be used in a remotely sensed satellite image setting to find a specific type of land use/land cover, such as forest or residential areas, road network etc.

A GIS database is composed of all of the geographic/spatial information (maps, imagery) and associated attribute information (tables, reports) that are linked in such a way that we can extract either the spatial or attribute information by requests based on the location or characteristics of the data features either singly or as related to other features. Two main data models are used for storing spatial or geographic features in a GIS: the raster and the vector model. Vector and raster data get the most attention from geospatial vendors. However, location-specific tabular data is far and away the most common kind of spatial data. This is because most data about physical things are inherently location-specific -- everything must be somewhere. Examples include data about customers, inventory, vehicles, cell phone users, houses, roads, rivers, weather, or any other physical object. It is essential that a GIS should have the capability to able to retrieve these location-based data from the raster data itself.

Through its ability to merge spatial data (maps) and non-spatial data (attribute information) in one location, GIS provides a framework for efficient data storage and data retrieval, intuitive display of information in a spatial context, and combining various types of information so that the data may be analyzed further.

The increased volume of digital imagery necessitates the development of novel methods to efficiently retrieve data associate with it. The use of digital imagery is ever increasing for GIS applications due to enabling technology to integrate digital imagery with other land related digital data.

The purpose of this research is to establish a technique that lets the end user of a GIS application to select the area of his interest (AoI) on a digital image and be able to view data (spatial or non spatial) associated with it. In contrast from traditional raster-image based approaches that are static and allow very little user interaction, this technique allows the user to select portions of an image and then deliver vector geographic data and attribute data dynamically. Since the user receives only the data he wants, a lot of processing power and response time is reduced. Furthermore, if the system is used over the Internet, network traffic could be reduced significantly when the user is dealing with large amounts of vector data.

The rest of the thesis is outlined as follows. In chapter II, the problem statement is stated. Chapter III highlights the background of GIS, GIS concepts, and literature review, current state of technology and data models used in this research work. This chapter also discusses some of the work that has been done in the problem domain. Chapter IV contains the proposed solution with a flowchart, block diagram, algorithms and describes how raster cells can be selected dynamically by user and retrieves vector data associated with them. Chapter V contains the implementation and output of the application developed. Chapter VI states challenges faced during the implementation.

CHAPTER II

PROBLEM STATEMENT

Sometimes just looking at a map will tell us what we want to know. Maps not only tell us where things are, but also what's special about them. Maps are not static displays; they're interactive. We can browse a map—taking a closer look at a particular area—and point at features to find out more about them. As we work with a map, we change how we view the data it contains. When we are just browsing a map, we might want to pan and zoom around the data to investigate different areas and features.

Generally, in any GIS application, while viewing any map with imagery in it, users start from a general view of the image and zoom to the interested area. When a user defines an area of interest for image visualization, any existing desktop GIS tools such as ArcGIS [7] will use the geographical bounding box of the defined area and retrieve vector data that covers the area (Figure 1). Geographical bounding box is a rectangle, oriented to the x and y axes, which bounds a geographic feature or a geographic data set. It is specified by two coordinates: xmin, ymin and xmax, ymax.

To zoom into an area of interest, the user needs to select the zoom-in tool. Once he selects the zoom-in tool, he can press the left mouse button at the top-left corner of the area to zoom in and drag the cursor to the lower-right corner of the zoom-in area, drawing a rectangle that represents the area for zooming.

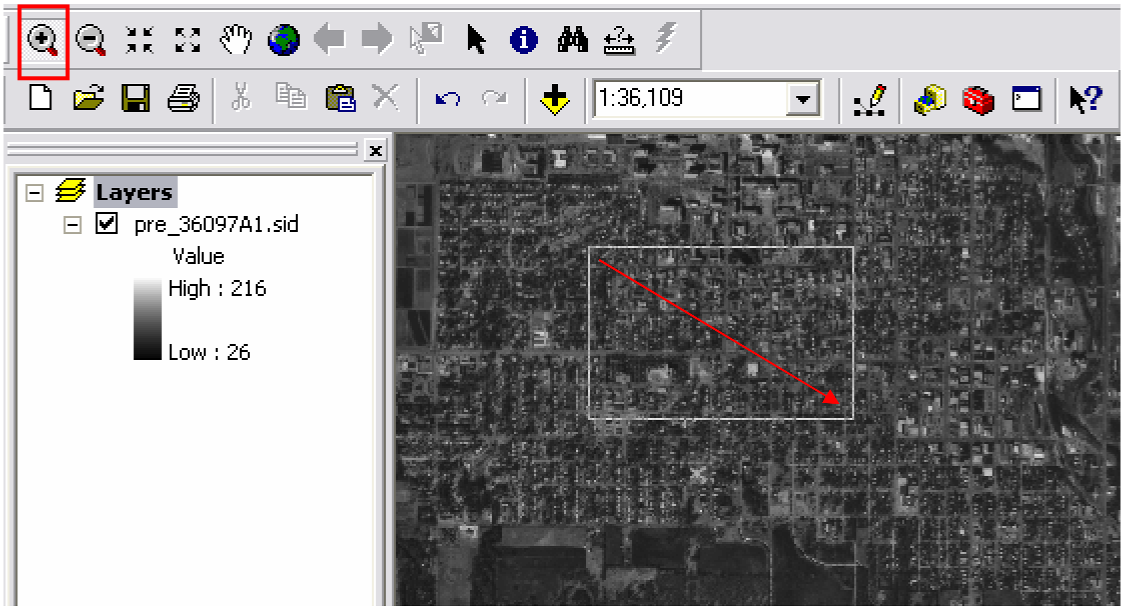


Figure 1. Bounding Box

However, if the user is not interested in looking at vector data of the entire defined area, currently there is no way the user can select a portion of the image and get the vector data that belongs to just that portion. The vector data comes for the portion of the image that falls into the geography of the bounding box (Figure 2).

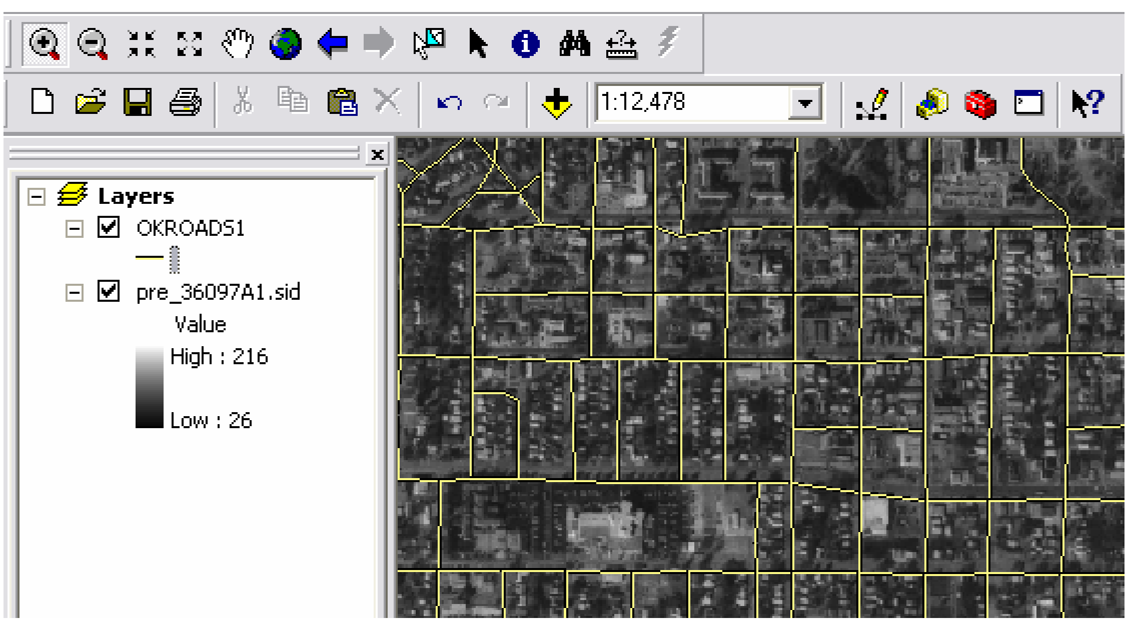


Figure 2. Zoomed area with vector data

ArcIMS, an Internet mapping solution, allows for centrally hosting and serving GIS maps, data, and applications for use on the Internet. It is used for delivering dynamic maps and data over the web. The user selects data layers from an interactive menu based on which the raster layer is displayed on the web-browser. The user selects vector maps as overlays for raster map layer. Interactive zoom/pan capability allows the user to view the displayed maps in greater details or to choose different areas for display. The bounding box extents go to the Web server, which in turn requests a new map image from the ArcIMS server. The ArcIMS server processes requests for maps and related information and generates the image, which goes back to the user.

In the entire process mentioned above, the user doesn’t have the capability to work on the area of interest in the raster image without zooming into that area. As an example, take a case where a user is viewing a raster image (aerial photo) covering the entire city of Stillwater, Oklahoma. The user is only interested in the region covered by the OSU campus. The level of zooming is such that the user sees not only the OSU campus but also large areas surrounding the campus. At this zoom level the user cannot just work on his area of interest unless he zooms into the image to view just the campus. If he wants to see all the streets inside the campus he would get the streets of the entire region in the view. The user is therefore given much more data/information than necessary. The problem gets worse if there are multiple areas of interest, which are apart from each other. For example, the OSU campus and the Airport. Suppose the user wants to do some comparison among these areas of interest. The current system would return data (spatial/non spatial) covering the entire region instead of filtering out areas that are not of interest. This makes the complex nature of the interrelated data more difficult to use. In this situation the user will have more data than he wants, and the system will need to remove the excess.

No available technology allows selection of raster data by the user and only retrieves the vector data associated with it. It is therefore essential to create a functionality that allows the user to select raster data in user friendly way and automatically retrieve the data associated with the selected rater data.

CHAPTER III

LITERATURE REVIEW

This chapter focuses on the current state of technology and further describes various GIS concepts, data models and background to help in understanding the research work. A modern GIS software system comprises an integrated suite of software components, including end user applications, geographic tools and data access components. The Environmental Science Research Institute (ESRI) is the leading developer of geographic information systems software. ESRI’s GIS software packages could be classified into two groups based on the functionality and type: Desktop GIS and Internet GIS. In this research study we will be using a desktop GIS platform to implement our proposed idea to the problem stated in previous chapter. The future work may involve implementing the proposed solution to Internet GIS.

3.1 Desktop GIS

A Desktop GIS focuses on data use, rather than data creation, and provides excellent tools for making maps, reports, and charts. A well-know example is ESRI’s ArcGIS desktop. ArcGIS desktop is used to perform many GIS tasks like mapping, geographic analysis, data edition and compilation on a desktop environment. Users can also develop their own extensions to ArcGIS desktop by working with ArcObjects, the ArcGIS software component library. Users develop extension and custom tools using standard window programming interfaces such as Microsoft’s VBA (Visual Basic

Application), Java, C++. These programming interfaces are used to create stand-alone executables or components that can plug into existing GIS applications. ArcMap is the central application in ArcGIS Desktop for all map-based tasks including cartography, map analysis, and editing. ArcMap is a comprehensive map authoring application for ArcGIS Desktop.

3.2 Internet GIS

Internet GIS focuses on display and query applications, as well as mapping. Examples include ESRI's ArcIMS (Internet Mapping Server). ArcIMS, an Internet mapping server, is used for delivering dynamic maps and GIS data over the Web. Its primary focus is Web delivery of geographic data and maps. It provides a highly scalable framework for GIS Web publishing that meets the needs of corporate Intranets and the demands of worldwide Internet access.

3.3 Map data representation

The central component of GIS is a collection of maps and associated information in digital form. GIS data consists of two components. The first one is the spatial data that describes the geography (shape and position) of the earth's surface. The second is the attribute data, which describe the characteristics or qualities of the geographic locations in question. The attributes may include elements like altitude, population, plant biomass, landform types, etc.

In GIS, all data is defined as either:

- Points (one person's address location)
- Lines (a street or stream)
- Polygons (a parcel or a census tract), or

- Rasters (digital photos or other data made up of pixels or grids)

These "layers" form the core of map-based GIS - by stacking them, you can select between them, easily creating many different views of a place. Layers in GIS typically contain just one type of data (e.g., not points and lines together, but points and lines separately). The basic organization principle of a GIS is the data layer. Rather than storing all spatial features in one place, as on a paper map, groups of similar features are combined in one of a number of these data layers. A comprehensive GIS database will include layers of physical features such as roads, rivers and buildings, as well as layers of defined features such as administrative boundaries or postal zones which cannot be observed on the ground [12]. Other layers may include a climate layer, soil layer etc. as required. These data layers can be represented using vector and raster data models. Each representation has its own methods of presenting and using maps and the data associated with the maps.

3.4 Feature and Raster Geometry

A GIS typically represents geographic location using rasters or vectors (feature geometry). In addition to vector features and raster data sets, all other spatial data types can be managed and stored in the relational tables, allowing you the opportunity to manage all geographic data in an RDBMS.

Vector features (geographic objects with vector geometry) are a versatile and frequently used geographic data type, well suited for representing features with discrete boundaries such as wells, streets, rivers, states, and parcels. A feature is simply an object that has a location stored as one of its properties (or fields) in the row. Typically, features are spatially represented as points, lines, polygons, or annotation, and are organized into feature classes (Figure 3). Feature classes are collections of features of the same type with a common spatial representation and set of attributes (e.g., a line feature class for roads).

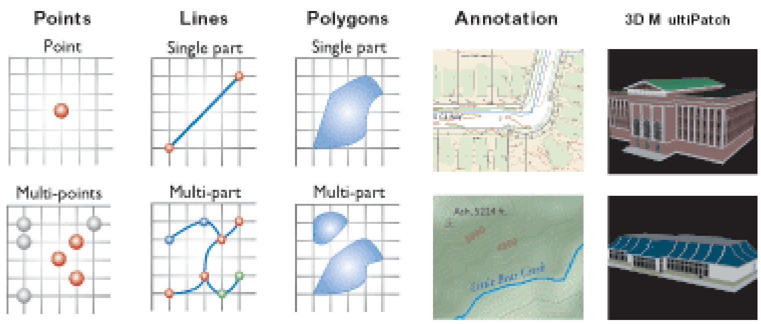


Figure 3. Common vector feature representations in a GIS [from 8].

Rasters (Figure 4) are used to represent continuous layers, such as elevation, slope and aspect, vegetation, temperature, rainfall, plume dispersion, and so on. Rasters are most commonly used for the storage of aerial photographs and imagery of various kinds.

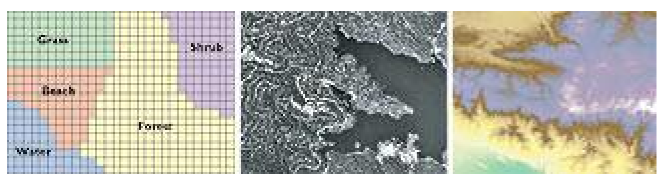


Figure 4. Raster data sets representations [from 8].

3.5 Vector Data

The Vector data model represents real-world features using a set of geometric primitive points, lines and polygons. A point is represented in a computer database by an x, y coordinate called a point feature. Points may be connected to form lines called line feature. A line feature is a sequence x, y coordinates, whereby the end points are usually called nodes and the intermediate points are termed vertices. Polygons are represented by a closed series of lines such that the first point equals the last point of the loop. Points might be used to represent houses, wells or geodetic control points; lines describe such features as roads and rivers; enumeration areas or districts, for example, are represented by polygons (Figure 5). The point, line and polygon features of vector data are stored in a special structure file format that can be readable by different GIS applications; this special structure file format is known as a ShapeFile. A ShapeFile stores the location, shape, and attribute information of geographic features. Where shape can be a point, line or polygon. A ShapeFile stores the geometry and attribute information for the spatial feature in a data set. This dataset is a combination of 3 other files; the other files are called main file, index file, and a dBASE file. The main file is a direct access, variable-record-length file in which each record describes a shape (point, line, and polygon) with a list of its vertices. In the index file, each record contains the offset of the corresponding main file record from the beginning of the main file. The dBASE table contains feature attributes with one record per feature. dBASE file is similar to database file which is indexed with other files in a ShapeFile.

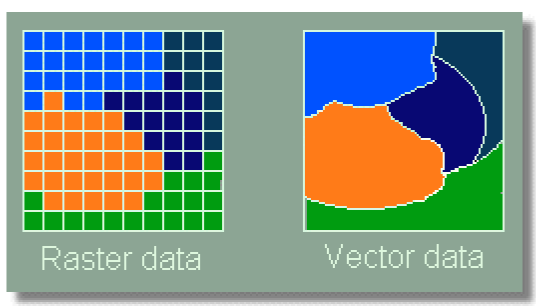


Figure 5. Raster Data and Vector Data [from 11]

3.6 Raster Data

In raster representation, the graphic and attributes are merged into a unified data file. The study area is subdivided into a fine mesh of grid cells (pixels). Each cell is given a number that may represent a quantitative or qualitative value that characterizes the site. For example the value may represent the light reflectance of a surface in different wavelengths or a number that represent a given characteristic. In case of images (photograph type images) reflectance in different wavelengths are stored in different layers and the image reconstructed by “overlaying” them. Raster files are data intensive since they have to record information at every point but since they mimic the computer data architecture they can be rapidly evaluated. Furthermore raster images better represent spatial data and thus raster systems have substantially more analytical power. Raster systems are the best for the analysis of natural resources and agriculture data primarily because satellite images, which are the basis for most GIS uses in the field, are stored in raster form. Organization GIS databases are organized in a form similar to a collection of maps. The main difference is that in the case of raster systems the images are divided in individual layers, which can be combined, if so desired, through overlaying. The advantage of this arrangement is that a researcher can view an image only through a particular wavelength, for example near infrared, if that is the layer that provides the most pertinent information. Composite images can be created with a combination of both vector and raster layers. An advantage of GIS maps is that they can be displayed at any scale, which means that data layers that were derived from paper maps of different scales but covering the same region can be easily combined. All spatial data in GIS is georeferenced based on the location of an image or layer in space as defined by a given coordinate system.

Vector layers have features such as points, lines and polygon. However raster layer do not contain features per se. The closest equivalent is pixel. The main difference between them is that in vector space, objects are usually well defined, attributed and their spatial relations determined, whereas in raster space, typically this is not the case. The distinction is important because when object extents and spatial relations are well-defined a-priori, the operations and techniques for manipulating and querying their properties are different than for raw objects on a raster image.

When a raster spatial database does contain additional textual, information about the imagery it contains, it is referred to as "metadata". Currently, there isn't a generally accepted standard format the image metadata, although an attempt is being made to address this problem Metadata therefore, in the sense used in this research, consists of a listing of potential values for a set of attributes which describe general properties of the image itself, but not about any image-object or other details that the image may contain. These attributes may include such additional information such as: date and time of image acquisition; date and time of introduction in the database; scale/resolution; location of the image, expressed in hierarchically arranged geographic entities like state, country, city, etc.; and/or sensor information and/or imagery type, e.g. black & white, color, color infrared, etc.

3.7 Interactive Selection of Features in ArcGIS

ArcGIS provide many ways to retrieve information about features in ArcMap [17]. The current desktop GIS technology provides various tools to select vector data interactively. This chapter discusses some of the techniques in selecting feature data. The user can identify features by clicking on them in order to display their attributes. The user can select features by clicking on the features to highlight them and look at their records in the layer attribute table. The user can find features by using known information about the feature in order to search the map for that particular feature. The user can select features interactively with the mouse pointer by clicking them one at a time or by dragging a box around them. Before selecting features with one of these methods, the user can specify the layers he wants to select from the selectable layers. The user can select from the layers that are checked in the Selection tab at the bottom of the table of contents or in the Set Selectable Layers dialog box, which he can open by clicking the Selection menu and clicking Set Selectable Layers.

3.7.1 Identifying Features

Perhaps the fastest way to get information about a single feature is to identify it, using the Identify Tool. To use the Identify tool, the user must select it from the Tools Toolbar. Within the map, the user must click on the feature of interest in order to view the attribute information for that particular feature.

3.7.2 Selecting Features

If the user wants to compare information about several features, the best way is to select the features on the map and look at their records in the layer attribute table. The easiest way to select multiple features is by using the Select Features Tool on the Tools Toolbar. To use the Select Features Tool, the user must select it from the Tools Toolbar. On the map, all features of interest may be selected by holding down the shift key and clicking on the various features of interest. The selected features will be outlined in blue. If a feature is selected by mistake it can be de-selected by holding down the shift key and clicking the feature again. All features that have been selected can be cleared by clicking the Selection menu from the Standard Toolbar and selecting the Clear Selected Features option. To view the selected features' attribute table, the user must right-click on the data layer where features have been selected. The Open Attribute Table option should then be clicked.

3.7.3 Finding Features

When the user has a piece of information about a feature, but is not sure where that feature is on the map, the user can search the map for that feature using the known piece of information. The user can find a feature, by selecting the Find tool on the Tools toolbar. When the Find dialog Box appears, the Features tab should be selected. The known attribute information should be typed in the Find box. In the In Layers drop down box, the layer that the user wishes to find features in should be selected. In the Search options, the user should choose to either search all fields in the attribute table or a specific field. Once all parameters are set, the Find button should be clicked.

3.8 Map overlay

Map overlay is the combination of two separate spatial datasets (points, lines, polygons, or images) to create a new output dataset. These overlays are similar to mathematical Venn diagram overlays. A union overlay combines the geographic features and attribute tables of both inputs into a single new output. An intersect overlay defines the area where both inputs overlap and retains a set of attribute fields for each. A symmetric difference overlay defines an output area that includes the total area of both inputs except for the overlapping area. Data extraction is a GIS process similar to vector overlay, though it can be used in either vector or raster data analysis. Rather than combining the properties and features of both datasets, data extraction involves using a "clip" or "mask" to extract the features of one dataset that fall within the spatial extent of another dataset.

In raster data analysis, the overlay of datasets is accomplished through a process known as "local operation on multiple rasters" or "map algebra," through a function that combines the values of each raster's matrix. This function may weigh some inputs more than others through use of an "index model" that reflects the influence of various factors upon a geographic phenomenon.

Figure 6. Overlay of Vector and Raster data [from 13].

The easiest way to think of raster data is as an unintelligent picture such as a satellite image or an aerial photograph. This data does not have the intelligence to allow you to do complex analysis on individual map features. It is best suited for use as a backdrop under a Vector data map to improve the appearance (Figure 6).

3.9 Types of Information in a Digital Map

Any digital map is capable of storing much more information than a paper map of the same area, but it’s generally not clear at first glance just what sort of information the map includes. For example, more information is usually available in a digital map than what you see on-screen. Furthermore, evaluating a given data set simply by looking at the screen can be difficult. For example, what part of the image is contained in the data and what part is created by the GIS program’s interpretation of the data? We must understand the types of data in our map so we can use it appropriately. [10]

Three general types of information can be included in digital maps:

- Geographic information,

  which provides the position and shapes of specific geographic features.

- Attribute information,

  which provides additional non-graphic information about each feature.

- Display information,

  which describes how the features will appear on the screen.

Some digital maps do not contain all three types of information. For example, raster maps usually do not include attribute information, and many vector data sources do not include display information.

3.9.1 Geographic Information

The geographic information in a digital map provides the position and shape of each map feature. For example, a road map's geographic information is the location of each road on the map.

In a vector map, a feature's position is normally expressed as sets of X, Y pairs or X, Y, Z triples, using the coordinate system defined for the map (see the discussion of coordinate systems, below). Most vector geographic information systems support three fundamental geometric objects: (as discussed before)

- Point: A single pair of coordinates.
- Line: Two or more points in a specific sequence.
- Polygon: An area enclosed by a line.

Some systems also support more complex entities, such as regions, circles, ellipses, arcs, and curves.

3.9.2 Attribute Information

Attribute data describes specific map features but is not inherently graphic. For example, an attribute associated with a road might be its name or the date it was last paved. Attributes are often stored in database files kept separately from the graphic portion of the map. Attributes pertain only to vector maps; they are seldom associated with raster images.

GIS software packages maintain internal links tying each graphical map entity to its attribute information. The nature of these links varies widely across systems. In some the link is implicit, and the user has no control over it. Other systems have explicit links that the user can modify. Links in these systems take the form of database keys. Each map feature has a key value stored with it; the key identifies the specific database record that contains the feature's attribute information.

3.9.3 Display Information

The display information in a digital-map data set describes how the map is to be displayed or plotted. Common display information includes feature colors, line widths and line types (solid, dashed, dotted, single, or double); how the names of roads and other features are shown on the map; and whether or not lakes, parks, or other area features are color-coded.

However, many users do not consider the quality of display information when they evaluate a data set. Yet map display strongly affects the information you and your audience can obtain from the map -- no matter how simple or complex the project. A technically flawless, but unattractive or hard-to-read map will not achieve the goal of conveying information easily to the user.

Oddly enough, many common data sets contain no display information. For example, USGS Digital Line Graph files provide no display information at all. Each feature contains an attribute that describes the entity but does not indicate display features. Users, and their GIS software, must interpret those attributes and decide how each will look on the final display.

3.10 ESRI Shapefile

A shapefile is a special data file format that stores non-topological geometry and attribute information for the spatial features in a data set. The geometry for a feature is stored as a shape comprising a set of vector coordinates in a location list data structure (LLS). Shapefiles can support point, line, and area features. Area features are represented as closed loop polygons a shapefile must strictly conform to the ESRI (Environmental Systems Research Institute) specifications [9]. It consists of a main file, an index file, and a dBASE table. The main file is a direct access, variable-record-length file in which each record describes a shape with a list of its vertices. In the index file, each record contains the offset of the corresponding main file record from the beginning of the main file. The dBASE table contains feature attributes with one record per feature. The one-to-one relationship between geometry and attributes is based on record number. Attribute records in the dBASE file must be in the same order as records in the main file.

3.11 Georeferencing Vector and Raster

Given a raster data set and a vector data set, we address the problem of extracting information from raster data over a set of boundaries. The assumption of this problem is that the raster and vector data sets overlap geographically and both data sets are described by sufficient georeferencing information for data fusion [2].

Data Sets Georeferencing, or geographic referencing, is the name given to the process of assigning values of latitude and longitude to features on a map. Latitude (lat) and longitude (long) describe points in three-dimensional (3D) space, while maps are inherently two-dimensional (2D) representations. With the advent of computers, modern maps are usually stored as digital images since they represent raster information similar to 2D image information. The steps involved in georeferencing a digital map image can vary among map image specification types, but the end result is the ability to retrieve the lat/long coordinates for any point on the georeferenced map. This capability is useful because the lat/long coordinates precisely define the position of an object on the Earth. Lat/long coordinates define a point on a 3D model of the Earth, while map coordinates represent a pixel – a row and column location on a 2D grid obtained from projecting some 3D model of the Earth onto a plane. 2D maps are easy to display and facilitate distance measurement, while 3D coordinates are accurate but cumbersome and have no standard length for different degrees of latitude and longitude. The best way to define the position of an object on a map is with relative horizontal (column) and vertical (row) distances: "Three kilometers north of object A and two kilometers west of object B". In contrast, the best way to specify the position on a 3D sphere is with relative angular offsets: "Five degrees north and six degrees west of A".

A software package ArcGIS, with components that include ArcView and ArcExplorer provided by ESRI, is one of the GIS systems that can accept geospecific data and provide geographic referencing.

There is nearly limitless information available describing georeferencing [15][13], but the level of understanding necessary to correctly georeference a single image can be rather daunting. The difficulty is due in part to the complications of projecting a three-dimensional surface, the Earth, onto a two dimensional object, a map. Understanding something about this process goes a long way towards understanding why certain parameters are needed to geographically orient a digital image.

3.12 Georeferencing Coordinate Transformations

The georeferencing system is built around coordinate transformations to and from three types of location representations: 2D map pixels (column and row), 3D lat/long coordinates, and 2D Universal Transverse Mercator (UTM) values. These transformations are shown in Figure 7. The motivation for the transformations between 2D map and 3D lat/long coordinate systems is fairly obvious, because one would like to know the lat/long values of any pixel in a georeferenced image. The third location representation, the UTM coordinate system, is a Cartesian coordinate system useful for specifying a number of points on a map without having to refer to latitude and longitude. Furthermore, UTM values facilitate metric distance calculations, which are difficult in lat/long coordinates where the distance between two adjacent degrees is dependent upon their location relative to the equator and the prime meridian.

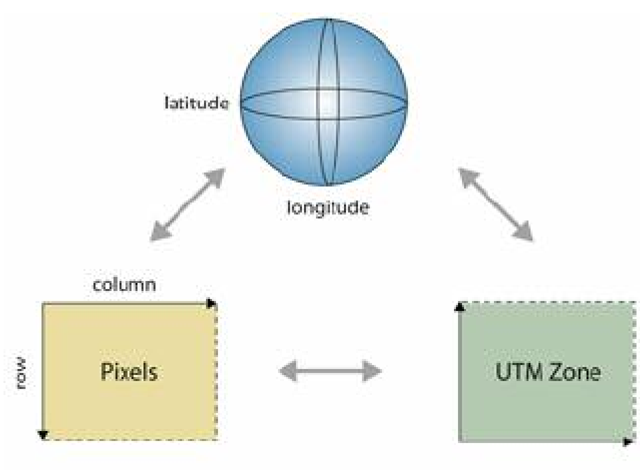


Figure 7. Types of coordinate transformations [from 10].

CHAPTER IV

PROPOSED SOLUTION

The problem stated in problem statement section, has two parts to consider. First is the selection of raster cells from the imagery and second is the extraction of vector data that belong to the geography of selected cells. As mentioned in chapter III section 3.6, current GIS technology has many ways to select vector data on maps, however selection of raster data is still an area of research. We propose a grid-based approach to select AoI on the raster imagery. To enhance the map browsing process and to disburden the user from complex mind maps of the image, a transparent grid is provided over the top of image. The advantage of using the proposed grid becomes more apparent if interaction is considered. Specifically, if we use this technique in mobile devices which suffer from a number of limitations like small screen size, limited processing power and bandwidth, it can substitutes sliders and menus, and bring fast and intuitive interaction to zoom and pan techniques. Grid interaction can be basically understood as virtual equivalent for related physical tasks as moving or folding maps. To navigate and interact with grid, an arbitrary pointing device supporting moving and selecting is required. The user can simply click on cells according to AoI and selected cells get highlighted. With the help of World file (explained in following sections), which is a separate ASCII file that comes with Raster Imagery, the proposed solution creates a temporary polygon shapefile containing the geography of selected cells. This shapefile is called Clipper, which is in turn used in clipping the target vector data. A GIS clip method is performed which uses a clipper theme like a cookie cutter on the target theme. The target theme's attribute is not altered and the user gets the desired output data. The following sections describe the solution explained above in detail with the help of algorithms, flowcharts and diagrams.

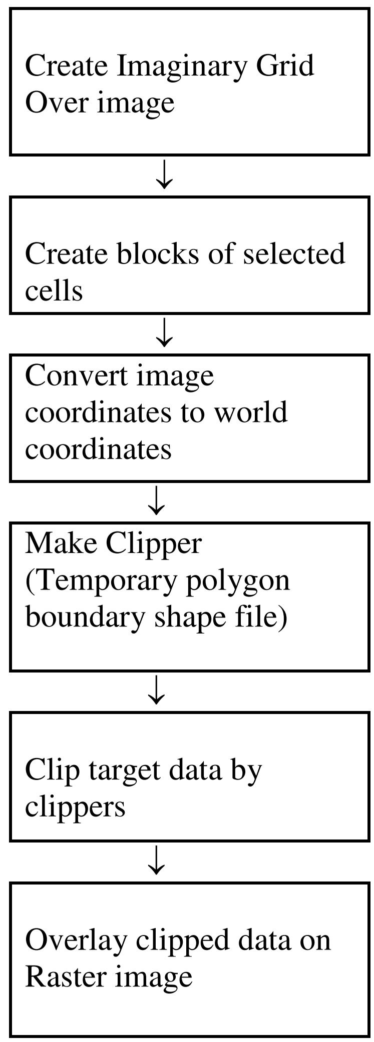


Figure 8. Block Diagram

The above block diagram (Figure 8) shows the major steps taken into consideration of proposed solution.

4.1 Create Imaginary Grid on Raster Data

Algorithm for Creating Grid:

CellSize - size of each cell defined in a grid.

Rows - number of rows in a grid.

Cols - number of columns in a grid.

MaxX, MaxY - Maximum x, y coordinates of Raster Image.

MinX, MinY - Minimum x, y coordinates of Rater Image.

Row - row number of selected cell.

Col - column number of selected cell.

RowColList - list of row and column number of selected cells.

CellCount - number of cells selected

1. Set CellSize. By default, CellSize = actual cell size of raster imagery.
2. Calculate number of rows in a grid.

$$\text{Rows} = \frac{\text{MaxY} - \text{MinY}}{\text{CellSize}}$$

3. Calculate number of columns in a grid.

$$\text{Cols} = \frac{\text{MaxX} - \text{MinX}}{\text{CellSize}}$$

4. Draw Grid (Rows, Cols, CellSize)

Raster data is an abstraction of the real world where spatial data is expressed as a matrix of cells or pixels with spatial position implicit in the ordering of the pixels. With raster data model, spatial data is not continuous but divided into discrete units. This makes raster data particularly suitable for certain types of spatial operation, for example overlays or area calculations. To achieve the selection of AoI on raster image, an imaginary grid is provided. We will create an imaginary grid superimposed over an image-based raster data. The grid is mainly used for interaction that helps users in identifying cells of rater image. The grid is not stored in any database so we will call it imaginary grid. The image area is divided into rows and columns, which form a regular grid structure. A grid defines a geographic space as a matrix of identically sized square cells. The size of the grid will determine by the image extent, which is top, left, bottom and right coordinates. The grid outline is defined to represent the proportions of image to explore. The user, depending upon the minimum area of selected portion over imagery determines the cell size. E.g. if the cell size is 300 meter then the minimum selectable area will be 300 meter x 300 meter. Cell size can be change depending on the user's need. The pixel equivalent is usually referred to as a cell element or grid cell. Pixel/cell refers to the smallest unit of information available in an image or raster map. This is the smallest element of a display device that can be independently assigned attributes such as color. As in real world, grid is necessary to locate a desired image section. Additionally, a number of intuitive features are inherent if interacting with the grid. Nevertheless, if not needed grid can be hidden.

4.2 Raster cells selection

Algorithm for Selecting Grid cells:

```
1.      While (select more cells = true)
        {
                If (CellSize <= 0) Then Error Message "Select Cell size"
                //Convert mouse click to map units
                Set pPoint = DisplayTransformation.ToMapPoint(X, Y)
                // Calculate row and column number of selected cell
                Row = Ceil (Abs (pPoint.Y - YMax) / cellsize)
                Col = Ceil (Abs (pPoint.X - XMin) / cellsize)
                RowColList.Add (Row, Col)
                CellCount = CellCount  + 1

        } // End While
```

Cell-based systems divide the world into discrete uniform units called cells, based on a grid structure. Every cell represents a certain specified portion of the earth, such as a square kilometer, hectare, or square meter. Each cell is given a value to correspond to the feature or characteristic in which it is located, or describes the location, such as an elevation value, soil type, or residential classification. Once a grid is defined, the user can select cells from it depending on the area AoI on image. Clicking mouse on image selects cells as shown in Figure 9.

Placing an imaginative grid over the image not only allows selecting cells but also help in finding geographical location of each selected cells. Given the information about the bounding coordinates of the grid and number of rows/columns, the location of each cell can be calculated. Therefore, no explicit location value is needed for each cell.

To make the processing efficient and fast we will create blocks of selected cells. A block will consist of a group of neighboring selected cells. Each square (in the grid) will have something of significance in it, which might be a building, a street or a lake etc.

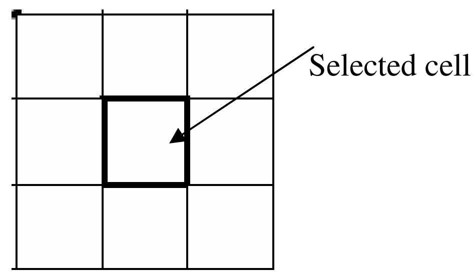


Figure 9. Selected cell in a Grid

4.3 Create clipper

Algorithm for creating Clipper:

This algorithm creates a clipper file for selected grid cells. A polygon feature is created for each cell selected.

1. Define new Feature Class "Clipper"
2. For (n = 0 to n = CellCount)

```
{
        //Get row, column for the cell
        Row, Col = RowColList (n)
        //Calculate coordinates for each corner of the cell
        Point1 (x, y) = MinX + (Col - 1) * CellSize, MaxY - Row * CellSize
        Point2 (x, y) = MinX + (Col - 1) * CellSize, MaxY - (Row -1) * CellSize
        Point3 (x, y) = MinX + Col * CellSize, MaxY - (Row - 1) * CellSize
        Point4 (x, y) = MinX + Col * CellSize, MaxY - Row * CellSize
        Polygon (Point1, Point2, Point3, Point4)
        Define new Feature Class "pFeature"
        PFeature.Shape = Polygon
        //Merge pFeature into Clipper
        Clipper.Merge (pFeature)

} // End For
```

To get the vector spatial data, which will overlay over a raster image, we need to create "clipper" shape file, which will clip the desired vector data. The shape of a clipper will be a polygon that is nothing but a block containing selected grid cells. We will find geographical coordinates of every corner of a block and will create a polygon boundary shape file using these sets of points. The following diagram (Figure 10) shows a grid with a block containing selected cells in it.

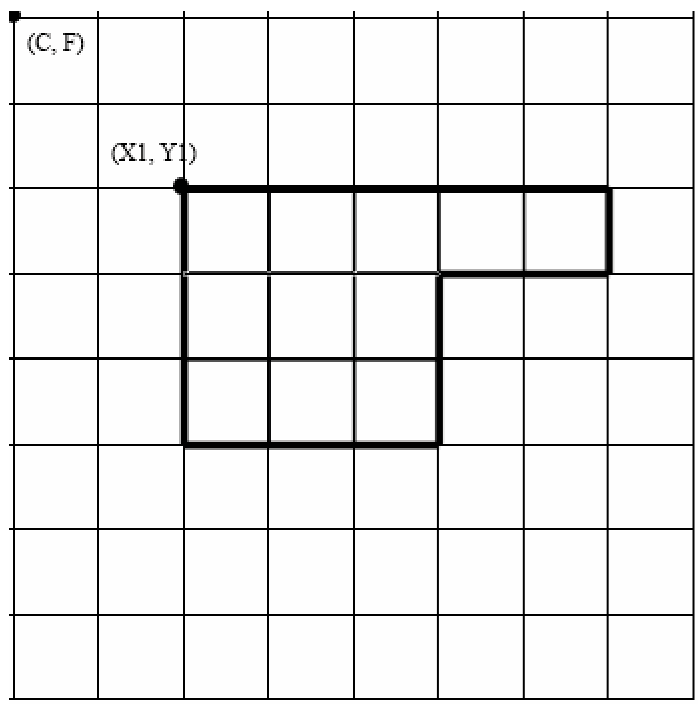


Figure 10. Clipper in a Grid

We might have multiple clippers because the user can select grid cells all over the image according to his need, and group of cells can be separated from one another. This will result multiple blocks or clippers as shown in diagram (Figure 11).

Figure 11. Grid Blocks [from 16]

4.3.1 Georeferencing with World Files

The vector data Shape files are stored in real-world coordinates. World coordinates are measurement of a location on the earth's surface expressed in degrees of latitude and longitude. In order to display raster images with shape files, it is necessary to establish an image-to-world transformation that converts the image coordinates to real-world coordinates. This transformation information is typically stored with the image. Images store this information in a separate ASCII file. This file is generally referred to as the world file, since it contains the real-world transformation information used by the image.

World files are a simple mechanism for associating georeferencing (world coordinates) information with raster files. The world file contents look like the following. The first coefficient is the X pixel size. The second and third are rotational/shear coefficients (and should normally be 0.0). The fourth is the Y pixel size, normally negative indicating that Y decreases as you move down from the top left origin. The final two values are the X and Y location of the center of the top left pixel. This example is for an image (fig 2) with a 2m x 2m pixel size, and a top left origin at (673200E, 4000000N):

2

0.0000000000

0.0000000000

-2

673200.000000

4000000.000000

The name of the world file is based on the file it relates to. For instance, the world file for doq.jpg might be doq.jpgw.

4.3.2 World file naming conventions

It's easy to identify the world file, which should accompany an image file: world files use the same name as the image, with a "w" appended. For example, the world file for the image file mytown.tiff would be called mytown.tiffw and the world file for redlands.rlc would be redlands.rlcw.

4.3.3 How the georeferencing information is accessed

The image-to-world transformation is accessed each time an image is displayed (e.g., when you pan or zoom) and when a piece of vector data get overlayed over raster image. The image-to-world transformation is a six-parameter affine transformation in the form of:

$x1 = Ax + By + C$

$y1 = Dx + Ey + F$

Where

$x1$ = calculated x-coordinate of the pixel on the map

$y1$ = calculated y-coordinate of the pixel on the map

x = column number of a grid cell in the image

y = row number of a grid cell in the image

A = x-scale; dimension of a pixel in map units in x direction

B, D = rotation terms

C, F = translation terms; x, y map coordinates of the center of the upper-left pixel

E = negative of y-scale; dimension of a pixel in map units in y direction

The y-scale (E) is negative because the origins of an image and a geographic coordinate system are different. The origin of an image is located in the upper-left corner, whereas the origin of the map coordinate system is located in the lower-left corner. Row values in the image increase from the origin downward, while y-coordinate values in the map increase from the origin upward.

4.4 Clipping

Algorithm for Clipping:

This algorithm clips the target data with the help of Clipper file.

1. Define new Output Feature Class "Output"
2. Check projection method

    If Clipper.Projection is same as TargetFeature.Projection

        "OK" Go to Step 3.

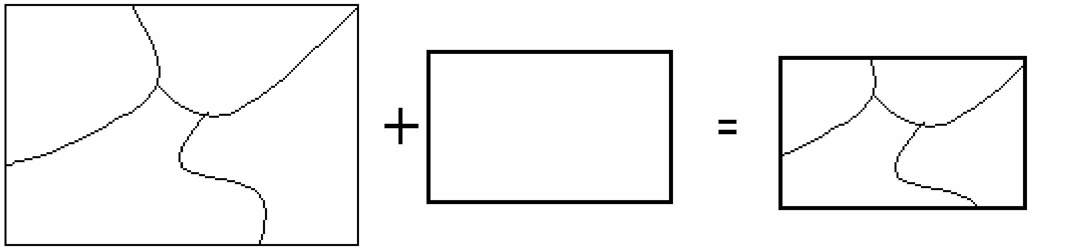
    Else

        "Error"

3. Perform Clipping operation

    Output = Clip (TargetFeature, Clipper)

Once we have clippers created we are ready to clip an existing shape file by using a polygon shape file (like using a cookie cutter) to create a new shape file. All features are clipped at the border of the cookie cutter. This has the advantage of not including anything outside your area of interest but the disadvantage of chopping some features up which perhaps should not be chopped (e.g., parcels). To use the clip method, we will need to have a polygon shape file, which will act as the "clipper" (the cookie cutter) in addition to the "clippee" shape files. This is a temporary shape file which will be covering boundary area, and then perform the clip as outlined below. We want to clip out only the portion that the user will need to cover the portion of image selected. The following Figure 12 shows the clipping method.



Clippee shapefile Clipper shapefile Output shapefile

Figure 12. Clipping

The following flowchart (Figure 13) shows the workflow design that gives detailed information of the solution proposed.

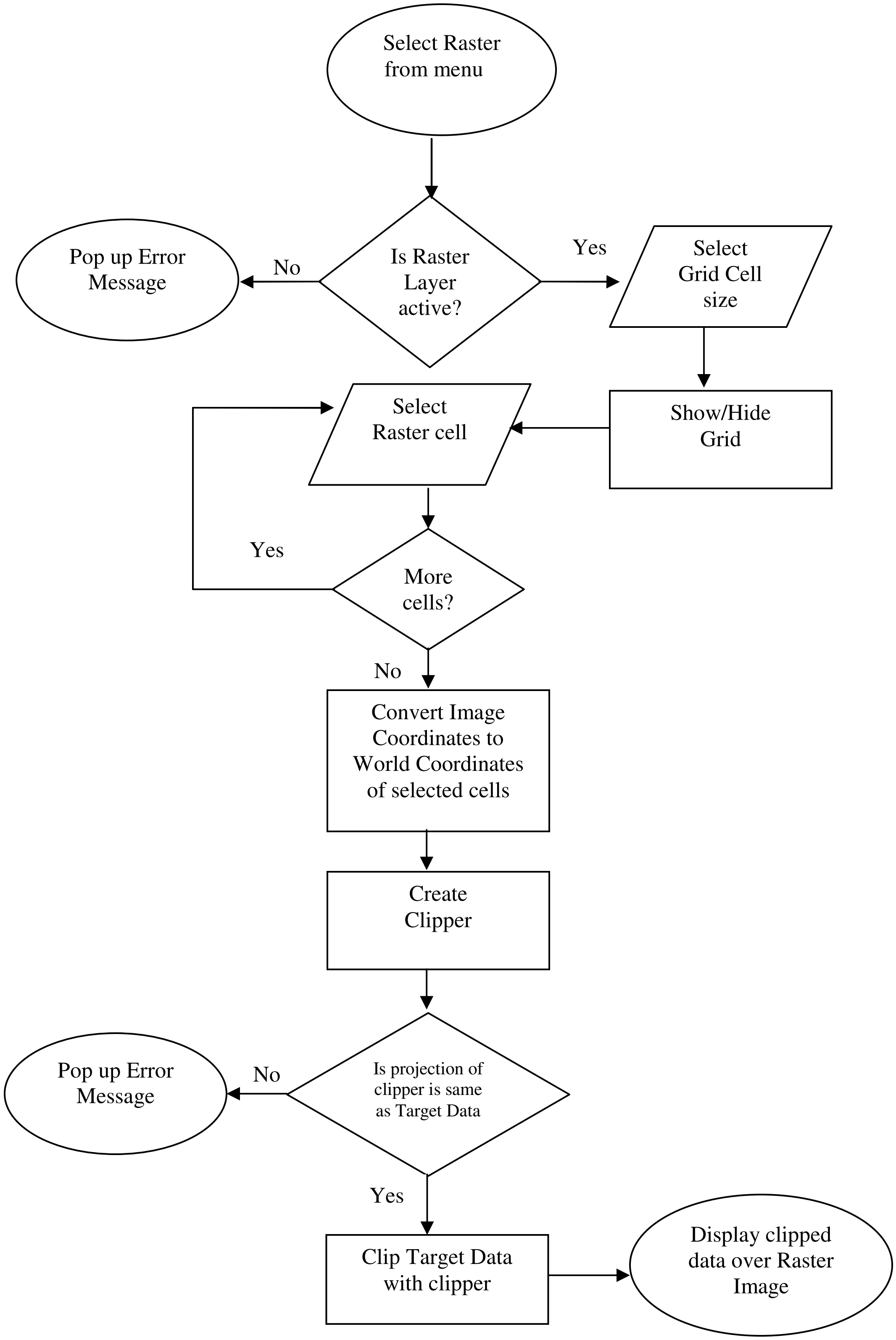


Figure 13. Flow Chart Design

CHAPTER V

IMPLEMENTATION AND RESULTS

The proposed method of selecting raster data cells and able to retrieve vector data for the selected cells has been successfully implemented and tested.

ArcObjects is the development platform used for this research work in conjunction with ArcGIS family of application such as ArcMap. The ArcObjects software components expose the full range of functionality available in ArcInfo and ArcView to software developers. The most common way that developers will customize the ArcGIS Desktop applications is through Visual Basic for Applications (VBA), which is embedded within ArcCatalog and ArcMap. Through VBA, we can leverage the application framework that already exists in ArcMap for general data management and map presentation tasks and extend ArcGIS with our own custom commands, tools, menus, and modules. VBA is not a standalone program. It provides an integrated programming environment, the VBE that lets us write a Visual Basic (VB) macro and debug it right away in ArcMap. A macro can integrate some or all of VB's functionality with the extensive object library available through ArcMap. The ESRI Object Library is always available to us in the VBA environment [1]. For specialized applications, developers with sufficient skill can bypass the application framework of ArcMap and ArcCatalog and instead build their own-targeted applications. The Map control provides a good point of entry, allowing access to the remainder of ArcObjects [6].

The programming Interfaces are used to create stand-alone executables or components that can plug into existing GIS application.

The following figures are screenshots of the customized application created in ArcMAP. All the data used in the research is provided by Oklahoma Center for Geospatial Information [14]. A raster layer image, which covers a part of City of Stillwater and vector layer of Roads, is added to the project as shown in Figure 14. A new toolbar with buttons and menus is created which runs the scripts written for this application.

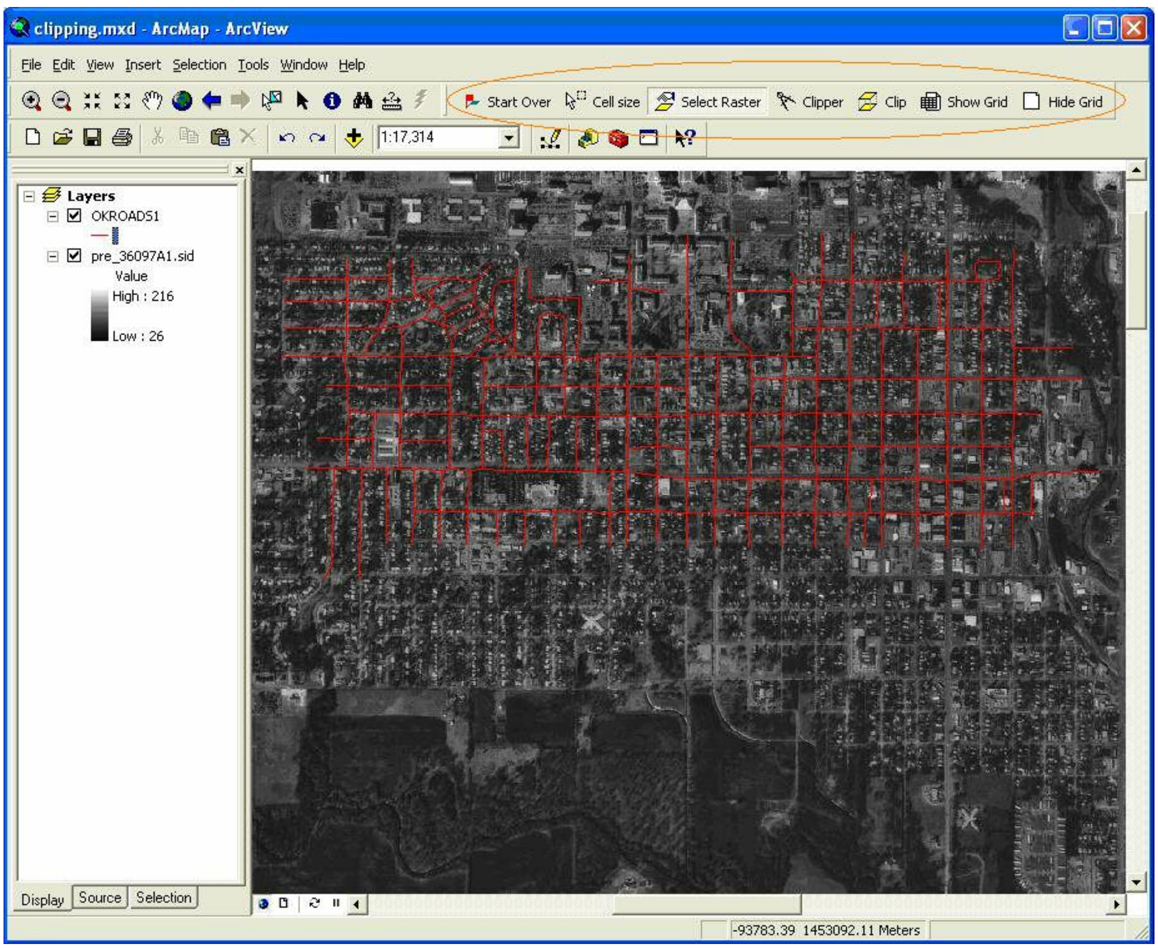


Figure 14. Customized Arcmap application

The user can set the cell size by clicking the Cell Size button. Once cell size is defined we can draw grid over the image by clicking Show Grid button as shown in figure 15.

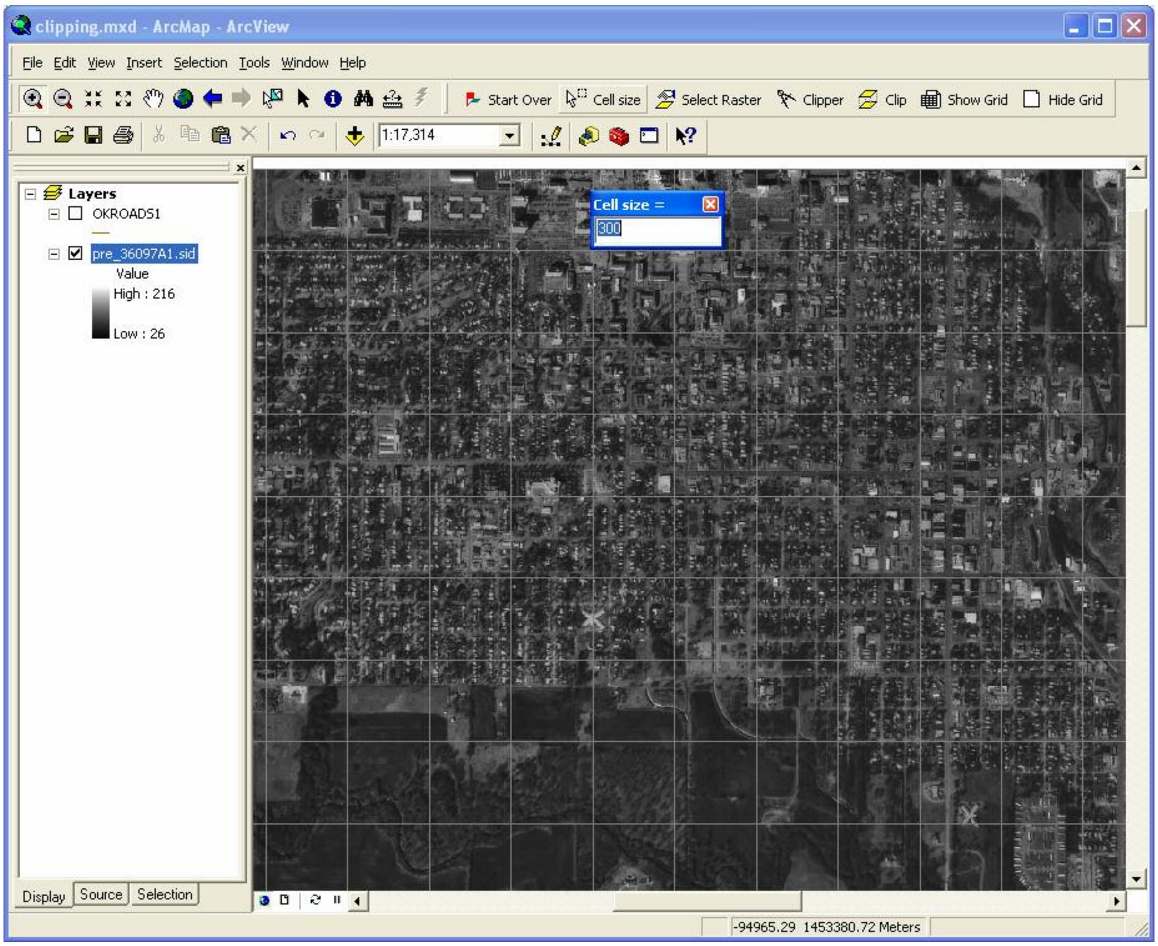


Figure 15. Select Cell size and Show Grid

By selecting Select Raster menu the user can pick cells according to his interested area over the image as shown in figure 16.

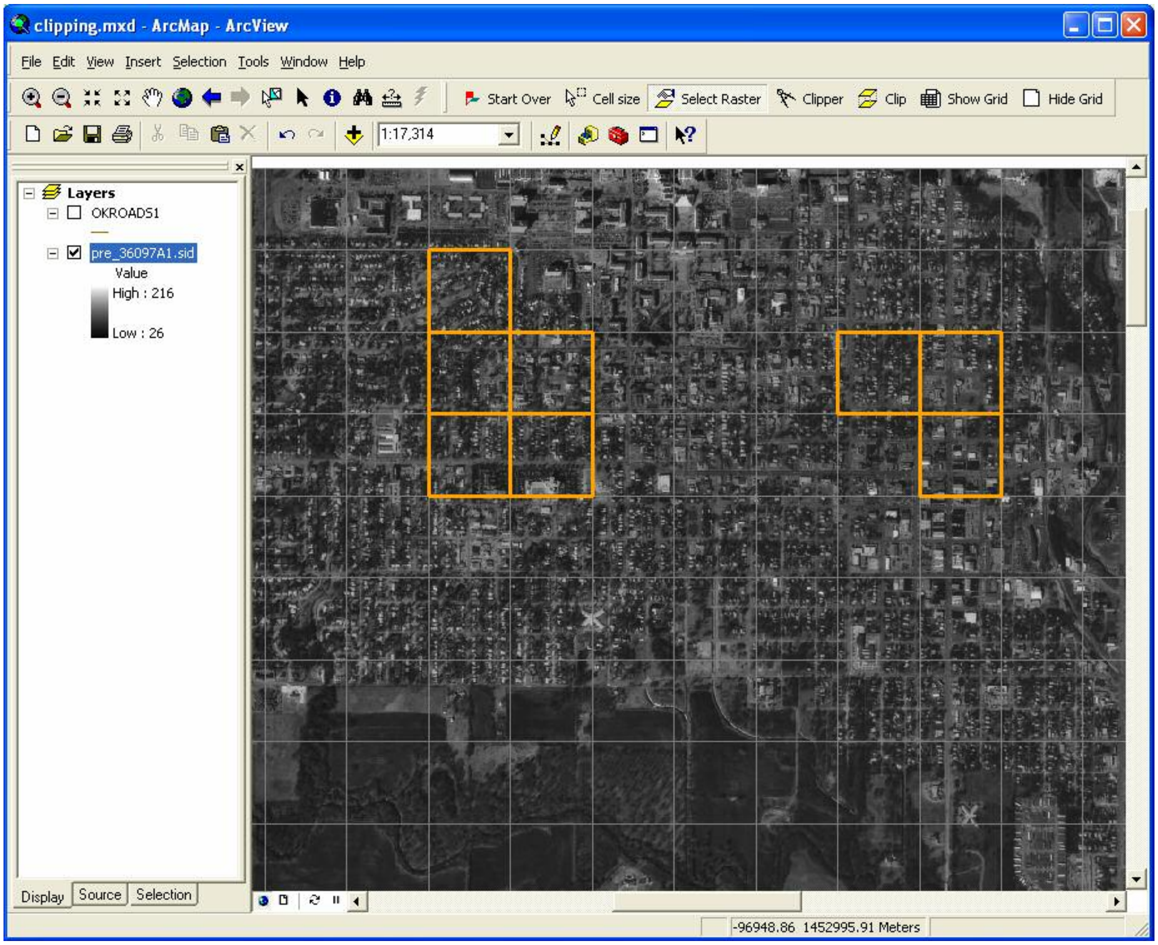


Figure 16. Selected Raster cells

Collecting selected cells in figure 17 after clicking Clipper button creates clipper polygon shapefile.

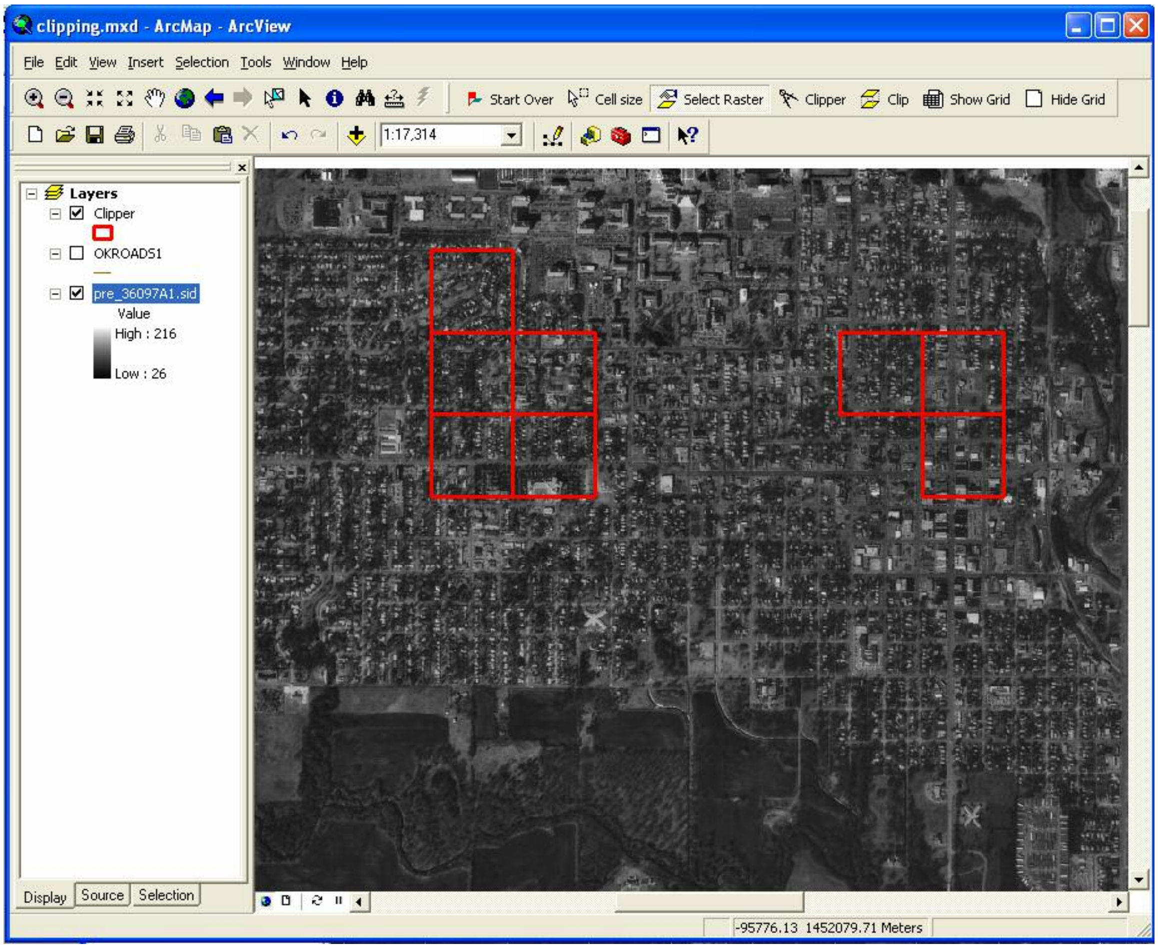


Figure 17. Clipper shapefile

Road layer shapefile is clipped by clipper shapefile and gives the desired output.

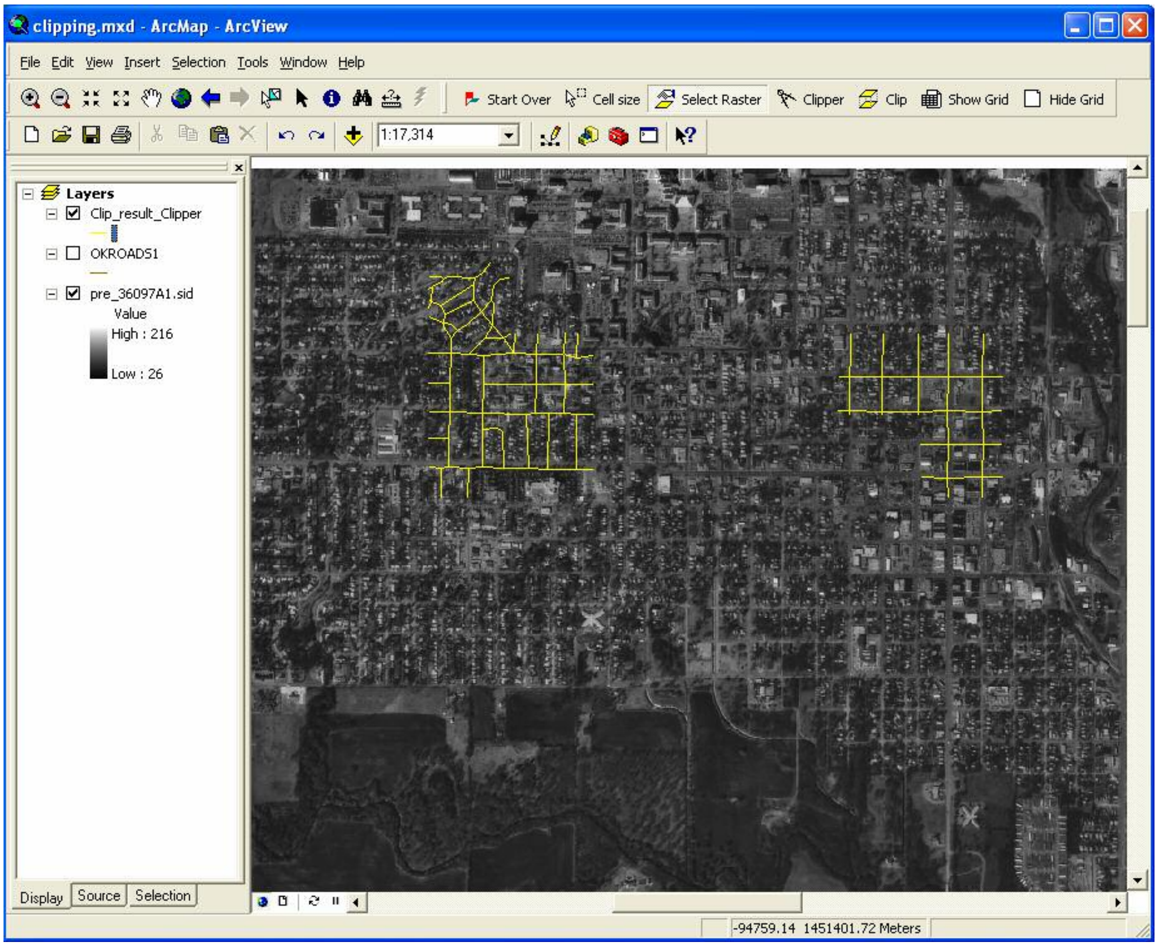


Figure 18. Vector data for the cells selected

CHAPTER VI

CHALLENGES

The following assumptions have been taken in this research work.

- The geo-coordinates of raster image is specified i.e. world file.
- The projection method used for raster and vector data is the same.

The grid based approach uses world file for reading the geographical coordinate of the selected cells. If an image lacks a world file, the user can create his own world file, using a text editor. This is generally practical only when the image does not require any rotation or rectification to be properly georeferenced. The ESRI ArcInfo commands REGISTER and RECTIFY, as well as GRIDIMAGE and CONVERTIMAGE create a world file.

The most challenging task of integrating maps with imagery is having a common projection method defined in the datasets. There is a wide variety of geo-spatial data available on the Internet that provides satellite imagery and maps of various regions. The National Map, MapQuest, University of Texas Map Library, Microsoft TerraService, and Space Imaging are good examples of map or satellite imagery repositories [5]. In addition, a wide variety of maps are available from various government agencies, such as property survey maps and maps of oil and natural gas fields. Satellite imagery and aerial photography have been utilized to enhance real estate listings, various military targeting applications, and other applications.

By integrating these spatial datasets, one can support a rich set of knowledge discovery queries that could not have been answered given any of these datasets in isolation. For example, when you are looking for a park in a neighborhood, the satellite imagery may provide you better view of the park, while the map is essential to see the surrounding streets and how to get to the park. However, accurately integrating maps and imagery from different data sources remains a challenging task. This is because spatial data obtained from various data sources may have different projections and different accuracy levels. If the geographic projections of these datasets are known, then they can be converted to the same geographic projections. However, the geographic projection for a wide variety of geo-spatial data available on the Internet is not known. The fact that many of the online maps sources do not provide the geo-coordinates of the maps makes the integration even more complicated.

REFERENCES

[1] ArcObjects Online, Copyright © Environmental Systems Research Institute, Inc. http://edndoc.esri.com/arcobjects/8.3/, Date of last access: Nov 1 2006

[2] Bajcsy Peter, Peter Groves, Sunayana Saha, and Tyler Alumbaugh, Decision Support Using Data Mining Methods and Geographic Information, Technical Report NCSA-ALG-03-0002, February 2003

[3] Bajcsy Peter, Tyler J. Alumbaugh, Georeferencing Maps With Contours, National Center for Supercomputing Applications, National Center for Supercomputing Applications, Technical Report NCSA-ALG-02-0001, October 2002

[4] Carswell James D., Using Raster Sketches For Digital Image Retrieval, Ph.D thesis in Spatial Information Science and Engineering, The University of Maine, May 2000

[5] Chen Ching-Chien, Craig A. Knoblock, Cyrus Shahabi, and Snehal Thakkar, Automatically and Accurately Conflating Satellite Imagery and Maps, University of Southern California, http://www.isi.edu/integration/papers/chen03-ng2i.pdf, Date of last access: Nov 13 2006

[6] Chudiwale Varun, Development Of Processes For Online GIS Data Selection, Extraction, And Aggregation Using ArcIMS and .NET Technology, Master thesis, Computer Science, Oklahoma State University, 2005

[7] ESRI, ArcGIS product description, http://www.esri.com/software/arcgis/index.html Date of last access: Oct 12 2006.

[8] ESRI, http://www.esri.com/software/arcgis/geodatabase/about/geometry.html, Date of last access: Nov 12 2006

[9] ESRI Shapefile Technical Description, An ESRI White Paper—July 1998, http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf, Date of last access: Nov 7 2006

[10] GIS Data Depot, GeoCommunity, http://data.geocomm.com/helpdesk/formats.html, Date of last access: Oct 15 2006

[11] GIS & Remote Sensing for conservation, Africanconservation.org, http://www.africanconservation.org/gis.html, Date of last access: Nov 18 2006

[12] Guidelines on the Application of GPS in Modern Mapping and GIS Technologies to Population Data, United Nations Economic and Social Commission for Asia and the Pacific (ESCAP), http://www.unescap.org/stat/pop-it/, Date of last access: Nov 16 2006

[13] Integrating Maps and Imagery, Information Integration Research Group of the Intelligent Systems Division of the Information Sciences Institute (ISI) http://www.isi.edu/integration/TerraWorld/, Date of last access: Nov 17 2006

[14] Oklahoma Center for Geospatial Information (OCGI) Department of Geography at Oklahoma State University (OSU), http://www.ocgi.okstate.edu/ Date of last access: Sept 1 2006

[15] Sheikholeslami Gholamhosein and Aidong Zhang Ling Bian, Geographical Image Classification and Retrieval, Proceedings of the 5th ACM international workshop on Advances in geographic information systems, Las Vegas, Nevada, United States, Pages: 58 - 61, Year of Publication: 1997

[16] Terraserver, Microsoft, http://www.terraserver.microsoft.com, Date of last access: Sept 2 2006

[17] Using Arcmap, The ArcGIS User manual ArcGIS 9 Copyright © 2000–2005 ESRI, http://www.gsd.harvard.edu/gis/manual/projections/tutorial/index.htm, Date of last access: May 12 2006

[18] What is GIS? The guide to Geographic Information Systems, GIS information on web, http://www.gis.com/whatisgis/index.cfm, Date of last access: Sept 14 2006.

APPENDIX A

GLOSSARY

Aerial photograph	A photograph of the earth's surface taken from a platform flying above the surface but not in orbit, usually an aircraft. Aerial photography is often used as a cartographic data source for basemapping, locating geographic features, and interpreting environmental conditions.
ArcIMS	ESRI software that allows for centrally hosting and serving GIS maps, data, and applications for use on the Internet. The administrative framework lets users author configuration files, publish maps, design Web pages, and administer ArcIMS spatial servers. ArcIMS supports Windows, Linux, and UNIX platforms and is customizable on many levels.
ArcObjects	A library of software components that make up the foundation of ArcGIS. ArcGIS Desktop, ArcGIS Engine, and ArcGIS Server are all built using the ArcObjects libraries
Bounding box	The rectangle, aligned with the coordinate axes and placed on a map display, that encompasses a geographic feature or group of features or an area of interest. It is defined by minimum and maximum coordinates in the x and y directions and is used to represent, in a general way, the location of a geographic area.

Cartesian coordinate system (in the context of this thesis)	A two-dimensional, planar coordinate system in which horizontal distance is measured along an x-axis and vertical distance is measured along a y-axis. Each point on the plane is defined by an x, y coordinate. Relative measures of distance, area, and direction are constant throughout the Cartesian coordinate plane.
Coordinate system	A reference framework consisting of a set of points, lines, and/or surfaces, and a set of rules, used to define the positions of points in space in either two or three dimensions. The Cartesian coordinate system and the geographic coordinate system used on the earth's surface are common examples of coordinate systems.
Cell	The smallest unit of information in raster data, usually square in shape. In a map or GIS dataset, each cell represents a portion of the earth, such as a square meter or square mile, and usually has an attribute value associated with it, such as soil type or vegetation class.
Clip	A command that extracts features from one feature class that resides entirely within a boundary defined by features in another feature class.
Feature	A representation of a real-world object on a map. For example Point, Line and Polygon.
Feature Class	A collection of geographic features with the same geometry type (such as point, line, or polygon), the same attributes, and the same spatial reference. Feature classes can be stored in geodatabases,

	shapefiles, coverages, or other data formats. Feature classes allow homogeneous features to be grouped into a single unit for data storage purposes. For example, highways, primary roads, and secondary roads can be grouped into a line feature class named "roads."
Georeferencing	Aligning geographic data to a known coordinate system so it can be viewed, queried, and analyzed with other geographic data. Georeferencing may involve shifting, rotating, scaling, skewing, and in some cases warping, rubber sheeting, or orthorectifying the data.
GIS	Acronym for *geographic information system*. An integrated collection of computer software and data used to view and manage information about geographic places, analyze spatial relationships, and model spatial processes. A GIS provides a framework for gathering and organizing spatial data and related information so that it can be displayed and analyzed.
Grid	In cartography, any network of parallel and perpendicular lines superimposed on a map and used for reference. These grids are usually referred to by the map projection or coordinate system they represent, such as universal transverse Mercator grid.
Identify	In ArcGIS, a tool that, when applied to a feature (by clicking it), opens a window showing that feature's attributes.
Latitude	The angular distance, usually measured in degrees north or south of the equator. Lines of latitude are also referred to as parallels.

Layer	The visual representation of a geographic dataset in any digital map environment. Conceptually, a layer is a slice or stratum of the geographic reality in a particular area, and is more or less equivalent to a legend item on a paper map. On a road map, for example, roads, national parks, political boundaries, and rivers might be considered different layers.
Longitude	The angular distance, usually expressed in degrees, minutes, and seconds, of the location of a point on the earth's surface east or west of an arbitrarily defined meridian (usually the Greenwich prime meridian). All lines of longitude are great circles that intersect the equator and pass through the North and South Poles.
Map	A graphic representation of the spatial relationships of entities within an area.
Orthograph	An aerial photograph from which distortions owing to camera tilt and ground relief have been removed. An orthophotograph has the same scale throughout and can be used as a map.
Orthophotoquad	An orthophotograph that has been formatted as a USGS 1:24,000 topographic quadrangle with little or no cartographic enhancement.
Overlay	A spatial operation in which two or more maps or layers registered to a common coordinate system are superimposed, either digitally or on a transparent material, for the purpose of showing the relationships between features that occupy the same geographic space.

Raster data model	A representation of the world as a surface divided into a regular grid of cells. Raster models are useful for storing data that varies continuously, as in an aerial photograph, a satellite image, a surface of chemical concentrations, or an elevation surface.
Remote Sensing	Collecting and interpreting information about the environment and the surface of the earth from a distance, primarily by sensing radiation that is naturally emitted or reflected by the earth's surface or from the atmosphere, or by sensing signals transmitted from a device and reflected back to it. Examples of remote-sensing methods include aerial photography, radar, and satellite imaging.
Spatial	Related to or existing within space.
Theme	A set of related geographic features such as streets, parcels, or rivers, along with their attributes. All features in a theme share the same coordinate system, are located within a common geographic extent, and have the same attributes.
Vector data model	A representation of the world using points, lines, and polygons. Vector models are useful for storing data that has discrete boundaries, such as country borders, land parcels, and streets.
Shapefile	A vector data storage format for storing the location, shape, and attributes of geographic features. A shapefile is stored in a set of related files and contains one feature class.
UTM	Acronym for *universal transverse Mercator*. A projected coordinate system that divides the world into 60 north and south zones, 6

degrees wide.

World file — A text file containing information about where an image should be displayed in real-world coordinates. When an image has a properly configured world file, GIS software can use the information (a total of six values, including the starting coordinates, the cell size in both x- and y-dimensions, and any rotation and scaling information) to accurately overlay the image with any other data already in a projected or geographic coordinate system.

APPENDIX B

PROGRAM LISTING

The customized Arcmap application is written in VBA and is listed as follows.

' The purpose of this application is to give the user of a GIS application ability to select ' Raster image cells and then able to fetch the vector data associated with the selected ' cells.

```
' Select Raster Cells

Public rowcollist() As String
Public cellcount As Integer
Public cellsize As Integer
Public totalcells As Long
Public pEnv As Envelope
Public lCols As Long
Public lRows As Long

Private Sub SelectRaster_MouseDown(ByVal button As Long, ByVal shift As Long,
ByVal X As Long, ByVal Y As Long)

   If (cellsize <= 0) Then
      MsgBox "select cell size", vbExclamation
      Exit Sub
   End If
' x and y define the point the user clicked in device units
   Dim pPoint As IPoint
   Dim pApp As IMxApplication
   Set pApp = Application
   Set pPoint = pApp.Display.DisplayTransformation.ToMapPoint(X, Y)
  ' Get the Map
   Dim pMxDoc As IMxDocument
   Set pMxDoc = ThisDocument
   Dim pMap As IMap
```

```
Set pMap = pMxDoc.FocusMap
    'Create a raster layer
    Dim pRasterLy As IRasterLayer
    Set pRasterLy = New RasterLayer
    Set pRasterLy = pMap.Layer(FindRasterLayer)
    ' Set raster property
    Dim pRasProp As IRasterProps
    Set pRasProp = pRasterLy.Raster
    Dim height, width As Integer
    Dim Col, Row As Integer
    If (pPoint.X > pRasProp.Extent.XMax) Or (pPoint.X < pRasProp.Extent.XMin) Then
        Exit Sub
    End If
    If (pPoint.Y > pRasProp.Extent.YMax) Or (pPoint.Y < pRasProp.Extent.YMin) Then
        Exit Sub
    End If
    height = pRasProp.height
    width = pRasProp.width
    Row = UpIt(Abs(pPoint.Y - pRasProp.Extent.YMax) / cellsize)
    Col = UpIt(Abs(pPoint.X - pRasProp.Extent.XMin) / cellsize)
    Dim iEnv As IEnvelope
    Set iEnv = New Envelope

    iEnv.PutCoords pRasProp.Extent.XMin + (Col - 1) * cellsize, pRasProp.Extent.YMax
- Row * cellsize, pRasProp.Extent.XMin + Col * cellsize, pRasProp.Extent.YMax - (Row
- 1) * cellsize

    Dim pElement As IElement
    Dim pGC As IGraphicsContainer
    Dim pActiveView As IActiveView
    Dim pScreenDisplay As IScreenDisplay

    Set pMxDoc = Application.Document
    Set pActiveView = pMxDoc.FocusMap
    Set pGC = pMap
    Set pElement = New RectangleElement
    pElement.Geometry = iEnv

    Dim pFillShapeElem As IFillShapeElement
    Set pFillShapeElem = New RectangleElement
    Set pFillShapeElem = pElement

    'Create a fill symbol for the elem
    Dim pFillSym As ISimpleFillSymbol
    Set pFillSym = New SimpleFillSymbol
```

```
    Dim pRGB As IRgbColor
    Set pRGB = New RgbColor
    With pRGB
        .Red = 255
        .Green = 165
        .Blue = 0
    End With


    Dim pOutline As ILineSymbol
    Set pOutline = New SimpleLineSymbol
    pOutline.Color = pRGB
    pOutline.width = 2


    pFillSym.Outline = pOutline
    pFillSym.Style = esriSFSHollow
   pFillShapeElem.Symbol = pFillSym

    'Add the element to the map
    pGC.AddElement pFillShapeElem, 0

    'Refresh the map
    Set pActiveView = pMap
    pActiveView.PartialRefresh esriViewGraphics, pElement, Nothing
    rowcollist(cellcount) = Row & "," & Col
    cellcount = cellcount + 1
End Sub

Private Function MergeClipper() As Boolean
On Error GoTo EH
' Get the first layer in the map
    Dim pMxDoc As IMxDocument
    Set pMxDoc = ThisDocument
    Dim pLayer As ILayer

    Dim inputArray As esriSystem.IArray
    Set inputArray = New esriSystem.Array
    Dim Col, Row As Integer
    Dim RowCol() As String

    RowCol = Split(rowcollist(0), ",")
    Row = CInt(RowCol(0))
    Col = CInt(RowCol(1))

    'Create a new ShapefileWorkspaceFactory object and open a shapefile folder
```

```
    Dim pWorkspaceFactory As IWorkspaceFactory
    Dim pFeatureWorkspace As IFeatureWorkspace
    Set pWorkspaceFactory = New ShapefileWorkspaceFactory
    Set pFeatureWorkspace = pWorkspaceFactory.OpenFromFile("C:\TEMP\shape\", 0)
    'Create a new FeatureLayer and assign a shapefile to it
    Dim pFeatLayer As IFeatureLayer
    Set pFeatLayer = New FeatureLayer
    Set pFeatLayer.FeatureClass = pFeatureWorkspace.OpenFeatureClass("Clipper_" +
CStr(Row) + "," + CStr(Col))
    pFeatLayer.Name = pFeatLayer.FeatureClass.AliasName


    'Set pLayer = pMxDoc.FocusMap.Layer(0)
    'Dim pFeatLayer As IFeatureLayer
    Set pLayer = pFeatLayer
    Dim pFirstFeatClass As IFeatureClass
    Set pFirstFeatClass = pFeatLayer.FeatureClass

    ' Get the first layer's table
    ' Use the Itable interface from the Layer (not from the FeatureClass)
    ' This table defines which fields are to be used in the output
    Dim pFirstTable As ITable
    Set pFirstTable = pLayer

    ' Error checking
    If pFirstTable Is Nothing Then
        MsgBox "Table QI failed"
        Exit Function
    End If

    inputArray.Add pFirstTable

        Dim i  As Integer
    For i = 1 To cellcount - 1
        RowCol = Split(rowcollist(i), ",")
        Row = CInt(RowCol(0))
        Col = CInt(RowCol(1))

        ' Get the second layer and its table
        ' Use the Itable interface from the Layer (not from the FeatureClass)
        'Set pLayer = pMxDoc.FocusMap.Layer(1)
        Set pFeatLayer = New FeatureLayer
        Set pFeatLayer.FeatureClass = pFeatureWorkspace.OpenFeatureClass("Clipper_" +
CStr(Row) + "," + CStr(Col))
        pFeatLayer.Name = pFeatLayer.FeatureClass.AliasName
        Set pLayer = pFeatLayer
```

```
    Dim pSecondTable As ITable
    Set pSecondTable = pLayer
    ' Error checking
    If pSecondTable Is Nothing Then
        MsgBox "Table QI failed"
        Exit Function
    End If
     ' Build the input set/array - these are the layers to be merged

    inputArray.Add pSecondTable
Next i

' Define the output feature class name and shape type
Dim pFeatClassName As IFeatureClassName
Set pFeatClassName = New FeatureClassName

With pFeatClassName
    .FeatureType = esriFTSimple
    .ShapeFieldName = "Shape"
    .ShapeType = pFirstFeatClass.ShapeType
End With

' Set the output location and feature class name
Dim pNewWSName As IWorkspaceName
Set pNewWSName = New WorkspaceName

With pNewWSName
    .WorkspaceFactoryProgID = "esriCore.ShapefileWorkspaceFactory.1"
    .PathName = "C:\TEMP\shape\"
End With

Dim pDatasetName As IDatasetName
Set pDatasetName = pFeatClassName
pDatasetName.Name = "Clipper"

Set pDatasetName.WorkspaceName = pNewWSName



' Perform the merge
Dim pBGP As IBasicGeoprocessor
Set pBGP = New BasicGeoprocessor
Dim pOutputFeatClass As IFeatureClass
Set pOutputFeatClass = pBGP.Merge(inputArray, pFirstTable, pFeatClassName)

' Add the output to the map
```

```
   Dim pOutputFeatLayer As IFeatureLayer
   Set pOutputFeatLayer = New FeatureLayer
   Set pOutputFeatLayer.FeatureClass = pOutputFeatClass
   pOutputFeatLayer.Name = pOutputFeatClass.AliasName
   pMxDoc.FocusMap.AddLayer pOutputFeatLayer

   Dim temp As Object
   Set temp = CreateObject("Scripting.FileSystemObject")

    For i = 0 To cellcount - 1
      RowCol = Split(rowcollist(i), ",")
      Row = CInt(RowCol(0))
      Col = CInt(RowCol(1))


         If temp.FileExists("C:\TEMP\shape\Clipper_" + CStr(Row) + "," + CStr(Col) +
".shp") Then
           temp.DeleteFile ("C:\TEMP\shape\Clipper_" + CStr(Row) + "," + CStr(Col) +
".*")
         End If

   Next i

   'remove selected cells drawings
   Dim pGC As IGraphicsContainer
   Set pGC = pMxDoc.FocusMap
   pGC.DeleteAllElements

   MergeClipper = True
   Exit Function
EH:
   MsgBox "error : " & Err.Source & "," & Err.Description
End Function

Public Function FindRasterLayer() As Long
   Dim pMxDoc As IMxDocument
   Set pMxDoc = ThisDocument

   Dim pMap As IMap
   Set pMap = pMxDoc.FocusMap
   Dim idx As Long
   idx = 0
   Dim pEnumLayer As IEnumLayer
   Set pEnumLayer = pMap.Layers

   Dim pLayer As ILayer
```

```
    Dim pLayerReq As ILayer

    Set pLayer = pEnumLayer.Next
    Do
        If pLayer Is Nothing Then Exit Do
        If TypeOf pLayer Is IRasterLayer Then
            Exit Do
        End If
        idx = idx + 1
        Set pLayer = pEnumLayer.Next
    Loop
     FindRasterLayer = idx
End Function
Private Function createShapefile() As Boolean
    'create shapefile from a cell drawn
    Const strFolder As String = "C:\TEMP\shape\"
    Const strShapeFieldName As String = "Shape"
    Dim strName As String
    Dim Col, Row As Integer
    Dim RowCol() As String

    Dim pMxDoc As IMxDocument
    Set pMxDoc = ThisDocument
    Dim pMap As IMap
    Set pMap = pMxDoc.FocusMap
    'Create a raster layer

    Dim pRasterLy As IRasterLayer
    Set pRasterLy = New RasterLayer
    Set pRasterLy = pMap.Layer(FindRasterLayer())
    ' Set raster property
    Dim pRasProp As IRasterProps
    Set pRasProp = pRasterLy.Raster
    Dim i  As Integer
    For i = 0 To cellcount - 1
        RowCol = Split(rowcollist(i), ",")
        Row = CInt(RowCol(0))
        Col = CInt(RowCol(1))
        Dim pPointCollection As IPointCollection
        Set pPointCollection = New Polygon
        Dim pPoint1 As IPoint
        Set pPoint1 = New Point
        pPoint1.PutCoords pRasProp.Extent.XMin + (Col - 1) * cellsize,
pRasProp.Extent.YMax - Row * cellsize
        pPointCollection.AddPoint pPoint1
        Dim pPoint2 As IPoint
```

```
    Set pPoint2 = New Point
    pPoint2.PutCoords pRasProp.Extent.XMin + (Col - 1) * cellsize,
pRasProp.Extent.YMax - (Row - 1) * cellsize
    pPointCollection.AddPoint pPoint2
    Dim pPoint3 As IPoint
    Set pPoint3 = New Point
    pPoint3.PutCoords pRasProp.Extent.XMin + Col * cellsize, pRasProp.Extent.YMax
- (Row - 1) * cellsize
    pPointCollection.AddPoint pPoint3
    Dim pPoint4 As IPoint
    Set pPoint4 = New Point
    pPoint4.PutCoords pRasProp.Extent.XMin + Col * cellsize, pRasProp.Extent.YMax
- Row * cellsize
    pPointCollection.AddPoint pPoint4

    pPointCollection.AddPoint pPoint1

    Dim pPolygon As IPolygon
    Set pPolygon = pPointCollection

    ''.....................................................................
    ''create a new Shapefile

    strName = "Clipper_" + CStr(Row) + "," + CStr(Col)
    Dim file1 As New FileStream
    Dim temp As Object
    Set temp = CreateObject("Scripting.FileSystemObject")
    If (Dir("C:\TEMP\shape\" + strName + ".*") <> "") Then
      temp.DeleteFile ("C:\TEMP\shape\" + strName + ".*")
    End If



    ' Open the folder to contain the shapefile as a workspace
    Dim pFWS As IFeatureWorkspace
    Dim pWorkspaceFactory As IWorkspaceFactory
    Set pWorkspaceFactory = New ShapefileWorkspaceFactory
    Set pFWS = pWorkspaceFactory.OpenFromFile(strFolder, 0)


    ' Set up a simple fields collection
    Dim pFields As IFields
    Dim pFieldsEdit As IFieldsEdit
    Set pFields = New Fields
    Set pFieldsEdit = pFields
```

```
Dim pField As IField
Dim pFieldEdit As IFieldEdit

' Make the shape field
' it will need a geometry definition, with a spatial reference
Set pField = New Field
Set pFieldEdit = pField
pFieldEdit.Name = strShapeFieldName
pFieldEdit.Type = esriFieldTypeGeometry

Dim pGeomDef As IGeometryDef
Dim pGeomDefEdit As IGeometryDefEdit
Set pGeomDef = New GeometryDef
Set pGeomDefEdit = pGeomDef

'MsgBox pMxDoc.FocusMap.SpatialReference.Name

Dim pSpatRefFact As SpatialReferenceEnvironment
Set pSpatRefFact = New SpatialReferenceEnvironment
Dim m_pWGS1984 As ISpatialReference
Set m_pWGS1984 = pMxDoc.FocusMap.SpatialReference '
pSpatRefFact.CreateGeographicCoordinateSystem(esriSRGeoCS_WGS1984)

With pGeomDefEdit
.GeometryType = esriGeometryPolygon
Set .SpatialReference = m_pWGS1984
End With
Set pFieldEdit.GeometryDef = pGeomDef
pFieldsEdit.AddField pField

' Add another miscellaneous text field
Set pField = New Field
Set pFieldEdit = pField
With pFieldEdit
.Length = 30
.Name = "Text"
.Type = esriFieldTypeString
End With
pFieldsEdit.AddField pField

' Create the shapefile
' (some parameters apply to geodatabase options and can be defaulted as Nothing)
Dim pFeatClass As IFeatureClass
Set pFeatClass = pFWS.CreateFeatureClass(strName, pFields, Nothing, _
                        Nothing, esriFTSimple, strShapeFieldName, "")
pPolygon.Envelope.width = 3
```

```
        Dim pFeature As IFeature
        Set pFeature = pFeatClass.CreateFeature 'This creates a new blank feature
        Set pFeature.Shape = pPolygon  'set its shape to the rubber polygon

        pFeature.Store 'now store
        pPolygon.Close
    Next i
    MsgBox "Clipper shape file created."
    createShapefile = True
End  Function


Private Sub btnClip_Click()
If (cellsize <= 0) Then
        MsgBox "select cell size", vbExclamation
        Exit Sub
    End If
Dim pMxDoc As IMxDocument
  Set pMxDoc = ThisDocument
'  'remove old clip result layer
  Dim pLayerDel As Ilayer
'  Set pLayerDel = FindLayerRef("Clip_result_Clipper") ' Pass the requied layer name.
'  If Not pLayerDel Is Nothing Then
'    pMxDoc.FocusMap.DeleteLayer pLayerDel
'  End If
'
'  'delete old clip result file
  Dim temp As Object
  Set temp = CreateObject("Scripting.FileSystemObject")
  If temp.FileExists("C:\TEMP\shape\Clip_result_Clipper.shp") Then
    temp.DeleteFile ("C:\TEMP\shape\Clip_result_Clipper.*")
  End If
'

' Get the input layer and feature class.

  Dim pLayer As ILayer
  'Set pLayer = pMxDoc.FocusMap.Layer(0)
  Set pLayer = FindLayerRef("OKROADS1")
  Dim pInputFeatLayer As IFeatureLayer
  Set pInputFeatLayer = pLayer
  Dim pInputTable As ITable
  Set pInputTable = pLayer

  ' Get the input feature class.
  ' Use the Itable interface from the Layer (not from the FeatureClass)
```

```
' The Input feature class properties, such as shape type,
' will be needed for the output.
Dim pInputFeatClass As IFeatureClass
Set pInputFeatClass = pInputFeatLayer.FeatureClass

' Get the clip/overlay layer
' Use the Itable interface from the Layer (not from the FeatureClass)
'Set pLayer = pMxDoc.FocusMap.Layer(1)
Set pLayer = FindLayerRef("Clipper")
Dim pClipTable As ITable
Set pClipTable = pLayer

' Error checking
If pInputTable Is Nothing Then
   MsgBox "Table QI failed"
   Exit Sub
End If

If pClipTable Is Nothing Then
   MsgBox "Table QI failed"
   Exit Sub
End If

' Define the output feature class name and shape type (taken from the
' properties of the input feature class)
Dim pFeatClassName As IFeatureClassName
Set pFeatClassName = New FeatureClassName
With pFeatClassName
   .FeatureType = esriFTSimple
   .ShapeFieldName = "Shape"
   .ShapeType = pInputFeatClass.ShapeType
End With

' Set output location and feature class name
Dim pNewWSName As IWorkspaceName
Set pNewWSName = New WorkspaceName
pNewWSName.WorkspaceFactoryProgID = "esriCore.ShapeFileWorkspaceFactory.1"
pNewWSName.PathName = "C:\TEMP\shape\"

Dim pDatasetName As IDatasetName
Set pDatasetName = pFeatClassName
pDatasetName.Name = "Clip_result_" + pLayer.Name

Set pDatasetName.WorkspaceName = pNewWSName
```

```
 ' Set the tolerance.  Passing 0.0 causes the default tolerance to be used.
 ' The default tolerance is 1/10,000 of the extent of the data frame's spatial domain
 Dim tol As Double
 tol = 0#

 ' Perform the clip
 Dim pBGP As IBasicGeoprocessor
 Set pBGP = New BasicGeoprocessor
 Dim pOutputFeatClass As IFeatureClass
 Set pOutputFeatClass = pBGP.Clip(pInputTable, False, pClipTable, False, _
   tol, pFeatClassName)

 ' Add the output layer (clipped features) to the map
 Dim pOutputFeatLayer As IFeatureLayer
 Set pOutputFeatLayer = New FeatureLayer
 Set pOutputFeatLayer.FeatureClass = pOutputFeatClass
 pOutputFeatLayer.Name = pOutputFeatClass.AliasName
 pMxDoc.FocusMap.AddLayer pOutputFeatLayer

 'remove the clpper layer
 Set pLayerDel = FindLayerRef("Clipper") ' Pass the requied layer name.
 If Not pLayerDel Is Nothing Then
   pMxDoc.FocusMap.DeleteLayer pLayerDel
 End If

 'delete clipper file

 Set temp = CreateObject("Scripting.FileSystemObject")
 temp.DeleteFile ("C:\TEMP\shape\Clipper.*")
 cellcount = 0
End Sub

Public Function FindLayerRef(LayerName As String) As ILayer

   Dim pMxDoc As IMxDocument
   Set pMxDoc = ThisDocument

   Dim pMap As IMap
   Set pMap = pMxDoc.FocusMap

   Dim pEnumLayer As IEnumLayer
   Set pEnumLayer = pMap.Layers
```

```
    Dim pLayer As ILayer
    Dim pLayerReq As ILayer

    Set pLayer = pEnumLayer.Next
    Do
        If pLayer Is Nothing Then Exit Do
        If pLayer.Name = LayerName Then
            Set pLayerReq = pLayer
        End If
        Set pLayer = pEnumLayer.Next
    Loop

    Set FindLayerRef = pLayerReq
End Function


Private Sub btnCreateGrid_Click()
    If cellsize <= 0 Then
        MsgBox "select cell size", vbExclamation
        Exit Sub
    End If
    Dim pMyDoc As IMxDocument
    Set pMyDoc = Application.Document
    Dim pRasterLy As IRasterLayer
    Set pRasterLy = New RasterLayer

    Dim pMap As IMap
    Set pMap = pMyDoc.FocusMap
    Set pRasterLy = pMap.Layer(FindRasterLayer())
    ' Set raster property
    Dim pRasProp As IRasterProps
    Set pRasProp = pRasterLy.Raster
    Dim pEnv As IEnvelope
    Set pEnv = New Envelope
    Set pEnv = pRasProp.Extent.Envelope

    Dim pPoint2 As IPoint
    Dim pPoint3 As IPoint
    Dim pLine As ILine
    Dim pElement As IElement
    Dim pGC As IGraphicsContainer
    Dim pAV As IActiveView
    Dim pSegmentCollection As ISegmentCollection

    Set pGC = pMyDoc.FocusMap
    Dim i As Integer
```

```
  i = 0
   Dim pLineElement As ILineElement
    Dim pRGB As IRgbColor
  Set pRGB = New RgbColor
  With pRGB
     .Red = 127
     .Green = 127
     .Blue = 127
  End With
Dim pLineSymbol As ILineSymbol
  Do
     If i = lCols Then Exit Do
     'create line
     Set pLine = New Line
     Set pPoint2 = New Point
     Set pPoint3 = New Point
     pPoint2.PutCoords pEnv.XMin + (cellsize * i), pEnv.YMax
     pPoint3.PutCoords pEnv.XMin + (cellsize * i), pEnv.YMin
     pLine.FromPoint = pPoint2
     pLine.ToPoint = pPoint3
     'add line to the Segmentcollection
     Set pSegmentCollection = New Polyline
     pSegmentCollection.AddSegment pLine
     'add polyline to the graphicscontainer

     Set pLineElement = New LineElement
     Set pElement = pLineElement
     pElement.Geometry = pSegmentCollection

     ' Create a line symbol

     Set pLineSymbol = New SimpleLineSymbol
     pLineSymbol.width = 1
     pLineSymbol.Color = pRGB
     ' Set line style
     pLineElement.Symbol = pLineSymbol

     pGC.AddElement pElement, 0
     i = i + 1
  Loop
  i = 0
  'rows
  Do
     If i = lRows Then Exit Do
     'create line
     Set pLine = New Line
```

```
        Set pPoint2 = New Point
        Set pPoint3 = New Point
        pPoint2.PutCoords pEnv.XMin, pEnv.YMax - (cellsize * i)
        pPoint3.PutCoords pEnv.XMax, pEnv.YMax - (cellsize * i)
        pLine.FromPoint = pPoint2
        pLine.ToPoint = pPoint3
        'add line to the Segmentcollection
        Set pSegmentCollection = New Polyline
        pSegmentCollection.AddSegment pLine
        'add polyline to the graphicscontainer
        Set pLineElement = New LineElement
        Set pElement = pLineElement
        pElement.Geometry = pSegmentCollection

        ' Create a line symbol

        Set pLineSymbol = New SimpleLineSymbol
        pLineSymbol.width = 1
        pLineSymbol.Color = pRGB
        ' Set line style
        pLineElement.Symbol = pLineSymbol
        pGC.AddElement pElement, 0
        i = i + 1
    Loop
    'refresh activeview
    Set pAV = pMyDoc.FocusMap
    pAV.PartialRefresh esriViewGraphics, Nothing, Nothing

End Sub
```

VITA

Amar Gupta

Candidate for the Degree of

Master of Science

Thesis: GRID BASED RASTER SELECTION AND VECTOR EXTRACTION

Major Field: Computer Science

Biographical:

Personal Data: Born in Indore, India on September 24, 1979, son of Bhawar Lal Gupta and Girja Gupta.

Education: Received the degree of Bachelors of Engineering in Computer Engineering from Institute of Engineering & Technology, D.A.V.V, Indore, India, in May 2002; completed the requirements for the Master of Science degree at Computer Science department at Oklahoma State University, Stillwater, Oklahoma, in December 2006.

Experience: Research Assistant in Computer Science department, Oklahoma State University, from May 2003 to May 2004. Programmer at Biology Department, Oklahoma State University, from July 2003 to May 2004. GIS Engineer at SST Development Group, Stillwater, Oklahoma, from July 2004 to present.

Name: Amar Gupta Date of Degree: December, 2006

Institution: Oklahoma State University Location: Stillwater, Oklahoma

Title of Study: GRID BASED RASTER SELECTION AND VECTOR EXTRACTION

Pages in Study: 68 Candidate for the Degree of Master of Science

Major Field: Computer Science

Scope and Method of Study: The increased volume of raster imagery necessitates the development of methods to effectively retrieve data associated with it. The use of digital imagery is ever increasing for GIS (Geographical Information System) applications due to enabling technologies to integrate digital imagery with land related digital data. Integration of imagery with vector data is now a necessity for a full-featured GIS system. Imagery was once thought to be the exclusive domain of image processing systems, but is now often required as a backdrop for vector, or other data types. This research addresses the limited capability of current GIS applications while working with raster images. It proposes a method of giving the user more control over raster images and be able to retrieve data associated with it. The current GIS tools provide various ways to select vector data interactively by the user. However, selecting raster data from the GIS application is an area still to be explored by researchers. In this thesis we propose an effective grid based approach to select an Area of Interest (AoI) on the raster image and extract vector data from it. The new contribution of this work combines the design of a customized GIS application that lets the user select raster image cells and retrieve vector data that falls into the geography of selected cells.

Findings and Conclusion: The proposed method of selecting raster data cells and retrieving the associated vector data for the selected cells has been successfully implemented and tested. Arcobjects is the development platform used for this research work in conjunction with the ArcGIS family of application such as ArcMap. A customized ArcMap application is created that provides a grid based approach to select an AoI on the raster image and extract data for it.

Advisor's Approval: Dr. Johnson P Thomas______________________________________