

Chapter 2

Algorithm for Digitization of Tea Garden maps

2.1	Introduction	36
2.2	The Scheme and implementation	40
2.3	Comparison with existing digitization techniques	53
2.4	Discussion	54

Algorithm for Digitization of Tea Garden maps

2.1 Introduction

Digitizing, or digitization, is the process of turning an analog signal into a digital representation of that signal. The term is often used for the scanning of analog sources, such as printed photos and taped video into computers for editing, but it also can refer to audio (where sampling rate is often measured in kilohertz) and textures map transformation. The term digitizing is also used to describe the process of making a digital representation of features within a raster image using a geographic information system.

The raster spatial data model is one of a family of spatial data models described as tessellations. In the raster world, individual grid cells are used as the building blocks for creating images of points, line, area, network and surface entities. The size of the grid cell is very important as it influences how an entity appears.

The vector spatial data model uses two dimensional Cartesian (x,y) co-ordinates to store the shape of a spatial entity. In the vector world the point is the basic building block from which all spatial entities are constructed. The simplest spatial entity, the point, is represented by a single (x,y) co-ordinate pair. Line and area entities are constructed by connecting a series of points into chains and polygons. The more complex the shape of a line or area feature the greater the number of points required representing it. Selecting the appropriate number of points to construct an entity is one of the major dilemmas when using the vector approach. If too few points are chosen the character, the shape and spatial properties of the entity (for example, area, length, and perimeter) will be compromised. If too many points are used, unnecessary duplicate information will be stored and this will be costly in terms of data capture and computer storage.

In the vector data model the representation of networks and surfaces is an extension of the approach used for storing line and area features. However, the method is

more complex, and closely linked to the way the data are structured for computer encoding.

Section 2.1.1 deals with the raster data structure whereas information about vector data structure is given in section 2.1.2. Glimpse of digitization and its utility is given in section 2.1.3.

2.1.1 Raster Data Structures

In the raster world a range of different methods is used to encode a spatial entity for storage and representation in the computer. The figure 2.1 shows the most straightforward method of coding raster data. The cells in each line of the image are

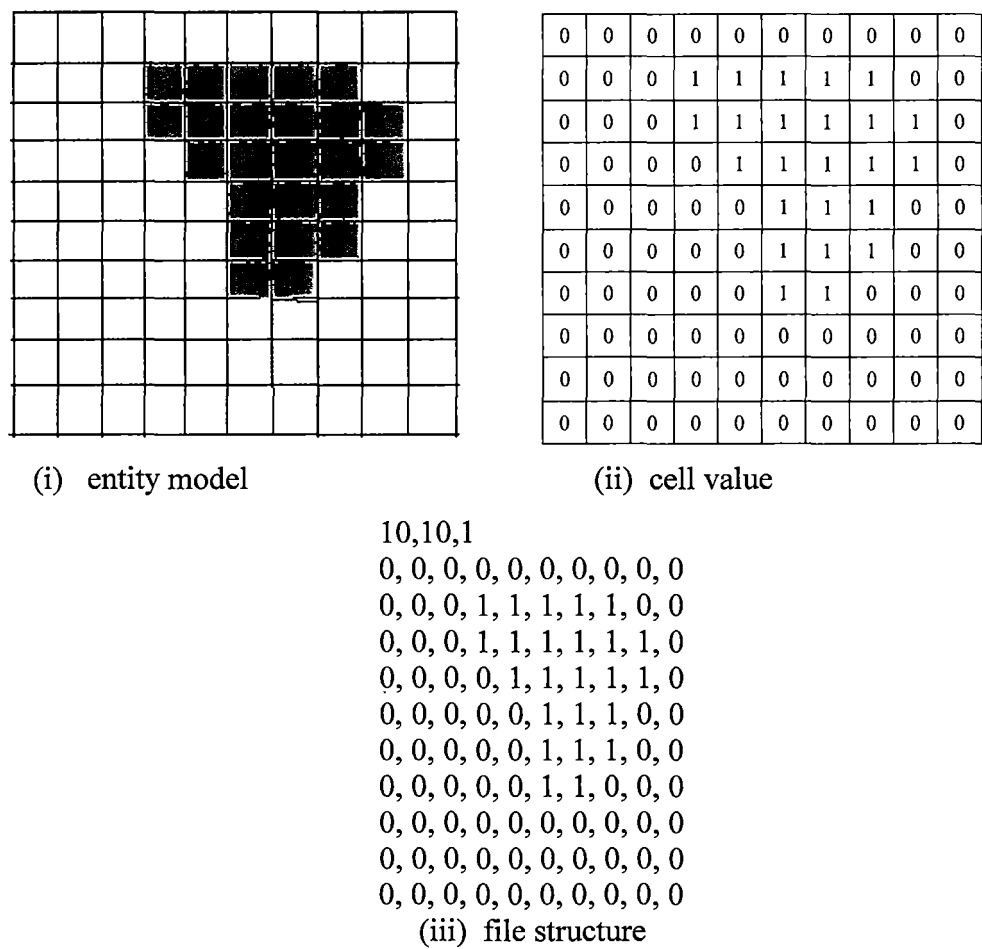


Figure 2.1: A simple raster data structure: (i) entity model (ii) cell value (iii) file structure

mirrored by an equivalent row of numbers in the file structure.

The first line of the file tells the computer that the image consists of 10 rows and 10 columns and the maximum cell value is 1. In the given example a value of 0 has been used to record cells where the entity is not present, and a value of 1 for cells where the entity is present.

One of the major disadvantages with raster data sets is their size, because a value must be recorded and stored for each cell in an image. Thus, a complex image made up of a mosaic of different features (such as a soil map with 20 distinct classes) requires the same amount of storage space as a similar raster map showing the location of a single forest.

2.1.2 Vector Data Structure

The simplest vector data structure that can be used to reproduce a geographical image in the computer is a file containing (x,y) co-ordinate pairs that represent the location of individual point features (or the points used to construct lines or areas). The vector data structure is given in figure 2.2.

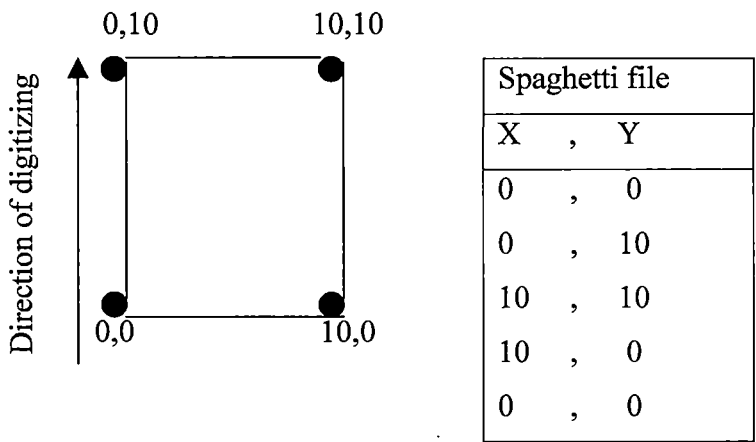


Figure 2.2: Simple vector data structure

The limitation of simple vector data structures start to emerge when more complex spatial entities are considered. The simple data structure could be used to graphically reproduce the network without any duplication of data.

2.1.3 Digitization

All data in analog form need to be converted to digital form before they can be input into GIS. Some common methods of digitization are keyboard entry, manual digitizing, automatic digitizing and scanning. Keyboard entry may be appropriate for tabular data or for small number of co-ordinate pairs read from paper map or pocket GPS. The most common method of encoding spatial features from paper maps is manual digitizing. It is an appropriate method when a selection of features is required from a paper map. Manual digitizing requires a spatial hardware called table digitizer that is linked to a computer workstation.

Most manual digitizers may be used in one of two modes: point mode or stream mode. In point mode the user begins digitizing each line segment with a start node, records each change in direction of the line with a digitized point and finishes the segment with an end node. So, a straight line can be digitized with just two nodes. For more complex lines, a greater number of points are required between the start and end nodes.

In stream mode the digitizer is set up to record points according to a stated time interval or on a distance basis. Once the user has recorded the start of a line the digitizer might be set to record a point automatically every 0.5 seconds and the user must move the cursor along the line to record its shape. At end node is required to stop the digitizer recording further points. The speed at which the cursor is moved along the line determines the number of points recorded. Thus, for more complex line where the cursor needs to be moved more slowly and more care, a greater number of points will be recorded.

The manual digitizing of paper maps is one of the main sources of positional errors in GIS. The accuracy of data generated by this method of encoding is dependent on

many factors including the scale and resolution of the source map and the quality of the equipment and software being used.

Manual digitizing is a time consuming and tedious job. If a large number of complex maps such as land use maps, contour maps containing thousands of lines or polygons need to be digitized then this manual process will be more expensive. Two common automatic digitizing methods are scanning and automatic line following. Scanning is the most commonly used method of automatic digitizing. Scanning is an appropriate method of data encoding when raster data are required, since this is the automatic output format for most scanning software.

A scanner is a piece of hardware for converting an analog source document into digital raster format. All scanners work by sampling the source document using transmitted or reflected light. The accuracy of scanned output data depends on the quality of the scanner, the quality of the image- processing software used to process the scanned data and the quality (and complexity) of the source document.

Another type of automatic digitizer is the automatic line follower. This encoding method might be appropriate where digital versions of clear, distinctive lines on a map are required. Whereas scanners are raster devices, the automatic line follower is a vector device and produces output as (x,y) co-ordinate strings. The data produced by this method are suitable for vector GIS. Automatic line followers are not as common as scanners, largely due to their complexity. In addition, difficulties may be faced when digitizing features such as dashed line or contour lines. Considerable editing and checking of data may be necessary [8,9,10].

2.2 The Scheme and implementation

In this section an algorithm for digitization process has been proposed [4]. The technique is based on dragging of mouse on a line/curve/polygon where the position of the mouse pointer transmit an (x,y) co-ordinate. This coordinates are emulated as attributes and the information is placed into table. When the mouse pointer is moved on the scanned map from one point to next neighboring point coordinate for this neighboring

point is taken and used as attribute for neighboring pixel (point) and stored as subsequent records into the database table.

A map may contain different characters like road, river, border etc. To identify these features a unique color attribute for each feature is assigned and as the same has also been kept into a separate table with a unique key management.

There is provision for segmentation of the scanned map, as it physically exists in a tea garden maps to obtain the facility of retrieving a particular segment or the projection of multiple segments on the same screen.

When someone needs to use those coordinates, it retrieves from the database and converts into image objects. The collection of the object looks as a complete image.

The schematic diagram of the technique is given in the figure 2.3 and the process/algorithm proposed for digitization is given in section 2.2.1.

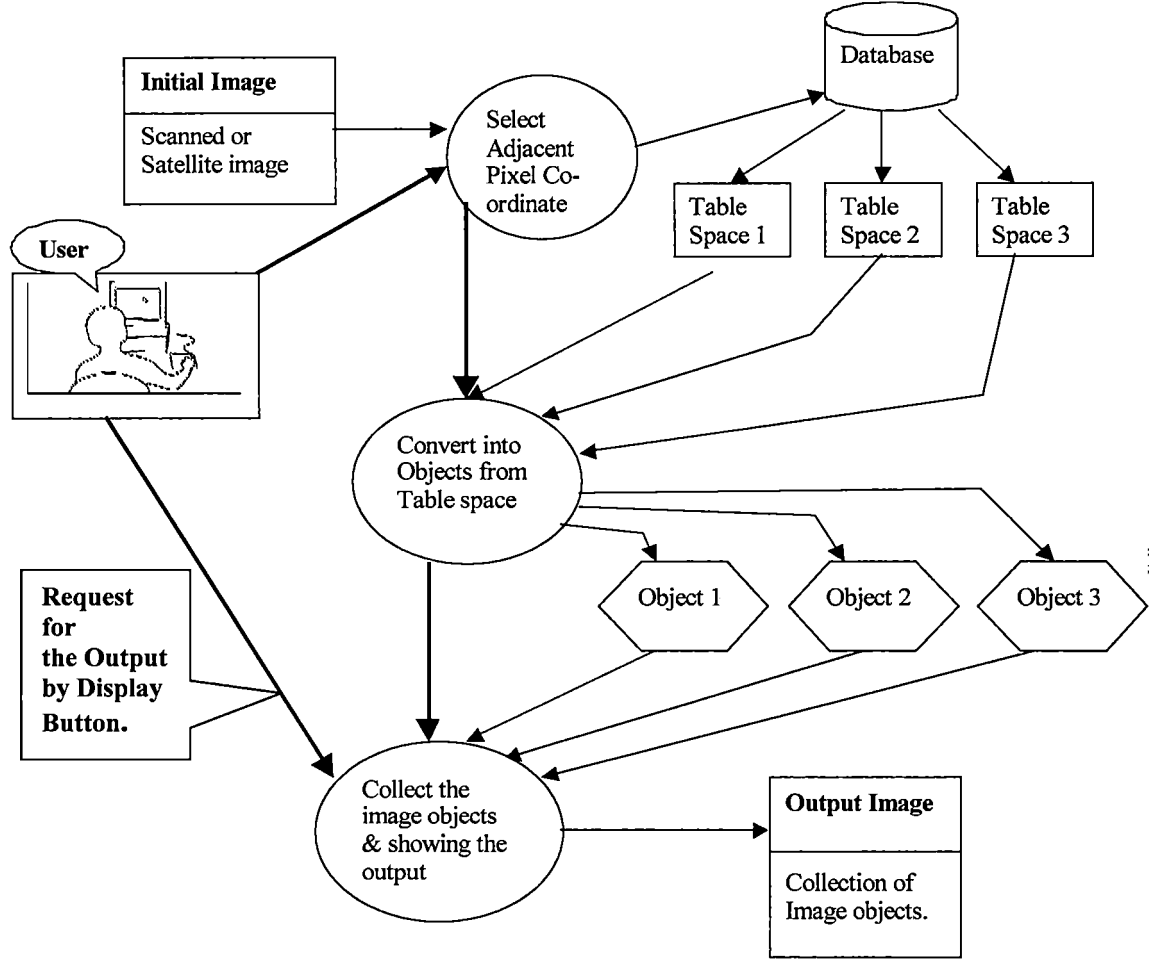


Figure 2.3: Schematic diagram of the digitization process

2.2.1 Digitization Process

The initial image captured by a scanner is expressed as a continuous function $f(x,y)$ of two co-ordinates in the plane. These co-ordinates, in the form of a table space, are further transformed into its corresponding object, which, as a result, gives the output of the image. The operation is performed in following steps:

- (i) Placement of image in the frame.
- (ii) Vectorization of polygons.
 - (a) Association of image object name with vectorized polygons.
 - (b) Association of color attribute to polygons.
- (iii) Regeneration of image from database.

These are describing in details in the section 2.2.1.1 to 2.2.1.3.

2.2.1.1 Placement of image in the frame

A Scanned image is considered as a label object in Java Swing that is placed into a Panel object of Java swing frame. The algorithm for insertion of the scanned image is given in section 2.2.1.1.1.

2.2.1.1.1 Algorithm

- Step 1: Declare the frame object*
- Step 2: Set the frame size of the frame object*
- Step 3: Declare the label object*
- Step 4: Declare the panel object*
- Step 5: Set panel layout of the panel object*
- Step 6: Set panel background of the panel object*
- Step 7: Add label object into panel object*
- Step 8: Add panel object into frame object*
- Step 9: Declare the button object*
- Step 10: Add button object into frame object*

Figure 2.4 shows the result of attachment of scanned image/map to frame. Figure 2.5 shows the schematic mapping of the coordinates of the image and display device. The extent of the image/map is taken as $[0,0],[0,N],[M,0]$ and $[M,N]$, where M and N are pixel values of number of rows and columns of the scanned image/map (EFGH in the

figure 2.5). ABCD is the initial frame of the screen then the initial point of the image is at a distance (x,y) from the beginning of the screen.

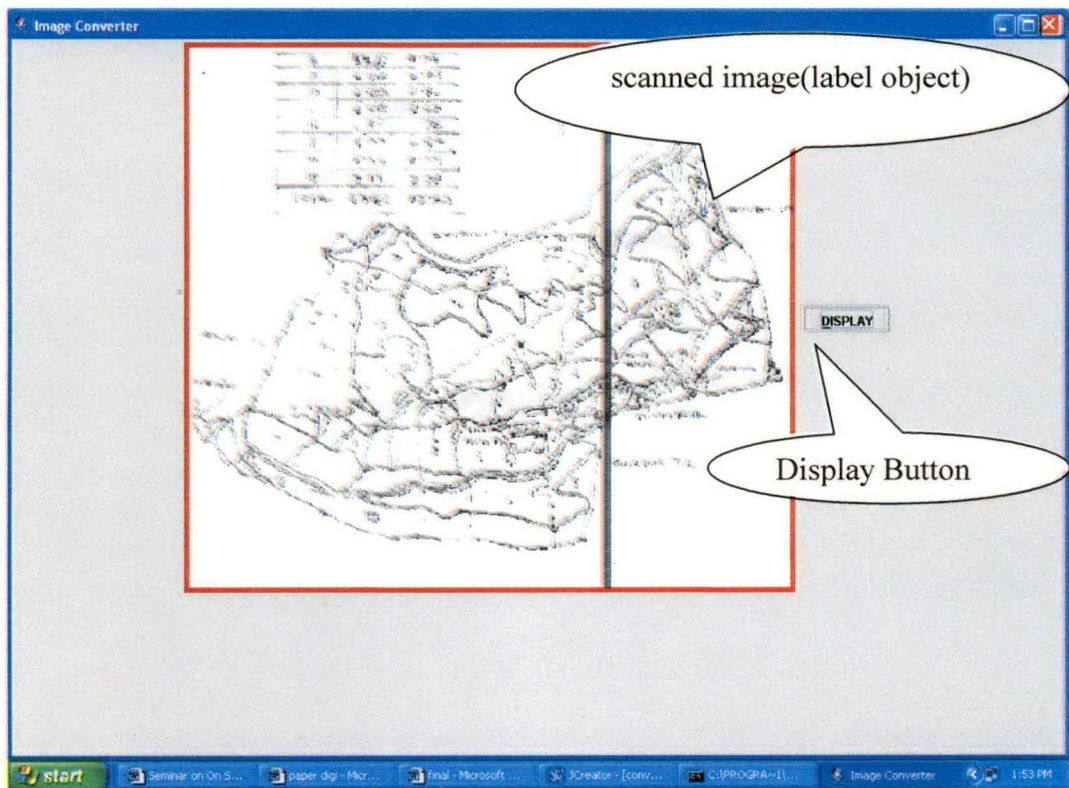


Figure 2.4: Java frame with the scanned image

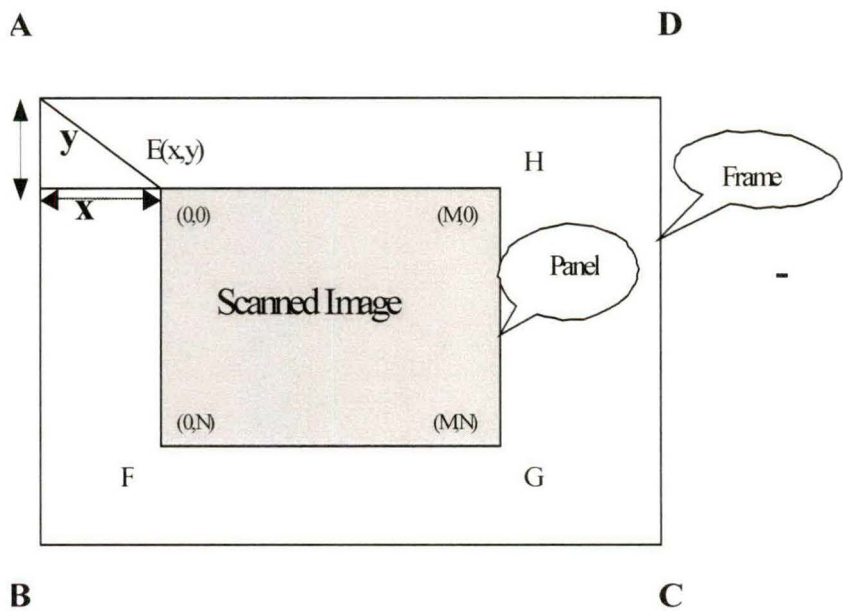


Figure 2.5: Java frame with the scanned image added into the Panel

2.2.1.2 Vectorization of polygon/curve

Select the required adjacent pixels (as given in figure 2.6) co-ordinate value ($f(x,y)$) with the help of the left mouse button. The user needs to drag over the required pixels.

The first step towards converting the original image to the final image begins by acquiring the image from the source and feeding it into the system. Once the image has been fed the digitization process can begin. The user first selects portions from the image that has to be digitized by dragging the left clicked mouse over the coordinates to be considered for digitization (i.e. he selects intervals within the image to form a polygon). Once this is over automatically a window appears that prompts the user to enter what the image represents from amongst various choices. After this if he chooses to proceed, he is asked to choose a unique color (RGB code) from the color information regarding the polygon is then stored in a global database and various table space are created. These tables hold information regarding the selected portion of the original image.

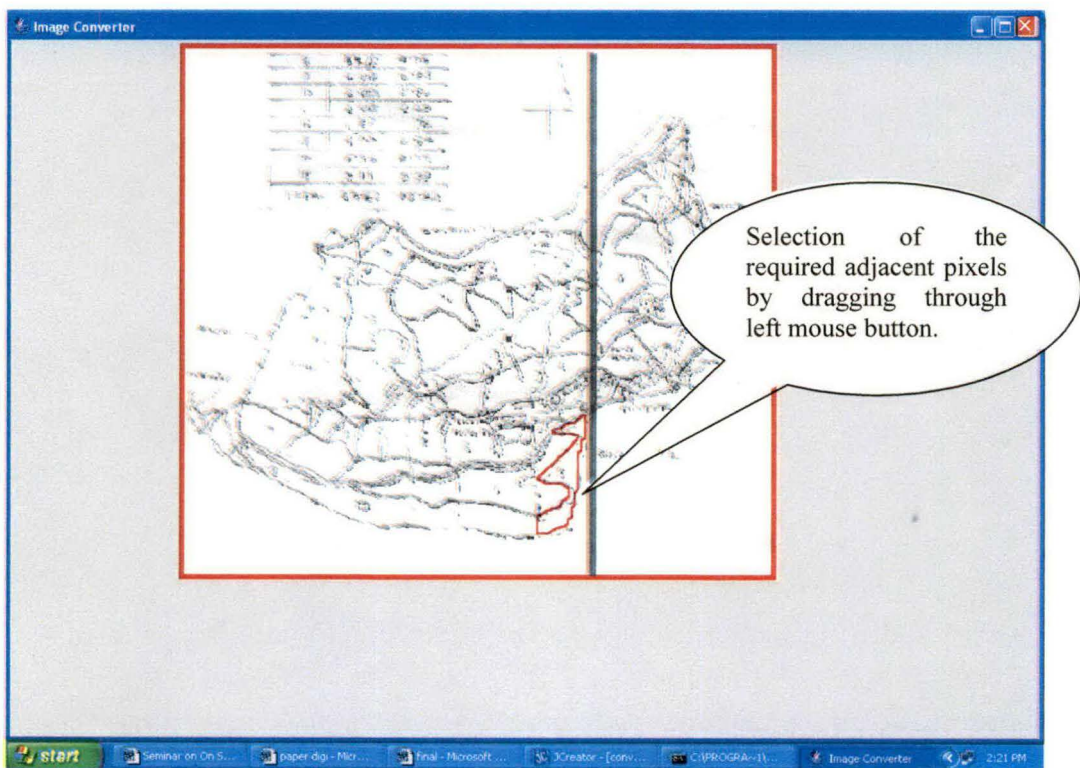


Figure 2.6: Selection of adjacent pixels

Association of the image object attributes with the vectorized polygon and association of color attribute with vectorized polygon is describe in detail in section 2.2.1.2.1 and 2.2.1.2.2.

2.2.1.2.1 Association of image object name with vectorized polygon/curve

At the time of selection of the adjacent pixels (points), the pixel values of X-axis and Y-axis are stored into two different arrays ($X[i]$, $Y[i]$). After completion of the selection process, the user needs to choose the corresponding image object name (Segment, Road, River, Border etc.) through a dialog box (as given in figure 2.7). A new table is automatically created in Oracle whose name is linked with the image object name (as given in figure 2.8). The two different array elements are comprised to be stored into that new table. The mapping of the adjacent pixels with the elements of database table is shown in figure 2.9.

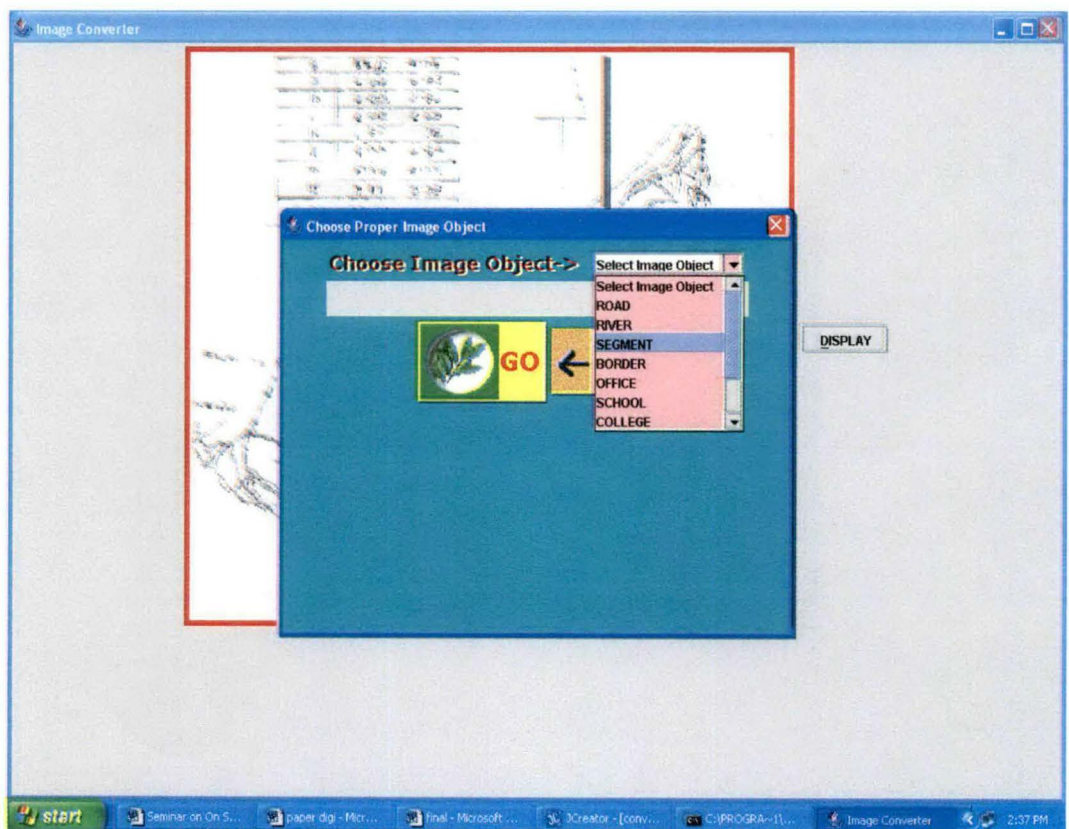


Figure 2.7: Dialog Box to choose the Image Object name

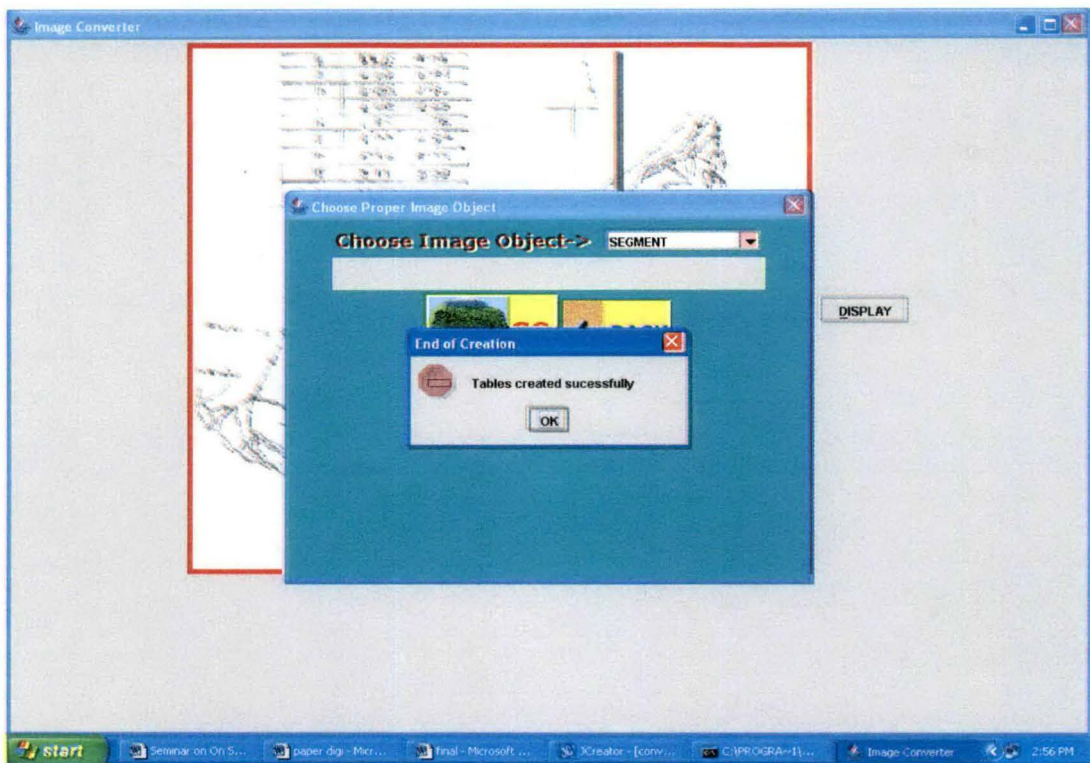


Figure 2.8: Confirmation message of table creation

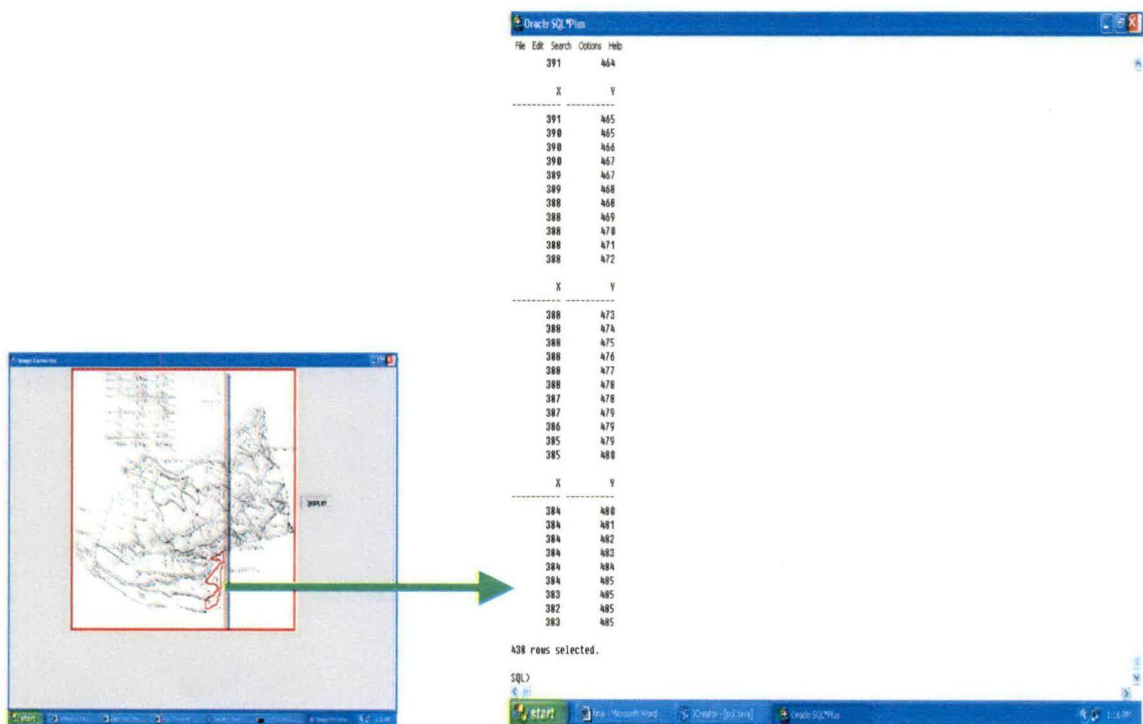


Figure 2.9: Adjacent coordinates are mapped into database

2.2.1.2.2 Association of color attribute with the vectorized polygon/curve

After completion of the storing process, a user interface has been created for association of color attribute to polygon. The user needs to choose the corresponding color of the image object through a dialog box (as given in figure 2.10). Further the name of the new table created along with the RGB value of the color corresponding to that image object is stored into another table defined earlier by the administrator whose name is same as the image object name (as given in figure 2.11).

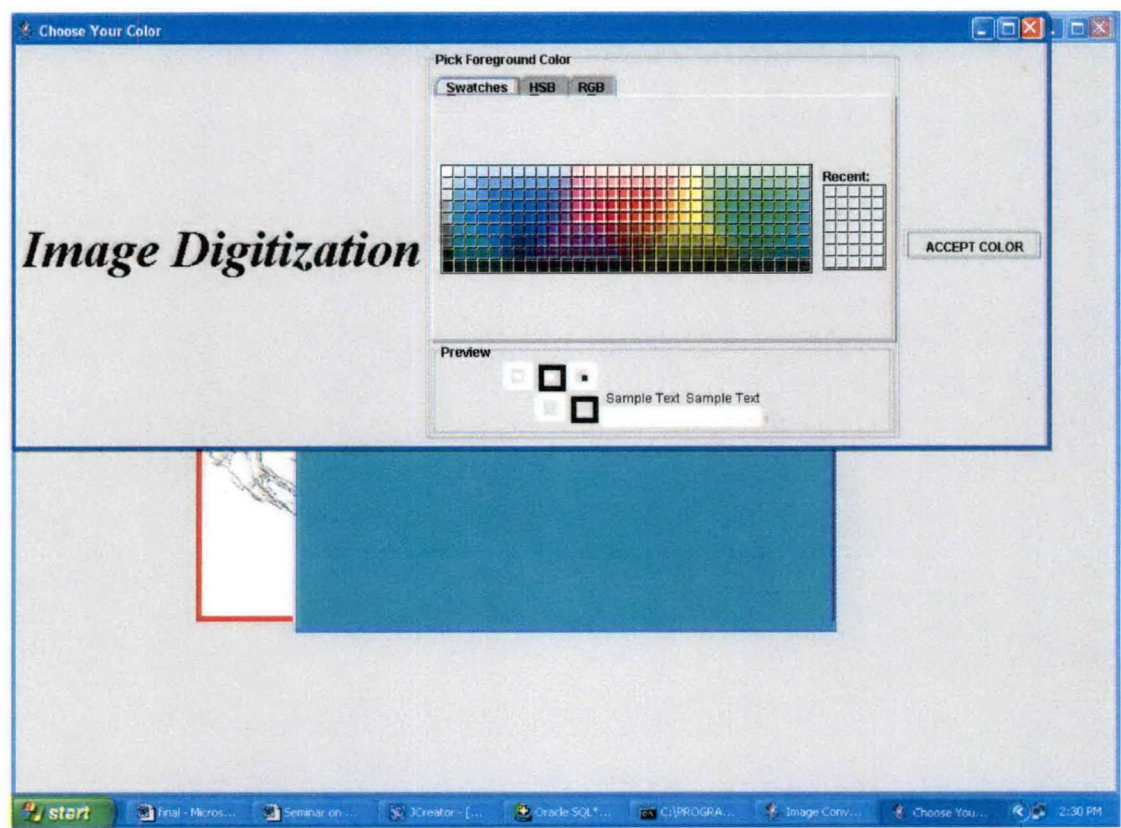


Figure 2.10: Dialog Box to choose the color of the image objects

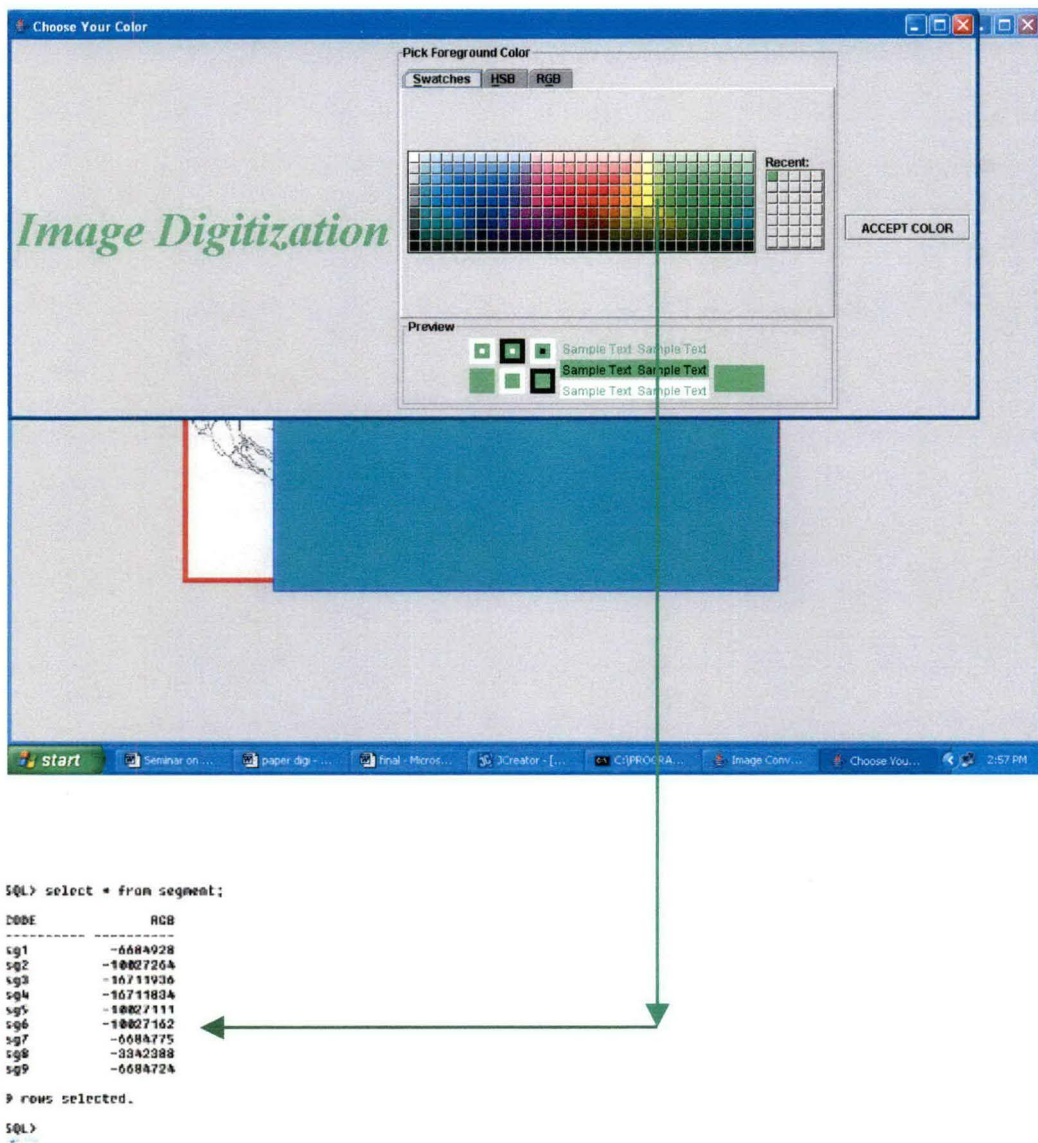


Figure 2.11: Color code mapping into the database.

The algorithm of the overall step is given in section 2.2.1.2.3.

2.2.1.2.3 Algorithm

Step 1: Get the pixel value of X-axis & Y-axis.

Step 2: Put the pixel value into two different arrays.

Step 3: Increment the counter of the array.

Step 4: Repeat Steps 1, 2 and 3 until mouse button is released.

Step 5: Choose the corresponding Image object through a dialog box.

Step 6: The two different array elements are comprised to be stored into a newly created table, which is then linked with the image object name.

Step 7: Choose the corresponding color of the Image object through a dialog box.

Step 8: The name of the table created above and the RGB value of the chosen color object are stored into another table with the same image object name.

2.2.1.3 Regenerating of image from database tables

Once the image is created in the digital form (i.e. information regarding selected image) and request is made to display the output. The process retrieves all the necessary information from the global database tables and forms the digitized image objects (Segments, Roads, Rivers, and Borders etc).

The main table of this package is the table named GIS that has as its fields the names of all the sub tables that are the image object names (such as Segment, Road, Border, River, etc). Each of these sub tables are again divided into two portions (has two fields) a unique table name and RGB value where the unique table is the name of a sub-table of the same name that contains the X and Y co-ordinates of the object described and the RGB portion contains the RGB code for the given data object (as given in figure 2.12).

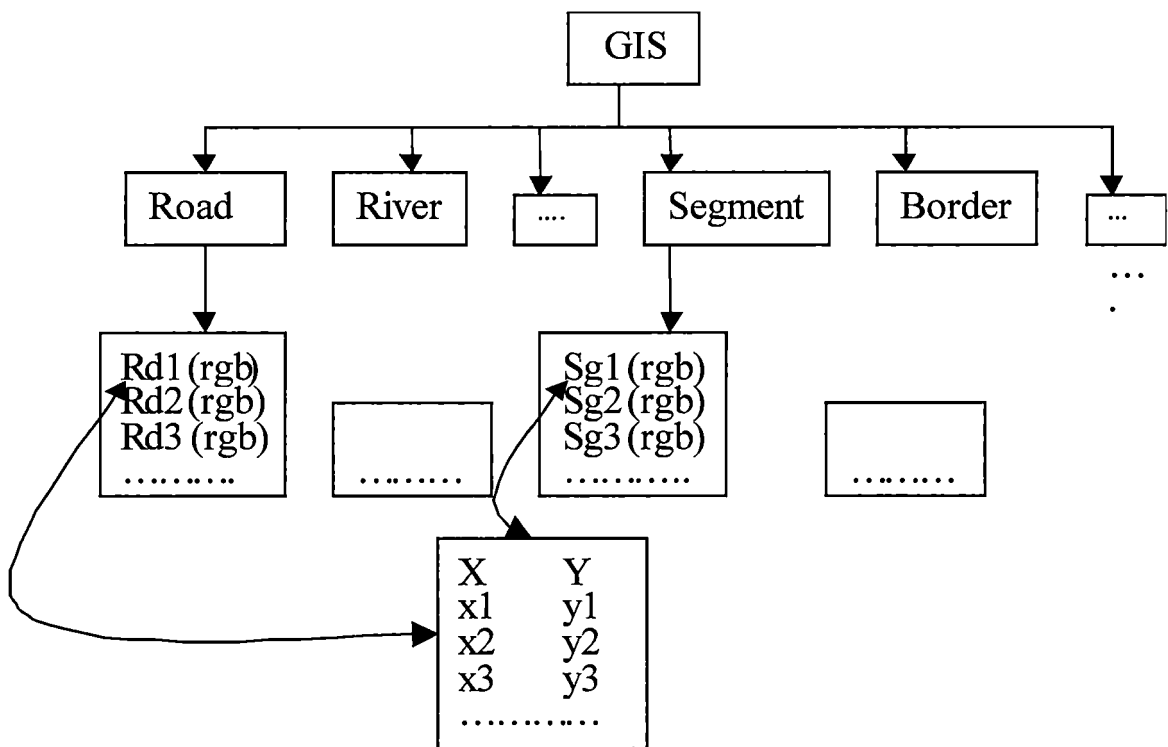


Figure 2.12: Relationship between the tables

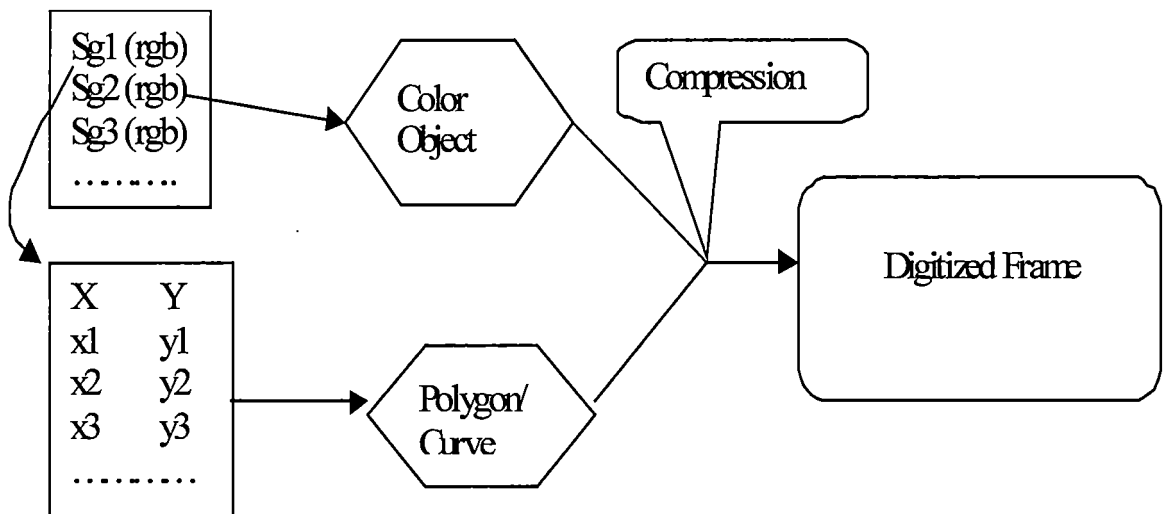


Figure 2.13: Figure object (Polygon/Curve) and color object are compressed into digitized frame

Two different arrays are used to get the values from the tables, which are containing the X and Y coordinates. A Figure object (polygon, curve) is regenerated with the help of those array elements and a color object is also generated from the RGB value. Each figure object is combined with the corresponding color object and it compressed into the final digitized frame (as given in figure 2.13). After regenerating all the data from tables the digitized frame would be like the Figure 2.14.

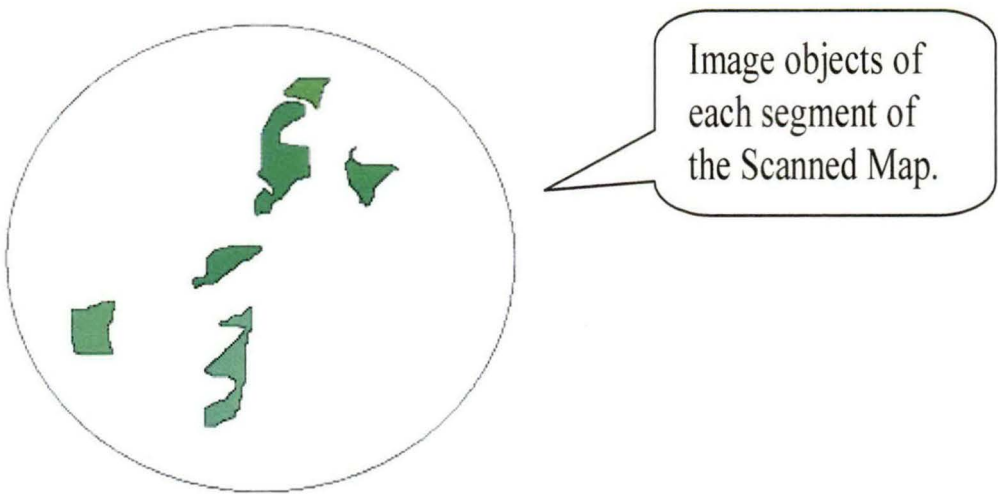


Figure 2.14: Digitized frame with the digitized image objects

The image pixel values of the initial frame are mapped with the whole digitized frames pixel values (as given in figure 2.15). Therefore a point (m, n) of the initial frame will be consider $(m-x, n-y)$ of the digitized frame where (x,y) denotes the corresponding distance of the panel(i.e. the y^{th} row and x^{th} column) from the position $(0,0)$ on the initial frame.

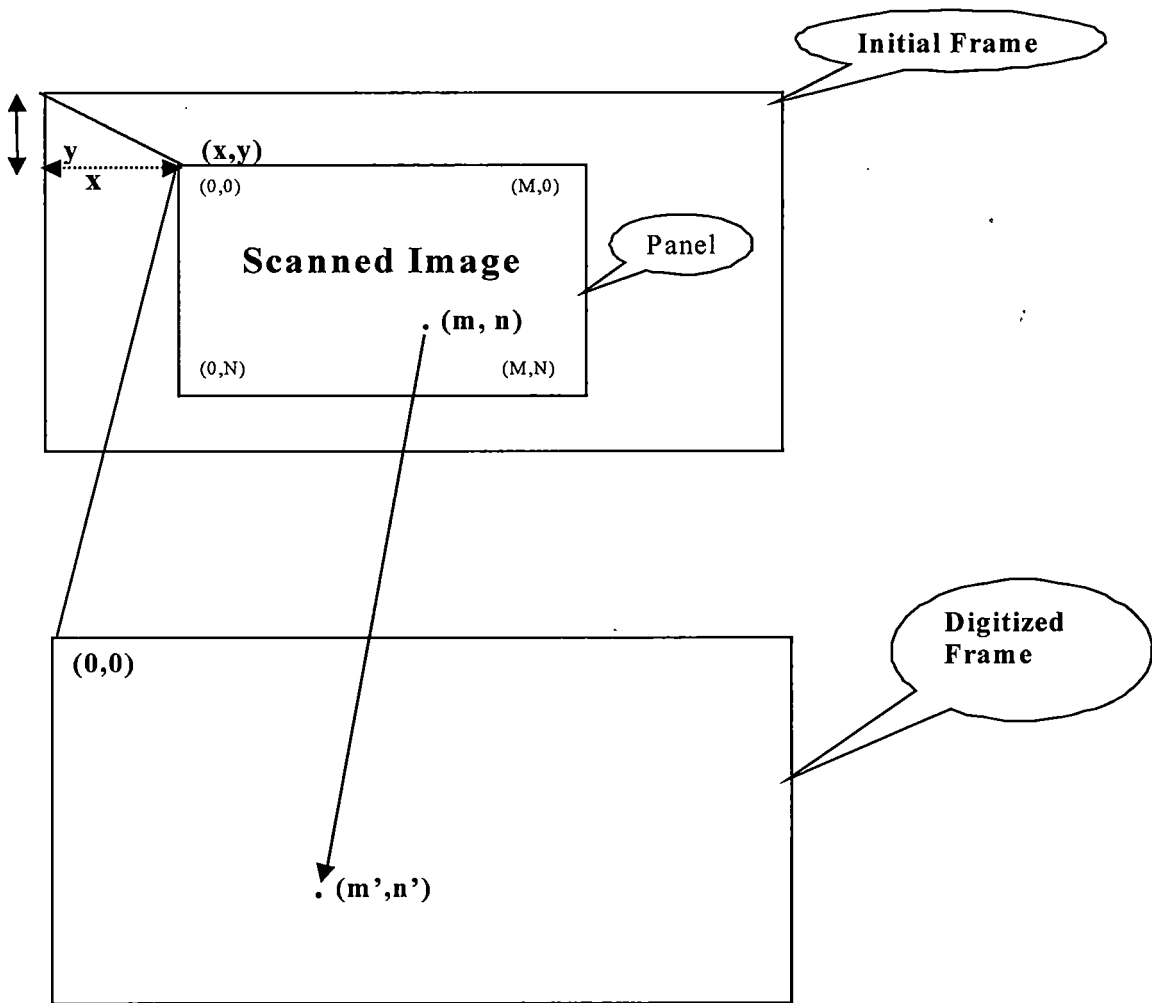


Figure 2.15: Point mapping between the initial frame and digitized frame

The algorithm for Regenerating of image from database is given in section 2.2.1.3.1.

2.2.1.3.1 Algorithm

- Step 1: Open the main table that contains the table names same as image object name.
- Step 2: Open each table, which are specified in the main table.
- Step 3: Generate the color objects from the RGB values specified in the table.
- Step 4: Open those tables which containing x, y coordinates values are specified in the above-mentioned table.
- Step 5: Transfers those x, y coordinates values in two different arrays.
- Step 6: Generate the figure objects (Polygon, Curve) from the arrays.
- Step 7: Combined the figure object with the corresponding color object.
- Step 8: Compressed the colored figure object into the digitized frame.

2.3 Comparison with existing digitization techniques

Table 2.1 shows some of the characteristics of proposed system, which may lead to better digitization process for the proposed tea garden management system.

Table 2.1: Comparison of proposed digitization technique with existing systems

Proposed technique	Existing digitization techniques
1. It produces a collection of objects. Objects may be a separate section of the tea garden and we can use these objects independently whenever needed in the program and it may reduce the cost and increase the program efficiency.	1. Final output is a unique digitized image. Objects within the image can not be isolated.
2. A digitized segment may be called independently in any management system under the complete MIS Scheme without loading complete image of the tea garden, which requires less memory and storage during execution. Hence the proposed technique is time and space efficient in terms of CPU scheduling.	2. Full digitized image is to be loaded into the memory while processing.
3. The digitization technique is independent to other part of the proposed thesis i.e. digitization tool (as proposed) may be used independently.	3. All digitization techniques are integral part of the package. The digitized image / data cannot be exported to other system hence they are not kept as independent tools.
4. The software for digitization will be available as open source and any tea garden may use/customize this as open source. Implementation cost is very low.	4. The cost of the software where digitization is integrated, as a tool is very high and cannot be used as open source for implementation in any tea garden.
5. The software is free and available as open source.	5. Software is copy protected.

2.4 Discussion

Using this technique one can digitize a particular section (portion) of the map uniquely and identify it. Data for decision-making regarding this selected portion of the map may also be entered, retrieved or searched just like in any other digitization technique.

Unlike most other digitization technique the final output of this technique does not produce a unique digitized image for the selected portions. It rather produces a collection of objects, so even if the user wants he/she will not be able to get a digitized image of the map using this technique.

To use this technique a user must be well versed with the use of computers and an expert in using computer aided drawing tools, because he has to accurately move the mouse over the border of the selected portion in order to select the pixels accurately.

Presently this technique uses Oracle 9i for storing the data. This makes the package costly. However this can be reduced by storing the data as (.dat) or (.txt) files.