TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

CENTRAL CAMPUS, PULCHOWK

THESIS NO.: 069MSCS667

Developing a Faster LOS Algorithm for Faster Viewshed Generation

By

Subin Shrestha

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRONICS AND
COMPUTER ENGINEERING IN PARTIAL FULFILMENT OF THE
REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE IN
COMPUTER SYSTEMS AND KNOWLEDGE ENGINEERING

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING
LALITPUR, NEPAL

NOVEMBER, 2014

Developing a Faster LOS Algorithm for Faster Viewshed Generation

By

Subin Shrestha

(069/MSCS/667)

Thesis Supervisor

Sanjeeb Prasad Panday (Ph. D.)

(M. Sc. Program Coordinator)

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Systems and Knowledge Engineering

Department of Electronics and Computer Engineering

Institute of Engineering, Central Campus, Pulchowk

Tribhuvan University

Lalitpur, Nepal

November, 2014

COPYRIGHT

The author has agreed that the Library, Department of Electronics and Computer Engineering, Institute of Engineering, Central Campus, Pulchowk may make this thesis freely available for inspection. Moreover, the author has agreed that the permission for extensive copying of this thesis work for scholarly purpose may be granted by the professor(s) who supervised this work recorded herein or, in his/her absence, by the Head of Department wherein this thesis was done. It is understood that the recognition will be given to the author of this thesis and to the Department of Electronics and Computer Engineering, Institute of Engineering, Central Campus, Pulchowk in any use of the material of this thesis. Copying or publication or other use of this thesis for financial gain without approval of the Department of Electronics and Computer Engineering, Institute of Engineering, Central Campus, Pulchowk and author's written permission is prohibited.

Request for permission to copy or to make any use of the material in this thesis in whole or part should be addressed to:

Head

Department of Electronics and Computer Engineering

Institute of Engineering

Central Campus, Pulchowk

Lalitpur, Nepal

TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

CENTRAL CAMPUS, PULCHOWK

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

The undersigned certify that they have read, and recommended to the Institute of Engineering for acceptance, a thesis report entitled “Developing a Faster LOS Algorithm for Faster Viewshed Generation” submitted by Subin Shrestha in partial fulfillment of the requirements for the degree of Master of Science in Computer Systems and Knowledge Engineering.

(SIGNED)____

Supervisor, Dr. Sanjeeb Prasad Panday
M. Sc. Program Coordinator
Department of Electronics and Computer Engineering

(SIGNED)____

External Examiner, Krishna Prasad Bhandari
Senior Engineer
Nepal Telecom

(SIGNED)____

Committee Chairperson, Dr. Shashidhar Ram Joshi
Professor
Department of Electronics and Computer Engineering

2ND NOVEMBER, 2014_________

Date

DEPARTMENTAL ACCEPTANCE

The thesis entitled “Developing a Faster LOS Algorithm for Faster Viewshed Generation”, submitted by Subin Shrestha in partial fulfillment of the requirements for the award of the degree of “Master of Science in Computer Systems and Knowledge Engineering” has been accepted as a bona fide record of work independently carried out by him in the department.

(SIGNED)__________________________

Dr. Dibakar Raj Pant

Assistant Professor and Head of Department

Department of Electronics and Computer Engineering

Institute of Engineering, Central Campus, Pulchowk

Tribhuvan University

Lalitpur, Nepal

ABSTRACT

The R3 viewshed algorithm is widely used in the GIS industry as it is an accurate method for viewshed generation. However, the computational speed of the R3 algorithm is slow. Other algorithms, although faster, only provide an approximation of the viewshed. So, an accurate viewshed algorithm which provides a faster result is anticipated.

A faster viewshed algorithm based on R3 method has been formulated and developed in this study. The improvement is based on the process that separates shadow regions, which reduces number of intermediate points for visibility analysis.

The faster viewshed algorithm has been tested and verified to confirm the improvements brought over the general viewshed algorithm. Experimental tests were carried over the geographic database of 48000 sq. km which included terrains varying from 58 m to 8000 m. The results obtained from all these tests have suggested a clear and significant gain in the computation time.

Keywords:
Viewshed, Viewshed algorithm, Faster LOS, Faster viewshed, Faster R3 algorithm

ACKNOWLEDGEMENT

It is my pleasure to take this opportunity provided by the Department of Electronics and Computer Engineering to conduct this study.

I would like to express my sincere gratitude to my thesis supervisor, Dr. Sanjeeb Prasad Panday for his indispensable steer, support and inspiration. Without his assistance and instruction in every step, this dissertation would not have been accomplished.

I am indebted to Prof. Dr. Shashidhar Ram Joshi for his kind encouragement and aspiring input to start this work. I am also grateful to Dr. Aman Shakya for his guidance and for providing me with an insight on the possibilities to extend this study into applications.

I wish to express my warm thanks to my friends for their invaluable support and feedback.

Finally, a special thanks to my family for supporting me and encouraging me with their blessings to accomplish this work.

Subin Shrestha
Kathmandu, Nepal
November 2014
(subin.shrestha@gmail.com)

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

DEM	Digital Elevation Model
GIS	Geographic Information System
IDE	Integrated Development Environment
LOS	Line of Sight
SRTM	Shuttle Radar Topographic Mission
SRTM-3	Shuttle Radar Topographic Mission 3 arc

CHAPTER 1

INTRODUCTION

1.1 Background

Visual perception has had a profound effect on humankind thriving, since the latter has been deriving meaning from its surroundings; in other words, for us, humans ‘to see is to understand and prosper’. Living in the natural environment, though, entails active visual landscape experience which is an inherently dynamic process and by which we apprehend and understand the most of our surroundings.
Information about points of the landscape that are mutually visible or parts of a region which are visible from one or more points of observation is valuable for several reasons and applications. Yet, such information neither is synthesized by itself to a perceivable and meaningful context, nor it involves the active visual experience. [1]

The viewshed of a location may be defined as the distribution of an area of the surrounding region visible from that location. As an important terrain parameter, viewshed is widely used in engineering, environmental management and the military. The generation of outlines of the mountains and hills is also an interesting application of the viewshed.

The main idea of the viewshed is based on Line of Sight (LOS) calculation which analyzes the regional scope for a given view-point. Various algorithms for computation of viewshed have been developed. These algorithms have been developed for the efficient computation of visible landscape from a particular point.

The major algorithms for the visibility analysis are R3 and R2 viewshed algorithm. The R3 is the most accurate and simple and direct algorithm for viewshed analysis but it is also computationally time-consuming. The R3 algorithm analyzes the visibility of each grid point in the regional view-point. The R2 algorithm requires computations proportional to the number of grid elements, resulting in substantial computational savings. In the R2 algorithm, computations from previous inter-visibility analysis are used to assist subsequent computations, resulting in reduced total computational cost. [2]

The viewshed analysis is a spatial analysis. The viewshed analysis using digital elevation model (DEM) has become one of the standard functions of the Geographic Information System (GIS). The elevation models derived from the Shuttle Radar Topographic Mission (SRTM) are popularly used for the spatial analysis.

In this study, a faster LOS algorithm for viewshed generation has been proposed. This betterment in the LOS method would be carried out by separating the shadow region in the LOS profile whose visibility does not need to be checked as the range falling in the shadow region are not visible to the observer.

Using the faster LOS algorithm, a faster viewshed generation algorithm has also been devised and proposed.

1.2 Problem definition

The major algorithms for viewshed generation are based on Line of Sight (LOS) computation method. The widely used viewshed algorithms - R3 and R2 both uses the LOS method in which the visibility of each point in the elevation profile is checked for its visibility.

During this LOS check, the conventional method does not take advantage of shadow regions whose visibilities do not need to be checked individually. This method checks visibility of every point in the shadow region too which has no role in the visibility of the final target. Hence, an accurate but faster viewshed algorithm is anticipated.

1.3 Objectives

The main objectives of this research are:

- To take advantage of shadow region to reduce the time taken in LOS computation
- To make viewshed generation faster by developing faster algorithm for viewshed computation based on R3 method

1.4 Scope of the work

The proposed LOS and viewshed algorithm can reduce the computation time of the viewshed. When the proposed algorithm is implemented, all algorithms which are based on sightline method would become faster and more efficient. The popular and widely used viewshed algorithms - R3 and R2 algorithm would also benefit from the above method. Hence, the generation of the viewshed of an area would be computed more efficiently in lesser time with the proposed algorithm.

1.5 Organization of the report

This report is divided into five chapters. A brief description of each chapter is presented below:

- Chapter 1: This chapter has described a brief about the research study and the objectives. The definition of the problem has been included in this chapter. The outline of this report is also described here.

- Chapter 2: The relevant literatures studied and referred in course of this research work has been included in this chapter.

- Chapter 3: The ways and methods of generating faster viewshed using proposed algorithm has been described in this chapter.

- Chapter 4: This chapter accounts for the steps proceeded in order to carry out of this study and explains how each procedure had been realized.

- Chapter 5: This chapter presents the results obtained from the study and includes analysis and verification of the obtained outputs.

- Chapter 6: The conclusion of this research along with recommendations derived from this study has been presented in this chapter.

CHAPTER 2

LITERATURE REVIEW

2.1 Viewshed

A viewshed is the area on the ground that is visible from a specific location. It may be an area of land, water, or other environmental element that is visible to the human eye from a fixed vantage point. Viewsheds are often spaces that are readily visible from public areas such as from public roadways, public parks or high-rise buildings. Computing the viewshed of an observer's viewpoint is a common requirement for tourism, land management, architectural simulation and many other applications.

Viewshed analysis using Digital Elevation Models (DEMs) has become a standard function of Geographical Information System (GIS). The analysis uses the elevation value of each cell of the DEM to determine visibility to or from a particular cell. The location of this particular cell varies depending on the needs of the analysis. For example, viewshed analysis is commonly used to locate communication towers or determining the view from a road.

Viewsheds can be calculated using an individual point such as a tower or multiple points such as a line representing a road. For example, when locating a landfill, the analysis can determine from where the landfill is visible to keep it hidden from view. Viewshed can also be used to optimize the number of guards to safeguard a region. It can also be used in planning a cell tower for the prediction of the signal coverage.

The generation of the outlines of the mountains and hills as seen in the vertical plane is an interesting application of the viewshed analysis. This 360° generation of the profile lines of the mountains and hills can be used for identification of the landscape as seen from the observer located at particular location.

2.2 Viewshed algorithms

Various studies have been carried out to develop a faster algorithm to efficiently compute the viewshed as it comprises of manipulation and calculation of huge database. Local viewshed computation using grid based Digital Elevation Models (DEMs) has been worked out using various algorithms.

The terrain is commonly represented by a regular grid of points called Digital Elevation Model (DEM), where the longitude and altitude of each point are known with exactitude. The visible area from a specific observer situated at a high 'h' from the ground is called viewshed. To calculate this viewshed, all the points of the DEM are possible obstacles. [3]

The line-of-sight (LOS) method is the most basic and exact method to compute the viewshed. LOS or sightline is defined as the line from the view point to the target. If the sightline is blocked by any part of the terrain surface lying between the viewpoint and the target, then the target is invisible to the viewpoint. The elevation profile along the direction joining the viewpoint and each grid position is determined by appropriate interpolation of the grid, and inter-visibility analysis is performed along this profile.

2.2.1 R3 algorithm

Viewsheds on grids are usually modeled in a discrete way: each point in the grid is marked as visible or invisible, and the viewshed of v is defined as the set of all grid points that are visible from v. The standard method for computing viewsheds on grid terrains is the algorithm R3 by Franklin and Ray [4]. R3 determines the visibility of each point in the grid as follows: it computes the intersections between the horizontal projection of line-of-sight and the grid lines (horizontal and vertical), and computes the elevation of the terrain at these intersection points by linear interpolation. This is considered to be the standard model and R3 is considered to produce the "exact" viewshed [2]. This algorithm requires computations proportional to N^3 for an N by N grid, because they require the direct examination of upto N points to determine occlusion between the viewpoint and each of the N^2 grid positions. As described by Franklin and Ray, R3 runs in O (n^3) time, which is too slow in practice, especially for multiple viewshed computations.

2.2.2 R2 algorithm

The algorithm named R2, proposed by Franklin and Ray [4], is an optimization of R3 that runs in O (n^2) time. The idea of R2 is to examine the lines-of-sight only to the grid points on the boundary of the grid; a grid point that is not on the boundary is considered to be visible if the nearest point of intersection between a grid line and one of the examined lines-of-sight is determined to be visible. Overall R2 is fast and, according to its authors, produces a good approximation of R3 that outweighs its loss in accuracy [4].

CHAPTER 3

PROPOSED FASTER VIEWSHED ALGORITHM

3.1 Elevation profile generation

The elevation profile from the observer to the target is generated with the help of Digital Elevation Model (DEM). The latitude and longitude of the observer and the target is supplied as inputs to this process. From the input co-ordinates, the slope and distance from the observer to the target is determined. After that, from the observer point, the interval is increased in pre-set resolution distance towards an angle obtained from the slope.

The elevation at each interval is read from the DEM database upto the target point. The array with interval position as its index and respective elevation as its value would generate the elevation profile.

3.2 LOS computation

The line of sight (LOS) at each interval in the elevation profile is calculated. The first interval point is always visible. The slope of the line from the observe point to n^{th} intermediate point shall be referred to as the slope of the n^{th} interval. A line is drawn from the observer point to the 2^{nd} intermediate point. If the slope of 2^{nd} point is greater than the slope of the 1^{st} point, then the 2^{nd} point is marked as visible. The slope of the last visible intermediate point is stored as the highest slope. If the slope of 2^{nd} point isn't higher than the slope of 1^{st} point, then the 2^{nd} point is marked as invisible.

The slope of the n^{th} interval is compared with the slope of last visible intermediate point (highest slope). If the slope of the n^{th} interval is greater than the slope of last visible point, then the n^{th} point is marked as visible and the maximum slope is updated as the slope of the n^{th} point. If the slope of the n^{th} interval is less than or equal to the highest slope, then the n^{th} interval is marked as invisible. The above process is repeated for all consecutive intervals upto the target interval.

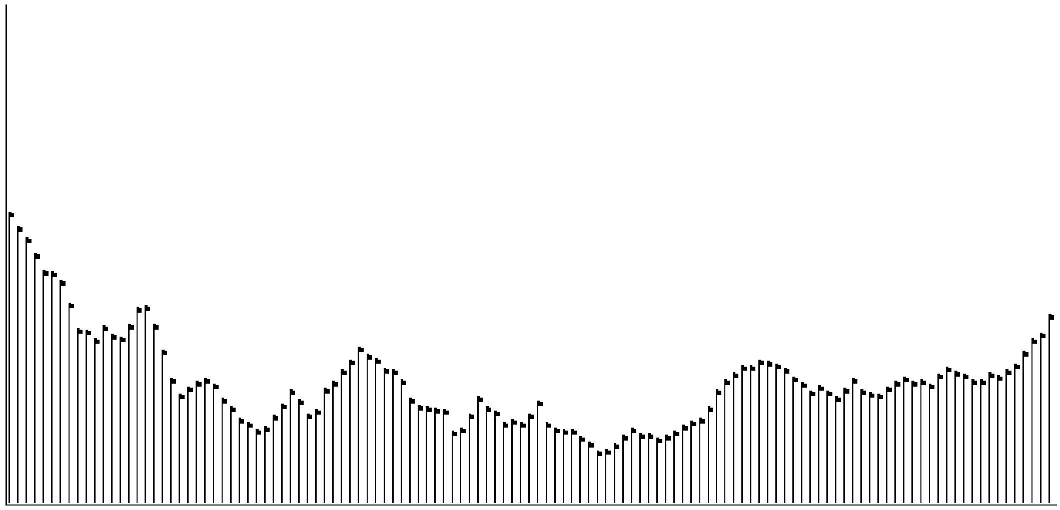

Figure 3.1: Elevation profile generation

In figure 3.1, the left-most and right-most interval represents the observer and the target position respectively.

The algorithm for the above process, which is a general LOS computation method, is described below:

```
1. Start
2. Generate elevation profile from observer to target
3. Mark 1st intermediate point as visible
4. Determine slope of 1st point
5. Set highest_slope = Slope of 1st point
6. Set n = 2
7. Determine slope of nth point
8. If slope of nth point > highest_slope
   Set,
           highest_slope = Slope of nth point
           profile[n] = Visible
   Else Set,
           profile[n] = Invisible
```

9. Check next n^{th} point, n = n + 1
10. Go to Step 7
11. Repeat step 5 to 8 for all values of n-intervals from observer to the target
12. End

3.3 Faster computation of LOS

The computation of the LOS could be carried out faster if we take advantage of the shadow region in the elevation profile. To take this advantage, we find out all the maxima point in the elevation profile.

Each intermediate point is tested for maxima. The maximas are determined by comparing the elevation value of the intermediate interval with the elevation value of the preceding and following interval. If the elevation of the intermediate interval is greater than or equal to both the elevation of the preceding and following interval, then that intermediate interval is marked as maxima.

The position of the latest visible point is set equal to the position of 1st interval at the start. For each intermediate point, a test is conducted to determine whether the point is a maxima. When the n^{th} intermediate point is detected as maxima, the visibility test is done from the latest visible point up until a invisible point is determined from slope comparison. The invisible point thus obtained is labeled as the starting point of the shadow (invisible) region.

Then, the slope of the detected maxima is compared to determine its visibility from the observer. If this maxima is visible, then the visibility of points from the maxima is checked in decreasing interval order up until an invisible point is determined by calculation of the slope. The position of the obtained invisible interval would be marked as the end point of the shadow region. During computation, each interval points that are determined to be visible are labeled as ‘visible’. The points that are not labeled as ‘visible’ are supposed to be ‘invisible’ by default.

The range of intermediate points from start point to end point are noted as ‘invisible’ in a single step as these points lie in the shadow region and are not visible to the observers.

The above process of shadow region determination is carried out for all values of ‘n’ until it reaches the target interval. Other shadow regions are also determined in the same fashion.

Hence, the LOS profile is generated from the observer to the target without having to check the visibility of the points lying in the shadow region.

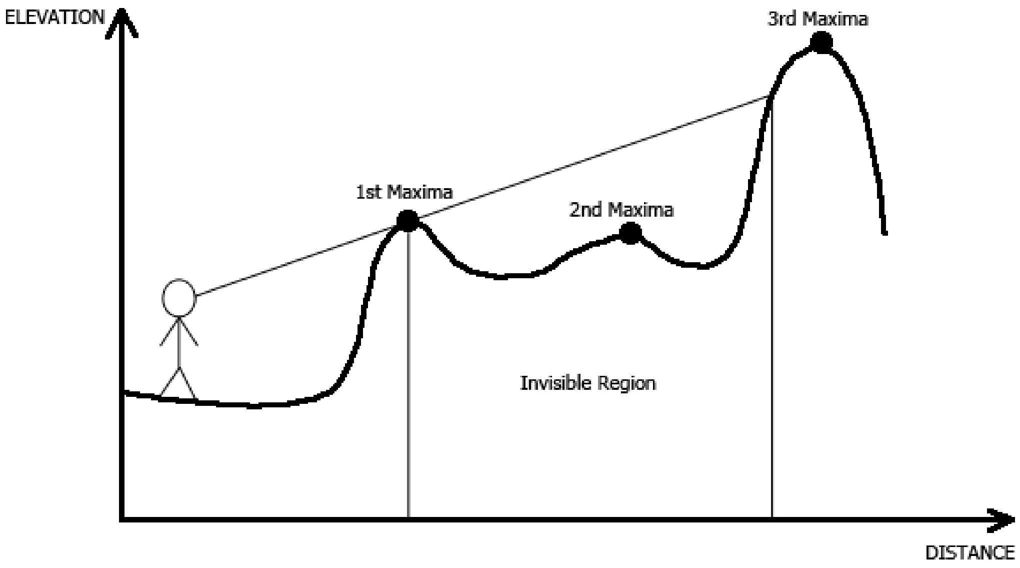

Figure 3.2: Separating invisible region using maxima points

By using this method, the LOS computation becomes faster and more efficient as the total no. of intermediate points whose visibility is to be checked is reduced.

The algorithm for faster LOS computation is described below:

1. Start
2. Generate elevation profile from observer to target

3. Set latest visible point, lvp = 1
4. Set intermediate interval position, n = 1
5. Look out for maxima in the elevation profile
6. If n^{th} intermediate point is a maxima
    a. Test visibility from the ‘lvp’ until an invisible point is determined and mark each visible point detected.
    b. Mark the determined invisible point as the starting point of the invisible shadow region.
    c. If n^{th} intermediate point maxima is visible,
        i. Set lvp = n
        ii. Test the visibility from the n^{th} point in decreasing interval order until an invisible point is determined. Mark each visible point detected and set the position of the latest visible point to the variable ‘lvp’.
        iii. Mark the determined invisible point as the end point of the invisible shadow region.
7. Move to next intermediate point, n = n + 1
8. Repeat steps 5 to 7 for all values of n ranging from 1 to target interval.
9. Finally, check the remaining visibility from ‘lvp’ upto the position of the target interval and mark each visible point detected.
10. End

3.4 Viewshed computation

The standard and accurate method for the computation of the viewshed is R3 viewshed computation method. In this method, each elevation grid in the Digital Elevation Model (DEM) within the area of interest is tested individually for its visibility. The elevation profile along the direction joining the observer’s point and each grid position is determined and visibility analysis is carried out along this profile.

The observer latitude/longitude and range of area to be considered for viewshed analysis is input by the user. From the provided range of area, the number of rows and columns to be considered in the DEM is determined. For every grid in the DEM, an elevation profile is generated from the observer to that grid. Using the LOS computation method, the visibility of each grid is determined.

The algorithm for viewshed computation is as follows:

```
1. Start
2. Set observer co-ordinate, user_lat and user_long
3. Set boundary co-ordinates calculated from the range of area to be considered
4. Obtain row size (row_num) and column size (col_num) of the elevation model from boundary co-ordinates
5. For i = 1 to row_num
    a. For j = 1 to col_num
        i. Generate elevation profile from observer co-ordinate to the grid ($i^{th}$ row, $j^{th}$ column)
        ii. Check visibility of the grid point using LOS computation method
6. Generate viewshed by plotting the each visible grids
7. End
```

3.5 Proposed algorithm for faster viewshed computation

Using the faster LOS computation method, the time required for the generation of the viewshed is reduced as some computation steps of the shadow region are saved. Hence, the viewshed algorithm works faster when the faster LOS computation method is applied with slight modification primarily focusing the R3 algorithm.

The algorithm for faster viewshed computation is as follows:

```
1. Start
2. Set observer co-ordinate, user_lat and user_long
3. Set boundary co-ordinates calculated from the range of area to be considered
```

```
4. Obtain row size (row_num) and column size (col_num) of the elevation model from boundary co-ordinates
5. For i = 1 to row_num
    a. For j = 1 to col_num
        i. Generate elevation profile from observer co-ordinate to the grid ($i^{th}$ row, $j^{th}$ column)
        ii. Determine the interval point that is a maxima and lies above the LOS line in the elevation profile.
            • If the slope of the interval point is greater than previous highest slope of the visible point, then the point is visible, else, the point is invisible. If the point is visible, update the slope as highest slope of the visible point.
            • Increment interval position by 2 as two maximas cannot occur consecutively.
6. Check the visibility from last maxima that lies above the LOS line upto the intermediate grid interval using LOS computation method.
7. If the final target grid is visible, mark that grid as visible. Else, mark the grid as invisible
8. Plot the visibility profile of each grid for viewshed generation
9. End
```

The faster viewshed computation method starts drawing a straight line from the observer to the intermediate grid, which is also known as LOS line. This process is preceded by generating the elevation profile from the observer to the intermediate grid. Now, in this elevation profile, the interval point whose elevation value is greater than the minimum value of LOS line is determined.

When the interval point satisfies the above criterion, it is further checked to know if that interval point is a maxima. If both criteria are satisfied, i.e. the interval point is above LOS line and is a maxima, then, slope of this point is compared with the previous highest slope of the latest visible point. If the slope of this interval point is greater than the previous highest slope, the interval point is marked as ‘visible’.

The test is repeated for every interval points in the elevation profile of the intermediate grid. After the last point is determined that satisfies the above two conditions, the general LOS computation is performed for visibility analysis from that maxima point upto the interval position of that intermediate grid. At the end of this process, if the interval position of the grid is found visible, the intermediate grid is labeled as ‘visible’. Else, the intermediate grid is labeled as ‘invisible’

The above process is performed for all the intermediate grids to determine their visibility. This would generate the complete viewshed of the region. This method of viewshed computation is faster as the method does not need to consider every interval points of the intermediate grid for the visibility point. The visibility analysis is conducted only for interval points that lie above the minimum value of LOS line and are maximas. Hence, a faster method for viewshed generation is obtained.

CHAPTER 4

METHODOLOGY

4.1 Digital Elevation Modeling (DEM)

For Digital Elevation Modeling (DEM), the SRTM 3 arc data has been used. In SRTM-3 data, 1 degree in latitude/longitude corresponds to 1201 rows/columns. The geographic data of the elevation has an approximate resolution of 90 meters in latitude and longitude. The 90 meters resolution is equivalent to 0.000833° in latitude or longitude. As the resolution of the data is 90m, the same resolution has been used in the computation of LOS and the viewshed. The grid-based digital elevation modeling has been used in this study. The geographic elevation database tested and used in this study ranges from latitude of 26° N to 28°N and longitude of 85°E to 87°E covering an area of about 48000 sq. km. The related SRTM-3 tiles are N27E085, N26E085, N28E085 and N28E086.

4.2 Determination of elevation profile

Equal spacing method has been used for the selection of points along the elevation profile.

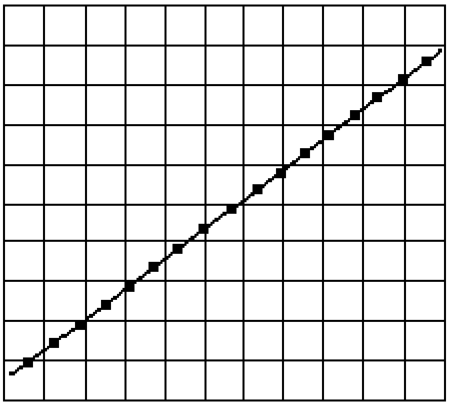

Figure 4.1: Equal spacing of points along profile

In this method, we specify the fixed interval spacing or resolution whose value is set as 90m in accord with the resolution of the SRTM data. For the interval point that does not exactly lie in the elevation grid, the elevation value has been approximated to the nearest row and column of the grid.

4.3 Measurement of computation time

The computational time is measured using standard programming library function. The time required for the computation of LOS is observed to lie in the range of microseconds. The computation time which is in such small range cannot not be measured precisely as other processes running in the background can easily affect the measurement process.

To deal with this fact, the computation of LOS is repeated for 10000 iterations and the computational time taken is measured. Then, the actual computational time is determined by dividing the measured computation time by the number of repetitions carried out.

The processing time required for viewshed computation can be divided into two parts. The first part is the time required for loading of the elevation values of the all intermediate profile lines. The second part is the time required for viewshed analysis of each intermediate profile line.

The time taken for viewshed analysis cannot be measured directly as the computation can be proceeded only after loading of the elevation profiles. So, to measure the computational time taken for viewshed analysis only, the time taken for the first part is subtracted from the total computation time.

In every experimental test, the time taken for the computation is measured for 10 times and the average value of the time taken is considered as the final measurement value.

4.4 Procedures for output comparison and verification

For comparison and verification of the proposed algorithm, the viewshed analysis result obtained from the general method is taken as the reference that represents the correct output.

The output of the viewshed analysis is obtained in terms of the number of visible and invisible points. The visibility results obtained from the proposed algorithm is examined for its similarity with the results obtained from the general method.

The result from the two algorithms is also compared by varying the observer height to vary the number of visible points and to observe its effect in the output.

For the calculation of any possible error, the visibility of each intermediate point/grid as determined by general method and proposed method is compared. If the point that is marked as 'visible' by general method is marked as 'invisible' by the proposed method, then that point is marked as false invisible point. Similarly, if the point that is marked as 'invisible' by general method is marked as 'visible' by the proposed method, then that point is marked as false visible point.

The sum of false visible points and false invisible points would represent any error present in the viewshed generated using the proposed algorithm. The false visible and invisible points are also plotted in the geographic co-ordinate to graphically examine the possible error/s.

For the comparison of the proposed computation method with the general computation method in terms of computational speed, the time taken for generation of the viewshed is determined using both methods and is compared to determine the faster method. Hence, any improvement in the computation of the proposed algorithm is figured out.

The result obtained from both the methods is also compared graphically with the output obtained from industry standard GIS software. ArcGIS has been used as the standard GIS tool for generating standard viewshed outputs.

4.5 Programming platform

The programming work required to compare and verify the results of various algorithms in this research has been carried out in Java Version 7 in 32 bit Windows 7 operating system. Eclipse IDE for Java developer has been used for the development of the program along with WindowBuilder Core Java package for Graphical User Interface.

4.6 Specification of the hardware

The following is the specification of the hardware in which the computational tests and experiments have been carried out:

Hardware type	Laptop
Processor	Intel® Core™ i7-2620M CPU @ 2.70Ghz 2.70 Ghz
Installed memory (RAM)	4.00GB (3.24 GB usable)
System type	32-bit Operating System

CHAPTER 5

RESULTS AND DISCUSSIONS

5.1 Outputs of LOS computation obtained at different terrains

The computation of LOS has been carried out in varying terrains to include the geography of plain lands, hills and the mountains. The distance between the observer and the target has been varied from nearly 40 km upto 110 km. The offset height of the observer has been set to 20m. This value has been chosen mainly to rise above any deviation in elevation value that would arise from differences in interpolation scheme.

The input parameters used for the experiments to compute LOS at various terrains has been show in Table 5.1.1.

Table 5.1.1 Input parameters used for the test

Experiment No.	Observer Co-ordinate (Lat/Long/ Elevation)	Target Co-ordinate (Lat/Long/ Elevation)	Observer Offset Height	Distance between Observer and Target
1	27.155375°N 85.037994°E 136 m	27.895387°N 85.701864°E 3195 m	20 m	107.37 km
2	27.352870°N 85.981451°E 468 m	27.934648°N 85.147450°E 603 m		109.82 km
3	27.974115°N 85.866868°E 4055 m	27.570912°N 85.406289°E 2759 m		66.11 km
4	28.371009°N 85.746943°E 6751 m	28.946363°N 85.795484°E 5645 m		62.36 km

Experiment No.	**Observer Co-ordinate (Lat/Long/ Elevation)**	**Target Co-ordinate (Lat/Long/ Elevation)**	**Observer Offset Height**	**Distance between Observer and Target**
5	26.076257°N 85.048036°E 58 m	26.840064°N 85.620534°E 90 m	20 m	103.09 km
6	27.944487°N 85.992231°E 3256 m	27.306795°N 85.003897°E 441 m		127 km
7	27.570912°N 85.406289°E 2759 m	27.829444°N 85.103715°E 1540 m		42.98 km
8	28.983482°N 85.584188°E 4580 m	28.436682°N 85.811188°E 5926 m		63.94 km
9	26.076257°N 85.048036°E 58 m	26.426037°N 85.305018°E 51 m		46.88 km
10	26.076257°N 85.048036°E 58 m	26.538824°N 85.389250°E 66 m		62.08 km
11	27.570912°N 85.406289°E 2759 m	27.843007°N 85.004313°E 538 m		52.42 km
12	27.570912°N 85.406289°E 2759 m	27.925452°N 85.002375°E 1521 m		58.04 km

The time required for the computation of LOS using general LOS method has been presented in Table 5.1.2. Along with time taken, the number of visible and invisible points with the percentage of visibility has been included in the table.

Table 5.1.2 Time required for general LOS computation

Experiment No.	Average time taken for computation (μs)	No. of visible points	No. of invisible points	Visibility (%)
1	30.93	55	1137	4.61
2	29.85	16	1204	1.31
3	18.00	176	558	23.98
4	17.30	191	501	27.60
5	29.31	130	1015	11.35
6	36.13	23	1388	1.63
7	11.17	222	255	46.54
8	18.28	332	378	46.76
9	12.65	87	433	16.73
10	17.77	110	579	15.97
11	13.65	237	345	40.72
12	15.36	243	401	37.73

The time required for LOS computation using the faster LOS method has been presented in Table 5.1.3. The number of visible and invisible points determined from the computation along with the number false visible and false invisible points has been presented.

Table 5.1.3 Time required for faster LOS computation

Experiment No.	Average time taken for computation (μs)	No. of visible points	No. of invisible points	False visible points	False invisible points
1	28.49	55	1137	0	0
2	27.29	12	1208	0	4

Experiment No.	Average time taken for computation (µs)	No. of visible points	No. of invisible points	False visible points	False invisible points
3	17.59	176	558	2	2
4	17.07	189	503	0	2
5	27.29	130	1015	0	0
6	32.56	23	1388	0	0
7	12.00	222	255	0	0
8	17.83	329	381	1	4
9	12.81	87	433	0	0
10	17.14	110	579	0	0
11	14.29	235	347	0	2
12	15.70	243	401	0	0

The comparison of time taken for computation of LOS along with visibility analysis between the general and faster LOS method at various terrains has been presented in Table 5.1.4. The computation of factor of improvement brought over by the faster LOS method has also been included in the table. It can be seen from the table that, the faster LOS method brings improvement for the target distance greater than 60 km.

Table 5.1.4 Comparison of the computation time

Experiment No.	Average time taken for computation (µs)		Improvement in computational time using Faster LOS (%)
	Using General LOS	Using Faster LOS	
1	30.93	28.49	7.89
2	29.85	27.29	8.58
3	18.00	17.59	2.28

Experiment No.	Average time taken for computation (μs)		Improvement in computational time using Faster LOS (%)
	Using General LOS	Using Faster LOS	
4	17.30	17.07	1.33
5	29.31	27.29	6.89
6	36.13	32.56	9.88
7	11.17	12.00	-7.43
8	18.28	17.83	2.46
9	12.65	12.81	-1.26
10	17.77	17.14	3.55
11	13.65	14.29	-4.69
12	15.36	15.70	-2.21

The observed number of false visible and false invisible points is summed up for the calculation of the error in the visibility analysis. The numbers of these false points are very small and negligible. The observed data for the false points has been given in Table 5.1.5.

Table 5.1.5 Error Calculation for faster LOS computation

Experiment No.	Total interval points	No. of false visible points	No. of false invisible points	Total error points	Error in Viewshed (%)
1	1192	0	0	0	0.00
2	1220	0	4	4	0.33
3	734	2	2	4	0.54
4	692	0	2	2	0.29
5	1145	0	0	0	0.00
6	1411	0	0	0	0.00
7	477	0	0	0	0.00

Experiment No.	Total interval points	No. of false visible points	No. of false invisible points	Total error points	Error in Viewshed (%)
8	710	1	4	5	0.70
9	520	0	0	0	0.00
10	689	0	0	0	0.00
11	582	0	2	2	0.34
12	644	0	0	0	0.00

5.2 Analysis of the LOS outputs at different terrains

The experiments to test the performance of the faster LOS computation method have been carried out in various geographic terrains including the plain, hilly and the mountainous landscapes with elevation varying from 58 m to nearly 6800 m.

From the experiments, significant improvement in the computation time has been observed when the faster LOS method is implemented over the general LOS method.

It was found from the experiments that the time taken by faster LOS method was equal or even greater than the time taken by general LOS when the target distance was less than 75 km. From the test, it was also found that the faster LOS yields significant gain for target distance greater than 90 km.

This result can be accounted for less probability of occurrence of shadow regions in short target distance and higher probability of occurrence of shadow regions in the greater target distance. The presented results of the experiments have also reflected this fact. Considering higher target distance of 60km or more, the average percentage of the improvement was observed to be nearly 5%.

The error incurred in the faster computation of the LOS can be considered as negligible as the percentage of the error was below 0.2%. The reason for this error can be accounted for very few points inside the shadow regions that appear to be visible usually because of error incurred during interpolation and elevation approximation.

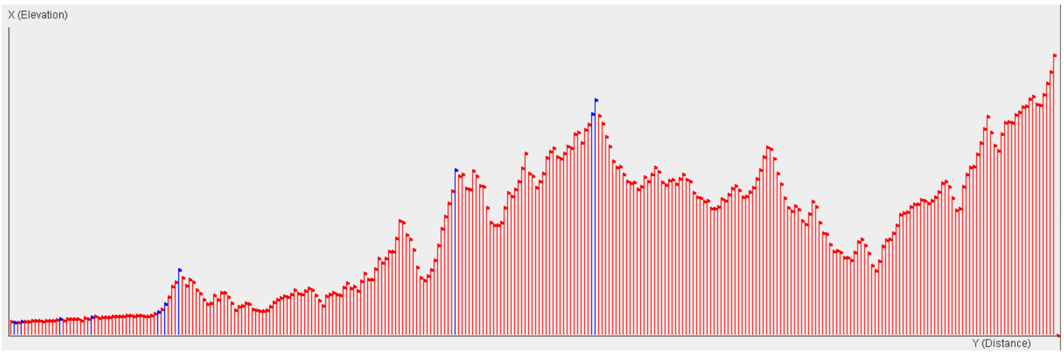

Figure 5.1: Elevation profile of Experiment 1

(The elevation profile of other experiments have been presented in Appendix A.1)

In the generated elevation profiles, the observer is located at the left-most position and the target is located at the right-most position of the graph. The blue lines represent the visible intervals whereas the red lines represent the invisible intervals.

5.3 Outputs of LOS computation at various visibilities

The input parameters for computation of LOS and its basic visibility analysis at various visibilities have been shown in Table 5.3.1.

The visibility has been varied from 0 % to approximately 95 % covering almost full range of the elevation profile under study. The visibility has been varied by varying the offset height of the observer.

Table 5.3.1 Input parameters used for the test

Experiment No.	Observer Co-ordinate (Lat/Long/ Elevation)	Target Co-ordinate (Lat/Long/ Elevation)	Approx. visibility obtained by varying observer offset height	Distance between Observer and Target
1	26.076257°N 85.048036°E 58 m	26.840064°N 85.620534°E 90 m	0% / 0 m	103.09 km
2			10% / 20 m	
3			25% / 65 m	
4			36% / 130 m	
5			50% / 230 m	
6			65% / 420 m	
7			80% / 1000 m	
8			95% / 1650 m	

The time taken data read for computation of LOS has been presented in Table 5.3.2. The numbers of visible and invisible points are also shown.

Table 5.3.2 Time required for general LOS computation

Experiment No.	Average time taken for computation (μs)	No. of visible points	No. of invisible points
1	28.44	6	1139
2	29.31	130	1015
3	30.30	286	859
4	30.68	415	730
5	30.43	570	575
6	30.75	739	406
7	30.18	904	241
8	29.89	1085	60

The time taken for faster LOS computation along with its visibility has been shown in Table 5.3.3. It can be observed that that, for all eight experiments conducted, no false visible point or false invisible point was observed.

Table 5.3.3 Time required for faster LOS computation

Experiment No.	Average time taken for computation (µs)	No. of visible points	No. of invisible points	False visible points	False invisible points
1	26.41	6	1139	0	0
2	27.29	130	1015	0	0
3	28.26	286	859	0	0
4	28.67	415	730	0	0
5	29.10	570	575	0	0
6	29.46	739	406	0	0
7	29.24	904	241	0	0
8	29.65	1085	60	0	0

The comparison of time taken for computation of LOS between general and faster LOS method has been presented in Table 5.3.4. The measure of improvement using the faster method has also been calculated in the table. It can be seen from the table that factor of improvement is greater when the visibility is lesser.

Table 5.3.4 Comparison of the computation time

Experiment No.	Average time taken for computation (µs)		Improvement in computational time using Faster LOS (%)
	Using General LOS	Using Faster LOS	
1	28.44	26.41	7.14
2	29.31	27.29	6.89
3	30.3	28.26	6.73

Experiment No.	**Average time taken for computation (µs)**		**Improvement in computational time using Faster LOS (%)**
	Using General LOS	**Using Faster LOS**	
4	30.68	28.67	6.55
5	30.43	29.1	4.37
6	30.75	29.46	4.20
7	30.18	29.24	3.11
8	29.89	29.65	0.80

5.4 Analysis of the LOS outputs at various visibilities

The faster LOS computation method was also studied for its performance by varying the visibility of the profile. The visibility in the elevation profile was varied by varying the observer's offset height. The visibility was varied from completely invisible profile to nearly 95% visible profile.

From the experiments, it was found that the improvement in computation using the faster method was more prominent when the profiles had lesser visibility.

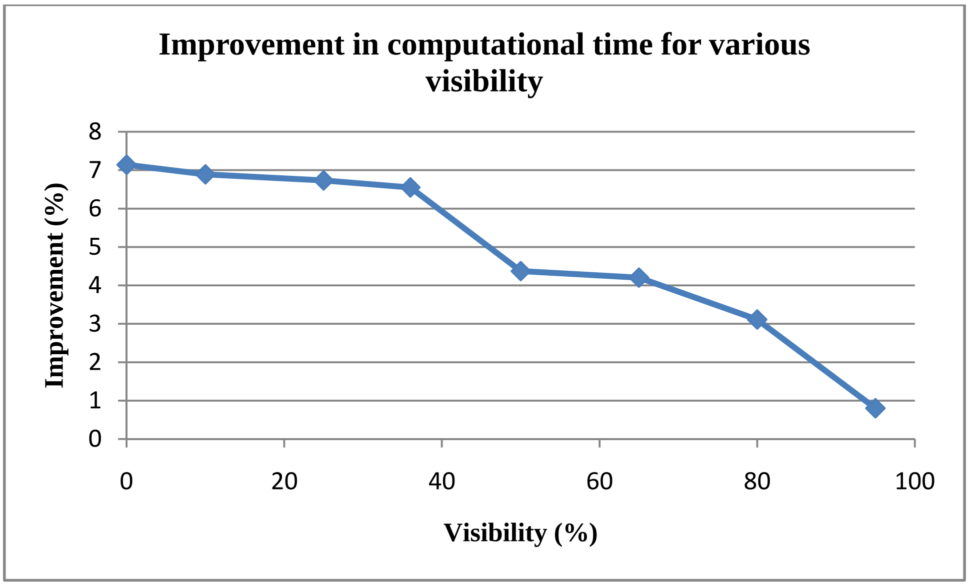

Figure 5.2: Improvement in the computation measured at various visibilities

For profiles with visibility greater than 95%, the computation time taken by general and faster LOS method was nearly equal. The faster LOS method was found to bring significant improvement in computation time when the visibility of the profile was less than 80%.

In all the experiments conducted with varying visibility, no error was observed in the computations performed using the faster LOS method.

5.5 Outputs of viewshed computation at various terrains

The computation of viewshed at various terrains has been carried out. The details of the areas considered for the test have been shown in Table 5.5.1. The areas for the experiments have been chosen to include varying geography. An average area of 340 sq. km has been studied and tested in each experiment.

Table 5.5.1 Input parameters used for the test

Experiment No.	Observer Co-ordinate	Area of computation	Observer offset height
1	27.541770°N 85.303562°E	540 rows X 830 columns	20 m
2	27.019912N 85.011929°E	600 rows X 1000 columns	
3	27.570912°N 85.406289°E	500 rows X 700 columns	
4	27.596693°N 85.083194°E	480 rows X 1000 columns	
5	26.119167°N 85.036200°E	480 rows X 1000 columns	
6	28.503620°N 86.295102°E	550 rows X 800 columns	
7	27.650083°N 85.256270°E	400 rows X 800 columns	

Experiment No.	**Observer Co-ordinate**	**Area of computation**	**Observer offset height**
8	26.730689°N 85.132806°E	300 rows X 600 columns	20 m
9	28.361512°N 86.060062°E	300 rows X 800 columns	
10	27.700513°N 85.311870°E	350 rows X 800 columns	

The time taken for loading of required elevation data has been determined and presented in Table 5.5.2. This is the measurement of time required to load elevation value of all intermediate points for entire grids.

Table 5.5.2 Time taken for loading of elevation data

Experiment No.	**Average time taken for loading of elevation data (ms)**
1	4760.52
2	7453.12
3	3286.72
4	5571.31
5	5569.89
6	4587.83
7	2976.53
8	1269.81
9	2093.94
10	2511.82

The measurement of average time taken for computation of viewshed in each experiment using the general method has been shown in Table 5.5.3. The table also includes the calculation of percentage of visibility for each test. The measurement of time taken obtained includes the elevation loading time as well.

To obtain the average computation time for viewshed computation only, the elevation loading time from Table 5.5.2 has been subtracted from the actual measurement.

Table 5.5.3 Time taken for computation using general viewshed method

Experiment No.	**Average time taken for computation (ms)**		**No. of visible points**	**No. of invisible points**	**Visibility (%)**
	Including elevation loading time	**Excluding elevation loading time**			
1	5995.32	1234.80	56714	392857	12.62
2	10428.13	2975.01	62634	538967	10.41
3	4320.30	1033.58	71017	280184	20.22
4	6799.64	1228.33	33101	448380	6.87
5	7600.48	2030.59	36768	444723	7.64
6	5628.34	1040.51	37429	403992	8.48
7	3879.56	903.03	47152	274049	14.68
8	1778.28	508.47	17826	163975	9.81
9	2433.91	339.97	197	240904	0.08
10	3170.68	658.86	10897	270254	3.88

The average time taken for computation of viewshed using the proposed method has been presented in Table 5.5.4. The time taken has been presented with and without the elevation loading time. The number of false visible and false invisible points for error calculation has also been obtained.

Table 5.5.4 Time taken for computation using faster viewshed method

Experiment No.	Average time taken for computation (ms)		No. of visible points	No. of invisible points	False visible points	False invisible points
	Including elevation loading time	Excluding elevation loading time				
1	5482.44	721.92	55172	394399	0	1542
2	8007.58	554.46	56896	544705	16	5754
3	4049.53	762.81	58846	292355	11	12182
4	6207.75	636.44	29478	452003	14	3637
5	6281.47	711.58	32891	448590	0	3877
6	5340.27	752.44	35711	405640	28	1746
7	3153.72	177.18	37370	283831	10	9792
8	1492.24	222.43	14315	166586	0	3511
9	2250.51	156.57	223	240878	28	2
10	2794.42	282.60	9509	271642	42	1430

In Table 5.5.5, the overall time required for viewshed computation using general and proposed method has been compared. The comparison has been done considering the time required for viewshed computation including the elevation loading time. The highest overall improvement obtained was nearly 23% while the lowest observed improvement was about 5%.

Table 5.5.5 Comparison of computation time (including elevation loading time)

Experiment No.	Average time taken for computation (ms)		Improvement in computational time using Faster Viewshed (%)
	Using General Viewshed	Using Faster Viewshed	
1	5995.32	5482.44	8.55

Experiment No.	Average time taken for computation (ms)		Improvement in computational time using Faster Viewshed (%)
	Using General Viewshed	Using Faster Viewshed	
2	10428.13	8007.58	23.21
3	4320.3	4049.53	6.27
4	6799.64	6207.75	8.70
5	7600.48	6281.47	17.35
6	5628.34	5340.27	5.12
7	3879.56	3153.72	18.71
8	1778.28	1492.24	16.09
9	2433.91	2250.51	7.54
10	3170.68	2794.42	11.87

The computation time taken for viewshed computation only has been compared between the general and faster method. The measured computation time has been presented in Table 5.5.6. The percentage of improvement brought by the proposed viewshed method has also been calculated and shown in the table. As can be seen from the table, in experiment no. 2, the percentage of improvement in viewshed computation was as high as 81%. The average percentage of improvement observed in the experiments was 24%.

Table 5.5.6 Comparison of computation time (excluding elevation loading time)

Experiment No.	Average time taken for computation (ms)		Improvement in computational time using Faster Viewshed (%)
	Using General Viewshed	Using Faster Viewshed	
1	1234.8	721.92	41.54
2	2975.01	554.46	81.36
3	1033.58	762.81	26.20

Experiment No.	Average time taken for computation (ms)		Improvement in computational time using Faster Viewshed (%)
	Using General Viewshed	Using Faster Viewshed	
4	1228.33	636.44	48.19
5	2030.59	711.58	64.96
6	1040.51	752.44	27.69
7	903.03	177.18	80.38
8	508.47	222.43	56.26
9	339.97	156.57	53.95
10	658.86	282.6	57.11

The calculation of error for the viewshed generated using the proposed algorithm has been presented in Table 5.5.7. Total error in computation has been obtained by adding the number of false visible and false invisible points. The percentage of error has been calculated by dividing the total error with total number of interval points/grids. From the table, it can be noticed that the percentage of error in the experiments is not so substantial with average error percentage being less than 1.25%.

Table 5.5.7 Error calculation for the viewshed generated

Experiment No.	Total no. of interval points	No. of false visible points	No. of false invisible points	Total error points	Error in viewshed computation (%)
1	449571	0	1542	1542	0.34
2	601601	16	5754	5770	0.96
3	351201	11	12182	12193	3.47
4	481481	14	3637	3651	0.76
5	481491	0	3877	3877	0.81
6	441421	28	1746	1774	0.40

Experiment No.	Total no. of interval points	No. of false visible points	No. of false invisible points	Total error points	Error in viewshed computation (%)
7	321201	10	9792	9802	3.05
8	181801	0	3511	3511	1.93
9	241101	28	2	30	0.01
10	281151	42	1430	1472	0.52

5.6 Analysis of the viewshed outputs at various terrains

Various viewshed results were obtained to assess the performance of the faster viewshed algorithm on wide geographic terrains. The viewshed computation process consists of two parts: the first part being the task of loading the required elevation profiles and the second part being actual viewshed analysis operation.

The time taken for the first part of the viewshed process is equal for both the general viewshed algorithm and faster viewshed algorithm. So, the comparison of the computational time taken for the processing can be done based on time required for the execution of the second part, i.e. analysis of the visibility.

The improvement in computation is greater when the time taken for viewshed generation only is considered. This consideration would only account for the actual time taken for viewshed computation and does not consider the elevation loading time as the elevation loading time is the basic task required by both the algorithms.

It has also been observed from experiments that, in general, 70% of the total computation time is spent on the process of loading of the elevation value.

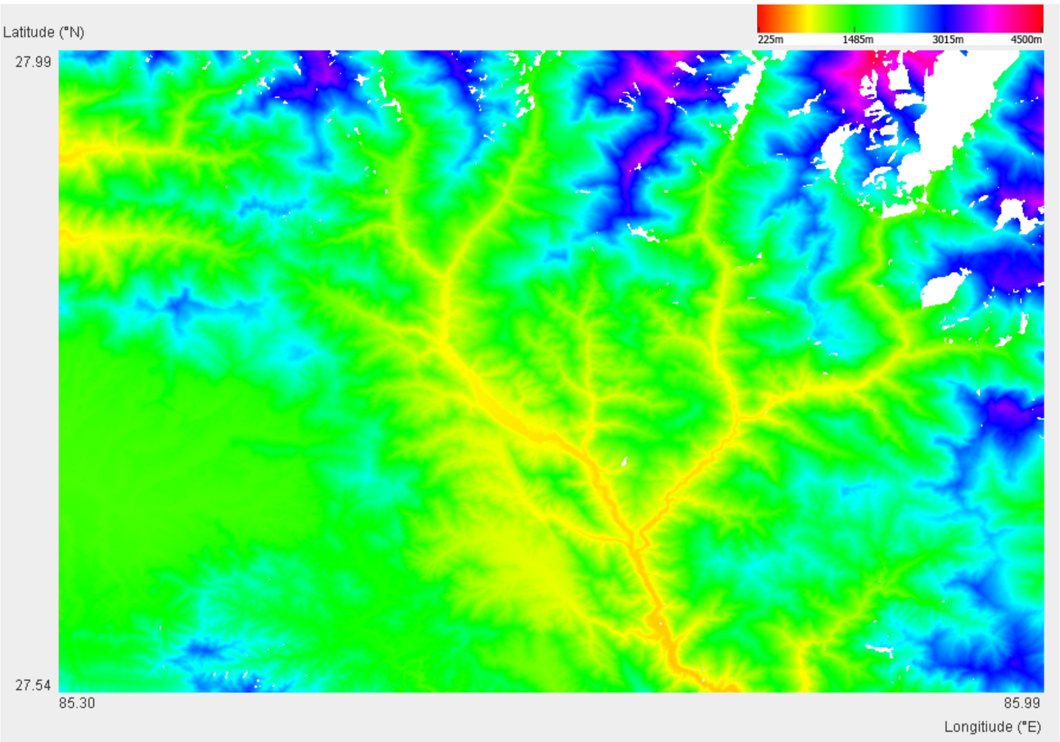

Figure 5.3: Digital Elevation Model (DEM) for Experiment 1

(The DEM of other experiments have been presented in Appendix A.2)

From the experiments conducted, it was observed that the computation vividly became faster when the faster viewshed method was implemented. In one of the experiment, an improvement in computation speed as high as 81 % was also found. The lowest observed improvement was 26%. The average improvement in the computational speed was nearly 54%.

The greatest error incurred in the computation was less than 3.5% which can be accounted for the very few visible points lying in the shadow which is usually because of error caused due to interpolation and elevation approximation. The average error was observed to be 1.23%.

For the analysis, the observer is supposed to be positioned at the bottom-left corner of the graph.

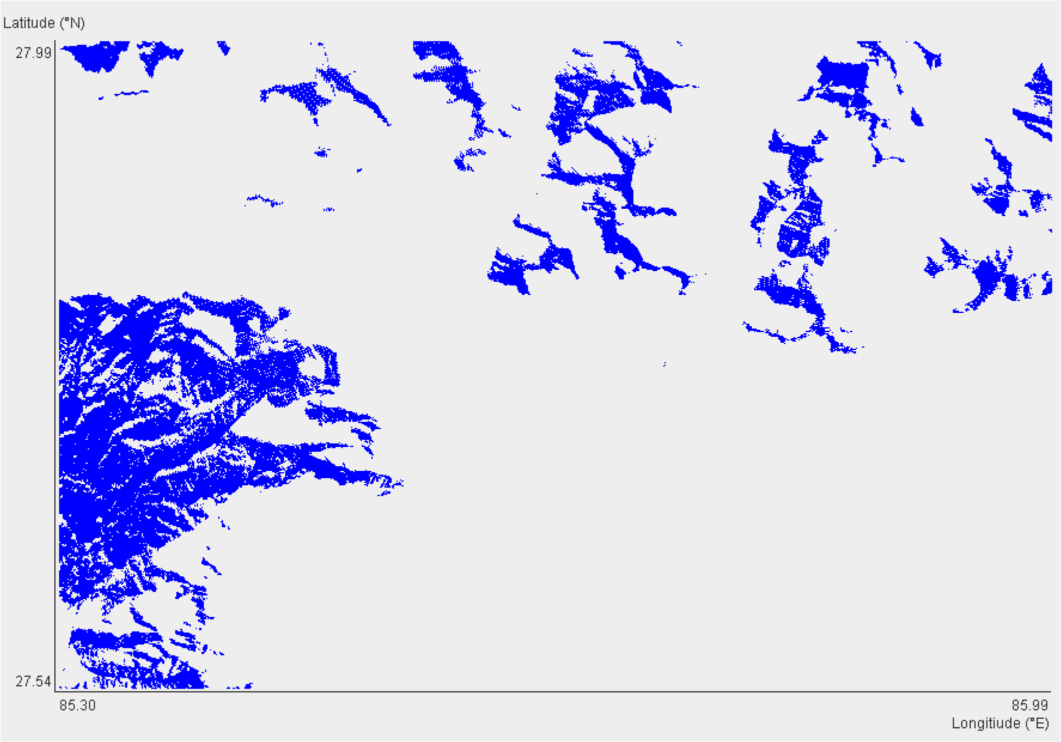

Figure 5.4: Viewshed generated in Experiment 1

(The viewshed generated in other experiments have been presented in Appendix A.2)

In figure 5.5, the blue points in the viewshed diagram represent the visible points in the elevation model. The red points, which are practically invisible, because of very small / insignificant error count represent the error in computation of the viewshed incurred when the proposed algorithm is used.

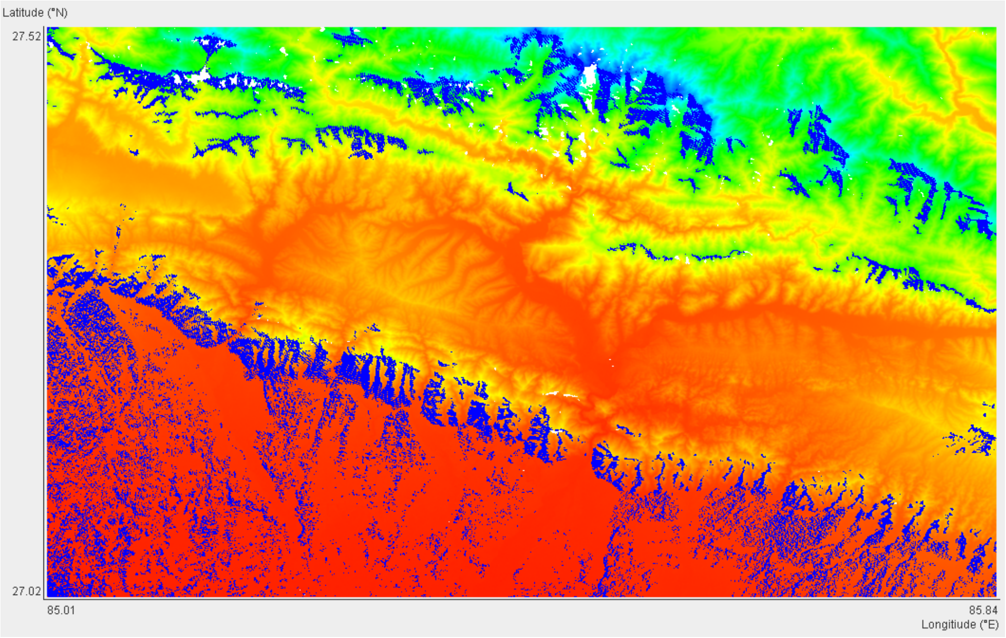

Figure 5.5: Graphical analysis of the generated viewshed for Experiment 2

5.7 Outputs of viewshed computation at various visibilities

For studying the performance of the proposed algorithm at varying visibility, experiments have been conducted by generating the viewshed of the same area with different percentage of visibility. The test has been carried out in an area of nearly 540 sq. km. The visibility has been varied from approximately 3 % to 84 % by supposedly varying the offset elevation height of the observer. The details of the area considered for this study has been presented in Table 5.7.1.

Table 5.7.1 Input parameters used for the test

Experiment No.	**Observer Co-ordinate**	**Area of computation**	**Approx. visibility obtained by varying observer offset height**
1	27.019912°N 85.011929°E	600 rows X 1000 columns	2.71% / 0m
2			10.41% / 20 m
3			16.26% / 100m

Experiment No.	Observer Co-ordinate	Area of computation	Approx. visibility obtained by varying observer offset height
4	27.019912°N 85.011929°E	600 rows X 1000 columns	28.19% / 500m
5			37.69% / 1200m
6			43.95% / 2000m
7			52.30% / 3500m
8			59.62% / 5500m
9			67.21% / 8500m
10			83.16% / 20000m

The average time taken for loading of elevation data has been presented in Table 5.7.2. As the area considered for all experiments is same, the average time taken has been computed only once.

Table 5.7.2 Time taken for loading of elevation data

Experiment No.	Average time taken for loading of elevation data (ms)
1 - 10	7453.12

In Table 5.7.3, the time taken for computation of viewshed at various visibilities has been shown. The computation has been done using the general viewshed method. The actual number of visible and invisible points has also been obtained.

Table 5.7.3 Time taken for computation using general viewshed method

Experiment No.	Average time taken for computation (ms)		No. of visible points	No. of invisible points
	Including elevation loading time	Excluding elevation loading time		
1	8777.29	1324.17	16325	585276
2	10387.09	2933.97	62634	538967

Experiment No.	Average time taken for computation (ms)		No. of visible points	No. of invisible points
	Including elevation loading time	Excluding elevation loading time		
3	10921.16	3468.04	97839	503762
4	10432.27	2979.15	169599	432002
5	9647.05	2193.93	226758	374843
6	9431.13	1978.01	264410	337191
7	9543.52	2090.40	314649	286952
8	9561.48	2108.36	358648	242953
9	9569.75	2116.63	404363	197238
10	9179.82	1726.70	500308	101293

The time taken for computation of viewshed using the faster viewshed method at different visibility cases has been presented in Table 5.7.4. From the experiment, the computation time taken for overall process including the elevation load time has been obtained. Then, the elevation loading time obtained from Table 5.7.2 is subtracted from the overall computation time to obtain the actual time taken for viewshed process only. The number of visible and invisible points obtained from the experiments has also been presented in the table.

Table 5.7.4 Time taken for computation using faster viewshed method

Experiment No.	Average time taken for computation (ms)		No. of visible points	No. of invisible points	False visible points	False invisible points
	Including elevation loading time	Excluding elevation loading time				
1	8035.43	582.31	14573	587028	78	1830
2	8249.94	796.82	56999	544602	119	5754

Experiment No.	Average time taken for computation (ms)		No. of visible points	No. of invisible points	False visible points	False invisible points
	Including elevation loading time	Excluding elevation loading time				
3	8752.67	1299.55	88896	512605	108	8951
4	8010.16	557.04	155983	445618	91	13707
5	8263.47	810.35	210239	391362	58	16577
6	9370.69	1917.57	245472	356129	41	18979
7	9605.51	2152.39	291915	309686	131	22865
8	9654.96	2201.84	332659	268942	238	26227
9	9894.89	2441.77	375254	226347	300	29409
10	10377.68	2924.56	466125	135476	277	34460

In Table 5.7.5, the time taken for viewshed computation at various visibility cases has been compared between the general and faster viewshed method. The elevation loading time has been considered in the computation time. The comparison of the overall computation has been shown in the table. It can be seen from the table that the improvements are significant upto the visibility of 44%.

Table 5.7.5 Comparison of computation time (including elevation loading time)

Experiment No.	Average time taken for computation (ms)		Improvement in computational time using Faster Viewshed (%)
	Using General Viewshed	Using Faster Viewshed	
1	8777.29	8035.43	8.45
2	10387.09	8249.94	20.58
3	10921.16	8752.67	19.86
4	10432.27	8010.16	23.22

Experiment No.	Average time taken for computation (ms)		Improvement in computational time using Faster Viewshed (%)
	Using General Viewshed	Using Faster Viewshed	
5	9647.05	8263.47	14.34
6	9431.13	9370.69	0.64
7	9543.52	9605.51	-0.65
8	9561.48	9654.96	-0.98
9	9569.75	9894.89	-3.40
10	9179.82	10377.68	-13.05

The comparison of time taken by general and proposed algorithm for viewshed computation by excluding the elevation loading time has been presented in Table 5.7.6. When the elevation loading time is excluded from the overall computation time, the result obtained is the time required solely for viewshed computation. The improvement in computation time has been observed for experiments with visibility below 50%.

Table 5.7.6 Comparison of computation time (excluding elevation loading time)

Experiment No.	Average time taken for computation (ms)		Improvement in computational time using Faster Viewshed (%)
	Using General Viewshed	Using Faster Viewshed	
1	1324.17	582.31	56.02
2	2933.97	796.82	72.84
3	3468.04	1299.55	62.53
4	2979.15	557.04	81.30
5	2193.93	810.35	63.06
6	1978.01	1917.57	3.06
7	2090.40	2152.39	-2.97

Experiment No.	Average time taken for computation (ms)		Improvement in computational time using Faster Viewshed (%)
	Using General Viewshed	Using Faster Viewshed	
8	2108.36	2201.84	-4.43
9	2116.63	2441.77	-15.36
10	1726.70	2924.56	-69.37

Table 5.7.7 presents the calculation of error for the viewshed generated using the proposed method at different visibility scenarios. The average percentage of error has been computed to be nearly 3%. The percentage in error has been observed to increase as the percentage of visibility increases.

Table 5.7.7 Error calculation for the viewshed generated

Experiment No.	Total no. of interval points	No. of false visible points	No. of false invisible points	Total error points	Error in viewshed computation (%)
1	601601	78	1830	1908	0.32
2		119	5754	5873	0.98
3		108	8951	9059	1.51
4		91	13707	13798	2.29
5		58	16577	16635	2.77
6		41	18979	19020	3.16
7		131	22865	22996	3.82
8		238	26227	26465	4.40
9		300	29409	29709	4.94
10		277	34460	34737	5.77

5.8 Analysis of the viewshed outputs at various visibilities

In the tests conducted to learn the performance improvement by varying the visibility, the visibility was varied from 3% to 84% which was obtained by varying the offset elevation of the observer.

From the test, it has been observed that the viewshed computation using proposed method is faster for the visibility upto 50%. The improvement in computational time is greater for area with less viewshed. In most practical scenario, the visibility of the region is usually far less than 50%. So, the proposed algorithm for faster viewshed computation is applicable for practical viewshed analysis and study.

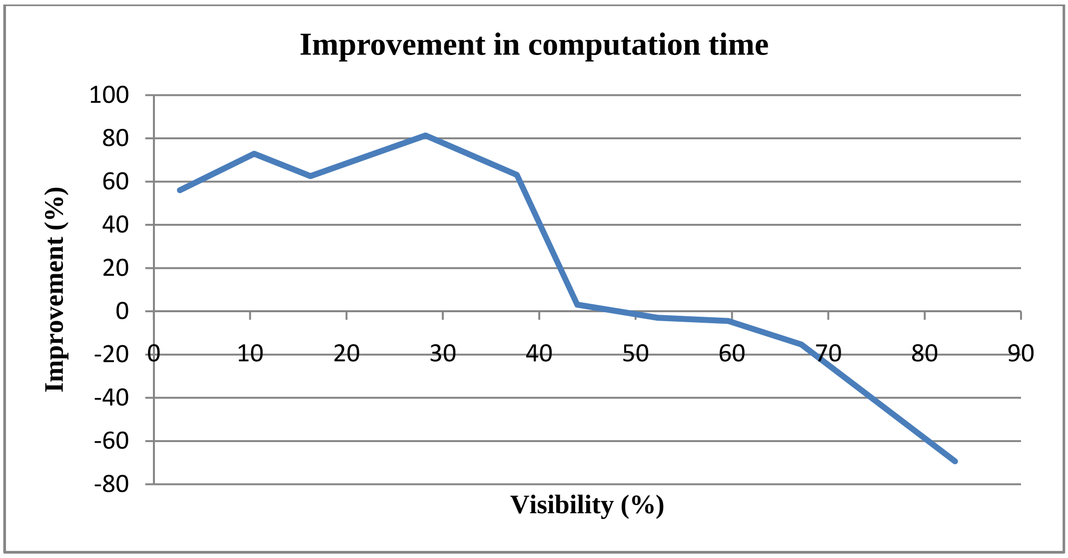

Figure 5.6: Improvement in the computation measured at various visibilities

5.9 Comparison with standard output

The LOS as well as the viewshed generated by the proposed algorithm has been validated by graphically comparing the output with standard output from the ArcGIS.

5.9.1 Comparison of LOS elevation profile generated with standard ArcGIS output

For this comparison, a LOS profile has been generated using the method devised in this study and has been graphically compared with the LOS output from GIS software - ArcGIS. Figure 5.7 presents the standard output obtained from the ArcGIS. The output from the faster LOS method is shown in Figure 5.8. In the presented outputs, the observer and the target is supposed to be on the leftmost and rightmost of the graph respectively. The X-axis represents the distance range whereas the Y-axis represents the elevation value.

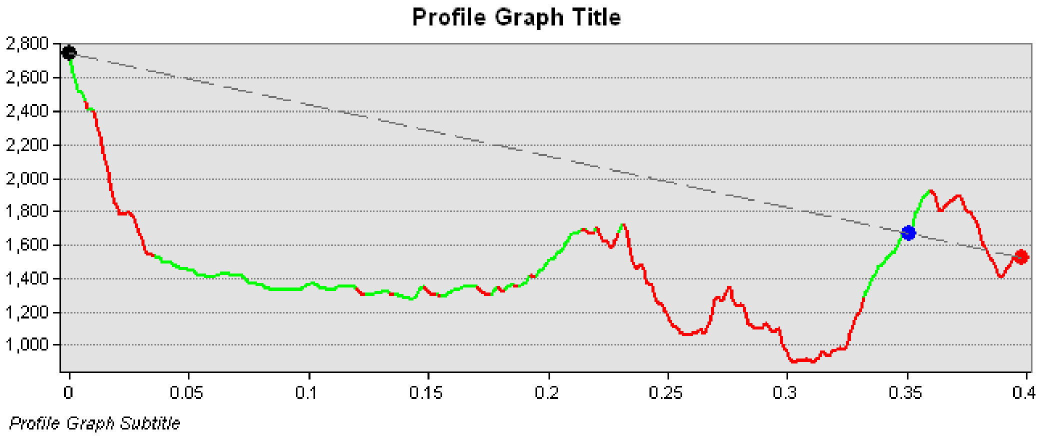

Figure 5.7: Elevation profile with visibility analysis generated using ArcGIS

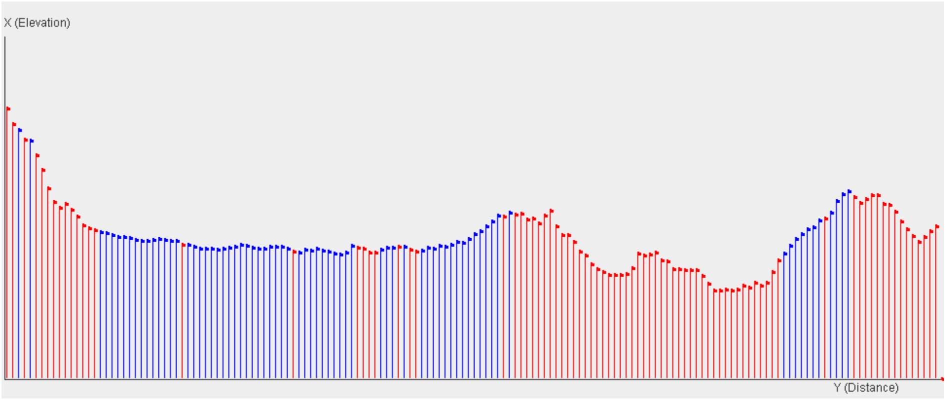

Figure 5.8: Elevation profile with visibility analysis generated using faster LOS

5.9.2 Comparison of viewshed generated with standard ArcGIS output

The viewshed generated from ArcGIS and the proposed algorithm has been shown in Figure 5.9 and Figure 5.10 respectively. For the comparison, the observer has been supposed to be positioned at the bottom-left of the image for both the cases.

Figure 5.9: Viewshed generated from ArcGIS

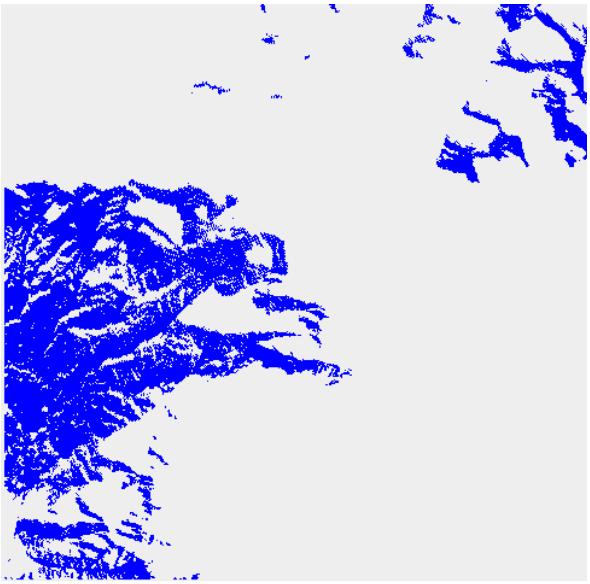

Figure 5.10: Viewshed generated using proposed viewshed algorithm

The mathematically comparison of the results generated from the GIS software is not expected as the exact algorithm and the interpolation method implemented in the software is not known as a part of the software confidentiality and proprietary. Thus, the correctness of the outputs has been graphically verified.

CHAPTER 6

CONCLUSION AND RECOMMENDATION

6.1 Conclusion

The faster viewshed algorithm has been successfully tested and verified to confirm the improvements brought over the general viewshed algorithm. Experimental tests were carried out in the geographic database of 48000 sq. km which included geographic terrains varying from 58 m to nearly 8000 m.

A clear and significant improvement in the performance of the computation of the viewshed using the devised algorithm has been observed. During the experiments, an improvement in overall computational speed as high as 23% was also observed, while the lowest observed improvement was 5%. An average of nearly 12% improvement in the overall viewshed computation was found.

From the experiments and analysis, it has been found that the devised algorithm is computationally faster for region having visibility upto 50%. In most practical cases, the visibility of the region is usually far less than 50%. In general, the visibility of a region would become greater than 50% only when the offset elevation of the observer is raised very high, but this height cannot be realized in the real world. So, the proposed algorithm for faster viewshed computation is applicable for practical viewshed analysis and study.

The rational ground behind the gain in computational time is to be accounted to the process of separating the invisible shadow regions by determining their boundary points. This process has been observed to greatly reduce the number of intermediate intervals for visibility analysis, hence making the computation faster.

The generation of the viewshed of an area would be computed more efficiently in lesser time with the algorithm developed. Hence, a faster viewshed algorithm has been successfully formulated and developed in this study.

6.2 Recommendation

The devised algorithm can be implemented to all viewshed algorithms that are based on Line-of-sight (LOS) method. The popular viewshed algorithm used for approximate viewshed computation – the R2 algorithm would also become faster if the concept of separating shadow regions is applied. The scale of improvement in these various viewshed algorithms can be measured and verified.

The viewshed generated using the proposed algorithm is an exact viewshed of an area. The algorithm can also be modified to generate approximate viewshed by skipping few maximas.

REFERENCES

1. Loukas-Moysis Misthos, 2014, Mountainous Landscape Exploration Visualizing Viewshed Changes In Animated Maps
2. David Izraelevitz, 2003, A Fast Algorithm for Approximate Viewshed Computation, Photogrammetric Engineering and Remote Sensing, Vol. 69, No. 7, July 2003, pp.767-774
3. S. Tabik, A. R. Cervilla, E. Zapata, L. F. Romero, Efficient Data Structure and Highly Scalable Algorith for Total-Viewshed Computation
4. W. Randolph Franklin and Clark Ray, 1994, Higher isn’t necessarily better: Visibility algorithms and experiments. In Proc. 6th Symp. Spatial Data Handling (SDH 1994), pages 751–763, 1994
5. J. Wang, G. J. Robinson and K. White, 2000, Generating viewshed without using sightlines, Photogrammetric Engineering and Remote Sensing, Vol. 66, No.1, pp. 87-90

BIBLIOGRAPHY

1. Benedict, M. L., 1979, To take hold of space: isovists and isovist fields, Environment and Planning B, 6, pp. 47 - 65
2. GRASS GIS manual http://grass.osgeo.org/grass70/manuals/r.viewshed.html
3. H. Haverkort, L. Toma, B. P. Wei, On IO-efficient viewshed algorithms and their accuracy
4. J. Wang, G. J. Robinson and K. White, 1996, A Fast Solution to Local Viewshed Computation Using Grid-Based Digital Elevation Models, Photogrammetric Engineering and Remote Sensing, Vol. 62, No. 10, October 1996, pp. 1157-1164
5. R. Franklin, C. K. Ray, S. Mehta, 1994, Geometric Algorithms for Siting of Air Defense Missile Batteries
6. Tandy, C. R. V., 1967, The isovist method of landscape survey, in Methods of Landscape Analysis Ed. H C Murray (Landscape Research Group, London)
7. U. Pyysalo, J. Oksanen and T. Sarjakoski, 2009, Viewshed Analysis and Visualization of Landscape Voxel Models, Proc. Of 24th Int. Cartographic Conf.
8. Y. Zhi, L. Wu, Z. Sui, H. Cai, 2011, An Improved Algorithm for Computing Viewshed Based on Reference Planes
9. Z. Xu, Q. Yao, 2009, A Novel Algortihm for Viewshed Based on Digital Elevation Model

APPENDIX

A.1 Elevation profile generated for testing LOS computation at different terrains

Some of the elevation profiles generated in course of testing the faster LOS computation at various landscapes have been presented below:

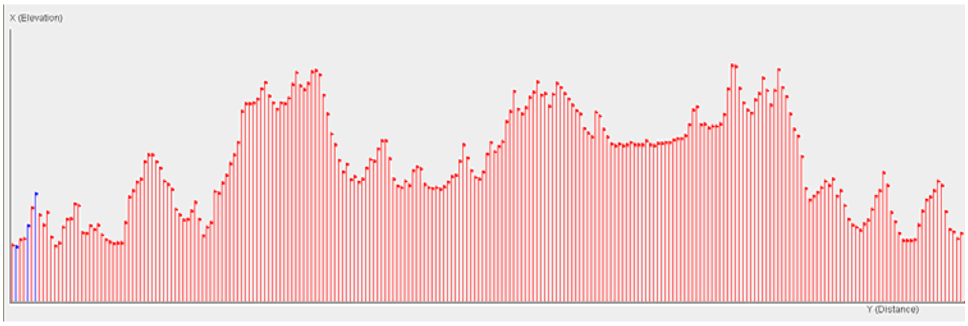

Figure A.1.1: Elevation profile of Experiment 2

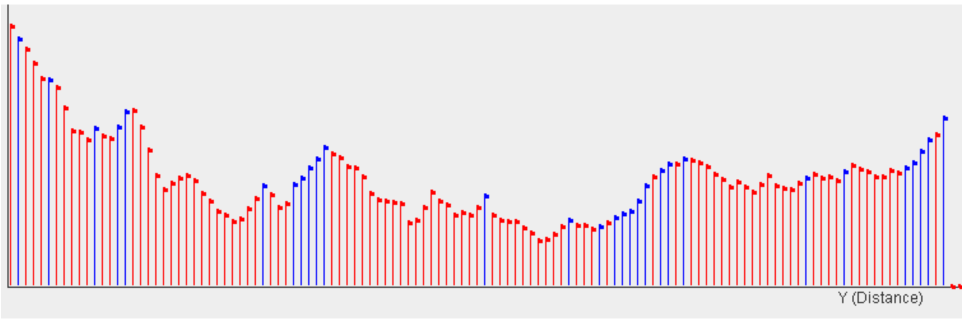

Figure A.1.2: Elevation profile of Experiment 3

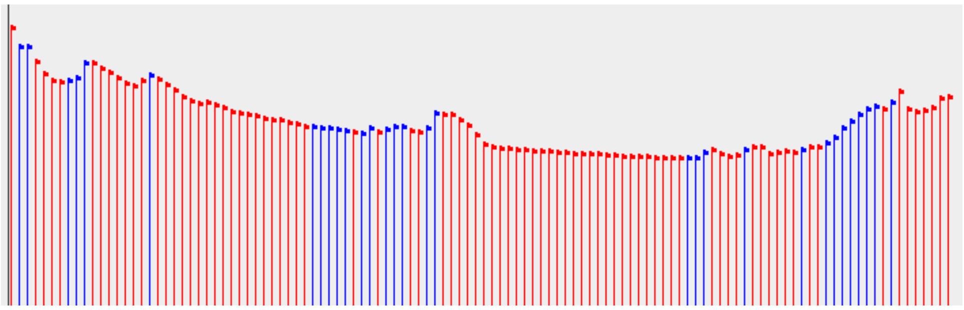

Figure A.1.3: Elevation profile of Experiment 4

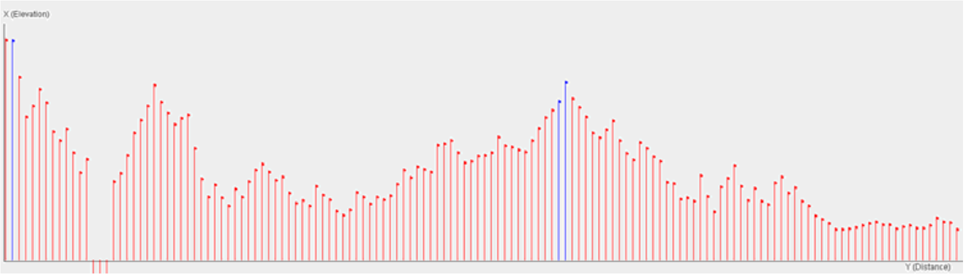

Figure A.1.4: Elevation profile of Experiment 6

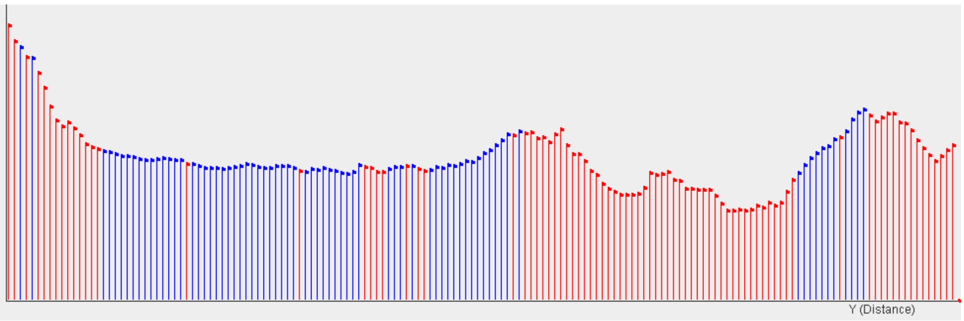

Figure A.1.5: Elevation profile of Experiment 7

A.2 DEM generated for testing viewshed at different terrains

Some of the digital elevation maps generated for testing the performance of proposed viewshed algorithm have been presented below:

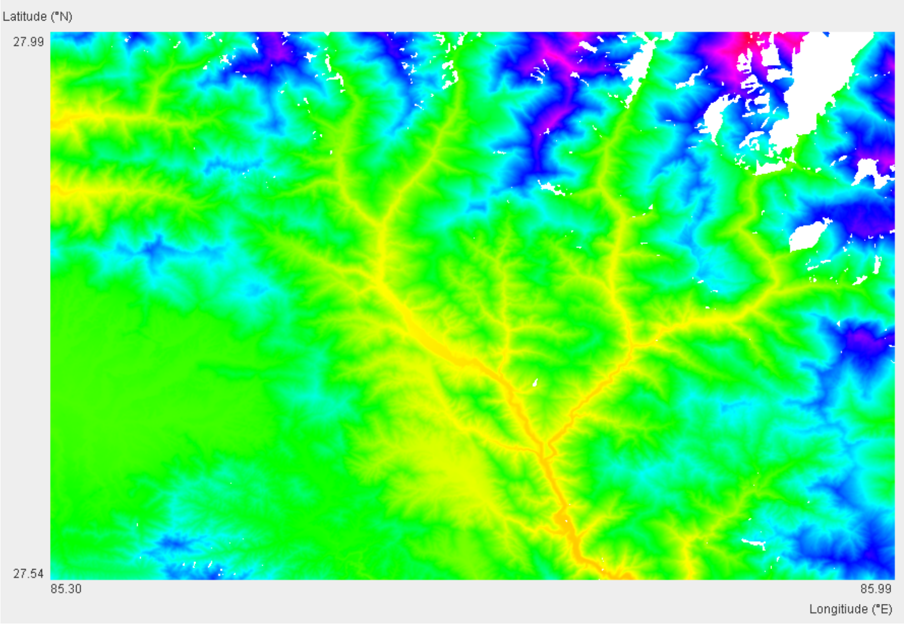

Figure A.2.1: Digital Elevation Model for Experiment 2

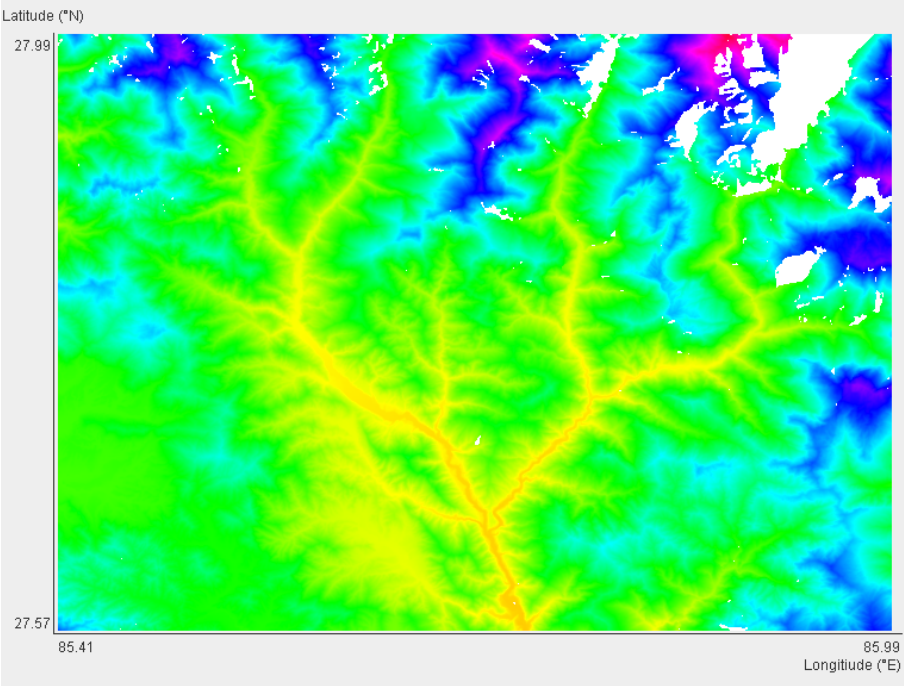

Figure A.2.2: Digital Elevation Model for Experiment 3

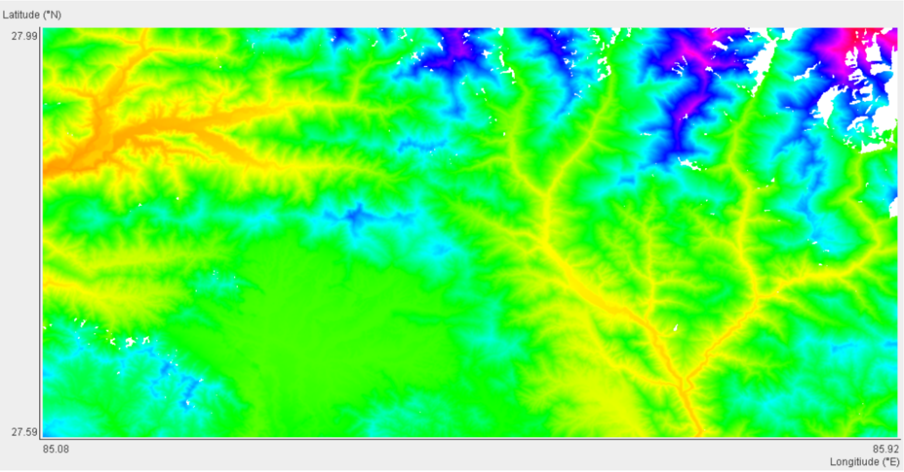

Figure A.2.3: Digital Elevation Model for Experiment 4

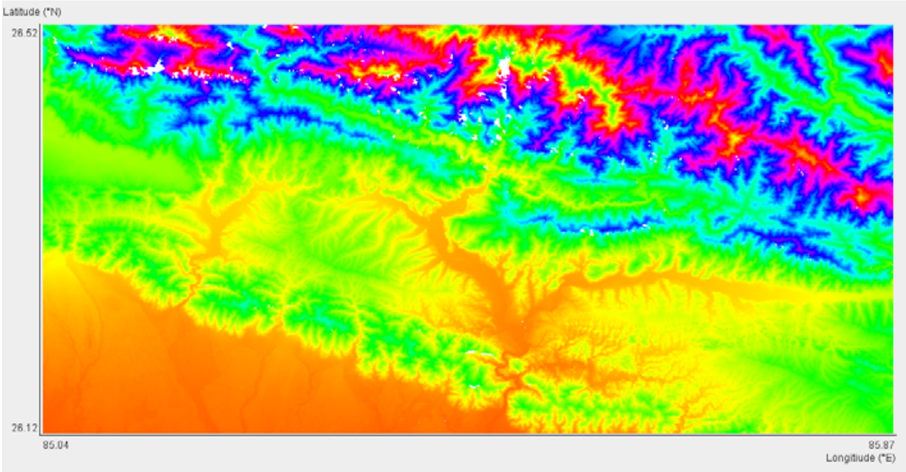

Figure A.2.4: Digital Elevation Model for Experiment 5

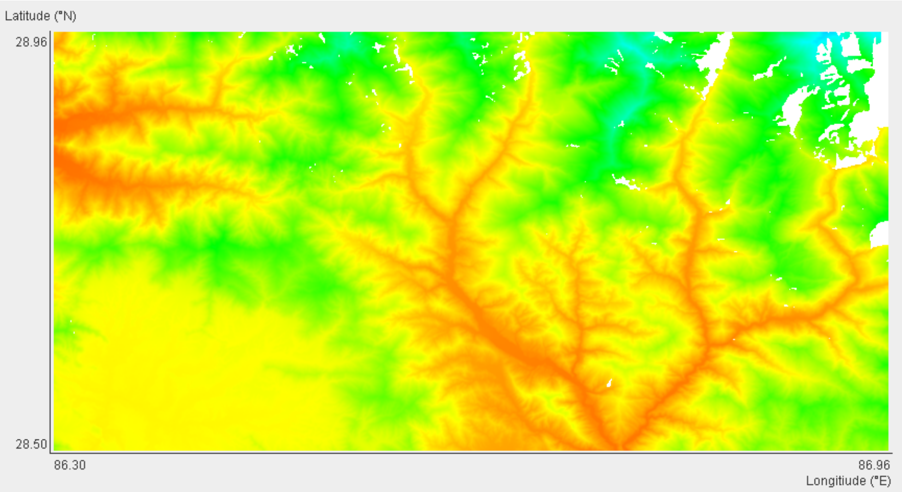

Figure A.2.5: Digital Elevation Model for Experiment 7

A.3 Viewshed generated for testing viewshed computation at different terrains

In course of experiments to analyze the viewshed generated, some of the viewshed outputs generated for graphically verifying the proposed algorithm at various terrains has been presented below:

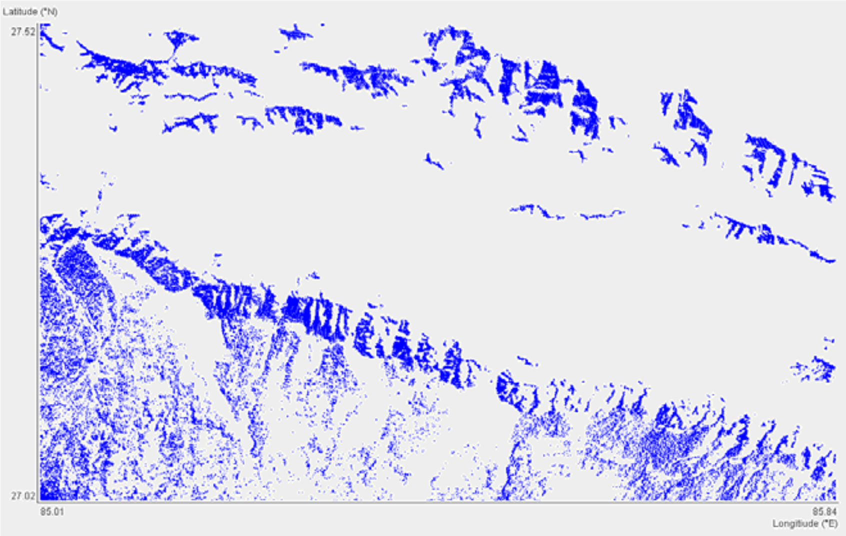

Figure A.3.1: Viewshed generated for Experiment 2

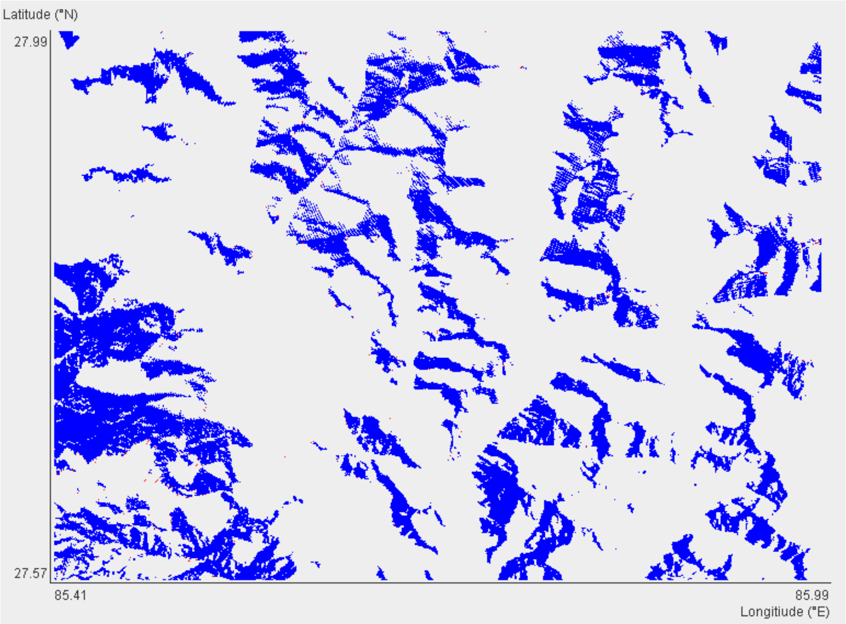

Figure A.3.2: Viewshed generated for Experiment 3

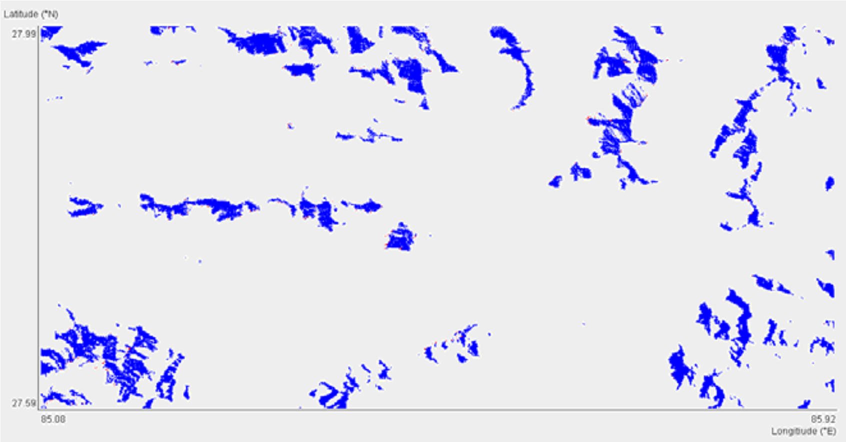

Figure A.3.3: Viewshed generated for Experiment 4

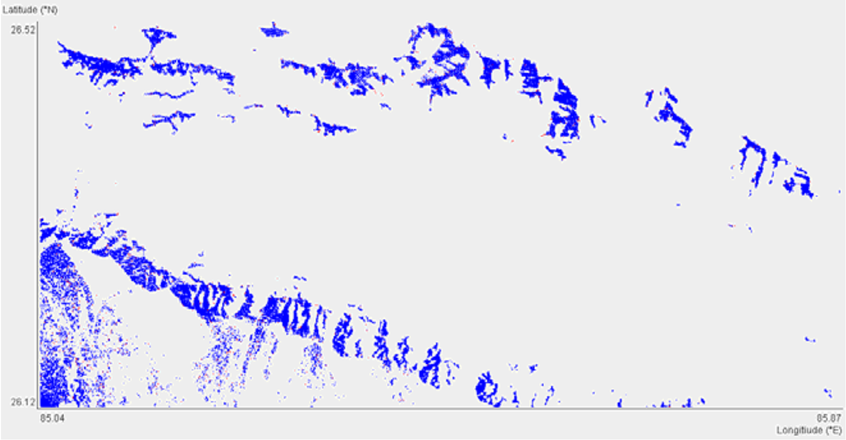

Figure A.3.4: Viewshed generated for Experiment 5

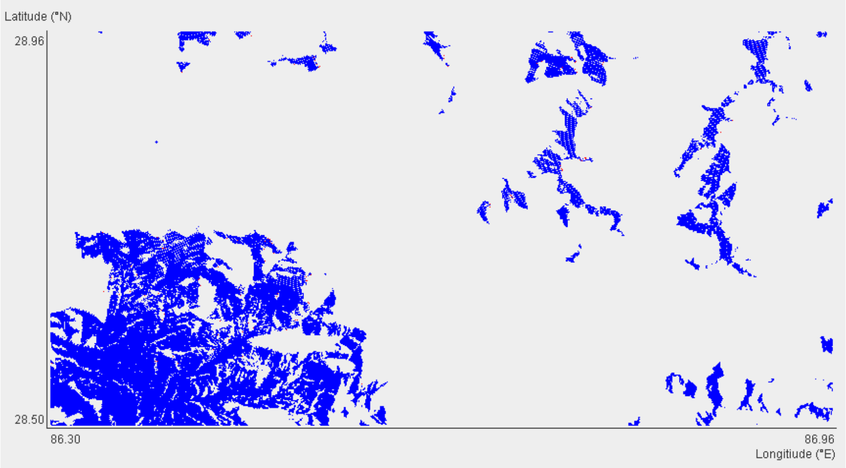

Figure A.3.5: Viewshed generated for Experiment 7

A.4 Shuttle Radar Topographic Mission (SRTM)

The Shuttle Radar Topography Mission (SRTM) is an international research effort that obtained digital elevation models on a near-global scale from 56° S to 60° N, to generate the most complete high-resolution digital topographic database of Earth prior to the release of the ASTER GDEM in 2009. SRTM consisted of a specially modified radar system that flew on board the Space Shuttle Endeavour during the 11-day STS-99 mission in February 2000, based on the older Spaceborne Imaging Radar-C/X-band Synthetic Aperture Radar (SIR-C/X-SAR), previously used on the Shuttle in 1994.

To acquire topographic (elevation) data, the SRTM payload was outfitted with two radar antennas. One antenna was located in the Shuttle's payload bay, the other – a critical change from the SIR-C/X-SAR, allowing single-pass interferometry – on the end of a 60-meter (200-foot) mast that extended from the payload bay once the Shuttle was in space. The technique employed is known as Interferometric Synthetic Aperture Radar.

The elevation models are arranged into tiles, each covering one degree of latitude and one degree of longitude, named according to their south western corners. It follows that "n45e006" stretches from 45°N 6°E to 46°N 7°E and "s45w006" from 45°S 6°W to 44°S 5°W. The resolution of the raw data is one arcsecond (30 m), but this has only been released over United States territory. A derived one arcsecond dataset (with trees and other non-terrain features removed) covering Australia was made available in November 2011; the raw data are restricted for government use. For the rest of the world, only three arcsecond (90 m) data are available. Each one arcsecond tile has 3,601 rows, each consisting of 3,601 16 bit bigendian cells. The dimensions of the three arcsecond tiles are 1201 x 1201.

The Shuttle Radar Topography Mission is an international project spearheaded by the U.S. National Geospatial-Intelligence Agency (NGA) and the U.S. National Aeronautics and Space Administration (NASA).