Finding the Corridor of a Straight Line Segment and its Application in Estimation of error in Area Calculation of a Digital Figure

A dissertation submitted in partial fulfillment of the requirements for the completion of degree of the Master of Technology in Computer Science (2005-2007).

By

Koushik Bhattacharyya

Indian Statistical Institute

Kolkata.

Under The Supervision of

Dr. Bhargab B Bhatacharya

2007

<u>INDIAN STATISTICAL INSTITUTE</u>

203, Barrackpore Trunk Road

Calcutta – 700 108

Diss/07/01/196

Certificate

This is to certify that the dissertation entitled "**Finding the Corridor of a Straight Line Segment and its Application in Estimation of error in Area Calculation of a Digital Figure**" submitted by Koushik Bhatacharyya towards partial fulfillment for the degree of M.Tech in Computer Science at Indian Statistical Institute, Kolkata, embodies the work done under my supervision.

Dated : 2007

Signed:

(Bhargab B Bhattacharya)

Supervisor

External Expert

Diss/07/01/196

Acknowledgement

It is my privilege and pleasure to convey my gratitude to my guide **Dr. Bhargab. B. Bhattacharya** for his guidance, support and encouragement throughout the course of the dissertation without which it would have been impossible to complete this research endeavor. I express my gratitude to **Dr. S.C. Nandy**, ACMU for his constructive suggestions throughout the year. I convey my special thanks to **Dr. Sandip Das** ACMU for his active help and guide during my work.

Koushik Bhatacharyya.
(mtc0517)

Diss/07/01/196

Abstract

Digitization of Continuous objects cause loss of information because we find limitation to represent infinite precession of information of actual object. This limitation ultimately results in a many to one map from a group of objects of the real world into a single object of the Digital World. So when we make some quantitative estimation of Original object from its digital version we inevitably invite some possibility of error. By estimating the error of a digital straight line segment and approximating an arbitrary figure by sequence of straight line segment, we ultimately find an approximate estimate of error in general estimation.

Key Words: Digital Geometry, Corridor DSS, Digital Equivalence, Digital Straight line Segments, T-Corridor, W-Corridor, Error Estimation

Diss/07/01/196

Synopsis

"Finding the Corridor of a Straight Line Segment and its Application in Error Estimation in Area Computation"

Koushik Bhattacharyya (MTC 0517)

Under the supervision of

Prof. Bhargab B. Bhattacharya

Motivation:

As the digital world is making its more and more place in several aspect of life, we are creating digital version of real life objects and playing more and more games with them. Digitization of continuous objects causes loss of information because we find limitation to represent infinite precession of information of actual object. This limitation ultimately results in a many to one map from a group of objects of the real world into a single object of the Digital World. So when we make some quantitative estimation of original object from its digital version we inevitably invite some possibility of error.

By estimating the error of a digital straight line segment and approximating an arbitrary figure by sequence of straight line segment, we ultimately find an approximate estimate of error in area computation of a closed figure. The corridor of a Digital Line Segment will give such area where a straight line segment can move without changing its digital representation.

Till date there is no algorithm that finds the corridor of a Digital Straight line segment. This is the initial motivation to find such algorithm with proof of correctness. Here we have presented an algorithm for finding corridor of a digital straight line segment. And the proof of correctness is also stated here. After getting such algorithm next step is to apply this in error estimation in area computation gradually from simple polygon to any closed figure.

Problem definition:

Assuming that a straight line segment is identified by its chain code and fixed boundaries (segment boundaries) of its end points, we define Translation-Corridor and Weak-Corridor which are informally the areas where the original line segment of Euclidean Plane can Translate and Move without changing its chain code and end boundary segments (termed as Left and Right Wall in Literature).

Our problem is:

Given a straight line segment in Euclidean Plane, devise an algorithm to find Translation-Corridor and Weak-Corridor of the Straight line segment.

Next problem, which is addressed after solving above, contains design of some technique or some algorithm that estimates error in area computation of area of the original figure from its digital representation, which is a typical job in Digital Geometry (extracting quantitative information from Digital Image). Initial algorithm is for a simple polygon. Then it is extended to handle any closed figure where no two portions of boundary curve intersect.

Overview of Work:

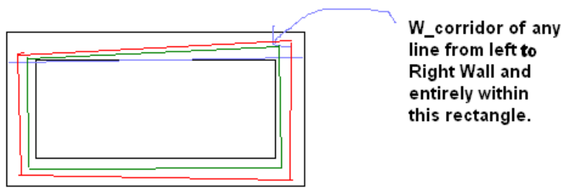
Our contribution can be described in terms of three interrelated tasks. The main task is to give algorithms for finding two types of corridors with proof of correctness. Second task contains development of Digital Straight Line Segment investigation Tool which along with several other important algorithms related to Digital Straight line segments, implements the above algorithm to find Corridors along with Display. Third part of the task is to propose some techniques in the form of algorithm to estimate error in area computation of simple polygon to more complex objects.

We have described point set representation of Straight line segment, their boundary digitization scheme, tunnel (which play a very crucial role in the theoretical analysis) and digital equivalence of two line segments. Our main goal was to show that the set of all line segments which are digitally equivalent to the originally given straight line segment is same as the set of all line segments entirely contained in the Tunnel and sharing Left and Right Wall with the original straight line segment. This fact plays the main role in proving the correctness of Weak-Corridor algorithm. To prove this main theorem we have used the result that each line itself belongs to its own Tunnel.

After proving the uniformity of thickness of Translation Corridor, we have derived the expression for thickness. Also described the fact that a range of slopes in the R^2 plane is mapped to an intermediate slope in the digital plane. That intermediate slope will be a rational number in the case of some typical pattern of chain code. Finally we have described a modified rotation in the algorithm for Weak-Corridor before stating and proving the algorithm. This algorithm actually finds the Weak-Corridor by rotation of a line segment but always keeping it within the tunnel. Whenever it touches the tunnel wall during rotation in a particular direction with respect to a fixed point on the line, called pivot, the touch-point on the wall of Tunnel becomes a boundary point of weak corridor. Collecting all those points we find a polygon that is in fact the weak corridor.

The Digital Straight Line Segment (DSS) investigation tool automates the generation of several information related to DSS like grid points of digitization, chain code, corridor polygon and its (average in the case of non-uniform) width, and area. Ultimately we find the estimation of error using the proposed algorithm described later.

For a simple polygon without hole, each side is digitized and the corridor of each side describes the area where line in Euclidean plane can move. If the actual line is in one end of that area but from digital representation we assume that the line was in the other end of the area, then error contributed by this line in computation of area of the polygon from its digital representation is equal to the area of the Corridor in the case of Translation-Corridor. Except for some cases same is true for Weak-Corridor.



Lastly we approximate any curved boundary by a sequence of approximate straight line segments (ADSS) for which our above theory of corridor is also valid (since all assumption used in our theory was also maintained in ADSS).For each ADSS in the sequence we find corridor and get estimation of desired error.

Conclusion: The algorithms are proved to be true finder of Translation and Weak Corridor which are finally used to find error in area computation and implemented with graphical display option for their applications.

Content

Introduction

1.1 Digital Geometry:

Definition1.1: Subject dealing with GEOMETRY of DISCRETE SET (usually point set) which are considered to be DIGITIZED model / image of object.

Definition1.2: Digital geometry is the study of geometric or topological properties of set of pixel or voxels. It is often an attempt to obtain quantitative information about object by analyzing Digitized pictures (2D or 3D) in which the objects are represented by such set.

1.2 Main Aspect of Study:

Simply put, digitizing is replacing an object by a discrete set of its points. The images we see on the TV screen, the raster display of a computer, or in newspapers are in fact digital images.

Constructing digitized representations of objects, with the emphasis on precision and efficiency (either by means of synthesis, see, for example, Bresenham's line algorithm or digital disks, or by means of digitization and subsequent processing of digital images).

Study of properties of digital sets; see, for example, Pick's theorem, digital convexity, digital straightness, or digital planarity.

Transforming digitized representations of objects, for example (A) into simplified shapes such as (i) skeletons, by repeated removal of simple points such that the digital topology of an image does not change, or (ii) medial axis, by calculating local maxima in a distance transform of the given digitized object representation, or (B) into modified shapes using mathematical morphology.

Reconstructing "real" objects or their properties (area, length, curvature, volume, surface area, and so forth) from digital images.

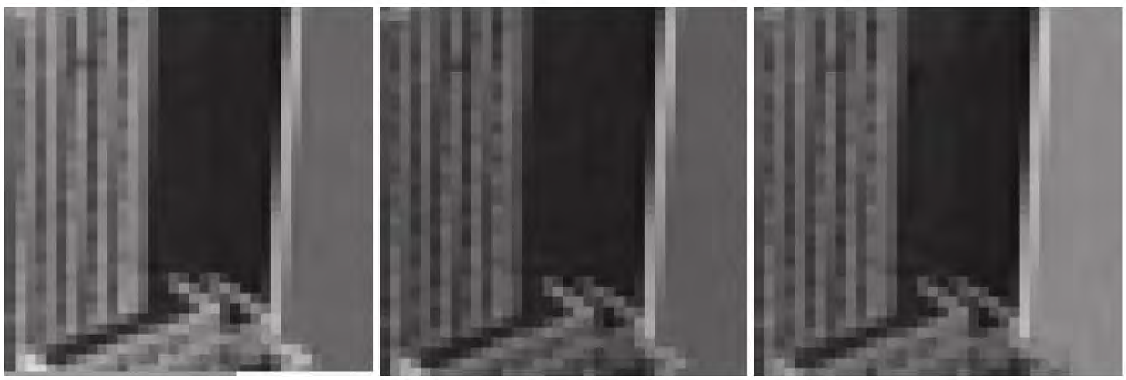
1.3 Applications of Digital Geometry:

Digital Geometry is Finding its application I the area of Computer graphics, Pattern recognition, Image analysis, Medical imaging, Industrial image analysis, Robot vision, and possibly many more in the coming days. For a precise and simple example Component labeling Algorithm has got good application in Image Segmentation.

Grids and Digitization

2.0 : Digital Pictures

A digital color picture can be thought as an array of triples (R,G,B) of integers ranging from 0 to G_{Max}. An RGB picture is composed of three single-valued channels, each of which can be shown as gray-level pictures.



red green blue

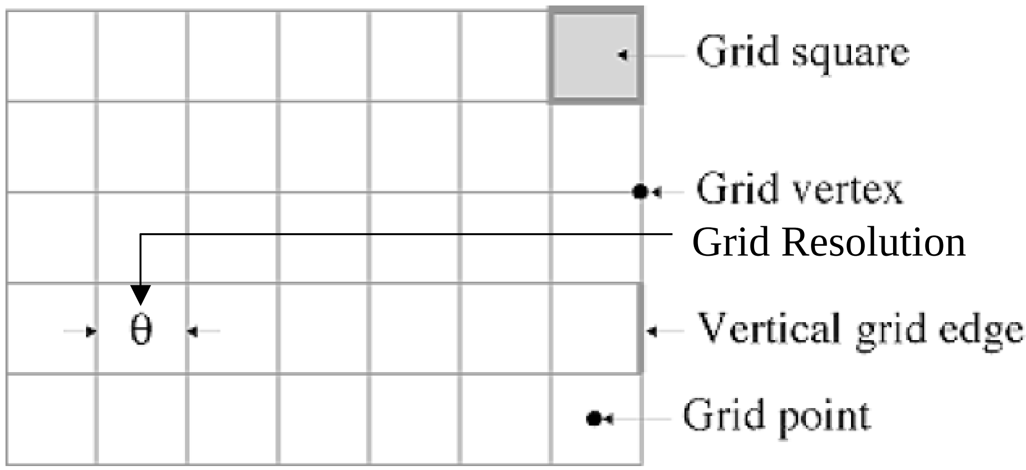
2.1 : Grid

A 2D Grid **G** is $\mathbf{Z}^2$ and in 3D Grid is $\mathbf{Z}^3$**.** All our further discussion will be based on 2D Image unless otherwise stated.

A Grid Vertex is shifted by **(0.5, 0.5)** with respect to Grid Points (elements of $\mathbf{Z}^2$). Precisely Grid Vertices or 0-Cells are $\{(i + 0.5, j + 0.5) \mid (i, j) \in \mathbf{Z}^2\}$**.**

The Line Segment Joining a pair of adjacent Grid Vertices i.e. Grid Vertices whose Euclidian distance is 1 is called a Grid Edge or **1-Cell**.

The Square formed four Grid Edge is called Grid Square or **2-Cell**.



2.2 : Adjacency

Definition 2.2.0 : Two 2-cells, $\mathbf{c_1}$ and $\mathbf{c_2}$, are called **1-adjacent** iff $\mathbf{c_1} \neq \mathbf{c_2}$ and $\mathbf{c_1} \cap \mathbf{c_2}$ is a 1-cell; two grid points $\mathbf{p_1 = (x_1, y_1)}$ and $\mathbf{p_2 = (x_2, y_2)}$ are called **4-adjacent** iff $\mathbf{|x_1 - x_2| + |y_1 - y_2| = 1}$.

Definition 2.2.1 : Two 2-cells, $\mathbf{c_1}$ and $\mathbf{c_2}$, are called **0-adjacent** iff $\mathbf{c_1} \neq \mathbf{c_2}$ and $\mathbf{c_1} \cap \mathbf{c_2}$ contains a 0-cell; two grid points $\mathbf{p_1 = (x_1, y_1)}$ and $\mathbf{p_2 = (x_2, y_2)}$ are called **8-adjacent** iff $\mathbf{\max\{ |x_1 - x_2| , |y_1 - y_2| \} = 1}$.

Definition 2.2.2 : Adjacency sets $\mathbf{A_1(c)}$ and $\mathbf{A_4(p)}$ of one 2-cell **c** or one grid point **p** are respectively the set of all the 2-cells that are **1-adjacent** with **c** and the set of all the grid points that are **4-adjacent** with **p** the corresponding neighborhoods **N1(c)** and **N4(p)**, also containing **c** or **p** itself in addition.

A 2D grid of size m×n is defined by

$$\mathbf{G_{m,n} = \{(i , j) \in Z^2 : 1 \leq i \leq m \text{ and } 1 \leq j \leq n\}} \text{ -- } (1.1)$$

Rectangular set of Grid Squares

$$\mathbf{G_{m,n} = \{ \text{Grid Square } c : (i , j) \text{ is center of } c \text{ and } 1 \leq i \leq m \text{ and } 1 \leq j \leq n\}} \text{- } (1.2)$$

- In the grid point model, a 2D grid **G** is either the infinite grid $\mathbf{Z}^2$ or an m×n rectangular sub array of $\mathbf{Z}^2$; on $\mathbf{G}_{m,n}$. Similarly, a 3D grid is either $\mathbf{Z}^3$ or an l×m×n cuboidal sub array of $\mathbf{Z}^3$.
- In the grid cell model, a 2D grid **G** is either $\mathbf{C}_2$ or an m×n "block" of 2-cells whose union $\cup$ **G** is a rectangular region of the Euclidian plane $\mathbf{E}^2$; on $\mathbf{G}_{m,n}$

in the Equation 1.2. Similarly, a 3D grid is either $\mathbf{C}_3$ or an $l\times m\times n$ set of 3-cells whose union is a cuboid in Euclidian space $\mathbf{E}^3$.

Following proposition is proved in [7.0.0] : The grid defined by $m\times n$ 2-cells and adjacency relation A_1 (A_0) is isomorphic to the grid defined by $m\times n$ grid points and adjacency relation A_4 (A_8). Either of these grids will be denoted by $\mathbf{G}_{m,n}$.

2D Straightness

Basic Definition:-

We consider the grid-intersection digitization of a ray

$$\gamma_{\alpha,\beta} = \{(x, \alpha x + \beta) : 0 \le x < +\infty\}$$

in the set $\mathbf{N}^2 = \{(i,j) : i,j \in \mathbf{N}\}$ of grid points with nonnegative integer coordinates or in the set of 2-cells that have centers in $\mathbf{N}^2$. Because of the symmetry of grid, we can assume that $0 \le \alpha \le 1$.

$\gamma_{\alpha,\beta}$ has a sequence of intersection points $p_0, p_1, p_2, \ldots$ with the vertical grid lines at $n \ge 0$. Let $(n, \boldsymbol{I}_n) \in \mathbf{Z}^2$ be the grid point closest to p_n, and let the following be true:

$$\boldsymbol{I}_{\alpha,\beta} = \{(n, \boldsymbol{I}_n): n \ge 0 \wedge \boldsymbol{I}_n = \lfloor \alpha n + \beta + 0.5 \rfloor\}$$

If there are two closest grid points, we choose the upper one: $\boldsymbol{I}_{\alpha,\beta}$ is the set of centers of a set of grid squares $\mathbf{R}(\gamma_{\alpha,\beta})$. The differences between successive $\boldsymbol{I}_n$ s define the following *chain* codes:

$$\mathbf{i}_{\alpha,\beta}(\mathbf{n}) = \boldsymbol{I}_{n+1} - \boldsymbol{I}_n = \begin{cases} 0 & \text{if } \boldsymbol{I}_n = \boldsymbol{I}_{n+1} \\ 1 & \text{if } \boldsymbol{I}_n = \boldsymbol{I}_{n+1} - 1 \\ & \text{for } n \ge 0 \end{cases}$$

In accordance with our assumption that $0 \le \alpha \le 1$, we need to use only the codes 0 and 1. We recall that code 0 is a horizontal increment and code 1 is a diagonal increment; in the following figure…

Figure

Definition 3.0: $\mathbf{i}_{\alpha,\beta} = \mathbf{i}_{\alpha,\beta}(0), \mathbf{i}_{\alpha,\beta}(1), \mathbf{i}_{\alpha,\beta}(2)$……. is a *digital ray* (in the grid point model) with slope α and intercept β.

This definition can easily be adapted to handle straight lines instead of rays. The code sequence of a *digital straight line* (DSL) is infinite in both direction.

Definition 3.1: Digital rays are infinite words over **{0,1}**. We recall a few basic definitions from the theory of words. A (finite) word defined on (or "over") an alphabet **A** is a finite sequence of elements of A. The length **|u|** of the word $\mathbf{u} = \mathbf{a_1}, \mathbf{a_2}, \mathbf{a_3}, \ldots\ldots, \mathbf{a_n}$ **(where each** $\mathbf{a_i} \in \mathbf{A}$**)** is the number n of letters a_i in u. The empty word **ε**

has length zero. The set of all words defined on **A** is denoted by $\mathbf{A}^*$. A word **v** is a factor of a word **u** iff there exist words $\mathbf{v_1}$ and $\mathbf{v_2}$ such that $\mathbf{u = v_1 v v_2}$. **v** is a subword of u iff v = $a_1, a_2, a_3,\ldots\ldots\ldots,a_n$ and there exist words $\mathbf{v_0, v_1, v_2,\ldots\ldots\ldots,v_n}$ such that $\mathbf{u = v_0 a1 v_1 a_2 v_2 a_3 \ldots\ldots\ldots a_n v_n}$.

Let $\mathbf{X} \subset \mathbf{A}^*$. The set of all *infinite words* $\mathbf{w = u_0 u_1 u_2\ldots\ldots}$ **(where each** $\mathbf{u_i} \in \mathbf{X} - \{\varepsilon\}$**)** is denoted by $\mathbf{X}^w$. If all of the $\mathbf{u_i}$**s** are equal, for example to **v**, we write $\mathbf{w= v}^w$. For all $\mathbf{v} \in \mathbf{A}^*$ and $\mathbf{w} \in \mathbf{A}^w$**, v** is a prefix and w a suffix of the concatenation **vw**.

An integer $\mathbf{k \geq 1}$ is a *period* of a word $\mathbf{u = a_1, a_2, a_3,\ldots\ldots\ldots,a_n}$ **if** $\mathbf{a_i = a_{i+k}}$ $\mathbf{(i = 1, 2,\ldots., n - k)}$. The smallest period of u is called the period of **u**. An infinite word $\mathbf{w} \in \mathbf{A}^w$ is called periodic if it is of the form $\mathbf{w = v}^w$ for some nonempty word $\mathbf{v} \in \mathbf{A}^*$. A word $\mathbf{w} \in \mathbf{A}^w$ is eventually periodic if it is of the form $\mathbf{w = uv}^w$ for some $\mathbf{u} \in \mathbf{A}^*$ and some nonempty $\mathbf{v} \in \mathbf{A}^*$. A word $\mathbf{w} \in \mathbf{A}^w$ is called is *aperiodic* if it is not eventually periodic.

The digitization of a ray $\gamma_{\alpha,\beta}$ in the grid point model is periodic if α is rational and aperiodic if it is irrational.

We state the following theorem without proof.

Theorem 3.0 : Rational digital rays are periodic, and irrational digital rays are aperiodic.

Definition 3.2: A digital straight line segment (DSS for short) is a nonempty factor of a digital ray.

A DSS **u** connects two points $\mathbf{p = (m_p, n_p)}$ and q = $\mathbf{(m_q, n_q)}$ of $\mathbf{N}^2$ $\mathbf{(m_p < m_q)}$ iff the geometric interpretation of $\mathbf{u = u(1)\ldots\ldots.u(m_p - m_q + 1)}$ defines a sequence of horizontal and diagonal steps from **p** to **q**. Let $\mathbf{u = u(1)u(2)\ldots\ldots\ldots u(n)}$ be an 8-arc of length **n**, and let $\mathbf{G}$(u) = $\mathbf{\{p_0, p_1, \ldots\ldots\ldots, p_{n-1}\}}$ be the *assigned* set of grid points such that $\mathbf{p_0 = (0,0)}$ and **u** connects $\mathbf{p_0}$ with $\mathbf{p_{n-1}}$ via a sequence of horizontal and diagonal steps through $\mathbf{p_1, \ldots., p_{n-2}}$.

We again state a Theorem without proof.

Theorem 3.1 : A word $\mathbf{u} \in \mathbf{\{0, 1\}}^*$ is a DSS iff **G**(u) lies between or on two parallel lines with a distance apart (in the y direction) that is less than 1.

Corridor of DSS

4.0 Point Set Representation of Straight Line Segment

Definition 4.0.0: Straight Line Segment **l** with end points $P(x_1,y_1)$ and $Q(x_2,y_2)$ is set of points in $\mathbf{R}^2$ plane defined by $\mathbf{l} = \{(x,y) \mid x = \alpha.x_1 + (1-\alpha).x_2$ and $y = \alpha.y_1 + (1-\alpha).y_2$ and $\alpha \in [0,1]\}$.

Definition 4.0.1 : A Straight Line Segment $\mathbf{l_1}$ is said to be obtained by translation of Straight Line Segment **l** iff there exist a fixed (h,k) dependent on $\mathbf{l_1}$ s.t $(x_1,y_1) \in \mathbf{l_1}$ iff $\exists$ $(x,y) \in \mathbf{l}$ satisfying

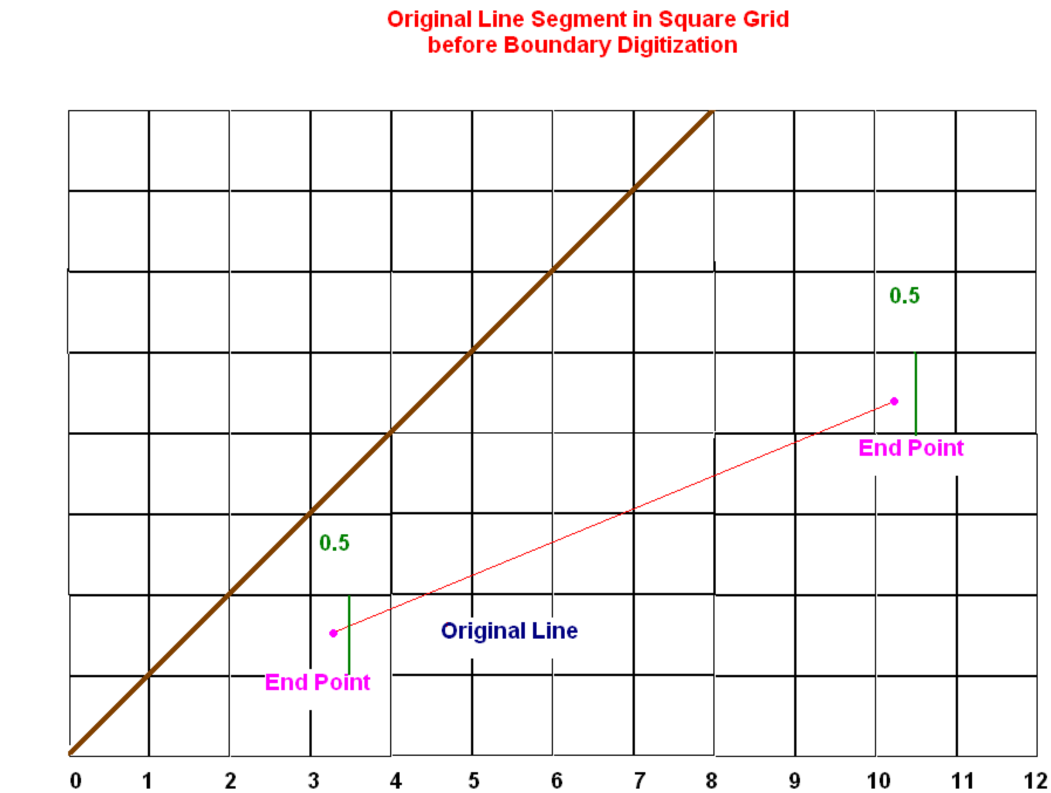
$$x_1 = x + h$$
$$y_1 = y + k$$

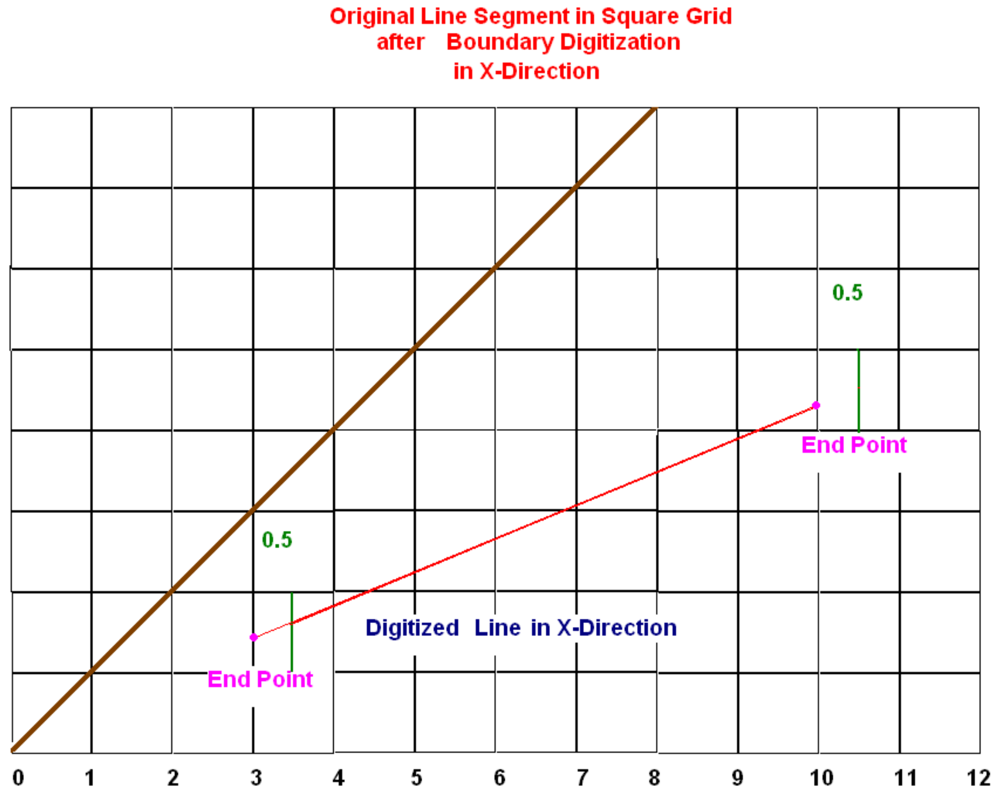
Definition 4.0.2 : Translational Area of Straight Line Segment **l** is denoted by **T(l)** and defined by

$$\mathbf{T(l)} = \bigcup_{\mathbf{l_1} \text{ is obtained by translation of } \mathbf{l}} \mathbf{l_1}$$

4.1 Boundary Digitization

Definition 4.1 : Let **l** be a line segment in $\mathbf{R}^2$ has two end points at $\mathbf{P(x_1 ,y_1), Q(x_2 ,y_2)}$, where $\mathbf{x_1 ,y_1, x_2 ,y_2}$ are Real Numbers. Digitization of **l** in X-Direction is another St Line Segment say $\mathbf{l_1}$ which is defined by the end points $\mathbf{P(h_1 ,y_1), Q(h_2 ,y_2)}$ where h_i is the Rounded value of $\mathbf{x_i}$ **(i=1,2)**.Similar definition can be given for Digitization of **l** in Y-Direction and also for both Direction.


Original Line Segment in Square Grid
before Boundary Digitization
0.5
0.5
End Point
End Point
Original Line
0
1
2
3
4
5
6
7
8
9
10
11
12



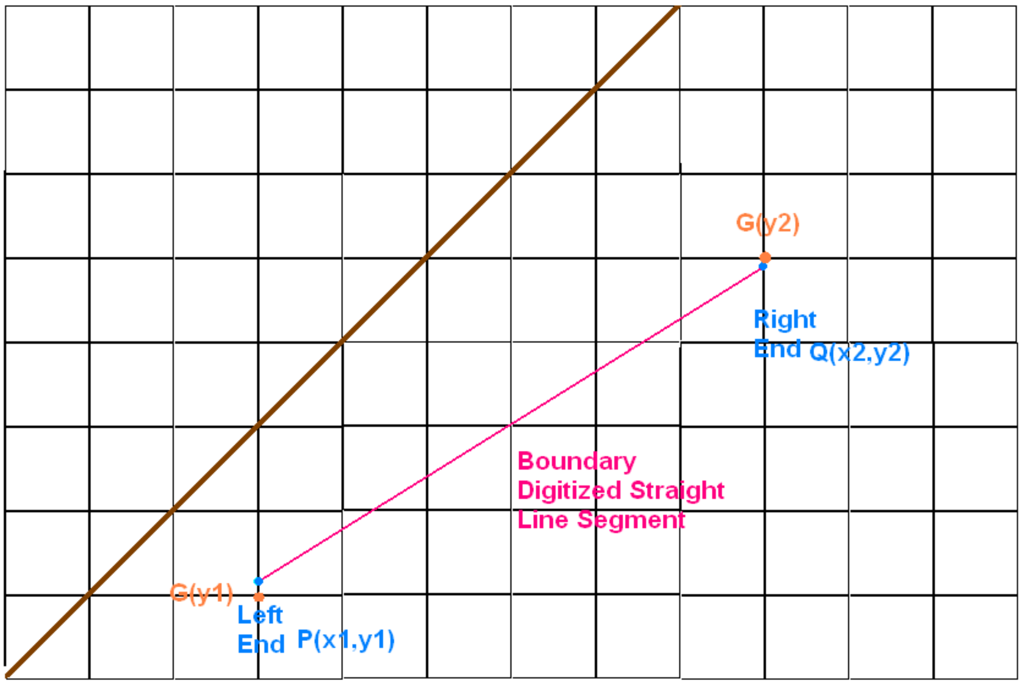
4.2 <u>Tunnel of a Digital Straight Line Segment</u>

Unless otherwise mentioned we assume that all the straight line segments discussed in the following discussion are Digitized in X-Direction and we would work with that digitized version. So all straight line segment appearing in our coming discussion are having integral X-Coordinates at both end points.

Definition 4.2.1 : Let **l** be a straight line segment whose slope **s** lies in the closed interval **[0,1]**.Then the end point with less value of X-Coordinate is called *Left End* and the end point with Greater value of X-Coordinate is called *Right End* of the straight line segment. For lines **l** with slope **s** lies in the open interval **(1,∞) U (-∞, -1),** *Left End* and *Right End* are respectively the end points of **l** with lesser and higher Y-Coordinates.
Finally, For lines **l** with slope **s** lies in the closed interval **[0,1]** *Left End* and *Right End* are respectively the end points of **l** with higher and lesser value of X-Coordinate.(Opposite to their physical locations).

Definition 4.2.2 : Let ***fr***(*y*) = Fractional Part of **y = [y] +1 –y**,(where **[y]** is the box function denoting the greatest integer less than or equal to y).Then, given a point **P(x,y)**, ***G***(*x*) and ***G***(*y*) denotes the abscissa and ordinate respectively of the grid point which represents the point P(x,y) in grid point i.e

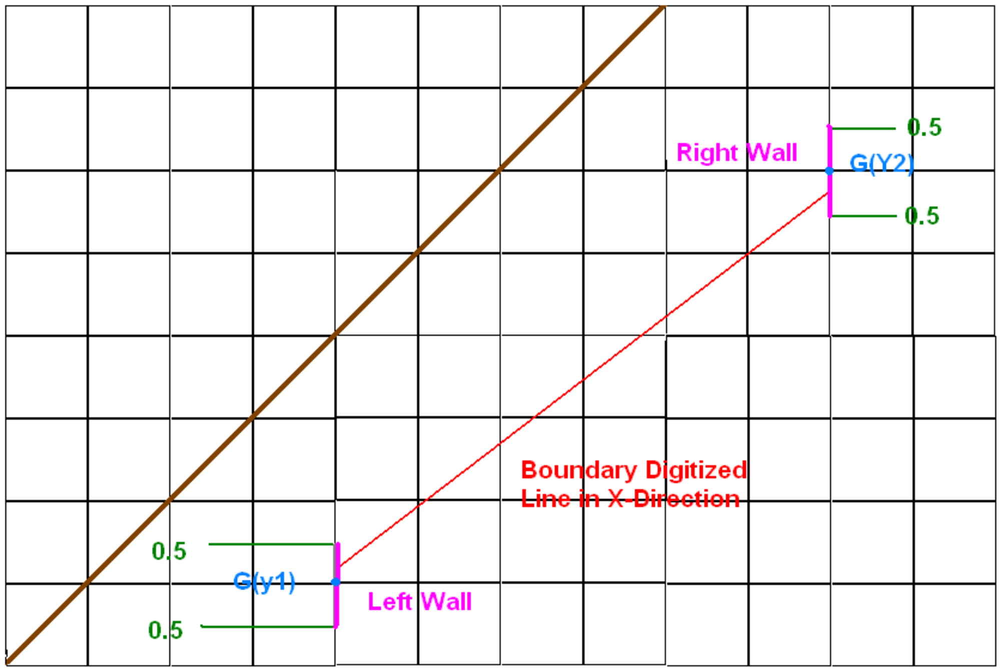
$$G(y) = [y] \quad \text{iff} \quad fr(y) \in [0, 0.5)$$
$$= [y] + 1 \quad \text{iff} \quad fr(y) \in [0.5, 1)$$



Definition 4.2.3 : Given a straight line segment **l** whose slope **s** lies in [0,1] U [-1,0], let P(x,y) be its *left end (i.e x ∈* **Z***)*.Then *Left Wall* of **l** is the straight line segment defined by end points **LW1**(x, **G**(y) – 0.5) and **LW2**(x, **G**(y) + 0.5).

For the case of straight line segment **l** whose slope **s** ε (1,∞) U (-∞, -1) *(i.e* y *∈* **Z***)*, the *Left Wall* of **l** is the straight line segment defined by end points **LW1**(**G**(x) - 0.5, y) and **LW2**(**G**(x) + 0.5, y).

Definition 4.2.4 : Given a straight line segment **l** whose slope **s** lies in [0,1] U [-1,0], let Q(x,y) be its *right end (i.e x* ε **Z***)*.Then Right *Wall* of **l** is the straight line segment defined by end points **RW1**(x, **G**(y) – 0.5) and **RW2**(x, **G**(y) + 0.5).
For the case of straight line segment **l** whose slope **s** ε (1,∞) U (-∞, -1) *(i.e* y *ε* **Z***)*, the *Right Wall* of **l** is the straight line segment defined by end points **RW1**(**G**(x) - 0.5, y) and **RW2**(**G**(x) + 0.5, y).



Definition 4.2.5 : Let S be the set of all Straight Line Segment in the plane R^2. Let $\boldsymbol{\rho} \subseteq (S \times S)$ be a binary relation on S defined by :

For all $l_1, l_2 \in S$, $l_1 \rho l_2$ iff all three of the following holds :

(i) l_1, l_2 have same Left Wall

(ii) l_1, l_2 have same Right Wall

(iii) l_1, l_2 have same Chain Code

All proofs and discussions in the following text are based on the straight line segments with slope **s** $\in$ **[0,1]** unless otherwise mentioned**.** All the ideas can be readily extended to the case of all real slopes **s** $\notin$ **[0,1]** due to symmetry of axes**.**

Lemma 4.2.1 : ρ defined above(in Definition 4.2.5) is Equivalence Relation.

Proof : $\forall\, l_1, l_2, l_3 \in$ **S**, We observe the following,

(a) Reflexivity: $l_1 \rho l_1$ since (i), (ii), (iii) are readily satisfied

(b) Symmetricity : $\mathbf{l_1 \rho l_2} \Rightarrow \mathbf{l_1}, l_2$ have same Left Wall $\Rightarrow l_2, l_1$ have same Left Wall and similarly other two conditions are satisfied .

$$\therefore l_1 \rho l_2 \Rightarrow l_2 \rho l_1$$

(c) Transitivity : $l_1 \rho l_2$ And $l_2 \rho l_3 \Rightarrow l_1, l_2$ have same Left Wall and l_2, l_3 have same Left Wall $\Rightarrow l_1, l_3$ have same Left Wall , other two conditions are satisfied similarly.

$$\therefore l_1 \rho l_2 \text{ And } l_2 \rho l_3 \Rightarrow l_1 \rho l_3$$

Since, ρ is Reflexive, Symmetric and Transitive, hence from the definition of Equivalence Relation, ρ is Equivalence Relation. We call this relation as *Digital*

Equivalence. So two straight line segments $\mathbf{l_1}, \mathbf{l_2} \in \mathbf{S}$ are *Digitally Equivalent* iff $\mathbf{l_1 \,\rho\, l_2}$

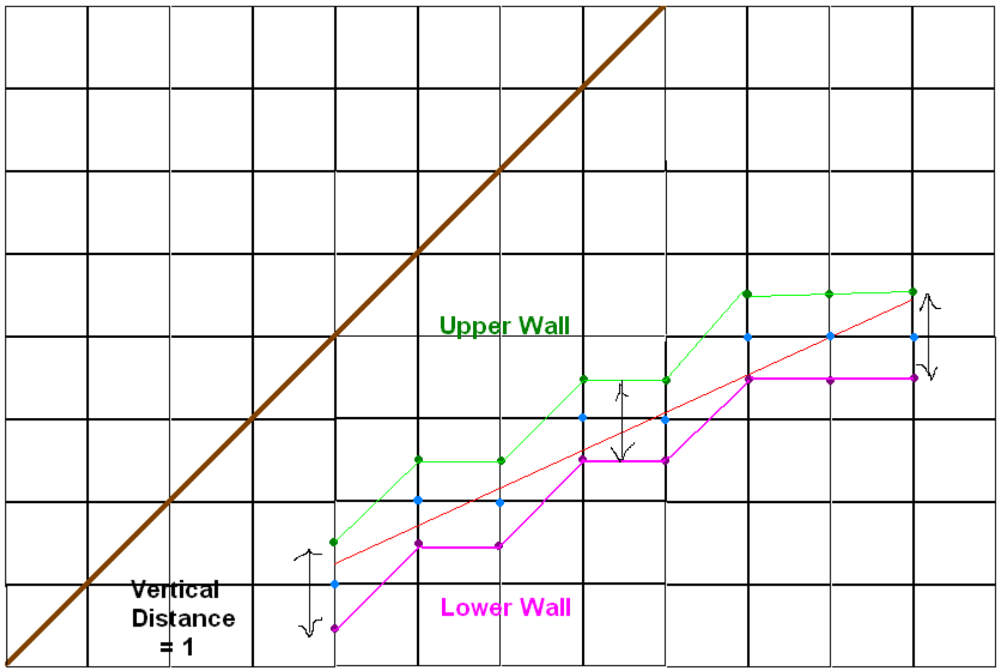
Definition 4.2.6 : Given a straight line segments $\mathbf{l_1} \in \mathbf{S}$, we define *Equivalence class* $\boldsymbol{C}(l_1)$ Of $\mathbf{l_1}$ by $\mathbf{C(l_1) = \{l \in S \mid l_1 \,\rho\, l\}}$.

Since Equivalence relation determines partition on original set **S** so by **ρ**, **S** is partitioned into subsets of straight line segment.

Definition 4.2.7 : Given a straight line segment $\mathbf{l}$ $(\mathbf{y = s.x} + \beta)$whose slope **s** lies in [0,1] with Left and Right ends at $\mathbf{P(x_1,y_1)}$ and $\mathbf{Q(x_2,y_2)}$ respectively. Upper Wall of **l** is the line obtained by sequentially joining the points each with the next $\mathbf{P(x_1,G(y_1) + 0.5)}$, $\mathbf{P1(x_1 + 1, G(y_1 + s) + 0.5)}$, $\mathbf{P2(x_1 + 2, G(y_1 + 2.s) + 0.5)}$, $\mathbf{P1(x_1 + 3, G(y_1 + 3.s) + 0.5)}$, **…,** $\mathbf{Q(x_2,G(y_2) + 0.5)}$.
Alternatively, if $\{(\mathbf{n, I_n}) \mid \mathbf{n = x_1 \text{ to } x_2 \text{ and } I_n} = \lfloor \mathbf{s.n} + \boldsymbol{\beta} + \mathbf{0.5} \rfloor\}$ be the sequence of point which are closest straight line segment **l** then Upper Wall of **l** is given by the sequence $\{(\mathbf{n, I_n + 0.5})\}$.

Definition 4.2.7 : Given a straight line segment $\mathbf{l}$ $(\mathbf{y = s.x} + \beta)$whose slope **s** lies in **[0,1]** with Left and Right ends at $\mathbf{P(x_1,y_1)}$ and $\mathbf{Q(x_2,y_2)}$ respectively. Lower Wall of **l** is the line obtained by sequentially joining the points each with the next by a straight line segment $\mathbf{P(x_1,G(y_1) - 0.5)}$, $\mathbf{P1(x_1 + 1, G(y_1 + s) - 0.5)}$, $\mathbf{P2(x_1 + 2, G(y_1 + 2.s) - 0.5)}$, $\mathbf{P1(x_1 + 3, G(y_1 + 3.s) - 0.5)}$, **…,** $\mathbf{Q(x_2,G(y_2) - 0.5)}$.

Alternatively, if $\{(\mathbf{n, I_n}) \mid \mathbf{n = x_1 \text{ to } x_2 \text{ and } I_n} = \lfloor \mathbf{s.n} + \boldsymbol{\beta} + \mathbf{0.5} \rfloor\}$ be the sequence of point which are closest straight line segment **l** then Upper Wall of **l** is given by the sequence $\{(\mathbf{n, I_n - 0.5})\}$.



Lemma 4.2.2 : The vertical separation between Upper and Lower Wall is always constant and equal to 1.
Proof : At the points of integral abscissa the proof is trivial since vertical separation Between the points $(\mathbf{n}, \mathbf{I_n + 0.5})$ and $(\mathbf{n}, \mathbf{I_n - 0.5})$ is always $\mathbf{I_n + 0.5 - (I_n - 0.5) = 1}\ \forall\ \mathbf{n = x_1}$ **to** $\mathbf{x_2}$.

For other points, let us take nay point with non integral abscissa, say **P(h,k)**on the Upper Wall. Let **P** be within vertical lines $\mathbf{x = x_i}$ **and** $\mathbf{x = x_{i+1}}$.
Then $\exists\ \alpha \in (0,1)$ **s.t**

$$h = \alpha\ .x_i + (1-\alpha).\ x_{i+1}$$

since **P(h,k)** lies inside the segment joining $(\mathbf{x_i, I_{xi} + 0.5})$ and $(\mathbf{x_{i+1}, I_{xi+1} + 0.5})$, hence

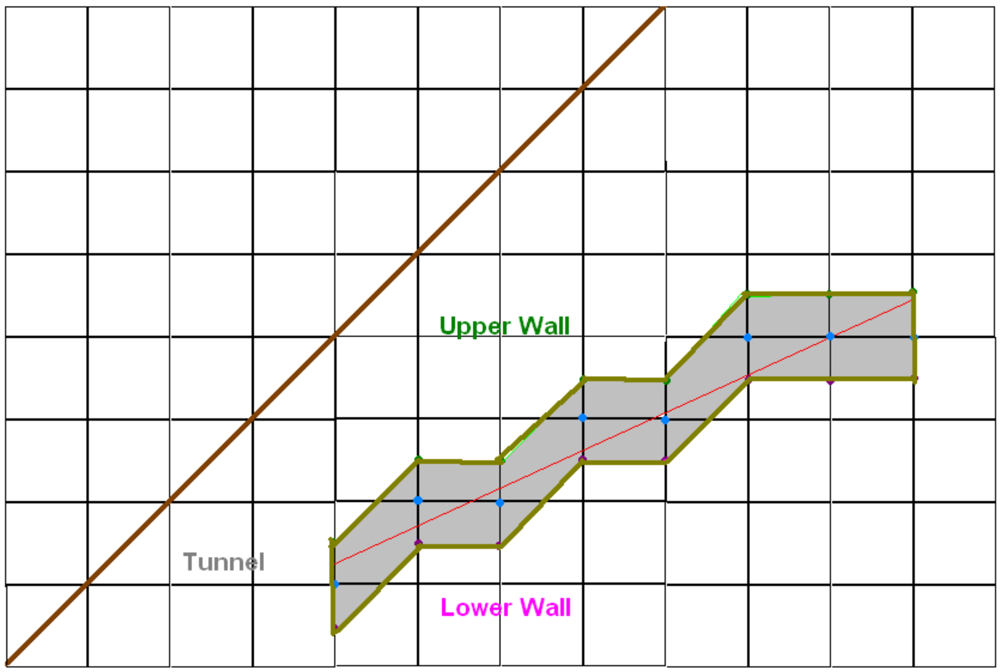
$$k = \alpha\ .(Ix_i + 0.5) + (1-\alpha).(I\ x_{i+1} + 0.5)$$

Now if $\mathbf{Q(h,k_1)}$ be the corresponding point on the Lower Wall with same abscissa then $\mathbf{Q(h,k_1)}$ lies inside the segment joining $(\mathbf{x_i, I_{xi} - 0.5})$ and $(\mathbf{x_{i+1}, I_{xi+1} - 0.5})$, hence

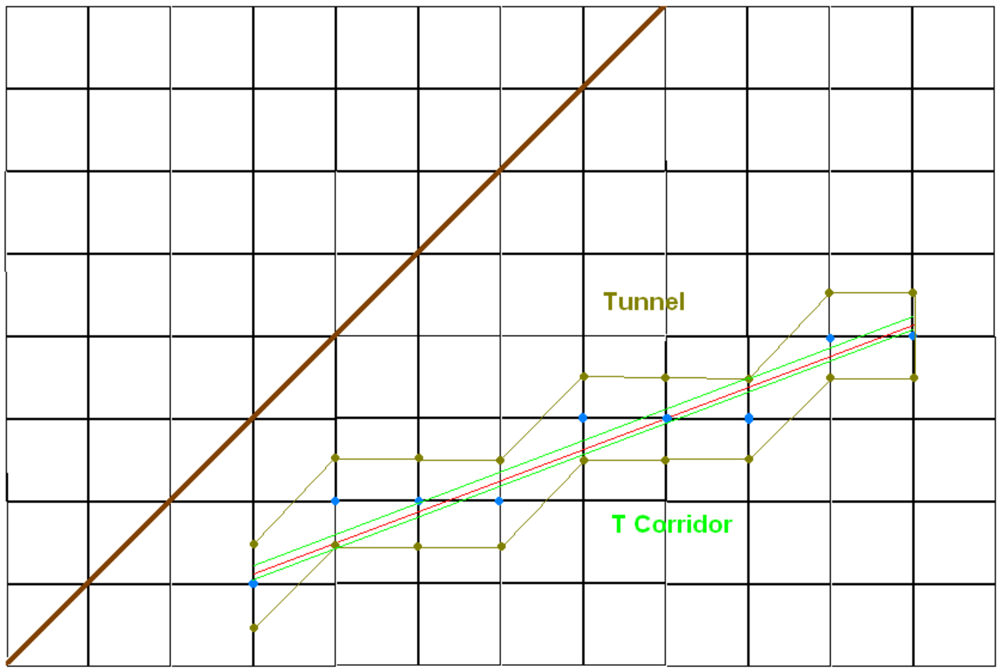
$$k_1 = \alpha\ .(Ix_i - 0.5) + (1-\alpha).(I\ x_{i+1} - 0.5)$$

$$\therefore\ k - k_1 = \alpha.1 + (1-\alpha).1 = 1.$$ Hence Provrd.

Definition 4.2.6 : Tunnel of a straight line segment **l** is the area enclosed by the four wall Viz. Left, Upper, Lower and Right Wall and is denoted by **Tunnel(l)**.



Definition 4.2.7 : T-Corridor of a straight line segment **l** is denoted by **T-Cor(l)** and defined by **T-Cor(l)** = { $\mathbf{l_1 \in S \mid l_1 \in C(l) \cap T(l)}$ }



Lemma 4.2.3 : Let $P_1(x_1,y_1)$ and **$P_2(x_2,y_2)$** respectively be the Left and Right End of a straight line segment **l** and **$P_1'(x_1, y_1')$** and **$P_2'(x_2, y_2')$** be that of **l'**. Then **$y_i \geq y_i'$** **($i = 1,2$) $\Rightarrow k \geq k'$ $\forall$ $(h,k) \in l$** and **$(h', k') \in l'$**.

Proof : Since **$(h,k) \in l$ $\exists$ $\alpha \in [0,1]$ s.t.**

$$h = \alpha. x_1 + (1 - \alpha). x_2$$

$$k = \alpha. y_1 + (1 - \alpha). y_2 \qquad \text{------------------------------------(1),}$$

but in **l'** for the point **(h, k'),** we again have

$$h = \alpha. x_1 + (1 - \alpha). x_2$$

hence

$$k' = \alpha. y_1' + (1 - \alpha). y_2' \qquad \text{------------------------------------(2),}$$

Since, **$\alpha \geq 0$** hence **$\alpha. y_1 \geq \alpha. y_1'$** ------------------------------------**(3),**

Again **$\alpha \in [0,1]$** hence **$(1 - \alpha) \geq 0$,**

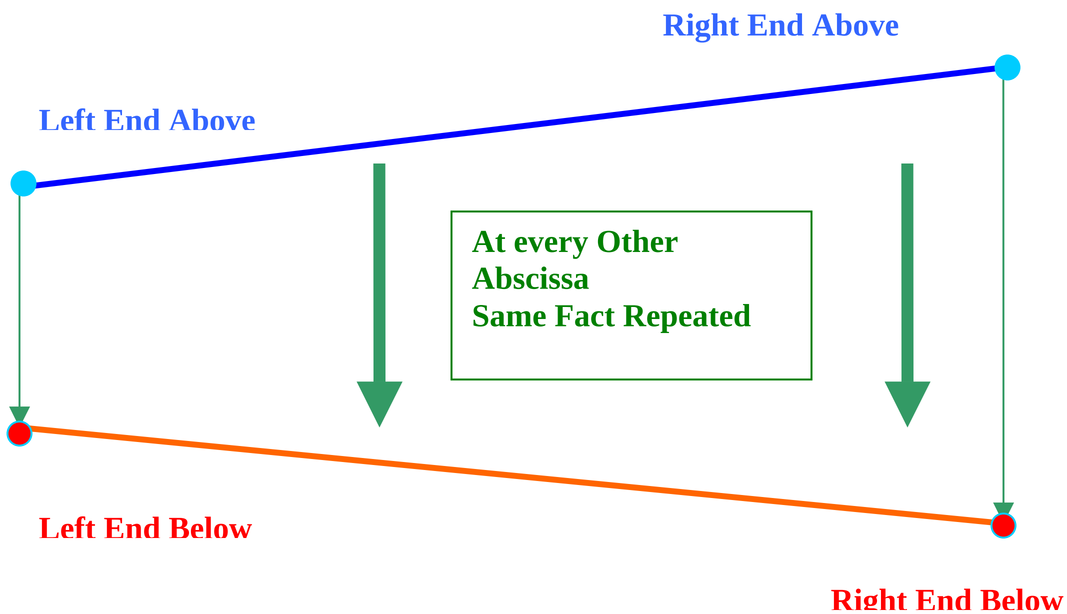
$\therefore$ **$(1 - \alpha). y_2 \geq (1 - \alpha). y_2'$** ------------------------------------**(4),**

Adding **(3) and (4),**

$$\alpha. y_1 + (1 - \alpha). y_2 \geq \alpha. y_1' + (1 - \alpha). y_2'$$

$$\Rightarrow \quad k \geq k'$$

Hence Proved.



Lemma 4.2.4 : **l** is entirely contained within **Tunnel(l)**

Proof : Let $\mathbf{P(x,y)} \in \mathbf{l}$. Let $\mathbf{P_1(x_1,y_1)}$ and $\mathbf{P_2(x_2,y_2)}$ be the respectively the Left and Right End of **l** . **Hence,** $\mathbf{x = (1 - \alpha).\ x_2 + \alpha.\ x_1}$ for some $\boldsymbol{\alpha} \in \mathbf{[0,1]}$.

Now, $\mathbf{x = (1 - \alpha).\ x_2 + \alpha.\ x_1 \geq (1 - \alpha).\ x_1 + \alpha.\ x_1}$ **[Since** $\mathbf{x_2 > x_1}$**]**

$$= \mathbf{x_1}.$$

$$\therefore\ \mathbf{x_1 \leq x}$$

Similarly, $\mathbf{x \leq x_2}$.

$$\therefore \mathbf{x} \in \mathbf{[x_1, x_2]}.$$

$\therefore$ **l** is contained within vertical lines $\mathbf{x = x_1}$ **and** $\mathbf{x = x_2}$.

Now it is sufficient to prove that l is below Upper Wall and above Lower Wall.

Case I : In **P(x,y)** , $\mathbf{x} \in \mathbf{Z}$**,** say $\mathbf{x = n}$**.** Using the definition and notation given in the Definition 4.2.7, $\mathbf{G(y) = I_n}$ (where **G(y)** is defined in the Definition 4.2.2). Now from the definition of **G(y)** , $\mathbf{fr(y)} \in \mathbf{[0, 0.5)} \Leftrightarrow \mathbf{G(y) = [y]}$ ------------------------------------(1),

Now $\mathbf{y = [y] + fr(y) = G(y) + fr(y) = I_n + fr(y) < I_n + 0.5}$ **[from (1)].**

Again, $\mathbf{G(y) = [y] + 1} \Leftrightarrow \mathbf{fr(y)} \in \mathbf{[\,0.5, 1)}$ ---------------------------------------(2)

$\mathbf{y = [y] + fr(y) = [y] + 1 + fr(y) - 1 = I_n + fr(y) - 1}$ **[**Since, $\mathbf{G(y) = I_n}$ and from **(2)** **]** $\geq \mathbf{I_n - 0.5}$ **[** from **(2)].**

But Upper and Lower Wall are defined by the sequences $\mathbf{\{(n, I_n + 0.5)\}}$ and $\mathbf{\{(n, I_n - 0.5)\}}$ where $\mathbf{n = x_1}$ **to** $\mathbf{x_2}$.

$\therefore$ for the Case when $\mathbf{x} \in \mathbf{Z}$ part of straight line segment **l** is Within Upper and Lower Wall.

Case II : In **P(x,y)** , $\mathbf{x} \notin \mathbf{Z}$**,** then say $\mathbf{x} \in \mathbf{(x_i, x_i + 1)}$**,** open interval where $\mathbf{x_i, x_i + 1} \in \mathbf{Z}$ and $\mathbf{x_i, x_i + 1} \in \mathbf{[x_1, x_2]}$**.** Then Case I has already proved that part of **l** is within Upper and Lower Wall at the points corresponding to the vertical lines $\mathbf{x = x_i}$ and $\mathbf{x = x_i + 1}$**.**

Now, using Lemma 4.2.4, part of **l** at **x = x** is also below Upper Wall and above Lower Wall.

So, combining the Case I and case II we complete the proof.

Theorem 4.2.0: C(l) is entirely contained in **Tunnel(l)** $\forall$ **l** $\in$ **S**.
Proof : We first Prove that $\mathbf{l_1} \in \mathbf{C(l)} \Rightarrow \mathbf{l_1}$ and **l** have Identical **Tunnel i.e.**

$$\mathbf{Tunnel(l_1) = Tunnel(l).}$$

Let $\mathbf{l_1} \in \mathbf{C(l).}$ Then from the definition of **C(l),** we have $\mathbf{l\ \rho\ l_1}$ hence **l** and $\mathbf{l_1}$ have identical Left and Right Wall ((i) and (ii)).It remains to prove that **l** and $\mathbf{l_1}$ have identical Upper and Lower Wall. Let $\mathbf{\{(n, I_n)\}}$ and $\mathbf{\{(n, I_n')\}}$ be the grid points where **l** and $\mathbf{l_1}$ are digitized(**n** varying from abscissa of Left Wall to that of Right Wall say $\mathbf{x_1}$ and $\mathbf{x_2}$ respectively).
Since, $\mathbf{l\ \rho\ l_1}$ therefore they have same chain code

$$\therefore\quad \mathbf{I_{n+1} - I_n = I_{n+1}' - I_n'} \qquad \forall\ \mathbf{n = x_1, x_1+1, \ldots, x_2 - 1} \text{ ------------}\mathbf{(1)}$$

Due to identical Left Wall,

$$\mathbf{I_{x1} = I_{x1}'}$$

Now successive application of fact **(1)** we sequentially conclude that

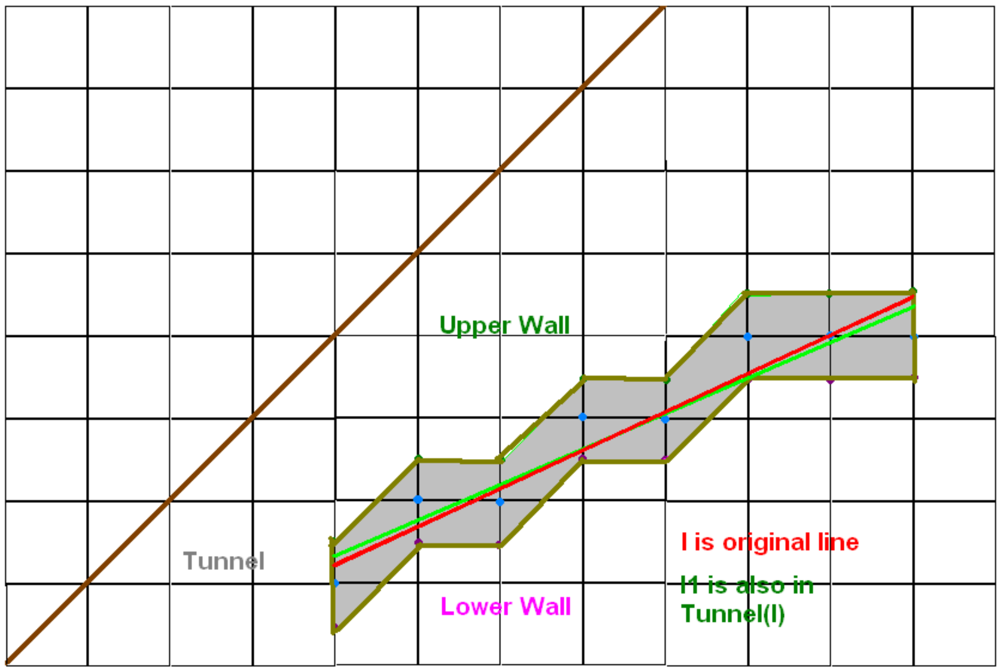
$$\mathbf{I_{x1+1} = I_{x1+1}'}$$
$$\mathbf{I_{x1+2} = I_{x1+2}'}$$
$$\bullet$$
$$\bullet$$
$$\bullet$$
$$\mathbf{I_{x2-1} = I_{x2-1}'}$$
$$\mathbf{I_{x2} = I_{x2}'}$$

$$\therefore\quad \mathbf{I_n = I_n'}\ \forall\ \mathbf{n = x_1, x_1+1, \ldots, x_2}$$

So, Upper Wall of **l** given by $\mathbf{\{(n, I_n + 0.5)\}}$ is same as $\mathbf{\{(n, I_n' + 0.5)\}}$ which is the Upper Wall of $\mathbf{l'}$ and similarly for the Lower Wall.
Hence, $\mathbf{Tunnel(l_1) = Tunnel(l).}$

Now, $\mathbf{l_1}$ is entirely contained in $\mathbf{Tunnel(l_1)}$ from Lemma 4.2.4 and $\mathbf{Tunnel(l_1) =}$ **Tunnel(l)** hence $\mathbf{l_1}$ is entirely contained in **Tunnel(l).**
But $\mathbf{l_1}$ was chosen arbitrarily in **C(l)** hence every line of **C(l)** are entirely in **Tunnel(l).**
Ultimately, **C(l)** is entirely in **Tunnel(l).** (Proved).

.



Corollary 4.2.0: : **T-Cor(l)** is entirely contained in **Tunnel(l)**.
Proof : From Theorem 4.2.0, we have **C(l)** is entirely contained in **Tunnel(l)** $\forall$ **l** $\in$ **S**.
Let $\mathbf{l_1} \in$ **T-Cor(l),** then from Definition 4.2.7**,** $\mathbf{l_1} \in \mathbf{C(l)} \cap \mathbf{T(l)} \Rightarrow \mathbf{l_1} \in \mathbf{C(l)}$ **;** but from Theorem 4.2.0, we have **C(l)** is entirely contained in **Tunnel(l)** $\forall$ **l** $\in$ **S**.
$\therefore$ $\mathbf{l_1}$ is also entirely contained in **Tunnel(l)**.

Theorem 4.2.0: Any straight line segment $\mathbf{l_1} \in \mathbf{S}$ having same Left and Right Wall with **l** and entirely lying within **Tunnel(l)** must be $\in$ **C(l) [i.e** $\mathbf{l_1} \in \mathbf{C(l)}$**].**
Proof : Let $\mathbf{l_1}$ be such a straight line segment. For the shake of contradiction let us assume that $\mathbf{l_1} \notin \mathbf{C(l)}$**.** Left Wall of both **l** and $\mathbf{l_1}$ are same hence $\mathbf{I_{x1}} = \mathbf{I_{x1}}'$ in $\{(\mathbf{n}, \mathbf{I_n})\}$ and $\{(\mathbf{n}, \mathbf{I_n}')\}$**.** Since Left and Right Wall of both **l** and $\mathbf{l_1}$ are identical, so only possibility is that they differ in the chain code (since **l** and $\mathbf{l_1}$ are NOT related by $\boldsymbol{\rho}$).Let the first difference in the chain code occurs at the abscissa $\mathbf{x = x_i}$. Since all the chain code before (if exist) $\mathbf{x = x_i}$ are identical so $\mathbf{I_n = I_n}'$ $\forall$ $\mathbf{n = x_1}$ to $\mathbf{x_i - 1}$ but $\mathbf{I_{xi} \neq I_{xi}}'$**. Since $\mathbf{I_{xi}}$, $\mathbf{I_{xi}}'$ are both integers, so,** $|\mathbf{I_{xi} - I_{xi}}'| \geq \mathbf{1}$**.**Without loss of generality let $\mathbf{I_{xi}' = I_{xi} + 1}$(minimum possible distance is 1)**.** Then $\mathbf{l_1}$ cuts $\mathbf{x = x_i}$ with ordinate value within $[\mathbf{I_{xi}' - 0.5}, \mathbf{I_{xi}' + 0.5}) = [\mathbf{I_{xi} + 1 - 0.5}, \mathbf{I_{xi} + 1 + 0.5}) = [\mathbf{I_{xi} + 0.5}, \mathbf{I_{xi} + 1.5})$ which is above the corresponding Upper Bound of **Tunnel(l)** because upper bound of **Tunnel(l)** at that abscissa is $\mathbf{I_{xi} + 0.5}$**.**

Algorithm to Find T-Corridoor(l) :

Input : Straight Line segment **l** by its end points
Output : **T-Corridor** as an Enclosing Polygon

Step1. Find **Tunnel(l)**

Step2. For Each point **p(i)** in the Upper Wall with integral abscissa
2.1) Find the Vertical Distance of point **p(i)** from line **l**
Step3. Find Minnimum among them say **dUp** say at **p(U)**

Step4. For Each point **q(i)** in the Lower Wall with integral abscissa
2.1) Find the Vertical Distance of point **q(i)** from line **l**
Step5. Find Minimum among them say **dDown** say at **q(D)**

Step6. Find **LU RU**, the points of intersection of the line parallel to **l** but **dUp** distance apart in upward direction from **l (** passing through **p(U))**, to the left and right wall of **l** respectively.

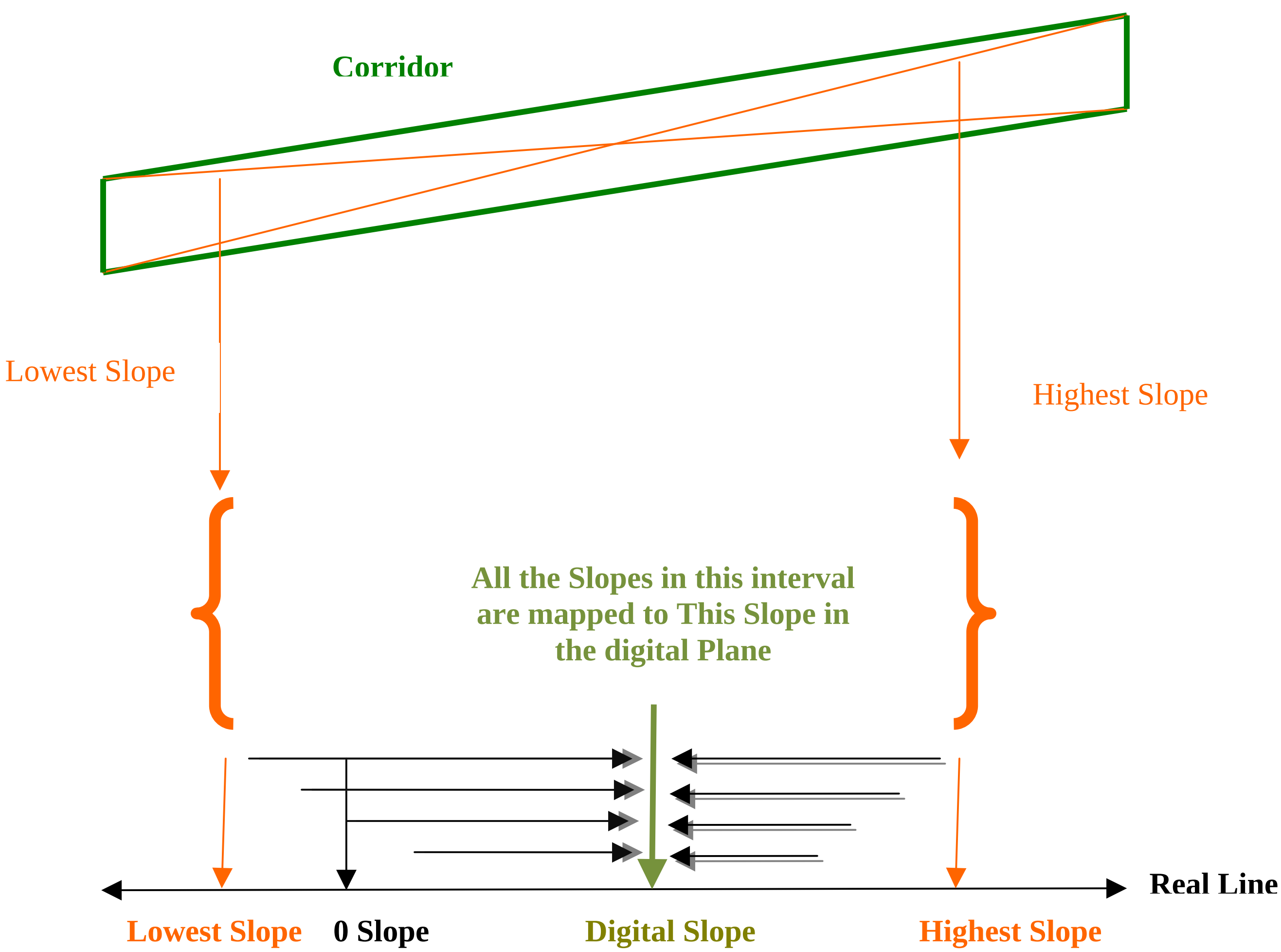
Step7. Find **LL RL**, the points of intersection of the line parallel to **l** but **dDown** distance apart in Downward direction from **l (** passing through **q(D))**to the left and right wall of **l** respectively.

Step8. Then **(LL,LU,RU,RL)** gives the polygon which exatly encloses the **T-Corridoor(l)**. Return **(LL,LU,RU,RL)**.

Step9. End

Theorem 4.2.0: Any straight line segment $\mathbf{l_1} \in \mathbf{S}$ having same Left and Right Wall with **l** and entirely lying within **Tunnel(l)** must be $\in$ **C(l) [i.e** $\mathbf{l_1} \in$ **C(l)].**

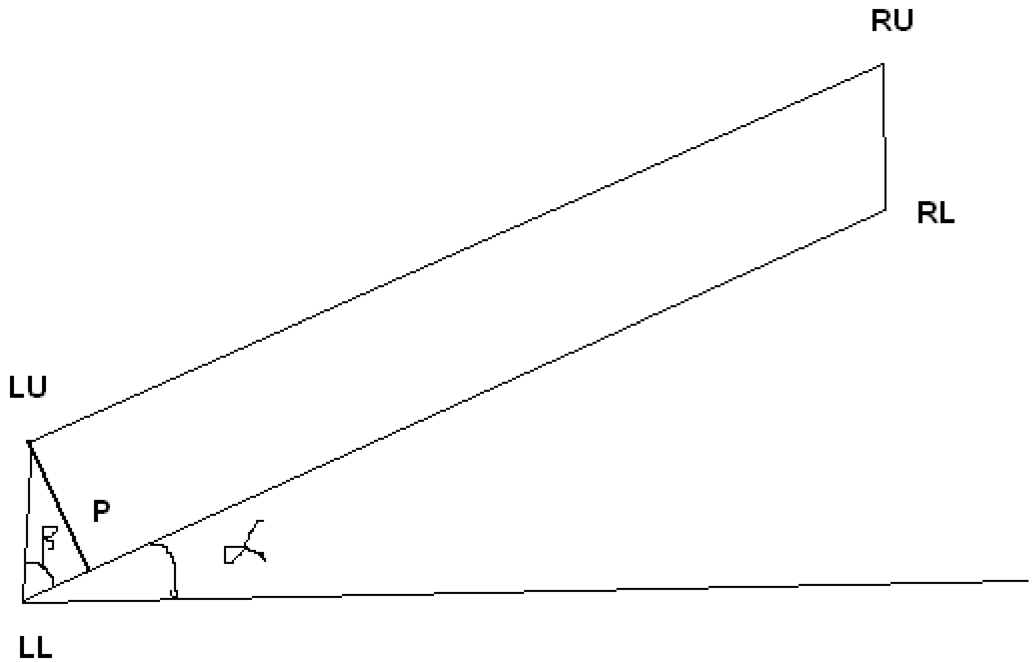
Proof : Let $\mathbf{l_1}$ be such a straight line segment. For the shake of contradiction let us assume that $\mathbf{l_1 \notin C(l)}$. Left Wall of both $\mathbf{l}$ and $\mathbf{l_1}$ are same hence $\mathbf{I_{x1}} = \mathbf{I_{x1}'}$ in $\mathbf{\{(n, I_n)\}}$ and $\mathbf{\{(n, I_n')\}}$. Since Left and Right Wall of both $\mathbf{l}$ and $\mathbf{l_1}$ are identical, so only possibility is that they differ in the chain code (since $\mathbf{l}$ and $\mathbf{l_1}$ are NOT related by $\boldsymbol{\rho}$).Let the first difference in the chain code occurs at the abscissa $\mathbf{x = x_i}$. Since all the chain code before (if exist) $\mathbf{x = x_i}$ are identical so $\mathbf{I_n = I_n'}$ $\forall$ $\mathbf{n = x_1}$ to $\mathbf{x_i - 1}$ but $\mathbf{I_{xi} \neq I_{xi}'}$. **Since** $\mathbf{I_{xi}}$**,** $\mathbf{I_{xi}'}$ **are both integers, so,** $\mathbf{| I_{xi} - I_{xi}' | \geq 1}$**.**Without loss of generality let $\mathbf{I_{xi}' = I_{xi} + 1}$(minimum possible distance is 1)**.** Then $\mathbf{l_1}$ cuts $\mathbf{x = x_i}$ with ordinate value within $[\mathbf{I_{xi}' - 0.5}, \mathbf{I_{xi}' + 0.5}) = [\mathbf{I_{xi} + 1 - 0.5}, \mathbf{I_{xi} + 1 + 0.5}) = [\mathbf{I_{xi} + 0.5}, \mathbf{I_{xi} + 1.5})$ which is above the corresponding Upper Bound of **Tunnel(l)** because upper bound of **Tunnel(l)** at that abscissa is $\mathbf{I_{xi} + 0.5}$**.**



All slopes in the interval Rational/Irrational are mapped to Digital slope. In the case of W-Corridor it is readily clear that Lowest and Highest Slopes both are rational. But for T-Corridor we can realize it considering the segment of infinite length when the initial line have rational slope. If the initial line has irrational slope then digital slope will be irrational since upper and lower boundaries of T-Corridor are both parallel to the original line and their slope is the Digital slope.

Comment on Width T-Corridor :

Algorithm has found **LL, LU, RU, RL**. Each of **LL-RL**, **LU-RU** are parallel to the original input line **l**. Hence **LL-RL || LU-RU.** Again **LL-LU || RL-RU** since they are part of tow vertical lines viz. Left and Right Wall. Therefore **LL, LU, RU, RL** forms a parallelogram.



With reference to above figure slope of original line **s = tan α.** Let **p** be the foot of perpendicular from **LU** on **LL-RL.** Let the angle(**LU LL P**) be **β**. $\alpha + \beta = 90^o$.
From this width of corridor **=** Length of **LU-P** = Length of **LU-LL .Sin β**
= Length of **LU-LL .Cos α =** Length of **LU-LL/** $(1+s^2)^{1/2}$

Proof of Algorithm :
Let $\mathbf{l_1}$ be any line obtained by translation of **l** with translation amount **(h,k)**.Since line has to be within Left and Right Wall so **h = 0.** If **k > 0** and goes beyond the minimum distance found in the step3 i.e. **dUp** then resulting line will cross the boundary point **p(U)** so will be outside the **Tunnel(l)** and from Corollary 4.2.0, is goes outside the **T-Cor(l).** If $\mathbf{k \leq dUp}$ then it is within **T-Cor(l)** provided it is above the Lower Wall of **Tunnel(l).** Similarly reasoning with **dDown** we conclude that $\mathbf{l_1}$ will be in **T-Cor(l)** if $\mathbf{k \in [-dDown , dUp]}$ (noting that **dDown** is a positive number so negative sigh is added before it).

Only if part : Let $\mathbf{l_1}$ be any line obtained by translation of **l** with translation amount **(0,k)** **s.t.** $\mathbf{k \in [-dDown , dUp],}$ then due to parallelism every point of $\mathbf{l_1}$ is below the the every point of **LU-RU** and also above the every point of **LL-RL** each of which are within **Tunnel(l)** hence $\mathbf{l_1}$ is too.
Above two parts leads to the conclusion that $\mathbf{l_1}$ is within **T-Cor(l)** iff $\mathbf{k \in [-dDown , dUp]}$, and from algorithm, iff $\mathbf{l_1}$ is within Polygon **(LL,LU,RU,RL).**
This infers the correctness of the algorithm.
Proof Complete.

4.3 : Weak Corridor :

To solve the problem with T-Corridor we seek for some extended version of corridor that will contain all the straight line segment that has got same digital representation with Original line in the sense that digital version of both are identical i.e. in the sense of $\boldsymbol{\rho}$.

Definition 4.3.0 : *Weak Corridor* of a straight line segment **l** is denoted by **W-Cor(l)** and defined simply by **W-Cor(l) = C(l).**

Before presenting the Algorithm we want to state the meaning of Rotation in this Context, which is a little bit different from common concept of rotation. Whenever we Say that we are rotating a line we do trim or extension of line in either end at each stage of rotation so that the end points have same abscissa with the initial line.

It can be shown that the rotating line discussed below meets Upper and Lower Wall will Always meet the Tunnel at some point(s) having integral abscissa, using the similar reasoning that is given in the proof of Lemma 4.3.9.

We use the notational convention that for a point **P, P.x** and **P.y** gives the abscissa and ordinates of point **P.**

Algorithm 4.3.0

/* To Find Four Corners of W-Cor(l) */

Input : Straight Line segment **l** by its end points
Output : **Left Lower, Left Up, Right Up, Right Lower** end points **W-Corridor** as **LL,LU,RU,RL**.
{we here only finds **LL** and **RU**, other two points **LU** and **RL** can be analogously found with similar steps**}**

Step1. Find **Tunnel(l)**

Step2. Rotate line **l** by pivoting at its left end in anticlockwise direction So that the line go upward towards Upper Wall and away from the Lower Wall, up to the position when it first meet the Upper Wall say at $\mathbf{U_k}$.

Step3. Now Pivot at $\mathbf{U_k}$ and Rotate line **l** from its new position in anticlockwise direction until it meet Lower or Upper Wall of **Tunnel(I).**

Step4. If (Upper Wall is met before Lower Wall) Then
4.1) $\mathbf{U_k}$ ← New meeting point of Upper Wall
4.2) Goto Step3.
End If

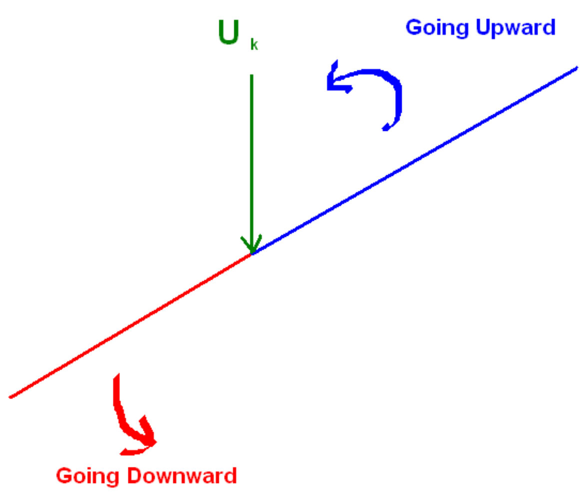
Step5. If (Lower Wall is met before Upper Wall) OR
(Lower Wall and Upper Wall met simultaniously) Then
5.1) Call the Lower Wall meeting point as $\mathbf{L_j}$
5.2) call the furthest (from pivot) Upper Wall meeting Point as $\mathbf{U_k}$.

Step6. Call the Left and Right end points of current position of rotating line as **LL** and **RU**.

Step7. End

Lemma 4.3.0: In the algorithm 4.3.0 $\mathbf{U_k.x} > \mathbf{L_j.x}$ i.e. U_k is at right of L_j

Proof : Consider at any U_k step before current line $\mathbf{l_1}$ touches L_j.Since we pivot at U_k and rotate $\mathbf{l_1}$ anticlockwise, so, the portion of $\mathbf{l_1}$ which is at right of $\mathbf{U_k}$ is going Above i.e. away from the Lower Wall so cannot touch the Lower Wall before the portion of $\mathbf{l_1}$ which is at left of $\mathbf{U_k}$ and is going down towards the Lower Wall. This left portion, Therefore can touch the Lower wall first. Consequently the Point $\mathbf{L_j}$ which is touched by the left portion of $\mathbf{U_k}$ will be at left of $\mathbf{U_k}$.



Lemma 4.3.1 : Line joining **LL** and **RU** will have greatest slope among all straight line segment in **C(l)** .

Proof : From Algorithm1 **LL-RU** passes through U_k and L_j and L_j is at right of U_k. Let l_2 be any other line of **C(l)**. To pass through the Tunnel it must pass through the segments $U_k L_k$ and $U_j L_j$.

Slope of the line $U_k L_j = s_0 = (U_k.y - L_j.y) / (U_k.x - L_j.x)$.
If l_2 intersect $U_k L_k$ and $U_j L_j$ respectively at **P** and **Q** then $\exists$ $\alpha, \beta \in [0,1]$ s.t.

$$P.y = U_k.y - \alpha$$
$$Q.y = L_j.y + \beta.$$

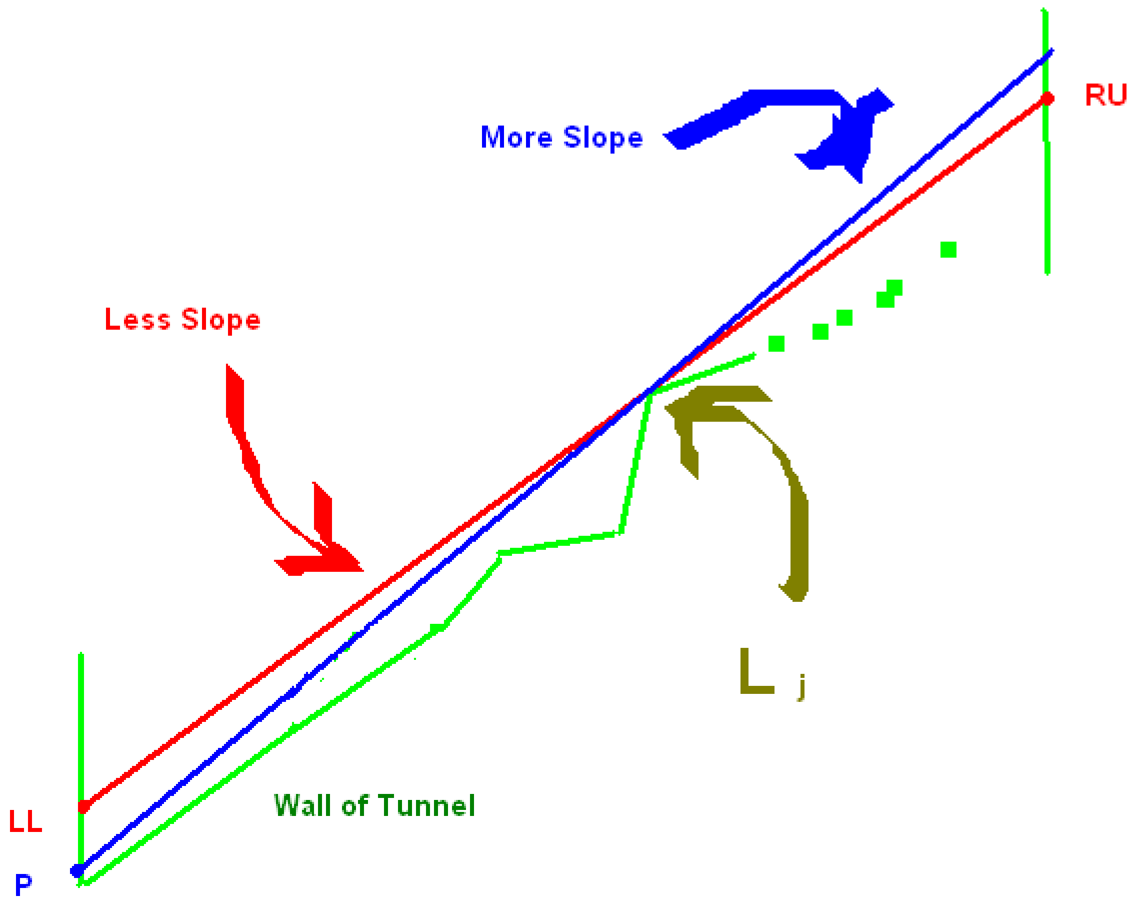
Slope of the line $l_2 = (U_k.y - \alpha - L_j.y - \beta) / (U_k.x - L_j.x)$
$\leq (U_k.y - L_j.y) / (U_k.x - L_j.x)$
$= s_0$

Lemma 4.3.2 : Maximum slope is uniquely attained by the line joining **LL** and **RU.**
Proof : Line passing through **LL** and **RU** also passing through U_k and L_j has Maximum slope. Any other line must not pass through at least one of U_k or L_j , as two points determines a straight line. Then Using the notation of Lemma 4.3.1, at least one of α or β is > 0, which results in strict inequality i.e. Slope of the line $l_2 <$ Slope of the line joining **LL** and **RU.** Hence any other line will have strictly lesser slope and therefore we conclude the required uniqueness.

Lemma 4.3.4 : No point lower than **LL** and on the Left Wall is in **C(l).**
Proof : If **LL** is on the Lower Wall we are done. If not, let **P** be any point on Left Wall below **LL** above or on the Lower Wall. Then,

$$P.y = LL.y - \gamma \text{ for } \gamma > 0$$

Now, slope of **LL-RU** = Slope of $LL - L_j$
$= (L_j.y - LL.y) / (L_j.x - LL.x)$
But slope of $P - L_j = (P.y - L_j.y) / (P.x - L_j.x) = (L_j.y - LL.y + \gamma) / (L_j.x - LL.x)$
which is greater than slope of slope of **LL-RU** due to same denominator but greater numerator. But this is the least possible slope of aline passing through **P**. Hence all lines through **P** that is within the Tunnel will have greater slope than slope of line through **LL** and **RU.** This leads to a contradiction to the Maximality of slope stated in the Lemma 4.3.1.



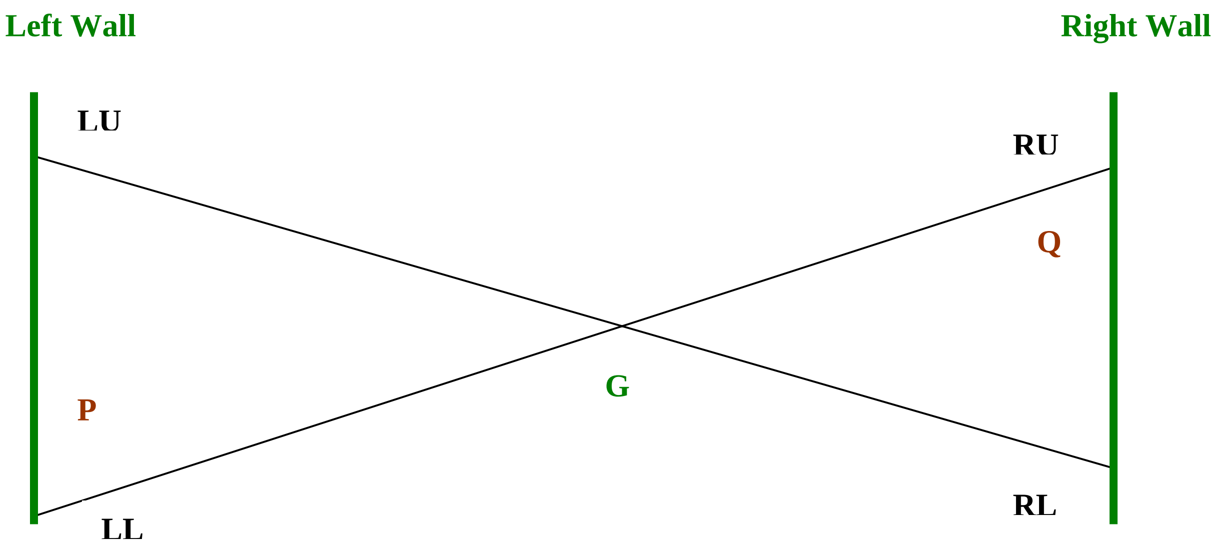
Similarly we can prove the following three Lemmas

Lemma 4.3.5 : No point above **LU** and on the Left Wall is in **C(l).**
Lemma 4.3.6 : No point above **RU** and on the Right Wall is in **C(l).**
Lemma 4.3.6 : No point below **RL** and on the Right Wall is in **C(l).**

Before tracing entire boundary of **C(l)** we establish that we have found boundaries on Left Wall and Right Wall. Following Lemma assures that.

Lemma 4.3.7 : Every Point of **LL-LU** and every Point of **RL-RU** are in **C(l).**
Proof: We only prove for **LL-LU.** Other part is similar.
Let **G** be intersection point of **LL- RU** and **LU –RL.** We can rotate **LL- RU** in clockwise direction pivoting at **G** up to **LU –RL** continuously so that Left End Point of Rotating line continuously moves on Left Wall starting from **LL** to **LU** touching every point of the segment within **LL-LU.** Similarly Right End point of the Rotating line continuously moves on Right Wall starting from **RL** to **RU** touching every point of the segment within **RL-RU.**

Let **P** be any point on the segment **LL-LU.** In some intermediate position of rotation the rotating line touches **P**. Let the other end of rotating line at that time be denoted by **Q**. Now it only remains to prove that **PQ** is entirely within Tunnel .**PG** is enclosed within **LU-G** and **LL-G** each of which are within **Tunnel(l)**. Hence **PG** is within **Tunnel(l)**. Again **GQ** is enclosed within **RU-G** and **RL-G** each of which are within **Tunnel(l)**. Hence **GQ** is within **Tunnel(l)**. Combining we get entire **PQ** is within **Tunnel(l).**



Finally we observe that Algorithm4.0 terminates because only step4 has conditional loop and that condition can be true for finite times since Tunnel length is finite.

Now we have found four points of **C(l).** To find the remaining boundaries of **C(l)** we propose the following Algorithm4.1 and present proof afterwards.

Algorithm 4.1

/* To Find Entire W-Cor(l) */

Input : All inputs and outputs of Algorithm1
Output : **W-Corridor** as an Enclosing Polygon

Initial points (LL,l_j) = WL
1. l = LL – RU
2. Pivot ← l_j
3. Rotate l, with respect to the Pivot clockwise up to maximum extent so that Right part of l from l_j is always above Lower wall & Left part of l from l_j is always below Upper wall and also within boundaries of LL, LU at Left Wall and also RU, RL

at the Right Wall. Let l´ be the final position where l first touch Lower or Upper Wall.

4. If (l´ has touched Lower Wall at l_{j1} before touching Upper Wall) Then
   A) Add the point l_{j1} at the end of Sequence WL.
   B) Pivot ← l_{j1}
   C) Set l ← l´
   D) go to 2.
5. If (l´ has touched Upper Wall before touching Lower Wall) Then
   Only possibility is that it will touch the point LU before any other point of the Upper Wall.
   A) Add the point RL at the end of Sequence WL.
   B) Return WL
6. If (l´ meets Upper and Lower Wall together) Then
   A) Add the point RL at the end of Sequence WL.
   B) Return WL
7. Return WL

Analogously we get **WU** as sequence of points giving the Upper Bound of Weak Corridor and finally properly combining **WL** and **WU** we get a polygn which we denote by **W**.

Proof of Algorithm :

Initially we state and prove few Lemma which will be useful I the proof of the actual algorithm.

Lemma 4.3.8 : No point of **C(l)** can be below the line segment $\mathbf{L_1}$- $\mathbf{L_2}$, where $\mathbf{L_1, L_2}$ are any two points on Lower Wall with integral abscissa and No point of **C(l)** can be above the line segment $\mathbf{U_1}$- $\mathbf{U_2}$, where $\mathbf{U_1, U_2}$ are any two points on Upper Wall with integral abscissa.

Proof: If L_1, L_2 both have same ordinate then $\mathbf{L_1}$- $\mathbf{L_2}$ defines a portion of the Lower Wall. Any point below that is readily outside **Tunnel(l)** hence outside **C(l).**
If $\mathbf{L_1.y \neq L_2.y}$ then take any line $\mathbf{l_2}$ in **C(l).** It must intersect $\mathbf{U_1L_1}$ and $\mathbf{U_2L_2}$ respectively say at **P,Q**.
Since,

$$\mathbf{P.y \geq L_1.y}$$
$$\mathbf{Q.y \geq L_2.y}$$

∴ using Lemma 4.2.3 **PQ** is entirely above $\mathbf{L_1}$- $\mathbf{L_2}$ hence all point of **PQ.** Second part can be proved similarly.

Lemma 4.3.9: In **step5** of the Algorithm 4.1, if l_1 has touched Upper Wall before touching Lower Wall then Only possibility is that it will touch the point LU before any other point of the Upper Wall.
Proof : Let $\mathbf{U_1, U_2, \ldots, U_n}$ be the points on the Upper Wall sequentially from the left to right. Also **LU** be the point on left Wall below or at equal height of $\mathbf{U_1}$.Rotating line is pivoted at $\mathbf{l_j}$ which is at the right of $\mathbf{U_1, U_2, \ldots, U_r}$ (say).Since the rotating line is coming upward from the bottom, among two points with same abscissa $\mathbf{U_1}$ and **LU** it will meet the lower point first. Hence among $\mathbf{U_1}$ and **LU** it will meet **LU** first (since lesser angle has to be rotated).
If $\mathbf{U_i}$ is at left of $\mathbf{U_j}$ then rotating line will meet the point $\mathbf{U_i}$ first since lesser angle has to be rotated to meet be.

So, among all possible meeting point $\mathbf{\{ U_1, U_2, \ldots, U_r \}}$, $\mathbf{U_1}$ is met before all the other and before $\mathbf{U_1}$, **LU** is met.

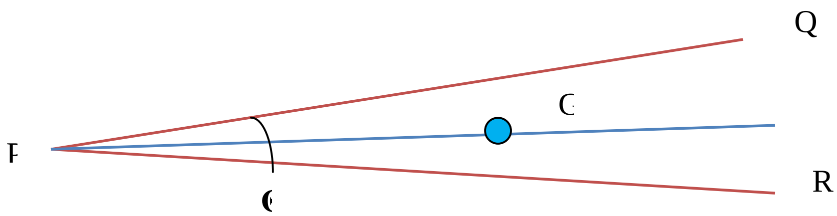
Lemma 4.3.10: Algorithm 4.1 terminates after finite time.
Proof : The Algorithm terminates if the **IF** condition in **Step4** is true for at most finite no. of times because all other steps are having no loop and each execute finite operation that is finished in the finite time.
Since, for any initial pivot ∃ finite no of point on the Lower Wall with integral coefficient (due to finiteness of Tunnel), so rotating line can meet at most finite no of points on Lower Wall in different loop iteration. Hence Proved.

Lemma 4.3.11: When Algorithm 4.1 terminates then **l'** coincides with **LU-RL** .
Proof: There are exactly two steps when the Algorithm 4.1 terminates viz **Step5** and **Step6**.
When Algorithm 4.1 terminates at **Step6** using the same reasoning given in the proof of Lemma 4.3.1 we conclude that **l'** has attained lowest slope and using the Lemma 4.3.2 , we conclude **l'** coincides with **LU-RL.**

When Algorithm 4.1 terminates at **Step5**, then **l'** is now between **LU** and some point viz $\mathbf{L_j}$ of the Lower wall each of which defines Upper and Lower bounds respectively in the corresponding abscissa. Hence using the same reasoning given in the proof of Lemma 4.3.1 we conclude that **l'** has attained lowest slope and using the Lemma 4.3.2 , we conclude **l'** coincides with **LU-RL.**(Proved)

Lemma 4.3.12: Let line segment **PQ** is rotated with pivot at P to some non zero angle and let is final position be **PR.** Let **G** be any point within **PQ** and **PR,** then ∃ an inter mediate position of rotating line that passes through **G.**

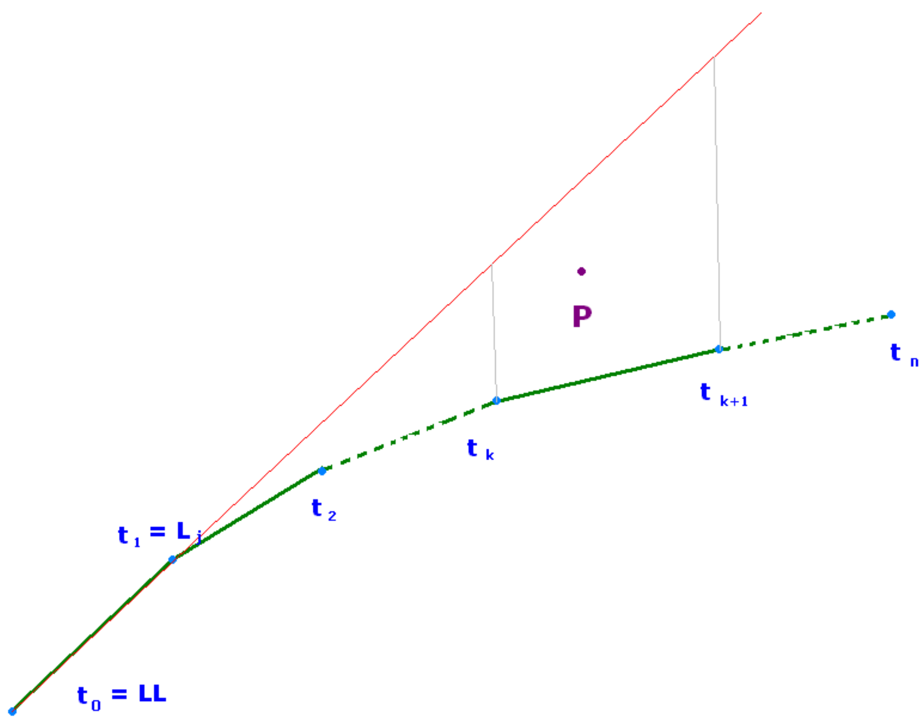
Proof : Join **G,P**. $\angle \mathbf{GPQ}$, $\angle \mathbf{GPR} \neq \mathbf{0}$ (since **G** is intermediate and NOT on any line of **PQ** or **PR**). $\angle \mathbf{GPQ} + \angle \mathbf{GPR} = \angle \mathbf{RPQ} = \mathbf{\Theta}$. Hence, $\mathbf{0} < \angle \mathbf{GPQ} < \mathbf{\Theta}$. Hence when **PQ** start



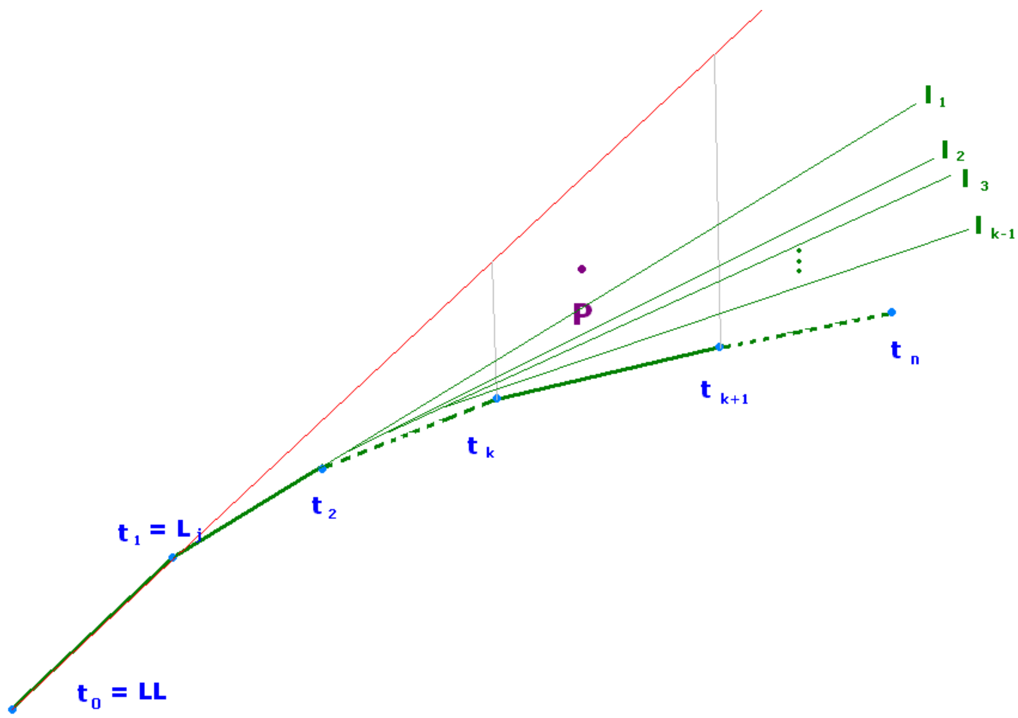
Rotation an rotate up to an angle **Θ** then its intermediate position makes all possible angles with **PQ** ranging from **0** to **Θ.** When rotating line makes ∠**GPQ** angle with **PQ** then its position coincides with **PG.**So ∃ an inter mediate position of rotating line that passes through **G.** (Proved)

Now, due to Lemma 4.3.8, no point of **C(l)** is out side the Polygon **W** obtained as output of Algorithm 4.1.(for the points **LL,LU,RU,RL** same reasoning can be applied).To complete the proof of correctness of the Algorithm 4.1 it remains to show that any point of **W** is also a point of **C(l).**

LL-RU is a diagonal of **W,** so divides **W** into two parts**.** We here presents the proof for the points of **W** that are below the diagonal **LL-RU.** Proof for the other half of **W** will be analogous.



Let us denote the point **LL** by $\mathbf{t_0}$ and other points of **WL** sequentially from the left to right by $\mathbf{t_0, t_1, \ldots, t_n}$ respectively. Let **P** be any point within Polygon **W** that are below the diagonal **LL-RU.** Let $t_k.x \leq P.x < t_{k+1}.x$. The trapezoid **Δ** enclosed by the lines t_k-t_{k+1}, **LL-RU** and two vertical lines through t_k and t_{k+1}, encloses **P**. Let l_i be the position of rotating line when it passes through t_i and t_{i+1}.



Claim : If l_i is within Tunnel then all the points that rotating line segment pass through in traversal from l_i to l_{i+1} **(**inclusive**)** is so.

Proof of Claim : l_i is rotated with pivot at t_{i+1} unless it touch the Tunnel Wall for first time. Since l_i was in the Tunnel so it cannot go outside (by its any part) the Tunnel without touching the Tunnel wall for first time. Since before getting into the position Of l_{i+1} neither Upper nor Lower Wall is touched hence in all intermediate position the rotating line is within Tunnel.
Now at l_{i+1} position rotating line has just touched the Tunnel is has not cross the Tunnel boundaries (Upper and Lower Wall are inclusive). Therefore l_{i+1} is also with the Tunnel. (Claim Proved)

We note that Left and Right End of each l_i are respectively on Left and Right Wall since that invariant is maintained during our rotation. So each l_i are in **C(l)** and all intermediate Position of rotating line while rotating from position l_i to l_{i+1} will also be in **C(l).**
Now the trapezoid **Δ** is divided into several parts by the lines $l_1, l_2, \ldots, l_{k-1}$.When point **P** fall on any of the lines $l_0, l_1, l_2, \ldots, l_k$ then above claim establishes that **P** is in **C(l).** When point **P** fall between any of the lines say l_f and l_{f+1}, then Lemma 4.3.12 establish that ∃ an intermediate position of rotating line when rotating from l_f to l_{f+1}that passes through

P. At that position, rotating line again entirely within Tunnel and ends at Left and Right Wall implying rotating line at that position is within **C(l).**

This completes the proof of correctness of the algorithm.

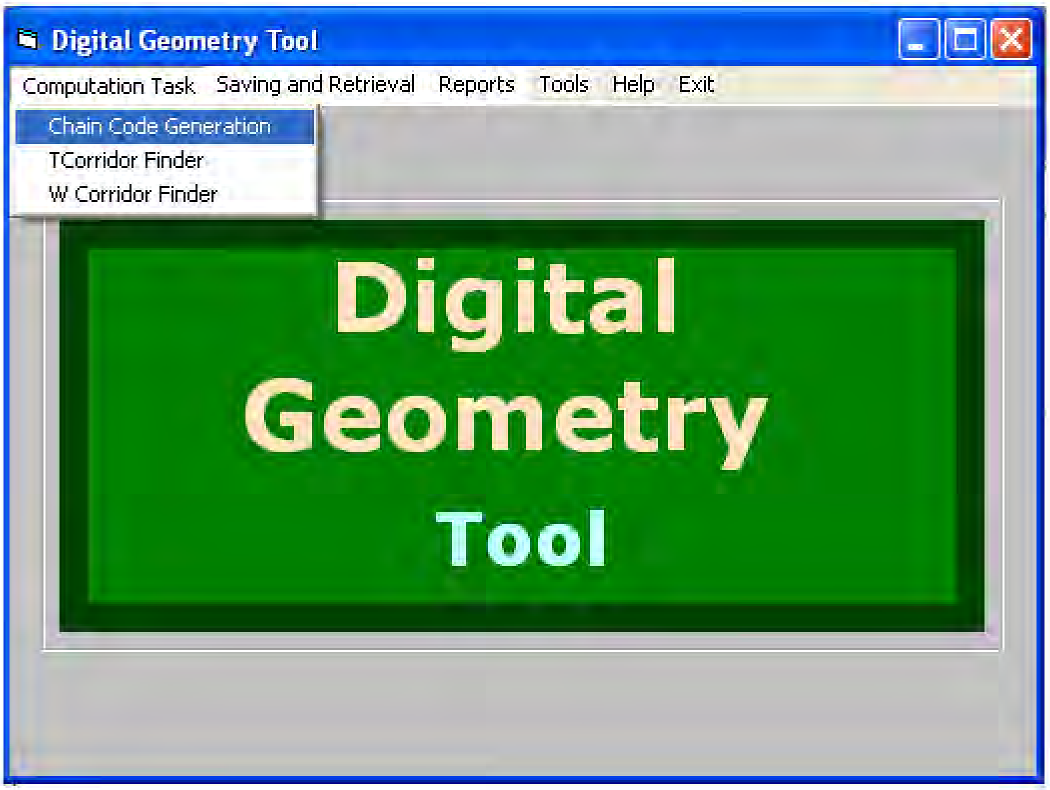
Complexity of Algorithms :

Complexity of T-Corridor is **O(n)** where n is # vertical lines of the grid that the given straight line segment **l** intersect i.e. n is a measure of length of the given straight line segment. This is because each of the Steps 2,3,4,5 takes **O(n)** time and **O(n) + O(n) + O(n) + O(n) = O(n).**
All the other steps takes constant time.

Both the Algorithm used for W-Corridor finding will have similar type of complexity analysis.
In Algorithm 4.1 Step1 takes **O(n)** time , rotation takes **O(n)** time. And due to loop there are maximum **O(n)** no of rotation hence the total complexity is **O(n^2)** in worst case**..**

Similar analysis shows that Algorithm 4.1 has same worst case complexity. But Algorithm 4.1 is having output sensitive runtime since total number of rotation is exactly total no of edge found by this algorithm, each rotating line stops at some position where one edge is added in the output list. So more precise complexity would be **O(h.n)** where **h** is the # edge reported by this algorithm.

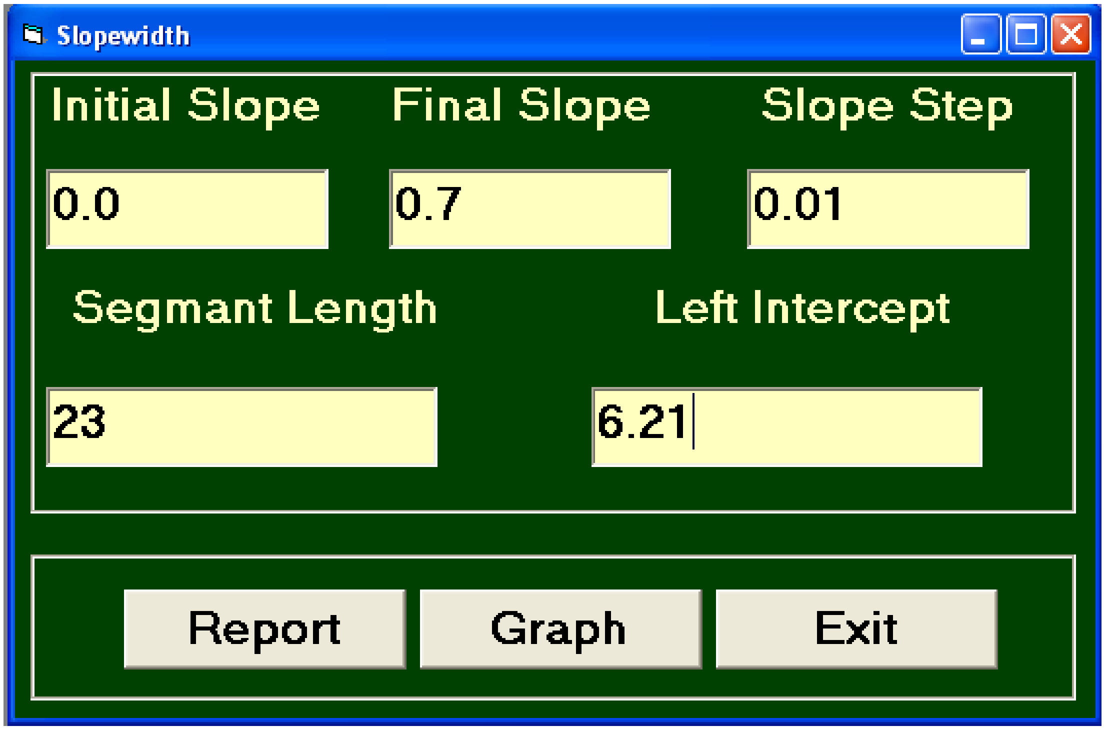
Simulation Tool and Experiments



We have developed a version of DSS investigation tool which is used in our simulation of several algorithms described above and also for elementary tasks related to them like chain Code Generation. We here briefly describe the tool's functionalities. Some Report And Graphs Generated by the Tool is also presented at the end as Experimental Results.

Above is the Screenshot of the initial Screen called Main Menu.

Following is the Screenshot of the Report Generator Which will tell us how Width varies With slope when Straight Line Segment have fixed Length. There are option for Graphical plotting.



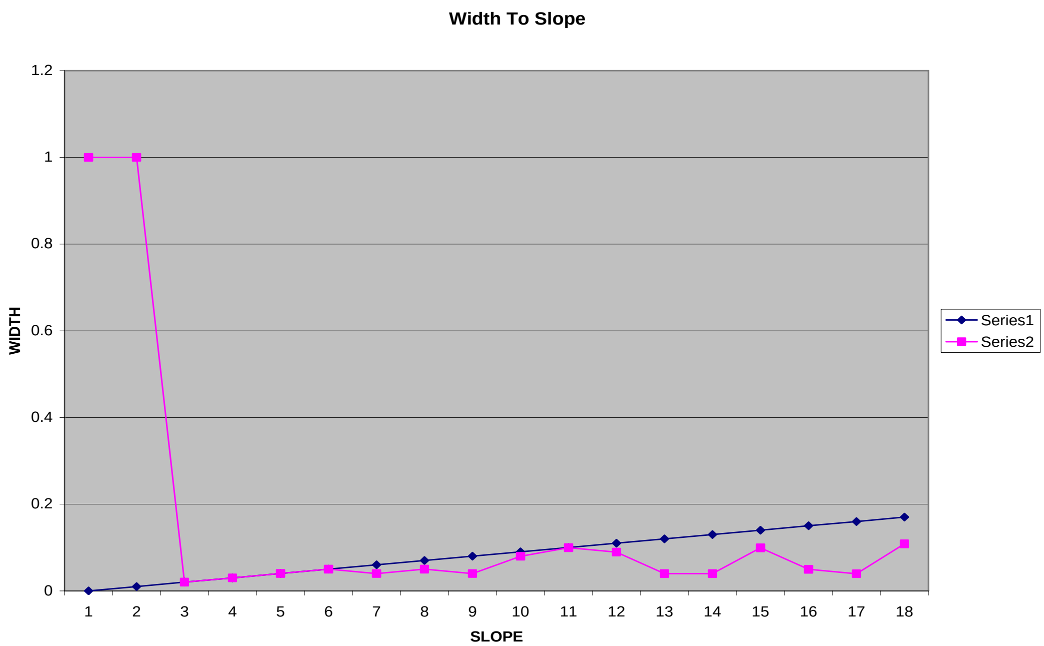
Here we Show a Sample Repot Generated by above interface in the following page. The variation of slope is with 0.0 to 0.17 with Step 0.01.Y-intercept is 6.21 and Length of the Line is 23.Left and Right End indicates X- Coordinate of the Left and Right End Respectively.

Then we present a graph generated by the system plotting the slope with T-Corridor Width.

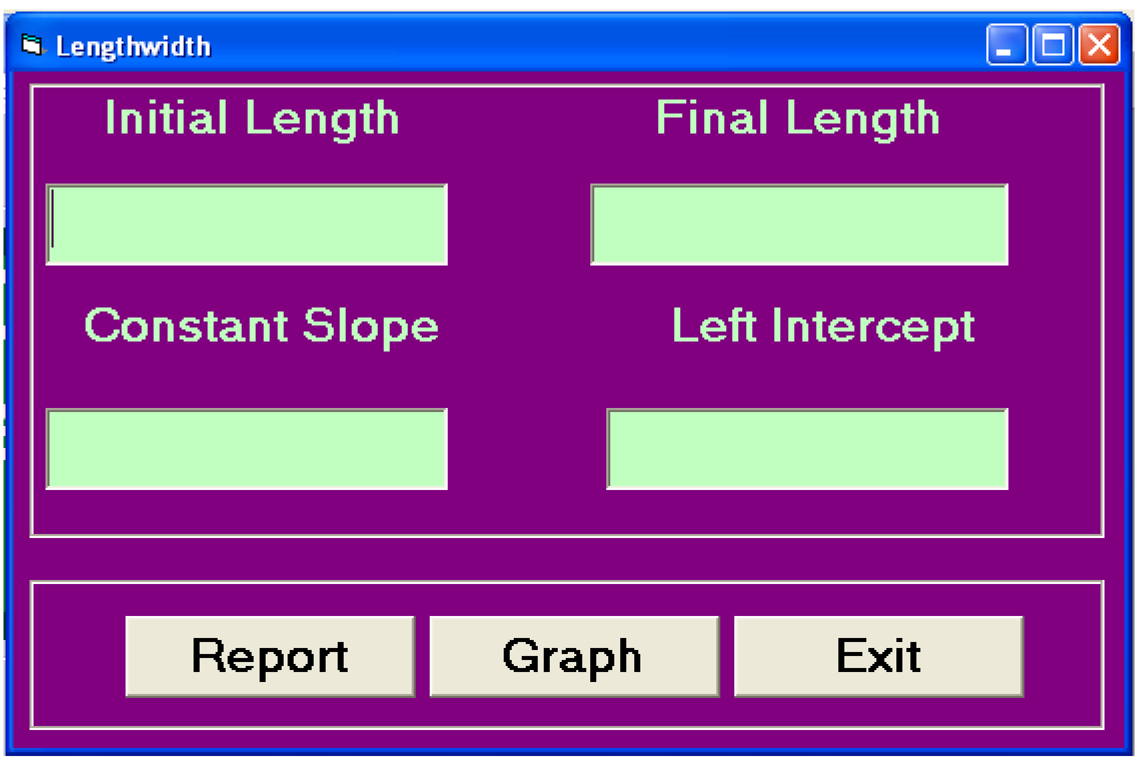
Digital Geometry Report
Slope vrs. Width Table

Y-Intercept: 6.21			
Length of Line: 23			
Left End: 2			
Right End:25			
Slope	**T-Cor. Width**	**T-Cor. Area**	**Chain Code**
0	1	23	0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0.01	1	23	0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0.02	1.99960011995997E-02	0.459908027590793	0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0.03	2.99865091056713E-02	0.689689709430439	0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0.04	3.99680383488716E-02	0.919264882024047	0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0.05	4.99376169438921E-02	1.14856518970952	0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0.06	3.99281938186312E-02	0.918348457828517	0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
0.07	4.98779483570812E-02	1.14719281221287	0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
0.08	0.039872611141445	0.917070056253236	0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
0.09	7.96779551074584E-02	1.83259296747154	0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0.1	9.95037190209986E-02	2.28858553748297	0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
0.11	8.94603920341888E-02	2.05758901678634	0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
0.12	3.97150735394769E-02	0.913446691407969	0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
0.13	3.96662246937295E-02	0.91232316795578	0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
0.14	9.90341746674327E-02	2.27778601735095	0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
0.15	4.94468176434147E-02	1.13727680579854	0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0,
0.16	3.94976252766682E-02	0.90844538136337	0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0,
0.17	0.108444143133679	2.49421529207463	0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,

Here is the Graph generated by the above system.

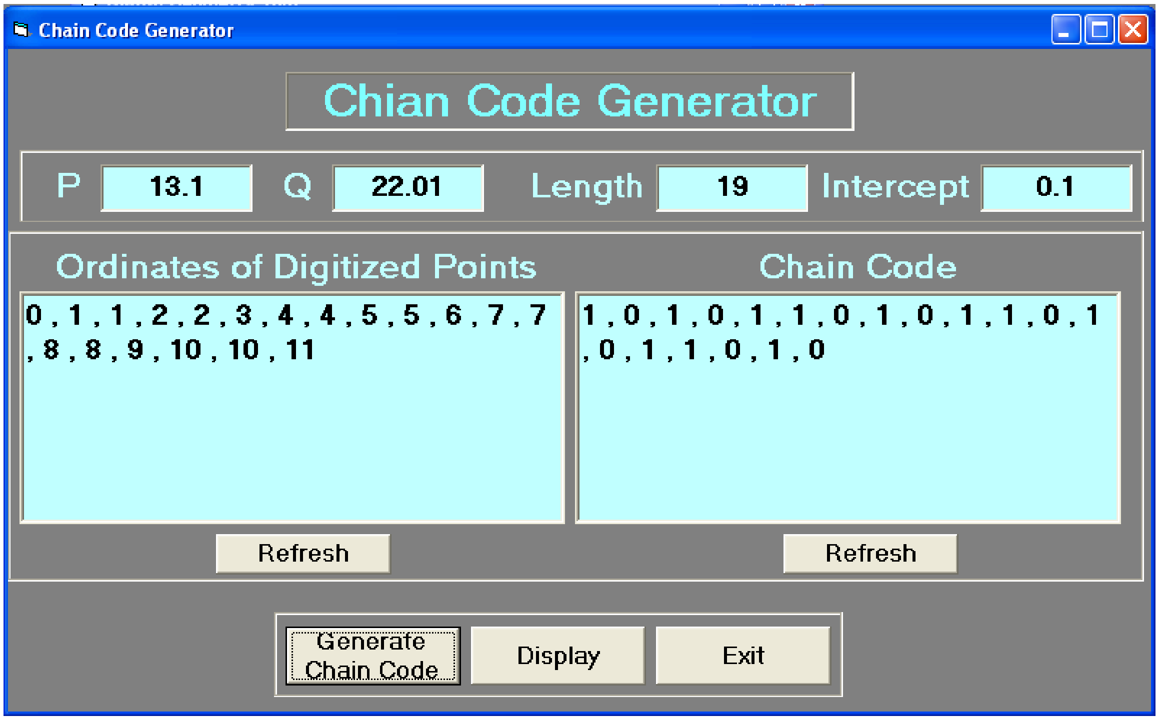


Similar form for Reporting and Plotting Length of Segment with Width when Slope Is Fixed.

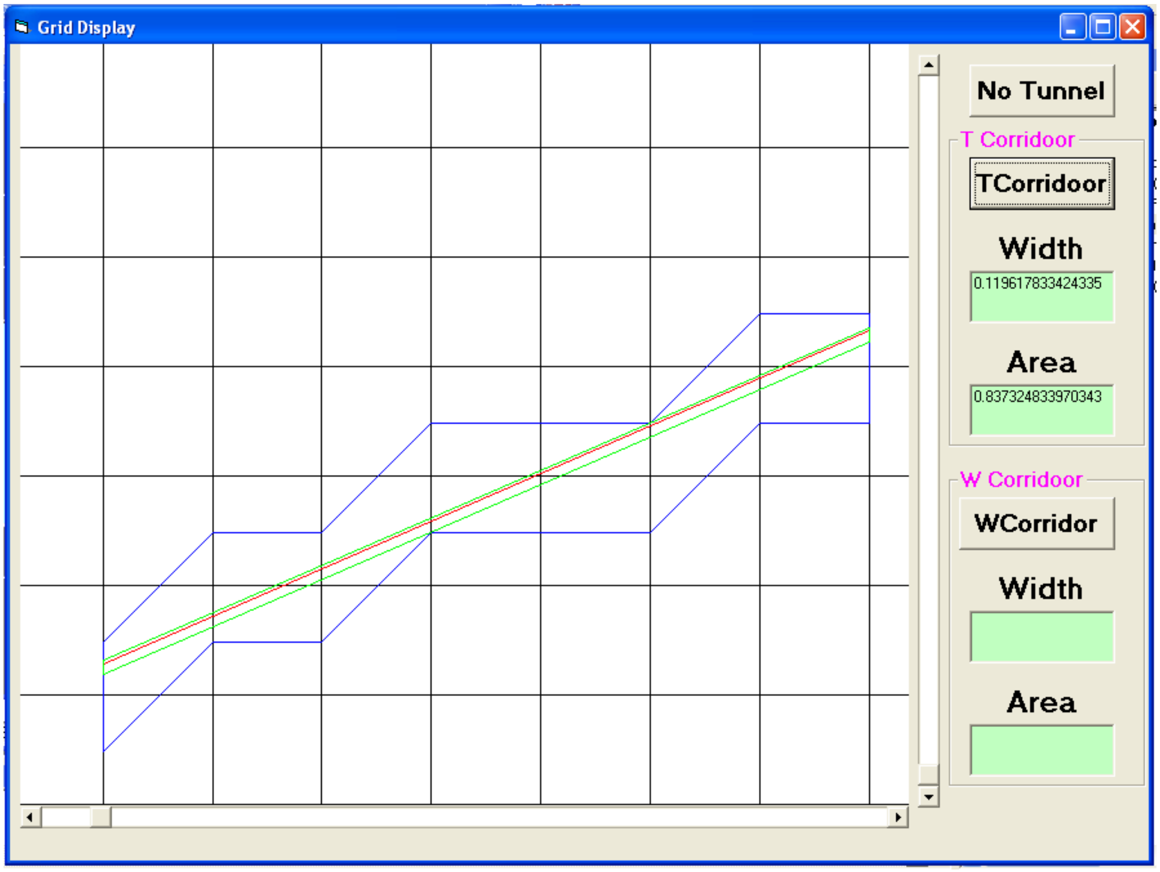


Diss/07/01/196

This is a form for Ordinate of Digitized Grid Point Finder and Chain Code generator.



The Display button of the above displays the following graphical representation of Tunnel , T-Corridor and W-Corridor.


Grid Display
No Tunnel
T Corridoor
TCorridoor
Width
0.119617833424335
Area
0.837324833970343
W Corridoor
WCorridor
Width
Area

References

7.0 Test Books :

7.0.0) Digital Geometry – Reinhard Klette and Azriel Rosenfeld
Morgan Kaufman Publishers

7.1 Papers :

[1] Partha Bhowmick and Bhargab B. Bhattacharya. Fast Polygonal Approximation of Digital Curves Using Relaxed Straightness Properties. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE.

[2] Ulrich Eckhardt1 and Helene Reiter. Polygonal Representations of Digital Sets. Algorithmica © 2003 Springer-Verlag New York Inc.

[3] D. Coeurjolly, I. Sivignon, F. Dupont, F. Feschet, J.-M. Chassery . Ondigital plane preimage structure. Discrete Applied Mathematics 151 (2005) 78 – 92. ELSEVIER.

[4] L. Wu. On the chain code of a line. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-4:347-353,1982.

[5] P. D. Stephenson and B. Litow. Why step when you can run: Iterative line digitization algorithms based on hierarchies of runs. IEEE Computer Graphics and Applications, 20(6):76-84, November 2000.

[6] A. Rosenfeld. Digital straight line segments. IEEE Transactions on Computers, C-23(12):1264-1269, Dec. 1974.

[7] Peter Stephenson and Bruce Litow .Making the DDA run:Two-dimensional ray traversal using runs and runs of runs.

[8] Martin N. Huxley and Jovisa Zunic.The Number of N-Point Digital DiscsIEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 29, NO. 1, JANUARY 2007

[9] L. Dorst, A.W.M. Smeulders. Discrete straight line segments: parameters,primitives and properties. In: R. Melter,P. Bhattacharya, A. Rosenfeld(eds): Ser. Contemp. Maths., Amer. Math. Soc. 119:45{62, 1991.

[10] F. Feschet L. Tougne.On the min DSS problem of closed discrete curves.Technical report LLAIC - 2002/06.

[11] T. Asano, Y. Kawamura, R. Klette, and K. Obokkata. Digital curve approximation with length evaluation. IEICE Trans. Fundamentals Electronics Communication Computer Sciences, 86:987–994, 2003.

[12] N. Aziz and S. Bhat. On the computation of integral properties of objects. Advances Engineering Software, 12:174–180, 1990.

[13] E. Barcucci, A. Del Lungo, and R. Pinzani. Deco polyominoes permutations and random generation. Theoretical Computer Science, 159:29–42, 1996.

[14] C. Berenstein and D. Lavine. On the number of digital straight line segments.IEEE Trans. Pattern Analysis Machine Intelligence, 10:880–887, 1988.

[15] G. Bertrand and R.Malgouyres. Topological properties of discrete surfaces. In: S. Miguet, A. Montanvert and S. Ub´eda, editors. Discrete Geometry for Computer Imagery, 325–336. Lecture Notes in Computer Science 1176, Springer, Berlin, Germany, 1996.

[16] Y. Bertrand, G. Damiand, and C. Fiorio. Topological encoding of 3-D segmented images. In: G. Borgefors, I. Nystr¨om, and G. Sanniti di Baja, editors. Discrete Geometry for Computer Imagery, 311–324. Lecture Notes in Computer Science 1953, Springer, Berlin, Germany, 2000.

[17] G. Bongiovanni, F. Luccio, and A. Zorat. The discrete equation of the straight line. IEEE Trans. Computers, 24:310–313, 1975.

[18] A.M. Bruckstein. Self-similarity properties of digitized straight lines. Contemporary Mathematics, 119:1–20, 1991.

[19] V. Caglioti. On the uncertainty of straight lines in digital images. CVGIP: Graphical Models Image Processing, 55:255–270, 1993.

[20] S. Chattopadhyay and P.P. Das. A new method of analysis for discrete straight lines. Pattern Recognition Letters, 12:747–755, 1991.

[21] L. Dorst and R.P.W. Duin. Spirograph theory: A framework for calculations on digitized straight lines. IEEE Trans. Pattern Analysis Machine Intelligence, 6:632–639, 1984.

[22] L. Dorst and A.W.M. Smeulders. Discrete representations of straight lines. IEEE Trans. Pattern Analysis Machine Intelligence, 6:450–462, 1984.

[23] L. Dorst and A.W.M. Smeulders. Discrete straight line segments: Parameters, primitives and properties. Contemporary Mathematics, 119:45–62, 1991.

[24] M. Gaafar. Convexity verification, block-chords and digital straight lines.

Computer Graphics Image Processing, 6:361–370, 1977.

[25] C.E. Kim. On cellular straight line segments. Computer Graphics Image Processing, 18:369–381, 1982.

[26] H. Klaasman. Some aspects of the accuracy of the approximated position of a straight line on a square grid. Computer Graphics Image Processing, 4:225–235, 1975.

[27] J. Koplowitz, M. Lindenbaum, and A.M. Bruckstein. The number of digital straight lines on an N×N grid. IEEE Trans. Information Theory, 36:192–197, 1990.

[28] H.C. Lee and K.S. Fu. Using the FFT to determine digital straight line chain codes. Computer Graphics Image Processing, 18:359–368, 1982.

[29] M. Lindenbaum. Compression of chain codes using digital straight line sequences. Pattern Recognition Letters, 7:167–171, 1988.

[30] C. Ronse. A simple proof of Rosenfeld's characterization of digital straight line segments. Pattern Recognition Letters, 3:323–326, 1985.

[31] R. Shoucri, R. Benesch, and S. Thomas. Note on the determination of a digital straight line from chain codes. Computer Vision, Graphics, Image Processing, 29:133–139, 1985.

[32] A.M. Vossepoel and A.W.M. Smeulders. Vector code probability and metrication error in the representation of straight lines of finite length. Computer Graphics Image Processing, 20:347–364, 1982.

[33] K. Voss. Coding of digital straight lines by continued fractions. Computers Artificial Intelligence, 10:75–80, 1991.