Measurement of Shortest Distance between any two vertices in a Trianglar Grid

Suman Kumar Chanda*

*Dept. of Computer Science, Panchakot Mahavidyalaya, Purulia, W.B.-723121, India;
email:sumanchanda1991@gmail.com

Abstract

The triangular grid places points at the vertices of equilateral triangles. The length of each grid side is dependent on the area of the sampling space. The area is then divided into N number of blocks, where N is the number of desired samples. Each block is then treated as an equilateral triangle. An algorithm to find the shortest distance between two points is presented here. The Cartesian co-ordinate system of the triangular grid is also discussed here.

Keywords: Triangular Grid, Co-ordinate System, G_6 Graph, G_8 Graph

Introduction

A graph is a triangular grid graph if it is an induced sub graph of a tiling of the plane with equilateral triangles [1,2]. It is also called an isometric grid which is a grid formed by tiling the plane regularly with equilateral triangles. This is a graph with the intersection of grid lines being the vertices and the line segments between two opposite co-ordinates. Let us assume a Cartesian co-ordinate system (x, y), it forms a triangular grid if there is an interconnection between the co-ordinate (x, y) and $(x + 1, y + 1)$ or (x, y) and $(x - 1, y + 1)$. In this work, considered the interconnection between the co-ordinate (x, y) and $(x + 1, y + 1)$ to get a triangular grid.

Co-ordinate System

Let $v(x, y)$ is a central vertex with diagonal in one direction in triangular grid then vertex v has its six neighbors $(x + 1, y)$, $(x + 1, y + 1)$ or $(x - 1, y + 1)$, $(x, y + 1)$, $(x - 1, y)$, $(x - 1, y - 1)$ or $(x + 1, y - 1)$, $(x, y - 1)$. These co-ordinates can be represented as the following figure:

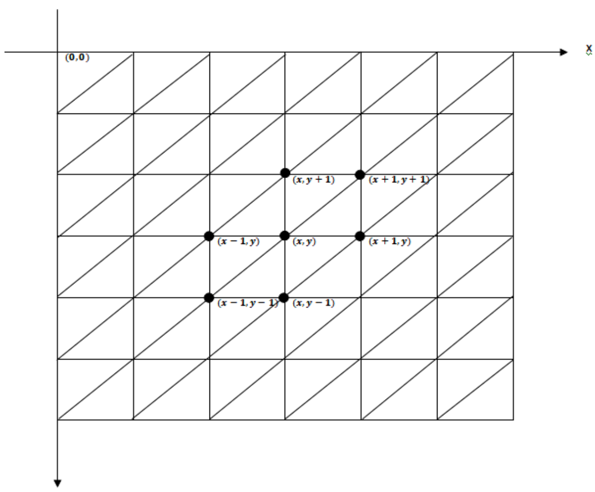

Fig.1: Co-ordinate Representation of G_6

Let $v(x, y)$ is a central vertex with diagonal in both direction then vertex v has its eight neighbors $(x+1, y)$, $(x+1, y+1)$, $(x-1, y+1)$, $(x, y+1)$, $(x-1, y)$, $(x-1, y-1)$, $(x+1, y-1)$, $(x, y-1)$. These co-ordinates can be represented as the following figure:

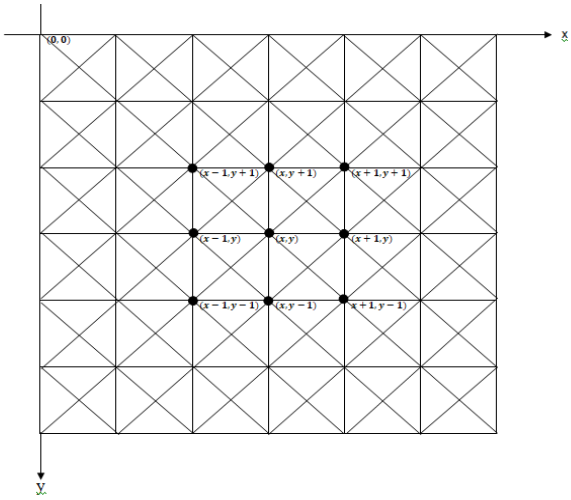

Fig.2: Co-ordinate Representation of G_8

Properties of Triangular Grid

1. A triangular grid graph is a finite induced sub graph of the infinite graph associated with the two-dimensional triangular grid.
2. Each vertex in this graph has same degree 6 (with one diagonal) and degree 8(with two diagonals), so this is a regular infinite graph.
3. Each vertex (x, y) in a triangular grid has six neighbours$(x+1, y)$, $(x+1, y+1)$ or $(x-1, y+1)$, $(x, y+1)$, $(x-1, y)$, $(x-1, y-1)$ or $(x+1, y-1)$, $(x, y-1)$ when we consider diagonal in one direction of the triangular grid.
4. Each vertex (x, y) in a triangular grid has eight neighbours $(x+1, y)$, $(x+1, y+1)$, $(x-1, y+1)$, $(x, y+1)$, $(x-1, y)$, $(x-1, y-1)$, $(x+1, y-1)$, $(x, y-1)$ when we consider diagonal in both direction of the triangular grid.
5. Triangular infinite graph consist of infinite number of equilateral triangles.

1. Algorithms

- Algorithm Name: Distance_g_6.
- **Input:**$(i_1, j_1)$and (i_2, j_2),where (i_1, j_1) and (i_2, j_2) are co-ordinates of two vertices of a infinite graph**.**
- **Output**: Shortest Distance between given two vertices.

<u>Steps of Algorithm:</u>

Step 1. if $(i_1 < 0 || i_2 < 0 || j_1 < 0 || j_2 < 0)$

then Input co-ordinate does not exist**.**

Step 2. if $(j_1 == j_2)$

then $distance = abs(i_2 - i_1)$;

Step 3. if $(i_1 == i_2)$

then $distance = abs(j_2 - j_1)$;

Step 4 .if $((i_1 < i_2)$ && $(j_1 > j_2))$

then $\boldsymbol{swap}(i_1, i_2)$ and $\boldsymbol{swap}(j_1, j_2)$;

Step 5. if $((i_1 > i_2)$ && $(j_1 > j_2))$

then $\boldsymbol{swap}(i_1, i_2)$ and $\boldsymbol{swap}(j_1, j_2)$;

Step 6. if $\big((abs(i_1 - i_2) == abs(j_1 - j_2)\ \&\&\ (i_1 > i_2)\big)$

then $distance = (abs(i_1 - i_2) + abs(j_1 - j_2))/2$;

Step 7. if $((i_1 > i_2)$&&$(j_1 < j_2))$ then check

Step 7.1 if $(abs(i_1 - i_2) > abs(j_1 - j_2))$

then $j == j_2$;

$i = i_1 - abs(j_1 - j_2)$;

else

$i = i_2$;

$j = abs(i_1 - i_2) + j_1$;

$distance1 = (abs(i_1 - i) + abs(j_1 - j))/2$;

$distance2 = (abs(i_2 - i) + abs(j_2 - j))$;

$distance = distance1 + distance2$;

Step 8. if $((i_1 < i_2)$ && $(j_1 < j_2))$

then $distance = \big(abs(i_1 - i_2) + abs(j_1 - j_2)\big)$;

- Algorithm Name: Distance_g_8
- **Input:** (i_1, j_1) and (i_2, j_2), where (i_1, j_1) and (i_2, j_2) are co-ordinates of two vertices of a infinite graph.
- **Output**: Shortest Distance between given two vertices.

Steps of Algorithm:

Step 1. if $(i_1 < 0 || i_2 < 0 || j_1 < 0 || j_2 < 0)$

then Input co-ordinate does not exist

Step 2. if $(j_1 == j_2)$

then $distance = abs(i_2 - i_1)$;

Step 3. if $(i_1 == i_2)$

then $distance = abs(j_2 - j_1)$;

Step 4. if $((i_1 < i_2)$ && $(j_1 > j_2))$

then $\boldsymbol{swap}(i_1, i_2)$ and $\boldsymbol{swap}(j_1, j_2)$;

Step 5. if $((i_1 > i_2)$ && $(j_1 > j_2))$

then $\boldsymbol{swap}(i_1, i_2)$ and $\boldsymbol{swap}(j_1, j_2)$;

Step 6. if $((i_1 < i_2)$ && $(j_1 < j_2))$

then $\boldsymbol{swap}(i_1, i_2)$ and $\boldsymbol{swap}(j_1, j_2)$;

Step 7. if $\big((abs(i_1 - i_2) == abs(j_1 - j_2)\ \&\&\ (i_1 > i_2)\big)$

then $distance = (abs(i_1 - i_2) + abs(j_1 - j_2))/2$;

Step 8. if $\big((i_1 > i_2)\ \&\&\ (j_1 < j_2)\big)$ then check

Step 8.1 if $(abs(i_1 - i_2) > abs(j_1 - j_2))$

then $j = j_2$;

$i = i_1 - abs(j_1 - j_2)$;

else

$i = i_2$;

\`$j = abs(i_1 - i_2) + j_1$;

$distance1 = (abs(i_1 - i) + abs(j_1 - j))/2$;

$distance2 = (abs(i_2 - i) + abs(j_2 - j))$;

$distance = distance1 + distance2$;

Step 9. if $\big((i_1 > i_2)\ \&\&\ (j_1 > j_2)\big)$ then check

Step 9.1 if $(abs(i_1 - i_2) > abs(j_1 - j_2))$

then $j = j_2$;

$i = i_1 - abs(j_1 - j_2)$;

$d_1 = abs(i - i_2) + abs(j - j_2)$;

$d_2 = (abs(i_1 - i) + abs(j_1 - j))/2$;

else

$i = i_1$;

$j = abs(i_1 - i_2) + j_2$;

$d_1 = \big(abs(i_1 - i) + abs(j_1 - j)\big)$;

$d_2 = (abs(i - i_2) + abs(j - j_2))/2$;

$distance1 = \big(abs(i_1 - i) + abs(j_1 - j)\big)$;

$distance2 = (abs(i - i_2) + abs(j - j_2))/2$;

$distance = distance1 + distance2$;

Step 10. if $\big((abs(i_1 - i_2) == abs(j_1 - j_2)\ \&\&\ (i_1 < i_2)\big)$

then $distance = (abs(i_1 - i_2) + abs(j_1 - j_2))/2$;

Step 11. if $\big((i_1 < i_2)\ \&\&\ (j_1 < j_2)\big)$

then $distance = \big(abs(i_1 - i_2) + abs(j_1 - j_2)\big)$;

Note: The *abs()* returns the absolute value.

- **Distance between any two vertices with degree 6(G_6):**

When any two vertices are given then it checks their coordinate positions. There may be different co-ordinates combinations. Distance can be measured depending on the co-ordinate positions.

i. When first co-ordinate positions of two given vertices are same then perform absolute difference between second co-ordinate positions of given vertices which can be mathematically described as:

$d = abs(j_2 - j_1)$; where d is calculated distance, j_1 & j_2 are second co-ordinate positions of the given vertices.

ii. When second co-ordinate positions of two given vertices are same then perform absolute difference between first co-ordinate positions of given vertices which can be mathematically described as:

$d = abs(i_2 - i_1)$; where d is calculated distance, i_1 & i_2 are second co-ordinate positions of the given vertices.

iii. When there is a diagonal between given vertices then distance can be measured as:

$d = (abs(i_1 - i_2) + abs(j_1 - j_2))/2$; where d is calculated distance, $i_1, i_2$and j_1, j_2 are first and second co-ordinate positions of the given vertices

iv. When first position of first vertex is greater than first position of second vertex and second position of first vertex is less than second position of second vertex then check whether absolute difference between (i_1, i_2) is greater than absolute difference between (j_1, j_2),if it is true then set the new co-ordinate position of diagonal as:

$$j = j_2.$$

$i = i_1 - abs(j_1 - j_2)$; where i & j are new co-ordinate position.

Otherwise set the new co-ordinate position of diagonal as

$$i = i_2.$$

$j = abs(i_1 - i_2) + j_1$, where i & j are new co-ordinate position.

Now we can calculate the distance as

$$d_1 = (abs(i_1 - i) + abs(j_1 - j))/2.$$

$$d_2 = (abs(i_2 - i) + abs(j_2 - j)).$$

$d = d_1 + d_2$, where d_1 & d_2 are two intermediate values of distance calculation and d is the final resultant distance.

v. When first position of first vertex is less than first position of second vertex and second position of the first vertex is less than second position of the second vertex then distance can be calculated as:

$d = (abs(i_1 - i_2) + abs(j_1 - j_2))$, where d is calculated distance.

- **Distance between any two vertices with degree 8(G_8)**:

When any two vertices are given then it checks their coordinate positions. There may be different co-ordinates combinations. Distance can be measured depending on the co-ordinate positions.

i. When first co-ordinate positions of two given vertices are same then perform absolute difference between second co-ordinate positions of given vertices which can be mathematically described as:

$d = abs(j_2 - j_1)$, where d is calculated distance, j_1 & j_2 are second co-ordinate positions of the given vertices.

ii. When second co-ordinate positions of two given vertices are same then perform absolute difference between first co-ordinate positions of given vertices which can be mathematically described as:

$d = abs(i_2 - i_1)$, where d is calculated distance, i_1 & i_2 are first positions of the given two vertices.

iii. When there is a diagonal between given vertices then distance can be measured as:

$d = (abs(i_1 - i_2) + abs(j_1 - j_2))/2$, where d is calculated distance, i_1, i_2 and j_1, j_2 are first and second co-ordinate positions of the given vertices.

iv. When first position of first vertex is greater than first position of second vertex and second position of first vertex is less than second position of second vertex then check whether absolute difference between (i_1, i_2) is greater than absolute difference (j_1, j_2), if it is true then set the new co-ordinate position of diagonal as

$j = j_2$.

$i = i_1 - abs(j_1 - j_2)$, where i & j are new co-ordinate position.

Otherwise set the new co-ordinate position of diagonal as

$i = i_2$

$j = abs(i_1 - i_2) + j_1$, where i & j are new co-ordinate position.

Now we can calculate the distance as

$d_1 = (abs(i_1 - i) + abs(j_1 - j))/2$.

$d_2 = (abs(i_2 - i) + abs(j_2 - j))$.

$d = d_1 + d_2$, where d_1 & d_2 are two intermediate values of distance calculation and d is the final resultant distance.

v. When first position of first vertex is greater than the first position of the second vertex and second position of the first vertex is also greater than second position of the second vertex then check wheather absolute difference of (i_1, i_2) is greater than absolute difference of (j_1, j_2), if its true then set the new co-ordinate position as

$$j = j_2.$$

$$i = i_1 - abs(j_1 - j_2).$$

Now distance can be calculated as

$$d_1 = abs(i - i_2) + abs(j - j_2)\ .$$

$$d_2 = (abs(i_1 - i) + abs(j_1 - j))/2,$$

where d_1 & d_2 are two intermediate values of distance calculation .

otherwise, set new co-ordinate position as

$$i = i_1.$$

$$j = abs(i_1 - i_2) + j_2.$$

Now distance can be calculated as

$$d_1 = \left(abs(i_1 - i) + abs(j_1 - j)\right).$$

$d_2 = (abs(i - i_2) + abs(j - j_2))/2$, where d_1 & d_2 are two intermediate values of distance calculation.

Now final distance can be calculated as $d = d_1 + d_2$, where d is the final distance.

vi. When absolute difference of (i_1, i_2) is equal to absolute difference of (j_1, j_2),and first position of the first vertex is less than the first position of the second vertex then distance can be measured as :

$d = (abs(i_1 - i_2) + abs(j_1 - j_2))/2$, where (i_1, j_1) and (i_2, j_2) are co-ordinates of given vertices and d is the distance.

vii. When first position of the first vertex is less than the first position of the second vertex and second position of the first vertex is less than second position of the second vertex then distance can be measured as:

$d = \left(abs(i_1 - i_2) + abs(j_1 - j_2)\right)$, d is the distance.

Conclusion

The algorithm can be extended for negative co-ordinate position also. An application of these properties arises in telecommunications and computer vision (problems of determining the shape of an object represented by a cluster of points on a grid), in molecular biology (protein folding) [3], and in configurational statistics of polymers .Cyclic properties of triangular grid graphs can also be used in the design of cellular networks since these networks are generally modelled as induced subgraphs of the infinite two - dimensional triangular grid.

References

1. Valery S., Gordon, Y. L. and Orlovich, F. W. (2008). Hamiltonian properties of triangular grid graphs, Discrete Mathematics, 308(24), 6166-6188.
2. Gardner, M. (1986). *Knotted Doughnuts and Other Mathematical Entertainments.* New York: W. H. Freeman.
3. Agarwala, R.S., Batzoglou, V., Dancik, S.E., Dwcatur, M., Farach, S. and Hannenhalli, S. (1997). Skiena, Local rules for protein folding on a triangularlattice and generalized hydrophobicity in the HP model, Journal of Computational Biology, 4, 275-296.