

# COMPARATIVE ANALYSIS OF MANHATTAN AND EUCLIDEAN DISTANCE METRICS USING A\* ALGORITHM

Shrawan Kumar Sharma<sup>1</sup> and Shiv Kumar<sup>2</sup>

<sup>1</sup>Assistant Professor, Department of Computer Science Vision Group of Collages, Chittorgarh, India

<sup>2</sup>Assitant Professor Department of Computer Science of Engineering, Mewar University, Gangrar, Chittorgarh, India  
{E-mail : <sup>1</sup>shrawansharma7070@gmail.com, <sup>2</sup>shivkumar004@gmail.com}

## Abstract

A\* algorithm is mostly used to find the shortest path searching for Grid and map based path like road and rail network. A\* algorithm generally find the path using Heuristic technique. The main idea behind the A\* find the shortest path is the calculating the path (start to destination) very fast. The main work of this paper is that study of two distance metrics viz. Euclidean and Manhattan. We can perform some those metrics experiments in A\* algorithm to validate the study. The result is show the comparative analysis of those metrics.

**Key Words :** A\* Algorithm, Shortest path, Euclidean, Manhattan, Heuristic

## 1. Introduction

In artificial intelligence problem is solved in two steps. In first steps, problem is represented by state space search tree format with the help of problem description attributes like initial state, goal state, operators etc. In Second steps, optimal solution is searched to reach the goal, will be discussed in this chapter in sufficient details. Heuristic function is used to solve complex problem and working of this function will also be addressed. [1].

Searching path from a given or presented road networks from a source location to destination place is day to day problem for everyone. Tourists as well as general public are facing this type of problem every day. There are number of application available in market that needs to solve a huge number of such route queries. Current commercial solutions usually are slow or inaccurate [2].

Heuristic search: A heuristic is a technique to solve a problem where goal state is fully or partially known to the user with initial state. Heuristics play a major role in solving complex problem with optimality as well as completeness. Heuristic algorithm helps to reduce the number of alternatives from an exponential number to polynomial number. Application areas of heuristic search algorithm are decision making, planning, searching shortest path etc. [3].

$f(n)$  is the sum of  $g(n)$  and  $h(n)$  (1)

## 2. Research Methodologies

### 2.1 Distance metrics overview

Euclidean distance is calculated by using following formula [4]. For example let  $X_1, Y_1$  is coordinates of X while  $X_2, Y_2$  is coordinates of Y then distance between X & Y is given by:

$$\begin{aligned} DXY &= \sqrt{(X_1 - Y_1)^2 + (X_2 - Y_2)^2} & (2) \\ &= \sqrt{(2-3)^2 + (3-5)^2 + (4-7)^2 + (5-9)^2} \\ &= \sqrt{(-1)^2 + (-2)^2 + (-3)^2 + (-4)^2} \\ &= \sqrt{30} \\ &= 5.5 \end{aligned}$$

Similarly Manhattan distance calculation between X and Y is given by.

$$\begin{aligned} DXY &= |X_1 - Y_1| + |X_2 - Y_2| & (3) \\ &= |2-3| + |3-5| + |4-7| + |5-9| \\ &= 1 + 2 + 3 + 4 \\ &= 10 \end{aligned}$$

It computes the absolute difference between coordinates of the two objects [4].

### 2.2 A\* Algorithm

The A\* algorithm is combination of uniform-cost search and greedy search algorithm. Implementation of A\* algorithm is just like Best-first search algorithm except the cost estimation process i.e.  $f(n) = g(n) + h(n)$ , where  $g(n)$  is the actual cost from initial state to current state and  $h(n)$  is the estimates cost from current state to a goal state[5]. That is why  $f(n)$  is minimum distance path of any solution path going through node n. At each and every step lowest  $f(n)$  is chosen for node n. The algorithm terminates when a goal states found. If  $h(n)$  is less then  $g(n)$  then A\* provides best solution. The performance of A\* similar to Dijkstra algorithm if  $h(n)=0$ [6].

#### 2.1.1 Algorithm Pseudo code:

- Initialization the Start and End Position.
- Search open list and check list is empty or not

- While there are still more possible next steps in the open list
  - A) Look for the node which has the lowest f on the open list. Refer to this node as the current node.
  - B) Remove it from the open list and add it to the closed
  - C) Consider each neighbor of the step. For each neighbor:
    - i) Calculate the path cost of reaching the neighbor
    - ii) If the cost is less than the cost known for this location then remove it from the open or closed lists (since we've now found a better route)
- Search goal and generate path

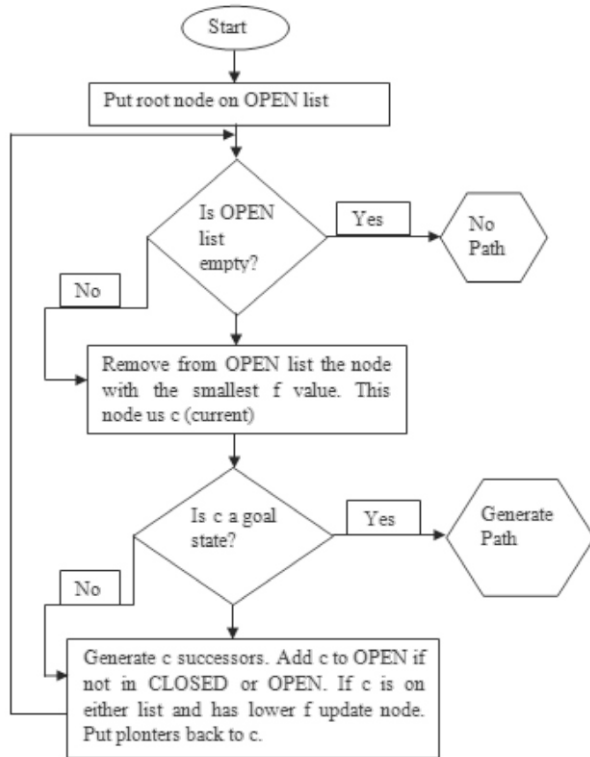


Fig. 1 : Flow chart of A\* Algorithm

### 3. Experimental Work

In this section, we conduct an experiment to find the shortest path in A\* algorithm between two points. The calculation of shortest path done by Euclidean and Manhattan distance. In this we can calculate the Time and number of operation is performed to calculate those distance metrics and find which distance metrics is best for calculation the fast shortest path.

### 3.1 A\* with Manhattan and Euclidean distance

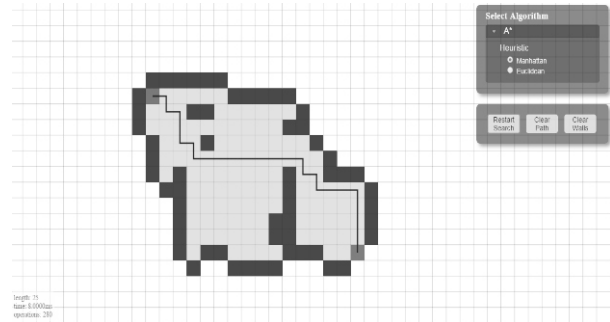


Fig. 2 : A\* with Manhattan distance

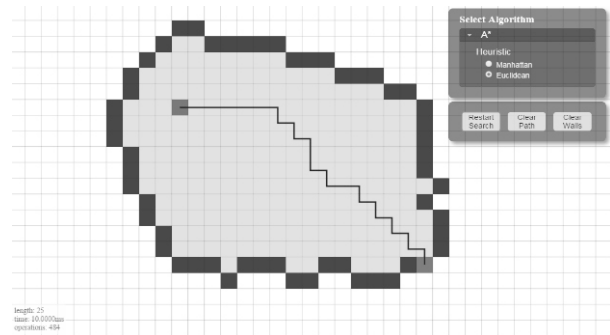


Fig. 3 : A\* with Euclidean distance metrics

### 4. Result Analyses

Searching time is calculated for three cases as shown in table 1 (Comparative analysis of Manhattan and Euclidean distance) as well as figure (fig no.). case 1 case 2 and case 3 have path length of 25, 34 and 48 respectively.

Where:

Length = Total number of minimum f value grid  
 Time = End time – Start time

Table 1 : Comparative analysis of Manhattan and Euclidean distance

	Case 1		Case 2		Case 3	
	Manhattan	Euclidean	Manhattan	Euclidean	Manhattan	Euclidean
Length	25	25	34	34	48	48
Time(ms)	8	10	14	21	18	26

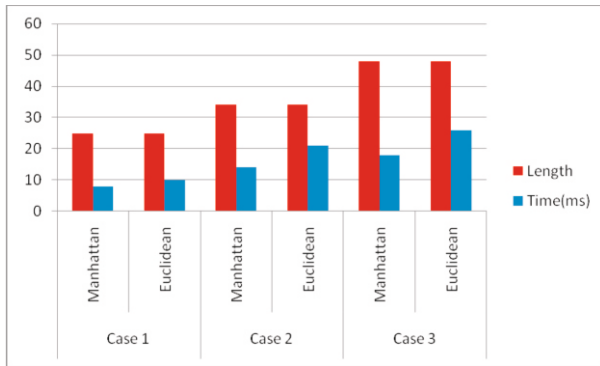


Fig. 4 : Graphical representation of comparative analysis of Manhattan and Euclidean distance

As per experimental analysis based on execution time, A\* using Manhattan distance is better than the Euclidean distance.

## 5. Conclusions

Manhattan distance metrics performs better than the Euclidean distance metrics in A\* algorithm. The main reason of better performance of Manhattan distance is fast calculation of the all parameters as well as generation of fast shortest path.

## Acknowledgement

Foremost, I would like to express my sincere gratitude to my advisor Shiv Kumar for the continuous support of my study, for his patience, motivation, enthusiasm, and immense knowledge. Besides my advisor, I would like to thank the rest of my Friends and well wisher: Last but not the least; I would like to thank my brother Nand Lal Sharma and my wife Neelam Sharma supporting me spiritually throughout my life.

## References

- [1] Amit Konar, "Artificial Intelligence and Soft Computing Behavioral and Cognitive Modeling of the Human Brain", 2000 by CRC Press LLC p.116
- [2] Shrawan Kumar Sharma, B.L.Pal, "Shortest Path Searching for Road Network using A\* Algorithm", IJCSMC,
- [3] Ben Coppin, "Artificial Intelligence Illuminated", Copyright © 2004 by Jones and Bartlett Publishers, page 91
- [4] Kahkashan Kouser, Sunita, "A comparative study of K Means Algorithm by Different Distance Measures", IJIRCCE, Vol. 1, Issue 9, November 2013
- [5] Sreekanth Reddy Kallem, "Artificial Intelligence Algorithms", IOSRICE Volume 6, Issue 3 (Sep-Oct. 2012), pp. 01-08
- [6] Gardenwiki, "Path Finding Tutorial", "[http://wiki.gamegardens.com/Path\\_Finding\\_Tutorial](http://wiki.gamegardens.com/Path_Finding_Tutorial)", December 2006