Y. Singh, S. Sharma, R. Sutton, and D. Hatton. Towards use of Dijkstra algorithm for optimal navigation of an unmanned surface vehicle in a real-time marine environment with results from artificial potential field. *TransNav: International Journal on Marine Navigation and Safety of Sea Transportation* **12**(1):125-131, Mar. 2018.

Publisher URL:
<http://www.transnav.eu/Article2_Towards_use_of_Dijkstra_Algorithm_Singh,45,795.html>.
doi:10.12716/1001.12.01.14

TRANSNAV
http://www.transnav.eu
the International Journal
on Marine Navigation
and Safety of Sea Transportation
Volume 12
Number 1
March 2018
DOI: 10.12716/1001.12.01.14

Towards use of Dijkstra Algorithm for Optimal Navigation of an Unmanned Surface Vehicle in a Real-Time Marine Environment with results from Artificial Potential Field

Y. Singh, S. Sharma, R. Sutton & D. Hatton
Autonomous Marine Systems Group, Plymouth University, United Kingdom

ABSTRACT: The growing need of ocean surveying and exploration for scientific and industrial application has led to the requirement of routing strategies for ocean vehicles which are optimal in nature. Most of the op-timal path planning for marine vehicles had been conducted offline in a self-made environment. This paper takes into account a practical marine environment, i.e. Portsmouth Harbour, for finding an optimal path in terms of computational time between source and end points on a real time map for an USV. The current study makes use of a grid map generated from original and uses a Dijkstra algorithm to find the shortest path for a single USV. In order to benchmark the study, a path planning study using a well-known local path planning method artificial path planning (APF) has been conducted in a real time marine environment and effectiveness is measured in terms of path length and computational time.

1 INTRODUCTION

With the growing advances in navigation technologies, there is a greater need to explore oceans for resources as well as for the future needs. Autonomous unmanned vehicles have shown the potential towards various missions of scientific and military significance depending upon the requirement, environment and cost involved (Serreze et al., 2008 and Legrand et al., 2003). Unmanned vehicles can be classified into four categories namely, unmanned aerial vehicles (UAVs), unmanned underwater vehicles (UUVs), unmanned ground vehicles (UGVs) and unmanned surface vehicles (USVs). USVs are watercraft of small (<1 tonnes) or medium (100 tonnes) size in terms of water displacement.

The general architecture for an USV operation in a maritime environment has three basic systems namely, control and path planning, communication and monitoring and obstacle detection and avoidance (ODA), which are responsible for mission planning and execution as shown in figure 1. Path planning is one of the basic subsystems in the maritime operation of USVs to generate way-points for a safe navigation within a desired environment from start to end point. Research and development in areas of artificial intelligence has provided larger scope for development in this territory of marine navigation (Campbell et al., 2012). The abstraction of path planning for an USV is summarized in figure 2.

Until now in path planning of an USV, global and local approaches have been adopted, which has been summarised in figure 3. In global approaches, the complete information of environment is well known while in the local approach only partial information about the environment is known. Under global approaches, grid map-based path planning techniques are the best known since they generate sub optimal trajectories with the fastest computation time

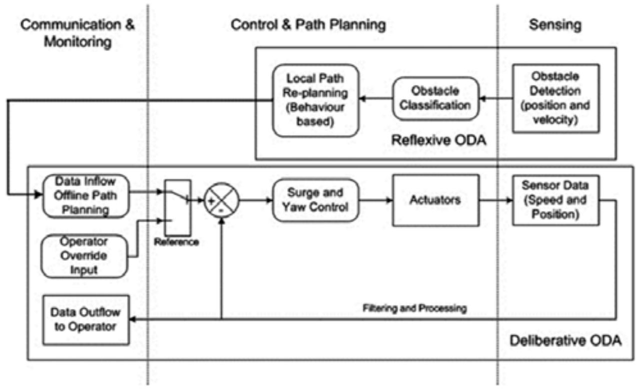

Figure 1. General architecture of USV operation in a maritime environment (Campbell et al., 2012)

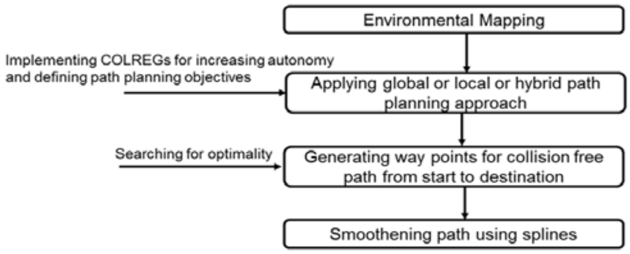

Figure 2. Path planning abstraction for USVs

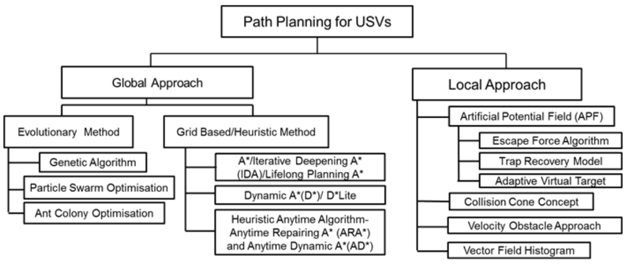

Figure 3. Path planning approaches for USV

1.1 *Literature review*

Dijkstra (1959) initiated the work in the area of grid map-based path planning algorithm by describing the shortest path between two nodes specified on a map. This was later improved by Hart et al. (1968) who introduced A^*, which is an extended version of Dijkstra algorithm. In the last two decade, many variants of A^* have been introduced by various researchers to improve the performance of robots working in various environments. Stentz (1995) introduced the first major improvement of A^*, focused D^* algorithm for real time path replanning which was later improved for partially unknown environment by induction of D^* Lite (Koenig and Likhachev, 2002). Another improvement by fixing infelicities of A^* in a dynamic environment was introduced by Likhachev et al. (2005) through Anytime Dynamic A^*. Since these algorithms do not consider the heading and dynamics of a robot in account, another major improvement was introduced in the form of Theta* (Nash et al., 2007). This algorithm accounts heading angle and yaw rate of a robot in the path planning, which is a necessity for USV path planning since it cannot follow an unrealistic path with sharp turns (Kruger et al., 2007; Prasanth Kumar et al., 2005; Yang et al., 2011). Advanced approaches like the ant colony algorithm (ACO) (Song, 2014) and particle swarm optimization (PSO) (Song et al., 2015) have been adopted for USV navigation but cannot generate trajectories in real time due to high computational load. Along with this, these algorithms do not give consideration to vehicle dynamics and turning radius.

In robotics, various local path planning approaches such as Collision Cone Concept (Chakravarthy and Ghose ,1998), Velocity Obstacle Approach (Fiorini and Shiller ,1998), Vector Field Histogram (Borenstein and Koren ,1991), and APF (Khatib, 1986), have been proposed. As most of the robotics problem is real time, the need to have a very fast and simple motion planner is evident. The simplicity enables fast development and deployment of a robot, whereas the computationally inexpensive nature allows the algorithm to be implemented in robots with minimum sensing capabilities. APF is one of the simplest methods, and the method is capable of autonomously moving a robot in realistic obstacle framework.

After APF was introduced by Khatib (1986), many researchers have attempted to improve the APF, which suffers from trap situation in local minima, oscillations in narrow passage and goals non-reachable with obstacles nearby (GNRON) (Koren and Borenstein, 1991). Ge and Cui (2002) included velocity terms for target and obstacles within APF to compute potential to correct the problem of GNRON. Baxter et al. (2007, 2009) used APF for multiple robots in order to correct the sensor errors. Tu and Baltes (2006) used a fuzzy approach within APF to solve the problem of oscillations within narrow passage. Fahimi et al. (2009) used the concept of fluid dynamics within APF to correct the issue of a trapped situation in local minima.

Until now in the literature, very few studies associated with the path planning of USV have made use of the APF in a practical marine environment. Most of these studies have been conducted in self-simulated environment. The present paper makes an effort to understand the effectiveness of APF in path planning of USV in a practical marine environment.

1.2 *Major contribution*

Many studies in marine navigation have been conducted but most of them have been related to collision avoidance rather than the path planning problem (Tam et al., 2009). Even the studies conducted on optimal USV navigation have been struggling with the high computational load and are inapplicable in generating trajectory in real time. Until now in the literature, path planning approaches have been applied on a self-simulated Euclidean $SE(2)$ grid map with no consideration to real time environment. This study presents the use of the Dijkstra algorithm in a real time environment with minimum computational load to generate a trajectory within a real time operation. This approach is well suited for optimal USV navigation in a static environment with minimum computational requirement. In order to benchmark the present study, a well-known local path planning approach

APF has been chosen for USV path planning in a static environment and its effectiveness is measured in terms in terms of path length and computational time.

The paper has been organized in four sections. The section after the introductory material comprising of literature review and major contribution explains the methodology and the Dijkstra algorithm used for the study. The third section deals with implementation of Dijkstra algorithm in a real time marine environment and explains its effectiveness. The fourth section explains the concept and implementation of APF in a real time marine environment with results. The final section discusses the results and provides conclusions with recommendations towards future work.

2 METHODOLOGY

2.1 *Dijkstra Algorithm*

There are various variants of the Dijkstra algorithm. The variant used in this study fixes a source node which is the start point of the USV and finds the shortest paths from source node to all other nodes in the graph leading to shortest- path tree. In order to reduce the computational load in the original variant, a sparse graph i.e. graph with fewer edges approach has been adopted leading to more efficient storage of graph nodes. The algorithm is defined in Algorithm 1 (Ahuja, 1990).

Algorithm 1. Dijkstra(Graph, source)

function Dijkstra(*Graph*, *source*):
create vertex set Q
for each vertex *v* in *Graph*: // *Initialization*
dist[*v*] ← INFINITY // *Unknown distance from source to v*
prev[*v*] ← UNDEFINED // *Previous node in optimal path from source*
add *v* to *Q* // *All nodes initially in Q (unvisited nodes)*
dist[*source*] ← 0 // *Distance from source to source*
while *Q* is not empty:
u ← vertex in *Q* with min dist[u] // *Node with the least distance will be selected first*
remove *u* from *Q*
for each neighbour *v* of *u*: // *v is still in Q.*
alt ← dist[*u*] + length(*u*, *v*)
if *alt* < dist[*v*]: // *A shorter path to v has been found*
dist[*v*] ← *alt*
prev[*v*] ← *u*
return dist[], prev[]

2.2 *Environmental mapping*

Environmental mapping is the first step in the abstraction of path planning as shown in figure 2. In order to use a practical environment, Portsmouth harbor has been considered as shown in figure 4. The map is organised as a weighted occupancy map using a cell decomposition method (Latombe, 1991).

This map represents obstacles as black and free space as white in a matrix of black and white as shown in figure 5. A 800×800 pixel map size has been used for the simulation with a resolution of 3.6 m/pixel.

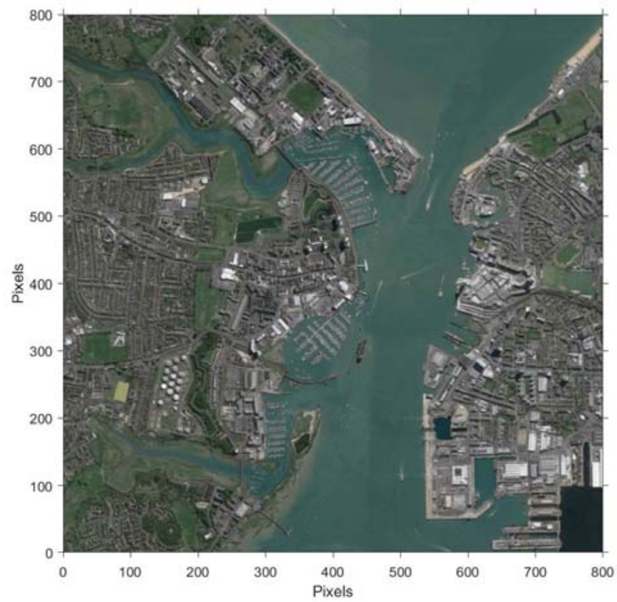

Figure 4. Aerial view of the simulation area (Source: Google Maps

Figure 5. Grid map of the simulation area

Figure 6. The *Springer* USV

Table 1. Specifications of Springer

Configuration	Values
Length (m)	4.2
Width (m)	2.3
Displacement (tonnes)	0.6
Maximum speed (m/s)	4

3 SIMULATION

The proposed simulation was executed using MATLAB 2015a on Intel i5 2.80GHz quad core with a 16 GB RAM. In these simulations, computational time of the simulation for three different start nodes and a fixed goal node have been compared in order to determine the effectiveness of the algorithm in terms of computational time to find an optimal trajectory in a practical marine environment. The simulations are assumed to be used by *Springer*, a USV available with Plymouth University whose specifications have been given in Table 1. Figure 6 shows the *Springer* USV. Figure 7 shows the three cases of three different start nodes within the grid map having a fixed goal node. These starting nodes are chosen arbitrarily within grid map on different positions within the simulation area to show the effectiveness of the algorithm in finding different trajectories with least computational load.

Table 2. Performance analysis for three cases in terms of computational time

Cases	Computational Time (s)
Case 1 (Figure 6(a))	6.801
Case 2 (Figure 6(b))	5.579
Case 3 (Figure 6(c))	6.141

Table 2 shows the comparison of computational time for three cases as shown in figure 7. The results show that the trajectories generated by the Dijkstra algorithm within a huge grid map from any source nodes satisfy the computational efficiency. All cases are able to generate a complete path in less than 7 seconds which in turns lead to the generation of path in less than 1 second per metre length of the distance covered by USV. Henceforth, such an algorithm is applicable in a real time operation where faster optimal trajectories are needed to be generated.

Since the maximum speed of the USV for which the algorithm is designed is 4 m/s, henceforth, the proposed approach satisfies the dynamic constraints of the platform. Although various factors such as vehicle dynamics and heading angle have not been considered in the approach, the basic objective of the study towards generation of trajectory with minimum computational load has been accomplished.

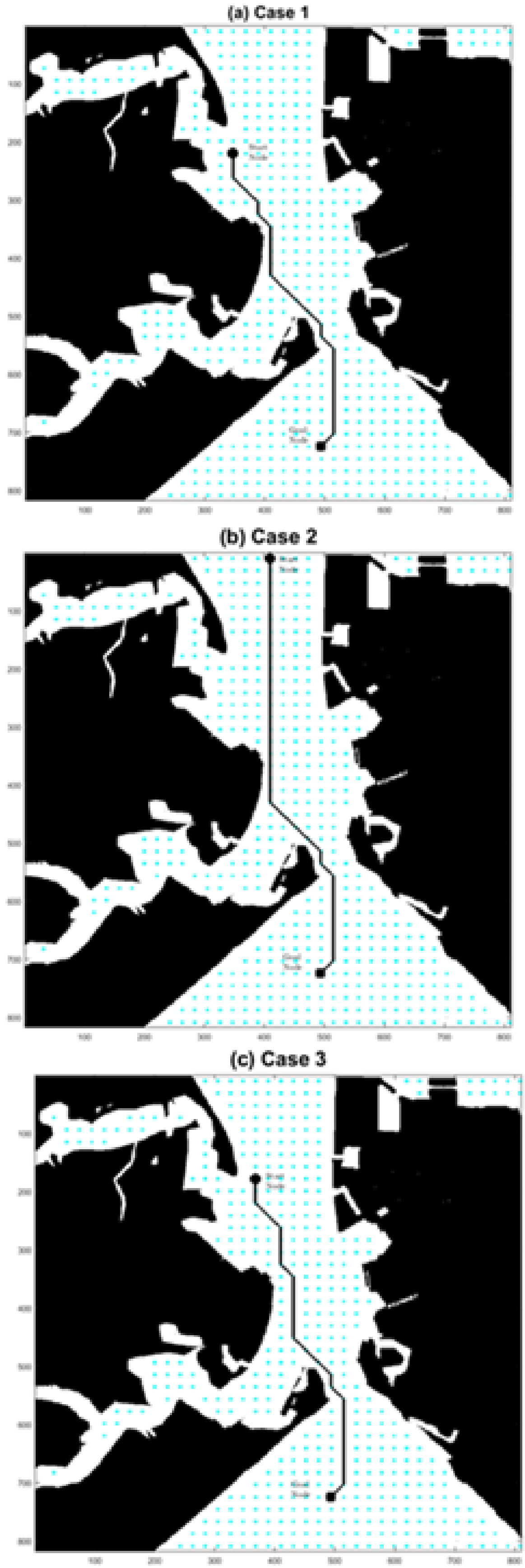

Figure 7. Simulation results for three cases

4 APF: CONCEPT AND METHODOLOGY

APF solves the problem assuming all obstacles are a source of repulsive potential, with the potential inversely proportional to the distance of a robot from the obstacle while the goal attracts it by applying an attractive potential, Kala (2016). The derivative of the potential gives the value of the virtual force applied on the robot, based on its movement, Kala (2016). The motion is completely reactive in nature. A schematic of the APF is shown in figure 8.

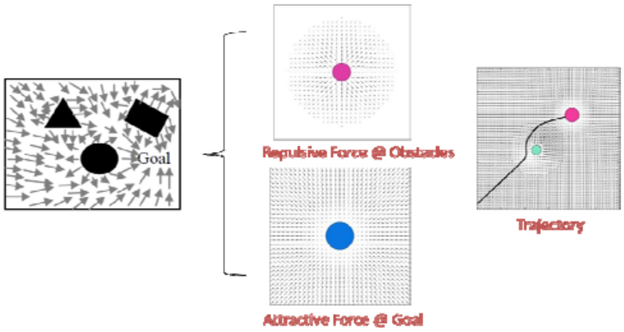

Figure 8. Schematic of the APF

4.1 *Attractive Potential*

The attractive potential is applied by a single goal to direct the robot towards itself. The attractive potential is directly proportional to the distance between the current position of the robot and the goal. This causes the potential to tend to zero as the robot approaches the goal and hence it slows down as it approaches the goal (Kala, 2016). The potential in this study is taken as, quadratic potential, represented in Equation (1)

$$U_{att}(x) = \frac{1}{2} k_{att} \|x - G\|^2 \tag{1}$$

where x is the current position of the robot and G is the goal. $\|.\|$ is the Euclidean distance function and k_{att} is the proportionality constant, whereas the degree is taken as 2.

The driving force is a vector whose magnitude is measured through the derivative of the potential function and direction as the line which maximizes the change in potential, which is given by Equation (2)

$$\begin{aligned} F_{att}(x) &= \nabla U_{att}(x) = k_{att} \|x - G\| . u(x - G) \\ &= k_{att} \|x - G\| \frac{(x - G)}{\|x - G\|} \\ &= k_{att}(x - G) \end{aligned} \tag{2}$$

u() is the unit vector.

4.2 *Repulsive Potential*

The repulsive potential is applied by obstacles which repel the robot coming close and repelling it to avoid collision. The potential is inversely proportional to the distance so that potential tends to infinity if robot comes near obstacle leading to repulsion. Obstacles at a certain distance d^* are considered in modeling the potential (Kala, 2016).

The repulsive potential is given by Equation (3).

$$U_{rep}(x) = \begin{cases} \frac{1}{2} k_{rep} \left(\frac{1}{\|x - o_i\|} - \frac{1}{d^*} \right)^2 & if \ \|x - o_i\| > d^* \\ 0 & if \ \|x - o_i\| \le d^* \end{cases} \tag{3}$$

where, x is the current distance of the robot and o_i is the position of the obstacle. $\|.\|$ is the Euclidian distance function and k_{rep} is the proportionality constant, whereas the degree is taken as 2.

The repulsive force is given by Equation (4), which is a derivative of the repulsive potential

$$\begin{aligned} F_{rep}(x) &= \nabla U_{rep}(x) \\ &= -k_{rep} \left(\frac{1}{\|x - o_i\|} - \frac{1}{d^*} \right) \frac{1}{\|x - o_i\|^2} u(x - o_i) \\ &= -k_{rep} \left(\frac{1}{\|x - o_i\|} - \frac{1}{d^*} \right) \frac{1}{\|x - o_i\|^2} \frac{(x - o_i)}{\|x - o_i\|} \\ &= -k_{rep} \left(\frac{1}{\|x - o_i\|} - \frac{1}{d^*} \right) \frac{(x - o_i)}{\|x - o_i\|^3} \end{aligned} \tag{4}$$

4.3 *Resultant Potential*

The resultant potential is given by sum of attractive and repulsive potential. This final force is henceforth, the derivative of the resultant potential. This is given in Equation (5).

$$\begin{aligned} U &= U_{att} + U_{rep} \\ F &= \nabla U = \nabla U_{att} + \nabla U_{rep} = F_{att} + F_{rep} \end{aligned} \tag{5}$$

4.4 *Methodology*

In the present study, APF is used for USV navigation within a practical marine environment i.e. Portsmouth Harbour having a start and goal point as shown in figure 9.

A binary map of 800 x 800 pixel grid resolution, figure 10, is taken into account with a USV available from Plymouth University named, *Springer*, shown in figure 5 being considered in terms of kinematic constraints for the purpose of path planning. Parameters used in APF for path planning of *Springer* are shown in Table 3.

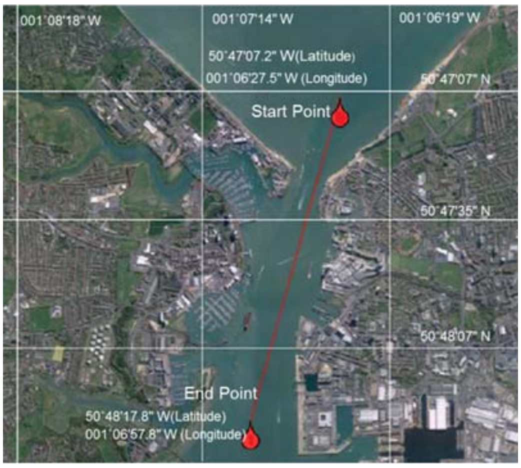

Figure 9. Simulation area- Portsmouth Harbour (Source: Google Maps)

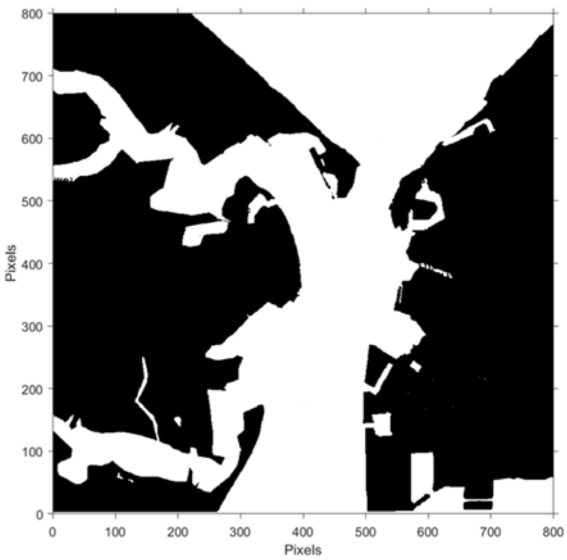

Figure 10. Binary map of the simulation area (1 Pixel = 3.6 m)

Table 3. Parameters used in APF for path planning of *Springer*

Parameters	*Values*
Attractive Potential Scaling Factor (k_{att})	300000
Repulsive Potential Scaling Factor (k_{rep})	300000
Safety Distance from Obstacles (d^*)	30 pixels
Maximum Turn Rate	10 pi/180°
Initial Heading of USV	-pi/2

Evaluation of the APF performance for USV path planning in terms of path length and computational time is described in Table 4. Simulation records movement sequences of the USV within map. Figure 11 shows the sequence of USV motion from start to goal point at different time of the motion. The overall trajectory shows that such algorithm is efficient in generating safe path for USV in a practical marine environment.

Table 4 shows that USV is able to find a safe trajectory of length 3075 m within 32.608 s which means, less than 1 s is required by USV to find a path of 1m. Thus real time implementation of such algorithm is possible within a practical marine environment. Since the APF is a parameter dependent algorithm, there is a need to find right set of parameters for different case scenarios. In addition to this, such algorithm is not complete (i.e. guarantees finding a path in all scenarios) and is more intensive computationally than global path planners to be used in marine robots which have limited on board capability.

Table 4. Performance of APF in *Springer* navigation

Parameters	Value
Path Length	3075 m
CPU Time	32.608 s

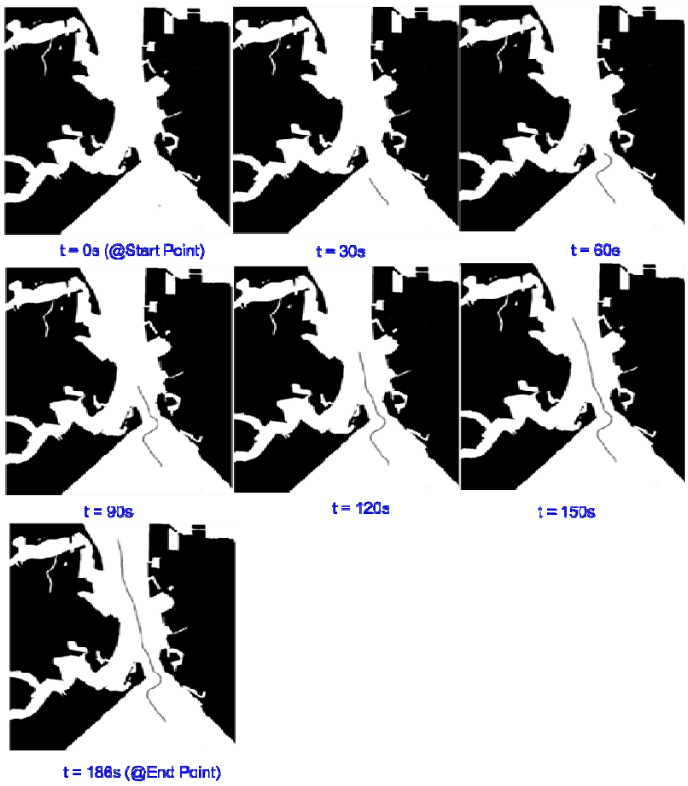

Figure 11. Sequence of USV motion from start to end point

5 CONCLUSION

In this paper, a computationally efficient Dijkstra algorithm to find a path between a source and goal node on a grid map is proposed. The performance was measured in terms of computational time for three different cases, where source points where chosen arbitrarily. The results show that the proposed approach satisfies the computational requirement of the path planning in a real time environment. In order to benchmark the study, a well-known local path planning algorithm APF was also studied and results were presented. Dijkstra algorithm was found more effective in terms of finding path optimally and computationally. In conclusion, this new approach is suitable for global path planning of an USV in a static environment. Towards future work, vehicle dynamics and environmental disturbances can be included in the grid map to better understand the applicability of this approach in a dynamic environment.

6 ACKNOWLEDGEMENTS

This research is supported by the doctoral grant of the Commonwealth Scholarship Commission, United Kingdom tenable for doctoral studies in the School of Engineering at Plymouth University.

REFERENCES

Ahuja, R.K.., Mehlhorn, K., Orlin, J.B., and Tarjan, R.E. 1990. Faster Algorithms for the Shortest Path Problem, *Journal of Association for Computing Machinery (ACM)*, 37 (2):213–223.

Baxter, J.L., Burke, E.K., Garibaldi, J.M., and Norman, M. 2007. Multi-robot search and rescue: A potential field based approach, *Autonomous Robots and Agents, Springer*, 9-16.

Baxter, J.L., Burke, E.K., Garibaldi, J.M., and Norman, M. 2009. Shared potential fields and their place in a multi-robot co-ordination taxonomy, *Robotics and Autonomous Systems*, 57(10):1048-1055.

Borenstein, J., and Koren, Y. 1991. The vector field histogram-fast obstacle avoidance for mobile robots, *IEEE Trans. Robotics and Automation*, 7:278-288.

Campbell, S., Naeem, W. and Irwin, G.W. 2012. A review on improving the autonomy of unmanned surface vehicles through intelligent collision avoidance manoeuvres, *Annual Reviews in Control*, 36: 267–83.

Chakravarthy, A., and Ghose, D. 1998. Obstacle avoidance in a Dynamic Environment: A Collision Cone Approach, *IEEE Trans. Systems, Man and Cybernetics, Part A: Systems and Humans*, 28:562-74.

Dijkstra, E. 1959. A note on two problems in connexion with graphs, *Numer. Math.* 1, 269-271.

El-Hawary, F. 2000. The Ocean Engineering Handbook, *CRC Press*, 1-416.

Fahimi, F., Nataraj, C., and Ashrafiuon, H. 2009. Real-time obstacle avoidance for multiple mobile robots, *Robotica*, 27(2): 189-198.

Fiorini, P., and Shiller, Z., 1998. Motion planning in dynamic environments using velocity obstacles, *Int. J. Robotics Research*, 17:760-772.

Ge, S.S., and Cui, Y.J., 2002. Dynamic motion planning for mobile robots using potential field method, *Autonomous Robots*, 13(3):207-222.

Hart, P.E., Nilsson, N.J., and Rapheal, B. 1968. A formal basis for the heuristic determination of minimum cost paths, *IEEE Transactions on System and Scientific Cybernetics* 4, 100-107.

Kala, R. 2016. Potential-Based Planning, *On-Road Intelligent Vehicles*, Ch.11:318-356.

Khatib, O. 1986. Real time obstacle avoidance for manipulators and mobile robots, *Int. J. Robotic Research*, 5:90-98.

Koenig, S., and Likhachev, M. 2002. D* Lite, *AAAI/IAAI*, 476-483.

Kruger, D, Stolkin, R., Blum, A., and Briganti, J. 2007. Optimal AUV path planning for extended missions in complex, fast-flowing estuarine environments, *In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 4265-4271.

Latombe, J. 1991. Robot Motion Planning, *Klumer Academic Publishers*, Norwell, USA.

Legrand, J., Alfonso, M., Bozzano, R., Goasguen, G., Lindh, H., Ribotti, A., Rodrigues, I., Tziavos, C. 2003. Monitoring the Marine environment: Operational Practices in Europe, *In Third International Conference on Euro GOOS*.

Likhachev, M., Ferguson, D.I., Gordon, G.J., Stentz, A. and Thrun, S. 2005. Anytime Dynamic A*: An Anytime, Replanning Algorithm, *ICAPS*, 262-271.

Nash, A., Daniel, K.., Koenig, S. And Felner, A. 2007. Theta*: any-angle path planning on grids, *In Proceedings of the National Conference on Artificial Intelligence*, 1177-1183.

Prasanth-Kumar, R., Dasgupta, A., Kumar, C. 2005. Real-time optimal motion planning for autonomous underwater vehicles, *Ocean Engineering* 32, 4265-4271.

Serreze, M.C. 2008. Arctic sea ice in 2008: standing on the threshold, *In MTS/IEEE OCEANS'08 Conference.*

Song, C.H. 2014. Global Path Planning Method for USV System Based on Improved Ant Colony Algorithm, *Applied Mechanics and Materials*, 568-570.

Song, L., Mao, Y., Xian, Z., Zhou, Y., and Du, K. 2015. A Study on Path Planning Algorithms Based upon Particle Swarm Optimization, *Journal of Information and Computational Science* 12, 673-680.

Stentz, A. 1995. The Focussed D* Algorithm for Real-Time Replanning, *IJCAI* 95, 1652-1659.

Tam, C., Bucknall, R., and Greig, A. 2009. Review of collision avoidance and path planning methods for ships in close range encounters, *Journal of Navigation* 59, 27-42.

Tu, K., and Baltes, J., 2006. Fuzzy potential energy for a map approach to robot navigation, *J. Robotics and Autonomous Systems*, 54(7):574-589.

Yang, Y., Wang, S., Wu, Z., and Wang, Y. 2011. Motion planning for multi-HUG formation in an environment with obstacles, *Ocean Engineering* 38, 2262-2269.