AMBIENTE VIRTUAL PARA AGENTES ROBÓTICOS COLABORATIVOS.

Juan Sebastian Bareño Agudelo

Trabajo de grado presentado como requisito para optar al título de:

Ingeniero Mecatrónico

Director:

Robinson Jiménez Moreno

UNIVERSIDAD MILITAR NUEVA GRANADA

FACULTAD DE INGENIERÍA

PROGRAMA MECATRONICA

BOGOTÁ, 18 DE ABRIL 2018

Contemporary Engineering Sciences, Vol. 10, 2017, no. 28, 1365 - 1376
HIKARI Ltd, www.m-hikari.com
https://doi.org/10.12988/ces.2017.711163

Virtual Environment for Collaborative Robotic Agents

Juan Sebastián Bareño Agudelo, Robinson Jiménez Moreno and Ruben Dario Hernández

Davinci Research Group, Mechatronics Department
Faculty of Engineering, Universidad Militar Nueva Granada, Carrera 11# 101-80
Bogotá, Colombia

Abstract

This article describes the development of a virtual environment in the VRML platform, used to simulate the selection, classification and storage of wares that has a destination to people in a specific location around the world. This environment is based in two rooms, the first room or delivery room, where the package is received and classified by place of shipment, in turn is moved to the second room or warehouse, where all the merchandise that arrives is stored, being classified by destination. As methodology of the project, the idea is, by means of an algorithm, to classify the merchandise by destination, with this, it is achieved that it does not find lost merchandise or take the merchandise to a wrong destination.

Keywords: virtual environment, simulation, classification, VRML, platform

1 Introduction

Currently, virtual environments are a fundamental tool in engineering since it helps to set the behavior of artificial intelligence designs and algorithms, seeking to find efficient, economic and accurate solutions. These tools help the designer and engineer to establish the best solutions to a given problem. However, there are virtual applications focused on training as presented in [1], which presents a collaborative virtual environment for the training of security agents in large events, this system allows studying the reactions to threats related to suspicious materials. In [2] a work focused on the development of a collaborative hybrid virtual environment is presented, which allows the interaction of two users with a cube object, using operations such as scaling, rotate and translate, allowing each user to interact with the six degrees of freedom of the object. Similarly, [3] presents the development of a collaborative virtual reality environment for training and travel on a university campus, which presents three ways to control the behavior of the crowd by means of rules of robotic agents simulated by computer, which provide the controls to navigate within the environment as autonomous agents and generating responses in real time.

Recent applications of the use of robotic agents with virtual environments are presented in [4], where it is developed a virtual environment of medical surgery with the interaction of the da Vinci robot, analyzing the benchmarks of the performance of each surgeon allowing an evaluation of the techniques used. In this way, the implementation of a virtual system depends on the mechanical system, the conditioning of the robotic prototype and the artificial intelligence algorithm (robotic agent) developed for the user's interaction with the machine [5] [6]. Nowadays, interactive virtual worlds provide a means of experimental learning and entertainment, since they incorporate bodily virtual agents similar to humans and with different degrees of intelligence, obtaining what today are called Intelligent Virtual Agents (IVA). Since the collaboration between agents can be important in relation to what it is wanted to know about the environment of each agent using awareness models such as CSCW (computer-supported cooperative work) [7].

In areas, such as autonomous vehicles, there are works such as [8] in which an algorithm is developed to generate routes in a given space, which finds the best solution so that the vehicle can move from a point A to a point B without losing the planned trajectory and without hitting possible static obstacles that are in the workspace. Similarly, in [9], the design of a virtual environment that allows the simulation of freight transport is shown, where a package is received and classified according to the destination location. There, the classification algorithm is based on the RRT (Rapidly exploring random tree) method, which is designed to efficiently search for convex spaces and, by means of a random construction, samples of possible trajectories are created to travel effectively [10].

Nowadays, in most processes that involve the classification and storage of merchandise, it is necessary the intervention of the operator who can be in charge of the manipulation of tools such as forklifts or similar, so this requires a fast and efficient process to avoid problems of loss of goods, damage, deterioration or delays in delivery [11] [12]. This is why the need to automate this type of processes is generated so that the manufacturer can guarantee the delivery and distribution of their product. Hence, the developments of virtual test environments become relevant, since they facilitate the analysis of joint workspace, the validation of the trajectories of the collaborative robotic agents, or generation of test times or adjustment of the development algorithms of the plant.

For the reasons previously explained, the present paper proposes the development and implementation of a virtual interface that allows the programming of robotic agents to validate the algorithms of trajectory planning for the development of joint works in a controlled environment. This paper is divided in four parts. The first section shows the relevant aspects to the development of virtual environments and robotic agents, followed by section two in which the virtual work environment and the algorithm developed for the storage of merchandise are described. In section three, the results obtained are presented and finally the conclusions reached in the development of the proposed research.

2 Virtual Environment for Selection and Storage of Merchandise

Collaborative work of two robotic agents, where the first robot must receive the merchandise of an operator who must indicate its destination, through a room. Once this trajectory is finished, the agent must deliver the merchandise to a second robot, where it must store the merchandise in the company's warehouse according to the instructions of the operator, since it must be classified according to its destination. In this work, different scenarios are presented where it is possible to observe the collaborative work of two robotic agents as their individual performance performing the complete task from the moment they receive the merchandise until it is deposited in the warehouse. In the same way, it can be observed how the agents avoid the collision both with rigid objects and with moving objects, where the complete work of classification of 10 types of merchandise (boxes) will also be displayed, which must be stored in different positions of the warehouse for later transportation and delivery.

Using work tools such as programming software and graphic design, among which is Matlab®, where the entire trajectory algorithm was programmed and the results were visualized, SolidWorks® that allowed designing both the robotic agents and the objects that will be immersed in the rooms, and V-Realm Builder, where a virtual environment was created with two rooms of a shipping company where the first is the receipt of merchandise and the second the warehouse where it would be stored.

As a first step, it must have a floor plan view of the room to proceed to create an aerial image of it, which in turn must be passed to the black and white scale in order to show what objects are inside it.

Figure 1 shows the method of capturing the room in which the path will be calculated so that the objects that are in it are highlighted in such a way that the algorithm can have them as a reference when calculating the route. Since the project consists of two rooms, each of them must be captured, and the starting and arrival point of each robotic agent must be taken into account.

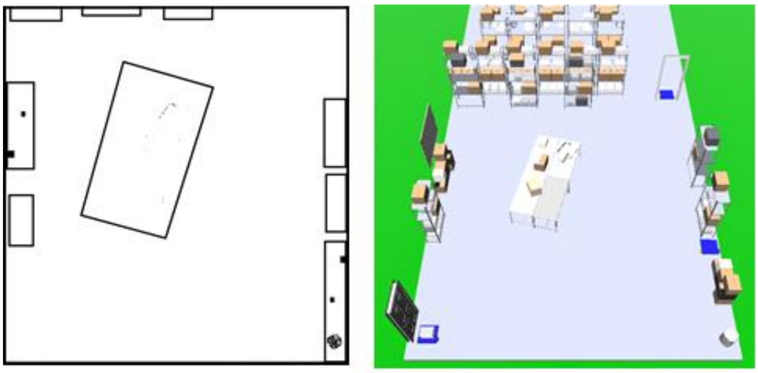

Figure 1. Aerial capture of the room.

With the starting point and arrival point of the robotic agents, it is proceeded to their respective solution by the RRT method, as shown next, the starting points of each agent and a respective solution of the trajectory that must be followed so that it travels through the room avoiding collisions or losses in the trajectory are observed.

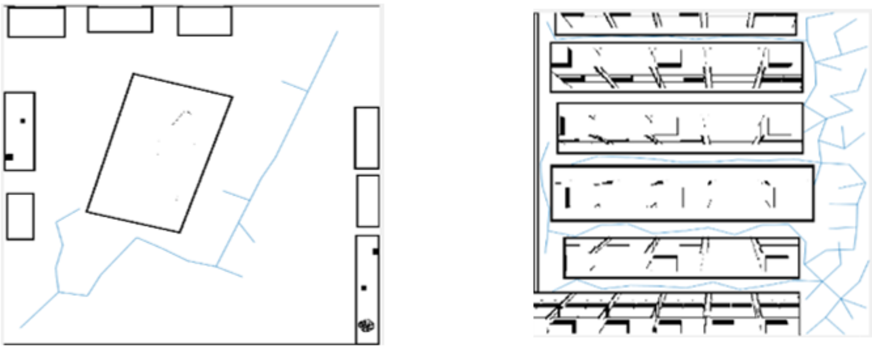

Figure 2. Possible solutions of the respective rooms.

After having filtered the erroneous solutions of Figure 2, it can be seen that in Figure 3 said trajectory is represented in a matrix of points, which will be used for the robotic agent to do the trajectory in the virtual environment.
The path construction procedure is based on the growth of a tree, using random samples within the search space. As each sample is drawn, it is verified whether it is feasible to be part of the solution, this is because the sample does not cross or arrive at a black section. As a result, a new state is added to the tree in such a way that the algorithm finds the solution within the constraints presented by the image, which in this case is the aerial capture of the working rooms of the robotic agents [12].

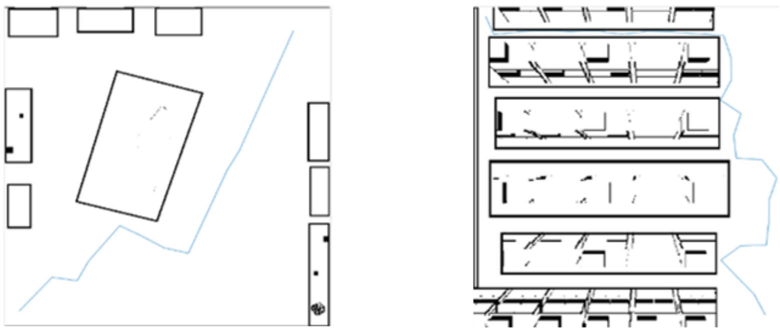

Figure 3. Solutions of the trajectories.

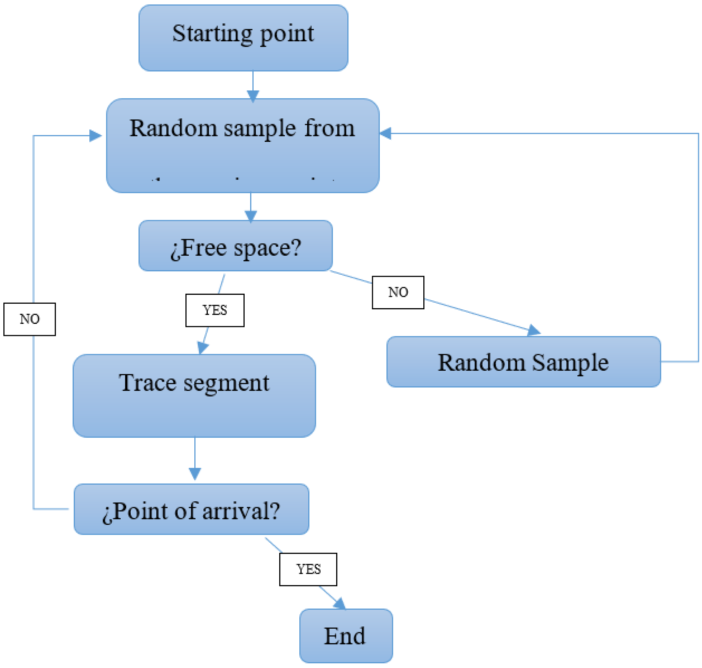

Figure 4. Basic model of the algorithm.

In Figure 4 it can be seen how the algorithm works. Random lines are generated in such a way that it creates the route from the point of start until its arrival, counting the obstacles as black points inside the previously loaded image [13].
Once the solution has been acquired, the algorithm will take the trajectory and the delivery by means of a matrix of points, where in the rows it will be possible to observe the points in X and in Y that the robotic agent must travel to perform the displacement.
This method is based on traversing the matrix by rows in ascending order, the first two rows are selected and the starting angle of the robotic agent is calculated to start its trajectory. For this, the slope of the segment traced by the points in X and in Y of the selected rows must be calculated.

With the calculated slope, it is proceeded to determine the inverse tangent in degrees, to obtain the angle of inclination of the segment. The 'atand ()' command that calculates the inverse tangent of the segment does not differentiate between the quadrants of the Cartesian plane, so conditionals were created that indicate in which quadrant the segment is located. For this, the sign must be taken into account in the subtraction of the points of the selected rows of the matrix both in X and in Y and in its slope m, once the angle is obtained, the conversion is done from degree to radians since the virtual environment works with the angles in radians.

$$m = \frac{(Y_{+1} - Y)}{(X_{+1} - X)} \tag{1}$$

$$AG = atand(m) \tag{2}$$

$$AR = degtorad(AG) \tag{3}$$

Where, m is the calculated slope, AG, at the angle of the slope in degrees, AR, the angle AG in radians.
For the movement of the robotic agent, it is done the same procedure of taking two rows in a row in the order that the matrix takes and subtract the points, respectively. This calculation is done for both X-axis and Y-axis. It is fragmented the subtraction of the selected points of the rows in a base n, where n is selected by the user under the criterion of how realistic the trajectory is, to more points, the real path of the robotic agent will be (the recommended value is 40). The trajectory is executed starting at points X and Y of the first of the two selected rows, growing in (X+1 – X)/n up to X+1 for the two axes. This calculation yields a matrix of 2xn, which is used for the robotic agent to move through the virtual environment, traversing the rows of the matrix previously calculated in ascending order.

Table I. Matrix of points

X	Y
A	B
C	D

$$for\ RX = A : \frac{(C - A)}{n} : C \tag{4}$$

$$for\ RY = B : \frac{(D - B)}{n} : D \tag{5}$$

RX is the matrix of points that indicates the path of the robotic agent in X and RY, the path of the robotic agent in Y, the results of the routes are stored in a vector for both X and Y, are traveled in parallel within a for loop, so that the trajectory of the agent is ideal.

3 Analysis of Results

As an analysis of the project, it is proceeded to execute the algorithm so that the robotic agents go through the rooms avoiding the objects that appear in each of them. In this way, it is then observed how, from the rooms in Figure 5, the aerial capture must be done, to identify the obstacles and allow the algorithm to select them so that based on their position in the rooms, the robotic agent can perform its displacements.

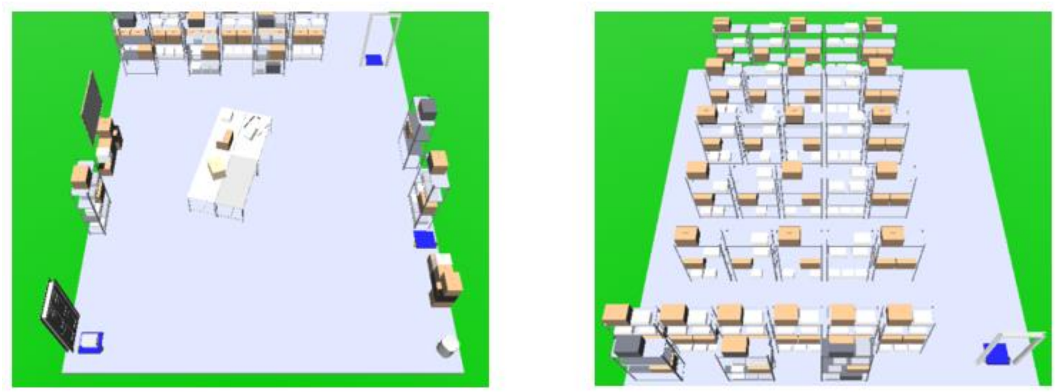

Figure 5. Virtual Environment of Rooms 1 and 2.

The screenshots of rooms 1 and 2 taken from an aerial view are:

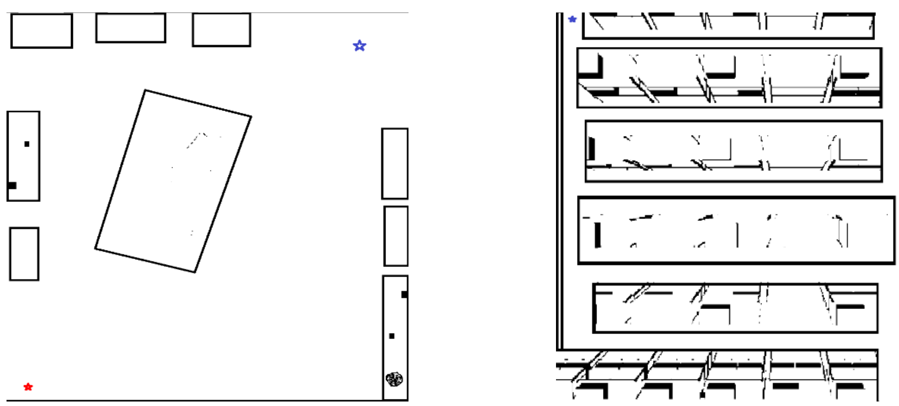

Figure 6. Aerial capture of the rooms.

As shown in Figure 6, it can be seen red stars that indicate the starting points of the robotic agents and blue stars that indicate the arrival points of each robotic agent, in the same way, the time in which the algorithm to find a trajectory was taken. There were done 10 tests to find an average time.

Table II. Path calculation times.

	Room 1, Time (s)	Room 2, Time (s)
1	0.83828	9.47386
2	0.96377	2.54407
3	1.13656	3.16544
4	1.11192	2.12234
5	0.95960	4.44894
6	0.86872	2.00061
7	1.01809	2.57647
8	0.95689	1.86987
9	0.89336	3.17336
10	1.16097	6.01721
	Average	
	0.99081632	3.7392173

As the name implies, Rapidly Random Tree, the algorithm is based on random samples, this indicates that it will never follow a pattern which indicates what route it will take or what the trajectory will be like, but that each time the program is run, it will calculate a totally different trajectory, taking as a priority to reach its destination, regardless of whether it is the most optimal route or not. Given this, it can be seen that the times in which it performs the trajectory for each room obey a totally different pattern since in the first room the times handle a difference no greater than 0.2s while the times in room 2 are quite irregular, this is because the trajectory is much more complex to calculate due to the distributions of the objects inside the room.

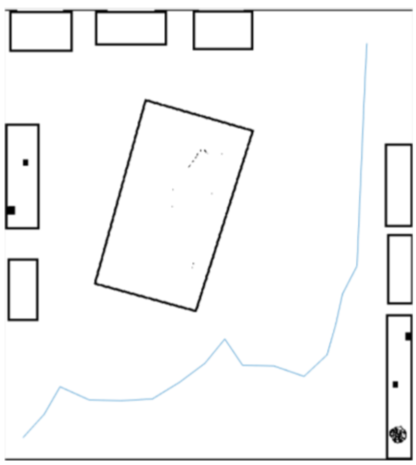

Figure 7. Path for room 1.

As illustrated in Figure 8, there is a possible solution of the trajectory. It can be observed how it avoids collision with the objectives of the room, in such a way that it can reach its destination successfully.

Table III. Matrix of points of the trajectory and its angles.

Trajectory No. 1		Angles No. 1	
X	Y	Degrees	Radians
0.3	0	49.4506	0.70752054
0.77	-0.52	61.7133984	0.49349433
1.12	-1.13	-22.8171382	1.96883051
1.76	-0.84	-0.07724666	1.57194454
2.46	-0.82	4.82981831	1.4863001
3.16	-0.86	33.2200815	0.99079653
3.76	-1.23	38.429228	0.89987977
4.32	-1.65	52.3856724	0.65629386
4.76	-2.19	-54.5232391	2.52220637
5.15	-1.61	-0.07724666	1.57194454
5.85	-1.59	-17.6533414	1.87870526
6.51	-1.36	44.8236258	0.78827647
7.02	-1.84	75.7269438	0.24891183
7.21	-2.51	79.1197684	0.18969587
7.36	-3.19	64.2597582	0.44905197
7.68	-3.81	89.1052987	0.01541548
7.71	-4.51	89.1052987	0.01541548
7.74	-5.21	89.1052987	0.01541548
7.77	-5.91	89.1052987	0.01541548
7.8	-6.61	89.1052987	0.01541548
7.83	-7.31	89.1052987	0.01541548
7.86	-8.01	88.4652723	0.02658605
7.9	-8.75		

The table III shows the matrices of points that indicate the trajectory that the robotic agent must travel in room 1 to reach the final point, as indicated in Figure 6, where the starting point and the point of arrival for room 1 can be observed.

Figure 8. Illustrations of the trajectory.

In Figure 8 it can be seen illustrations of how the robotic agents move in the rooms, from the points previously indicated in Figure 6, their movement and the delivery of the goods for storage.

4 Conclusions

The algorithm provides a path as a solution to travel through proposed rooms 1 and 2, complying with the specified delimitations, such as not crashing and reaching the final point, however, it does not mean that it is the best way to go through the rooms. As can be seen in the results of room 1, the generated trajectory that the robotic agent must travel is very random, determining that the algorithm, although it gives solution to the problem, does not do it in a very efficient way, since it does not calculate the smallest possible distance, in turn it does not travel in the shortest time. Thanks to the trajectory algorithm, it is enough to take an aerial capture of any room to determine the objectives that are in it and select the starting and arrival points of the robotic agent to obtain a trajectory that can be traversed through the room avoiding the recognized objectives.

Acknowledgements. The authors are grateful to the Nueva Granada Military University and the Davinci research group.

References

[1] C. Passos, M. H. da Silva, A. C. A. Mol and P. V. R. Carvalho, Design of a collaborative virtual environment for training security agents in big events, *Cognition, Technology and Work*, **19** (2017), 315–328. https://doi.org/10.1007/s10111-017-0407-5

[2] L. P. Soares, T. V. De Oliveira, V. A. Sangalli, M. S. Pinho and R. Kopper, Collaborative hybrid virtual environment, *2016 IEEE Symposium on 3D User Interfaces, 3DUI*, (2016), 283–284. https://doi.org/10.1109/3dui.2016.7460081

[3] S. Sharma, S. P. Rajeev and P. Devearux, An immersive collaborative virtual environment of a university campus for performing virtual campus evacuation drills and tours for campus safety, *2015 International Conference on Collaboration Technologies and Systems, CTS*, (2015), 84–89. https://doi.org/10.1109/cts.2015.7210404

[4] T. S. Lendvay, T. C. Brand, L. White, T. Kowalewski, S. Jonnadula, L. D. Mercer, D. Khorsand, J. Andros, B. Hannaford, R. M. Satava, Virtual reality robotic surgery warm-up improves task performance in a dry laboratory environment: A prospective randomized controlled study, *J. Am. Coll. Surg.*, **216** (2013), no. 6, 1181–1192. https://doi.org/10.1016/j.jamcollsurg.2013.02.012

[5] F. Ribeiro and A. M. Lopes, Haptic guidance in a collaborative Robotic system, Chapter in *Robotics in Smart Manufacturing*, Communications in Computer and Information Science, vol. 371, Springer-Verlag Berlin Heidelberg, 2013, 101–112. https://doi.org/10.1007/978-3-642-39223-8_10

[6] Y. Sung and K. Cho, Collaborative programming by demonstration in a virtual environment, *IEEE Intell. Syst.*, **27** (2012), no. 2, 14–17. https://doi.org/10.1109/mis.2012.28

[7] P. Herrero and A. de Antonio, A Human Based Perception Model for Cooperative Intelligent Virtual Agents, *On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE*, Lect. Notes Comput. Sci., vol. 2519, 2002, 195–212. https://doi.org/10.1007/3-540-36124-3_12

[8] S. M. LaValle, Rapidly-Exploring Random Trees: A New Tool for Path Planning, **129** (1998), 98–11.

[9] J.J. Kuffner, S.M. LaValle, RRT-Connect: An Efficient Approach to Single Query Path Planning, *Proceedings 2000 ICRA. Millennium Conference*. *IEEE International Conference on Robotics and Automation Symposia Proceedings (Cat. No.00CH37065)*, **2** (2000). https://doi.org/10.1109/robot.2000.844730

[10] S. M. LaValle and J. J. Kuffner, Rapidly-exploring random trees: Progress and prospects, *4th Work. Algorithmic Comput. Robot. New Dir*., (2000), 293–308.

[11] A. Konstantinidis, Th. Tsiatsos and A. Pomportsis, Collaborative virtual learning environments: Design and evaluation, *Multimed. Tools Appl*., **44** (2009), no. 2, 279–304. https://doi.org/10.1007/s11042-009-0289-5

[12] C. B. Moon and W. Chung, Kinodynamic planner dual-tree RRT (DT-RRT) for two-wheeled mobile robots using the rapidly exploring random tree, *IEEE Trans. Ind. Electron*., **62** (2015), no. 2, 1080–1090. https://doi.org/10.1109/tie.2014.2345351

[13] R. Kala, Code for Robot Path Planning using Rapidly-exploring Random Trees’, Indian Institute of Information Technology Allahabad, 2014, Available in: http://rkala.in

Received: November 17, 2017; Published: December 10, 2017