RESEARCH ARTICLE OPEN ACCESS

Predicting Movements On A Plane For Robot With Depth Camera

Jose Manuel Mejia Perales[1], Rodolfo Romero Herrera[2], Alfonso Fernandez Vazquez[3]
[1, 2, 3] (Seccion de Estudios de Posgrado e Investigación Escuela Superior de Computo Instituto Politécnico Nacional, México D.F.

ABSTRACT
This paper presents the results of location and prediction of movement made by an object. This probability is calculated for a matrix made of 5 x5 with the use of Markov processes. After some interactions the computer does not fail in any case. It is possible increase the size of the matrix but requires more computation and the results are repeated again, only changing the odds. To detect moving robots is captured by a Kinect depth camera the color of a ball (red), thereby calculate their coordinates in a plane. The robot receives the calculated probability and thus the robot's movements are programmed to catch the ball.
Keywords - Location, Markov, Kinect, Robot, tracking.

I. INTRODUCTION

Since 1948 until today, it is more common to find toys, household appliances, power tools, space explorers as the Phatfinder or Curiosity, which are derivatives of robotics and although I have more than a decade with us and are becoming more common, we continue to amaze, marking the transition from science fiction to reality.

The robot control by computer systems is done through various programming languages that allow the machines to adapt to different tasks, being an important factor in these languages the simplicity of operation. Robotic programming is explicit and may be of two types [1]:

Textual programming [1], [2], which is a series of instructions that tell the robot actions to be executed and the order must follow. Shares may be given by equations of motion and shock sensors supplemented or presence calculations and allowing fluid movements and perfect, and permits communication with the environment where it is located; this type of programming is ideal for precision tasks. The problem with textual programming is that it requires programming expertise even to correct a simple path for it.

Programming direct gestural [2]. It is generally used in the programming of robotic arms guide is directly tracing the path to the system later to repeat these movements. The programming problem gestural is the need for the robot to perform the programs, besides not being adaptable to the environment in real time.

The first correct mathematical construction of a Markov process with continuous paths was performed for N. Wiener in 1923. The general theory of Markov processes was developed by the decades of the 30s to 40's by A. N. Kolmogorov, W. Feller, W. Doeblin, P. Levi, J. L. Doob and others [1][2].

Markov analysis originated in studies of A. A. Markov (1906 – 1907), experiments on the chain sequence and attempts to mathematically describe physical phenomena known as Brownian motion, which can be compared with the daily activities of people and the many variables that can alter. Markov analysis is a way of analyzing the current movement of some variable, to predict the future movement of the same [3].

The fundamental characteristic of Markov chains is the probability of a studied system; it is in a particular condition that depends only on its current condition [4].

The reasons that were used for Markov chains are [5]:

- The adjustment between real problems and Markov chain models is sufficient to find relationships between variables that are useful to interpret the context.
- The underlying mathematical basis is well developed and allows the numerical solution of individual problems.

Almost any mobile device currently covers the computational requirements needed for its implementation.

II. METHODOLOGY

T This experiment, consisting in "a cooperative robot control", that allows a robot or more to play football in the goalkeeper position or other positioning using as input a matrix of transitions, this matrix contains information of where a ball has been, and can generate routes and also use this information to predict future position where he might be.

Input. The way to acquire information from the environment in this case was using a Kinect sensor, you wanted to know the position of a ball within the visual field, and it was the same as the path of the golf ball. How it is obtained the position of a ball identifying the color red is the same, obtaining the value of each component of RGB saturation of a pixel that has the ball. Then analyzing each pixel of the image thus compares the value of the RGB components by calculating the Euclidean distance between the sample value and the value of each pixel of the image. The Euclidean distance between the components must be less than 10.

Transition tables. Division intends any surface that generates Table 1. If we move any object on this surface, different frequencies will then step in the boxes. Matrices frequency and probability are constructed considering the positions and changes from one period to another between them, generating an array of 5 x 5.

TABLE I
FREQUENCY MATRIX

	A	B	C	D	E
A	408	62	0	0	0
B	61	406	68	0	0
C	0	63	402	60	0
D	0	0	59	301	55
E	0	0	0	55	256

In this matrix we find that the cell in row A to column B indicates how many times the ball is in B since it was in A, the cell located in row D and column D indicates the time that the continuous ball D because D was in previously. The transition probability matrices likely handle as shown in Table 2

TABLE 2.
LIKELIHOOD MATRIX OF TRANSITION ONE-STEP OR FIRST ORDER.

	A	B	C	D	E
A	0.8680	0.1319	0	0	0
B	0.1140	0.7588	0.1271	0	0
C	0	0.12	0.7657	0.1142	0
D	0	0	0.1421	0.7253	0.1325
E	0	0	0	0.1612	0.8387

The above matrix called transition matrix is a step and gives the probability that the system moves from one state to another in a step that is in a transition. Also be used to specify the probability that a system moves from one state to another in any number of steps or transitions, if a step is not likely to change with time. Using the following example you can see how they relate Bayes networks and answer the question: If the system is in state B, what is the probability that the state is in C after two stages? Figure 3 shows the calculations.

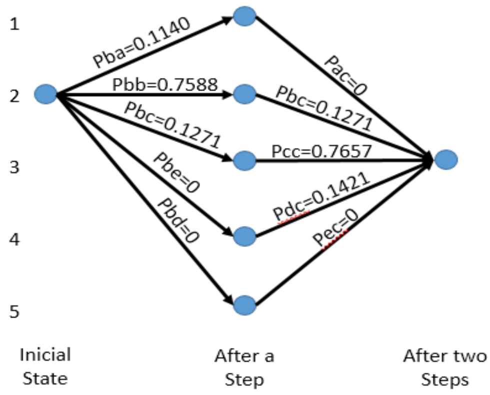

Figure 3. Movement from "b" to "c"

The number of each arrow represents the probability of taking that particular step. For example in one of the arrows B the system goes to B and then from B to C. The probability of the first step is 0.7588 and the second step is 0.1277. Then multiply the two probabilities that the trip system that route. The probability of any possible route may be calculated in the same manner.

$P^2_{B,C}$= (0.1140x0)+(0.7588x0.1271)+(0.1271x0.7657)+(0x0.1421)+(0x0) = 0.1937

In general terms, if $P_{i,\ j}$ is the probability that the system will from state i to j in a step and $P^2_{i,\ k}$ is the probability that the system will IAK in two steps, then:

$$P^2{}_{i,k} = \sum_{todo\ j} (P_{i,j})(P_{j,k}) \qquad (1)$$

Equation 1 can be used to calculate $P^2_{i,\ k}$ for all possible combinations of i k. May occur after the values in matrix form as shown in Table 3 and will have built a transition matrix in two steps [6][7].

TABLE 3.
LIKELIHOOD MATRIX OF A MARKOV CHAIN OF SECOND ORDER

	A	B	C	D	E
A	0.7685	0.2146	0.0168	0.0000	0.7685
B	0.1855	0.6061	0.1938	0.0145	0.1855
C	0.0137	0.1829	0.6178	0.1703	0.0137
D	0.0000	0.0171	0.2119	0.5636	0.0000
E	0.0000	0.0000	0.0229	0.2521	0.0000

The process shown in equation (1) can be used to calculate the transition matrix of three steps. If $P^3_{i,\ j}$ is the probability of moving from i to j in three steps, then:

$$P^3{}_{i,j} = \sum_{todo\ k} (P_{i,k})(P_{k,j}) \qquad (2)$$

By calculating the transition probabilities of two, three or n steps, we obtain a list of all routes. It can give you a simple way to get the transition probabilities by matrix multiplications.

Transition matrices and especially the multi-step transition is not only useful for finding a number of questions on the control system, is also a basic building element to answer questions about the powerful system called steady state.

Communications. Communication between the module of transition table and output modules were made for UDP (User Datagram Protocol) for speed, and allows sending messages Broadcast mode, which permits messages to communicate different multiple devices.

Bookseller Processing offers hypermedia, which tolerates up to 254 machines within a local network, the advantage here over an Internet connection, is that there is no packet loss.

Outputs. The output of this system is the position of the robot after the order sent him where he was positioned. By getting tested, you are first shipment the order that the robot had to find a position near the center of the plan.

The figure 1 shows the modules of the system.

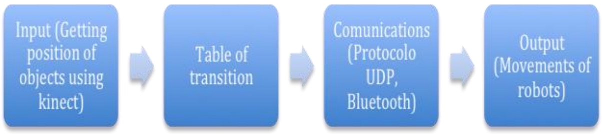

Fig. 1. Experiment Modules

Programming and communication in the system

The system consists of a Kinect sensor that allows for the joint information of the user, the sensor is connected to a computer through a program which created in Processing to calculate the rotation angle must perform each servo to reach the position specified by the user, this program in addition to the calculation of the rotation angles, must send the number of servo and angle to move through serial communication and a protocol established by the company ROBOTIS. Figure 2 shows how the system is constituted. Communication protocol dictates that the data packet should be constituted by:

Code indicating that the control is remote (30)

- Connection speed between the computer and the controller (4)
- Number

Servo to move desired angular position

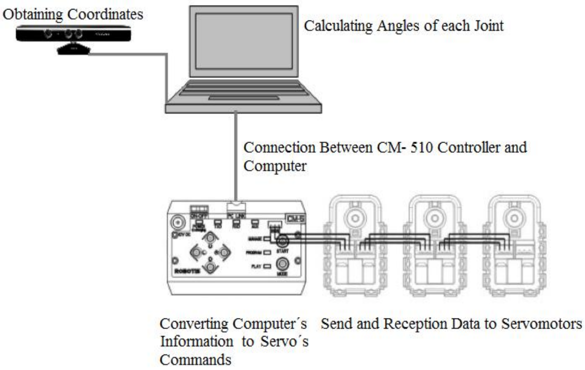

Figure 2. Projection of the vector formed by the right arm in the Z, Y.

The robot controller consists of a family AVR 2651, manages a separate communication with the computer and servomotors, this driver must perform a program that allows you to read and write data on the servo and also communicate with the computer to obtain operating data, the communication is 150000 bps. The robot controller must have programmed a helper function, such as in case any of the servomotors current warming sounds an alert and there is a kick operation.

The numbers of servomotors are 18, all connected between them and the controller via serial communication. Electric motors have a resolution of 10 bits / revolution.

III. RESULTS

After making system programming, we proceeded to make tests, which yielded the following results. Kinect for video games have a linear output as shown in Figure 6 and operates in a range of distances of 1 to 5 meters, the Kinect computer has a distance range of 50 to 500 cm. It can work at a distance of 50 cm to 250 cm with the Tracking Skeleton with an acceptable error of less than 10 cm, as the distance between the user and the error increases Kinect surges exponentially, this is shown in Figure 3.

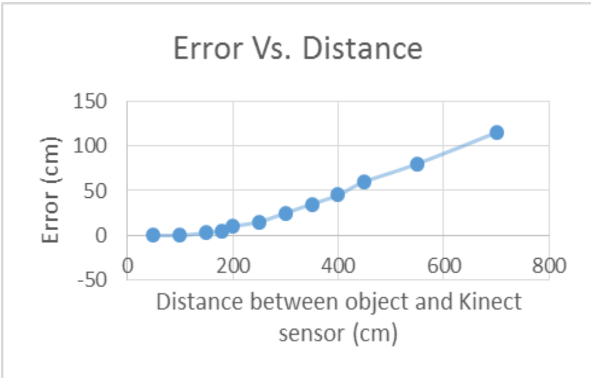

Fig. 3. Kinect Sensor Output

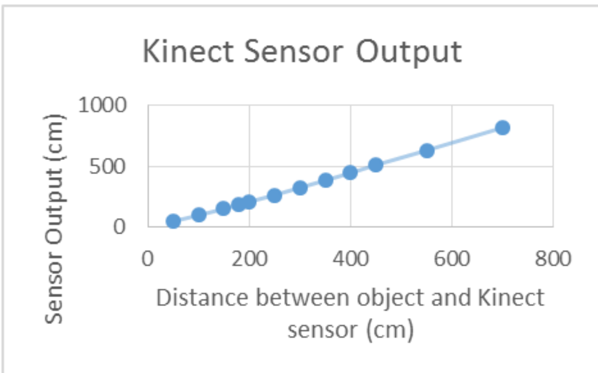

Figure 4. Error sensor output.

The robot used is Lego mind storms brand, which with some modifications communicates serially and the computer is used [8]. For positioning data, block programming the CPU to allow comparison of your current position with the one sent by the

ISSN : 2248-9622, Vol. 3, Issue 6, Nov-Dec 2013, pp.464-468

computer and be positioned again and thus move towards the ball, this is by subtraction of the current position less initial. To take both the position of the ball and the robot used a kinect sensor.

The technique for tracking of objects or colors used (Euclidean distance) is good, however when brightness changes are less effective, because the value of the sample is altered as there are cases of illumination change. This causes our system sometimes does not recognize the red ball; other times the system ignores because to light sources. It also confuses a color objects whose combination of saturation components to approximate to 0 (Euclidean distance between colors). Controlling these objections the system has no fault.

Another important point to consider is the relationship entries cm versus Pixel, this relationship allows us to know what is the size of our visual field and well able to handle speed ratios. In the graph we can see the linear relationship between pixels and centimeters. See figure 5.

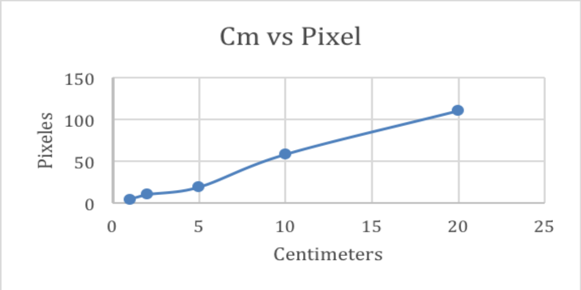

Figure 5. Relationships between Centimeters and Pixel

However, when the distance between the sensor and the plane changes, so does this relationship. Graphical equation for a distance between the sensor and the plane 2 meters is showed in figure 6.

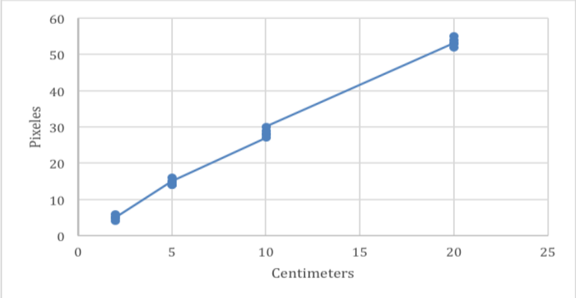

Figure 6. Relationship cm versus Pixels

The graphs and equations it follows that the relationship between pixels and inches changes according to the depth or distance between the sensor and the plane in which the measurements are made. If you have an environment where the distance between the sensor and the plane changes (such as most real life circumstances) these relationships are not as functional as it would involve setting at all times, so it is necessary establish a relationship between the variables centimeters, depth and pixels, again resort to multiple linear regression.

cm = -5.6272 +0.15107 * pixels +0.56252 * depth (in feet)

This relationship allows us to know the extent of the objects in centimeters depending on the value in pixels, which according to what I have caught on camera at different depths. This relationship handles error 4 centimeters to two meters maximum depth, however as our field of view is 1.20 meters error is 1 cm and is not relevant to our experiment, so that the expression is valid.

In graphical figure 7 is a distinguished robot reach the position at a time of about 4 seconds, also distinguished variations 10 pixels, however not by movement of the robot but for the image information captured from Kinect.

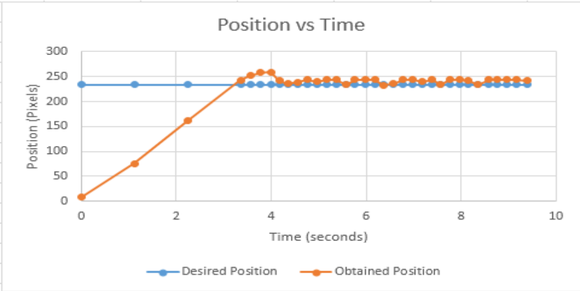

Figure 7. Time location of an object

After these tests robot-positioning costs were tested by multiple positions, we obtained the graphic of figure 8:

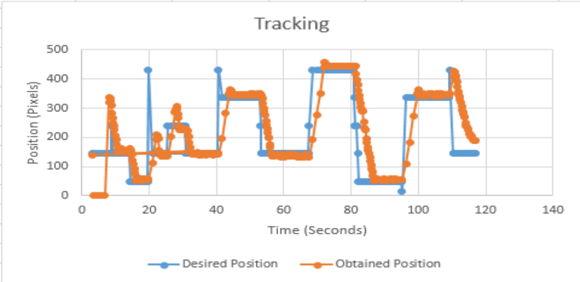

Figure 8. Multiple testing.

In these graphs shows that the mobile target tracking is performed, even when there are abrupt changes this ignores long distances. However, the abrupt changes in the game rarely appear. Figure 9 shows the lay robot used.

Figure 9 Robot used for the proposed system

IV. CONCLUSION

Communication between the computer and the controller is serial, the data packets contain the address of the booster, the direction of rotation and the number of degrees you have to turn, the robot controller suitable to transmit this information on a protocol to servomotors.

Because the program execution time and the transmission time data, the response of joint motion of the robot with respect to the beginning of joint movement of the demonstrator, has a delay less than 20 ms.

Will implement a system based on a grid plan that allows the location of a moving object based on Markov processes. Therefore, obtaining a matrix of transition likelihood to the step in each.

The same system is applied to a real robot where you get the same results. The robot can calculate the trajectory of the ball smoothly, even if the environment where change, with the only condition that a grid is retained to perform the Markov process. However lighting conditions affected by or to be controlled

A drawback is observed that when increasing the size of the plane, the matrix must be increased, so the calculation is also increased. Then the algorithm is limited by the processing speed of the computer.

V. ACKNOWLEDGEMENTS

We are recognition to the IPN (Instituto Politécnico Nacional) for their support for the project.

REFERENCES

[1] Angulo J. M. Historia y evolución de la robótica. En Robótica práctica. Tecnología y aplicaciones. Pp 13-14. Ed. Paraninfo. 5 ed. Madrid. 1999.

[2] K.S. FU, R.C González, C.S.G. LEE. *Robotica: Control, Deteccion, Vision e Inteligencia*. Ed. Mc Graw Hill, 1993

[3] J. Zhou, L. Liu, Z. Pan and H. Yan, *Scanning 3D Full Human Bodies Using Kinects*, IEEE Transactions on Visualization and Computer Graphics, vol. 18, no. 4, pp. 643-650, 2012.

[4] Livingston, M.A.; Sebastian, J; Zhuming Ai; Decker, J.W. *Performance measurements for the Microsoft Kinect skeleton*. Virtual Reality Short Papers and Posters (VRW), 2012 IEEE. pp. 119 - 120. 4-8 March 2012

[5] K. K. Biswas and S. K. Basu. *Gesture recognition using Microsoft Kinect.* In Automation, Robotics and Applications (ICARA), pp. 100-103, 2011.

[6] Bjorn Bottecher, Rne L. Schilling, “Approximation of Feller processes by Markov chains with Lévy Increment”, Stochastics and Dynamics, Vol. 9, No. 1 (2009) copyright World Scientific Publishing Company.

[7] Radu Zaharopol, Invariant Probabilities of Markov-Feller Operators and Their Supports, Kluwer Academic Pubhlishers, Netherlands, 2002.