In Proceedings of 9th Workshop on Humanoid Soccer Robots
14th IEEE-RAS International Conference on Humanoid Robots (Humanoids), Madrid, Spain, 2014.

Robot navigation by fusing a spatio-temporal video descriptor with a robust humanoid motion control for kicking a ball

Kevin Infante[1], Fabio Martinez[1], Mario Arbulu [1]

***Abstract*— Cooperation work among robots is a very popular and active research area, in which the synergy among different dynamic methods, communication strategies and feedback control algorithms allows that a group of robots work together to achieve a common goal. Although, the last years have been reported important advances in this line, there exist many limitations on the develop of these collaborative works because the extremely variability of the scenes, the task challenge and some robot's hardware issues. This work develops the specific tasks of locating, reaching and kicking a ball for a humanoid robot, and proposes a new approach which solves stability and illumination problems.**

Vision tracking feedback to locomotion algorithm will be employed in those tasks. Vision algorithm is based on spatio-temporal characterization, which give the input to the motion planning robot algorithm. The motion planning algorithm introduces a novel control approach which allows to reach the ball and kick it stably. A validation of the proposed method in terms of tracking algorithm error with motion control, in both real and simulated indoor environment allows to evidence the proper performance of the proposed approach.

I. INTRODUCTION

Cooperation works between robots is an open area of research. As an example, humanoid robots playing soccer is a case, in which the robots should interact between them. For solving the problem of the robots soccer match challenge, it is possible to divide in several tasks, such as: tracking the ball, tracking the partner, approaching to the ball, selecting the goal, kicking the ball, and so on. At this stance, it is clear that the researchers must to integrate the solutions of those subproblems, that is, vision and motion planning algorithms are integrated, in order to achieve the robots interaction. In this work, vision algorithms will be integrated with motion planning strategies for the specific tasks of tracking, approaching and kicking a ball.

Computer vision strategies have widely used as input to control robot trajectories and also to reconstruct road maps to navigate in unknown scenarios, Moeslund [1]. These computer vision algorithms are dedicated to detect and label different objects into scenes, that allows to understand different current location and take decisions about the robots motion.

Motion planning of humanoid robots could be solved by many proposals. In this work, two kinds of motion planning are solved, the first one is stable walking motion to any goal; the second one is kicking a ball stably.

Kicking a ball approaches could be described as following: Muller et. Al approach deals with Bezier curves which are changed online, in order to get stable motion [2], this approach does not take in account the whole body robot dynamics, at only the center of mass position as a stability criteria. Bucley [3] proposes an open loop motion patterns which take into account the robot balance off line, this approach can not solve the uncertainty disturb on the robot, due to surface, or external force by other robot. Urieli et. Al [4] propose an optimization algorithm for each robot ability, but it was not tested on real robots. Hester et. Al introduce an innovative approach which allows to the robots generate motion by learning from previous ones, the control system could be adapted to random situation, and stable motion could be generated after several tests, so the problem with this method is that, the robot could fall down while it is learning. Ferreira et. Al [5] introduce the omnidirectional method in order to generate the vector direction of motion, but it stability criteria is not clear.

Regarding the walking motion planning, Kanehiro et. Al [6] propose a global robot motion algorithm which allows the robot navigate around small areas at very short control cycle, this approach allows to the robot react quickly to suddenly external effort. Arbulu et. Al [7], [8], [9] propose a local stable motion planning for humanoid robots, which allows efficient stable motions; that approach could be used in this research.

The main contribution from a computer vision standpoint it was a simple but efficient strategy that accurately follow a ball to perform a kicking. Such strategy it is based on an statistical tracking that follows a current ball observation mapped from a hough space and then adjusted with previous states exponentially weighted from past times.

On the other hand, the principal contributions on motion planning algorithm were:

- Efficient kinematics and dynamics models avoid singularities, analytical closed solutions are implemented.
- Stable walking patterns are developed, taking into account no linear robot dynamics.
- Dynamically Stable kicking ball motion was performed

This work is organized as following: the section 2 describes the problem statement, the section 3 shows the

*This work was supported by Engineering Faculty of Universidad de La Sabana

[1]Kevin Infante is undergraduate student with Faculty of Engineering, Informatics Program, Universidad de La Sabana, Chia, Colombia, South America KevinInHe@unisabana.edu.co

[1]Fabio Martinez, MSc is Part time Professor with Faculty of Engineering, Informatics Program, Universidad de La Sabana, Chia, Colombia, South America fabio.martinez1@unisabana.edu.co

[1]Mario Arbulu, PhD is Full Professor with Faculty of Engineering, Informatics Program, Universidad de La Sabana, Chia, Colombia, South America mario.arbulu@unisabana.edu.co

spatio-temporal video characterization, in the section 4 an approach for motion control will be described; in section 5 the proposed integrated algorithms are detailed; in the section 6 experimental results will be shown, and finally in section 7 the conclusions and future works will explained.

II. PROBLEM STATEMENT

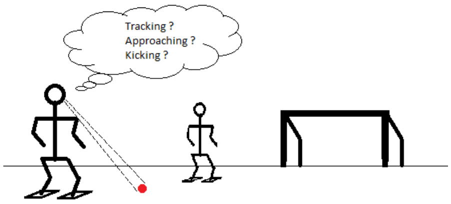

Fig. 1. Problem statement.

While the humanoid robots cooperation tasks are developing, some sub-problems should be solved, such as, vision tracking, locomotion control and planning, communications between robots, sensors navigation, and so on. This work proposes an approaching for solving the specific task of tracking a ball, approach and kicking it, which will be used on a soccer robots match. Thus, how to tracking a ball, how to select the goal, how to planning the walking robot motion and how to kick a ball stably sub problems will be solved in this work. The following sections will explain the proposed approaches, see Fig. 1 .

III. SPATIO-TEMPORAL VIDEO CHARACTERIZATION

A fast strategy to identify and track a ball was herein proposed by using a classical statistical EWMA (exponentially weighted moving average) tracking strategy under a sequence of frames captured into the humanoid robot. This strategy evolve in time the current ball position and size w.r.t to the robot reference space by characterizing an detecting at each frame the ball into the scene from a hough descriptor and then measuring the coherence of this estimation w.r.t previous history of the ball trajectory. The feature vector of position and size $x = [\overline{x}, \overline{y}, z^{'}]$ into the EWMA is then evolved as $\widetilde{x}_i = \alpha^2 x_{i-2} + \alpha(1-\alpha)x_{i-1} + (1-\alpha)x_i$ where α is a scalar number defined as $\alpha = [0-1]$.

The current observation of the ball x_i is measured by using a hough descriptor into the sequence of the images captured by the humanoid robot. For doing so, each frame is first mapped to a HSV color space and then filtered using a Gaussian kernel with a σ which is trained and adjusted in previous experiments.

Then it was computed over at each frame a Canny filter to detect the more relevant edges in the captured scene [1]. From this edge map it was computed search parametric circles with a interval range of localization and size bounded by previous estimation with a tolerance value, defined as $x_i = x_{i-1} + \delta$, in which δ represent an additional space of each center on the previous position x_{i-1}. The association of parametric circles by using the hough trasform is perfomed by groupings of edge points into object targets and then performing a special voting strategy over parametrized family of circles.

This tracking strategy allows to compute in real time the ball position and result robust to illumination noise and even to partial ball occlusion. Once is detected the best circle candidate that represents the ball, the proposed methodology take such detection as input to perform a robust motion planning control of the humanoid robot.

IV. MOTION CONTROL

Global and local motion could be solved at this stance. The global goal is the final position which the robot should achieve, and the local goal is the landing robot foot position.

Motion control could be divided in kinematic and dynamic models, [8], [9], and motion planning, [7].

Kinematic model which maps joint and cartesian spaces, is developed as following, Fig. 2:

$g_{st}(0)$: Initial manipulator configuration (right leg)
$g_{st}(\theta)$: Goal manipulator configuration (right leg)
$g_{th}(0)$: Initial manipulator configuration (right hand)
$g_{th}(\theta)$: Goal manipulator configuration (right hand)
θ_1 to θ_{11} : Degrees of freedom
θ_x^{rh} to $\theta_{\theta z}^{rh}$: Right hand position and orientation
θ_x^a ,θ_y^a ,θ_z^a, $\theta_{\theta x}^a$, $\theta_{\theta y}^a$, $\theta_{\theta z}^a$: COG position and orientation
θ_x^r to $\theta_{\theta z}^r$: Right foot position and orientation

So, the forward kinematics is given by the next equations:

$$g_{st}(\theta) = e^{\xi_x^r \wedge \theta_x^r}.e^{\xi_y^r \wedge \theta_y^r}...e^{\xi_{\theta z}^a \wedge \theta_{\theta z}^a}.g_{st}(0) \qquad (1)$$

$$g_{th}(\theta) = e^{\xi_x^a \wedge \theta_x^a}.e^{\xi_y^a \wedge \theta_y^a}.e^{\xi_z^a \wedge \theta_z^a}...e^{\xi_z^{rh} \wedge \theta_z^{rh}}.g_{th}(0) \qquad (2)$$

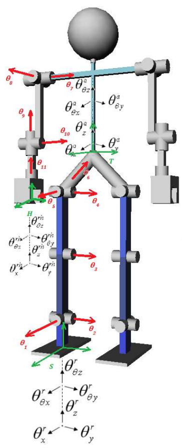

Fig. 2. NAO kinematics model, twists.

Motion local planning could be achieved by selecting the footprints, such as following, Fig. 3, the goal foot configuration (position, P^{n+1}, and orientation, θ^{n+1}) is the input parameter for taking the next step. It could been obtained by humanoid sensors or external command. Those input parameters could be generalized in order to compute the $n-th$ step in real time such as:

$$P^n = P^{n-1} + R(\theta_z^n)^T . L^n \tag{3}$$

where

$$P^n = \begin{pmatrix} p_x^n \\ p_y^n \\ p_z^n \end{pmatrix}, \; P^{n-1} = \begin{pmatrix} p_x^{n-1} \\ p_y^{n-1} \\ p_z^{n-1} \end{pmatrix},$$

$$R(\theta_z^n)^T = \begin{pmatrix} cos(\theta_z^n) & -sin(\theta_z^n) & 0 \\ sin(\theta_z^n) & cos(\theta_z^n) & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

$$L^n = \begin{pmatrix} L_x^n \\ -(-1)^n L_y^n \\ L_z^n \end{pmatrix}$$

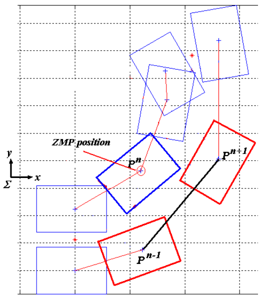

Fig. 3. Footprints planning.

In order to obtain stable step motion, it it possible to develop concentrated mass model such as proposed in [7], when the stable robot body motion is obtained, because its center of mass motion is constrained by the zero moment point reference. With that consideration, by composing several steps until achieve the global goal, the stable walking motion could be obtained.

V. INTEGRATED PROPOSED SOLUTION

Flowchart is shown in Fig. 4. As the flowchart shows, the program is divided into two threads, called ”Thread 1” and ”Thread 2”. The function of ”Thread 1” is to capture and analyse the images captured by the the humanoid robot's camera in order to get the necessary ball information in real time, and Thread 2 uses that information to make the humanoid robot move to the ball.

This work has three main objectives: locate the ball, and to make the humanoid robot to approach and kick it. To do this, as the flowchart shows, there are necessary the next steps:

- Extract the ball center coordinates (*y*, *z*) and its radius.
- Calculate the motion angles for the humanoid robot's head and body (ρ and γ)
- Make the humanoid robot move its body and kick the ball according to the head motion angles.

We use threads, because in this work the image capturing process is independent of the locomotion process, which allows the algorithm to capture images at a higher frame rate, so the whole algorithm can work faster.

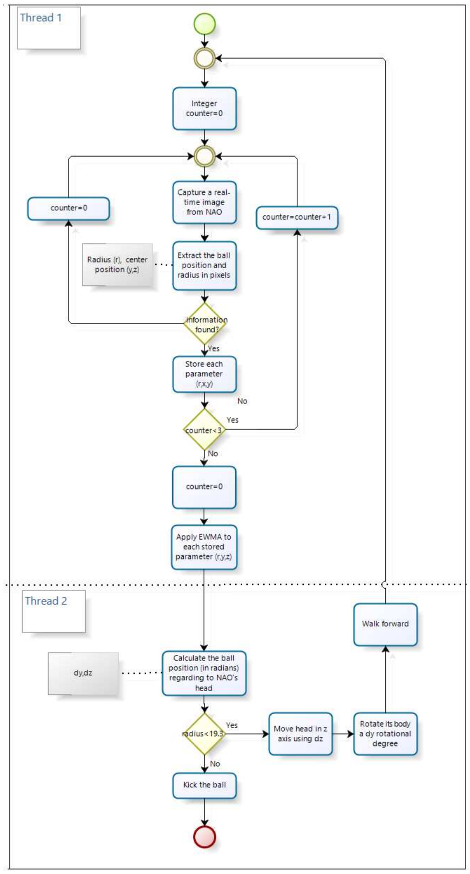

Fig. 4. Flowchart

A. Extract the ball position and radius in pixels

The aim of this step is to extract center coordinates (*y*, *z*) and radius measured in pixels of a red ball. To do that, image captured by the robot is filtered using both color filter and shape filter.

The color filter extracts from the image the pixels whose color value is the desired. This work uses HSV format to define the color value that will be filtered.

The result of the color filtering is an image with just black and white pixels. White pixels corresponds to those pixels whose color value was red. That image is the input for the shape filter, which will look for those group of pixels that look like a circle.

The output of the shape filtering is the data about the circles that it found in the image: center coordinates (*y*, *z*) and radius, measured in pixels, which is the objective of this step.

The parameters of both filters were found by experimentation.

B. Calculate the movement angles for the humanoid robot's head and body

The detected ball center coordinates of the ball (*y*, *z*) will be used to calculate the angles γ and ρ that control the head and body movements respectively. That ball center coordinates were measured based on a system coordinates related to the image captured by the humanoid robot, whose origin point is the top left corner of the image, and the values on the right and below of the origin point are positive.

Now, angles γ and ρ are measured by a different system coordinates related to the the humanoid robot's head, whose origin point represents when the humanoid robot's head is centered, its horizontal positive values represents when the humanoid robot has its head rotated clockwise, and its vertical positive values represents when the humanoid robot has its head turned downwards. This system coordinates is shown in Fig. 5

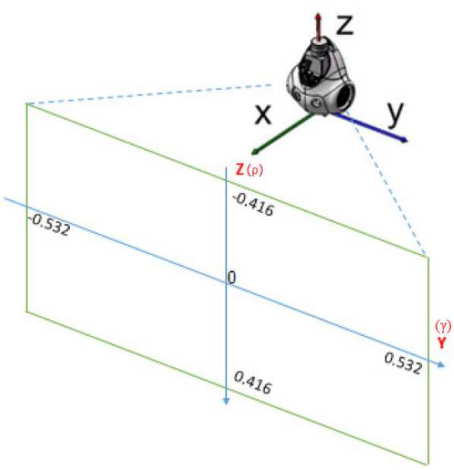

Fig. 5. System coordinates for degree values based on the image captured by the humanoid robot.

To calculate γ and ρ (rotation head angles in radians), the coordinates of the ball center *y* and *z* must be translated into the system coordinates of the humanoid robot's head. When it happens, *y* coordinate becomes the angle γ and *z* becomes the angle ρ. It is done using a calculation based on a rule of three, which is a little different for each axis because of the difference between both coordinates systems.

Image captured by the humanoid robot has a size in both pixels and radians, which allows making the rule of three. Actually the size of the image in radians is the the humanoid robot's visual field in both *y* and *z* axis. The image length in *y* and in *z* axis is respectively 320 x 240 in pixels and 60.97 x 0.831 in radians.

The following equation deals with the computation of ρ, which depends of z_pixel_value:

$$\rho = (((z\_pixel\_value/240) * 0.831) - 0.416) \quad (4)$$

The same procedure could be used for computing γ as a function of y_pixel_value.

C. *Moving the humanoid robot's head and body*

The previous subsection explained how the angles γ and ρ that controls the the humanoid robot's movements were obtained. This part explains how the algorithm uses the data obtained until now (*y*, *z*, *radius*, γ, ρ) to make the robot approach the ball and kick it.

To approach the ball, the humanoid robot must first turn its head and its body an angle γ and ρ respectively to make the ball be centered regarding to it, then it must walk forward until it is close enough to the ball to kick it. It is considered that ball is centered regarding to the humanoid robot's head and body if its center coordinates (*y*, *z*) are centered in the image captured by the humanoid robot. Center coordinates are centered in the image if they are inside of a rank of pixel values, which size was determined by experimentation. Taking into account the image size in pixels, ball is centered in y-axis if *y* is between 140 and 180 pixels, and for z-axis if *z* is between 100 and 140 pixels.

The algorithm uses radius value to determine if the humanoid robot is close enough to the ball to kick it. During experimentation, it was determined that the humanoid robot is close enough to the ball to kick it when radius is greater than 20.5.

As the algorithm is analyzing the image captured by the humanoid robot in real time, it can correct the the humanoid robot's walking direction if ball is moved or if the humanoid robot is deviating from its objective due to problems like low friction ground, engine warming, etc.

Finally, when the humanoid robot approach the ball, it is ready to kick it. Depending on the ball coordinate in y-axis (*y*), the humanoid robot will choose one of its legs to kick the ball. If *y* is greater than 160, the humanoid robot will use its right leg, on the contrary it will use its left leg.

VI. SIMULATION AND EXPERIMENTAL RESULTS

In this section the three tasks are shown: tracking, approaching and kicking a ball. For simulations, the simulator used was Webots. Results are shown below.

A. *Tracking a ball*

During experiments and simulations it was taken 55 samples at 11 different distances, which were the same for both environments. In experiments the algorithm could locate the ball 50 times (91% of effectiveness), while in simulations it could locate the ball in all locations at first (100% of effectiveness). The algorithm failed to detect ball in some samples in real environment because, despite the algorithm is capable to detect a ball with partial occlusion, as Fig. 7 shows, there are some specific locations in a middle-lightened environment in which illumination is too low so the humanoid robot cannot detect the ball. In contrast, the humanoid robot could detect the ball in all samples in the simulated environment because its illumination was high, as Fig. 6 shows.

B. *Approach a ball*

In this procedure the humanoid robot had to approach the ball 7 times in 7 different locations and to stop if it was close enough to kick the ball. In all samples in which the algorithm could locate the ball, the humanoid robot successfully turned its head and its body regarding to

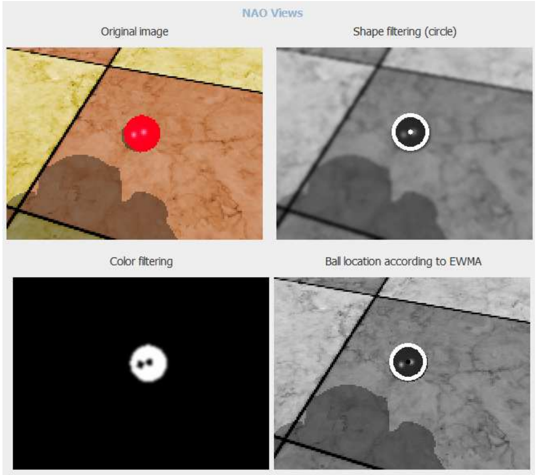

Fig. 6. Image captured by the humanoid robots camera with some filters and EWMA location at simulated location

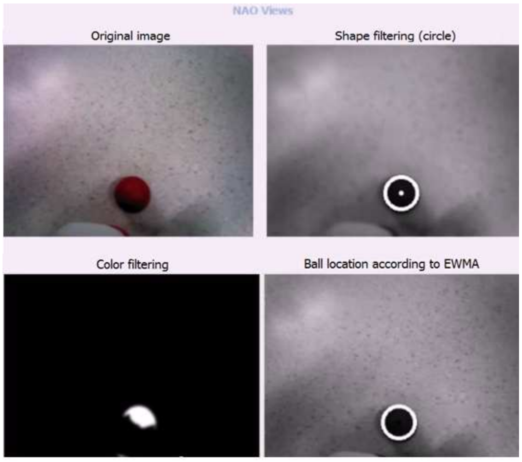

Fig. 7. Image captured by the humanoid robots camera with some filters and EWMA location at real location

the ball position, even if it was moved. The problem was to determine a radius value which indicates the humanoid robot when it was close enough to the ball, because there is no a direct relation between ball radius size and radius detected, due to the detected radius value is unstable, and it change even if ball is quiet. The radius value found by experimentation were 20.5 for real environment and 29.6 for simulated environment. The experiment used to find that values is detailed in Appendix.

The results of how close was the humanoid robot to the ball before kick it using the radius values found for both environments are shown in Table II (for real environment) and Table I (for simulated environment).

As Table II shows, the humanoid robot could reach and kick the ball in approximately an 84% of the samples, which proves the algorithm efficiency in the real environment. However, table I shows a lower efficiency of the algorithm, because the humanoid robot just could kick the ball in the 67% of the samples. It is because the simulator camera in some occasions shows the radius of the ball higher than it

TABLE I

MEASUREMENT OF THE DISTANCE BETWEEN THE HUMANOID ROBOT AND THE BALL IN THE SIMULATED ENVIRONMENT

Point	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5	Sample 6	Sample 7
1	0.03	**0.07**	**0.07**	**0.07**	0.02	-	0.02
2	0.04	**0.07**	-	0.02	**0.06**	0.04	0.05
3	0.01	0.04	0.04	**0.06**	**0.07**	0.04	0.05
4	0.05	0.01	0.02	0.01	0.05	0.04	**0.1**
5	0.03	0.03	0.04	**0.11**	**0.08**	**0.06**	0.05
6	0.05	**0.08**	0.01	0.03	**0.09**	0.03	0.01
7	0.05	0.02	0.01	**0.06**	0.01	**0.06**	0.05

TABLE II

MEASUREMENT OF THE DISTANCE BETWEEN THE HUMANOID ROBOT AND THE BALL IN THE REAL ENVIRONMENT

Point	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5	Sample 6	Sample 7
1	0.04	0.05	0.03	0.04	0.05	**0.06**	0.04
2	0.04	**0.09**	0.05	0.05	0.05	0.03	0.03
3	0.05	0.04	**0.06**	0.05	0.04	0.05	0.05
4	0.05	0.04	0.05	0.04	0.05	0.04	0.04
5	0.04	-	0.04	0.05	**0.06**	0.05	**0.08**
6	0.03	0.05	0.04	0.04	0.03	0.05	**0.06**
7	0.05	0.04	0.05	0.05	0.04	0.05	0.04

really is, which makes the robot to be too far from the ball to kick it.

In both simulated and real environment there are some samples in which the humanoid robot executed the kicking movement after it touched the ball, or even it did not located the ball. They are marked with a short line. There are three main causes for this: accumulated orders in the the humanoid robot's memory and WiFi signal problems.

Fig. 8 shows how the humanoid robot rotate its body to make the ball be centered.

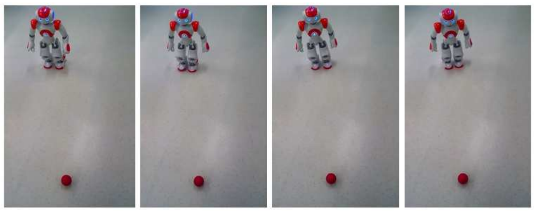

Fig. 8. the humanoid robot rotate its body to make the ball be centered regarding to it

C. Kick a ball

Kick the ball is the final task accomplished in this work. Once ball is located and the humanoid robot could reach it to a proper distance, it has to execute a set of movements to kick the ball, taking into account to maintain robot balance and to kick the ball as best as possible.

With the developed algorithm, the humanoid robot could kick properly the ball 27 of 31 times (87% of effectiveness) in real environment and 26 out of 31 times (83% of effectiveness) in simulated environment. The algorithm is enough robust to preserve the humanoid robots balance during kicking movement, however, facts like the engine warming influences the humanoid robots movements accuracy in real environment and simulator lag affects movement precision in simulated environment, so that kind of facts make the humanoid robot to lose its balance.

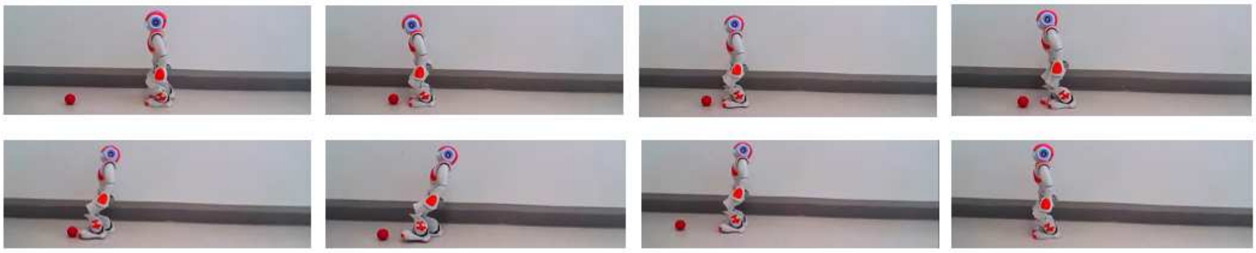

Fig. 9. Side view of ball localization algorithm running and kicking movement.

Fig. 9 shows the side view of how the robot the humanoid robot approach and kick the ball.

VII. CONCLUSIONS AND FUTURE WORKS

In this work it was presented a novel approach of control navigation to kicking a ball based on a robust vision tracking strategy and a robust control robot motion.

The previous results shown that the NAO humanoid robot could execute properly the tasks of tracking, approaching and kicking a ball in most of cases for both environments. For the task of tracking a ball, the humanoid robot had an effectiveness of 91% in real environment and 100% in simulated environment; for the approaching a ball task, the robot succeed at 84% of samples in real environment and just 67% in simulated environment, and the robot completed the kicking a ball task with an effectiveness of 87% in real environment and 83% in simulated environment. Despite this approach did not work well enough in the approaching a ball task in the simulated environment, the results shows that this work has a proper performance in real environment for whole complex task, so that it is capable to overcome the most serious obstacles in the execution of tracking, approaching and kicking a ball: robot stability, robot's deviation from its original path, and object partial occlusion.

The future works are oriented to overcome the other facts that affects the algorithm performance. First, is to improve some humanoid movements to make them to be faster, so avoid the order accumulation in its memory, and to develop a novel more reliable communication protocol between the computer and humanoid robot to prevent errors.

APPENDIX

NAO humanoid robot must reach the ball to a distance between 0 and 0.05m in order to kick it.

The experiment was made in order to determine the radius value, when the humanoid robot is close enough to the ball (between 0 and 0.05m from the ball). It was to measure 10 times the last five values of the radius detected by robot's camera before it got too close to the ball. In this way, the changes of the radius value detected by the humanoid robot's camera when it is moving its body are taken into account, making this a high-accuracy experiment. The radius values found by this experiment were *20.52* for real environment and *29.63* for simulated environment.

REFERENCES

[1] Moeslund, T. (2009, March 23). Canny Edge Detection [Online]. Available http://www.cse.iitd.ernet.in/ pkalra/csl783/canny.pdf.

[2] Muller, J., Laue, T., Rofer, T., Kicking a Ball Modeling Complex Dynamic Motions for Humanoid Robots, in RoboCup 2010: Robot Soccer World Cup XIV, Lecture Notes in Computer Science Volume 6556, 2011, pp 109-120.

[3] Buckley, A., Humanoid Robot Soccer Locomotion and Kick Dynamics: Open Loop Walking, Kicking and Morphing into Special Motions on the Nao Robot. Masters thesis, National University of Ireland Maynooth, 2013.

[4] Urieli, D., MacAlpine, P., Kalyanakrishnan, S., Bentor, Y., and Stone, P. On optimizing interdependent skills: a case study in simulated 3D humanoid robot soccer. In The 10th International Conference on Autonomous Agents and Multiagent Systems - Volume 2 (AAMAS '11), Vol. 2. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 769-776. 2011.

[5] Ferreira, R., Reis, L.P., Moreira, A.P., Lau, N., Development of an Omnidirectional Kick for a NAO Humanoid Robot,Advances in Artificial Intelligence IBERAMIA 2012, Lecture Notes in Computer Science Volume 7637, 2012, pp 571-580.

[6] Kanehiro, F.; Yoshida, E.; Yokoi, K., "Efficient reaching motion planning and execution for exploration by humanoid robots," Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on , vol., no., pp.1911,1916, 7-12 Oct. 2012, doi: 10.1109/IROS.2012.6385587

[7] Arbulu, M.; Kheddar, A; Yoshida, E., "An approach of generic solution for humanoid stepping over motion," Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on, pp.474,479, 6-8 Dec. 2010, doi: 10.1109/ICHR.2010.5686345

[8] Arbulu, M.; Balaguer, C.; Monge, C.; Martinez, S.; Jardon, A, "Aiming for multibody dynamics on stable humanoid motion with special euclideans groups," Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, pp.691,697, 18-22 Oct. 2010, doi: 10.1109/IROS.2010.5649923

[9] Arbulu, M.; Padilla, A; Ramirez, F., "Geometric balancing control of humanoid robots," Robotics and Biomimetics (ROBIO), 2013 IEEE International Conference on, pp.2136,2141, 12-14 Dec. 2013 doi: 10.1109/ROBIO.2013.6739785

[10] Santosh, D.; Achar, Supreeth; Jawahar, C.V., "Autonomous image-based exploration for mobile robot navigation," Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on, pp.2717,2722, 19-23 May 2008, doi: 10.1109/ROBOT.2008.4543622.

[11] Beinhofer, M.;Muller, J.; Burgard, W. "Effective landmark placement for accurate and reliable mobile robot navigation". Robotics and Autonomous Systems, vol.61, no.10, pp. 1060-1069. DOI: 10.1016/j.robot.2012.08.009