Hanuman-KMUTT: Team Description Paper

Nakarin Suppakun, Sathit Wanitchaikit, Wisanu Jutharee, Chonlatit Sanprueksin,
Atiraj Phummapooti, Narongsak Tirasuntarakul and Thavida Maneewarn

King Mongkut's University of Technology Thonburi,
Institute of Field Robotics (FIBO)
126 Pracha u-tid Rd., Bangmod, Thungkru, Bangkok 10140, Thailand
praew@fibo.kmutt.ac.th
http://www.fibo.kmutt.ac.th/teamkmutt/index.html

Abstract. This paper describes the design and development of the kid-sized humanoid robots of the Hanuman-KMUTT team for RoboCup 2014. The omni-directional walking algorithm which has been improved since last year together with the localization capability, allow the striker robot to approach the ball with faster and greater precision. The collaborative localization method between the goalkeeper and the strikers is also explored as the enhancement to the existing localization algorithm.

1 Introduction

Institute of Field Robotics(FIBO) at King Mongkut's University of Technology Thonburi (KMUTT) has developed the humanoid robots to participate in RoboCup humanoid kid-sized league since 2005 under the name 'Team KMUTT' and later on 'Pheonix'. The successor, 'Hanuman-FC', won Thailand Humanoid Soccer Robot Championship 2012 and participated in World-RoboCup 2013 (Eindhoven, Netherlands) under the name 'Hanuman-KMUTT'. In 2013, our team has passed into the quarter-final round of the World RoboCup. At the end of 2013, Hanuman-KMUTT has also successfully defended the champion title in the Thailand Humanoid Soccer Robot Championship 2013 and will prepare to be the representative of Thailand in the World-RoboCup 2014. Since Institute of Field Robotics, KMUTT, has hosted the Thailand Humanoid Soccer Robot Championship for three years since 2010. Our team members have gained lots of experience in organizing and have been familiar with refereeing the robot soccer under the RoboCup rules and regulations.

In this paper, we will describe the recent development of our kid-sized humanoid robots. Section 2 gives an overview of the system design in our robots which comprise of two strikers and one goalie. In section 3, the vision based navigation system of the robot will be explained. Section 4 discusses the game control and decision-making system. The last section concludes the paper.

2 System overview

Our team has 3 robots which have the same mechanical design: Hanuman, Tripetch, and Muchanu as shown in Fig 1. Each robot is composed of mechanical hardware, sensors, and computing hardware. The structure of all robots is made of aluminum alloy sheet metal. Each robot uses 20 servo-motors (details are provided in the robot specification sheet).

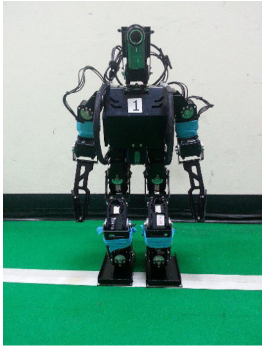

Fig.1. 'Hanuman' striker robot.

All striker robots use the same set of sensors. These are 9dof IMU (Sparkfun Sensor Stick SEN-10724) and a Logitech USB camera. The camera installed on the robot is a single lens USB webcam which has the horizontal field of view at 60 degrees. The capture resolution is 640x480 pixels. The workspace of pan-tilt unit is -120 to 120 degrees in panning and 0 to -80 degrees in tilting. The accelerometer tells the robot if there is any longitudinal and/or transversal tilt. The three rate gyros measure angular velocity in three axes of rotation. The body angle is estimated from gyro and accelerometer information. The angular position and velocity information will be used to adapt the attitude of the body during walking. The camera is used to track the ball and other objects of interest, which is crucial for navigation decision-making software.

The main computer for the striker robots are the PICOITX (PICO820) SBC with ultra low power Intel AtomTM processor Z500 (1.6GHz CPU) . The PICO820 board computer receives information from the CCD camera via the USB port. The computer computes the walking path and sends locomotion command to the ARM 7 (60MHz RISC microprocessor) motor controller via RS485 port.

In the current locomotion system, the high level sends the control input in form of planar velocity (Vx, Vy) and angular velocity (ω) to the low level locomotion module. The locomotion module will map the input velocity value to walking gait parameters and generate the desired trajectory for all motors in each legs using 6 DOF inverse kinematics and then convert these trajectories into command for the servo motors. The high level control decides the walking command and specifies the time duration or the distance of each command as the command parameters. In addition, our robot also has special commands such as kick and get up.

3 Vision based navigation system

To interact with objects in the soccer field, the robot shall be able to distinguish and identify many types of objects in the playing environment. These objects are the ball, goals, opponents, and teammates. Moreover, other useful information associated with the objects such as position and speed shall be determined. Once the information is determined, the robot can compute an appropriate action for the soccer game. The following subsections describe how the robot recognizes the objects in the field and compute the position of the objects.

3.1 Object Recognition

Different objects in the field have different colors. Thus we can roughly distinguish each object type by using the color segmentation technique. The segmentation algorithm used in our robot is based on CMVision [1]. Because HSV color space is more sophisticate for defining color description by human compare to RGB or YUV color space. Therefore we modified the original algorithm which uses YUV color space as input to HSV color space model. See Fig 2. for the segmentation result.

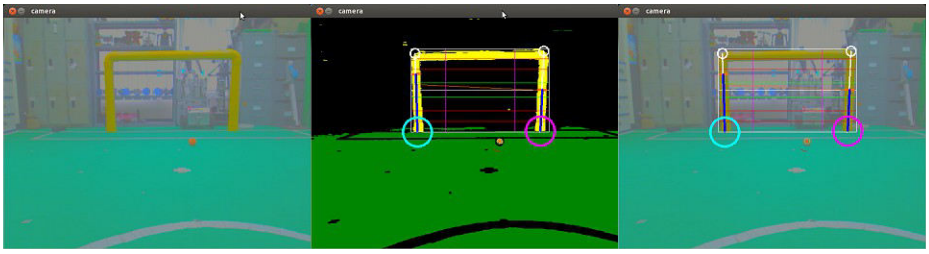

Fig. 2. Color segmentation for identifying search region for each object type.

The above result is used for reducing the search region in the identification process. For example, the orange segment surrounded by the green field would be considered as the candidate of the ball object.

For goalposts detection, we search for the base of the post(s) in order to find the position of the post(s). First we search for the left and right post by horizontal scan-lines, and see if both posts were found on the image. For each post, we determine the alignment vector direction of the post. Finally we search along that vector in downward direction until found the base. As shown in the Fig 2, the cyan and magenta circles have been put in the image to indicate the base of the left and the right post respectively.

Since 2013, the goals at both ends of the field are yellow, thus the color of the goalpost cannot be used for determining the robot's bearing in the field. To distinguish these goals, we attach the magnetometer inside the robots in order to assisted the orientation estimation procedure. We use k-nearest neighbor for determining the current bearing.

3.2 Position Determination

Three dimensional position of an object can be estimated when the object is identified in an image. The necessary information consists of a selected pixel on image coordinate (u, v) that belongs to the object, a known height of the object from the floor, robot’s head angles for both tilt and pan, and robot specific properties and configurations. When the centroid of the object's region deviates from the camera center, the relative angle between the object and the camera could be calculated as a rotation of a virtual joint. Thus, two virtual joints would be used (for horizontal and vertical direction). The world-coordinate position of a target object can be determined using the forward kinematic of these two virtual joints.

3.3 Localization

We have developed the simulation system to test our localization algorithm. The position and orientation of the robot is estimated using particle filter algorithm at sampling time Δt . Initially, the position and orientation of the robot is estimated using 50 particles. When the locomotion command is sent to the robot in form of linear and angular velocity v_x, $\mathrm{v_y}$, $\mathrm{v_w}$. The new position of the robot can be estimated from a displacement in straight(x), side(y), and heading direction (ω) respectively as the following:

Let ζ represents the variable $\begin{bmatrix} x \\ y \\ \omega \end{bmatrix}$, thus $\Delta\zeta = \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta\omega \end{bmatrix} = \begin{bmatrix} v_x \cdot \Delta t \\ v_y \cdot \Delta t \\ v_\omega \cdot \Delta t \end{bmatrix}$

The noise displacement error N_ζ and noise error from movement E_ζ are defined as:

$$N_\zeta = F_\zeta \cdot Rand(-\Delta\zeta, \Delta\zeta) \quad , \quad E_\zeta = Rand(-e_\zeta, e_\zeta)$$

where F_ζ is constant factor; *Rand(a,b)* is a uniformly distributed random value from *a* to *b;* and e_ζ is maximum constant error.
We can estimate the new position of each particle by the following formula:

$$^{new}pos_\zeta = {}^{old}pos_\zeta + \Delta\zeta + N_\zeta + E_\zeta$$

The updated particle position is used to update the position of object in their field of view. In our system, the information for localization mainly comes from the goalposts. For instance, one goalpost found information can guide us a circumference of possible location for the robot, while two goalposts found information can guide us the approximate exact location of the robot.

Robot position confidence value can be calculated by summing all particle's weights before normalizing. For every particle that locates near to the estimated robot position less than a threshold, the confidence value of that particle will be biased with value 1.0. Then, the total confidence value will be normalized (dividing by the maximum value) to be within the range of [0,1].

In order to calculate particle weight, both distance and angle information from landmark (goalpost) observed by robot would be compared to each particle's distance and angle from the given landmark. The formula is based on Gaussian function, as below:

$$f(x) = e^{\frac{-(a_x - b_x)^2}{2c^2}}$$

where a = distance to landmark x, observed by robot (from real world)
b = distance to landmark x, observed by particle
c = constant value (represents standard deviation)
The particle's weight is calculated from the multiplication of the result from the aforementioned function for each found landmark as following:

$$w = f(x_1) * f(x_2) * f(x_3) * \ldots$$

All weights would be normalized so the sum of all particle weight equals to one. When we update each particle to the new location, the particle will be uniformly

distributed. Thus, the particles with higher weight are likely to be selected as the survived particle.
The position of the particle λ with the maximum weight becomes the estimated robot position.

$$\max_{\lambda} \Psi(\lambda) = w_{\lambda}$$

Furthermore, we have explore the idea of using the observed information from other teammate robots, especially from the goalkeeper. Since the goalkeeper stand at the same position most of the time, the observed distance from the goalkeeper to the object and other players is likely to have lower error. From the simulation, the result of localization performance seems to be improved. A further investigation of this collaborative localization technique applied in the real robot still need to be tested. The comparison between non-collaborative and collaborative localization can be shown in Fig 3. When a robot see other robot in the field, the position of the other robot will be registered as an unknown robot. This position will be compared with the position of the known teammate on the stored global map information. The new robot position will be updated as the known robot position when the distance between the robot and the known teammate stored on the global map information is minimum. The stored global map information will be broadcast among all robots in the team.

For each update loop, the teammate position information would be used to merge the current particles locations ρ_{ζ} by the following formula:

$$^{new}\rho_{\zeta} = M \cdot {}^{old}\rho_{\zeta} + (1 - M) \cdot {}^{fri}_{avg}\rho_{\zeta}$$

where M is a merging constant, ${}^{fri}_{avg}\rho_{\zeta}$ is the teammate position that have been

averaged among all perceived teammates. This merging step occurs before all particles weights are updated in the non-collaborative approach. Beside merging, the teammates information can be used when all particle's weights are zero and none of the goalpost has been found in the current update loop. A re-born particle could relocate itself around the position where the robot was last seen by its teammates. Furthermore, if only one goalpost has been found, this information could narrow down the boundary of the particle re-born location.

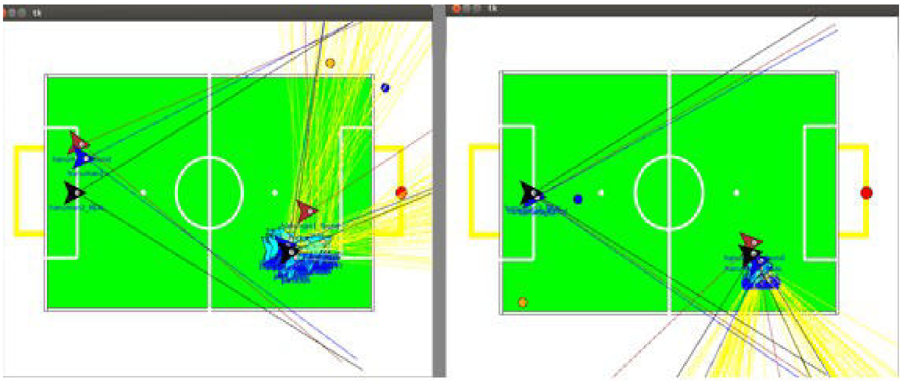

Fig. 3. Simulated localization; Left: each robot performs localization individually; Right: robot utilizes the sharing information from teammates, thus more accurate position estimation was achieved.

4 Game Control and Decision-making

4.1 Game controller

To control the operation during the game, all robots continuously receive the message from the referee's game controller. When the received message indicates the change of the game state, the current operation is terminated. Then the robot loads the new operation which associated with that state.

4.2 Decision Making System

The decision making system of all striker robots can be described by the finite state machine illustrated in Fig 4. The conditions for state changes of each state are determined based on the observation data from the vision system and internal sensor reading of the robot. In addition, the robot shares this information among the teammates by broadcasting mechanism via WLAN network. Therefore, the robot can cooperate to score the game as a team. The goalie decision can be described in a simpler state machine. The goalie is looking for the ball, if the ball heads toward the goalie on either direction (left/right). The goalie will fall in that direction in order to block the ball.

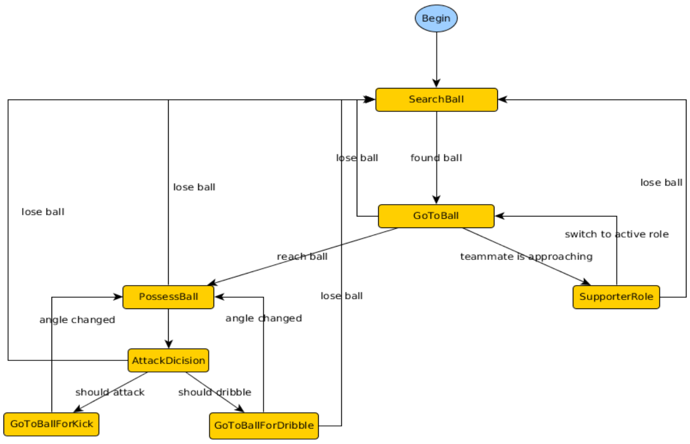

Fig. 4. State diagram of the decision making system for striker.

5 Conclusions

Since last year, our team has developed a stable humanoid robot system that can move fast and make decision quickly like 'Hanuman', the monkey character in Ramayana. This year, we focus our effort on creating more robust localization system that allow more accurate position estimation of the robot, thus results in more intelligent decision to be made in various situations.

References

1. Bruce, J.; Balch, T.; Veloso, M.: Fast and inexpensive color image segmentation for interactive robots. In: Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS' 00), vol.3, pp. 2061-2066. (2000)
2. Jutharee, W., Wanitchaikit, S., Maneechai, B., Kaewlek, N., Khunnithiwarawat, T., Polchankajorn, P., Suppakun, N., Tirasuntarakul, N., Phetruphant, T., Poungrat, A., Wichito, T., Natsupakpong ,S., Maneewarn, T.: Hanuman KMUTT: Team Description Paper. Robocup Humanoid 2013, http://application2013.germanteam.org/upload/0889d267ef49e8db22ea0f8881d37e78a09eb69c/TDP_TeamHanumanKmutt.pdf