Apollo3D Team Discription Paper

Juan Liu, Yue Hao, Hecheng Zhao, Junya Li,
Yue Jia, Ping Shen, Chonghu Zheng
Zhiwei Liang

College of Automation, Nanjing University of Posts and Telecommunications
liujuanhp@gmail.com

Abstract

Apollo3D is a team in RoboCup soccer simulation 3D league. We mainly aim at building a systematical architecture of intelligent and skillful robots. In the newest 11vs 11 version, due to the introduction of sensor noise and the expansion of the soccer field,a more accurate positioning and efficient upper strategy are need in order to avoiding robot being in a disorder. In the past year, our team Apollo3D successful devised a new localization system and a new set of cooperating tactics of the agents. In this paper, we introduce the mechanism of localization, communication and decision making system.

1 Introduction

Apollo Simulation 3D Team was established in 2006, and successfully attended several competitions. We have won the 1^{st} place in Robocup 2010 and the 3rd place in Robocup 2011 recently. The simulated Nao is much like the real one that attracts a large mount of students to devote to this field. Thanks to the devotion and cooperation of these students, several achievements had been achieved in the past years.

With the developing and improving of the RoboCup 3D platform, the number of players has increased to 11, and the field has expanded to 600 square meters. These changes urge us to reconsider the localization and communication problem. On this basis, in order to enhance the overall performance, we redesign the decision making system of our robots. Section 2 will introduce the self-localization of particle filter and how to use Kalman filter to track the ball and other agents. Section 3 will analyze the method that communicate between agents through a channel with limited capacity. Section 4 will discuss the hierarchical role assignment and multi-agent cooperation system.

2 Localization

2.1 Particle Filter Self-localization

Humanoid robot self-localization means estimating the positions and orientations of the local coordinates Σ_v relative to the world frame Σ_w (Fig. 1.This problem involves at least 6 configuration parameters (x, y, z, R, P, Y), and it is hard to build their correlations with the motion model using limited odometers and sensors. Meanwhile, most of the time that the robot actually needs to localize itself are when it walks upright on a flat surface and the hip joints are restricted in a horizontal plane. Thus the z, R, P (height, roll and pitch) of Σ_v are bounded in a small range. So the robot only needs to predict the 2D position $(x, y)$and the heading direction θ.

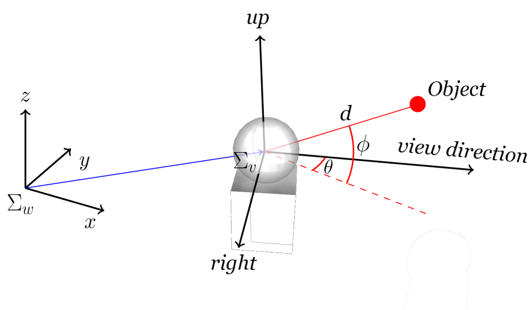

Figure 1: Diagram of the robot vision system

Particle filters estimate the posterior distribution of the state x_t of the dynamical system conditioned on the sensor measurement z_t and control information u_{t-1}, $Bel(x_t) \propto p(x_t|z_t, u_{t-1})$. This posterior can be computed recursively using Bayes rules and partially observable controllable Markov chains:

$$\overline{Bel}(x_t) = p(x_t|x_{t-1}, u_{t-1})Bel(x_{t-1}) \quad (1)$$

$$p(x_t|z_t, u_{t-1}) = \mu p(z_t|x_t)\overline{Bel}(x_t) \quad (2)$$

Equation (1) is called motion update phase. where the robot needs to predict the new state of position and orientation x_t basing on its motion u_{t-1} according to its odometers and the last state $Bel(x_{t-1})$. Equation (2) is the observation update phase. In this phase, the robots update to the current state on condition of the measurement of the sensors z_t.

The key idea of the particle filter is to represent the posterior $p(x_t|z_t, u_{t-1})$ by a set of weighted state samples:

$$S_t \quad = \quad \{\langle x_t^{(i)}, w_t^{(i)} \rangle\}_{i=1,\ldots,n} \tag{3}$$

where each $x_t^{(i)}$ stands for an instance of estimated state with $w_t^{(i)}$ being its weight.Theoretically, as $N \to \infty$ the distribution of these samples match the density of the posterior. In practice, we use 1000 particles to approximate the posterior. Algorithm 1 shows the details.

Algorithm 1 Partile_filter(S_{t-1}, u_{t-1}, z_t):

$S_t := \emptyset$, $N = 1000$, $w_{total} = 0$
for $i := 1$ to N **do**
  draw index $j(i)$ with probability $\propto w_{t-1}^{(j(i))}$ in S_t
  $x_t^{(i)} :=$ **motion_model**$(u_{t-1}, x_{t-1}^{(j(i))})$
  $w_t^{(i)} := p(z_t|x_t^{(i)})$
  $w_{total} := w_{total} + w_t^{(i)}$
  $S_t := S_t \cup \{\langle x_t^{(i)}, w_t^{(i)} \rangle\}$
end for
for $i := 1$ to N **do**
  $w_t^{(i)} := w_t^{(i)} / w_{total}$
end for
return S_t

Fially, the algorithm returns S_t, we simply calculate the average of $x_t^{(i)}$ to estimate the state at time t.

2.2 Kalman Filter Tracking

In RoboCup3D environment, the position of the ball and agents keep changing all the time. If each individual agent can accurately predict other agents' movement, it will better seize the initiative. Especially at the risk of opponents shooting, our goalie's quick reponse to stop the ball largely depend on its prediction of the velocity of the ball. The Kalman filter not only can increase the accuracy of tracking other objects, but also can help predict their other states like velocity.

3 Communication System

In RoboCup 3D simulation robot soccer games, 11 parallel agent processes cannot communicate directly.Instead, the server transmits the message through broadcasting. Only one agent can send message every two cycles, and the other agents receive it in the next cycle. The capacity of the channel is limited in the 20 byte ASCII message. In addition, not all ASCII characters are supported.

To compensate the restriction of the robot vision and make full use of the communication channel,we let the agents take turns to send information according to their own number order. Sharing other players' position and role information, each agent can precept the positions of unseen teammates and ball, and synchronize the roles, which are essential for both defending and offending. The format of messages is shown in Figure 2.

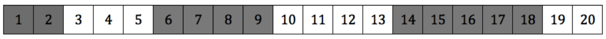

Figure 2: The format of 20 byte message.

- 1^{st} and 2^{nd} bytes: team signature.
- 3^{rd} byte: the player number of the sender.
- 4^{th} and 5^{th} bytes: binary variable such as if the player is fall or can see the ball.
- 6^{th} to 9^{th} bytes: ball position in sender's perception.
- 10^{th} to 13^{th} bytes: self position in sender's perception.
- 14^{th} to 18^{th} bytes: all players roles in sender's perception.
- 19^{th} and 20^{th} bytes: reserved bytes.

Notice that to compress the data, the field is divided into $5000 * 5000$ grids. We use the '*' to '$\sim$' total of 83 ASCIIs to encode our data, and thus we can transmit 83^{20} bit information.

4 Tactic

The team size of RoboCup 3D growth from 6 in 2010 to 9 in 2011, and finally 11 last year, which raised the concern of better multi-agent corporation.

Increased the number of players though, the robot who is on the ball is unique for any single moment. So far, distant passing skills between robots are still impractical for most teams, how the dribbler control the ball becomes the key to win a game.

The dribbler, called the Hero role in our model, bears the heaviest burden in a competition. In the tactic of Apollo3D, agents first select a formation according to the position of the ball, then choose a Hero. Since the vision of agents is restricted, and it has errors that every agent's perception of self and other players' locations, multiple Heroes may appear at the same time, causing chaotic collisions around the ball, which brings negative effects on controlling the ball. To solve this issue, we employed a voting method to select the best Hero, using the communication system to synchronize each agent's selection. Since it has time delays in the communication system, and agents cannot 100% sure about its selection, we gives each vote a weight describe by a probability value between $(0, 1)$.

When the Hero is dribbling, we make sure that every other player stay on a ascendant position to assist attacking. Each role (or position) is assigned according to the robot's location in the current formation. Meanwhile, we also have to synchronize other roles among agents, to prevent potential risk of collision and keep the order of attacking.

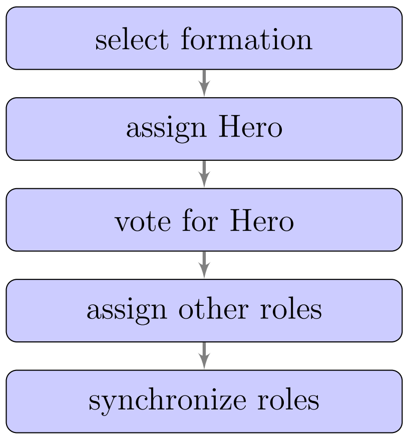

Figure 3: Flow chart of assigning roles

Here we use the following criteria for choosing the Hero:

- Whether the player is fall down.
- Whether the ball is visible to the player.
- Player's distance to ball.

- Whether the player is in front of the ball or behind it (players in front of the ball often need extra time for turning).
- Whether the player is goalie (competition rule stipulated the goalie has to be NO.1).
- Whether this player is Hero in last cycle.

5 Conclusion

In this paper, we discussed algorithms in mobile robot localization and multi-agent corporation. We proceed a large amount of experiments, and the results validated the reliability and superiority of these algorithms. Research on humanoid robot has gained popularity in Robotics, many researchers and engineers focus their research on this field. Individual skills, such as walking planning, will still dominate the RoboCup 3D game in the future. Our further work will focus on improving the controlling of the robot motions and stability of walking.

References

[1] Christopher G. Atkeson, Andrew W. Moore, and Stefan Schaal. Locally weighted learning. *Artif. Intell. Rev.*, 11(1-5):11–73, 1997.

[2] Dieter Fox. Kld-sampling: Adaptive particle filters. In *NIPS*, pages 713–720, 2001.

[3] A.; Ro 04fer T.; Graf, C.; Ha 04rtl and T. Laue. A robust closed-loop gait for the standard platform league humanoid. *Proceedings of the 4th Workshop on Humanoid Soccer Robots A workshop of the 2009 IEEE-RAS Intl. Conf. On Humanoid Robots (Humanoids 2009)*, 2009.

[4] E. Hashemi, M.G. Jadid, M. Lashgarian, M. Yaghobi, and R.N.M. Shafiei. Particle filter based localization of the nao biped robots. In *System Theory (SSST), 2012 44th Southeastern Symposium on*, pages 168 –173, march 2012.

[5] Qiang Huang, Kazuhito Yokoi, Shuuji Kajita, Kenji Kaneko, Hirohiko Arai, Noriho Koyachi, and Kazuo Tanie. Planning walking patterns for a biped robot. *IEEE Transactions on Robotics*, 17(3):280–289, 2001.

[6] P. Maybeck. The Kalman filter: An introduction to concepts. *Autonomous Robots*, 1990.

[7] Andrew Y. Ng, H. Jin Kim, Michael I. Jordan, and Shankar Sastry. Autonomous helicopter flight via reinforcement learning. In *NIPS*, 2003.

[8] Simon Thompson, Satoshi Kagami, and Koichi Nishiwaki. Localisation for autonomous humanoid navigation. In *Humanoids*, pages 13–19, 2006.

[9] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. MIT Press, Cambridge, Mass., 2005.

[10] Xinxin Wang, Xuesong Yan, Yongshan Zhang, and Siyu Pan. Kalman filter in the robocup 3d positioning. In *Computer Science and Electronics Engineering (ICCSEE), 2012 International Conference on*, volume 3, pages 47 –52, march 2012.

[11] P. MacAlpine, D. Urieli, S. Barrett, S. Kalyanakrishnan, F. Barrera, A. Lopez-Mobilia, N. S 05tiurc 02a, V. Vu, and P. Stone. UT Austin Villa 2011 3D Simulation Team report. Technical Report AI11-10, The Univ. of Texas at Austin, Dept. of Computer Science, AI Laboratory, December 2011.