BabyTigers 2003: Osaka Legged Robot Team

Noriaki Mitsunaga*+, Hideki Toichi*, Taku Izumi@, and Minoru Asada*+

{mitchy,toichi,izumi}@er.ams.eng.osaka-u.ac.jp, asada@ams.eng.osaka-u.ac.jp

*Dept. of Adaptive Machine Systems, Graduate School of Engineering, Osaka University,

+Handai Frontier Research Center,

@ Former student of the Graduate School of Engineering, Osaka University

2-1 Yamadaoka Suita, Osaka, 565-0871, Japan

1 Introduction

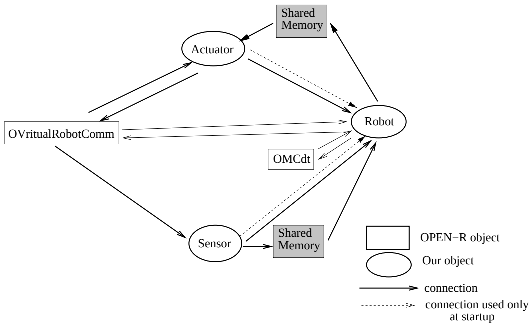
We are interested in learning issues such as action selection, observation strategy without 3D-reconstruction [1, 2, 3], emergence of walking, and cooperative behaviors. Basic implementations, such as vision, actions, and walking, are mostly derived from our previous work [4] and the actions and walking of the United Team of Kyushu (part of their work was based on the University of New South Wales one). Changes from last year are the following:

- speeding up of software color detection,
- import of walking and shooting from ASURA 2002,
- import of multi hypothesises tracking from CM-Pack'01,
- some strategy changes for attacker.

We will briefly describe our technical approach of this year in the sections from 2 to 6. Then we briefly describe our current work in cooperative behaviors in the section 7.

2 Inter object communication architecture

Figure 1 shows our inter object communication architecture (same as 2002). The *robot object* reads the information of the sensors from the shared memories, receive camera image from *OvirtualRobotComm*, determines its action, and writes it into the shared memory for *actuator object*. The *sensor object* preprocess the information from the sensors in back ground from the view point of *robot object*. The *actuator object* receives the action command from the shared memory and outputs joint angles every 8[ms] to the *OvirtualRobotComm*. In order to change the resolution of the camera image to be processed according to the game situation, we implemented the image processing in *robot object*.



,

Figure 1: Inter object communication between OPEN-R objects.

3 Color detection and object recognition

Since we adopted mostly vision based reactive approach, the frame rate of the image is very important. Therefore, we used both color detection by hardware (CDT) for 88×71 image and a software color detection mechanism for the higher resolution (176×143) image. We used a lookup table for color detection. Color detection is done by the following function inspired by the code of team ASURA (2002)[5]:

```
void do_tbldetect (byte *result, const byte *y,
 int width, int height, byte ***tbl)
{
  unsigned long *p = (unsigned long *)result;
  const byte *u = y + width;
  const byte *v = u + width;
  int skip = width*2;

  width /= 4;
  for (int i=height; i>0; i--) {
    for (int j=width; j>0; j--) {
      byte c0 = tbl[*(y++)][*(u++)][*(v++)];
      byte c1 = tbl[*(y++)][*(u++)][*(v++)];
      byte c2 = tbl[*(y++)][*(u++)][*(v++)];
      byte c3 = tbl[*(y++)][*(u++)][*(v++)];
      *p  = c0 | c1 << 8 | c2 << 16 | c3 << 24; // little endian
      p ++;
    }
    y += skip;
    u += skip;
    v += skip;
  }
}
```

The result of the color detection is returned in `byte *result`. To reduce the use of memory, the Y is quantized into 32 levels as in CDT. The pointers of the same Y level refer to the same `byte **`. The size of the table is about 2MB. Although we can specify arbitrary color detection area and up to 255 colors, we chose the rectangle area and 8 colors to keep the compatibility with CDT.

We used the same object recognition mechanism as in previous years. After the color detection, the eight neighbor concatenated blobs are detected, then blobs are labeled by the object names.

After object recognition, localization is done based on the multi-hypothesises tracking by the Kalman-filter of CM-Pack'01 [6]. However, it seems that the precision was not so good as other teams. We are investigating the reason.

4 Walking, Kicking, Diving, and Getting up

Currently we have five type of walking styles. We designed first three styles like followings. First we roughly specified the trajectories in joint angles without using the inverse kinematics, then we tuned the parameters by hand or the Genetic Algorithm (GA) based on the real robot.

Last year we designed another walking by specifying the rectangle trajectories of the soles so that the body motion can be smooth and the image changes can be reduced. Then, we tuned the parameters for walking by hand.

For the fifth walking, we imported UNSW type walking from ASURA 2002 from this year. We use mostly same patterns but some part of the code were re-written so that it will fit to our program.

We used kicking, diving, and getting up motions developed by us and some shooting behaviors were imported from ASURA 2002.

5 Strategies of the goal keeper and attackers

The strategy of the goal keeper is the same one used in last year's competition. Figure 2 shows the state transition of the goal keeper. The keeper did not use any explicit self-localization.

The attacker's strategy was simple one. The attacker tries to find the ball by circling in the field. When it finds the ball, it approaches and it stops before the ball. Then it looks around to check the direction and then shoot the ball. We used self-localization to keep illegal defender rule and determine the direction to shoot the ball.

6 RoboCup Challenge

6.1 Challenge 1 (B/W ball)

We tried two methods. One is the histogram based and the other is Hough transform based method. The histogram based method compares the histograms

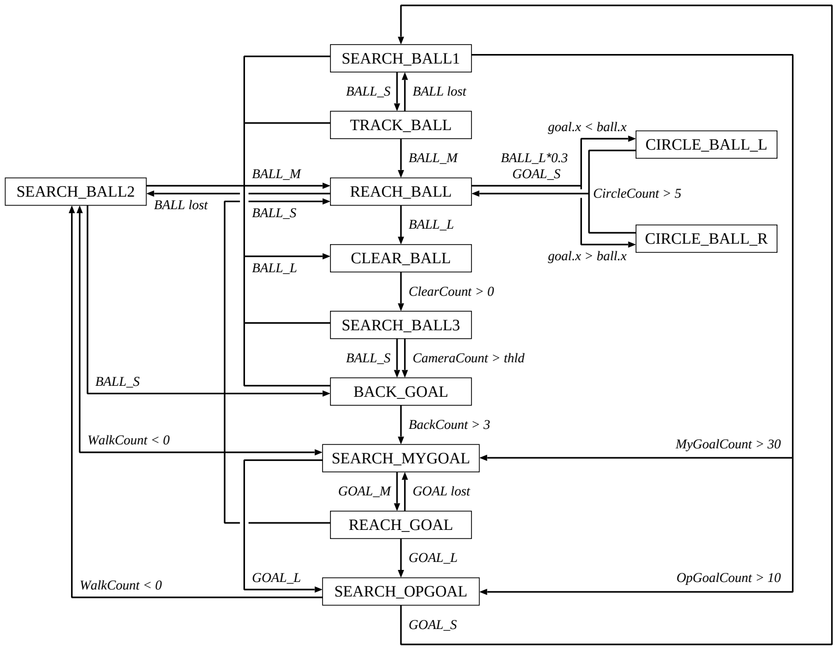


Figure 2: State transition map of the goal keeper.

with the template histogram of the ball. Since it compares the histograms of the various sizes and locations of rectangle in an image, we need not care for the size of the ball in the image in theory. Drawbacks are a) it sometimes make mistakes such as the ball is on the white side walls even if it is not (Figure 3, b) it is not good at finding a small ball image compared with current color based method, and c) calculation is slow. Although fast search method by Murase and Vinod [7] accelerated the speed by 10-100 times, it was still slow for full search in 176×144 image.

The Hough transform based method first detect edges and then execute Hough transform for circles. Figure 4 shows an example detection. The gray lines in Figure 4 (c) are the candidates of the ball circle and the white line is the best candidate. Drawbacks are a) it still make mistakes with white lines and the center circle, though it is better than histogram based method, b) it is not good at finding a small ball image, and c) calculation is slow. We tried to use the Hough transform based method in the Challenge but due to bug in the program, we used the histogram based one.

6.2 Challenge 2 (Localization without poles)

We used the multi hypothesis tracking method as in the games. We added to use the distance to the goal corner on the field but did not use the lines nor

(a) success

(b) failure

Figure 3: Results of the black and ball searching with histogram based method.

walls. The result of the localization was not good.

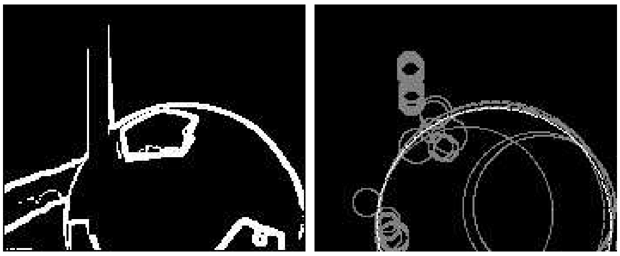
6.3 Challenge 3 (Obstacle avoidance)

Obstacle avoidance was done by following:

1. always rotate the head to watch around,
2. mark up the regions which are not the field in the camera image by the color,
3. project the points in the image onto the field and mark the occupancy grid (grid size is 50[mm]×50[mm]),
4. go forward, left, or right dependent on the size of free region.

The grid size was determined so that it will ignore the lines on the field. Although we need to accelerate the computation to use in the games (it was not computed at full frame rate), the performance of the method was good as shown in the Challenge.

(a) original image



(b) edge detected (c) result

Figure 4: An example of the black and ball searching with Hough transform based method.

7 Cooperative Behavior based on a Subjective Map with Shared Information

7.1 Background

In a multi robot system, it is expected that communications or information sharing between robots help acquiring the knowledge about their environment. When multi robots communicate with each other, they seem to need a reference coordinate system to exchange their information about the environment. In case of mobile robots, the world coordinate or the coordinate fixed to the environment are often used. To convert the observation to the world coordinate, each robot localizes itself. Assuming that localization errors are small or neglectable, information exchange between robots have no problem. However, localization errors often become too large to ignore.

Methods to localize itself and to acquire knowledge about environment by shared information from other robots are proposed [8, 9]. They use geometric constraints between several robots which are commonly observed from each other, calculates their positions, and share the environment map. The errors of self-locations are minimized so that shared observations conform to geometric constraints. To observe several robots at one time, they used omni-directional cameras rather than normal cameras with limited view angles. However, in case of robots with the latter cameras, there will be many situations that they will not be able to observe others. Then it becomes difficult to use such methods.

Although the representation of self-location by probabilistic form and use of beliefs are proposed and commonly used in order to handle the error of self-localization, it is difficult to obtain the accurate model to merge the maps by two or more robots and to maintain the shared map. Simple weighted average of information by robots may work when the errors are small. However when one of the robots has a large error on its self-localization it will affect the shared map used by all other robots. Designing weights or accuracy measurements to prevent such a case is difficult because there are always errors that is unknown to the designer and in many situations errors are not detectable by the robot itself. Then, we propose a subjective map based approach rather than shared map based one. A subjective map is for a multi-agent system to make decisions in a dynamic, hostile environment. It is maintained by each robot regardless of the objective consistency of representations among other agents. Owing to its subjectivity, the method is not affected by other agent's information which may include not negligible errors due to dynamic changes in the environment caused by accidents. A potential field is defined on the subjective map in terms of subtasks such as ball reaching and shooting, and is dynamically updated to make a decision to act.

7.2 A subjective map generation

Let us assume that there are two robots (robot A and robot B) and one object (a ball) in an environment. These robots have localized themselves and they are watching at the ball but they cannot observe each other due to their limited view angles (Fig.5). If we ignore the localization errors and put information on a map, there will be contradiction about the ball position as shown in Fig.6(a).

If we use the weighted average of the ball location $\hat{\boldsymbol{x}}$,

$$\hat{\boldsymbol{x}} = \frac{{}^{B}\sigma {}^{A}\boldsymbol{x} + {}^{A}\sigma {}^{A}\boldsymbol{x}}{{}^{A}\sigma + {}^{B}\sigma}, \tag{1}$$

where ${}^{A}\boldsymbol{x}$, ${}^{B}\boldsymbol{x}$, ${}^{A}\sigma$, ${}^{B}\sigma$ are the ball positions and their deviations estimated by robots A and B, respectively, assuming Gaussian distributions. Then we have a map shown in Fig.6(b). There is no contradiction in this map. However, this is not the true ball position in the world coordinate system. When robot A has correct estimation while robot B has incorrect one, robot A's estimation become worse because of the information sharing. Also there are cases the relative position to the robot itself is more important than the absolute position in the world coordinate system. Further, it becomes more complex when the robots can observe each other. If we can assume the simultaneous observations from several robots are available then we could use geometrical constraints and

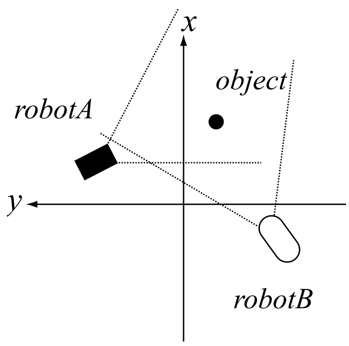


Figure 5: There are two robots watching at a ball.

reduce errors [8, 9]. In case of robots with limited view angle cameras and they are moving, we cannot assume it.

Here, we propose that each robot constructs its subjective map and determines its action based on it. For example, robot A believes its observation for the ball and calculate position of robot B from the relative position between the ball and robot B as,

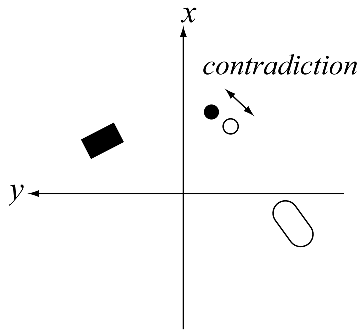
$$
\begin{aligned}
{}^{A}\hat{\boldsymbol{x}}_{A} &= {}^{A}\boldsymbol{x}_{A}, & (2)\\
{}^{A}\hat{\boldsymbol{x}}_{Q} &= {}^{A}\boldsymbol{x}_{ball}, & (3)\\
{}^{A}\hat{\boldsymbol{x}}_{B} &= {}^{B}\boldsymbol{x}_{B} + ({}^{A}\boldsymbol{x}_{ball} - {}^{B}\boldsymbol{x}_{ball}). & (4)
\end{aligned}
$$

Fig.7 shows the subjective maps of robots A and B. With these subjective maps, although reduction of localization error is not achieved, robot A is not affected by the localization error of robot B and can use the information from robot B. The subjective map method is expected to work for such a task and the environment that the relative positions is important rather than absolute ones, localization errors sometimes become large, and geometrical constraints are hard to use. In the following experiments, we compare the action decisions by shared map with average position method and decisions by subjective maps in a robot soccer environment with four legged robots. Robots determine their actions from potential field calculated from a shared or its subjective map.

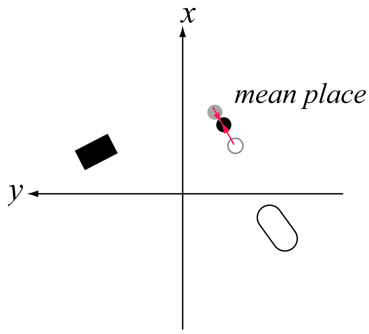
7.3 A potential field for making decisions

We define a potential field which depends on a subjective map of a robot. Each robot calculates the field from the map and determine its action according to the field. A robot has four actions, move forward, turn left, turn right and shoot a ball. It takes such an action that climbs the potential field if the ball is far and shoot it to the opponent goal as it approaches close.

The potential field $V(x, y)$ of robot i consists of three potentials. One is V_F which is the function of the position of a teammate j, ${}^{i}\boldsymbol{P}_j = ({}^{i}x_{R_j}, {}^{i}y_{R_j})$.



(a) with assumption of no localization error



(b) with averaged locations

Figure 6: Constructed map with assumption of no localization error and with averaged locations

The second is V_O which is the function of the position of an opponent k, ${}^i\boldsymbol{R}_k = ({}^i x_k, {}^i y_k)$. The last one is V_B which is the function of the ball position ${}^i\boldsymbol{Q} = ({}^i x_Q, {}^i y_Q)$. All the positions are derived from its subjective map. In the following, we give example potentials based on the setup shown in Fig.8.

Potentials by a teammate V_F and opponent V_O are calculated by,

$$V_F(x,y) = -\sum_{j(j\neq i)} f({}^i\boldsymbol{P}_j), \tag{5}$$

$$V_O(x,y) = -\sum_{k} f({}^i\boldsymbol{R}_k), \tag{6}$$

where,

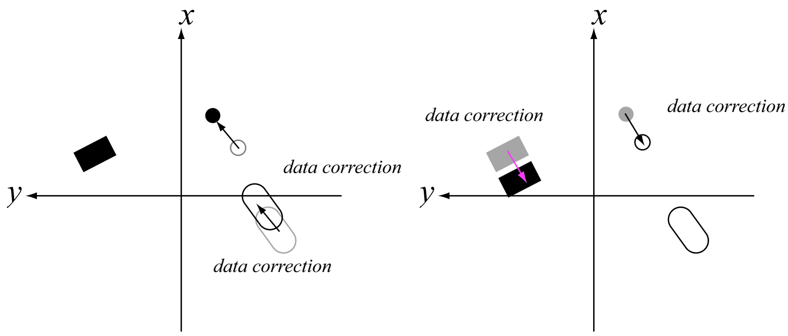
$$f(\boldsymbol{P}(\bar{x},\bar{y},\sigma_x,\sigma_y)) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-\frac{1}{2}\cdot\left(\left(\frac{x-\bar{x}}{\sigma_x}\right)^2+\left(\frac{y-\bar{y}}{\sigma_y}\right)^2\right)}. \tag{7}$$

These potentials are to avoid robots in the field. Fig.9 shows the V_F and V_O of robot A in the example setup.

The potential from the ball is defined so that the robot closer to the ball will reach ball, others will go to the position where it can back up the shoot. The potential function of robot i is switched depending on if i is the closest to the ball or not,

$$V_B(x,y) = \begin{cases} f({}^i\boldsymbol{Q}) & (\textit{robot } i \textit{ is the closest to the ball}), \\ f({}^i\boldsymbol{Q}') & (\textit{otherwise}), \end{cases} \tag{8}$$

where ${}^i\boldsymbol{Q}$ is the position of the ball, and ${}^i\boldsymbol{Q}'$ is the support position. ${}^i\boldsymbol{Q}'$ is



(a) the subjective map of robot A (b) the subjective map of robot B

Figure 7: The subjective map of robot A and robot B.

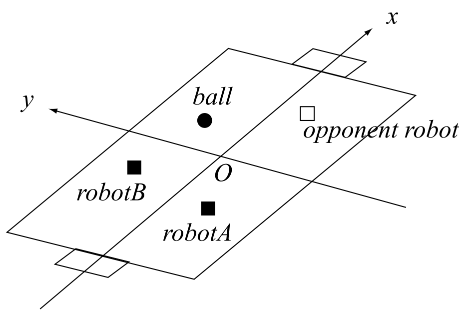


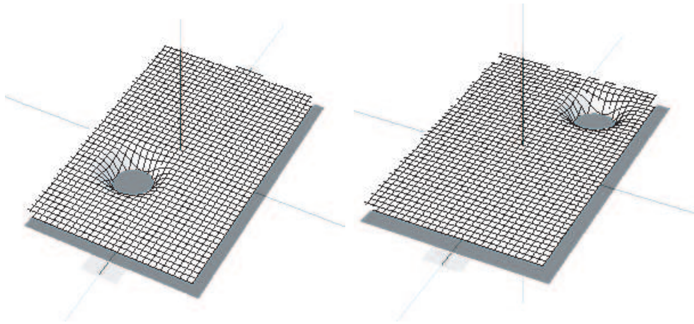
Figure 8: True object locations

defined as,

$$ {}^{i}\boldsymbol{Q}' = \frac{1}{2}\left({}^{i}\boldsymbol{Q} + \boldsymbol{G}\right), \tag{9} $$

where $\boldsymbol{G}$ is the position of target goal. The example potentials of robots A and B are shown in Fig.10. Final potential fields are shown in Fig.11.

Although we did not use this approach for the games in 2003, we have been experimenting this approach with the robots and the field for RoboCup SONY Legged Robot League. Experimental results will be presented in elsewhere.



(a) V_F of robot A (b) V_O of robot A

Figure 9: Potential diagram V_F and V_O of robot A.

Acknowledgment

We used the codes shared among the teams involved in RoboCup SONY Legged robot league. We thank Carnegie Melon University for their self-localization codes, the United Team of Kyushu for their motion and walking codes, and the University of New South Wales for their walking codes.

References

[1] Noriaki Mitsunaga and Minoru Asada. Observation strategy for decision making based on information criterion. In *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1038–1043. 2000.

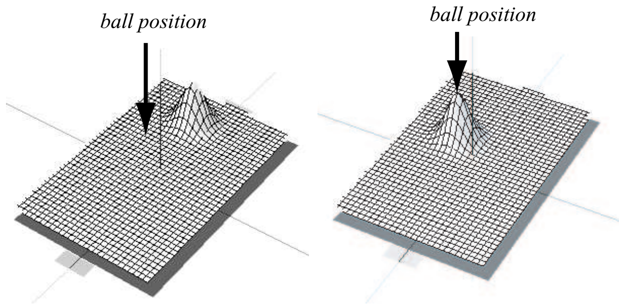
[2] Noriaki Mitsunaga and Minoru Asada. Sensor space segmentation for visual attention control of a mobile robot based on information criterion. In *Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1714–1719. 2001.

[3] Noriaki Mitsunaga and Minoru Asada. Visual attention control for a legged mobile robot based on information criterion. In *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 244–249. 2002.

[4] Noriaki Mitsunaga, Yukie Nagai, Tomohiro Ishida, Taku Izumi, and Minoru Asada. BabyTigers 2001: Osaka legged robot team. In Andreas Birk, Silvia Coradeschi, and Satoshi Tadokoro, editors, *RoboCup 2001: Robot Soccer World Cup V*, pages 685–688. Springer, Lecture Note in Artificail Intelligence (2377), 2002.

[5] http://www.asura.ac/

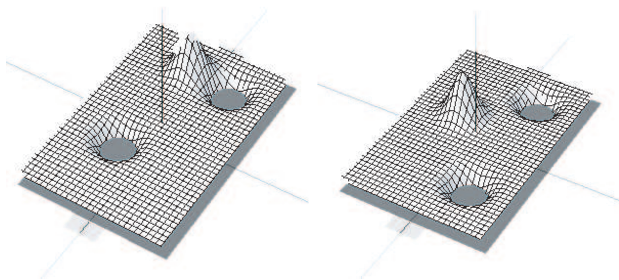
[6] William Uther, Scott Lenser, James Bruce, Martin Hock, and Manuela Veloso. CM-Pack'01: Fast legged robot waking, robust localization, and team behaviors. In Andreas Birk, Silvia Coradeschi, and Satoshi Tadokoro, editors, *RoboCup 2001:*



(a) V_B of robot A, it is highest at support position

(b) V_B of robot B, it is highest on the ball

Figure 10: Potentials from the ball (V_B)



(a) robot A

(b) robot B

Figure 11: Final potential fields of robots

Robot Soccer World Cup V, pages 693–696. Springer, Lecture Note in Artificail Intelligence (2377), 2002.

[7] H. Murase and V. V. Vinod. Fast visual search using focussed color matching - active search -. *Journal of the Institute of Electronics, Information and Communication Engineers D-II (in Japanese)*, J81-D-II(9):2035–2042, 9 1998.

[8] K. Kato, H. Ishiguro, and M. Barth. Identifying and localizing robots in a multi-robot system environment. In *Proceedings of the 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 966–971, 1999.

[9] T. Nakamura, A. Ebina, M. Imai, T. Ogasawara, and H. Ishiguro. Real-time estimating spatial configuration between multiple robots by triangle and enumeration constraints. In *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2048–2054, 2000.