SHARPKUNGFU TEAM DESCRIPTION 2006

QINING WANG*, CHUNXIA RONG*, YAN HUANG*, GUANGMING XIE*, LONG WANG*

*Intelligent Control Laboratory, College of Engineering,
Peking University, Beijing 100871, China
http://www.mech.pku.edu.cn/robot/fourleg/

Emails: qiningwang@pku.edu.cn, rongchunxia@163.com, huangyan1985@sina.com,
xiegming@mech.pku.edu.cn, longwang@pku.edu.cn

Abstract— This report is a short description of the four-legged robot soccer team sharPKUngfu with the specification of our work in 2006. This year, our sharPKUngfu Team participated in the Technical Challenge of RoboCup 2006. In this event, our Medal Awarding challenge got the $8th$ place in the Open Challenge and $16th$ in the sum. In October, we got the champion in RoboCup China Open 2006. We focus our research on collaborative localization in dynamic environment, quadrupedal gaits optimization and intelligent behaviors. To improve robot self-localization under more natural conditions, we create an experience-based collaborative localization approach to help robot localize under more natural conditions. Moreover, a Particle Swarm Optimization approach to generate high-speed quadrupedal gaits is proposed. In behavior module, our dynamic limit cycle based approach for obstacle avoidance is shown in detail. To perform better under the rule 2006, we create our own set of special actions and correlative strategies.

Keywords— Four-legged, localization, multi-robot, RoboCup, gait.

1 Introduction

This year, we focus our research on collaborative localization in dynamic environment, quadrupedal gaits optimization and intelligent behavior. To improve robot self-localization under more natural conditions, we create an experience-based collaborative localization approach to help robot localize under more natural conditions. Moreover, a Particle Swarm Optimization approach to generate high-speed quadrupedal gaits is proposed. In behavior module, our dynamic limit cycle based approach for obstacle avoidance is shown in detail. To perform better under the rule 2006, we create our own set of special actions and correlative strategies. Our code this year is based on GT2004, which is shared by the German Team.

In the following section, we show the improvement in localization with specification of human cognition inspired collaborative approach. In section 3, we present the work of our team in quadruped gaits optimization, especially of how to implement Particle Swarm Optimization approach in high-speed forward gaits generation. In section 4, strengthened collaboration among teammates is introduced. Moreover, improved obstacle avoidance method based on dynamic limit cycle in sharPKUngfu 2006 is presented.

2 Localization

To imitate human behaviors or even cognition procedures, most intelligent robots are designed to localize actively and automatically. However, it seems hard for robots to localize in a complex terrain especially an unknown one. To solve this problem, several methods have been applied for localization these years. They mainly differ in using probability to present the possible position and applying various types of sensory information. For example, [6] creates the probabilistic approach based on Markov chain, named Monte Carlo Localization (MCL). For vision-based robot, [5] applies the landmark based MCL to localize in dynamic environment. [3], [9] present the cooperative methods for autonomous position estimate. Whereas those landmark-based methods seem not sufficient if there is no such well recognized landmark considering the odometry error especially when a collision occurs. In that circumstance, the convergence of probability update procedure using MCL is not so satisfied. It may take quite a long time to work out the correct position. A recent work explained in [8] uses the approach of combining an image retrieval system with the Monte Carlo localization. However, the computational cost of this approach is expensive. Besides, the requirement of building a huge database is not so practical, especially in the complex environment.

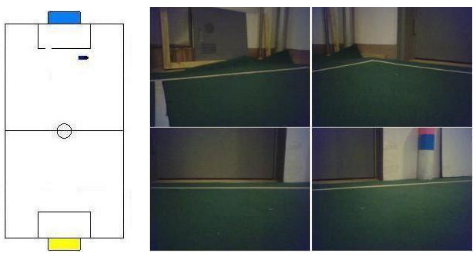
The cooperation in self-localization among multiple robots has many applications in real robot systems (see [7] for overview). For instance, [4] introduces a method for multi-robot localization with certain preconditions. Such robot systems need to identify individual robots. It is quite difficult to perform collaborative localization for robots dealing with situations where they can detect but not identify other robots. In addition, taking the uncertainty of sensors into account, the result of detecting individual robot is not so reliable. Those limitations of the approach make it not so applicable for real robots localizing in complex environments.

In the sharPKUngfu2006, we create a human cognition inspired collaborative approach that combines image database for experience without landmarks and real-time sensor data for a group of vision-based mobile robots to estimate their positions. We use the team message of dynamic reference object to improve the Markov localization for multiple mobile robots. On the one hand, our approach presents a fast and feasible system for vision-based mobile robots to localize in the dynamic environment even if there is no such recognized landmark to help. On the other hand, we show the collaborative method with introduction of Dynamic Reference Object to improve the accuracy and robustness of self localization, even in the circumstance that the robot can not localize individually or has no idea of who is nearby. Specific analysis can be found in [1].

Our method has been implemented on the sharPKUngfu Team in lab experiments and real soccer competitions. When the localization experiment is conducted in lab, the robot performs a scanning motion with its head (pan range $[-45°, 45°]$) to search landmarks and exploit experience.

2.1 *Individual Robot Localization*

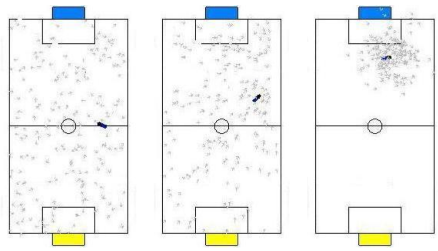
In this experiment, we use only one four-legged robot to perform localization in RoboCup environment. The robot is placed in the center of the field at first. Then we let robot walk to one corner of the field. On the way to the corner, we add *artificial collisions* to effect the odometry in a negative way. Specifically, we pick the legged robot up for a while. This procedure makes the odometry not so reliable to imitate real dynamic environment outdoor. After that, the robot stands at one point (shown in Fig. 1) where both landmark perception and experience exploiting can be activated. Then robot performs a head scanning mo-



(a) (b)

Figure 1: Real position for individual localization experiment. The solid symbol on the field presented in (a) is the real position of the robot. When stands at this position, the robot can obtain vision information using head camera within the head pan range. (b) is selected images in the view at the position when camera heading directions are $-45°$, $-10°$, $10°$ and $45°$.

tion to test different localization approach. If we only use landmark based MCL, specifically detecting beacons as new sensory information, the probability distribution converges not satisfied. The result of only using landmark based localization is shown in Fig. 2(b), where the robot can not get the localization result immediately. Then we use our hybrid system with landmark and experience based Markov localization. Using our approach, the robot can localize well after 5-10 seconds on average. The experimental result is shown in Fig. 2(c).



(a) $t = 0s$ (b) $t = 10s$ (c) $t = 10s$

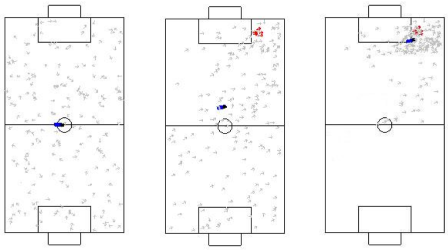
Figure 2: Comparison of particle distribution between our approach and only landmark(beacon) based method. Solid arrows indicate MCL particles(100). The calculated robot position is indicated by the solid symbol. (a) is the initial uniform distribution. (b) is the calculated result of using only landmark based MCL. (c) is the result of applying our approach.

2.2 *Collaborative Localization*

In this experiment, the orange ball used in the four-legged league is considered as the *dynamic reference object*. We use three robots to perform multi-robot localization. Every robot uses the hybrid system tested in the individual experiment mentioned above. We set one of the three robots as a sample to estimate our collaborative approach. The other two robots move randomly to catch the ball and broadcast the ball position with position possibilities. We receive the calculated result from the sample robot. To imitate the outdoor environment, this robot stands in a certain position on the field where we eliminate the landmark which the robot can easily detect. Only experience and collaboration can help the robot localize. The localization result of the sample robot which has used the collaborative approach is shown in Fig. 3. The probability distribution can converges quickly after 3-9 seconds when the dynamic reference object is taken into account.

3 Gaits Optimization

In sharPKUngfu 2006, we implement the Particle Swarm Optimization (PSO) based approach in high-speed gaits generation of Aibo. In our method, different to many existing gait optimization methods based on Genetic Algorithm, the initial values of parameters can be selected randomly



(a) $t = 0s$ (b) $t = 3s$ (c) $t = 9s$

Figure 3: The localization result of applying collaborative approach with dynamic reference object. Solid arrows indicate MCL particles(100). The calculated robot position is indicated by the solid symbol. (a) is the initial uniform distribution. (b) is the calculated result after 3 seconds. (c) is the well localization result after 9 seconds.

from a rational range. Those initial values need not any hand-tune parameters. To avoid motor abrasions of irrational gaits in first two generations, we artificially set fitness (forward speed) to be zero. At the 10th iteration, we have already achieved $390mm/s$ gait for Aibo, which is very impressive. Fig. 4 shows the specific result of optimization. By using this method, high-speed gaits can be generated relatively fast with no need of preset initial values. Specific implementation of our gaits optimization method can be found in our team report 2006 which is available on our web site.

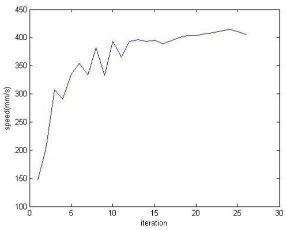


Figure 4: Forward gait optimization result. x-axis represents the iterative times, while y-axis is the best forward speed in each generation. The fastest forward speed in this experiment is $416mm/s$.

4 Behaviors

4.1 Obstacle Avoidance

In sharPKUngfu 2006, we introduce time-variable limit cycle to help robot avoid obstacles. To show the approach, we simply describe the shape of Aibo as a cycle in the two dimensional plane. Considering the following nonlinear system for dynamic limit cycle applying in Aibo:

$$\begin{aligned} \dot{\widetilde{x}} &= \rho(\widetilde{y} + \gamma\widetilde{x}(\frac{1}{4}\overline{v}^2 - \widetilde{x}^2 - \widetilde{y}^2)) \\ \dot{\widetilde{y}} &= \rho(-\widetilde{x} + \gamma\widetilde{y}(\frac{1}{4}\overline{v}^2 - \widetilde{x}^2 - \widetilde{y}^2)) \end{aligned} \quad (1)$$

where ρ is the character factor of the obstacle which is set to be a positive value. γ is the convergence factor. And $\overline{v}$ is the relative velocity to the obstacle which is dynamic when the robot moves. The size of limit cycle is changing when system(1) switches. To prove the circle $\widetilde{x}^2 + \widetilde{y}^2 = \frac{1}{4}\overline{v}^2$ is the dynamic limit cycle of the switched system(1), we use the common Lyapunov function:

$$V(\widetilde{x}, \widetilde{y}) = \widetilde{x}^2 + \widetilde{y}^2 \quad (2)$$

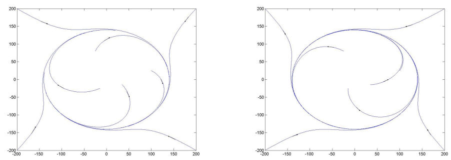
such that:

$$\dot{V}(\widetilde{x}, \widetilde{y}) = 2\rho\gamma(\frac{1}{4}\overline{v}^2 - \widetilde{x}^2 - \widetilde{y}^2)(\widetilde{x}^2 + \widetilde{y}^2) \quad (3)$$

For limit cycle, we can see that $\dot{V}(\widetilde{x}, \widetilde{y}) < 0$ when $V(\widetilde{x}, \widetilde{y}) > \frac{1}{4}\overline{v}^2$, while $\dot{V}(\widetilde{x}, \widetilde{y}) > 0$ when $V(\widetilde{x}, \widetilde{y}) < \frac{1}{4}\overline{v}^2$. This shows the following region is absorbing.

$$B = \{\rho_1 \leq V(\widetilde{x}, \widetilde{y}) \leq \rho_2, |0 < \rho_1 < \frac{1}{4}\overline{v}^2, \rho_2 > \frac{1}{4}\overline{v}^2\} \quad (4)$$

Since this argument above is valid for any $0 < \rho_1 < \frac{1}{4}\overline{v}^2$, and $\rho_2 > \frac{1}{4}\overline{v}^2$, when ρ_1, ρ_2 get close to $\frac{1}{4}\overline{v}^2$, region B shrinks to the circle $V(\widetilde{x}, \widetilde{y}) = \frac{1}{4}\overline{v}^2$.This shows that the circle is a periodic orbit as shown in Fig. 5(a) when $\overline{v} = 280$, $\rho = 0.01$, $\gamma = 0.0001$. This periodic orbit is called a limit cycle. We can see the trajectory from any point $(\widetilde{x}, \widetilde{y})$ moves toward and converges to the limit cycle clockwise when close. The counterclockwise



(a) clockwise (b) counterclockwise

Figure 5: Phase portrait of limit cycle

condition can be derived by the following system (shown in Fig. 5(b)):

$$\begin{aligned} \dot{\widetilde{x}} &= \rho(-\widetilde{y} + \gamma\widetilde{x}(\frac{1}{4}\overline{v}^2 - \widetilde{x}^2 - \widetilde{y}^2)) \\ \dot{\widetilde{y}} &= \rho(\widetilde{x} + \gamma\widetilde{y}(\frac{1}{4}\overline{v}^2 - \widetilde{x}^2 - \widetilde{y}^2)) \end{aligned} \quad (5)$$

Considering that the trajectory from any point $(\widetilde{x}, \widetilde{y})$ inside the limit cycle moves outward the cycle, and the trajectory from any point $(\widetilde{x}, \widetilde{y})$ outside the limit cycle approaches the cycle with distance determined by the relative speed $\overline{v}$, the limit cycle provides a method for obstacle avoidance among multiple mobile robots.

In RoboCup Four-legged League, there are many obstacles during the game. Robots can be considered as motive obstacles. When the dog approaches a teammate holding ball, it must stay out of the area where teammate handles ball, and be ready to perform cooperative strategies. If the dog holding ball encounters an opponent, it must control the ball and quickly avoid the approaching robot, especially when perform kicking ball in front of opponent goalie. Own penalty area is another one that can be taken for an obstacle. If the robot moves parallel to own ground line, it must avoid from walking into the own penalty area.

When the dog is in a safe region, by the dynamic limit cycle approach, it will move away the obstacle toward the safe circle with a radius relevant to the speed of the obstacle. Let α denote the orientation of the obstacle, (x_0, y_0) the center point of the obstacle. With the following transformation, we get the expression of system (8) in the original frame:

$$\begin{aligned} x &= \cos\alpha(\widetilde{x} + x_0) - \sin\alpha(\widetilde{y} + y_0) \\ y &= \sin\alpha(\widetilde{x} + x_0) + \cos\alpha(\widetilde{y} + y_0) \end{aligned} \tag{6}$$

Let v denote the translational velocity of the dog in the original frame, θ the direction of the motion. The kinematic model of the dog is described by:

$$\begin{aligned} \dot{x} &= v\cos\theta \\ \dot{y} &= v\sin\theta \\ \frac{\dot{x}}{\dot{y}} &= \tan\theta \end{aligned} \tag{7}$$

Then we can see:

$$\begin{aligned} v &= \sqrt{\dot{x}^2 + \dot{y}^2} \\ \theta &= \arctan\frac{\dot{x}}{\dot{y}} + \alpha \end{aligned} \tag{8}$$

Different obstacles have their own characters, with ρ matching to characters respectively. Using ρ in different values can control the magnitude of the absolute speed.

With the dynamic radius of the limit cycle, robot can perform more flexibly and rationally. Satisfactory results are shown in our application video. The implementation of this method is introduced in our team report 2006.

4.2 Perform Near Border

In 2006, new behaviors and strategies correlated to new border line are important. Any unappropriate behavior near border may cause a negative impact. For example, if the ball is near border in own half field, it is dangerous for the defender to handle ball inappositely to let it out of field. Because it may benefit the opponent striker to control the ball. To avoid this situation, we implement the near border behavior.

We define that for a player, if the distance to border line is less than $600mm$, it enters the *near border area*. It is simple that if the player handles ball near border, it can hold ball and move it along the direction vertical to borderline. However, actual test shows that different gaits along with grabbing ball motion may not help control ball well. Therefore, we divide the circle area around player into four parts. Fig. 6 shows the different parts of the *near border area* which may activate strategies respectively. We define the variable *robotPose.anlgle-to-border* which represents the absolute value of angle between robot's body direction and normal line to the border. Area 1 is the place where the *angle-to-border* is in range from 120° to 180°. In area 2 and 3, the angle is between 80° and 120°. Area 4 means the angle is less than 80°. In area 1, the robot grabs

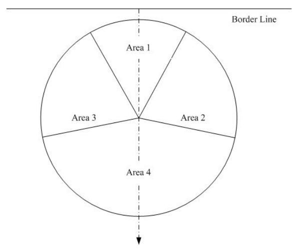


Figure 6: Strategy field in *near border area*

ball and adjusts its body direction first. Then the robot performs a sideways walk moving ball into field. In area 2 and 3, the robot performs a sideways walk directly. Player walks forward directly to the field if enters the area 4.

In GT2004, option *handle-ball* is used for grabbing ball and taking relevant actions. To apply our strategy in *near border area*, we rewrite the option *handle-ball* to control ball more appropriately. Fig. 7 shows the states switching in new option *handle-ball*. If the robot needs to

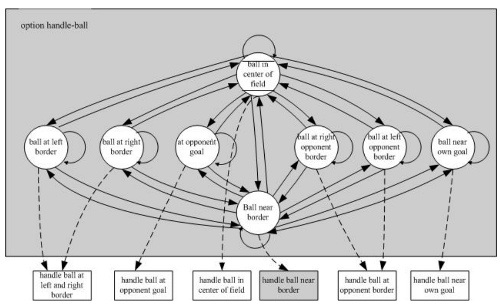


Figure 7: Option *handle-ball*

handle ball in *near border area*, option *handle-ball* will activate the newly created option *handle-ball-near-border*, shown in Fig. 8, which specifically deals with the ball near border. In option *handle-ball-near-border*, robot goes to the ball first and then another new option *back-and-turn-and-push* is activated. This option executes the basic ac-

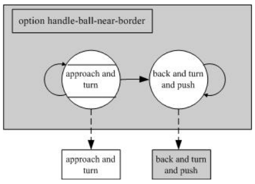


Figure 8: Option *handle-ball-near-border*

tions to handle ball in motion level. Fig. 9 describes the details for option *back-and-turn-and-push*. Applying new behavior and strategies men-

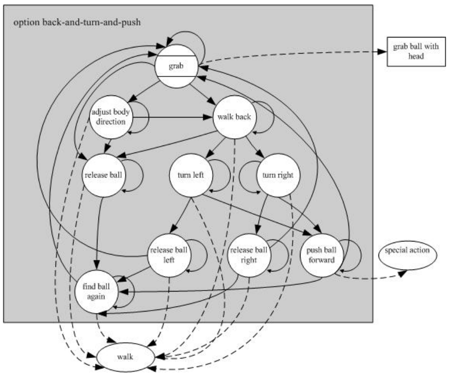


Figure 9: Option *back-and-turn-and-push*

tioned above, robot can play well on the field. And the probability of out-field ball caused by own players is relatively low.

4.3 Dribbling Ball

In a soccer game, the team that controls the ball more will probably score more. There are two ways to control the ball: a player's dribbling and two or more players' passing. Passing ball between robot players is still a difficult task, as shown in the result of the Passing Challenge 2006. Our team is also developing passing ball, but the result is not as good as that in developing dribbling ball. The dribbling ball behavior in sharPKUngfu 2006 is based on the gait called walking with ball. We use PSO based method to optimize the gait so that the robot can walk with ball smoothly and steadily in any direction: forward, backward, sideward and turning. Moreover, the robot can still see as far as possible when walking with ball, because its uses the chin to control the ball and keeps the head up (See Section 4.5 Chin Control). In the dribbling ball behavior, the robot controls the ball and walks towards the opponent goal. It will avoid obstacles and adjust direction when walking with ball. When the time is over (In Rule 2006 robots are allowed to hold the ball for up to 3 seconds), it will kick the ball to opponent goal or pass the ball to its teammate.

4.4 Special Actions

GT2004 presents a series of good special actions [2]. Most of them are useful, but they are not exactly applicable with consideration of new rule. New rule uses only white line instead of the border wall. This change not only affects vision and behaviors module, but also asks for appropriate special actions to perform near border, like handling ball.

To perform well in new field, we design our own set of special actions instead of those used in GT2004 except the *headLeft* and *headRight*. Those actions are tested in real games. All of these actions have their own features and advantages. Specifically for example, in action *chestSoft* robot first holds the ball and then push the ball out. The advantage of this special action is that the ball can be held firmly and the opponent can not easily capture the ball.

And we also changed the condition to activate special actions. For instance, when the ball is near the border line, any inapposite action may push the ball out of the field. When the ball near the border, we have two strategies using relative actions. In most cases, we use the method *Perform Near Border*. In a few occasions, when the behavior near border can not be activated, the special actions using paw will be inspired, like *slapLeft* or *slapRight*. The activated condition is carefully tested.

4.5 Chin Control

Chin Control is another part that we implement in the motion module of sharPKUngfu 2005. It is designed to perform in *Dribbling Ball* behavior. In this motion, the robot opens its mouth to an appropriate angle, and holds ball combining with the neck-tilt, the head-pan and the head-tilt motion. Using *chin control*, camera can be adjust to relatively high position. It is quite useful to avoid losing vision information when robot holds the ball. Fig. 10 compares the actions for grab ball used in different teams.

(a) (b)

Figure 10: Actions for grabbing ball in different teams. (a) is the chin control in sharPKUngfu 2006, while (b) is the grabbing ball used in GT2004

There is a problem in using chin to grab ball. The control area using head to grab ball like (b) in Fig. 9, is relatively larger than that one using chin. If the next action is turning body around or moving ball to left or right, the possibility of losing ball control is increasingly high. To solve this problem, we apply head control motion used in GT2004 to perform in the circumstance mentioned above. Chin control is only used in grabbing ball when perform *Dribbling Ball* behavior. Robot can use chin to help control ball to move forward or backward.

5 Conclusion

In sharPKUngfu 2006, we have made improvement in localization, locomotion and behavior modules. In RoboCup 2006, we perform our technical improvement in open challenge, passing ball and new goal challenge. The result makes us confident to perform in RoboCup Soccer competitions. After RoboCup 2006, we participated in RoboCup China Open 2006. Advantages in sharPKUngfu 2006 help our team make great success in this event. We got champions both in soccer competition and technical challenge. After the event, we focus our research on further study in collaborative localization, navigation and gaits optimization. All the improvement is explained above in detail. We have applied experience-based collaborative approach for localization which is important to make robots more rational and efficient. In gaits optimization, we implemented PSO based approach to get relatively high-speed forward gaits. To perform better under the soccer rule 2006, new behaviors and relevant actions have been created to hold ball in the field to get better performance. Besides, we tried to apply new approach to percept robots and avoid dynamic obstacles. Experiments in our lab show positive effect by using the real-time approach.

In 2007, we plan to let Aibo play in the environment without any landmark towards real human soccer conditions. Further study should be continued to exploit enough surrounding information to help self-localization. In vision module, we plan to implement color-edge based method to recognize beacons and goals which are newly defined in soccer rule 2007. Beside of forward gaits optimization, we will implement PSO in other different walking types to gain optimized motion parameters. In multi-robot coordination, the research on formation control will continue. In addition, we will continue to get involved in challenges of passing ball and obstacle avoidance. The final version of our code 2006 is now available on our web site.

Acknowledgement

This work was supported by National Natural Science Foundation of China (No. 60404001 and No. 60274001), National High Technology Research and Development Program of China (863 Program) (No. 2006AA04Z258), and 985 Project of Peking University.

Team members of sharPKUngfu, including Lianghuan Liu, Hua Li, Xin Guan, contributed works in real robot implementation and competitions.

References

[1] Q. Wang, L. Liu, G. Xie, L. Wang, Learning from human cognition: collaborative localization for vision-based autonomous robots, Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2006, pp. 3301-3306.

[2] T. Röfer, et. al., GermanTeam RoboCup 2004, Available online: http://www.germanteam.org/GT2004.pdf, 2004.

[3] T. Schmitt, R. Hanek, M. Beetz, S. Buck, B. Radig, Cooperative Probabilistic State Estimation for Vision-based Autonomous Mobile Robots. In IEEE Transactions on Robotics and Automation, vol. 18(5), 2002, pp. 670-684.

[4] D. Fox, W. Burgard, H. Kruppa, S. Thrun, A Probabilistic Approach to Collaborative Multi-Robot Localization. In Special Issue of Autonomous Robots on Heterogeneous Multi-Robot Systems, vol. 8(3), 2000.

[5] T. Röfer, M. Jüngel. Vision-Based Fast and Reactive Monte-Carlo Localization, Proc. of the IEEE International Conference on Robotics and Automation, 2003, pp. 856-861.

[6] D. Fox, W. Burgard, F. Dellaert, S. Thrun, Monte Carlo localization: Efficient position estimation for mobile robots, Proc. of the National Conference on Artificial Intelligence, 1999, pp. 343-349.

[7] R. C. Arkin, T. Balch, Cooperative multiagent robotic systems. In D. Kortenkamp, R.P. Bonasso, and R. Murphy, editors, Artificial Intelligence and Mobile Robots. MIT/AAAI Press, Cambridge, MA, 1998.

[8] J. Wolf, W. Burgard, H. Burkhardt, Robust vision-based localization by combining an image-retrieval system with Monte Carlo localization, IEEE Transactions on Robotics, vol. 21(2), 2005, pp. 208-216.

[9] R. Hanek, T. Schmitt, Vision-based localization and data fusion in a system of cooperating mobile robots, Proc. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 2, 2000, pp. 1199-1204.