MCT Susano Logics 2016 Team Description

Shota Aoki, Taihei Degawa, Kazuhiro Fujihara, Yuta Notsu, Toshiyuki Beppu

National Institute of Technology, Matsue College, 14-4, Nishi-Ikuma, Matsue, Shimane, 690-8518, Japan

beppu[at]matsue-ct.jp
http://www.matsue-ct.ac.jp/

Abstract. In the RoboCup Small Size League (SSL) games, teams of 6 robots which are autonomously controlled by a team's AI play soccer games with using an orange colored golf ball. At the SSL games, an image processing system (SSL-Vision) receives images from 4 cameras which are installed 4 meters above the field surface, identifies the robots and the ball positions, and sends these position data to the AI computers. Because the SSL-Vision identifies the robots and the ball from every camera images, multiple position coordinates are computed at the overlapped regions of the images. To determine the ball position from the multiple data, and to reduce the data discontinuity of the moving ball which goes across the boundary between camera areas, a multi-particle filters (MPF) was developed. The MPF consists of 4 particle filters estimates the position of the ball from the median point of the combined distribution. At experimental setups, the MPF smoothly connected two ball locus vectors of 17 millimeters distance. The distances between each sample of the MPF were almost same as both data from two cameras.

Keywords. Particle Filter, SSL-Vision.

1 Introduction

From RoboCup 2014 João Pessoa, Brazil, a "large field" of nominal size 9000 mm * 6000 mm was partially adopted for SSL games. At RoboCup 2015 Hefei, China, all SSL games were played on the large field.

On the large field, the SSL-Vision system identifies a ball and robots on the field from four camera images. The images overlap at border regions consequently (Figure 1). When the ball or the robots are in the overlapped region, the SSL-Vision calculates the coordinates of objects from every captured image [1]. At the game kickoff, the ball was placed on the center of the field, where four cameras capture images and the SSL-Vision calculates the coordinates of ball from all four images. Our team AI vacillated between them and could not control the kicker robot accurately, therefore. Furthermore, the overlapped region across the goal caused fluctuations in our AI's predicted ball course and resulted poor defense of our goalie. A multi-particle filter for the ball position was developed to minimize the discontinuity of the ball position data across the overlapped regions.

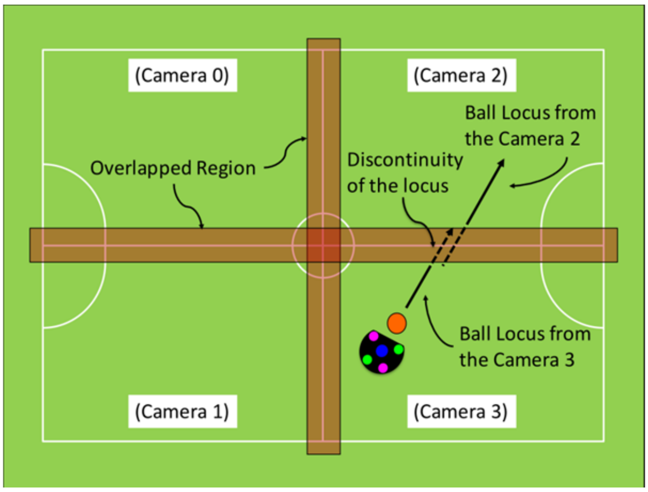

Fig. 1. Overlapped region of camera images on the SSL field

2 Multi-Particle Filter

2.1 Particle Filter

Particle Filter (PF) is a set of genetic-type particle Monte Carlo methodologies to solve the filtering problem [2]. Each particle in the PF has the state vector. The PF estimates the posterior density of the state vectors from given observation vectors.

Figure 2 shows the process of the PF [3]. The PF consists of three calculation steps of estimation, update and resampling. At the start of the loop of the process, the PF generates a set of particles based on the uniform distribution or the normal distribution. From the second loop, the PF uses the anterior filtered distribution of the system. The PF estimates a state distribution at time t from that of t - 1 with using the system model in the estimation step.

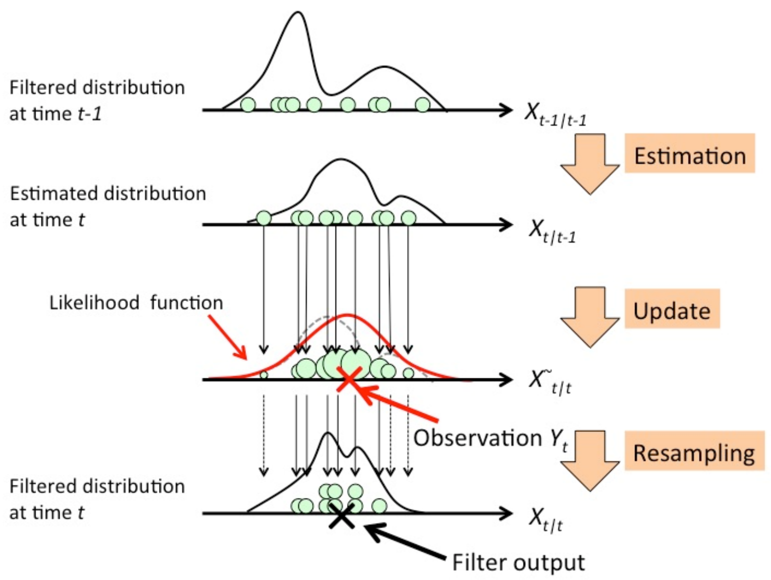

Fig. 2. Process of the particle filter

In the update step, the PF receives an observation vector $\boldsymbol{y}_t$ and evaluates the likelihood of each particle by utilizing a likelihood function. The likelihood function in Fig. 2 is based on Equation (1).

$$\omega_t = exp\left(-\frac{\left(\boldsymbol{y}_t - \boldsymbol{x}_{t|t-1}\right)^T R^{-1}\left(\boldsymbol{y}_t - \boldsymbol{x}_{t|t-1}\right)}{2}\right) \tag{1}$$

This function shows a normal distribution with the mean vector of $\boldsymbol{y}_t$. The function gives a strong likelihood to the particle whose state vector is close to $\boldsymbol{y}_t$. The value of the likelihood indicates the survival probability of the particle in the resampling step.

In the resampling step, the PF deletes particles with weak likelihood probabilities. The number of deletion is selected at random. Then the PF generates the same number of deleted particles with the likelihood function as a probability density function. After the resampling step, the particle distribution shows the filtered distribution at time *t*. The PF outputs the median point of the filtered distribution as the estimated state of the system.

2.2 System Model for the SSL-Vision

The two-dimensional state vector $\boldsymbol{x}_t$ identifies the ball position.

$$\boldsymbol{x}_t = [p_{xt} \quad p_{yt}]^T \tag{2}$$

An observation vector $\boldsymbol{y}_t$ is the coordinates received from the SSL-Vision. Observation noise $\boldsymbol{w}_t$ is the normal distribution with a standard deviation of 2.2 mm.

$$\boldsymbol{y}_t = \boldsymbol{x}_t + \boldsymbol{w}_t \tag{3}$$

A velocity vector $\boldsymbol{s}_t$ is calculated from $\boldsymbol{y}_t$, $\boldsymbol{y}_{t-1}$, and the sampling period of the SSL-Vision Δt.

$$\boldsymbol{s}_t = (\boldsymbol{y}_t - \boldsymbol{y}_{t-1}) \,/\, \Delta t \tag{4}$$

A system model is defined as follows.

$$\boldsymbol{x}_t = \boldsymbol{x}_{t-1} + \boldsymbol{s}_{t-1} * \Delta t + \boldsymbol{v}_t \tag{5}$$

An inaccuracy of the ball motion is expressed as the system noise $\boldsymbol{v}_t$. This system noise conforms to the normal distribution with the standard deviation of 1.7 mm.

2.3 Multi-Particle Filter

Figure 3 shows the process of our multi-particle filter (MPF) program. The MPF has four PFs and each PF is assigned for a camera. The SSL-Vision sends the ball position data with camera IDs. The MPF identifies the data from camera ID number and calculates a particle distribution for a camera independently. The filtered distributions are combined and a median point of the combined distribution shows the estimated ball position.

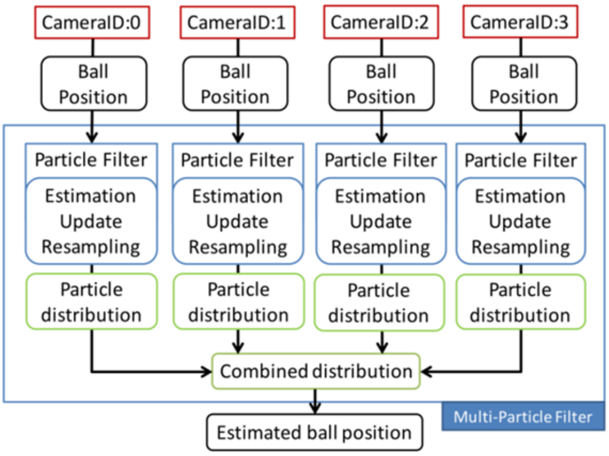

Fig. 3. Process of the Multi-Particle Filter

When the ball is in a camera region, a PF of data-received executes the filter process (Fig. 4). The PFs of continuous-missing do not execute the filtering steps. The MPF copies the particle distribution output from the executed PF to the combined distribution and other PFs as the anterior distribution for the next filtering. This technique reduces fluctuations of the estimated ball position for the time of the ball entry to the overlapped region.

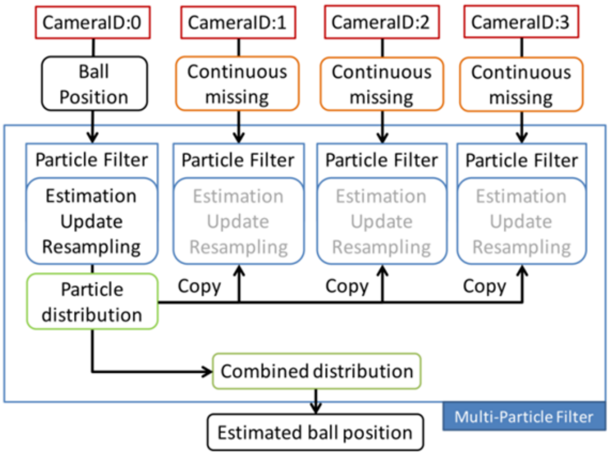

Fig. 4. MPF movement with a ball data

When the ball in the overlapped region, the SSL-Vision sends multiple ball positon. At this time, the PFs of data-received execute the filter and all the outputs are collected into the combined distribution.

If a data from a camera missed occasionally, the PF of occasional-missing outputs the particle distribution with the estimation step only (Fig. 5).

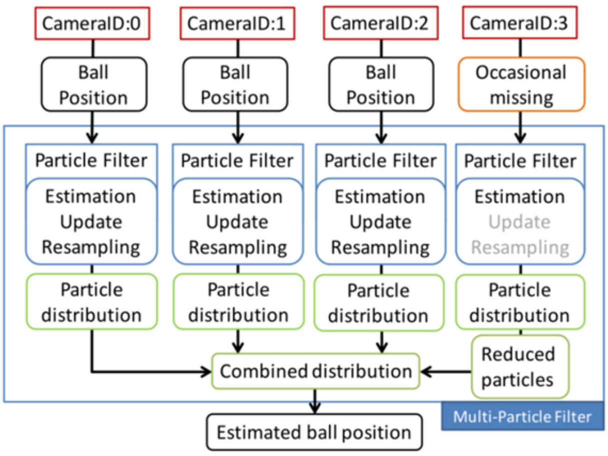

Fig. 5. MPF movement with an occasional data missing

In case of the continuous data missing, the probability of estimation deteriorates gradually. At that time, the number of the particles from the PF of occasional-missing to the combined distribution (*pNum*) is reduced in proportion to the number of missing cycles (*missingCount*).

$$pNum = pMAX * \left(1 - \frac{missingCount}{missingMAX}\right) \tag{6}$$

Where *pMAX* is the number of particles used for a PF, *missingMAX* is the maximum number of cycles that the PF executes the estimation without observations. By the reduction method of Equation (6), the position fluctuations of the filtered output at the time of the ball exits from the overlapped region to a single camera region is diminished.

Estimated ball position *outputX* is the center of the particles $\boldsymbol{x}_c$.

$$outputX = \frac{\sum_{c=0}^{cameraNum} \sum_{i=0}^{pNum_c} \boldsymbol{x}_c^{[i]}}{\sum_{c=0}^{cameraNum} pNum_c} \tag{7}$$

2.4 Experimental Result of the MPF

To evaluate the performance of the MPF, we examined the MPF outputs with two camera images. Two cameras (AVT Stingray F046C, 780x580) at a distance of 3 meters installed 3.4 meters above the surface of the field captured the ball at a speed of 1.9 meters per second. The number of particles *pMAX* was set to 300, and maximum number of cycles *missingMAX* was set to 15.

Figure 6 shows the result of an experiment. Two ball locus vectors of 17 millimeters distance were connected smoothly by the MPF. The distances between each sample were 31.4 ± 14.7 (Camera 0), 28.5 ± 8.6 (Camera 1), 29.5 ± 11.0 (MPF) millimeters in this trial.

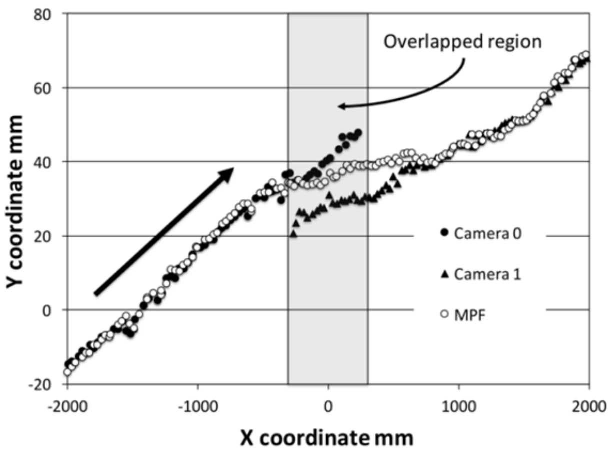

Fig. 6. Estimated ball positions with the MPF

3 Implementation of MPF to the Team Software

Figure 7 shows the software structure of MCT Susano Logics. Packet Reader receives vision packets from the SSL-Vision and sends the ball position with camera ID to the MPF. The estimated position is sent to both AI and Robots Controller. The software was developed by C++ (GCC 4.9). The process time of the MPF with 250 particles is 2.41 $\pm$ 0.37 milliseconds on Dell Inspiron 15 5000 (CPU: Intel Core i7-5500U 2.4GHz, memory size: 8GB) with Linux OS (Xubuntu 14.04). This time consumption is small enough for our software.

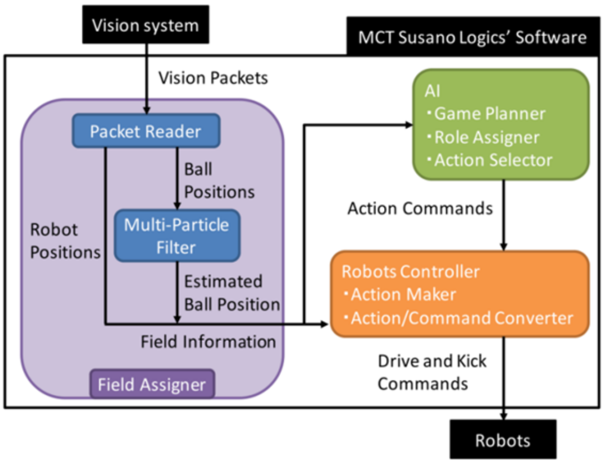

Fig. 7. Software structure with MPF

References

1. S. Zickler, T. Laue, O. Birbach, M. Wongphati, M. Veloso. SSL-Vision: The Shared Vision System for the RoboCup Small Size League. 2009. http://isites.harvard.edu/fs/docs/icb.topic859418.files/papers/sslvision2009-new.pdf
2. P. Del Moral. Non Linear Filtering: Interacting Particle Solution. Markov Processes and Related Fields 2 (4): 555–580, 1996.
3. T. Ikoma. Sequential Monte Carlo method and Particle Filter (in Japanese). Chapter 11, Statistical Science in 21th. Century, Japan Statistical Society. 2008. http://ebsa.ism.ac.jp/ebooks/sites/default/files/ebook/1881/pdf/vol3_ch11.pdf