Journal of Robotics, Networking and Artificial Life, Vol. 3, No. 2 (September 2016), 132-135

Improvement of Computational Efficiency of Unscented Particle Filter by Automatically Adjusting the Number of Particles

Kenta Hidaka, Takuo Suzuki, and Kunikazu Kobayashi
School of Information Science and Technology, Aichi Prefectural University
1522-3 Ibaragabasama, Nagakute, Aichi 480-1198, Japan.
Tel: +81-561-76-8782, Fax: +81-561-64-1108
kobayashi@ist.aichi-pu.ac.jp

Abstract

In RoboCup Standard Platform League (SPL), the method using unscented particle filter (UPF) has been proposed for self-localization. The UPF resolves a problem of particle filter which cannot be sampled appropriately when the likelihood is too high or low. This filter can estimate accurate position when the more number of particles is. However, the more, the more computation time is needed. In the present paper, we propose an automatic adjustment method for the number of particles in UPF. The proposed method uses three kinds of feature values with respect to particles, i.e. centroid, standard deviation, and weight. Through computer simulations, we confirmed the improvement of computational efficiency of UPF.

Keywords: RoboCup, Self-localization, Unscented particle filter, Kalman filter, Particle

1. Introduction

The RoboCup (Robot Soccer World Cup) project sets a goal that a fully autonomous robot team shall win against the most recent winning team of FIFA World Cup in soccer by 2050. In the present paper, we treat one of major research issues in RoboCup, i.e. self-localization. In RoboCup, self-localization is required in a variety of situations, such as passing a ball to a teammate and intercepting a pass.

The RoboCup Standard Platform League (SPL) is a league that all teams compete with the same standard humanoid robot called NAO developed by Aldebaran Robotics. The robot operates fully autonomously, that is with no external control, neither by humans nor by computers. In RoboCup SPL, the robot must process all the calculations on vision processing and decision making using low-end CPU (Intel Atom 1.6GHz). In addition, the robot must devote a lot of computation time to percept a white goal and a mostly white ball in vision processing.

In the present paper, we focus on self-localization for reducing the computational cost. In conventional method using UPF[1], since the number of particles is fixed during self-localization, it is possible to reduce the computational cost by changing the number of particles and rejecting redundant particles depending on the situation. In RoboCup SPL, there is no method which can change the number of particles in UPF.

Yu et al. proposed an adaptive unscented particle filter, in which relative entropy is used for changing the number of particles[2]. Yang et al. proposed another adaptive unscented particle filter, in which Kullbuck-Leibler divergence (KLD) sampling[3] is employed[4]. In contrast, the conventional self-localization method using UPF in RoboCup SPL uses 16 fixed particles[5]. In RoboCup Middle Size League (MSL), the number of particles in the conventional self-localization method using KLD sampling is fixed as 75[5]. In the present paper, in order to improve the computational efficiency of UPF, we propose an automatic adjustment method for changing the number of particles.

In Section 2, we explain fundamentals on UPF. Section 3 describes the proposed algorithm which can adaptively change the number of particles in UPF. In Section 4, we conduct some computer simulations and discuss the results. Section 5 summarizes our research and gives challenges for the future.

2. Unscented Particle Filter (UPF)

UPF has a combination of unscented Kalman filter (UKF)[6] and particle filter[7]. This filter can solve the problem in particle filter which resampling will fail if the new measurements appear in the tail of prior or if the likelihood is too peaked in comparison to the prior[1]. In this section, we outline particle filter and UKF briefly.

2.1. *Particle Filter*

The particle filter is a type of non-parametric Bayesian filter. This filter can apply to nonlinear state equations with non-Gaussian probability distribution.

The filter estimates the number of particles (state) by repeating the following three steps A, B, and C. This filter can precisely estimate state as more number of particles. Let the number of particles be N.

A. Motion update

The odometry information u_t obtained from robot estimates the current position $x_t=\{x_t^{[1]},\ldots,x_t^{[i]},\ldots,x_t^{[N]}\}$ from the previous position x_{t-1}.

$$p(x_t|u_t, x_{t-1}).$$

B. Measurement update

The weight w_t is calculated from x_t using measurement z_t.

$$w_t = \eta p(z_t|x_t),$$

where η is a constant.

C. Resampling

Some particles will be replaced with newly sampled ones.

draw i with probability $\propto w_t^{[i]}$,

add $x_t^{[i]}$ to state set X_t.

2.2. *Unscented Kalman Filter (UKF)*

The UKF is a filter that handles nonlinear state equation as the following equation. Probability distributions can be used when the distribution follows from normality. This filter continues to update one state. UPF updates the the motion update in particle filter using UKF.

$$x(t+1) = f(x(t)) + bv(t),$$
$$y(t) = h(x(t)) + z(t),$$

where x(t) is an n-dimensional vector, b is a constant. Assume that both variances of system noise v(t) and observation noise z(t) follow from normal distributions. UKF approximates the non-linear function f and g using unscented transform[1].

3. Proposed System

We propose an automatic adjustment method of the number of particles in UKF to reduce the computational cost. The proposed method realizes adjusting the number of particles by two steps, i.e. reset step and increase or decrease step as shown in Fig. 1.

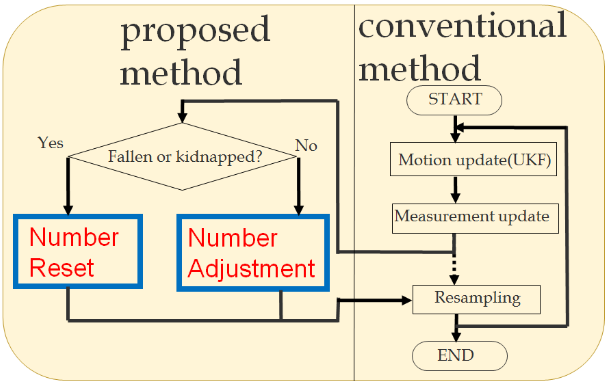


Fig. 1: Outline of the proposed method.

3.1. *Number Reset*

The number reset step will be applied to increase the number of particles when robot is fallen or kidnapped. In RoboCup SPL, since there are so many such cases, a fallen or kidnapped robot must recover from disordered states. We therefore have to increase the number of particles to improve self-localization.

3.2. *Number Adjustment*

In the number adjustment step, the number of particles can be changed according to the estimation accuracy of self-localization. The detailed algorithm is illustrated in Fig. 2. Let $X=\{x_1,x_2,\ldots,x_N\}$, $Y=\{y_1,y_2,\ldots,y_n\}$ and X_{new}

Algorithm 1 Number Adjustment

for $i = 1$ to N **do**
  if $\alpha \leq weight(x_i)$ **then**
    $X_{new} = X_{new} + x_i$
  end if
end for
if $|X_{new}| \neq 0$ **then**
  if $N = |X_{new}|$ **then**
    $Y = X_{new}$
    $e_1 = parmeter1$
    $e_2 = parmeter2$
  else
    $Y = X - X_{new}$
    $e_1 = parmeter3$
    $e_2 = parmeter4$
  end if
  $\mu = E[X_{new}]$
  $\sigma^2 = V[X_{new}]$
  for $c = 1$ to $|Y|$ **do**
    if $\mu - e_1\sigma < y_c < \mu + e_1\sigma$ **then**
      if $c = |Y|$ *and* $N \neq N_{min}$ **then**
        $X = X - min(weight(Y))$
        N = N-1
      end if
    else if $\mu - e_2\sigma < y_c < \mu + e_2\sigma$ **then**
      if $c = |Y|$ **then**
        break
      end if
    else
      if $N = N_{max}$ **then**
        break
      else
        $N = N + 1$
        break
      end if
    end if
  end for
end if

Fig. 2: Number adjustment step.

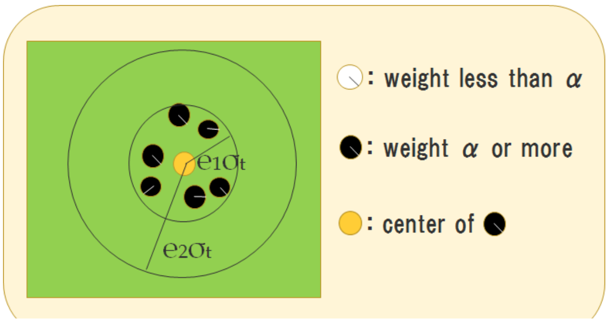


Fig. 3: One example of deleting a particle.

be sets. σ_n is the standard deviation of the particles with a weight more than α. e_1 and e_2 are any positive real

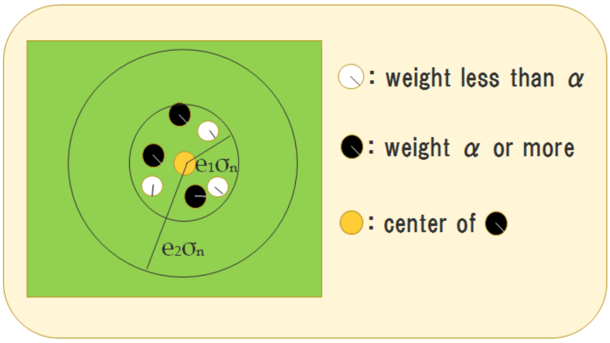


Fig. 4: Another example of deleting a particle.

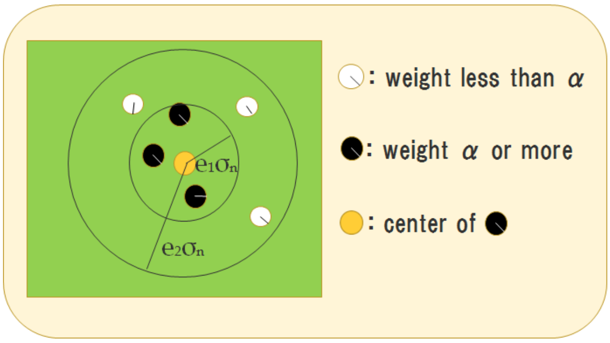


Fig. 5: An example of adding a particle.

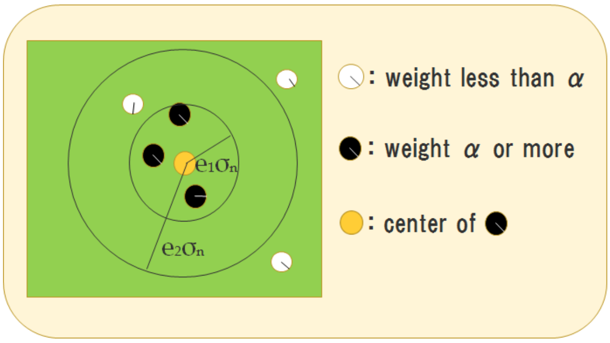


Fig. 6: An example of fixing the number of particles.

values and used in order to adjust the size of σ_n at time n. The proposed algorithm will reduce the number of particles if the situation as shown in Figs. 3 and 4. In Fig. 3, the white and black circles correspond to the particles with weights less than α and more than or equal to α, respectively and the yellow circle refers to the centroid of the black ones. If all the white circles are within the circle with radius $e_1\sigma_n$ and centered at the centroid of the black ones, i.e. the yellow circle, we can reduce the number of particles. On the other hand, if any of the white circles is beyond the circle with radius $e_2\sigma_n$, we cannot reduce any particles as shown in Fig. 6. If all the white circles are within the circle with radius $e_2\sigma_n$ and centered at the centroid of the black ones, we fix the number of particles.

4. Simulation

In this section, we show the validity of the proposed method through some computer simulations using SimRobot simulator[5]. Assume that the maximum number of particles is 16, i.e. N_{max}=16 as the same with the conventional method[5].The values of parameters are set as follows: N_{min}=4, parameter1=1, parameter2=2, parameter3=2, parameter4=3, and α=0.7.

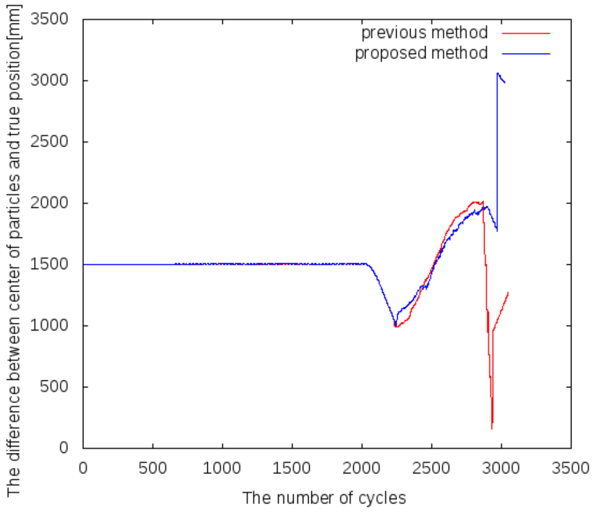
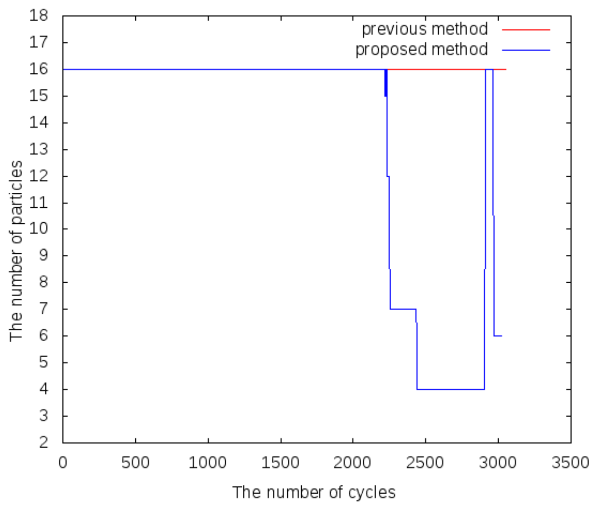


Fig. 7: Transition of the difference between the centroid of particles and true position.

Figure 7 illustrates the difference between the centroid



of particles and true position as the number of cycles. Figure 8 shows the transition of the number of particles. The estimation accuracy of the self-localization using the proposed method is able to maintain the same level as the conventional method. However, the computational results confirm that the self-localization error and the number of particles is not proportional. The number of cycles for processing the self-localization was measured for 10 seconds. That for the conventional method and the proposed method are 48,340 and 43,880 times, respectively. As a result, the computational efficiency was improved by about 10 percent.

5. Conclusion

In this paper, we focused on the self-localization in RoboCup SPL and proposed the automatic adjustment method of the number of particles in UPF. We suggested that the number of particles could be variable by using three values, i.e. the centroid, the standard deviation, and the weight of particles. Through computer simulations, we confirmed that the proposed method improved the computational efficiency as compared with the conventional method. As future issues, the number of particles and the self-localization error in the proposed method are required to be varied in proportion, because there is no need to increase the number of particles when the error is small.

References

1. R. van der Merwe et. al., The Unscented Particle Filter, *Advances in Neural Information Processing Systems*, pp.584-590 (2000).
2. W. Yu et. al., An Adaptive Unscented Particle Filter Algorithm through Relative Entropy for Mobile Robot Self-Localization, *Mathematical Problems in Engineering*, Vol.2013, Article ID 567373 (2013).
3. D. Fox, KLD-sampling: Adaptive Particle Filters, *Advances in Neural Information Processing Systems*. pp.713–720 (2001).
4. J. Yang et. al., A Fast Initial Alignment Method for SINS Used Adaptive Sample Size Unscented Particle Filter, *Proc. of International Conference on Chemical, Material and Food Engineering*, doi: 10.2991 (2015).
5. T. Röfer et. al., *B-Human Team Report and Code Release 2014* (2014).
6. S. J. Julier et. al., A New Extension of the Kalman Filter to Nonlinear Systems, *Proc. of AeroSense: The 11th Int. Symp. On Aerospace/Defence Sensing, Simulation and Controls*, pp.182-193 (1997).
7. N. J. Gordon et. al., Novel Approach to Nonlinear/non-Gaussian Bayesian State Estimation, *IEE Proceedings F*, Vol.140, No.2, pp.107-113 (1993).