International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:5, No:11, 2011

Hybrid Feature and Adaptive Particle Filter for Robust Object Tracking

Xinyue Zhao, Yutaka Satoh, Hidenori Takauji, and Shun'ichi Kaneko

Abstract—A hybrid feature based adaptive particle filter algorithm is presented for object tracking in real scenarios with static camera. The hybrid feature is combined by two effective features: the Grayscale Arranging Pairs (GAP) feature and the color histogram feature. The GAP feature has high discriminative ability even under conditions of severe illumination variation and dynamic background elements, while the color histogram feature has high reliability to identify the detected objects. The combination of two features covers the shortage of single feature. Furthermore, we adopt an updating target model so that some external problems such as visual angles can be overcame well. An automatic initialization algorithm is introduced which provides precise initial positions of objects. The experimental results show the good performance of the proposed method.

Keywords—Hybrid feature, adaptive Particle Filter, robust Object Tracking, Grayscale Arranging Pairs (GAP) feature.

I. Introduction

Object tracking is of great pertinence to the emerging applications such as visual surveillance, and intelligent traffic navigation etc. Tracking real-world objects is a challenging task due to the presence of noise, occlusion, clutter, dynamic background elements and confusing background colors. Particle filter based tracking has attracted considerable attention in recent years because of its powerful ability to deal with general non-linear and non-Gaussian problems. In the framework of particle filter, one of the most important parts is the observation model. The commonly used observation models built for particle filtering tracking are edge-based [1], color-based [3][2], and contour-based features [4]. However, algorithms relying on only one feature are less robust and suffer from various limitations in complex scenarios. The color feature is robust to noise and partial occlusion but suffers from illumination changes and the presence of confusing background colors. The edge and contour features are more robust to illumination variation compared to the color feature but are much sensitive to background clutter .

To overcome these problems, we introduce an adaptive hybrid observation model which integrates the Grayscale Arranging Pairs (GAP) feature [5][6] and color histogram feature in a static camera environment. The GAP feature was originally proposed for background subtraction. In this study, we improve the GAP feature to make it suitable for object tracking. Because of the outstanding performance of the GAP feature in extracting the foreground from the background, it has high sensitivity in distinguishing objects from the background, even in a complex environment (such as conditions with severe illumination changes and dynamic backgrounds). It makes use of both temporal information and global spatial information by considering stable relationships of intensity among multiple point pairs. Moreover, it represents the relative properties between objects and environment, which varies according to the positions of objects. Thus, the GAP feature provides better discrimination in many situations where a simple feature (such as the color feature) may fail, for example, under similar background conditions. Together with the GAP feature, we also utilize the color histogram feature [2], which has been widely utilized and is good at realizing performance to identify objects.

The hybrid model produces a good representation of the discrimination capabilities between objects and the background, and discernment capabilities on an object itself. Furthermore, we adopt an updating target model so that some external problems such as visual angles can be overcame well. An automatic initialization algorithm is also introduced which provides precise initial positions of objects.

The remainder of the paper is structured as follows: Section 2 discusses the function of particle filter. Section 3 introduces a hybrid features based adaptive particle filter algorithm for tracking. Section 4 describes experiments used to validate the algorithm. Section 5 discusses the selection of parameters. Section 6 contains concluding remarks.

Xinyue ZHAO and Shun'ichi KANEKO are with the Division of Systems Science and Informatics, Hokkaido University, Sapporo, Japan. e-mail: zhao@ssc.ssi.ist.hokudai.ac.jp.

Yutaka SATOH is with National Institute of Advanced Industrial Science and Technology (AIST), Japan.

Hidenori TAKAUJI is with Muroran Institute of Technology, Japan.

II. Particle filter for object tracking

A. Basic concept

Let $\mathbf{x}_{t-1}$ denote the state of a tracked object at time $t-1$, $\mathbf{z}_{t-1}$ be an observation at $t-1$, and $\mathbf{z}_{1:t-1}$ denote a set of all observations upto $t-1$. From a Bayesian viewpoint, all interesting information about the target's state $\mathbf{x}_{t-1}$ is encompassed by its posterior $p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1})$. During tracking, this posterior is recursively estimated as the new observation $\mathbf{z}_t$ arrives, which is realized in two major stages: prediction (1) and update (2):

$$p(\mathbf{x}_t|\mathbf{z}_{1:t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1})d\mathbf{x}_{t-1}. \quad (1)$$

$$p(\mathbf{x}_t|\mathbf{z}_{1:t}) \propto p(\mathbf{z}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{z}_{1:t-1}). \quad (2)$$

Recursions (1) and (2) for the posterior require a specification of a dynamic motion model that describes the state evolution $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ and a model that evaluates the likelihood of any state given the observation $p(\mathbf{z}_t|\mathbf{x}_t)$.

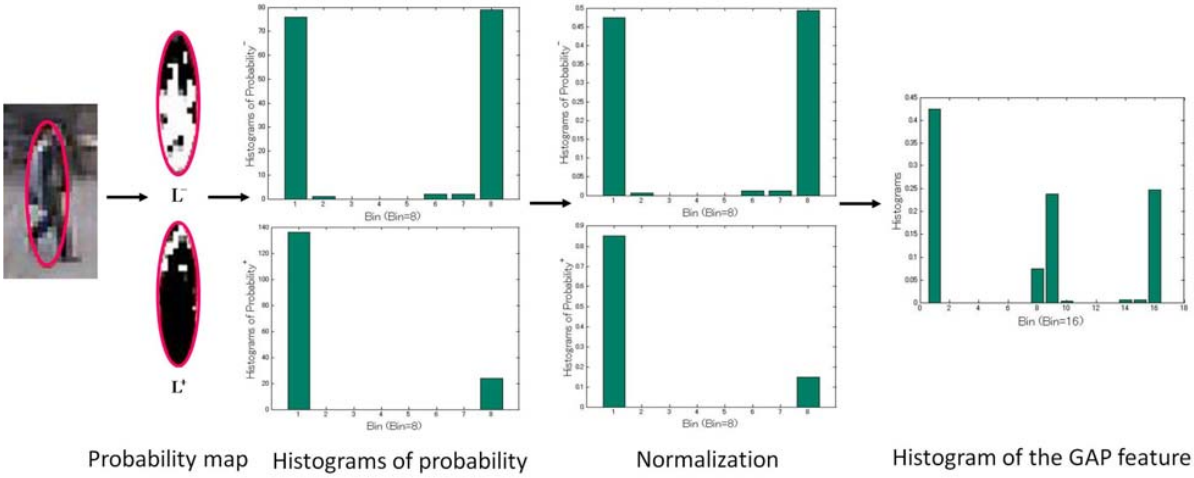


Fig. 1. The procedure of calculating the GAP feature. L^+ shows the relative brighter parts in objects compared to the background and L^- shows the relative darker parts. L^+ and L^- act together to distinguish objects from the background.

B. Dynamic motion model

The state of our tracker is defined as

$$\mathbf{x}_t = \{u_t, v_t, U_t, V_t, \dot{u_t}, \dot{v_t}, \dot{c_t}\}, \tag{3}$$

where (u_t, v_t) are the locations, (U_t, V_t) are lengths of half axes of particles, $(\dot{u_t}, \dot{v_t})$ are velocities, and $\dot{c_t}$ is the scale change factor. Hence the dynamic motion model is denoted as

$$\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + \mathbf{W}_{t-1}, \tag{4}$$

where $\mathbf{A}$ defines the deterministic component of the model and $\mathbf{W}_{t-1}$ is a multivariate Gaussian random variable.

C. Likelihood function

The posterior $p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1}) \approx \{\mathbf{x}_{t-1}^{(m)}, w_{t-1}^{(m)}\}_{m=1,\cdots,M}$ at time $t-1$ is estimated by a cloud of M weighted particles with the state $\mathbf{x}_{t-1}^{(m)}$ and the respective weight $w_{t-1}^{(m)}$. At time t, the particles are first re-sampled according to their weights. Then, they are propagated according to the dynamic model to obtain a representation of the prediction $p(\mathbf{x}_t|\mathbf{z}_{1:t-1})$. Finally, a weight is assigned to each particle according to the likelihood function $w_t^{(m)} \propto p(\mathbf{z}_t|\mathbf{x}_t^{(m)})$. All weights are normalized to sum to one, and the posterior at time t is approximated by a new weighted particle set $p(\mathbf{x}_t|\mathbf{z}_{1:t}) \approx \{\mathbf{x}_t^{(m)}, w_t^{(m)}\}_{m=1,\cdots,M}$. The procedure of determining likelihood is based on feature similarity. The features of the target region are compared with those of other candidate particle regions extracted in the last frame. In this paper, we will introduce a hybrid feature for determining likelihood. The details of the likelihood function will be introduced in Section 3.

D. State estimation

From the set of weighted samples, the current state $\hat{\mathbf{x}}_t$ can be estimated as

$$\hat{\mathbf{x}}_t = \sum_{m=1}^{M} w_t^{(m)} \mathbf{x}_t^{(m)}. \tag{5}$$

III. Proposed tracking algorithm

A. GAP feature

The GAP feature denotes the probability of one pixel belonging to the foreground. In our framework, unlike many traditional methods which consider the history states of a pixel to decide whether it belongs to the foreground, we do this by considering the relationship between this pixel and several statistically chosen reference pixels. For a target pixel P, suppose we have N positive reference pixels which statistically have higher intensity than P, and N negative reference pixels which statistically have lower intensity than P (refer to [6] for the details of statistically choosing the reference pixels). In a new frame, P is classified as the background when its intensity is normal: lower than those of the positive reference pixels and higher than those of the negative reference pixels; contrarily, P is classified as the foreground when its intensity is abnormal: higher than those of the positive reference pixels or lower than those of the negative reference pixels. In details, two probabilities that concern whether P belongs to the foreground are calculated as follows: positive probability $\xi^+ = n^+/N$ (n^+ denotes the number of positive reference pixels whose intensities are lower than that of P) and negative probability $\xi^- = n^-/N$ (n^- denotes the number of negative reference pixels whose intensities are higher than that of P). These two probabilities compose the GAP feature, and $0 \leq \xi^\pm \leq 1$.

In this paper, we define $L^\pm$ for the convenience of explanation as follows

$$L^\pm = \begin{cases} \pm\lceil(1-\xi^\pm)\cdot U\rceil & (\xi^\pm < 1), \\ \pm 1 & (\xi^\pm = 1), \end{cases} \tag{6}$$

where we use the top integral function for easy of calculation. Histograms provide a simple and efficient summary of data distribution and are widely used in describing the characteristics of different features. Here we adopt a histogram with the u-bin ($u \in \{-U, \cdots, -1, 1, \cdots, U\}$, and $2U$ is the total

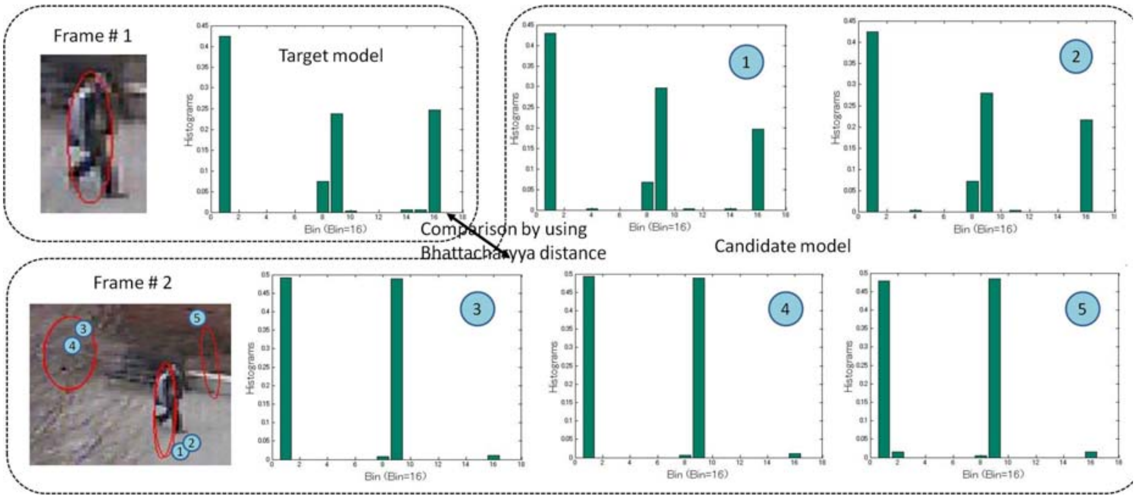


Fig. 2. The likelihood of the GAP feature between the target model and the candidate model is measured by the Bhattacharyya distance. Candidate models which are in the same location as the target object have the similar shapes with the target model. It shows that the GAP feature is robust at distinguish foreground objects from background.

number of bins) to represent the GAP feature. In Eq. 6, U is the half number of bins in the histogram.

L^+ shows the relative brighter parts in objects compared to the background and L^- shows the relative darker parts. L^+ and L^- act together to distinguish objects from the background. The histograms of $L^\pm$ in the region of a particle are denoted as $h(L^\pm)$. In our experiments, the histograms are typically calculated in the GAP space using 16 bins (from -8 to 8 without 0).

In the next step, histograms of $L^\pm$ are attached together and normalized. The histogram of the GAP feature in the state $\mathbf{x}_t$ is calculated as

$$F_{GAP}^{(u)} = \gamma \cdot \Bigg(\sum_{L^-=-U}^{-1} \frac{h(L^-)}{A} \delta(L^- - u) + \sum_{L^+=1}^{U} \frac{h(L^+)}{A} \delta(L^+ - u) \Bigg), \tag{7}$$

where A is the number of pixels in each region for normalization, δ is the Kronecker delta function, and γ is used to ensure that $\sum_{u=-U}^{U} F_{GAP}^{(u)} = 1$. The procedure of calculating the GAP feature is shown in Fig. 1.

B. Color feature

Color provides many cues and it achieves robustness to non-rigidity, rotation and partial occlusion of objects. We utilize the HSV color space to make the algorithm less sensitive to illumination changes, which is more robust than the RGB representation. The bin index $b(y_i)$ assigns the color at the location y_i to the corresponding bin. To increase the reliability when boundary pixels of an object are occluded, we use a weighting function $k(r)$ in which pixels that are far from the region center are assigned smaller weights. Then, the color feature $F_{color} = \{F_{color}^{(v)}\}_{v=1\cdots V}$ at the center point y of state $\mathbf{x}_t$ is calculated as

$$F_{color}^{(v)} = f \sum_{i=1}^{A} k\Bigg(\frac{|y - y_i|}{A}\Bigg) \delta\Big(b(y_i) - v\Big), \tag{8}$$

where A is the number of pixels in the region, and f is a normalization constant ensuring $\sum_{v=1}^{V} F_{color}^{(v)} = 1$.

C. Target model update

The tracking performance is influenced by lots of external factors, such as the visual angle, large geometric deformation of object, and environmental fluctuations. To deal with these difficulties, we adopt an updating target model in our method. The update of the target model is implemented by the equation

$$F_t^{\star(u)} = (1 - \alpha) \cdot F_{t-1}^{\star(u)} + \alpha \cdot F_{E_t}^{\star(u)}, \tag{9}$$

where $F_t^\star$ is the target model at time t, $F_{E_t}^\star$ is the histogram of estimated state for each bin u, and α is the update factor.

The target models of GAP feature and color feature are both calculated according to Eq. 4. Then, in the case with M particles, at time t, The likelihood $\rho_{GAP}(\mathbf{x}_t^{(m)})$ between adaptive target GAP feature model and candidate GAP feature model can be measured by the Bhattacharyya distance which is shown in Fig. 2. Candidate models which are in the same location as object have the similar shapes with the target model. It shows that the GAP feature is robust at distinguish foreground objects and background. As the same, the similarity of color histograms between the template and the current frame is also computed using the Bhattacharyya distance. The likelihood of the color feature can be denoted as $\rho_{color}(\mathbf{x}_t^{(m)})$.

D. Feature integration

In our approach, we use the properties of both color histogram feature and GAP feature mentioned above. The overall likelihood is shown as

$$\rho(\mathbf{x}_t^{(m)}) = (1 - \beta) \cdot \rho_{color}(\mathbf{x}_t^{(m)}) + \beta \cdot \rho_{GAP}(\mathbf{x}_t^{(m)}), \tag{10}$$

International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:5, No:11, 2011

Sequence S1 Sequence S2 Sequence S3

Fig. 3. Test sequences.

where β is a parameter to adjust the proportion between two features.

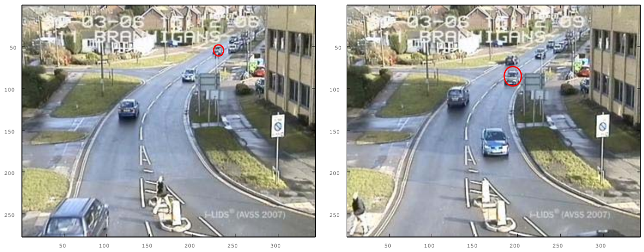
E. Automatic Initialization

For the initialization of the particle filter, the prior knowledge of the target is necessary. Manual initialization was effective and accurate previously, but can not satisfy the real-time surveillance system any more. Here, we propose an automatic initialization algorithm which can provide accurate information for particle filter.

As introduced in the above sections, the GAP feature is different from other common features, and it uses the spatial information which make itself strongly good at discriminating background and foreground. Because of this special property of the GAP feature, we can lock the moving object in the initial frame. The two probabilities ξ^+ and ξ^- of the GAP feature are used together to classify the character of each pixel. A preselected threshold T_w is used. In the case of $\xi^+ < T_w$ and $\xi^- < T_w$, the pixel is considered background, otherwise, the pixel is considered foreground. Then after the simple morphological image processing such as dilation and erosion, we can remove the noises and get connected components. Each component represents for a single moving object and is labeled differently. Then for each of them, its object properties will be extracted together with its location information. Particle filter will be initialized for each of single objects within the component. If the number of components is larger than the number of particle filter, a new object identifier will be assigned to current object. Likewise, the opposite condition is used to determine if an object is lost during the tracking.

IV. Experimental results

In order to evaluate our proposed method, we have done the experiments in different environments. They are taken from both indoor and outdoor scenes and vary with respect to viewpoint, illumination condition, and type of occlusion, demonstrating the robustness of our approach. The proposed method has been implemented in MATLAB and tested on a 3.2 GHz PC with 6 GB memory. The number of particle samples processed in the experiments is 100.



object tracking with target model update

object tracking without target model update

Fig. 4. The comparison of object tracking with and without target model update.

Fig. 4 shows the comparison of object tracking with and without target model update. As shown in Fig. 4, using adaptive model the scale of object appearance is getting adapted correctly compare to without using adaptive model.

The tracking result is presented in Fig. 5. Initially, we use the automatic method to get the initial position of the object. The object moves from the center of lobby under strong partial illumination. Large changes in body size and shape are also challenges in this database. We successfully tracked the object from the initial position until the end of the scene. Fig. 6 shows the comparison between true and estimated position of the tracked object.

We also compare the proposed method with color based tracking method using three different sequences shown in Fig. 3. The tracking results of the position error curve are shown in Fig. 7. It can be seen that the position error of the proposed method is smaller than that of the color-based tracking throughout almost the entire tracking process.

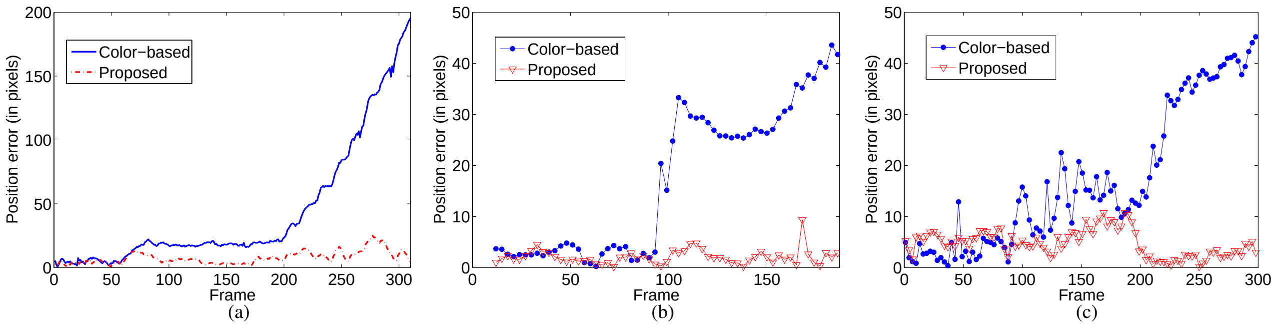


Fig. 7. Position error curves: (a) Results in S1; (b) Results in S2; (c) Results in S3.

Frame 023 Frame 200 Frame 309

Fig. 5. Tracking results in automatic initial position.

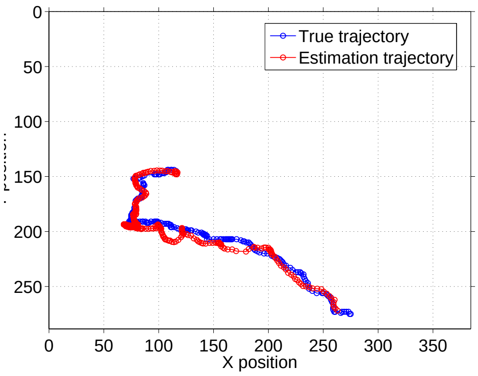


Fig. 6. True and estimate positions in Fig. 5.

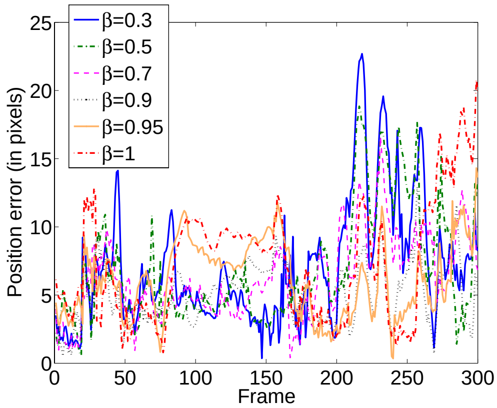


Fig. 8. β discussion in sequence S1.

TABLE I
MEAN AND VARIANCE OF POSITION ERRORS FOR DIFFERENT β.

	β=0.3	β=0.5	β=0.7	β=0.9	β=0.95	β=1
Mean	7.04	6.83	6.27	5.09	6.19	7.44
Variance	19.77	16.84	9.05	5.14	7.48	18.38

V. DISCUSSION OF PARAMETERS

There are mainly two parameters in the method: the update factor α and the integrating factor β. The decision of optimal value of each parameter is difficult. In this section, we will discuss the chosen of parameters.

The update factor α controls the rate of feature adaptation of the target model to the candidate model. The larger update rate shows the larger environmental fluctuations, while smaller update rate denotes fewer changes of environment. Experimental results show that setting $\alpha = 0.01$ is already efficient in different complex environments. So in this paper,we set $\alpha = 0.01$.

The integrating factor β is a parameter to adjust the proportion between two features which has the range from 0 to 1. Value of β is set according to different databases. Large β shows the high reliability of the GAP feature, while small β shows the high reliability of the color feature. We test the performance using different values of β in sequence S1, and the results are shown in Fig. 8. To compare results easily, we calculate the mean and variance of position errors in Table I and also show the relationship between them in Fig. 9. It is shown that the larger the β is, the smaller the mean and variance of position error are. But if β is too large (for example, $\beta = 1$), high mean position error (7.44) and high variance of position error (18.38) are caused. In this sequence, the optimal value of β is 0.9. This demonstrates that the GAP feature is more appropriate than the color one in this particular sequence. When one feature is undoubtedly more appropriate than the other one in a particular setting, it is better to attribute to it a larger weight in the weighted summation. For instance, in the case with confusing background or changing illumination, the GAP feature is more reliable than the color feature, so that the large β is set. On the other hand, in the case with crowded moving objects, the GAP feature is less reliable, so β is better to be set small. However, in most of the situations, it is difficult to determine which feature is more reliable since the real environment is complex, so we suggest $\beta = 0.5$ to obtain the equal weight.

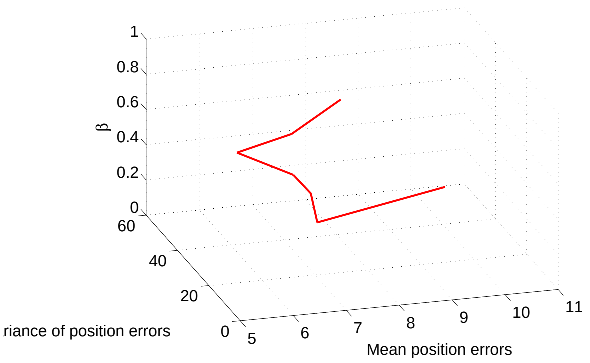


Fig. 9. The relations among β, Mean and Variance of position errors.

VI. Conclusions

We proposed a hybrid feature based adaptive particle filter algorithm for tracking. The observation model in particle filtering framework is built including two different types of features: the color histogram feature, which has high ability to accurately identify the detected object, and the GAP feature, which has high sensitivity in discriminating between the background and the objects. In addition, an updating target model is proposed to make the algorithm more robust. Meanwhile, different from the common algorithms, an automatic initialization algorithm is introduced which provides precise initial positions of objects. Experimental results show the good performance of the algorithm.

Acknowledgment

We would like to thank Prof. T. TANAKA for his helpful suggestions and assistance in our lab. We would also like to thank Dr. K. IWATA and Mr. R. OZAKI in National Institute of Advanced Industrial Science and Technology (AIST) for their helpful discussion and contributions in the development of this work.

References

[1] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 886–893, 2005.

[2] K. Nummiaro, E. Koller-Meier, and L. Van-Gool. An adaptive color-based particle filter. *Image and Vision Computing*, 21:99–110, 2003.

[3] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. Color-based probabilistic tracking. In *European Conference on Computer Vision (ECCV)*, pages 661–675, 2002.

[4] J. Wang, X. Chen, and W. Gao. Online selecting discriminative tracking features using particle filter. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1037–1042, 2005.

[5] Xinyue Zhao, Yutaka Satoh, Hidenori Takauji, Shun'ichi Kaneko, Kenji Iwata, and Ryushi Ozaki. Robust moving object detection based on a statistical model. In *The Sixteenth Korea-Japan Joint Workshop on Frontiers of Computer Vision*, pages 283–288, 2010.

[6] Xinyue Zhao, Yutaka Satoh, Hidenori Takauji, Shun'ichi Kaneko, Kenji Iwata, and Ryushi Ozaki. Object detection based on a robust and accurate statistical multi-point-pair model. *Pattern Recognition*, 44(6):1296–1311, 2011.