순차적 파티클 필터를 이용한 다중증거기반 얼굴추적

Probabilistic Head Tracking Based on Cascaded Condensation Filtering

김 현 우[1], 기 석 철[2]

Hyunwoo Kim[1], Seok-Cheol Kee[2]

Abstract This paper presents a probabilistic head tracking method, mainly applicable to face recognition and human robot interaction, which can robustly track human head against various variations such as pose/scale change, illumination change, and background clutters. Compared to conventional particle filter based approaches, the proposed method can effectively track a human head by regularizing the sample space and sequentially weighting multiple visual cues, in the prediction and observation stages, respectively. Experimental results show the robustness of the proposed method, and it is worthy to be mentioned that some proposed probabilistic framework could be easily applied to other object tracking problems.

Keywords : Human Robot Interaction, Face/Head Modeling And Tracking, Particle Filter, Multiple Evidence Fusion, Pose Estimation

1. Introduction

The visual analysis of human heads/faces is a key element of human robot interaction. The technical issues include facial feature detection/extraction and tracking, person identification, expression analysis, and 3-D model reconstruction. The applications cover personal and military security, ubiquitous computing and intelligent environment, high-level video semantics & understanding, and content-based video service.

In order to visually track human heads, researchers have been found several useful cues, such as color, shape, and motion. First of all, color cues have been widely developed because of its simplicity and real-time implementation ability. Comaniciu and Meer[1] showed a successful tracking system utilizing color cues, based on the Mean Shift algorithm. They modeled tracking objects as a color probability distribution, and in the tracking stage they estimated the position and scale of the object by searching candidate regions with a metric derived from the Bhattacharyya coefficient. Bradski[2] modified the algorithm to deal with dynamic distribution changes and they call it the CAMSHIFT (Continuously Adaptive Mean Shift) algorithm.

Differently, Isard and Blake[3] introduced a probabilistic framework called the Condensation (Conditional density propagation) algorithm as the transfer of the particle filter, and it has been highlighted as a new statistical framework for visual tracking. Originally they used shape cues of the object including hand, face, and shoulder, and then many researchers have utilized the Condensation framework by incorporating with color cues[4-6]. Especially, Jang and Kweon[6] reported their robust and real-time face tracking algorithm based on a skin-color model.

Vermaak et al.[7] utilized both motion and color cues to adapt face color and background model simultaneously. The approach is to selectively adapt the head color model based on the motion information, and it is performed using a

Received : Jun. 8. 2010; Reviewed : Jul. 20. 2010; Accepted : Aug. 6. 2010

※ 이 논문은 일부2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행됨(2010-0004359).

[1] 한독미디어대학원대학교 (KGIT; Korean German Institute of Technology) 뉴미디어학부 조교수 (Email: hwkim@kgit.ac.kr)

[2] ㈜만도 중앙연구소 산하 전자연구소 연구소장

stochastic EM (Expectation Maximization) algorithm. In spite of sound formularization and good performance, their algorithm is limited to fixed cameras in the assumption.

Nait-Charif and McKenna[8] combined color cue with shape (gradient) cue. They introduced the ILW (Iterative Likelihood Weighting) scheme, achieving accurate tracking even when the modeling of motion dynamics is poor. In the PETS-ICVS 2003 workshop, they presented outstanding results on the same video data set given by committee, compared with other participants.

In this paper, we present an accurate and efficient tracking algorithm, which utilizes color and shape cues based on the condensation filter. The proposed algorithm is intended for mobile platforms, so motion cue is excluded, and it is more focused on head tracking and face-related applications. Our head modeling covers the arbitrary head motion including the motion giving the backside of the head, and we herein extended the condensation algorithm to improve the robustness and speed. The modification includes the cascaded multiple cue usage and regularization of sample space. Moreover, head pose estimation gives more functionality for HCI applications.

Sections 2 and 3 describe the probabilistic model and the extra functionalities, respectively. Section 4 gives the experimental results and Section 5 concludes.

2. Probabilistic Head Modeling and Tracking

Our tracking system has three major components: initialization, prediction and update, like other standard tracking systems. In the initialization stage, human heads can be detected manually or automatically. Here we assume human heads are given already by a face/head detector. In the prediction stage, the motion parameters, i.e., the position and scale of each heads, are predicted using given (assumed or learned) system dynamics, and in the update stage the motion parameters are updated based on the image measurements. Finally, the prediction and update are performed iteratively in video streams.

We develop the probabilistic framework, a condensation algorithm to combine multiple cues in Bayesian rule, and in addition, the framework is improved to deal with multiple cues efficiently and to be robust to scale changes. The flow chart of the proposed algorithm is shown in Figure 1. When a frame is captured by a camera the heads/faces are assumed to be detected by the initialization stage. To track the detected heads, the prediction and update stages are followed. In the figure, our modification is shown in blue (shown in gray-filled box for b/w printer). The regularization of sampling space and the cascaded filtering for update will be described in the following Subsections.

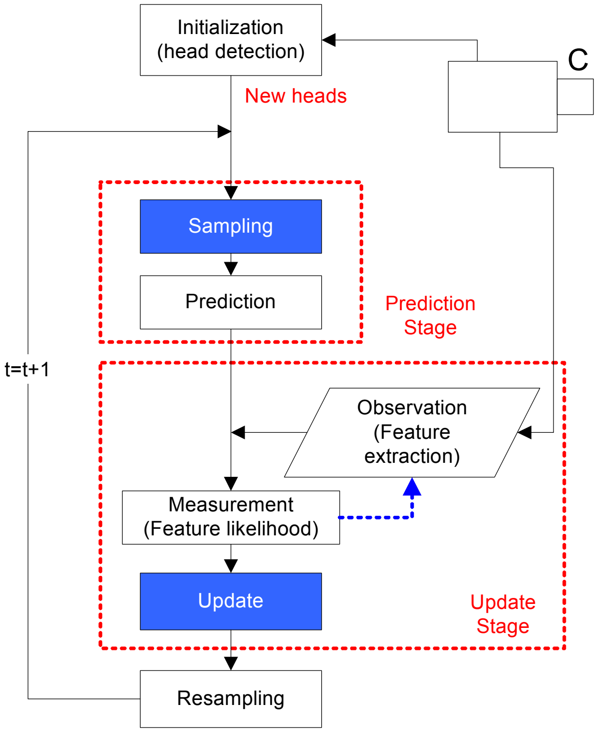


Fig. 1. The flow chart

2.1 Review of Condensation algorithm

Suppose that x_k and $D_k = \{y_0, y_1, \cdots, y_{k-1}, y_k\}$ denote the state vector and the measurement vector at the discrete time k. According to the Bayesian theory and probability propagation theory, a tracking framework can be formulated by

$$p(x_k \mid D_{k-1}) = \int p(x_k \mid x_{k-1}) p(x_{k-1} \mid D_{k-1}) dx_{k-1} \quad (1)$$

$$p(x_k \mid D_k) = \frac{p(y_k \mid x_k) p(x_k \mid D_{k-1})}{p(y_k \mid D_{k-1})} \quad (2)$$

Equation (1) and (2) correspond to prediction and update stages, respectively. Generally, the probabilities cannot be represented by deterministic models, so they are implemented by Monte-Carlo simulation. For details, refer to [3].

2.2. Prediction: Regularizing Sample Space

In the prediction stage, at first the sampling is done by the important sampling theory, then the samples are drifted as follows.

$$X(t) = A * X(t-1) + B * U + N(0,1) \quad (3)$$

where $X(t)$ is the system states, matrix A is the state transition matrix, matrix B couples the input into the system, N is the random noise to model the system diffuse. We call it the random drift method, because it assumes the dynamics systems can be modeled as random variable, i.e., consistent distributed Gaussian noise.

The random drift method, however, may be found to be unstable in their prediction and update stages. Especially, scale estimation is unstable because scale change does not exactly correspond to the image pixel change. Therefore, we propose a new drift method, called the regular drift method. The regular drift method regularizes the system drift at fixed distribution pattern, which also adapts itself to current state.

Compared with the conventional random drift, the regular drift is represented by

$$X(t) = A * X(t-1) + B * U + Factor * Pattern \quad (4)$$

where *Factor* denotes the coefficient adaptive to the current state. *Pattern* denotes the proposed samples distributions arranged as shown in Fig. 2, including the position and scale distribution.

In Fig. 2, the center point represents the samples transition state. Circle points represent the offsets of each sample based on its transition state. Although sometimes the difference is not very clear, it really exists showing the regular drift can achieve better than random drift.

In our visual face-tracking instance, the object's state is depicted as its position and scale, approximated with the center point (horizontal coordinate and vertical coordinate), short axis and long axis of ellipse. The scale also drifts regularly, red circles (bounded by rectangles) representing the increment and blue circles (bounded by diamonds) representing the decrement at the fixed ratio, and black circle representing zero changes of scale.

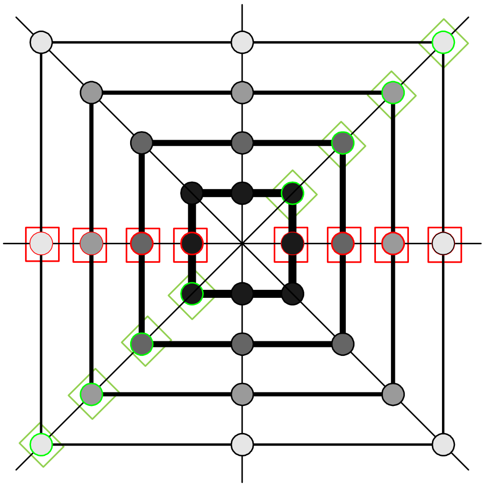


Fig. 2. Particle Diffusion Pattern

2.3. Update: Cascaded Weighting Filter

In the update stage, we propose the cascaded weighting filters, which can efficiently weight the samples. Since, in Monte-Carlo simulation, samples with a limited number approximate a distribution, the error between the approximations and its real function exists. In addition, more samples give more accurate approximation.

Our idea is to reduce the number of samples to be evaluated for weighting without the sacrifice of performance. It happens when a specific visual cue (e.g., color) can be coarsely evaluated than other cues (e.g., shape and scale). The coarse cue is firstly observed and the other cues are finely observed without increasing the number of samples. Assuming the sample number is fixed, we will show that we can find the best solution by changing weighting strategy. While conventional methods weight all of the samples with all evidence (cues), the samples are weighted in a cascaded way. For head tracking, the cascaded filters compose of contour filter, color filter, and scale filter, as shown in Fig. 3.

2.3.1 Color Filter

In the color filter, the hair color and skin color model

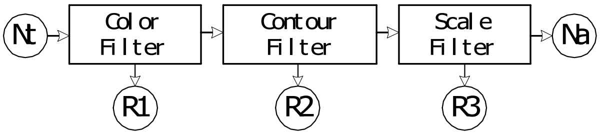


Fig. 3. Cascaded weighting filters. Nt and Na denote total sample number from the prediction and update stages, respectively. R1, R2, and R3 are the filter responses of color, contour, scale filters, respectively

are employed. Many objects in the natural world are similar to the human skin and human hair in color. But the combination of hair color model and skin model can weight the samples very accurately. In the area where each particle covered, the pixels are classified into three classes: skin pixels, hair pixels and distracter pixels. It is obvious the particles with explainable combination of hair blobs and skin blobs can get the high weight.

Suppose that *skin*(*i*), *hair*(*i*), and *t*(*i*) denote skin, hair, and total pixels at sample *i*, respectively. The weight component of color for each particle is calculated by

$$W^{color}(i) = Saturation(i) + THair(i) + TSkin(i) \quad (5)$$

where

$$Saturation(i) = (skin(i) + hair(i)) / t(i) \quad (6)$$

$$THair(i) = hair(i) / \sum_{n=1}^{N} hair(n) \quad (7)$$

$$TSkin(i) = skin(i) / \sum_{n=1}^{N} skin(n) \quad (8)$$

and N is the total particle number. In equation (6), the saturation indicates the percentage of pixels belonged to the head models in the ellipse. Equation (7) measures the percentage of hair pixels in one certain particle to the hair pixels of all particles. It is same to the equation (8), measuring the skin pixels percentage. After evaluating the individual, individual to all, the weigh can be assigned as equation (5).

2.3.2 Contour Filter

In the contour filter, the ellipses with certain long axis, the short axis and their aspect ratio are used to represent the head shape. Along the head contour, usually, there are strong image contrast and gradient features. Based on the observation map above, the gradient map can be obtained too. The contour filter will response powerfully to the particles with strong gradients at the perimeter, thus high weights can be assigned to them. Assuming $G(m)$ and $\vec{F}(m)$ represent the gradient magnitude and the gradient normal vector at perimeter *m*, respectively, contour response intensity $T_g(i)$ at that particle *i* is represented by

$$T_g(i) = \sum_{m=1}^{M} G(m) \cdot \overline{F}(m) \quad (9)$$

Therefore, the response function of contour filter for weight component can be written as

$$W^{contour}(i) = T_g(i) / \sum_{i=1}^{N} T_g(i) \quad (10)$$

2.3.3 Scale Filter

In the scale filter, we assume the objects scale changes continuously in the consecutive frames. A maximum zoom factor between two contiguous frames is set, and then the scale changes between frames are smaller than this number. Let $\{S^i,\ i = 1, 2, \ldots N\}$ be a particle set, $\{X_c,\ Y_c,\ SZ_x,\ SZ_y\}$ be state estimation at the last time slot, and $g^{MaxZoom}$ be maximum zoom factor. First, the size differences (δ_x^i and δ_y^i) between particle *i* and estimation for width and height are computed by

$$\delta_x^i = \left| SZ_x^i - SZ_x \right| \quad (11)$$

$$\delta_y^i = \left| SZ_y^i - SZ_y \right| \quad (12)$$

So, the covariance can be obtained as

$$\delta = \delta_x^i \cdot \delta_x^i / \sigma_x^2 + \delta_y^i \cdot \delta_y^i / \sigma_y^2 \quad (13)$$

$$\sigma = \sqrt{\sigma_x^2 + \sigma_y^2} \quad (14)$$

The current cut value for the allowed size shift for the object width and height can be calculated as

$$T_{szx} = (gMaxZoom - 1) \cdot SZ_x \tag{15}$$

$$T_{szy} = (gMaxZoom - 1) \cdot SZ_y \tag{16}$$

Weight component of scale for particle i can be evaluated in the section functions by

$$W^{scale}(i) = \frac{1}{\sigma} e^{-\delta} \text{ , if } \delta_x^i > T_{szx} \text{ and } \delta_y^i > T_{szy} \quad (17)$$

$$W^{scale}(i) = 1 \text{ ,} \qquad \text{otherwise} \quad (17)$$

So, the final cascaded weighting filters can be defined by

$$W^{(i)} = W^{color}(i) \cdot W^{contour}(i) \cdot W^{scale}(i) \tag{18}$$

3. Advanced Color Model and Pose Estimation

The human skin and hair have its individual color. Especially the human skin, although it has changes under different illumination and camera parameters, many researchers like to use it as the important features of human faces. In the real world, there are many scenes in which objects are similar to the human skin in color, it is so for the human hair. In order to reduce the negative effects from distracters and take best advantage of the human head information, we propose to use the combination information of human skin color, human hair color and explainable topology of them.

3.1. Advanced Color Model

For the natural skin color and hair color, each has its own distribution range. When imaging condition changes, the distribution will also change. So, there is no fixed color model for them. In order to model the color as accurately as possible without making it difficult to use friendly, we firstly train the color model offline; then adapt this offline model to the real scene by labeling the object manually in the first video frame. Then the algorithm will learn to discriminate the labeled human heads and other objects in the scene, aiming at the maximum classification accuracy and minimum miss-classifications. To the long time video, the algorithm can learn the tracking results at certain frequency and update itself automatically.

Normally, skin color models are widely used by researchers; hair color model is less used. In order to reduce the negative effects from distracters and take best advantage of the human head information, we propose here to use the combination information of human skin color, human hair color and explainable topology of them. In this section, we will discuss the benefits from the combination of two models. From the view of statistics, the probability of skin distracters and the probability of hair distracters are much higher than the probability of joint probability of hair and skin blobs in the real scenes. So, the random distracters can be dramatically reduced by the combination of two models. Assuming P^{skin}, P^{hair}, and P^{head} , which denote skin-like blobs probability, Hair-like blobs probability, and Human heads like distribution probability respectively, are given, the probability model can be described as

$$P^{head} = P^{skin} \cdot P^{hair} << P^{skin} \tag{19}$$

$$P^{head} = P^{skin} \cdot P^{hair} << P^{hair} \tag{20}$$

3.2. Pose Estimation

Similar to face recognition, face pose estimation is also an important issue. As the additional work in this paper, we can only use the color information to estimate the poses. The method is a little coarse, but the algorithm can work efficiently. But much more work is needed to make it robust.

The face-hair models at different poses are shown in Fig. 4. The head blobs are divided into 8 X 4 rectangles. The distribution patterns of rectangles at different poses can be characterized into its special patterns. Unfortunately, different person has different patterns, so there does not exist the common pattern from the training of samples. What we can do is to learn the individual patterns from the video frames. So it's helpful in the video editing,

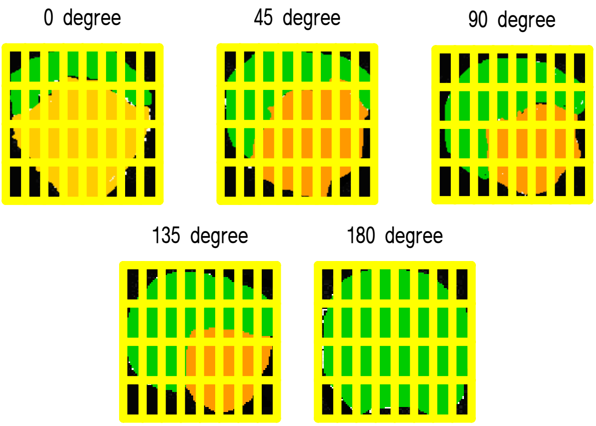
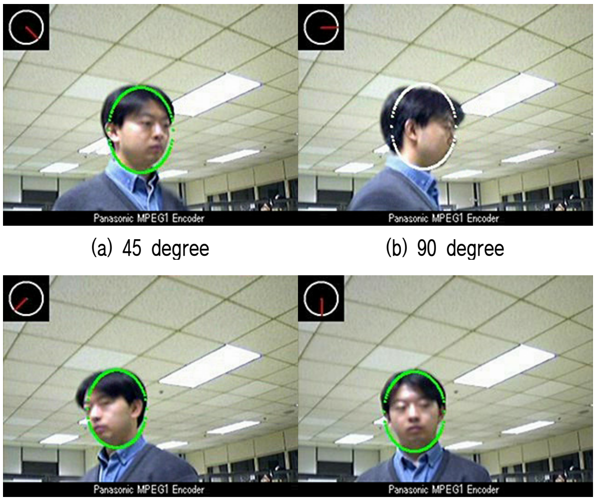


Fig. 4. Pose definition.

videoconferences, and smart room etc. We define the different pan poses, separating the view-space into eight parts. In Fig. 4, five poses are shown with grid masks. Fig. 5 is the face poses estimation result.



(a) 45 degree

(b) 90 degree

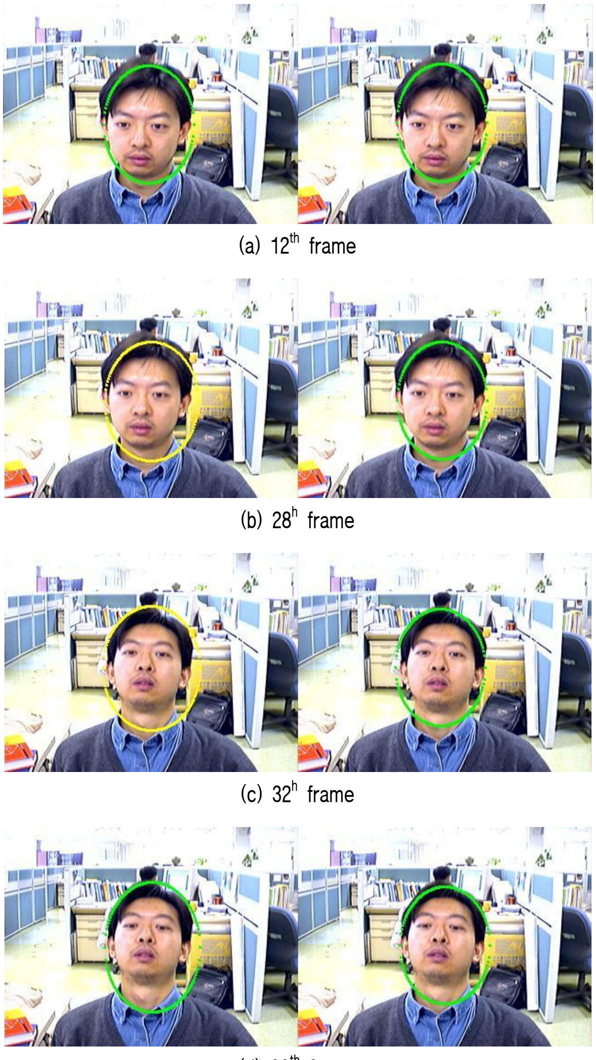
(c) -45 degree

(d) -90 degree

Fig. 5. Pose Estimation

4. Experimental Results

In this section, we compare the proposed tracking method with the random drift patterns and the usual weighting strategies. In Figure 6, we give an instance showing the results from different particles drift method. From the figure, we can find the better tracking performance of regularized drift pattern, comparing with the random drift method. Usually, we cannot precisely predict the objects state at next time point. So, the uniform distribution is the better choice to prevent its escaping. In order to incorporate the most current information, that is the final estimation at last time point, the particles are reordered according to their weights, and formed from the 4D space into the 2D pattern shown in Figure 2. The principle inside is the continuity in spatial and temporal will encourage the high weighting particles and punish the low weighting particles with more displacements in state space. Another function of this drift pattern is to compress the state space from 4D to 2D, initially reducing the particles number.



(a) 12th frame

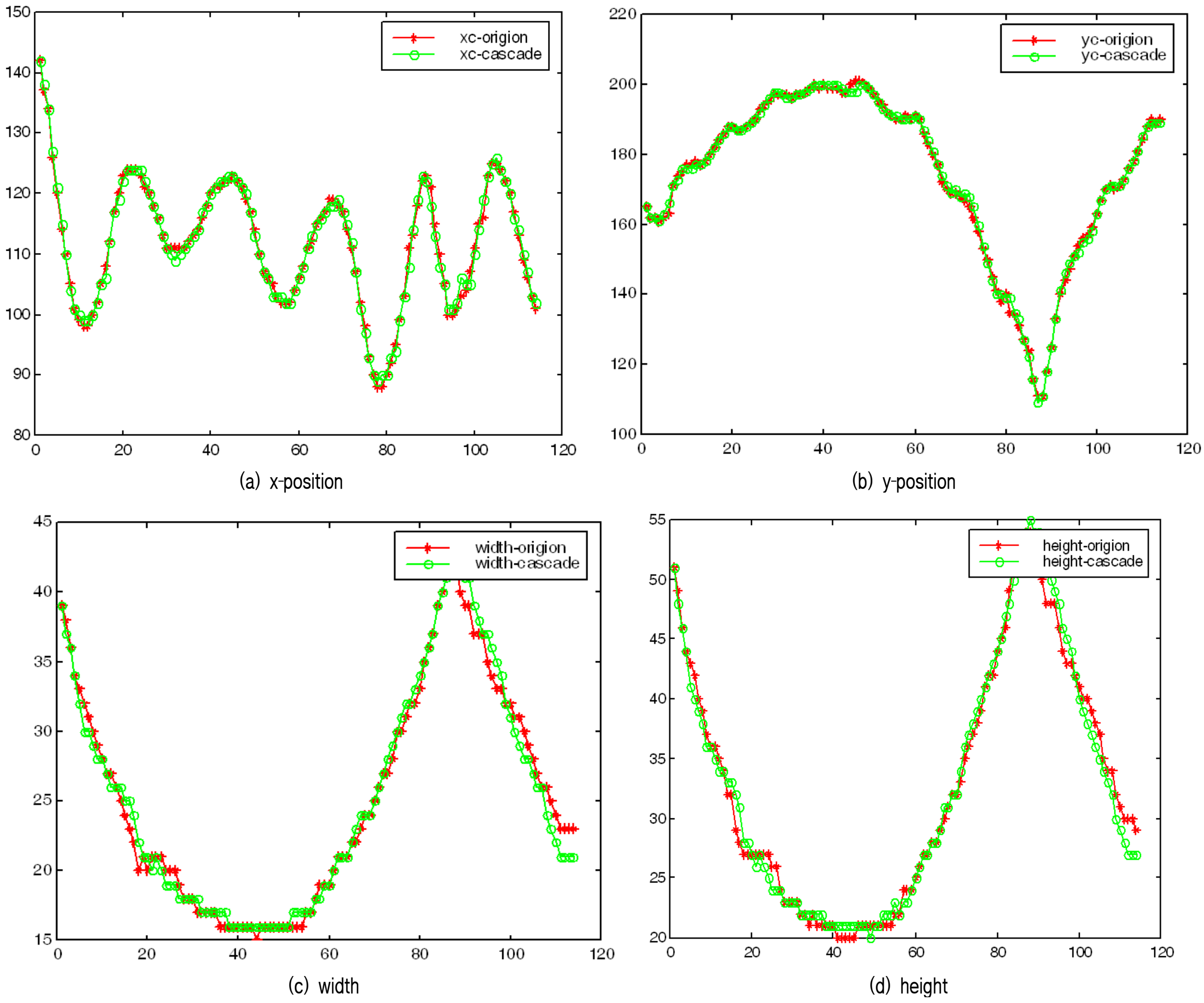
(b) 28h frame

(c) 32h frame

(d) 36th frame

Fig. 6. Comparison of different drift patterns. (Left) The results of Random drift, (Right) The results of Regular Drift.

In Figure 7, we compare in the curves the tracking performance with and without cascaded weighting strategies employed. Referring to figure 3, the particles number at



(a) x-position

(b) y-position

(c) width

(d) height

Fig. 7. Comparisons of tracking performance of different weighting strategies

each stage are as follows: $N_t = 64$, $R_1 = 12$, $R_2 = 8$, $R_3 = 8$, $N_a = 36$. No matter on the tracking of objects position or the scale, the two methods achieve almost the same performance. In order to eliminate the possibility of particles number selected more than the necessary, we reduce the particles number from 64 to 36, and evaluating and weighting them in usually method by turning off the function of cascaded weighting strategy, the performance becomes much more poor in tracking accuracy and robustness.

5. Concluding Remarks

In this paper, two tracking problems were addressed, one was how to better drift the particles in the probabilistic framework, and the other was how to weight the particles accurately and efficiently to reduce the computation cost without changing the tracking performance.

Experiment results demonstrated the proposed regularized drift pattern outperforms the random one, especially in the clutter scenes. The cascaded weighting strategies can reject the particles properly during the weights evaluation, thus the computation cost is drastically decreased and the high tracking accuracy and robustness are kept.

On the objects state representation, we formed the 4D state space into the 2D pattern space, cutting down the original necessary particles number for the high performance, so the necessary computation cost is also reduced greatly.

참고문헌

[1] D. Comaniciu, and P. Meer, "Mean Shift: A Robust Approach toward Feature Space Analysis," IEEE Trans. Pattern Analysis Machine Intelligence., 24(5): 603-619, 2002.

[2] G. R. Bradski, "Computer Vision Face Tracking For Use in a Perceptual User Interface," Intel Technology Journal Q2, 1998.

[3] M. Isard, and A. Blake, "Condensation-conditional density propagation for visual tracking," International Journal of Computer Vision, 29(1): 5-28, 1998.

[4] K. Nummiaro, E. Koller-Meier, and L. V. Gool, "An adaptive color-based particle filter," Image and Vision Computing, 21(1): 99-110, January 2003.

[5] P. Pérez, C. Hue, J. Vermaak and M. Gangnet, "Color-based probabilistic tracking," European Conference on Computer Vision, Copenhagen, Denmark, June 2002.

[6] G. J. Jang, and I. S. Kweon, "Robust Object Tracking Using an Adaptive Color Model," International Conference on Robotics and Automation, pp. 1677-1682, Seoul, Korea, May 2001.

[7] J. Vermaak, P. Pérez, M. Gangnet, and A. Blake. "Towards improved observation models for visual tracking: selective adaptation," European Conference on Computer Vision, Copenhagen, Denmark, June 2002.

[8] H. Nait-Charif, and S. J. McKenna, "Head Tracking and Action Recognition in a Smart Meeting Room," IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, Graz, Austria, 31 March 2003.

김 현 우

1994 한양대학교 전자통신공학과(공학사)
1996 포항공과대학교 전자전기공학과(공학석사)
2001 포항공과대학교 전자전기공학과(공학박사)
2001~2007 삼성종합기술원/삼성전자 책임연구원
2007~2008 ㈜한독산학협동단지 책임연구원
2008~2009 성균관대학교 연구부교수
2009~현재 한독미디어대학원대학교(KGIT) 조교수
관심분야 : 컴퓨터 비전, 인지 로보틱스, 증강현실, 영상처리

기 석 철

1987 서울대학교 제어계측공학과(공학사)
1989 서울대학교 제어계측공학과(공학석사)
2001 서울대학교 전기공학부(공학박사)
1989~2007 삼성종합기술원 수석연구원
2007~2010 ㈜로봇에버 연구소장
2010~현재 ㈜만도 중앙연구소 산하 전자연구소장
관심분야 : 컴퓨터 비전, 영상처리, 생체인식, 로봇비전, 지능형 자동차