한국항행학회 논문지 제10권 제1호 2006년 3월

복잡한 배경영상에서 효과적인 전처리 방법을 이용한 표적 중심 추적기

Efficient Preprocessing Method for Binary Centroid Tracker in Cluttered Image Sequences

조재수*

Jae-Soo Cho*

요 약

본 논문에서는 복잡한 배경영상에서 움직이는 물체를 자동으로 추적하는 표적중심 추적기의 효과적인 전처리 방법을 제안하였다. 이진 표적중심 추적기의 성능은 다음과 같은 요소가 추적성능을 좌우한다: (1) 효과적인 실시간 전처리 방법 (2) 복잡한 배경영상에서의 정확한 표적 추출방법 (3) 지능적인 표적창 크기 조절법. 본 논문에서 제안하는 표적중심 추적기는 배경과 움직이는 표적을 좀 더 쉽게 판별할 수 있도록 추적필터를 이용한 효과적인 실시간 전처리 방법에 의한 적응적인 표적분할방법을 사용한다. 효과적인 전처리 방법이란 추적필터에 의해 추정된 표적중심을 중심으로 입력영상에 다른 가중치를 줌으로써 표적과 배경을 더 쉽게 분리할 수 있다. 제안한 방법은 합성영상 및 실제 적외선 영상을 이용한 다양한 추적실험을 통하여 그 효용성 및 성능을 검증하였다.

Abstract

This paper proposes an efficient preprocessing technique for a binary centroid tracker in correlated image sequences. It is known that the following factors determine the performance of the binary centroid target tracker: (1) an efficient real-time preprocessing technique, (2) an exact target segmentation from cluttered background images and (3) an intelligent tracking window sizing, and etc. The proposed centroid tracker consists of an adaptive segmentation method based on novel distance features and an efficient real-time preprocessing technique in order to enhance the distinction between the objects of interest and their local background. Various tracking experiments using synthetic images as well as real Forward-Looking InfraRed (FLIR) images are performed to show the usefulness of the proposed methods.

Key words : Binary Centroid Tracker, Preprocessor, Target Segmentation

I. Introduction

Automatic video tracking systems are employed in a wide variety of missions and tracking environments, such as fire control, guidance, autonomous vehicle navigation[1]-[7]. Many different methods of estimation of a target location have been developed. The most common

* 한국기술교육대학교 인터넷미디어공학부(Internet Media Engineering, Korea University of Technology and Education)
· 제1저자 (First Author) : 조재수
· 접수일자 : 2006년 1월 5일

(and probably the best known) method among tracking algorithms is a centroid target

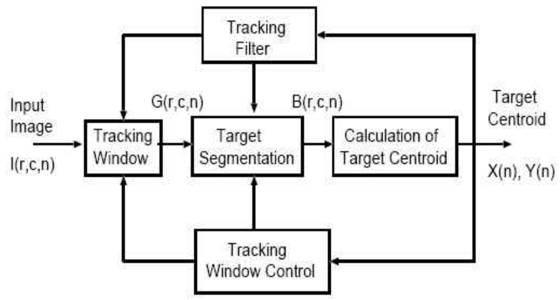


Fig. 1. Binary centroid tracker.

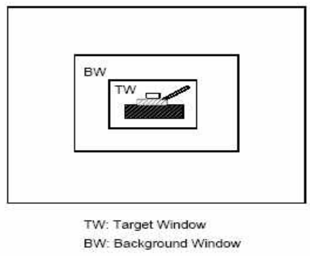


Fig. 2. General geometry of tracking window.

tracker[3]−[7]. There are two kinds of the centroid target tracker: One is a binary centroid tracker considered in this paper and the other is an intensity centroid tracker which uses the raw data directly. The centroid of the image presented by a target in the field of view (FOV) of a sensor is generally accepted as the aim point for autonomous fire and forget weapons. For hardened targets like tanks, proper aim point selection and guidance up to almost zero range is essential to achieve maximum kill probability. As the centroid offers a stable tracking point, it can be used as a reference to select the proper aim point. Figure 1 shows the block diagram of a general binary centroid tracker.

The target location estimate is often computed on a small subimage called a tracking window (target and background window), as shown in Figure 2, to sample the target and background intensities immediately adjacent to the target image. The target window ideally consists of all pixels located on the target object. The background window contains pixels in the background immediately surrounding the target. Based on the statistics of these two regions, a decision rule is formed so that each image pixel is classified as either a target pixel or a background one. This decision rule defines a mapping from the set of all possible pixel values into a binary set [3]−[7].

The performance of the binary centroid tracker can be increased by an efficient preprocessing method before target segmentation. The preprocessing method is applied to the digitized data stream in order to enhance the distinction between the objects of interest and their local background. A commonly−applied process was a local mean removal for the centroid target tracker reported in the literature[1]. Mean removal was used for tracking small targets against backgrounds with high horizontal noise content (such as horizon or thermal banding effects).

We will propose an efficient preprocessing algorithm in order to decrease th probability of pixel classification error in the target segmentation process. Furthermore, the segmentation effects on the target centroid estimation is also presented with some qualitative analysis.

II. Segmentation Effects on Target Centroid Estimation

When computing target position (aim point) from segmented binary images, a major problem is caused by noise, such as system and sensor noise, background clutters near the target, etc. If the segmentation process is accurate, most, if not all, of the target and background pixels will be correctly classified. In such cases the estimated centroid will be very close to the actual target centroid. However, if there is a large amount of sensor noise and/or if the target gray levels are similar to the surrounding background, segmentation will not be very good. This causes pixel classification errors and affects centroid computation.

An investigation of the effects of system and sensor noise on target tracking was presented in a previous study[3]. A simple way to model segmentation errors is to define a binary noise variable, $b(i,j)$, as follows:

$$b(i,j)=\begin{cases}0, & \text{if } pixel(i,j) \text{ is correctly classified,}\\ 1, & \text{if } pixel(i,j) \text{ is incorrectly classified.}\end{cases} \quad (1)$$

Therefore, for a given pixel (i,j), $b(i,j)$ is a Bernoulli distributed random variable:

$$p(b)=\Pr(e)^{b}[1-\Pr(e)]^{(1-b)},\ b=0,1 \quad (2)$$

where $\Pr(e)$ is the probability of a pixel classification error. The probability of a pixel classification error, $\Pr(e)$, is defined as the probability that a given pixel (i,j) in the target window will be misclassified. If the underlying PDFs of the target and the background pixels, $p(z|T)$ and $p(z|B)$, are known, $\Pr(e)$ can be estimated by integrating over the error regions of each PDF. For example, the corresponding PDFs are shown in Figure 3. The parameters m_t and m_b represent the respective means of the densities, and T_h represents the segmentation threshold computed by the binary centroid tracker. Background gray levels greater than T_h will be misclassified, as will target gray levels less than T_h. $\Pr(e)$ can be computed by using the formula

$$\Pr(e)=P(B)\int_{T_h}^{\infty}p(z|B)dz+P(T)\int_{-\infty}^{T_h}p(z|T)dz \quad (3)$$

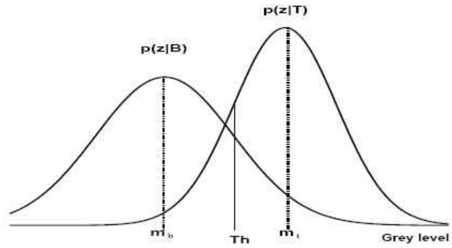


Fig. 3. Target and background PDFs.

where $P(B)$ is the probability that a pixel in the targe window is background, and $P(T)$ is the probability that a pixel in the target window is target.

The pixel classify function, $c(i,j)$, can be defined in terms of $b(i,j)$

$$c(i,j)=\begin{cases}1-b(i,j), & (i,j)\in Target\\ b(i,j), & (i,j)\in Background.\end{cases} \quad (4)$$

The x and y component of the target aim point error is redefined as

$$\overline{x}_t=x_t-\widehat{x}_t,\quad \overline{y}_t=y_t-\widehat{y}_t \quad (5)$$

where, $\overline{x}_t$ and $\overline{y}_t$ are target aim point error in x and y direction, respectively. x_t and y_t true target centroid points. $\widehat{x}_t$ and $\widehat{y}_t$ mean estimated target centroid points from the segmented image.

The expected value and the variance of the aim point error in x and y direction can be found by some approximations in [3] as follows:

$$\begin{bmatrix} E\{\overline{x_t}\} \\ E\{\overline{y_t}\} \end{bmatrix} \cong \begin{bmatrix} (x_t - \frac{N_b}{N_t} x_b)\Pr(e) \\ (y_t - \frac{N_b}{N_t} y_b)\Pr(e) \end{bmatrix} \quad (6)$$

$$\begin{bmatrix} var\{\overline{x_t}\} \\ var\{\overline{y_t}\} \end{bmatrix} \cong \begin{bmatrix} \frac{N_{tw}}{N_t^2} r_y^2 (\Pr(e) - \Pr(e)^2) \\ \frac{N_{tw}}{N_t^2} r_x^2 (\Pr(e) - \Pr(e)^2) \end{bmatrix} \quad (7)$$

where, x_b is x coordinate of true centroid of the background pixels in the target window, N_b is number of background pixels in the target window and N_{tw} number of pixels in the target window. r_x and r_y are Radiuses of gyration of the target window about x and y axes:

$$r_x^2 = \frac{\sum_{i \in TW} \sum_{j \in TW} j^2}{N_{tw}}, \quad r_y^2 = \frac{\sum_{i \in TW} \sum_{j \in TW} i^2}{N_{tw}} \quad (8)$$

Definition 1: A centroid error distance at the k^{th} image frame $Cerror(k)$ is defined as

$$Cerror(k) = \sqrt{\overline{x_t^2} + \overline{y_t^2}} = \sqrt{(x_t - \widehat{x}_t)^2 + (y_t - \widehat{y}_t)^2} \quad (9)$$

The centroid error distance, $Cerror(k)$, can be reduced by decreasing the probability of a pixel classification error, $\Pr(e)$. The results of target segmentation effects show if the centroid of the background pixels in the target window is equal to the centroid of the target pixels in the target window, the centroid bias is zero. Almost this condition does not happen in the ordinary target tracking scenarios. Notice that as with the aim point bias terms, the variance terms are directly proportional to the probability of a pixel classification error. It implies that if the segmentation algorithm can reduce the probability of a pixel classification error, then aim point jitter is decreased and the tracker performance is improved. Therefore, we will use the probability of a pixel classification error as a reference performance criterion to increase the performance of the binary centroid target tracker.

Ⅲ. Efficient Preprocessing Method

Figure 4 shows the proposed binary centroid

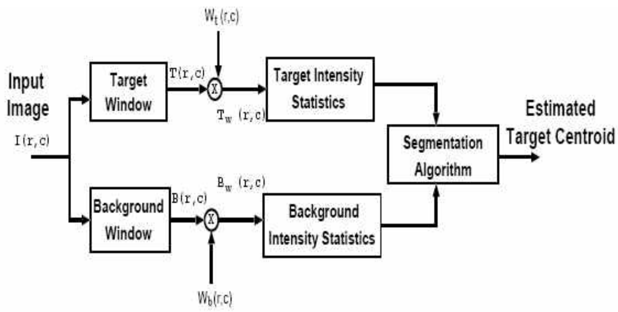


Fig. 4. The proposed binary centroid tracker.

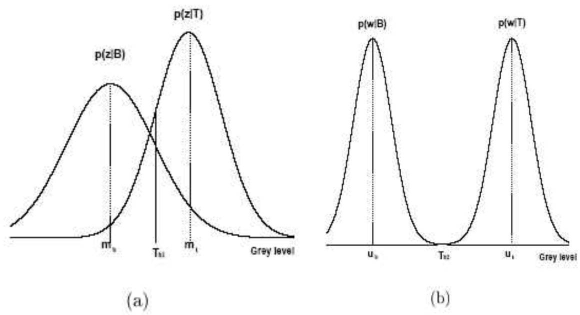


Fig. 5. (a) Target and background PDFs for the input image, (b) PDFs for the weighting windows of the respective target and background windows.

tracker with an efficient preprocessing. As shown in Figure 4, the windowed images of the target and background are spatially weighted by respective suppressing weighting windows, such as $w_t(x,y)$ and $w_b(x,y)$. The preprocessed images are simply modeled as:

$$g_t(x,y) = T[f_t(x-\hat{x}, y-\hat{y})] = w_t(x-\hat{x}, y-\hat{y}) \cdot f_t(x-\hat{x}, y-\hat{y}) \quad (10)$$

$$g_b(x,y) = T[f_b(x-\hat{x}, y-\hat{y})] = w_b(x-\hat{x}, y-\hat{y}) \cdot f_b(x-\hat{x}, y-\hat{y}) \quad (11)$$

where $f_t(x-\hat{x}, y-\hat{y})$ and $f_b(x-\hat{x}, y-\hat{y})$ are the gated target and background input image centered at the prediction pixel $(\hat{x}, \hat{y})$ such as in Figure 2, $g_t(x-\hat{x}, y-\hat{y})$ and $g_b(x-\hat{x}, y-\hat{y})$ are the respective preprocessed images, and T is an operator on f, defined over the tracking window. w_t and w_b are kinds of mask, in which the values of the coefficients are predetermined with the same size of the tracking window.

This masking process decreases the intensities of the background images near the target image more than those of the target image and, consequently, produces an increase in the tracking contrast. Therefore, we can decrease the probability of a pixel classification error. A simple graphical explanation of the proposed method is shown in Figure 5 using the related PDFs. The PDFs of the respective input images for the target and the background are shown in Figure 5(a). Figure 5(b) shows the resultant PDFs of the preprocessed target and the background images. The probability of a pixel classification error based on the preprocessed images can be reduced if we determine proper weighting windows which are mutually independent of the input corresponding images.

The principal objective of the preprocessing is to process the input image so that the result is more suitable than the original image for moving target segmentation. A flow chart of the proposed real-time centroid tracker is shown in Figure 6.

The procedure can be summarized as follows:

Step 1: Detect moving targets and initialize the required parameters. Using proper target recognition algorithms or detection techniques, it is required to detect a cenroid point of a moving target and to determine the initial parameters for the binary centroid tracker.

Step 2: Determine a weighting window based on the tracking windows.

Step 3: Weight the windowed input image by the determined weighting window in Step 2.

Step 4: Determine the pixel classification rule based on the statistics of the target and background.

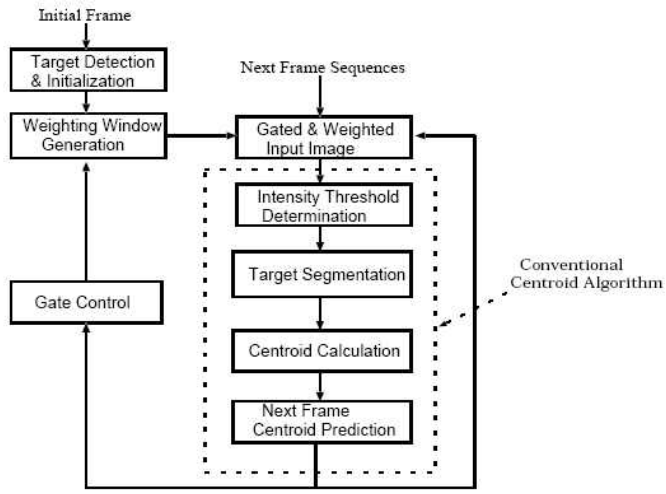


Fig. 6. Flow chart of the proposed method.

Step 5: Segment the target from the background in the target window and get the binary image.

Step 6: Calculate the centroid of target from the binary image.

Step 7: Predict the target centroid of next frame using the tracking filter and determine the target size in order to determine the sizes of the tracking window and go to Step 2.

The proposed tracker operates by first detecting target and initializing the other parameters, such as tracking window sizes, weighting window and the statistics of a moving target. Based on the statistics of the moving target, the target and background window of the next input image is spatially masked by the predetermined masking window based on the changing sizes of the tracking window. We used the Gaussian masking window.

Ⅳ. Experimental Results

In comparing trackers, a reference quantity of target characteristic with respect to background one for the target segmentation from the background images is needed. This quantity, Tracking Contrast(TC), is shown later to play a key role in the accuracy of the target segmentation. It is usually known that the higher one has the tracking contrast measure, TC, the lower probability of a pixel classification error can be obtained in the target segmentation.

$$TC = \frac{(\mu_t - \mu_b)^2}{\sigma_t^2 + \sigma_b^2} \tag{12}$$

where μ and σ are mean and standard deviation of the images, and the subscript t and b represent the target and background, respectively.

To show a performance of the proposed method, a number of experiments have been performed. In the experiments, various kinds of images were synthetically created in 256×256 size including a 50×50 rectangular target image. Various background images are also generated with respect to a Gaussian PDF, Target $N(150, 300)$, considering the tracking contrast. It is assumed that a predicted target center point from a tracking filter is centered at the real center point of target. The 100-independent Monte Carlo runs for each Tracking Contrast(TC) image are performed. In Figure 7, the probability of a pixel classification error by each segmentation method with the preprocessing method is compared with that of the original methods. The used segmentation methods are the ones described in the literature [5]-[7]. The overall results reveal that the proposed method has a lower probability of a pixel classification than those without preprocessing even for very low TC images.

The performance of the proposed method mostly depends on the ability of exact target prediction by a tracking filter. Experiments on various prediction errors by the tracking filter are performed in Figure 8. The experimental results show that the proposed method has good performances even in the abrupt prediction errors.

Figure 9(a) and (b) show a gated weight and a preprocessed image. Figure 9 (c) and (d) also show an overall weight and preprocessed image respectively. Figure 10(a) shows the histograms of the target and the background window for the statistics of the target and background in the original input image. Figure 10(b) also shows the same histograms in the preprocessed images. The contrast of the preprocessed image is increased by the suppressing weighting of the target-like background clutters. In Figure 11, some segmentation result by the optimal threshold are shown for a real image sequence. As shown in the above various experimental results, the proposed preprocessing method is very simple for a real-time application and has a good performance compared with that without preprocessing in the conventional binary centroid trackers.

Ⅴ. Conclusions

In this paper, we proposed an efficient preprocessing method for the binary centroid tracker. The purpose of the proposed preprocessing method is to increase the tracking contrast between the target intensity and the background one. The approach is to reduce the segmentation error by suppressing the target-like background clutters using the adaptive suppressing window. Eventually, the segmentation algorithm can reduce the probability of a pixel classification error and can increase the performance of the binary centroid tracker.

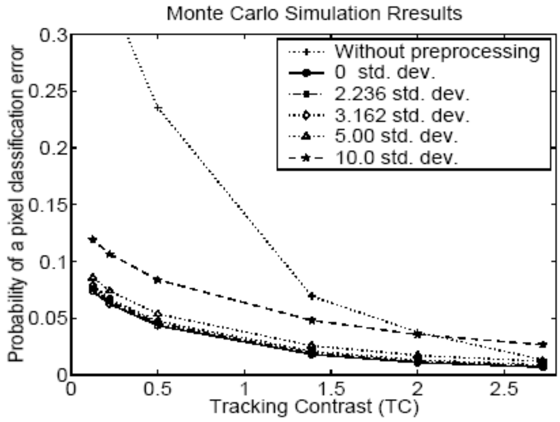
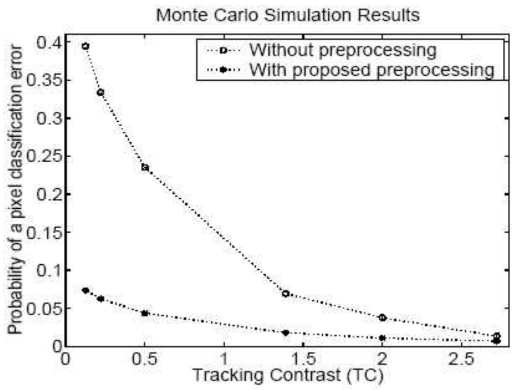
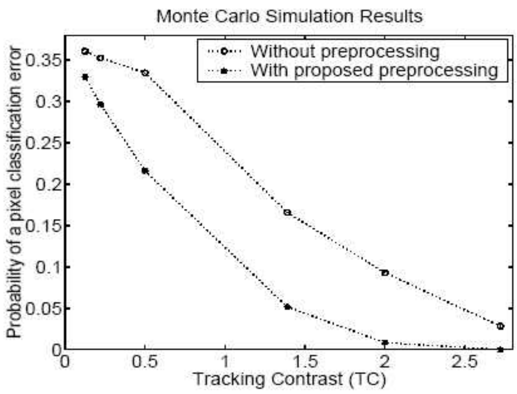


Fig. 8. Comparison of the probability of a pixel classification error for various prediction errors.

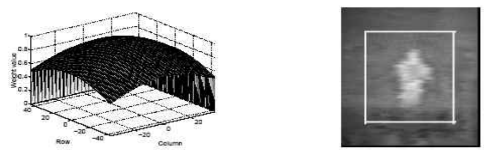


(a) Optimal threshold



(b) Optimal layering method

Fig. 7. Comparison of the probability of a pixel classification error.



(a) (b)

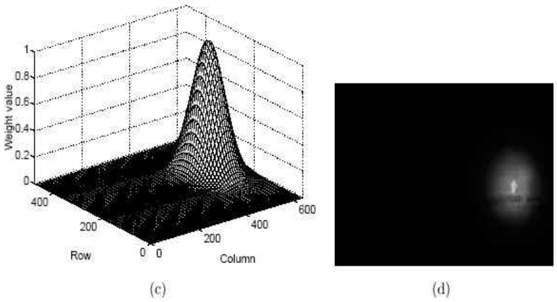


Fig. 9. Example of preprocessed image, (a) Gated weight, (b) Gated preprocessed image, (c) Weights for the overall image, (d) Overall preprocessed image.

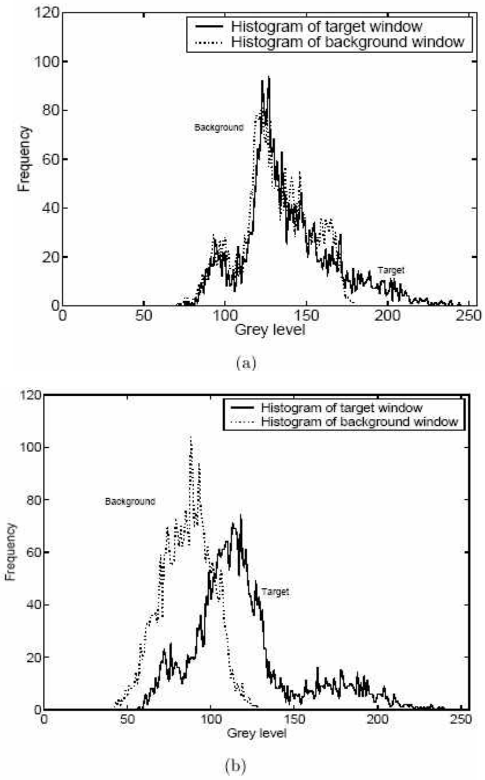


Fig. 10. (a) Target and background histograms for the input image, (b) Target and background histograms for the preprocessed image.

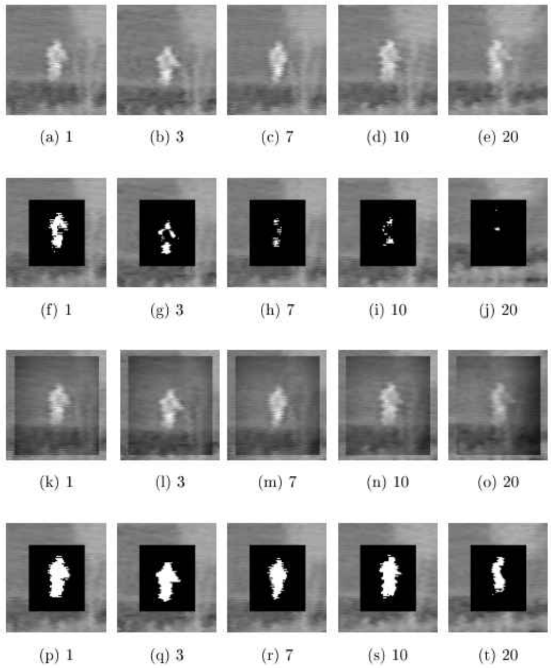


Fig. 11. Segmentation results of man image, (a)-(e): Real images, (f)-(j): Optimal threshold method, (k)-(0): Preprocessed images, (p)-(t): Optimal threshold method for the preprocessed images.

감사의 글

본 논문은 2005년도 정부재원(교육인적자원부 학술연구조성사업비)으로 한국학술진흥재단의 지원을 받아 연구되었음(KRF-2005-003-D00258)

참 고 문 헌

[1] M. C. Dudzik, Electro-optical systems design, analysis and testing, *The Infrared and Electro-Optical Systems Handbook,* SPIE, vol. 4, pp. 245-298.

[2] N. R. Pal and S.K. Pal, "A review on image segmentation techniques," *Pattern Recognition,* vol. 26, no. 9, pp. 1277-1294, 1993.

[3] D. R. Van Rheeden and R. A. Jones, "Noise effects on centroid tracker aim point estimation," *IEEE Trans. on Aerospace Electron. Syst.,* vol. 24, no. 2, pp. 177-185, March 1988.

[4] E. Oron, A. Kumar and Y. Bar-Shalom, "Precision tracking with segmentation for imaging sensors," *IEEE Trans. on Aerospace Electron. Syst.,* vol. 29, no. 3, pp. 977-986, July 1993.
[5] A. Kumar, Y. Bar-Shalom and E. Oron, "Precision tracking based on segmentation with optimal layering for imaging sensors," *IEEE Trans. on Pattern Anal. Mach. Intell.,* vol. 17, no. 2, pp. 182-188, Feb. 1995.
[6] J. S. Cho, D. J. Kim and D. J. Park, "Robust centroid target tracker based on new distance features in cluttered image sequences," *Visual Communications and Image Processing VCIP'2 000,* vol. 4067, pp. 1045- 1056, Perth, June 2000.
[7] J. S. Cho, D. J. Kim and D. J. Park, "Robust centroid target tracker based on novel distance features in cluttered image sequences," *IEICE Trans. Inf. and Syst.,* vol. E83-D no. 12, pp. 2142-2151, Dec. 2000.

조 재 수 (曺在洙)

1993년 2월 : 경북대학교 전자공학과(공학사)
1996년 2월: 한국과학기술원 전기및 전자공과(공학석사)
2001년 2월 : 한국과학기술원 전기 및 전자공학(공학박사)
2001년 3월~2003년 7월 : (주)삼성전자
2003년 9월~현재 : 한국기술교육대학교 인터넷미디어공학부 조교수
관심분야 : Automatic Video Tracking, Scalable Video Coding, Digital Watermarking