

논문 2004-41SP-6-12

# CCD 영상에서의 실시간 자동 표적 탐지 알고리즘

## (Real-Time Automatic Target Detection in CCD image)

유 정 재\*, 선 선 구\*\*, 박 현 옥\*\*\*

(Jung-Jae Yu, Sun-Gu Sun, and Hyun-Wook Park)

### 요 약

본 논문에서는 CCD(charge-coupled device) 영상 기반의 자동 표적 탐지 시스템(ATD System : Automatic Target Detection System)에 적합한 빠른 탐색 방법을 제안한다. 무기체계에서의 활용을 위해서는 빠른 연산이 주요한 변수인 만큼 이 논문에서는 적은 계산량으로 다양한 표적을 탐지할 수 있는 능력에 주안점을 두고 있다. 표적 훈련(train)단계에서는 구간별 수직 방향 프로젝션을 이용하여 1D의 템플릿을 구성하고 K-means clustering과 이진 트리 구조(binary tree structure)를 활용하여 실제 시험 단계에서 템플릿 정합하는 횟수를 최소화한다. 또한 Correlation-based Adaptive Predictive Search(CAPS)를 이용하여 각각의 템플릿에 적응적인 skip-width를 사용하여 탐색 속도를 높이고 클러터 제거 단계에서는 윤곽선으로부터 추출한 Fourier Descriptor계수를 비교함으로써 초기 탐지에서 타겟으로 오인된 클러터를 모양 정보에 기반해서 제거하는 방법을 사용한다.

### Abstract

In this paper, a new fast detection and clutter rejection method is proposed for CCD-image-based Automatic Target Detection System. For defence application, fast computation is a critical point, thus we concentrated on the ability to detect various targets with simple computation. In training stage, 1D template set is generated by regional vertical projection and K-means clustering, and binary tree structure is adopted to reduce the number of template matching in test stage. We also use adaptive skip-width by Correlation-based Adaptive Predictive Search(CAPS) to further improve the detecting speed. In clutter rejection stage, we obtain Fourier Descriptor coefficients from boundary information, which are useful to rejected clutters.

**Keywords :** Automatic target detection, template matching, CCD image, binary tree, Fourier Descriptor CAPS.

## I. 서 론

극한상황인 전투환경에서 얻어진 센서정보를 활용하여 무기체계 승무원들의 판단능력을 도와 효과적인 반응 수단을 제공하기 위한 목적으로 개발되는 시스템을 자동표적인식 (Automatic Target Recognition)이라 한다. 자동표적인식 시스템에서는 FLIR(Forward looking infrared), MMW(Milimeter wave), SAR (Synthetic aperture radar)와 CCD 센서 등이 독립적으로 혹은 정

보융합의 형태로 이용될 수 있다. 이중 CCD영상은 조명의 영향을 민감하게 받고 군차량의 위장으로 인해 특성이 상이하게 변화할 수 있으므로 기존의 ATR시스템에서 관심의 대상 밖이었다. 그보다는 야간에도 이용가능하고 조명, 위장의 영향을 적게 받는 적외선 영상을 기반으로 하여 표적을 인식하는 연구<sup>[1][2]</sup>가 많이 진행되어 왔다. 하지만 CCD영상에서 물체의 윤곽선이 FLIR 영상에서보다 훨씬 뚜렷하게 관찰된다는 점을 감안하면 CCD영상에 기반한 ATR시스템 역시 FLIR영상에 기반한 시스템을 보완하여 적용될 때 전체적인 성능향상에 기여할 수 있음을 알 수 있다. 이 논문에서는 기본적으로 CCD영상으로부터 얻어진 정보로부터 표적 탐지 과정이 독립적으로 이루어지고 이후 FLIR영상에 기반한 시스템의 결과와 융합되는 상황을 전제한다.

\* 학생회원, \*\*\* 정회원, 한국과학기술원  
(Korea Advanced Institute of Science and Technology)

\*\* 정회원, 국방과학연구소

(Agency for Defense Development)

접수일자: 2004년7월29일, 수정완료일: 2004년8월31일

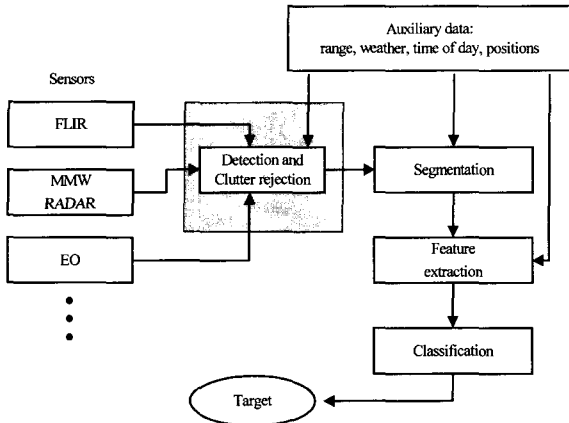


그림 1. 일반적인 ATR시스템의 구조  
 Fig. 1. Usual ATR system structure.

자동표적인식의 과정은 그림 1과 같이 다양한 센서로부터 획득된 영상에서 관심있는 표적이 있는지를 판별하는 표적 탐지와 탐지된 표적에서 정확히 배경과 표적을 분리하는 표적 분할, 표적의 불변특징 추출, 추출된 특징을 분류하여 표적을 인식/식별하는 일련의 단계로 구성된다. 이 논문에서는 CCD 흑백 영상을 기반으로 자동 표적 인식과정의 가장 초기 단계인 자동 표적 탐지(Automatic Target Detection)에 관한 빠르고 효과적인 알고리즘을 제안한다. 이후 이어지는 표적 분할, 특징 추출, 표적 인식/식별의 과정 모두 최초 표적 탐지 과정이 제대로 이루어진 것을 전제로 하기 때문에 효과적인 표적 탐지는 우수한 ATR 시스템의 구현을 위해 필수적인 요소이다. 또한 군사용으로 활용하기 위해서는 빠른 연산시간이 주요한 변수인 만큼 이 논문에서는 적은 계산량만으로 넓은 배경영역에서 표적을 탐지할 수 있는 능력에 주안점을 두고 있다.

윤곽선 정보에 기초해서 다양하게 변화하는 물체를 탐지하고자 하는 연구 분야에서는 유사한 template boundary를 입력영상의 gradient에 적응적으로 변화시키는 deformable template model<sup>[10]</sup>이 많이 사용되어 왔다. Tim Cootes는 얼굴 인식에 관한 연구에서 ASM(Active Shape Model)<sup>[11][12]</sup>을 이용하여 크기나 표정이 심하게 변화하는 환경에서 초기 모델 정보로부터 반복 연산을 거쳐 입력영상에서 얼굴의 윤곽선을 찾는 알고리즘을 제안하였다. 그러나 이러한 방법은 찾는 대상의 위치를 어느 정도 근사하게 알고 있어야 한다는 제한이 있으므로 이 논문에서 다루고 있는 상황처럼 초기에 넓은 배경에서 작은 표적의 위치를 초기 탐색하는 문제에는 적용하기 어렵다.

이 논문에서는 기본적으로 실전에서 관찰될 수 있는

표적 영상에 대한 사전 정보와 거리정보를 근사하게 알고 있다는 전제하에서 입력영상을 학습된 데이터와 빠르게 비교해서 표적의 위치를 결정한다. 문제는 물체의 수평 방향 회전과 미세한 크기 변화 등을 고려하면 이진화된 정보만 추출하더라도 상당히 많은 개수의 템플릿이 존재하게 된다는 점이다. 입력영상의 모든 위치에서 모든 템플릿들을 비교해야 한다면 이는 연산시간의 심각한 증가를 초래한다. 그렇기에 조금이라도 가능성이 있는 표적 후보는 누락시키지 않으면서 가급적 템플릿 정합 횟수를 최소로 줄여야 할 필요가 있다. 이를 위해 이 논문에서는 1D 템플릿의 활용과 이진 트리 구조를 이용한 템플릿의 선별적인 비교, 그리고 CAPS<sup>[4]</sup>를 이용한 다단계 탐색 방법을 제안한다.

먼저 II장에서는 실제 입력 영상이 들어왔을 때 표적을 검출하는 전체적인 알고리즘을 설명한다. 여기서는 전처리 과정과 초기 탐지 과정, 그리고 Fourier descriptor<sup>[5][6]</sup>를 이용한 클러스터 제거가 설명된다. 이 논문의 핵심이라 할 수 있는 III장에서는 초기 탐지를 위한 이진 트리 구조<sup>[3]</sup>의 학습 데이터를 구성하는 방법을 설명한다. 트리의 각각의 노드에는 그 안에 표적들을 빠짐없이 포함할 수 있는 탐색시의 오차 한도값, 그리고 CAPS<sup>[4]</sup>에 기초한 skip-width가 포함된다. 마지막으로 IV장에서는 실제 실험 결과 분석을 통해 성능을 검증한다.

## II. 제안 알고리즘

### 1. 제안 알고리즘

이 논문에서는 템플릿 정합 과정에서 연산시간을 줄이기 위해 표적 모델의 2D 템플릿 대신 1D 템플릿을 비교하여 표적의 위치를 찾는다. 먼저 입력 영상이 들어왔을 때 Yager의 Fuzzy measure<sup>[7]</sup>를 이용한 block thresholding을 적용해서 이진 영상을 만든다.

먼저 수식(1)과 같은 멤버쉽 함수값을 계산한다.  $\mu_0$ ,  $\mu_1$ 은 각각 임계값  $t$ 보다 작은 영역, 큰 영역의 평균값들이고  $C$ 는 intensity  $x$ 의 최대값과 최소값 간의 차이이다.  $\mu_x(x)$ 는  $x$ 로부터  $\mu_0$ ,  $\mu_1$ 까지의 거리의 역수에 상수를 첨가한 값이며 이는 intensity  $x$ 가 해당 클래스에 포함되는 정도를 나타내는 값으로 이해할 수 있다.

$$\mu_x(x) = \frac{1}{1+|x-\mu_0|/C} \quad \text{if } x \leq t$$

$$\text{or } \frac{1}{1+|x-\mu_1|/C} \quad \text{if } x > t \tag{1}$$

이러한 멤버쉽 함수값을 이용해서 fuzziness를 정의할 수 있고 그 fuzziness를 최소화 시키는 t를 찾아서 이진화 임계값으로 사용한다. 여기서는 Yager가 제안한 방법을 이용해서 식(2)의  $D_p(t)$ 를 최대화 하는 t를 찾고 그 값으로 이진화를 수행해서 이진 영상을 만든다. 이때 g는 입력 영상에서 각각의 픽셀의 intensity(0~255)를 의미하고  $\mu_x(g)$ 는 식(1)에서 구한 멤버쉽 함수값이다. p는 distance measure에 따라 여러 가지 값으로 정할 수 있으며 여기서는 p=2를 대입하였다.

$$D_p(t) = \left[ \sum_g |\mu_x(g) - \mu_x(g)|^p \right]^{1/p} \quad \text{when } \mu_x(g) = 1 - \mu_x(g) \quad (2)$$

이때 영상 전체를 대상으로 이진화를 수행하지 않고 블록으로 나누어 이진화를 수행한다. 블록의 크기를 작게 할수록 배경의 영향을 덜 받기 때문에 보다 정확한 이진화 영상을 얻을 수 있지만 임계값을 계산하는 횟수가 늘어나므로 연산시간이 증가한다. 또한 표적이 블록간의 경계에 걸쳐 있는 경우까지 고려해서 블록을 템플릿의 폭만큼 겹쳐서 이동시켰다. 블록의 크기는 템플릿의 세배 크기로 잡았으며, 이는 성능과 시간을 고려할 때 적절한 값을 실험적으로 확인하였다. 이진화 후 모포로지 처리를 통해 노이즈와 작은 요소들을 제거한다. 그리고 그림 2와 같이 분할된 영역에서 수직 방향으로 프로젝션 시킨 데이터를 얻는다.

$M \times N$ (M이 세로방향, N이 가로방향 길이)입력 영상의 경우, 가장 윗부분의  $m \times N$  ( $m < M$ )영역을 수직방향으로 프로젝션 시킨 데이터가 그림 2에서 화살표 표시한 회색 박스안의 그래프이다. (그림 3에서 상단 박스와 동일하다.) 이때 프로젝션 시키는 구간의 세로 방향 폭인 m은 다음과 같이 결정한다. 표적과의 거리 정보를 근사하게 알고 있다는 가정하에서 관찰 가능한 모든



그림 2. 입력 영상의 전처리 과정  
Fig. 2. Preprocessing of Input image.

표적을 포함할 수 있는 충분한 크기의 템플릿 크기를 결정하며 이 논문의 경우  $39 \times 74$ 로 설정하였다. 이러한 템플릿의 세로방향 길이 즉, 39가 그림 2에서 프로젝션 구간의 폭인 m이 된다. 이렇게 결정한 프로젝션 구간을 아랫방향으로 이동시키면서 각각의 프로젝션마다 위 아래로 들어오고 나가는 N개의 픽셀들만 고려함으로써  $2(M-m)N$ 의 연산만으로 프로젝션 데이터를 구성한다.

프로젝션을 시키지 않고 2D 템플릿 정합을 하는 경우와 연산시간을 비교하면 다음과 같다.  $m \times n$ 크기의 2D 템플릿 정합을 하는 경우  $(M-m)(N-n)mn$ 의 XOR연산과 +연산이 필요하다. 반면에 1D 템플릿 정합을 수행할 경우,  $(M-m)(N-n)n$ 의 +-연산이 필요하므로 연산시간 면에서 1D 템플릿 정합이 m의 비율만큼 연산량이 차이가 난다.

하지만 2D 템플릿을 1D템플릿으로 변환하는 과정에서 어쩔 수 없이 정보량의 감소가 수반되므로 하나의 템플릿으로 구분할 수 있는 대상의 정확도 역시 떨어진다. 하지만 이 논문의 주된 목적은 들어온 입력영상으로부터 관심영역을 빠르게 탐지하는 것이므로 그러한 손해를 감수하면서 속도를 높이기 위해 1D 템플릿을 사용하였다. 그리고 1D 템플릿 정합에서 일단 표적일 가능성이 있는 물체들을 빠트리지 않고 검출한 한다면 뒷단의 클러스터 제거 단계에서 클러스터들을 제거함으로써 1D 템플릿의 오차를 해결할 수 있다.

1D 템플릿을 구성하면서 발생하는 또 한 가지 문제는 정보량이 감소하는 과정에서 템플릿의 상당수가 유사한 특성을 갖게 된다는 점이다. 그에 따른 불필요한 연산을 줄이기 위해 III절에서 설명할 클러스터링 과정으로 유사한 템플릿들을 합쳐주고 적절한 오차 한도값과 skip-width를 결정해서 정합하는 횟수를 최소화한다.

## 2. 초기 표적의 후보 영역 탐색

앞 단계에서 프로젝션시킨 데이터를 그림 3과 같이 학습영상의 1D 템플릿과 비교하여 유사한 위치를 찾는 방식으로 최초 표적 후보를 결정한다. 유사도를 계산하는 방법은 상관도 계수(correlation coefficient)를 계산

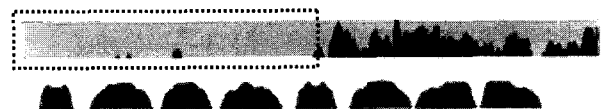


그림 3. 프로젝션 데이터 비교  
Fig. 3. Comparison of projection data.

할 수도 있겠지만 여기서는 연산시간을 줄이기 위해 단순히 차이를 더하는 방식을 취하였다. 식(3)에서  $x[i]$ 는 입력 영상의 프로젝션 데이터이고  $m[i]$ 는 1D 템플릿이다.

$$Difference = \sum_{i=1}^m |m[i] - x[i]| \quad (3)$$

그림 3은 입력 영상의 프로젝션 데이터와 예로 제시된 8개의 학습 데이터의 템플릿들을 보여준다.

입력영상의 프로젝션 데이터에서 점선 박스로 표시한 영역처럼 0이나 아주 미세한 성분만 이어지는 구간에서는 굳이 템플릿 정합을 수행하지 않아도 표적인 될 수 없음을 알 수 있다. 이와 같은 1D 템플릿 정합의 이점을 이용해서 어느 정도 가능성이 있는 위치에서만 실제 템플릿 정합을 수행함으로써 정합 횟수를 큰 폭으로 감소시킬 수 있다.

그리고 학습 데이터와의 연산횟수를 줄이기 위해 이진 트리 구조의 학습 데이터를 구성한다. 즉, 입력 영상의 프로젝션 데이터가 트리의 상단 노드의 템플릿과 일정한 값 이상의 상관도를 갖는 경우에만 왼쪽, 오른쪽 두개의 자식 노드 중 상관도가 높은 노드를 선택해서 비교를 계속한다. 실제로는 초기 탐색을 수행하는 대부분의 영역에서는 트리의 최상단 노드와만 비교를 수행하게 되므로 템플릿 정합 횟수는 큰 폭으로 줄게 된다.

이 때 왼쪽, 오른쪽 자식 노드의 템플릿과의 차이(식(3))만을 계산해서 진행방향을 결정한다면 Baysean rule에 어긋나는 잘못된 결과를 얻게 될 위험이 있다. 왜냐하면 왼쪽과 오른쪽의 자식 노드들 역시 더 많은 데이터들의 클러스터링 결과로 얻은 것들이며 각각의 노드에 포함된 데이터들의 개수가 다르기 때문이다. (III장 참조) 좀더 정확한 비교를 위해서는 더 높은 사후 확률을 갖는 노드를 선택해야 하며 그러기 위해서는 확률 밀도함수 예측 과정이 필요하다. 정확한 확률 밀도함수 예측을 위해서는 프로젝션 데이터와 동일한 차원에서 K-nearest neighbor와 같은 방법을 사용해야 하지만 여기서는 간략하게 템플릿간의 차이(식(3))를 변수  $x$ 로 갖는 1D 가우시안 분포로 근사화한다. 이때 노드에 포함되는 템플릿의 개수가 너무 적은 경우에는 가우시안 분포로 보는 것이 어렵기 때문에 식(4)와 같은 모델을 사용한다. 그림 4에서 보듯이  $P1(x)$ 는 노드에 포함되는 각각의 원소를 중심으로 국부적인 가우시안 분포들의 중첩이고  $P2(x)$ 는 전체 원소들을 하나의 분포로 보고 형성한 가우시안 모델이다. 계수  $\alpha$ 를 그림과

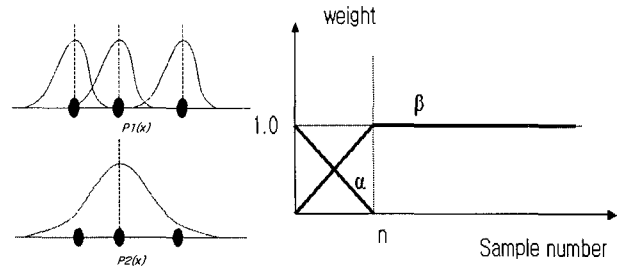


그림 4. 노드의 확률 밀도함수 근사화  
Fig. 4. Approximation of probability density function of each node.

같이 조절함으로써 템플릿 개수가 늘어날수록  $P1(x)$ 의 비중을 줄이고  $P2(x)$ 의 비중을 증가시킨다. 이런 방식으로 왼쪽, 오른쪽 노드에 포함된 데이터들에 대해서 확률 밀도함수를 근사화 하고 주어진 입력에서 더 높은 확률값을 갖는 노드를 선택한다.

$$P(x) = \alpha * P_1(x) + \beta * P_2(x) \quad (4)$$

이때 각각의 노드는 CAPS<sup>[4]</sup>에 기초해서 템플릿 특성에 적응적인 skip-width를 갖고 있다. 이 간격으로 듬성듬성하게 비교를 하고 상관도가 높은 위치에서만 부근 영역에서 자세한 탐색을 수행한다. 이러한 방법으로 연산횟수를 더욱 줄일 수 있었으며 학습 데이터를 구성하는 방법은 III장에서 다룬다.

### 3. 클러스터 제거

앞 단계에서 찾은 표적의 후보 영역들 주변으로 블록 이진화<sup>[7]</sup>를 다시 한번 수행해서 좀더 정확한 이진 영상을 얻고 무게중심을 구하는 방식으로 물체의 중심점을 계산한다.

Kauppinen의 실험<sup>[8]</sup>에 의하면 물체의 방향이 변화하고 노이즈가 존재하는 상황에서 Fourier descriptor를 이용해서 물체를 식별하는 여러 가지 방법 중 Contour Fourier 방법이 가장 우수한 성능을 보인다고 한다. 그러므로 본 연구에서도 식(5)처럼 무게중심에 상대적인  $x, y$  좌표를 각각 실수, 허수부분으로 보는 Complex coordinate function<sup>[8]</sup>을 사용해서 윤곽선을 표현한다.  $(x_i, y_i)$ 는 윤곽선 픽셀의 좌표이며  $(x_c, y_c)$ 는 무게중심의 좌표이다.

$$z_i = (x_i - x_c) + j(y_i - y_c) \quad (5)$$

이러한  $z(i)$ 를 Fourier transform(식(6))하여 Fourier descriptor (FD)<sup>[5][6]</sup>특징벡터(식(7))를 구성한다. 이때

F(0), F(1)은 각각 대상의 위치와 크기를 포함하는 정보이며 F(1)의 크기로 나누어 주는 것은 크기 변화에 둔감한 특징벡터를 얻기 위함이다.

$$F(n) = \frac{1}{N} \sum_{k=0}^{N-1} \alpha(k) e^{-j2\pi nk/N} \quad (6)$$

$$x = \left[ \begin{array}{c} |F(2)| \\ |F(1)| \end{array}, \frac{|F(3)|}{|F(1)|}, \dots, \frac{|F(N/2)|}{|F(1)|}, \dots, \frac{|F(N-1)|}{|F(1)|} \right]^T \quad (7)$$

이런 방식으로 앞 단에서 구한 표적 후보의 FD벡터를 구해서 학습 데이터의 FD 벡터들과 비교하여 차이를 계산한다. 차이가 오차 한도값 이하가 되는 학습 영상이 하나라도 존재하면 표적으로 간주하고 그렇지 않으면 클러스터로 간주한다. 식(8)에서  $E_k$ 는 k번째 학습 데이터와의 차이를 정규화한 값이며  $FD^{test}(i)$ 는 입력영상에서 물체의 i번째 FD 계수(식(7))를,  $FD^{train}_k(i)$ 는 k번째 학습 데이터의 i번째 FD계수를 나타낸다.

$$E_k = \frac{\sum_i (FD^{test}(i) - FD^{train}_k(i))^2}{\sum_i (FD^{test}(i)^2 + FD^{train}_k(i)^2) / 2} \quad (8)$$

### III. 학습 데이터 구성

템플릿 정합을 이용한 표적 탐지에서 학습 데이터를 어떻게 구성하느냐는 시스템 전체의 성능을 좌우할 만큼 중요한 변수이다. 제안한 방법에서 학습 데이터는 크게 이진 트리 구조로 구성된 프로젝션 데이터와 표적의 윤곽선으로부터 뽑아낸 FD 벡터들로 구성된다. 이 중 FD 부분은 앞장의 3절에서 설명한 방법대로 학습된 표적 영상에 대해 구한 것을 저장해서 비교할 뿐이므로 이 장에서는 추가 설명을 생략한다.

#### 1. K-means clustering

최초 학습 단계에서  $m \times n$  크기의 표적 영상은 표적을 수평방향으로 일정 각도씩 회전시키면서 촬영한 영상이다. 이러한 영상들로부터 다시 B-spline function<sup>[9]</sup> 보간법을 사용해서  $(m+1) \times (n+1), \dots, (m+a) \times (n+a)$ 의 약간씩 크기 변화가 있는 여러 영상들을 얻어낸다. 다음은 II장 1절에서 설명한 전처리 과정과 동일하게 이진 영상을 만들고 모포로지 처리를 해서 작은 요소들을 제거한다. 그리고 학습 데이터의 크기를 통일시키기 위해서 우선 다양한 크기의 영상을  $(m+a) \times (n+a)$  템플릿의 중

양에 위치시킨다. 그리고 남은 공간에 zero mapping을 한 뒤 수직 방향으로 프로젝션 시켜  $(m+a) \times 1$ 의 프로젝션 데이터들을 얻는다. 최초 표적의 종류를 C, 수평방향에 대하여 d도씩 회전시켜 촬영했다면 최종적인 프로젝션 데이터의 개수  $C'$ 은 다음과 같다.

$$C' = C * \frac{360}{d} * (a+1) \quad (9)$$

우리의 목적은 입력 영상에서 표적 위치의 초기 탐지이므로 표적의 종류는 고려할 필요가 없다. 그러므로 앞서 구한 표적 영상의 프로젝션 데이터들을 K-means clustering<sup>[13]</sup>을 이용해서 유사한 데이터들끼리 통합함으로써 그 숫자를 줄여준다. K-means clustering의 구체적인 방법은 다음과 같다.

- step 1.  $C'$ 개의 데이터들 중  $C'/K$  간격으로 하나씩 샘플링 해서 K개의 초기 평균 데이터들을 설정한다. (이때 임의로 선택해도 결과는 같다.)
- step 2.  $C'$ 개의 전체 데이터들을 K개의 중심 데이터들과 비교해서 K개의 클래스를 분류한다.
- step 3. 각각의 클래스 안에서 다시 평균 데이터를 계산한다.
- step 4. 새로 계산한 평균 데이터들과 기존의 평균 데이터들을 비교해서 차이가 모두 일정 값 이하이면 종료하고 그렇지 않으면 step 2로 돌아간다.

이렇게 해서 클러스터링 과정이 종료되면 평균 데이터와 클래스 안에 포함되는 데이터들과의 차이를 계산해서 가장 큰 차이값을 클래스의 에러값 e로 저장한다. 식(10)에서  $m$ 은 평균 데이터를,  $x_i$ 는 해당 클래스 안의 i번째 데이터를 나타낸다.

$$e = \max_i \left\{ \left| \bar{m} - x_i \right| \right\} \quad (10)$$

평균 데이터를 대표 템플릿으로 삼아 해당 클래스 안의 다른 데이터들을 대신해서 표적 탐지에 이용하고자 한다면 그 클래스가 내재하고 있는 에러 e를 고려해서 표적으로 인정하는 오차 한도값을 결정해야 한다. 즉 입력 영상의 프로젝션 데이터와 대표 템플릿과의 차이가 e의 범위 안에 있다면 일단 표적 후보로 지정한다.

#### 2. Correlation-based Adaptive Predictive Search

Correlation-based Adaptive Predictive Search

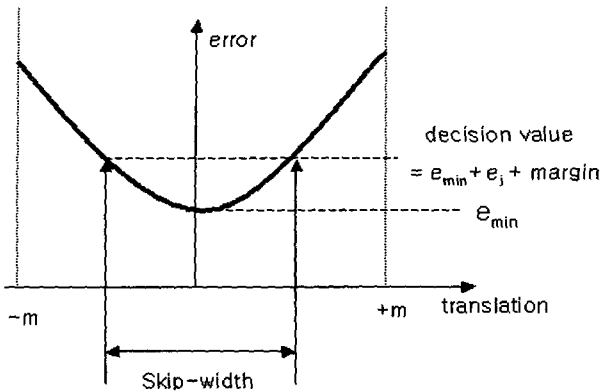


그림 5. skip-width 결정  
Fig. 5. Decision of skip-width.

(CAPS)<sup>[4]</sup>는 데이터의 자기 상관도(Auto-correlation)정보에 적응적인 skip-width를 정하고 그 간격으로 띄엄 띄엄 탐색한 후 해당 기준값 이상의 상관도를 보이는 곳에서만 자세한 탐색을 수행하는 방법이다.

skip-width를 결정하기 위해서는 상관도 함수를 알고 있는 상태에서 표적으로 인정하는 오차 한도값을 먼저 결정해야 한다. 기존의 연구<sup>[4]</sup>에서는 탐색과정에서 단일한 템플릿을 전체하고 임계값을 필요에 따라 인위적으로 결정한 뒤에 그에 따른 skip-width를 구하는 방법을 사용하였다. 여기서는 앞 절의 K-means clustering의 결과로 얻은 평균 데이터를 템플릿으로 삼아 탐색을 수행하기 때문에 클래스 안에 내재되어 있는 오차를 고려한 skip-width를 구해야 된다. 그림 5에서 포물선은 j번째 클래스를 대표하는 템플릿의 오차 그래프이며 이는 자기 상관도의 반대개념으로 볼 수 있다.  $e_j$ 는 식(10)에 따라 구한 최대 오차이며 margin은 노이즈와 영상의 부정확한 이진화로 인한 오차이다. 실제 트리 안에서 skip-width를 결정하는 것은 이보다 복잡하며 다음 절에서 설명한다.

### 3. 이진 트리 구성

1절에서 설명한 클러스터링의 결과로 얻은 C'개의 데이터 세트를 가지고 이진트리를 구성한다. 트리를 구성하는 방법은 전체 데이터를 상향식으로 묶어가는 방법과 하향식으로 나누어 가는 두 가지 방법이 있다. 상향식 방법을 사용해서 가장 근접한 두개씩 묶어 나가면서 트리를 구성한다면  $C'=2P$ 인 경우 P+1개의 레벨을 갖는 이상적으로 낮은 높이의 트리를 얻을 수 있다. 그러나 그 경우 각각의 노드에 포함된 데이터들의 분포를 예측해서 Maharanobis distance를 고려한 클러스터링을 수행해야 하기 때문에 시스템의 복잡도가 높아지게

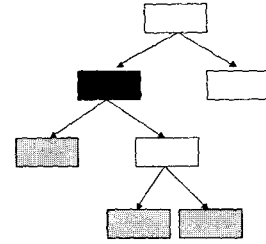


그림 6. 간단한 이진 트리 구조의 예  
Fig. 6. Simple example of binary tree structure.

된다. 그러므로 우리는 트리구조가 non-symmetric하게 된다는 단점을 감수하고 하향식 방식을 이용해서 트리를 구성하였다.

각각의 레벨에서 클래스를 분류하는 방법은 1절에서 설명했던 K-means clustering을 사용한다.

트리의 각각의 노드의 오차 한도값과 skip-width를 구하는 방식은 아래와 같은 간단한 예로 설명될 수 있다. 그림 6은 전체 데이터가 4개인 경우의 이진 트리를 구성한 것이다. 각각의 바닥 노드의 템플릿 역시 1절의 클러스터링을 거쳐 얻은 평균 데이터이며 식(10)에서의 오차를 내재하고 있다.

실제 탐색 단계에서 검은색 노드의 템플릿으로 탐색한 결과는 회색 노드의 템플릿들과 유사한 물체들 모두를 포함해야 한다. 그러므로 검은색 노드의 오차 한도값과 skip-width 모두 회색 노드들을 고려한 충분한 여유간격을 갖고 있어야 한다. 먼저 오차 한도값  $D_{criterion}$ 는 식(11)와 같이 구한다. 여기서 P와  $B_j$ 는 각각 검은 노드와 회색 노드들의 템플릿을 벡터로 표시한 것이며  $D(\cdot)$ 는 두 템플릿을 수평방향으로 translation시키면서 구한 최소 차이값(식(3))이고  $e_j$ 는 j번째 회색 노드에 내재된 오차값(식(10))이다.

$$D_{criterion} = \max \{ D(\bar{P}, \bar{B}_j) + e_j \} \quad (11)$$

skip-width를 구하는 과정은 다음과 같다. 먼저 그림 7과 같이 P와 각각의  $B_j$ 들과의 오차 그래프들을 그린다. 이들은 그림 7에서 세 개의 포물선에 해당하며 Cross correlation의 역할을 대신한다. 그리고 오차 한도값  $D_{criterion}$ 로 잘라 주었을 때 얻는 폭의 최소 길이(검은 양방향 화살표)를 그림 6의 검은 노드의 skip-width로 최종 결정한다.

이런 식으로 트리를 구성하면 어떤 노드에 포함되는 템플릿의 개수가 많으면 많을수록  $D_{criterion}$ 이 커지고 skip-width 역시 길어질 가능성이 높아진다. 결과적으로

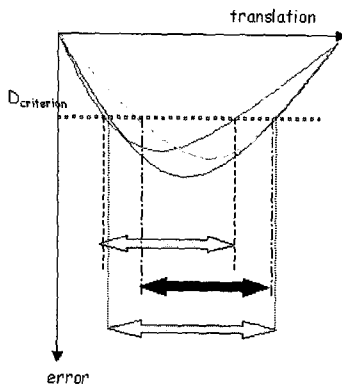


그림 7. skip-width의 결정  
Fig. 7. Decision of skip-width.

로 트리의 상단노드에서는 띄엄띄엄 템플릿 정합을 수행하면서 입력 영상의 데이터가 템플릿과 조금이라도 비슷하면 일단 표적후보로 인정하고 자식노드에게 전달한다. 반대로 하부 노드에서는 좁은 간격으로 샘플링하면서 엄격한 기준으로 비교하게 된다.

#### IV. 실험 결과 및 결론

##### 1. 실험결과

학습영상과 시험 영상 모두 Basler A301f CCD 카메라를 삼성 SENS노트북의 1394 포트에 연결하여 촬영하였다. 학습 영상은 1/25 크기의 프라모델 탱크 3대를 수평 방향으로 5도 간격씩 회전시키면서(식(9)에서  $d=5$ ) 70x35 사이즈의 영상을 총  $3 \times 72 = 216$ 장 촬영하였다. 전차라는 표적의 특성상 수직방향 회전은 실제 시험단계에서 크게 나타날 수 없으므로 여기서는 고려하지 않는다. 미세한 크기변화를 고려하기 위해 B-spline function<sup>[9]</sup>을 이용해서 72\*37, 74\*39 크기의 영상을 추가로 생성해서 총 648장의 학습 영상을 만들었다. 이진화와 프로젝션 과정을 거쳐 74\*1의 1D 데이터 648개를 만들고 이들을 다시 K-means clustering을 거쳐 64개의 클래스로 분류하고 각각의 평균 데이터를 생성하였다. 이 64개의 데이터들을 ID 템플릿으로 삼아 III장에서 설명한 이진 트리를 생성하였고 FD 벡터는 크기변화를 고려할 필요 없으므로 초기 216장의 학습 영상에 대해서만 생성하였다.

시험 영상은 학습 과정에서 사용했던 프라모델을 실제 야전 상황과 유사한 자연 환경에서 어느 정도 일관된 크기를 유지하되, 무작위 방향으로 배치하며 주간에 촬영하였다. 초기 탐색 단계에서 표적 후보로 선정된 물체는 박스 처리되었고 클러터 제거 단계에서 클러터로 판

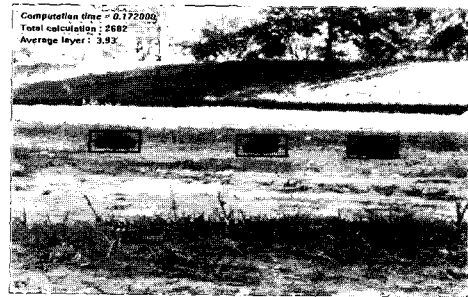


그림 8. 결과 영상  
Fig. 8. Result image.

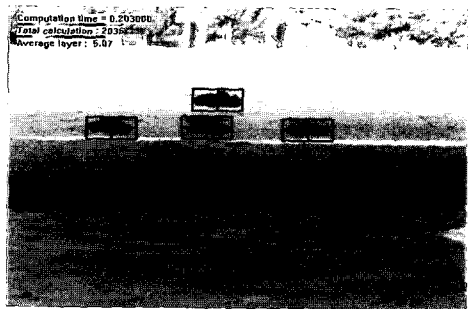


그림 9. 인접한 표적의 결과 영상  
Fig. 9. Result image of close targets.

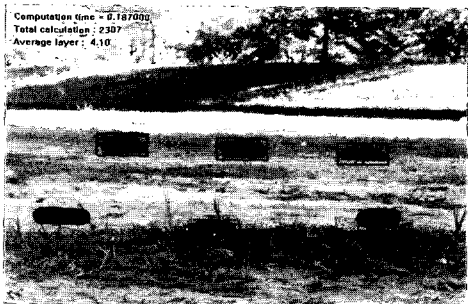


그림 10. 인위적인 클러터가 섞인 결과 영상  
Fig. 10. Result image with artificial clutters.

명된 물체는 밝은 포인트 처리, 표적으로 최종 선언된 물체는 십자 표시가 되었다. 그림 8과 그림 9는 각각 음지의 표적과 양지의 표적을 탐지한 결과이다. 그림 9의 경우 위 아래로 인접한 두개의 표적을 각각 탐지한 것을 알 수 있다. 그러나 초기 탐색이 수직방향 프로젝션 데이터에 의존하므로 표적 자체가 세로 방향으로 겹치는 경우에는 표적이 초기 탐색에서 검출되지 않는 한계가 있다. 그림 10은 인위적으로 첨가한 표적과 유사한 크기의 클러터를 구분해서 탐지한 결과이다. 그림 10에서 중앙부의 실제 표적들은 제대로 탐지되고 하단부의 클러터 세 개 중 하나가 초기 탐색 과정에서 표적 후보로 오인되었다가 클러터 제거 단계에서 클러터로 분류된 것을 보여준다.

실험은 45장의 영상에 총 90개의 표적과 45개의 인위적인 클러터를 포함시켜 실험하였다. Detection Rate

(DR)와 False Alarm Rate(FAR)는 식(12)과 같이 정의한다. Total number of clutter는 인위적으로 삽입한 클러터의 개수이며 False detected clutter는 삽입한 클러터와 배경성분으로부터 오인된 클러터의 총 합이다.

$$DR = \frac{\text{Correctly detected targets}}{\text{Total number of targets}}$$

$$FAR = \frac{\text{False detected clutters}}{\text{Total number of clutters}} \quad (12)$$

일단 초기 탐색에서 False reject가 발생할 가능성을 줄이기 위해 식(11)의 프로젝션 데이터의 오차 한도값에 충분한 마진을 두어 가급적 많은 표적 후보를 검출하도록 했다. 그리고 클러터 제거 단계에서 식(8)의 Ek에 대한 criterion value를 그림 11과 같이 변화시키면서 DR과 FAR을 기록하였다.

그림 11을 보면 criterion value가 0.15가 되는 곳에서 FAR은 절반 이하면서 DR은 90%를 기록한다. 여기서 FAR은 인위적으로 첨가한 표적과 비슷한 크기의 클러터를 고려했을 때의 수치임을 상기하기 바란다. FAR을 클러터를 첨가하지 않았을 때의 '자연 배경으로부터의 클러터 개수/전체 표적 개수'로 정의한다면 criterion value=0.15일 때 FAR은 4.4%에 불과하다. 640×480사이즈의 입력영상 한 장을 처리하는데 소요된 시간은 Pentium 2.8 GHz CPU 환경에서 평균 0.18초에 불과하다. 이 시간은 학습 데이터를 읽어오는 시간과 전처리 과정 등을 포함하는 시간이며 입력영상을 읽고 결과를 도출하는 시간은 포함하지 않는다.

이진 트리를 사용하지 않을 때와 시간 비교를 하면 다음과 같다. II절 2장에서 설명했듯이 이진 트리를 이용해서 탐지를 수행할 때는 매 위치에서 항상 트리의 bottom layer까지 내려가지는 않는다. 중간 단계에서도 노드의 템플릿과의 에러가 기준값 이상이면 클러터로

간주하고 비교를 중단할 수 있다. 그림 8, 9, 10에서 보다시피 비교를 수행하는 layer의 깊이는 최상단 노드를 0으로 볼 때 평균 4.3이다. 그러므로 매 위치에서 자식 노드의 좌, 우를 선택하는 확률 밀도함수 비교를 평균 4.3회 수행하고 식(3)의 에러 계산은 평균 5.3\*S회 수행한다.(S는 부모노드와 자식노드간의 skip-width 차이로 인해 증가하는 정합 횟수이며 보통 3 이하이다.) 이진 트리를 사용하지 않고 64개의 템플릿을 하나씩 비교하는 방식을 사용할 경우 DR과 FAR은 비슷하면서 연산시간은 평균 0.82초로 4배 이상 소요되는 것을 확인하였다.

### 2. 비교실험

비교실험으로 Quoc가 제안한 모포로지 연산과 Majority filter를 이용한 Sequential filter<sup>[14]</sup>의 방법을 CCD 영상에 적합하게 수정해서 실험하였다. 공평한 비교를 위해서 Quoc가 제안한 방법을 앞 단의 초기 탐색에서 프로젝션 데이터의 비교 대신 사용하고 FD를 이용한 클러터 제거는 동일하게 추가해서 성능을 비교하였다. Criterion value=0.15으로 설정 했을 때 DR=90.0%, FAR=46.6%로 제안된 방법과 큰 차이 없지만 연산시간은 평균 0.25초로 제안된 방법보다 39%정도 많이 소요되는 것을 알 수 있다. 이는 모포로지를 이용한 초기 탐색에서 많은 수의 표적 후보들이 검출되고 그만큼 클러터 제거 단계에서 많은 연산을 수행해야 하기 때문이다. 그림 12는 제안한 방법을 적용했던 그림 9의 영상과 동일한 영상에 대해서 비교실험의 방법을 적용한 경우이다. 다수의 표적 후보들이 초기에 검출되었다가 클러터 제거 단계에서 제거되어 십자 표시 없이 박스만으로 표시된 것을 알 수 있다.

클러터 제거 단계를 생략하고 모포로지 연산과 Majority filter만 사용하더라도 입출력을 제외하고 평균 0.21초가 소요되어 제안한 알고리즘보다 시간이 많이 걸리는 것을 확인하였다. 그리고 이 경우에는 크기 정보에만

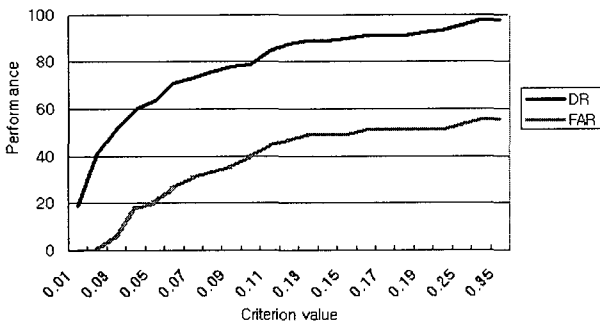


그림 11. Detection Rate와 False Alarm Rate  
Fig. 11. Detection Rate and False Alarm Rate.

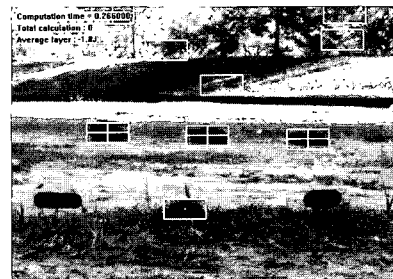


그림 12. 비교실험의 결과영상  
Fig. 12. Result image of comparison experiment.



의존해서 표적 후보를 찾으므로 FAR이 세 배 이상 크게 증가하므로 사실상 비교가 의미 없다.

그러므로 제안된 방법은 초기 탐색에서 기존 방법보다 더 빠르면서 표적일 가능성 있는 대상의 범주를 보다 좁게 찾을 수 있는 것이다.

### III. 결 론

시스템에 관해 전체적인 요약은 하자면 다음과 같다. 우선 학습 단계에서는 K-means algorithm을 통해 데이터의 숫자를 줄이면서 각각의 클래스에 내재된 오차 범위에 기반해서 이진 트리 구성단계에서 skip-width와 오차 한도값을 결정한다. 1D 프로젝션 데이터를 구성하면서 2D 정보의 일부는 사라지지만 그 대신 시험 단계에서의 정합과정이 훨씬 간단해지고 표적일 가능성이 현저히 낮은 영역을 사전에 제거하는 '영역 선택'이 가능해진다. 그 결과 빠른 시간 안에 입력 영상으로부터 표적후보를 검출하게 되고 이어서 FD을 이용한 클러스터 제거 단계에서는 모양 정보에 기반해서 표적으로 오인된 클러스터들을 제거함으로써 이후 이어지는 인식단계에서의 부담을 줄여준다. 이러한 모든 과정은 실제 군용 시스템에 활용될 수 있을 만큼 충분히 빠른 속도로 수행된다.

템플릿 정합의 가장 큰 약점은 역시 각각의 데이터를 모두 비교해야 하므로 많은 시간이 소요된다는 점이다. 이 논문에서는 K-means algorithm과 이진트리 구조, CAPS와 프로젝션 데이터의 사전 영역선택을 이용하여 표적 탐지에 필요한 정합횟수를 최소한으로 줄였다. 그리고 템플릿 정합을 이용하기 위해서는 먼저 관찰 가능한 모든 표적에 대한 학습 영상이 필요하며 이것은 실제로는 매우 어려운 과정이다. 그러므로 입력영상에서의 표적이 보유한 템플릿과 정확히 일치하지 않는 오차의 범위를 수치적으로 예측하여 실제 탐색 과정에서 정합을 수행하는 간격인 skip-width와 표적과 클러스터를 구분하는 오차 임계값을 적절하게 결정해야 한다. 템플릿 정합의 또 다른 약점은 크기변화에 민감하다는 점이며 이 점은 향후 계속적인 연구가 필요하다.

PCA나 MLP등을 이용한 표적 인식은 지금까지 많이 연구되어 왔다. 그러나 그러한 과정에 선행하는 초기 탐지에 관해서는 비교적 많은 연구가 이루어지지 못했다. 특히 그나마 발표된 연구도 대부분이 FLIR 영상에 기반한 탐지 알고리즘일 뿐, CCD 영상을 이용한 표적 탐지는 아직까지 연구가 거의 전무했다고 볼 수 있다.

이 논문에서 제안한 방법은 CCD 영상에 기반해서 초기 입력 영상으로부터 표적의 위치를 빠른 시간 안에 우수한 성능으로 탐지할 수 있다. 이는 실제 군용 시스템에 적용되어 FLIR영상에 기반한 ATR시스템을 보완할 경우 FLIR영상이 취약한 상황에서의 전체적인 시스템의 성능향상에 크게 기여할 수 있다.

### 참 고 문 헌

- [1] James A. Ratches, C.P. Walters, Rudolf G. Buser and B.D. Guenther, "Aided Automatic Target Recognition Based Upon Sensory Inputs From Image Forming Systems," IEEE Trans. on Pattern Analysis and Machine Intelligence., Vol. 19, no. 9, pp. 1004-1019, September 1997.
- [2] Syed A. Rizvi, Tarek N. Saadawi and Nasser M. Nasrabadi, "A Modular Clutter Rejection Technique for FLIR Imagery Using Region-Based Principal Component Analysis," IEEE Image Processing, Vol. 2, pp. 475-478, September 2000.
- [3] E. Ettelt and G. Schmidt, "Optimized Template Trees for Appearance Based Object Recognition," IEEE, Systems, Man and Cybernetics, 1998, Vol. 5, pp. 4536-4541, October 1998.
- [4] S. Sun, H. W. Park, David R. Haynor and Y. Kim, "Fast Template Matching using correlation-based adaptive predictive search," International Journal of Imaging Systems and Technology, vol. 13, Issue 3, pp.169-178, 2003.
- [5] Iivari Kunttu, Leena Lepisto, Juhani Rauhamma, and Ari Visa. "Multiscale Fourier Descriptor for Shape Classification," in Proc. of the 12th International Conf. on Image Analysis and Processing, pp. 536-541, September 2003.
- [6] F. Mokhtarian and A.K. Mackworth, "A Theory of Multiscale, Curvature-Based Shape Representation of Planar Curves," IEEE Trans. on Pattern Analysis and Machine Intelligence., Vol. 14, no. 8, pp. 789-805, August 1992.
- [7] J.R. Rarker, "Algorithms for Image Processing and Computer Vision", Wiley Computer Publishing, pp. 124-126, 1997.
- [8] Hannu Kauppinen, Tapio Seppanen and Matti Pietikainen, "An Experimental Comparison of Autoregressive and Fourier-Based Descriptors in 2D Shape Classification," IEEE Trans. on Pattern Analysis and Machine Intelligence., Vol 17, no 2, February 1995.
- [9] Anil K. Jain, "Fundamentals of Digital Image

Processing," Prentice Hall, pp. 364-366, 2003.

[10] Anil K. Jain, Yu Zhong and Sridhar Lakshmanan, "Object Matching Using Deformable Templates." IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 18, no. 3, pp. 267-278, March 1996.

[11] T. F. Cootes and C.J. Taylor, "Using Gray-Level Models to Improve Active Shape Model Search," Proc. of the 12th IARP Conference of Computer Vision & Image Processing, Vol 1, pp. 63-67, October 1994.

[12] T.F. Cootes, D. Cooper, C.J. Taylor and J. Graham, "Active Shape Models - Their Training and Application." Computer Vision and Image Understanding. Vol. 61, No. 1, pp. 38-59, and January 1995.

[13] Richard O.Duda, Peter E. Hart and David G. Stork, "Pattern Classification," Wiley Interscience, pp. 526-528, 2001.

[14] Quoc Henry Pham, Timothy M.Brosnan and Mark J.T.Smith, "Sequential Digital Filters for Fast Detection of Targets in FLIR Image Data," SPIE 1997.

저 자 소 개



유 정 재(학생회원)  
 2003년 연세대학교 전기전자공학  
 전공 학사 졸업.  
 2004년 현재 한국과학기술원  
 전기 및 전자공학전공  
 석사과정 재학 중

<주관심분야: 영상처리, 패턴인식, 자동표적인식>



신 선 구(정회원)  
 1987년 한양대학교 전자공학과  
 학사 졸업.  
 1989년 한양대학교 전자공학과  
 석사 졸업.  
 2003년 한국과학기술원  
 전기 및 전자공학전공  
 박사 졸업.

1989년~현재 국방과학연구소 연구원으로 재직  
 <주관심분야: 영상처리, 패턴인식, 자동표적인식>



박 현 옥(정회원)  
 1981년 서울대학교  
 전기공학과 학사졸업  
 1983년 한국과학기술원 전기 및  
 전자공학과 석사 졸업  
 1988년 한국과학기술원 전기 및  
 전자공학과 박사 졸업.

1989년~1992년 University of Washington 연구원  
 1992년~1993년 삼성전자 수석연구원.  
 1993년~현재 한국과학기술원 전자전산학과 전기  
 및 전자공학전공 교수.

<주관심분야: 영상처리, 영상압축, 의료영상시스템, 멀티미디어 시스템>