

論文2000-37SP-5-4

적응적으로 임계값을 결정하는 블록 기반의 디지털 감시 시스템용 움직임 검출 알고리즘

(A Block-based Motion Detection Algorithm with Adaptive Thresholds for Digital Video Surveillance Systems)

梁 允 碩 * , 李 東 濼 **

요 약

본 논문은 입력 영상에 따라 적응적으로 구해진 임계 값을 이용하여 움직임을 검출하는 블록 단위 움직임 검출 기법을 제안한다. 우선, 현재 영상을 블록의 크기에 따라 블록화 한 후 각 블록의 특징 값을 구하고 이전 영상에서 저장된 블록 특징 값과의 차이 값을 구한 다음 임계 값을 이용하여 움직임을 검출한다. 본 논문에서는 적응적인 임계 값을 구하기 위해서 움직임 벡터를 이용하여 움직임 블록과 배경 블록을 구분하고 각각의 영역에 대한 통계적인 분포를 해석하여 움직임 판별을 위한 각 특징 값의 임계 값을 입력 영상에 따라 자동 조정한다. 모의 실험을 통하여 블록의 크기가 움직임 검출 성능에 미치는 영향, 노이즈의 영향, 특징 값의 종류에 따른 검출의 정확도, 기존의 움직임 검출 알고리즘과의 성능을 비교 분석한다.

Abstract

This paper proposes a block-based motion detection algorithm for digital video surveillance system which adaptively decides the threshold according to the kinds of images. We first compute the features of a block after dividing each image into small sub-block regions, and analyze performance of the motion detection algorithm based on statistic features by using the proposed threshold-decision method. Motion vectors are used to analyze motion degree and adaptively determine the threshold. The simulation results show the performances of motion detection algorithms according to sub-block size, statistic features, noise, and threshold.

I. 서 론

최근 자동 감시 시스템의 수요의 증가와 컴퓨터나 인터넷과 연계한 네트워크 영상 감시 시스템의 개발에 힘입어 기존의 시스템에서 사용되던 아날로그 방식의

감시 카메라나 시스템들의 디지털화가 활발하게 이루어지고 있다. 디지털 방식의 감시 시스템은 이전에 기록된 감시 영상의 화질 열화를 방지할 수 있고 원하는 데이터만을 검색하거나 처리하기가 용이하며 통신망을 이용한 원격 감시 등 다양한 용도와 장점을 가지고 있다. 이러한 디지털 무인 감시 시스템에서 입력된 모든 영상의 데이터를 항상 저장해야 한다면 많은 메모리와 불필요한 시스템 자원의 낭비를 초래할 것이다. 따라서 영상의 효율적인 저장과 시스템 자원의 낭비를 막기 위해서는 침입자가 침입을 했거나 감시하는 영역 내에서 움직임을 감지했을 경우에만 영상을 저장하고 기록해야 할 것이다.

* 正會員, LG電子 DTV研究所
(DTV Research Lab, LG Electronics Inc.)

** 正會員, 漢陽大學校 電子컴퓨터工學部
(School of Electrical and Computer Eng, Hanyang University)

接受日字:2000年5月17日, 수정완료일:2000年7月25日

입력 영상으로부터 움직임을 검출하기 위한 대부분의 알고리즘은 이전에 입력된 참고 영상과 현재 입력된 입력 영상과의 화소 단위의 차를 구하여 얻어진 정보를 이용한다^{[1]-[5]}. 화소 단위의 차 영상을 이용하면 많은 정보를 얻을 수 있는 장점이 있지만 카메라 노이즈나 환경 변화에 민감하게 반응하는 단점이 있다. 또한 고정된 임계값을 이용하기 때문에 상황에 따라 움직임 검출의 성능이 안정적이지 못한 문제점이 있다. 본 논문에서는 임계 값을 자동적으로 조정하여주는 블럭기반의 움직임 검출 알고리즘을 제안한다. 블럭 단위 검출 알고리즘은 소요되는 메모리를 줄이면서 노이즈의 영향을 최소화하기 위해 블럭 단위로 통계적인 특징 값을 계산하여 움직임을 검출하는 알고리즘이다. 이때 블럭 단위 움직임 검출 알고리즘에서 중요한 요소는 움직임을 정량화 하기 위한 적절한 특징 값의 사용, 블럭의 크기, 움직임 유무를 판단하는 임계 값의 결정이다. 따라서 본 논문에서는 블럭의 크기가 움직임 검출에 미치는 영향을 체계적으로 분석하고 대표적인 통계적 특징 값(블럭 평균, 블럭 표준편차)들간의 성능 차이를 분석한다. 또한 적응적으로 임계 값을 자동 조정하기 위하여 움직임 벡터를 이용한 알고리즘을 제시한다. 일반적으로 움직임 검출 알고리즘에서 적절한 임계 값을 어떻게 결정할 것인가 하는 문제가 성능에 많은 영향을 미친다. 따라서 반복 실험을 통한 고정된 임의의 임계 값을 사용하지 않고 적응적으로 임계 값을 결정하는 알고리즘을 제안한다. 실험을 위한 입력 영상은 움직임 정도와 영역에 따른 다양한 경우의 영상을 이용하였다. 최종적으로 블럭 단위 특징 값은 입력 영상에 따라 적응적으로 계산된 임계 값과 비교되어 움직임 여부를 결정 한다. 임계 값을 적응적으로 구하기 위해서 움직임 벡터를 이용하여 움직임 블럭과 배경 블럭을 분류한 뒤 각각의 통계적인 분포를 산출하여 이 결과를 토대로 적절한 임계 값을 자동 조정한다.

본 논문의 전체 구성은 다음과 같다. 2 장에서는 기존의 움직임 검출 알고리즘들의 주요특징들에 대하여 설명하였고, 3 장에서는 제안하는 움직임 검출 알고리즘의 전체 구조와 임계 값의 결정 방법 등에 대하여 다루었다. 4 장에서는 모의 실험을 통한 제안된 알고리즘의 성능 분석과 계층별 복잡도, 노이즈의 영향, 기존의 움직임 검출 알고리즘과의 성능 분석 등을 다루었다. 마지막 5 장에서는 본 논문에 대한 결론을 맺는다.

II. 기존의 움직임 검출 기법

입력 영상으로부터 움직임을 검출하기 위한 대부분의 알고리즘은 이전에 입력된 참고 영상과 현재 입력된 입력 영상과의 차 영상을 이용한다. 차 영상으로부터 얻어진 화소 값으로부터 통계적 특징 값(히스토그램, 평균, 표준편차, 기타 특징 값)을 이끌어내고 실험을 통하여 얻어진 적절한 임계 값과 비교하여 움직임 여부를 판단한다. 그 외에 움직임 검출 방법에는 블럭화를 이용한 움직임 검출 방법, 기타 특정한 환경이나 목적에 따른 특정 용도의 움직임 검출 방법 등이 있다. 참고 논문^{[1]-[5]}은 기본적으로 차 영상의 화소 값을 이용하여 움직임과 상황 변화를 검출한다. 먼저 이전 영상(배경 영상, 참조 영상)과 현재 영상의 화소 단위의 차이 값을 구한 후 움직임 검출을 위하여 차 영상의 통계적 특성을 유도한다. 그림 1에서와 같이 얻어진 차 영상은 움직임 검출을 위한 각각의 목적과 환경에 맞도록 유도된 특징 값을 이용하여 임계 값과의 비교를 통하여 움직임 여부를 판단한다. 특징 값으로는 대표적으로 평균이나 표준 편차를 이용하거나 그 외 히스토그램이나 새로이 모델링한 특징 값 등을 사용한다. 참고 논문^[1]은 실외에서 환경의 변화를 고려한 화소 값 기반 움직임 검출 알고리즘을 제안하였다. 이전 영상과 현재 영상의 차 영상으로부터 화소 값들의 시간 축 상에서의 변화를 이용하여 3 가지의 새로운 특징 값(time correlation, background correlation, normalized distance)을 정의하고 이 값들을 이용하여 배경 영상을 검출한다. 검출된 배경 영상은 현재의 영상과 차이 값

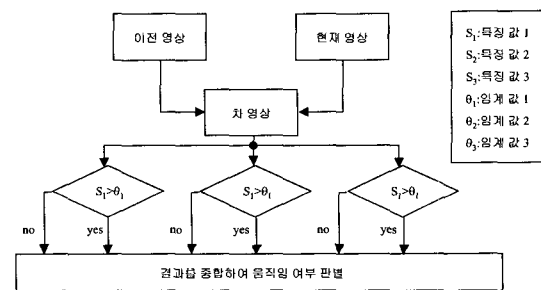


그림 1. 차 영상을 이용한 일반적인 움직임 검출의 흐름도

Fig. 1. Flowchart for motion detection based on frame difference.

을 구하여 움직임 여부와 이동 물체를 판별한다. 참고 논문^[2]은 주변의 환경에 적응적으로 대처 하고 실시간으로 동작하는 영상 감시 시스템을 제안하였다. 이는 참고 논문^[1]에서 제안된 방법을 개선한 것으로, 시간 축 상에서의 변화를 이용하여 개선된 새로운 특징 값을 유도하고 이 값을 이용하여 배경 영상을 추출한 후 현재 영상과의 화소 값의 차를 구하여 이동 물체 여부를 판별한다. 참고 논문^[4]는 카메라의 흔들림이 있는 상황에서 움직임을 검출하는 알고리즘을 제안하였다. 차 영상의 화소 값으로부터 평균과 표준 편차를 구하고 이 결과를 기본으로 카메라의 흔들림의 영향을 최소화하기 위하여 수리 형태론적 기법을 사용하여 카메라의 흔들림에 의한 에지 영상을 제거하고 상황 변화의 원인이 되는 물체를 검출한다. 수리 형태론적 방법을 사용하면 카메라의 흔들림에 의한 노이즈를 열림과 불림 연산을 통하여 제거할 수 있다는 장점을 지니지만, 반면에 열림과 불림 연산의 반복으로 인하여 연산량과 지연 시간이 증가하는 단점을 지닌다.

블럭 단위 움직임 검출 방법은 노이즈에 강인한 성질을 갖게 하기 위해서 영상을 일정한 블럭 단위로 분할하고 각 블럭의 통계적 특성을 분석하여 움직임 유무를 검출한다. 블럭 단위로 통계적 특징 값을 구하면 화소 값을 이용한 움직임 검출 보다 노이즈에 대한 영향을 감소시키고 카메라의 흔들림 등에 덜 민감한 효과를 얻을 수 있다. 참고 논문^[3]는 선박이나 자동차, 엘리베이터와 같이 흔들림이 있는 환경에서 움직임이나 상황을 검출하는 알고리즘을 제안하고 있다. 카메라 흔들림의 영향을 최소화하기 위하여 입력 영상을 블럭 단위로 블럭화 한 후 각 블럭에 해당하는 통계적인 특징 값을 이용하여 실험을 통해 얻어진 임계 값과 비교하여 입력 영상의 상황을 결정한다.

이러한 블럭 단위의 움직임 검출 알고리즘에서는 블럭의 크기에 따른 분석이 체계적으로 연구되지 않았고 기존의 알고리즘들은 고정된 임계값을 이용하기 때문에 입력 영상의 종류에 따라 움직임 검출의 성능이 일정하지 못하는 문제점이 있다.

III. 제안하는 움직임 검출 기법

그림 2는 본 논문에서 제안한 움직임 검출 기법의 전체적인 흐름을 나타낸다. 먼저 이전 영상과 현재 영

상을 블럭화하고 특징 값을 뽑아낸 후 이러한 블럭 특징 값들의 차이 값(블럭 평균의 차이 값 : *Difference of Block Mean(DBM)*, 블럭 표준 편차의 차이 값 : *Difference of Block Standard deviation(DBS)*)을 각각의 블럭에 대하여 구한다. 그리고 각 블럭의 *DBM*과 *DBS*를 이용하여 *DBM MSE(Mean Square Error)*, *DBS MSE* 값을 구하고 이 값을 움직임 벡터(*Motion Vector*)를 이용하여 예측된 임계 값과 비교하여 현재 영상에 대한 움직임 여부를 결정한다.

1. 움직임 검출을 위한 블럭의 구조 및 특징값

블럭 단위 움직임 검출 알고리즘에서 블럭의 크기는 움직임 검출의 성능에 가장 중요한 영향을 미친다. 블럭의 크기가 너무 작으면 조그마한 물체의 움직임에도 민감하게 반응하지만 블럭 내 화소 값의 수가 적기 때문에 이동 물체의 감지에 필요한 통계적 특성을 확보하기가 어렵고 외부 노이즈나 흔들림에 많은 영향을 받는다. 반면에 블럭의 크기가 너무 크면 블럭 내에서 움직이는 작은 물체에 대한 움직임 감지가 불가능하다. 따라서 블럭 단위 검출 기법에서는 블럭의 크기에 따른 검출의 성능 분석이 무엇보다도 중요하다. 본 논문에서 적용한 블럭의 크기는 4의 배수를 원칙으로 하고 16×16 보다 큰 블럭을 갖는 계층의 경우 블럭의 크기는 16의 배수를 원칙으로 한다. 또한 입력 영상의 크기

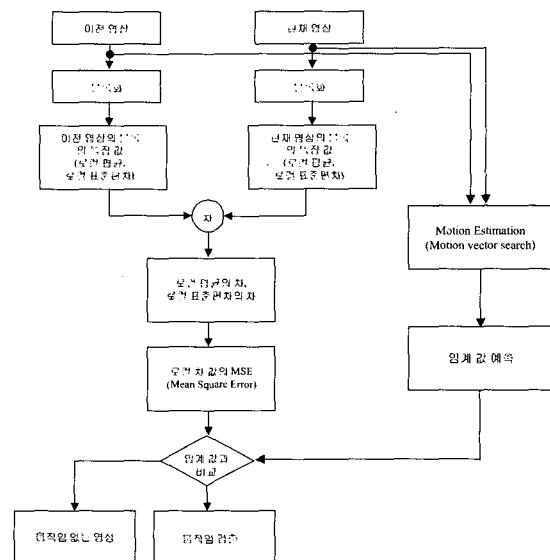


그림 2. 본 논문의 움직임 검출 흐름도
Fig. 2. Flowchart for motion detection proposed in this paper.

를 고려하여 블록화가 최대로 될 수 있는 크기로 각 계층별 블록의 크기를 정한다. 그림 3과 같이 352×240 입력 영상을 사용하였을 경우 총 7계층으로 나누어진 다. 표1에는 각 계층에 따른 프레임 내에서 전체 블록의 수를 나타내었다.

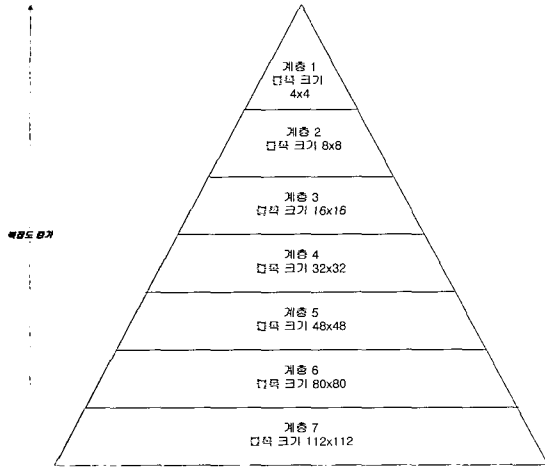


그림 3. 352×240 영상에서 움직임 검출의 계층
Fig. 3. Layers of motion detection for the image of 352×240.

표 1. 352×240 영상에서 각 계층별 블록의 수

Table 1. No. of blocks for each layer.

계층	계층1 (4×4)	계층2 (8×8)	계층3 (16×16)	계층4 (32×32)	계층5 (48×48)	계층6 (80×80)	계층7 (112×112)
블록의 개수	5280	1320	330	77	35	12	6

움직임 검출을 위해 사용되는 특징 값은 주로 통계적 특징 값을 많이 사용한다. 본 논문에서는 이전 영상과 현재 영상의 블록 단위의 평균, 표준 편차의 차이 값을 정의하고 이 특징 값을 이용하여 움직임 검출을 수행한다. 블록의 크기 $n \times n$ 일 때 블록 평균값(Block Mean)과 DBM, 블록 표준 편차 값(Block Standard Deviation), DBS를 구하는 식은 다음과 같다.

$$E_m(x) = \frac{1}{nn} \sum_{i=0}^n \sum_{j=0}^n x(i, j) \quad (1)$$

DBM (Difference of Block Mean)

$$= |E_m(x) - E_{n(t-1)}(x)| \quad (2)$$

$$S_m(x) = \sqrt{\frac{1}{nn} \sum_{i=0}^n \sum_{j=0}^n (x(i, j) - E_m(x))^2} \quad (3)$$

$$= \sqrt{E_m(x^2) - E_m^2(x)}$$

DBS (Difference of Block Standard deviation) =

$$|S_m(x) - S_{n(t-1)}(x)| \quad (4)$$

2. 임계 값의 조정

입력 영상의 움직임 여부는 특징 값(DBM, DBS)과 임계 값을 비교하여 결정된다. 이 때 입력 영상에 따른 적절한 임계 값을 조정하지 않고 고정된 임계 값을 사용하여 움직임을 검출한다면 입력 영상의 환경과 조건에 따라 너무 민감하거나 반대로 움직임이 있어도 검출이 제대로 되지 않는 결과를 초래할 수 있다. 따라서 입력 영상의 조건과 환경에 따른 적절한 임계 값을 얻는 것이 무엇보다도 중요하다. 본 논문에서는 입력 영상을 각 조건과 환경에 따라 분류하고 분류된 영상에 따라 적응적으로 임계 값을 구하여 움직임을 검출하는 알고리즘을 제안한다.

움직임 벡터를 이용한 블록의 분류

임계 값을 적응적으로 결정하는 방법은 움직임이 존재하는 움직임 블록인지 움직임이 없는 배경 블록 인지를 분류하고 분류된 블록의 통계적인 분포를 이용하여 임계 값을 결정한다. 여기서 움직임 블록과 움직임이 없는 배경 블록을 구분하기 위하여 움직임 벡터를 이용한다. 움직임 벡터를 구하는 움직임 추정 방법으로는 블록 정합 움직임 추정 알고리즘이 가장 널리 사용된다. 블록 정합 움직임 추정은 탐색 영역 내에서 현재 입력되는 기준 블록과 가장 잘 대응되는 블록을 찾는 것이다^{[10]-[12]}. 본 논문에서는 FS(Full Search)를 사용하여 16×16 블록 단위로 움직임 벡터를 찾는다. 만약 16x16 블록의 움직임 벡터가 (V_x, V_y) 라면, 16×16 블록의 움직임 벡터의 크기는 식 (5)에 의하여 표현된다. 이 때 블록의 움직임 벡터의 크기를 움직임 거리(moving distance)라고 정의한다.

Motion Vector : $MV = (V_x, V_y)$

$$Moving Distance : D = \sqrt{V_x^2 + V_y^2} \quad (5)$$

움직임 거리 D 가 0이면 움직임이 없는 배경 블록, 0이 아니면 움직임이 존재하는 움직임 블록으로 분류한다. 움직임 검출을 위한 블록의 크기는 각 계층마다 다른데, 16×16 보다 큰 블록의 움직임 거리는 16×16 블록의 움직임 벡터의 크기를 이용하여 유도한다. 방법은 블록의 내부에 속하는 16×16 블록의 움직임 벡터를 이용하여 전체 블록을 대표하는 하나의 움직임 거리 값으로 유도한다. 이 때 주어진 블록 크기 내에서 움직임이 있는 16×16 블록의 움직임 거리는 실제 주어진 블록의 움직임 거리에 영향을 미치지 않기 때문에 식 (6)과 같이 weight 를 0으로 주어서 제거한다. 블록의 크기가 16×16 보다 작은 경우에는 16×16 블록의 움직임 벡터를 각각의 블록에 적용한다.

$$D_n = (D_{16(1)} \times weight_{(1)} + D_{16(2)} \times weight_{(2)} + \dots + D_{16(n)} \times weight_{(n)}) / N_{16} \quad (6)$$

where D_{16} : 16×16 블록의 움직임 거리,
 N_{16} : 블록내의 16×16 블록의 개수,
 D_n : n 번째 블록의 움직임 거리

임계 값 구하기

움직임 벡터를 이용하여 분류된 움직임 블록과 배경 블록의 특성을 이용하여 움직임 검출을 위한 임계 값을 결정한다. 임계 값을 구하는 방법은 다음과 같으며 전체적인 흐름도는 그림 4과 같다.

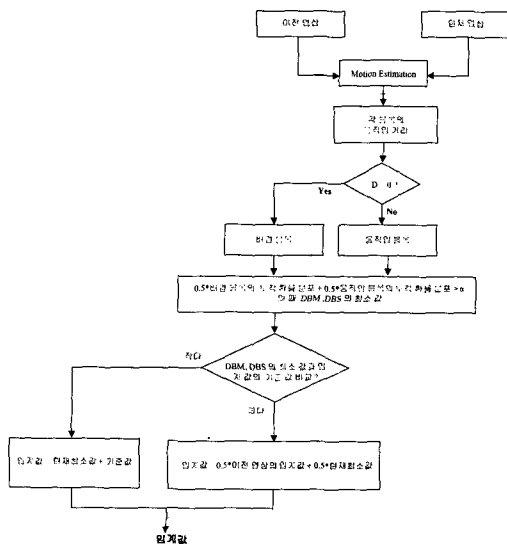


그림 4. 임계 값을 구하는 흐름도
 Fig. 4. Flowchart for deciding thresholds.

Step 1 : 분류된 배경 블록과 움직임 블록을 이용하여 다음식과 같은 배경 블록과 움직임 블록의 확률 분포를 계산한다.

$$P_{back}(dbm) = NB_{dbm} / NB_{total} \quad (7)$$

$$P_{back}(dbs) = NB_{dbs} / NB_{total} \quad (8)$$

$$P_{mov}(dbm) = NM_{dbm} / NM_{total} \quad (9)$$

$$P_{mov}(dbs) = NM_{dbs} / NB_{total} \quad (10)$$

where P_{back} : 블록 평균(dbm) 및 표준 편차(dbs)의 차이 값에 대한 배경 블록에 대한 확률 분포 ($0 \leq P_{back} \leq 1$)

NB_{total} : 배경 블록의 총 개수

NB_{dbm} : 블록 평균의 차이 값 (DBM) 에 대한 배경 블록의 수

NB_{dbs} : 블록 표준 편차의 차이 값 (DBS) 에 대한 배경 블록의 수

P_{mov} : 움직임 블록에 대한 확률 분포

($0 \leq P_{mov} \leq 1$)

NM_{total} : 움직임 블록의 총 개수

NM_{dbm} : 블록 평균의 차이 값 (DBM) 에 대한 움직임 블록의 수

NM_{dbs} : 블록 표준 편차의 차이 값 (DBS) 에 대한 움직임 블록의 수

Step 2 : 식 (11)와 같이 배경 블록과 움직임 블록의 누적 확률 분포의 합의 평균 값이 ($0 < \alpha < 1$) 보다 큰 DBM, DBS (특정값)의 최소값을 실험에 의해 결정된 임계 값의 기준 값(boundary)과 비교한다.

$$S_{dbm} = 0.5 \times SB_{dbm} + 0.5 \times SM_{dbm} > \alpha \quad (11)$$

where SB_{dbm} : 배경 블록 확률 분포 (B_{dbm})의 누적 값

SM_{dbs} : 움직임 블록 확률 분포 (M_{dbm})의 누적 값

그림 5는 움직임 블록과 배경 블록의 확률 분포의 예를 나타낸다. 이는 실험을 통해 Step2에서 제시한 방법에 대한 근거를 제시하기 위한 것이다. 실험에 의하면 배경 블록의 경우 특정값이 낮은 쪽에 대부분 분포한다. 즉 이전 영상과 현재 영상의 변화가 거의 없음을

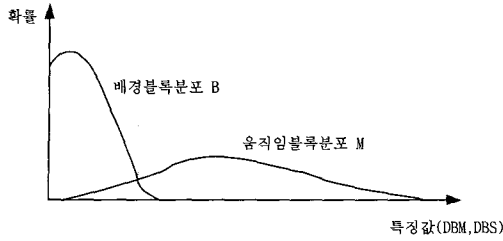
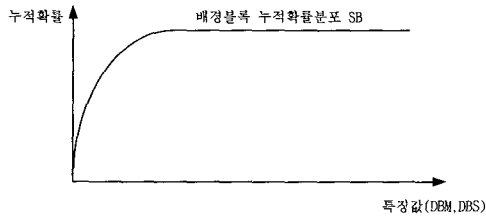
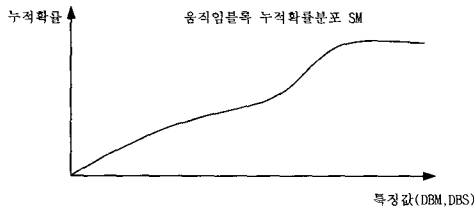


그림 5. 배경 블럭과 움직임 블럭의 확률 분포
Fig. 5. Probability distribution for background blocks and moving blocks.



(a)



(b)

그림 6. (a) 배경 블럭과 (b) 움직임 블럭의 누적 확률 분포
Fig. 6. Cumulative probability distribution for (a) background blocks, (b) moving blocks.

알 수 있다. 움직임 블럭의 경우 확률 분포는 움직임에 따라 넓은 영역에 걸쳐서 분포한다. 이러한 움직임 블럭과 배경 블럭의 확률 분포의 특성을 이용하여 임계값을 구하면 그림에서 점선 부분이 에러를 최소화하는 최적의 임계값의 영역이 된다. 이를 위하여 그림 6과 같은 누적 확률 분포를 사용하면 확률 분포에 비해 정량적인 분포를 얻을 수 있고 특히 움직임에 대한 특성을 잘 나타낼 수 있다. 따라서 움직임 블럭의 누적 확률 분포와 배경 블럭의 누적 확률 분포를 이용하여 적응적인 임계값을 유도한다. 방법은 움직임 블럭과 배경 블럭의 누적 분포의 합을 평균이 보다 큰 블럭 중에서 가장 작은 특정값(특정값의 최소값)을 구하여 이 값을 이용하여 임계값을 구한다. 그리고 기준 값을 설정하는 이유는 움직임이 없는 경우에 구해진 특정값이

매우 작은 값을 갖기 때문에 이 값을 그대로 임계값으로 사용할 경우 너무 낮은 임계값에 의해 움직임이 있는 경우로 오판할 수도 있다. 따라서 기준 값을 설정하여 기준 값보다 작은 최소값은 기준 값과 더하여 임계값으로 사용한다는 배경 영역과 움직임 영역을 검출하는 정도를 조절하는 파라미터 값이며 값이 너무 크면 임계값이 커지므로 작은 움직임을 검출하는 정확성이 떨어지고 값이 너무 작으면 임계값이 작아져서 움직임이 없는 배경 영상을 움직임 영상으로 인식하는 오류가 증가한다.

Step 3 : 위의 과정으로 구해진 임계값은 급격한 임계값의 변화를 방지하기 위하여 식 (12) (13)과 같이 이전의 임계값과 현재의 임계값의 평균값으로 결정한다.

$$TH_{dtm} = 0.5 \times TH_{n(t-1)} + 0.5 \times TH_{n_t} \quad (12)$$

$$TH_{dbs} = 0.5 \times TH_{n(t-1)} + 0.5 \times TH_{n_t} \quad (13)$$

Step 4 : 입력 영상의 이전 프레임과의 평균과 표준편차의 차에 해당하는 DBM, DBS의 프레임 내에서의 MSE(오차 제곱의 평균)와 비교하기 위하여 최종적인 임계값은 위의 임계값을 제공하여 사용한다. 움직임 블럭과 배경 블럭의 MSE를 따로 구한 후 이 두 값의 합을 전체 MSE로 정의한다. 그러한 이유는 움직임 블럭과 배경 블럭의 구분 없이 MSE를 구하면 움직임 영역이 작고 느린 움직임의 경우 움직임 블럭의 개수가 너무 작아서 전체 MSE 값에는 거의 영향을 미치지 못한다. 따라서 움직임의 영향을 효과적으로 반영하기 위하여 움직임 블럭의 MSE와 배경 블럭의 MSE를 각각 계산하여 전체 MSE를 구한다. DBS도 같은 방법으로 MSE값을 얻는다.

IV. 모의 실험 및 고찰

본 논문에서는 모의 실험을 위한 실험 테스트 영상을 위해 sharp VL-DC5 디지털 캠코더와 OnAir video capture board를 이용하여 환경과 조건에 따라 다양한 경우를 포함하는 총 630 frame의 영상을 만들었다. 입력 영상의 크기는 352×240 크기이며 테스트를 위한 입력 영상과 영상의 분류는 그림 7,8과 같다. 입력 영상의 분류에서 실제 움직임에 가장 영향을 많이 미치는

것은 프레임 내에서 차지하는 상대적인 움직임 영역의 크기이다. 움직임 영역이 큰 경우는 움직임 정도가 느리더라도 많은 영역에 걸쳐서 움직임이 나타나기 때문에 검출에 미치는 영향이 그만큼 크다.

임계 값은 α 값에 따라 변화하는데 α 값이 증가 할 수록 임계 값은 커지고 α 값이 감소 할수록 임계 값은 작아진다. 즉 α 값이 증가하면 움직임 검출의 민감도가 낮아지고 α 값이 감소 하면 움직임 검출의 민감도가 높아진다. 본 실험에서는 $\alpha=0.7, 0.8, 0.9$ 일 때 임계 값을 산출하여 실험하였고, 그 중 $\alpha=0.7$ 을 사용하였을 경우 최적의 성능을 얻었다. 그림 9는 16×16 블록 크기에서 값과 그 값에 따라 적응적으로 구해진 임계 값의 예를 보여준다. x축의 값은 입력 영상의 시퀀스를 나타내며 입력 영상은 그림 8에서 분류한 것과 같이 7 가지로 분류하여 case 1~7의 경우를 차례로 입력하여 실험하였다. case 1은(그림 7에서 1번째 영상) 움직임 없는 영상, case 2는(그림 7에서 2번째 영상) 움직임 영역 작고 움직임 정도가 느린 영상, case 3는(그림 7에서 3번째 영상) 영역 크고 느린 영상, case 4는(그림 7에서 4번째 영상) 영역 작고 보통의 움직임 영상, case 5는(그림 7에서 6번째 영상) 영역 크고 빠른 영상, case 6는(그림 7에서 7번째 영상) 영역 크고 보통 영상, case 7는(그림 7에서 8번째 영상) 영역 작고 빠른 영상이다. 그리고 y축 값은 임계 값을 나타낸다. 그림에서 살펴보면 값이 클수록 자동 조정되는 임계 값은 큰 값을 갖는다. 임계 값이 커지면 움직임 검출의 민감도가 떨어지기 때문에 너무 작거나 크지 않은 값을 사용해야 한다.

테스트 영상은 각 움직임 경우 별로 90 프레임을 사용하였는데, 종류별로 10 프레임 씩 편집하여 적용하였다. 실제 실험에서는 특징 값의 차 값이 이용되므로 10 프레임 당 차 값을 얻을 수 있는 경우는 9개가 된다. 그러므로 90 프레임 당 총 81 개의 경우에 대하여 검출을 하기 때문에 아래의 실험 결과 표에서 각 움직임 별 입력 영상을 81로 하였다. 그리고 성능 판단을 위해서 움직임이 없는 경우와 움직임이 있는 경우에 대한 결과를 종합하여 식 (14)와 같은 신뢰도를 유도한다. 신뢰도를 다음과 같이 정의하는 이유는 움직임을 검출하는데 있어서 움직임의 검출도 중요하지만 움직임이 없는 영상을 움직임이 없는 영상으로 검출하는 것도 중요하다. 따라서 움직임 영상을 검출하는 성능과 움직임이 없는 영상을 검출하는 성능을 각각 동등의 비율

로 반영하여 정의한다.

$$P = 100(\%) \times \frac{1}{2} \times \left(\frac{NB}{TNB} + \frac{NM}{TNM} \right) \quad (14)$$

where P : 성능 신뢰도 (Performance reliance)

NB (Number of Back frame) : 움직임 없는 영상에서 올바르게 검출된 frame 수

TNB (Total number of Back frame) : 움직임 없는 경우의 입력 프레임 수

NM (Number of Moving frame) : 움직임 있는 영상에서 올바르게 검출된 frame 수

TNM (Total Number of Moving frame) : 움직임 있는 경우의 입력 프레임 수

표 2 와 그림 10은 적응적으로 얻어진 임계 값을 이용하여 각 계층 별 블록의 크기에 따른 움직임 검출의 성능 결과를 나타낸다. 블록의 크기가 너무 작거나(4×4) 큰 경우 검출 성능이 감소하는 것을 볼 수 있다. 실제 실험에서 DBM을 이용한 움직임 검출은 적응적인 임계 값을 사용하였을 경우 매우 우수한 성능을 기대할 수 있었다. 특히 블록의 크기가 8×8 이나 16×16 인 경우에는 거의 대부분의 움직임에 대해서 올바르게 검출하였다. 반면에 DBS를 이용한 움직임 검출은 DBM을 이용한 결과에 비해 성능에 있어서 많이 떨어진다.



그림 7. 입력 영상
Fig. 7. Test input images.

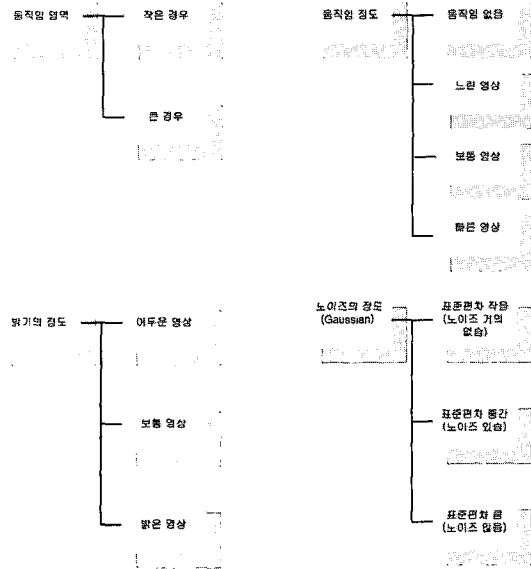


그림 8. 입력 영상의 분류
Fig. 8. Category of input images.

다음은 입력 영상에 대한 검출 성능을 기존의 방법과 비교 분석한 결과를 보여준다. 성능 비교는 일반적으로 적용되는 차 영상을 이용한 움직임 검출 방법과 차 영상에서 환경 변화에 적응적으로 대처하는 이동 물체 감지 기법에 관한 연구^[2]의 결과를 이용하여 본 논문의 결과와 비교 하였다. 표 3에서와 같이 차 영상을 이용하였을 경우에는 블럭 단위 검출과는 반대로 평균에 대한 검출 성능은 떨어지지만 표준 편차를 이용한 검출의 성능은 우수하다는 것을 알 수 있었다. 논문^[2]에서는 차 영상으로부터 3가지 특징 값을 유도하여 움직임 검출한다. 위의 논문의 알고리즘을 살펴보면 움직임 뿐만 아니라 조명의 변화등 다양한 상황도 검출하는데, 본 논문과의 비교를 위해서 움직임 검출에만 초점을 두어 결과를 비교 분석하였다. 신뢰도 P값이 가장 높은 결과(본 논문은 블럭 크기가 16×16 일 때의 결과, 차 영상을 이용한 경우는 임계 값이 3 인 경우, 참고 논문^[2]은 배경 거리 임계 값이 2% 인 경우)에 대하여 각 입력 영상의 움직임 영역과 움직임 정도에 따른 알고리즘의 성능을 비교 분석하였다. 특히 움직임 영역이 작은 경우 즉, 국부적인 움직임이 있는 경우 적응적 임계 값을 사용한 블럭 단위 움직임 검출 알고리즘의 성능이 타 논문 보다 매우 우수하였다. 이러한 성능의 차이는 고정된 임계 값은 낮은 임계 값이 요구되는 국부적이고 작은 움직임에 대한 움직임 특성을 적응적

로 반영하지 못하고 특히 영상 내에서 물체가 움직인 경우 검출 성능이 현저히 떨어지기 때문이다.

그림 12는 블럭 크기에 따른 외부 노이즈의 영향과 특징 값의 변화 정도를 실험한 결과를 나타낸다. 먼저 입력 영상에 대하여 Gaussian 노이즈(표준 편차 3~7)를 임의로 더한 후에 원 영상의 특징 값을 비교하여 노이즈가 미치는 영향을 분석하였다. 그림에서 각각의 값은 노이즈 영상의 특징 값과 원 영상의 특징 값 사이의 MSE를 나타낸다. 10개의 입력 영상을 샘플로 하여 노이즈의 평균적인 영향을 비교하여 보면 본 논문의 경우 타 알고리즘에 비하여 오차 값이 매우 작은 값을 갖는다는 것을 알 수 있다. 10 프레임에 대한 평균적인 노이즈의 영향 정도를 비교하면 본 논문의 경우 블럭 크기가 16x16일 때 0.53165(DBM 이용하는 경우)의 MSE값을 갖고 차 영상을 이용하였을 경우에는 2.5 (표준편차 이용하는 경우), 참고 논문^[2]의 경우 1.73% 정도의 값을 갖는다. 차 영상을 이용한 경우나 참고 논문 [2]의 경우에는 실험 결과에서 최적의 임계 값이 각각 3과 2%이므로 외부 노이즈에 따라 움직임 검출 성능이 달라질 수 있다. 특히 움직임이 느리거나

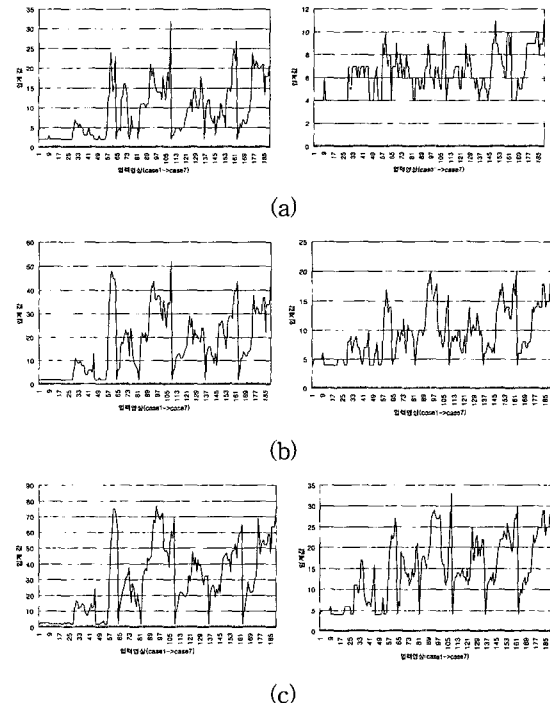


그림 9. DBM(좌), DBS(우) 임계 값의 자동 조정
Fig. 9. Automatic adjustment of DBM and DBS thresholds (a) $\alpha=0.7$, (b) $\alpha=0.8$, (c) $\alpha=0.9$.

움직임 영역이 작은 경우 특징 값은 임계 값과 거의 같은 범위에 있기 때문에 노이즈의 영향을 많이 받는다. 반면에 본 논문에서 제안한 블럭 단위 검출 기법은 블럭 단위 특징 값을 구하는 과정에서 노이즈의 값이

필터링되는 효과가 생기기 때문에(블럭의 평균을 구하기 때문) 노이즈에 대한 영향을 상대적으로 적게 받는다.

표 2. 움직임 검출 결과

Table 2. Results of motion detection.

	계 총	움직임 없음		느림		보통		빠름		검출 성능 신뢰도 P (%)
		작음	큼	작음	큼	작음	큼	작음	큼	
Total frame number		81	81	81	81	81	81	81	81	
움직임 검출 결과 (올바르게 검출된 영상 수)	DBM 이용									
	4 x 4	81	58	81	79	80	73	81	96.502058	
	8 x 8	81	68	81	81	81	73	80	97.119342	
	16 x 16	81	79	81	80	81	80	80	99.548148	
	32 x 32	81	60	77	67	81	70	77	93.82716	
	48 x 48	81	10	58	24	55	43	68	76.54321	
	80 x 80	81	2	38	9	50	30	63	69.753086	
	112 x 112	81	0	23	5	42	19	51	64.403292	
	DBS 이용									
	4 x 4	62	55	55	55	54	54	62	72.736626	
	8 x 8	59	58	63	59	54	61	63	73.251029	
	16 x 16	71	38	63	59	61	58	64	79.115226	
	32 x 32	75	10	57	37	53	52	61	74.074074	
	48 x 48	81	2	14	3	29	23	44	61.831276	
	80 x 80	80	0	8	0	14	4	36	55.761317	
	112 x 112	79	0	3	0	2	2	25	52.057613	

표 3. 차 영상을 이용한 움직임 검출 결과

Table 3. Results of motion detection based on frame difference.

16x16 level 3	임계값	움직임 없음		느림		보통		빠름		검출 성능 신뢰도 P (%)
		작음	큼	작음	큼	작음	큼	작음	큼	
Total frame number		81	81	81	81	81	81	81	81	
움직임 검출 결과 (올바르게 검출된 영상 수)	Mean									
	1	3	54	54	54	54	54	54	35.185185	
	2	29	72	72	72	72	72	72	62.345679	
	3	55	71	81	79	81	81	81	82.716049	
	4	81	36	75	50	78	55	77	88.168724	
	5	81	31	65	38	59	44	70	81.534382	
	6	81	30	62	34	57	42	65	79.835391	
	7	81	28	61	29	54	39	62	78.03642	
	8	81	27	59	29	53	35	62	77.263374	
	Variance									
	1	3	54	54	54	54	54	54	35.185185	
	3	81	63	81	75	78	72	78	93.045267	
	5	81	61	66	63	73	63	72	90.946502	
	7	81	56	63	59	55	62	63	86.831276	
9	81	41	63	59	54	61	63	85.082305		
10	81	36	62	59	54	61	63	84.465021		
11	81	32	62	52	54	55	63	82.716049		
12	81	32	62	46	54	47	63	81.27572		

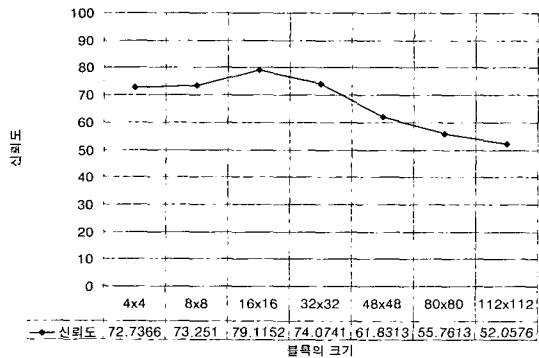
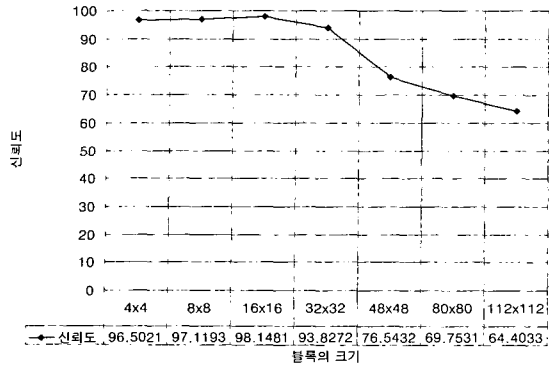


그림 10. DBM(좌), DBS(우)를 이용한 검출 성능 결과
Fig. 10. Results of motion detection performance using DBM and DBS.

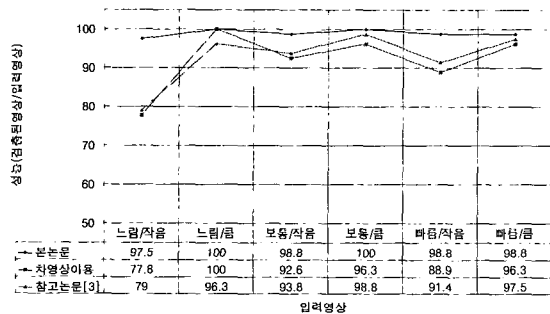


그림 11. 성능 결과 비교
Fig. 11. Performance comparisons with the existing methods.

표 4. 참고 논문[2]의 검출 결과
Table 4. Detection results of reference [2].

16x16 level 3	임계값(배경 가법)	움직임검출	느림		보통		빠름		검출성능 신비도P(%)
			작음	큼	작음	큼	작음	큼	
Total frame number		81	81	81	81	81	81	81	
움직임검출 결과 (움직임검출된 영상수)	1.00%	29	72	72	72	72	72	72	62.3669
	2.00%	81	64	78	76	80	71	79	95.5228
	3.00%	81	57	64	61	61	63	71	88.7808
	4.00%	81	37	63	59	55	61	63	84.7363
	8.00%	81	27	60	28	54	37	63	77.6767

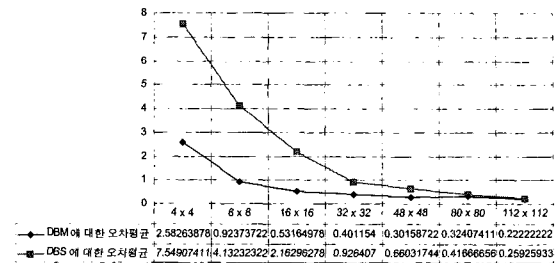


그림 12. 블록 크기에 따른 노이즈의 영향
Fig. 12. Noise effects according to the block size.

V. 결론

본 논문은 입력 영상과 이전 영상의 블록 단위 특징 값을 이용하여 움직임을 검출하는 블록 단위 움직임 검출 알고리즘을 제시하고 성능을 분석한다. 움직임 벡터를 이용하여 블록의 움직임 여부에 따라서 움직임 블록과 배경 블록으로 분류하고 각각의 통계적 분포를 해석하여 최적의 임계 값을 적응적으로 구하였다. 그리고 구해진 임계 값을 이용하여 블록의 크기와 특징 값에 따른 성능을 비교 분석하였다. 제안된 알고리즘은 블록화와 적응적인 임계 값을 사용함으로써 다양한 움직임 특성을 지닌 영상의 움직임 검출에 유용하게 이용될 수 있으며, 특히 국부적이고 작은 움직임에 대한 검출 성능이 기존의 연구 결과에 비해 우수함을 실험을 통해 검증하였다. 또한 기존의 차 영상 기반 알고리즘에 비해 노이즈에 강인한 특성을 보였다.

참고 문헌

[1] S.Nagaya, T.Miyatake, T.Fujita, W.Ito and H.Ueda, "Moving Object Detection by Time-

- correlation-based Background Judgement”, Proc. Of ACCV '95, pp.II-717-722, 1995.
- [2] Y.H.Joo and Y. S. Moon, “A Moving object detection algorithm for video surveillance”, ITC-CSCC '97, vol. 1, pp.165-168, 1997.
- [3] Y.W.Ha, H.T.Kim, K.H.Kang and S.C.Kim, “Event Detection Algorithm Based on Statistics of Subblock Images”, Journal of The Institute of Electronics Engineers of Korea, Vol.36-s, No.7, pp.124-133, 1999.
- [4] H.T.Kim, et. al. “Morphological detection algorithm of an event on board a ship”, ICSP'98, vol.2, pp. 1241-1244,1998.
- [5] 채욱삼, 이정현, 안용학, 이성욱, “신경망을 이용한 침입자 감시 및 추적 시스템”, 제 8 회 영상처리 workshop 논문집, pp. 167-172, 1996
- [6] Y.H.Kim, K.W.Lee, J.G.Jeon and K.T.Park, “Moving Object Discrimination and Tracking for Unmanned Surveillance System”, Proc. of Visual Communications and Image Processing '96, pp.175-186, 1996.
- [7] J.H.Yin, S.A.Velastin and A.C.Davies, “Image Processing Techniques for Crowd Density Estimation Using a Reference Image”, Proc. of ACCV 1995, pp. III-6-10, 1995.
- [8] M. Fathy and M. Y. Siyal, “An image detection technique based on morphological edge detection and background differencing for real-time traffic analysis”, Pattern Recognition Letters 16, pp. 1321-1330, 1995.
- [9] D.Gibbins, G.N.Newsam and M.J.Brooks, “Detecting Suspicious Background Changes in Video Surveillance of Busy Scenes”, Proc. Of Applications of Computer Vision '96, pp. 22-26, 1996.
- [10] Lia-Man Po and Wing-Chung Ma. “A Novel Four-Step Search algorithm for Fast Block Motion Estimation”, IEEE. Circuits sys. Vol. 6, No. 3, pp. 313-317, June 1996.
- [11] D.H. Lee. “modified four-step block-matching algorithm efficient for hardware implementation”, Electronics Letters, Vol. 35, No. 19, pp. 1622-1623, 1999.
- [12] Komarek and P.Pirsch. “Array architectures for block-matching algorithms”, IEEE trans. Circuits syst. Vol. 36, No. 10, Oct 1989.

 저 자 소 개

梁 允 碩(正會員)

1998年 한양대학교 제어계측공학과 공학사. 2000年 동대학원 제어계측공학과 공학석사. 2000年 2月 - 현재 LG전자 DTV 연구소 근무. 주관심분야는 DTV, 제한수신 시스템, 디지털 감시시스템

李 東 濬(正會員)

1986年 한양대학교 전자공학과 공학사. 1988年 Texas 대학 대학원 전기및컴퓨터공학과 공학석사. 1991年 Texas대학 대학원 전기및컴퓨터공학과 공학박사. 1991年 - 1994年 LG전자 중앙연구소 선임연구원. 1994年 - 현재 한양대학교 전자컴퓨터공학부 부교수. 주관심분야는 DTV, 영상처리 및 압축, 멀티미디어 시스템, ASIC