

이동 객체 감시를 위한 실시간 객체추출 및 추적시스템

강현중*, 이광형**

Realtime Object Extraction and Tracking System for Moving Object Monitoring

Hyun-Joong Kang*, Kwang-hyoung Lee**

요약

실시간 영상에서 객체 추적은 수년간 컴퓨터 비전 및 여러 실용적 응용 분야에서 관심을 가지는 주제 중 하나이다. 하지만 배경영상의 잡음을 객체로 인식하는 오류로 인하여 추출하고자 하는 객체를 찾지 못하는 경우가 있다. 본 논문에서는 실시간 영상에서 적응적 배경영상을 이용하여 객체를 추출하고 추적하는 방법을 제안한다. 입력되는 영상에서 배경영역의 잡음을 제거하고 조명에 강인한 객체 추출을 위하여 객체 영역이 아닌 배경영역 부분을 실시간으로 갱신함으로써 적응적 배경영상을 생성한다. 그리고 배경영상과 카메라로부터 입력되는 입력영상과의 차를 이용하여 객체를 추출한다. 추출된 객체는 내부점을 이용하여 최소사각영역을 설정하고, 이를 통해 객체를 추적한다. 아울러 제안방법의 성능에 대한 실험결과를 기존 추적알고리즘과 비교, 분석하여 평가한다.

Abstract

Object tracking in a real time image is one of interesting subjects in computer vision and many practical application fields past couple of years. But sometimes existing systems cannot find object by recognize background noise as object. This paper proposes a method of object detection and tracking using adaptive background image in real time. To detect object which does not influenced by illumination and remove noise in background image, this system generates adaptive background image by real time background image updating. This system detects object using the difference between background image and input image from camera. After setting up MBR(minimum bounding rectangle) using the internal point of detected object, the system tracks object through this MBR. In addition, this paper evaluates the test result about performance of proposed method as compared with existing tracking algorithm.

▶ Keyword : MBR(minimum bounding rectangle), 실시간객체추적, DRM

• 제1저자 : 김현중

• 접수일 : 2005.04.04, 심사완료일 : 2005.05.20

* 서일대학 인터넷정보과 부교수, ** 서일대학 인터넷정보과 전임강사

※본 논문은 2004년 서일대학 학술 연구비에 의해 연구되었음.

경영상으로 잡음을 현저히 줄였고 실시간으로 객체를 추출 및 추적함으로써 보안 및 감시 시스템으로 효용성을 향상시켰다.

I. 서론

컴퓨터 실시간 영상에서 객체 추적을 위해 컴퓨터 비전 기술을 이용하여 구현하기란 매우 어려운 일임에도 불구하고, 컴퓨터 성능의 발달로 인해 영상 처리 기법의 발전과 더불어 객체 인식과 객체 추적에 관한 연구가 활발히 진행되고 있다. 객체 추적은 카메라부터 입력된 영상에서 움직임을 보이는 객체를 인식하고, 그 움직임을 추정하여 추적하는 것이다. 객체를 추적하는 방법은 보안, 의료, 군사, 교통, 제어분야 등 여러 분야에 응용될 수 있어 그동안 많은 연구와 개발이 이루어지고 있다(1). 하지만, 기존의 실시간 객체 추적 시스템의 구현에는 많은 어려움이 존재한다. 우선 강한 실시간 객체 추적 시스템을 구현하기 위하여 고가의 장비를 요구한다. 움직임을 감지하기 위한 센서를 장착한 카메라나 pan/tilt로 움직이는 카메라 등의 요구이다. 또한 객체 추출이나 추적에 많은 연산을 필요로 하는 알고리즘 등을 적용하여 객체 추출에는 뛰어난 성능을 보이지만 실시간 객체 추적에는 부적합하다.

컴퓨터 비전 시스템을 이용한 인공 시각, 컴퓨터와 인간의 상호 작용, 영상 기반 제어 장치, 감시 시스템 분야에서 사용되는 객체 추적 방법은 각각 특정 환경에서 적합한 동작을 보이도록 몇 가지 제약 조건을 두어 설계되고 있다(2). 본 논문에서는 보안 및 감시 시스템 분야에서 적용되어 질 수 있는 방법을 제안하기 위하여, 저가형 PC카메라의 움직임이 고정되어 있고 외부 환경이 아닌 실내 환경이며 배경영상의 변화가 거의 없다는 특수 환경으로 제약 조건을 가진다.

제안하는 시스템은 시간의 경과에 따라 배경영역에서 잡음생성을 줄이고자 객체영역 이외의 영역을 배경영역으로 갱신하여 항상 최신의 배경영상을 유지하도록 하였다. 이와 같이 생성된 적응적 배경영상과 설치된 PC카메라로부터 실시간으로 입력되는 입력영상의 차를 이용하여 객체의 크기와 위치를 탐지함으로써 객체를 추출한다. 이때 연산의 양을 줄이기 위하여 그물식 탐색방법을 이용한다. 그리고 추출된 객체의 내부점들에 MBR(Minimum Bounding Rectangle)을 설정하여 객체를 실시간으로 추적한다.

본 논문은 실시간으로 입력되는 영상으로부터 적응적 배

II. 관련연구

2.1 기존연구

2.1.1 저수준 해석(Low-Level Analysis)

2.1.1.1 모서리들(Edges)

컴퓨터 비전 응용들에서 더 근본적인 특징으로서 모서리 표현은 Sakai 등에 의한 초기의 객체추출 연구에 적용되었다. Craw 등은 후에 사람의 머리 윤곽을 추적하기 위해서 Sakai 등의 연구에 기초하는 계층적 프레임워크를 제안하였다.

2.1.1.2 그레이 정보(Gray information)

모서리 세부들 이외에도 객체 내부의 그레이 정보 또한 특징들로서 사용될 수 있다. 눈썹, 눈동자 그리고 입술들과 같은 얼굴 특징들은 일반적으로 그것들을 둘러싸고 있는 안면 영역들보다 더 어둡게 나타난다. 이 특성은 다양한 얼굴 특징들을 구별하기 위해서 활용될 수 있다.

2.1.1.3 컬러(Color)

컬러는 객체 외양을 구분하는데 있어서 더욱 강력한 수단이다. 컬러가 가지는 여분의 차원들 때문에 유사한 그레이 정보의 두 형상들이 컬러공간에서 매우 다르게 나타날 수 있다. 가장 광범위하게 사용되는 컬러 모델들 중 하나는 RGB 표현이다. 피부외양에서의 주요한 변화는 대개 휘도 변화(밝기)에 기인하기 때문에, 일반적으로 정규화된 RGB 컬러들이 선호되며 그래서 휘도의 효과는 필터링되어 제거될 수 있다.

2.1.1.4 움직임(Motion)

만일 비디오 시퀀스를 사용할 수 있다면 움직임 정보는 움직이는 객체들의 위치를 결정하는데 편리한 수단들이다. 움직임 분할을 얻기 위한 직접적인 방식은 프레임 차이 해석에 의해서이다. 이 방법은 간단한 반면에 배경의 내용과 무관하게 움직이고 있는 전경을 효과적으로 식별할 수 있다. 얼굴과 신체 부분들을 포함하는 움직이는 실루엣들이 누적된 프레임 차이를 임계치와 함으로써 추출되거나, 얼

굴영역 이외에도 얼굴 특징들의 위치를 결정하기 위해서 프레임 차이를 채택하는 기법들이 시도되었다(3)(4).

2.1.2 특징 해석(Feature Analysis)

저수준 해석으로부터 생성되는 특징들은 애매하기 쉽다. 예를 들어서 컬러 모델을 사용하여 객체 영역들의 위치를 결정하는 것에서 유사한 컬러의 배경 객체들 또한 추출될 수 있다. 이것은 더 높은 수준의 특징해석에 의해서 해결될 수 있는 고전적인 다대일 정합문제이다. 많은 객체추출 기법들에서 객체형태의 지식은 그것들의 애매한 상태에서부터 다양한 특징들을 특성화하고 이어서 입증하기 위해서 채택되었다. 조사된 문헌들에서는 객체형태의 응용과 관련한 두 가지 방법들이 존재한다. 첫 번째 방법은 개인의 안면 특징들의 상대적인 위치결정에 기초하는 순차적 특징 탐색 전략들을 포함한다. 두 번째 방법그룹에 있는 기법들은 다양한 얼굴모델들을 사용하는 유연한 배열들을 특징으로 한다(5).

2.1.3 능동 형상 모델(Active Shape Models)

능동 형상 모델들은 특징들의 실제적인 물리적 및 더 높은 수준의 외관을 묘사한다. 일단 특징에 대해 가까운 근접 내에서 방출되면 능동 형상 모델은 국소 이미지 특징들(모서리들, 밝기)과 상호작용하고 점차적으로 특징의 형상을 얻기 위해서 변형할 것이다. 현대의 객체 추출연구에는 일반적으로 세 가지 종류의 능동 형상 모델들이 존재한다. 첫 번째 종류는 Kass 등에 의해서 1987년에 처음 소개된 snakes라 불리는 일반적인 능동 윤곽을 사용한다. 그 뒤에 객체 특징들의 선형성을 고려하고 snakes의 성능을 높이기 위해서 변형 가능한 템플릿들이 Yuille 등에 의해서 소개되었다. 그 후에 Cootes 등은 객체의 효율적인 해석을 제공하기 위해서 그들이 smart snakes 및 PDM으로 명명한 새로운 일반적이고 유연한 모델의 사용을 제안하였다(6).

2.1.4 선형 부분 공간 방법

1980년대 말에 Sirovich와 Kirby는 사람 얼굴들을 효과적으로 표현하기 위해서 PCA를 사용하는 기법들을 개발하였다. 다른 얼굴 이미지들의 조합이 주어지면 고유벡터들에 의하여 표현되는, 얼굴들의 분포의 주성분들을 찾는다. 그 후에 얼굴집합에 있는 각 개별 얼굴이 적절한 가중들을 사용하여 더 일반적으로는 고유 얼굴들로 나타내지는, 가장 큰 고유벡터들의 선형조합에 의해서 근사될 수 있다. 최근에 Turk와 Pentland는 얼굴인식을 위하여 이 기법을 발전시켰다. 그들의 방법은 개별 얼굴 표현에서 고유 얼굴들의 가중들의 다른 본성을 이용한다. 그들이 $\square\square$ 얼굴공간으로부터의 거리(distance-from -face-space) $\square\square$ (DFFS)라

는 용어를 붙인 이 잔여어리는 거리지도에서 전역 최소값의 관찰을 통해서 얼굴 존재성의 좋은 지표를 제공한다. Pentland 등은 최근에 훈련 집합에 있는 다양한 안면 특징 템플릿들로부터 얻어진 고유특징들로부터 생성되는 DFFS를 사용하는 얼굴 특징 추출기를 제안하였다. 특징 추출기는 또한 다른 이산적인 관찰들의 특징들이 훈련 동안에 사용되기 때문에 다른 관찰각도들 하에서 특징들을 설명하기 위한 더 우수한 능력을 가진다. DFFS 측도는 또한 얼굴특징 추출을 위하여 그리고 얼굴 및 안면 특징 추출을 위하여 Fisher의 선형판별식과 조합되어 사용되었다. 더 최근에 Moghaddam과 Pentland는 확률론 틀 내에서 이 기법을 더 발전시켰다. 표현을 위하여 PCA를 사용할 때 얼굴공간의 orthogonal complement를 무시한다. Moghaddam과 Pentland는 이것이 얼굴공간이 균일 밀도를 가진다는 가정에 이른다라는 것을 발견하였다(7).

2.1.5 신경망(Neural Networks)

얼굴추출을 위한 최초의 뉴럴 기법들은 MLPs에 기초하였는데 가능성 있는 결과들이 매우 간단한 데이터집합들 상에서 보고되었다. 크고 어려운 데이터집합 상에서 결과들이 보고된 첫 번째의 고급 뉴럴 기법은 Rowley 등에 의해서 수행된 연구였다. 최근에 Rowley 등은 이미지 평면에서 모든 각도들에서 얼굴들을 추출하기 위해서 라우터 신경망과 결합하였다. 그들은 얼굴의 각도를 결정하기 위해서 하나의 은닉층과 36개의 출력 유닛들(각 10도 마다 하나의 유닛)을 갖는 완전히 연결된 MLP를 사용한다. Feraud 등은 제약된 생성 모델(constrained generative model, CGM)에 기초하는 다른 뉴럴 기법을 제안했다. CGM은 300(15x20)개의 입력과 출력 유닛들(이미지의 크기에 대응하는)을 갖는, 가중들의 세 개의 층들을 갖는 auto associative fully connected MLP이다. Lin 등은 자동적인 얼굴인식 시스템을 확률론적 결정 기반의 신경망(probabilistic decision-based neural networks, PDBNN)에 기초하여 제안했다. PDBNN은 계층적 모듈형태 구조를 갖는 분류 신경망이다. 네트워크는 DBNN과 비슷하지만 그것은 추가적인 확률론적 제약을 가진다. SNoW(sparse network of windows)라 불리는 새로운 학습 아키텍처가 Roth 등에 의해 얼굴추출을 위하여 적용된다. 얼굴추출을 위한 SNoW는 두 개의 linear threshold units로 구성되는 신경망이다. 두 개의 LTUs는 부울 특징들의 입력공간상에서 작동한다(8).

2.1.6 통계적 방법(Statistical Approaches)

얼굴추출을 위한 몇 가지 다른 통계적 방법들이 존재한

다. 정보이론, support vector machine(SVM), 그리고 Bayes의 결정규칙에 기초하는 시스템들이다. 최대 우도 얼굴추출의 초기 연구에 기초하여, Colmenarez 및 Huang는 Kullback relative information에 기초하는 시스템을 제안하였다. Osuna 등은 SVM을 얼굴추출을 위하여 적용하였다. 제안된 시스템은 Sung과 Poggio에 의해서 개발된 것과 동일한 틀에 따른다. 커널함수로서 2계 다항식을 갖는 SVM은 전역적 최적성을 보장하는 분해 알고리즘을 가지고 훈련된다. 훈련은 부트-스트립 학습 알고리즘을 가지고 수행되며 이미지들은 전처리 된다. Kumar 및 Poggio은 최근에 얼굴들의 실시간 추적과 분석을 위하여 Osuna 등의 SVM 알고리즘을 시스템에 결합하였다. 그들은 과도한 스케닝을 피하기 위해서 입력 이미지들에서 분할된 피부영역들에 SVM 알고리즘을 적용한다. SVM은 또한 view sphere의 다른 부분들을 위하여 독립된 SVM을 생성함으로써 다중 뷰 얼굴추출을 위하여 사용되었다(9).

2.2 기존 연구 분석

기존의 많은 방법들이 객체 추적 및 얼굴 추출을 위하여 제안되어 왔다. 발표된 논문의 결과와 구현을 통한 각각의 방법들을 비교한 결과가 <표 1>에 나타나 있다. 객체 추적과 얼굴 추출에 우수한 성능을 보이는 방법은 상대적으로 처리 속도가 떨어짐을 볼 수 있다. 반대로 처리 속도가 빨라 많은 프레임을 처리할 수 있는 방법은 객체 추출과 얼굴 추출에 뛰어난 성능을 보이지 못하고 있다.

표 1 기존 알고리즘 성능 분석
Table. 1 Analysis of Performance existing algorithm

구분	FPS	Detection rate	Tracking rate
저수준 해석	◎	○	○
특징 해석	○	△	△
능동형상 모델	○	◎	○
선형 부분 공간 방법	△	◎	○
신경망	△	◎	○
통계적 방법	△	○	◎

(◎ : 우수, ○ : 양호, △ : 보통)

III. 실시간 객체 추출 및 추적

3.1 시스템 구조

컴퓨터 비전 시스템을 이용하여 객체를 추적하는 방법은 각각 특정 환경에서 적합한 동작을 보이도록 몇 가지 제약 조건을 두어 설계되어져 왔다. 본 논문에서도 효율적인 실시간 객체 추출 및 추적을 위하여 다음과 같은 제약 조건을 가진다. 센서 장착이나 움직임이 가능한 고가의 카메라가 아닌 저가형 PC카메라를 이용하는데 이는 카메라의 움직임이 고정됨을 의미한다. 또한 공원이나 거리 등의 야외 환경이 아닌 사무실이나 연구실, 기자재실, 자재창고, 주차장, 은행 365일 코너 등 배경영상의 변화가 미미한 특수 환경으로 환경을 제한한다.

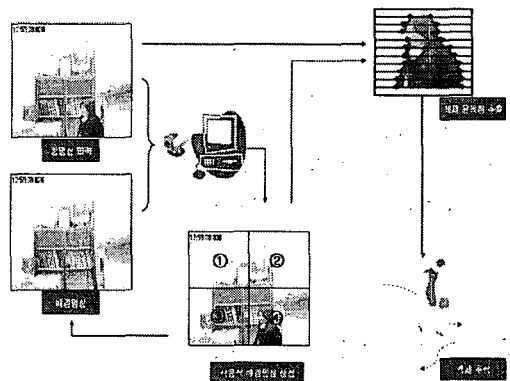


그림 1. 시스템 흐름도
Fig. 1 System flow

(그림 1)은 제안하는 방법의 전체적인 처리 흐름도이다. PC카메라로 실시간 받아들여지는 영상 중에서 객체의 출현이 감지되지 않는 영상을 '초기의 배경영상으로 설정하여 준다. 이 후, 연속적으로 입력되어지는 영상과 '초기 배경영상의 비교를 통하여 강인한 객체 추출을 할 수 있는' 적응적 배경영상을 생성한다. 계속적으로 갱신을 반복하는 배경영상과 실시간 입력되는 영상에서 그물식 탐색 방법을 이용하여 객체의 출현을 감지하고, 감지된 객체의 내부점을 추출한다. 추출된 객체 내부점을 이용하여 객체를 포함하는 최

소 사각형인 MBR을 설정해 주고, 연속되어지는 영상으로부터 일련의 처리를 반복하게 된다. 연속된 MBR의 설정을 추적함으로써 객체의 실시간 추적을 가능토록 하였다.

3.2 시스템 구현방법

3.2.1 적응적 배경 영상 생성

(그림 2)와 (그림 3)은 PC카메라로부터 입력되어진 초기 설정된 배경영상과 객체의 움직임이 감지되지 않은 상태에서 180초 후의 배경영상이다. 사람의 감지 능력으로는 두 그림이 똑같이 보이지만, 시스템에서는 둘을 다르게 인식한다. (그림 4)가 시스템이 이 두 그림을 다르게 인식함을 보이는데, 어떠한 변화도 없는 배경영상이 조명의 영향 등으로 시간의 경과에 따른 잡음 발생의 현상을 보인다. 본 시스템에서 사용되어지는 카메라는 저가이며 고성능이 아닌 일반 PC카메라이다. 저가의 시스템 구축을 위한 선택이었는데 고성능의 카메라가 아니기 때문에 빛에 대해 상당히 민감한 반응을 보인다. 1초, 30초, 60초, 120초 후 영상에서 각각 71개, 1657개, 2061개, 3891개의 최초 배경영상과의 차에서 임계값을 넘어가는 픽셀의 개수가 나타난다. 배경영상과 입력영상의 차이가 균집을 이루고 있으면 배경영상임에도 불구하고 새로운 객체로 인식하는 오류를 범할 수 있다. 여기서 우리는 잡음을 제거하여 보다 정확한 객체의 추출을 위해 적응적 배경영상을 생성한다.

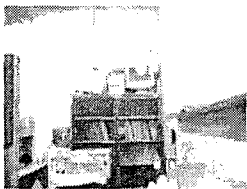


그림 2. 초기배경영상
Fig. 2 Initial image



3. 120초후 배경영상
Fig.3 After 120Sec image

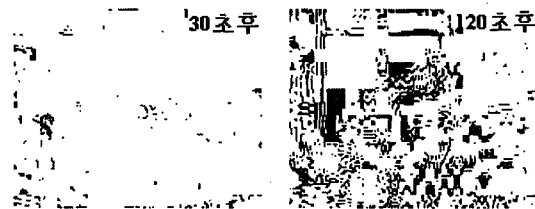


그림 4. 시변하는 배경영상에서의 잡음
Fig. 4 Noise of image the stream of times

전체적인 배경영상의 갱신은 많은 연산량을 필요로 하기

때문에 실시간 객체 추적 성능을 저하시킨다. 제안하는 시스템에서는 객체영역인 $n * m$ 의 영역을 제외한 나머지 영역을 배경영상으로 갱신하면서 객체 추적을 동시에 이루어 질 수 있도록 하였다.

(그림 5)는 적응적 배경영상 생성 방법을 보인다. 이전의 배경영상과 입력영상에서 객체 추출 후 객체영역을 제외한 나머지 영역을 새로운 배경영상으로 대체한다. (그림 5)는 객체의 위치에 따른 취할 수 있는 배경영상의 집합이다. 입력영상에서 객체를 찾아내고 객체의 위치에 따라 객체부분이 아닌 사각영역을 배경영상으로 판단한다. 즉 객체의 위치에 따라 ①~④ 부분으로 분류하고 입력영상을 배경영상으로 대체한다.

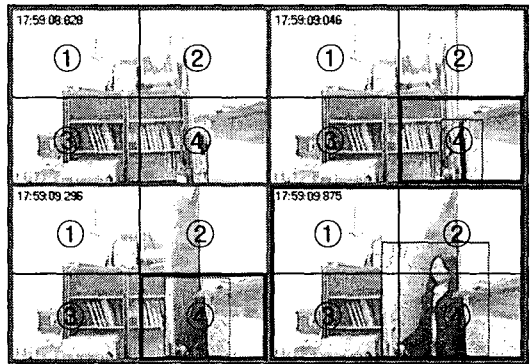


그림 5. 적응적 배경영상 생성
Fig. 5 Create of adjust background image

제안한 적응적 배경영상 방법을 이용하여 실험한 결과, 시변하는 배경영상내의 조명 등에 의한 잡음을 객체 내부에 포함되어 있는 배경영역을 제외한 나머지 영역에서 95% 이상 제거할 수 있었다. 객체내부에 포함되어 있는 배경영역 또한 객체의 움직임에 따라 바로 배경영상으로 적용되기 때문에 배경영상 내의 잡음을 객체로 인식하는 오류를 범하지 않는다.

3.2.1.1 객체의 내부점 추출

객체의 추적을 위해서 객체의 위치 결정을 위한 객체 추출이 우선적이다. 완전 차영상 방식을 도입한 기존 시스템과는 달리, 제안하는 시스템에서는 그물식 탐색 방식을 이용한다. 그물식 탐색 방식은 영상을 위에서 아래로 δ 픽셀 간격으로 탐색해 나가는 방법으로 완전 차영상을 이용하는 시스템보다 처리속도의 높은 향상을 보인다. (그림 6)은 그물식 탐색 방식 및 객체 내부점 추출이다. 객체의 내부점 추출은 그물식 탐색을 하면서 배경영상과 입력영상의

차영상을 이용하게 된다. 배경영상과 입력영상의 탐색라인상의 픽셀들을 각각 RGB채널로 분할한다. 식 (1)에 따라 RGB채널 각각의 차이가 임계값 β 보다 크다면, 객체 내부점 후보군으로 등록하게 된다. 제안하고자 하는 시스템은 보안 및 감시 시스템에 적용되어지기 때문에 사람이라는 객체 이외의 빛의 간섭 등으로 인한 배경에서의 작은 잡음 제거를 위해, 식 (2)를 이용하게 된다. 또한 객체의 일부분이지만 효율적인 잡음 제거를 위하여 어느 정도의 미세한 객체의 일부분은 무시를 한다. 식 (1)를 만족하는 객체 내부점 후보군에서 연속된 픽셀이 v 이하라면 잡음으로 판단하여 제거하고, 원하는 객체 내부점 $Obj(x_i, y_i)$ 를 추출하게 된다.

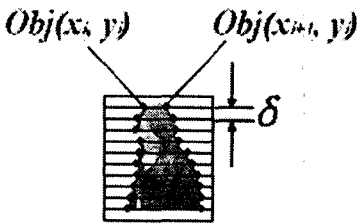


그림 6. 그물식 탐색 방식
Fig. 6 Netty Scan Method

* $Obj(x_i, y_i)$: 객체 내부점

* $Obj_Rec(left, top, right, bottom)$:

MBR의 최대, 최소 좌표

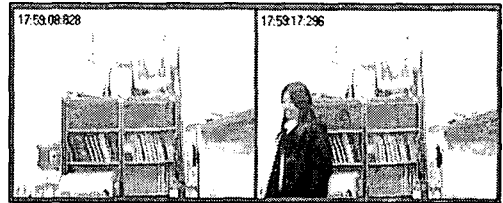
$$\begin{aligned}
 Obj_left &= \min [Obj(x_i)] - \delta \\
 Obj_top &= \min [Obj(y_i)] - \delta \\
 Obj_right &= \max [Obj(x_i)] + \delta \\
 Obj_bottom &= \max [Obj(y_i)] + \delta \dots\dots\dots (1)
 \end{aligned}$$

$$\begin{aligned}
 MBR &= [Obj_left, Obj_top, Obj_right, \\
 &Obj_bottom] \dots\dots\dots (2)
 \end{aligned}$$

(그림 7)은 계속적으로 갱신을 반복하는 적응적 배경영상과 실시간 입력되어지는 영상을 나타내며, (그림 8)은 배경영상과 입력영상에서 그물식 탐색 방식을 이용한 객체 내부점 추출을 나타낸다.

그물식 탐색 방식을 이용하여 객체 내부점을 추출한 것이 완전 차영상을 이용하여 객체를 추출하는 것에 비하여 강건한 객체 추출을 보이지는 않지만, 실시간 보안 및 감시 시스템에서 적용되어질 제안하는 시스템에서는 객체 추적의 대상인 사람의 추출을 빠르게 하는 목적에는 적합함을 볼

수 있고, 기존 시스템과 비교하여 객체 추출의 속도도 향상됨을 알 수 있다.



(a) 배경영상 (b) 입력영상

그림 7. 배경영상과 입력영상
Fig. 7 Background image & Input image

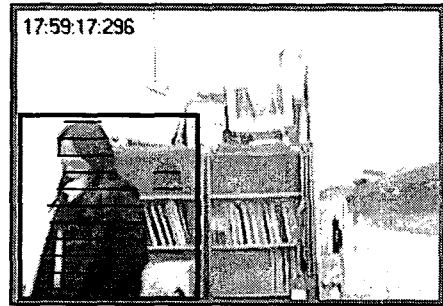


그림 8. 객체 내부점 추출

Fig. 8 extraction of object inner pointer

3.2.1.2 MBR을 이용한 객체 추적

추출된 객체 내부점을 이용하여 MBR(Minimum Bounding Rectangle)을 설정한다. MBR은 객체 추적을 위한 객체를 포함하는 최소 사각 영역으로써, 객체 영역을 추출할 때 속도를 향상시키기 위한 범위를 제한해 주는 역할을 수행한다.

보다 정확한 MBR의 설정을 위한 전처리 과정으로 Median Filtering을 통해 잡음을 제거하였다. 제안하는 시스템에서는 완전 차영상이 아닌 그물식 탐색 방법을 사용하기 때문에 일반적인 $n * n$ block median filter window를 사용하는 것이 아니라, horizontal median filter window를 이용한다. 또한 컬러 영상에 적용되는 median filtering을 위한 절차를 따른다.

컬러 영상을 필터링 할 때 미디언 값이 가지는 유일한 특징을 이용한다. 미디언 값과 집합에서 모든 다른 값들 사이의 차를 합한 것은 집합에서 임의의 다른 값에 대한 차를 합한 것보다 작다는 것이다. 식 (3)와 같이 각 화소 샘플에 대하여 각각의 RGB 채널에 대한 차들을 합한다. 이중 가장 작은 값을 가진 $Distance_i$ 는 필터의 출력인 x_i 에 대응된다.

- * Distance_i : 각 컬러 요소들에 대한 차
 - * N : 필터 윈도우에 표현되는 샘플수
 - * i : 처리되는 화소
 - * j : 다른 화소 샘플
- $x_{med} = x_i$ (단, i 는 Distance_i 중 최소값에 대응하는 값)

$$\sum_{i=1}^N |x_{med} - x_i| \leq \sum_{i=1}^N |y - x_i|$$

$$Distance_i = \sum_{j=1}^N (|red_i - red_j| + \dots) \quad (3)$$

객체 내부점 Obj(x_i, y_i) 의 x, y 좌표 중 식 (4)를 통하여 각각의 최대, 최소 좌표를 구하여 객체를 포함하는 최소한의 사각 영역을 설정해 준다. (그림 9)는 추출된 객체 내부점을 이용하여 설정된 MBR을 보인다. 이를 통하여 객체의 실시간 추적을 가능하도록 하였다.

- * Obj(x_i, y_i) : 객체 내부점
 - * Obj_Rec(left, top, right, bottom) :
MBR의 최대, 최소 좌표
- Obj_left = min [Obj(x_i)] - δ
 Obj_top = min [Obj(y_i)] - δ
 Obj_right = max [Obj(x_i)] + δ
 Obj_bottom = max [Obj(y_i)] + δ
 MBR = [Obj_left, Obj_top, Obj_right, Obj_bottom]

(4)

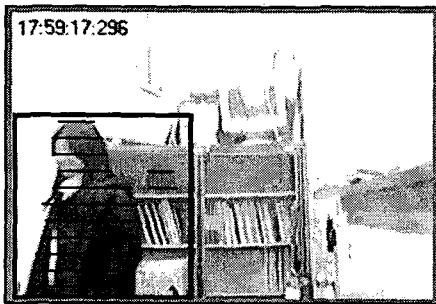


그림 9. 설정된 MBR
Fig. 9 MBR setting

IV. 구현 및 실험평가

5.1 실험환경

본 실험은 카메라의 움직임이 고정되어 있고 배경영상의 변화가 거의 없다는 환경으로 제약하고 입력영상을 실시간으로 받아들여 객체의 위치를 탐지하고, 이동 객체의 추적을 실험하였다.

본 논문에서 제안한 방법의 구현을 위해 Delphi 6.0을 이용하였고, Intel(R) Pentium 4 CPU 2.4GHz, 256M RAM의 PC에서 Microsoft Windows XP 운영체제 하에서 실험하였다. 저가형 PC카메라로부터 받아들이는 배경영상과 입력영상의 크기는 320 × 240의 24bit 컬러영상과 160 × 120의 24bit 컬러영상을 이용하였다.

5.1.1 시스템 구현

(그림 10)은 구현된 실시간 객체 추출 및 추적 시스템으로 입력받은 동영상에서 객체를 추출하고 MBR을 통한 실시간으로 객체를 추적한다.

인터페이스의 상단부에는 제안한 시스템에서 사용되어지는 임계값과 변수에 대한 값을 입력하여 환경과 상황에 가장 적합한 객체 추적을 가능하도록 하였으며, 입력되어지는 동영상의 크기, 해상도 등을 선택적으로 입력되어지게끔 하였다. 배경영상 부분은 계속적으로 갱신되고 있는 배경영상을 보이고, 입력영상 부분은 실시간 입력되어 지는 영상을 보여준다. 왼쪽 하단부는 처리되어진 프레임 수와 초당 처리 속도(FPS, Frame Per Second) 등을 나타내 준다.

객체추출 및 추적 부분은 객체를 추출하고 추적하는 모습을 보이는데, 입력된 영상으로부터 객체를 잡음 없이 추출하고 실시간 객체 추적이 가능하다. 시스템은 저장장치에 JPEG 파일형식으로 시간대별 실시간 객체 추적 과정을 저장함으로써 움직임이 추출된 후 객체의 움직임을 추후 검색이 가능하도록 하였다.

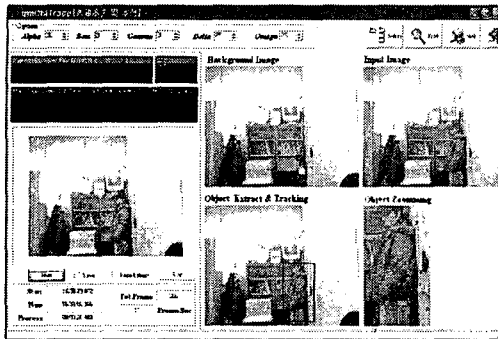


그림 10. 실시간 객체 추출 및 추적
Fig. 10 Extraction & tracking for realtime object.

5.1.2 실험결과

〈표 2〉는 본 논문에서 제안한 방법을 위한 임계값들과 변수에 대한 최적화 실험 결과이다.

표 2. 임계값과 변수에 대한 최적화 실험 결과
table. 2 Optimized Result to critical values

실험값 임계값	1	2	...	5	...	25	...	30	...
α	0.023	0.151	...	0.210	...	0.915	...	0.887	...
β	0.101	0.256	...	0.494	...	0.856	...	0.934	...
γ	0.495	0.531	...	0.926	...	0.349	...	0.324	...
δ	0.951	0.903	...	0.851	...	0.595	...	0.496	...

- α : 적응적 배경영상 생성 임계값
- β : 객체 내부점 후보군 추출 임계값
- γ : 객체 내부점 추출 임계값
- δ : 그물식 탐색을 위한 스캔 간격

〈표 2〉에서 보는바와 같이 α 가 25일 때 적응적 배경영상의 생성으로 많은 잡음을 제거할 수 있었고, β 가 30이었을 때 객체를 성공적으로 추출할 수 있었다. γ 가 5일 때에는 추출된 객체 내부점 후보군에서 많은 잡음 성공률을 보였고, δ 는 보다 조밀한 간격으로 탐색할 때 객체의 추출이 잘 이루어지나, 많은 연산량을 요구해 보다 빠른 탐색을 위하여 5를 선택하였다.

제안 시스템의 임계값들은 하나의 프레임을 분석하기 위한 수치이다. 적응적 배경영상 생성에 대한 임계값과 객체 내부점 후보군 추출 임계값은 기존의 배경영상과 실시간으로 입력되어진 영상의 각 R, G, B 값의 차이를 말하며, 한 번의 스캔으로 두 임계값을 계산 할 수 있다.

객체 내부점 추출 임계값은 배경영상과 입력영상의 후보 추출 임계값의 범위를 초과한 픽셀들의 개수를 말하며, 스캔간격은 그물식 탐색에 사용되는 임계값으로 전체영상을 스캔간격으로 분할해서 비교하기 위하여 사용되었다. (그림 11)은 입력된 동영상으로부터의 실시간 객체 추적 결과를 보이고 있다.

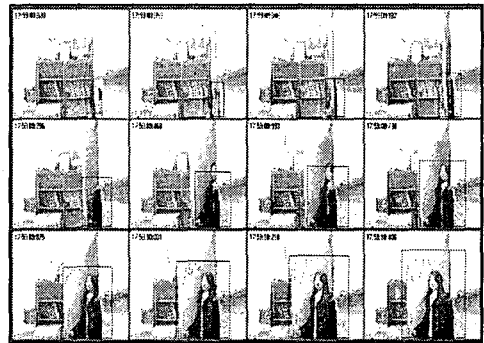


그림 11. MBR을 이용한 실시간 객체 추적
Fig. 11 Realtime object tracking using MBR

방법으로 변화된 장소, 카메라의 위치, 높이, 조명의 변화 등의 여러 상황에서 120여건의 Dataset에서 실험하였을 때 320 × 240 해상도의 영상은 초당 9~13 프레임을 처리하였으며, 160 × 120 해상도의 영상은 초당 17~23 프레임의 처리 성능을 보였다. 이는 기존의 객체 추적 방법에 비하여 처리 속도의 향상을 보이며, 제안한 알고리즘을 통하여 실시간 객체 추출 및 추적이 가능함을 보인다. 객체 추출 및 추적의 처리 속도는 추적 대상인 객체의 크기 등에 따라 민감하게 반응하며 객체가 작았을 경우, 각각 최대 11 프레임과 20프레임까지도 처리 속도를 보였다.

5.1.2.1 성능평가

〈표 3〉과 (그림 12)는 제안한 시스템과 기존 방법들과의 성능 평가이다. FPS는 처리속도 평가기준으로 초당 처리되는 프레임의 수를 나타낸다. 추출율은 하나의 프레임에서 19,200(160 × 120 해상도) 픽셀 중 실제 객체영역의 픽셀 수에 대한 추출된 객체영역의 픽셀 수를 나타낸다. 또한 추적율은 이전 프레임의 추출된 객체 영역 픽셀의 다음 프레임에서 추적된 성능을 나타낸다. 성능 평가에 사용된 영상은 160 × 120 해상도의 24 bit 컬러영상을 사용하였다. 〈표 3〉에서 보는 바와 같이 초당 처리하는 프레임 수와 객체 추출율, 추적율의 복합적인 면에서 기존의 방식들에 비해 향상됨을 알 수 있다.

표 3. 시스템 성능 평가
table. 3 System performance evaluation

구분	FPS	Detection rate	Tracking rate
저수준 해석	15 - 18	91%	90%
특징해석	11 - 16	84%	82%
능동 형상 모델	12 - 15	97%	93%
선형 부분공간 방법	5 - 9	98%	89%
신경망	6 - 9	96%	91%
통계적 방법	5 - 8	92%	97%
제안 시스템	17 - 23	97%	96%

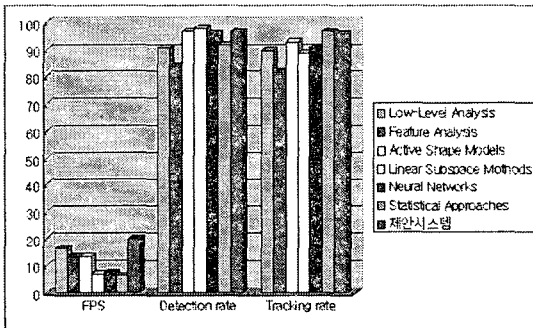


그림 12. 시스템 성능 비교

Fig. 12 Compare of System performance

V. 결론

본 논문에서는 입력되어지는 영상으로부터 배경영상의 실시간 갱신을 통해 객체의 위치를 탐지하여 객체를 추출하고 객체의 내부점을 이용한 MBR에서의 객체를 추적하는 방법을 제안하였다. 실험은 고정 PC카메라와 배경영상의 변화가 거의 없다는 제한된 환경 조건에서 실시간으로 배경영상의 갱신과 객체의 추적이 안정적인 결과를 보여주었다. 이는 객체의 추출 및 추적 알고리즘이 빠르게 수행되어 객체 인식과 결합하여 객체의 움직임 정보와 인식을 통한 보안 및 감시 시스템 등 응용분야에 적용되어 질 수 있다는 것을 기대할 수 있도록 하여 준다. 하지만 입력영상과 배경영상 각각의 R, G, B 값의 차를 이용, 객체의 일부분임을 인식할

때, 잡음과 조명의 영향으로 인하여 실험에 실패한 경우도 발생하여 이에 대한 보완이 필요하다. 또한 조명이 없는 경우 객체의 추출이 불가능하기 때문에 소리와 연계된 시스템이 필요하다. 따라서 배경소리와 새로이 입력되는 소리를 구분하여 조명을 구동할 수 있는 시스템의 개발이 연구 중이다.

참고문헌

- [1] 이희영·최재영·강동구·김홍수·차의영·전태수, "배경영상을 이용한 목표물 추적에 관한 연구", 한국멀티미디어학회 춘계학술발표논문집, 2권 1호, pp. 386-390, 1999.
- [2] 지정규·김용균, "가변 탐색창을 이용한 실시간 객체 추적 시스템", 한국OA학회논문지, 제7권 4호, pp. 14-19, 2002.
- [3] 이강호, "거리 영상에서 인식 특징을 이용한 경계선 검출 기법", 한국OA학회논문지, 제6권 2호, pp. 14-19, 2001.
- [4] E. Hjelm and B.K. Low, "Face Detection: A Survey," *Computer Vision and Image Understanding*, 83(3), pp. 236-274, 2001.
- [5] T.V. Pham, M. Worring, and A.W.M. Smeulders, "Face detection by aggregated Bayesian network classifiers," Technical Report 2001-04, Intelligent Sensory Information Systems Group, University of Amsterdam, 2001.
- [6] M. Rogers and J. Graham, "Robust active shape model search," *Proceedings of the 7th European Conference on Computer Vision*, (4), pp. 517-530, 2002.
- [7] K. Okada and C. von der Malsburg, "Analysis and synthesis of human faces with pose variations by a parametric piecewise linear subspace method," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (1), pp. 761-768, 2001.
- [8] R. Feraud, O.J. Bernier, J.E. Viallet, and M.

Collobert, "A Fast and Accurate Face Detection Based on Neural Network." IEEE Trans. Pattern Analysis and Machine Intelligence, 23(1), pp. 42-53, 2001.

- [9] R. L. Hsu, M. Abdel-Mottaleb, and A. K. Jain, "Face detection in color images," IEEE Trans. Pattern Analysis and Machine Intelligence, 24(5), pp. 696-706, 2002.
- [10] 조영석 · 이주신, "이동물체 고속 추적 알고리즘에 관한 연구", 한국OA학회논문지 제7권 1호, pp. 1-16, 2002.

저자 소개



강 현 중

1980년 성균관대학교 수학교육학 (학사)
 1986년 연세대학교 대학원 전자계산학과(이학 석사)
 1996년 2월 성균관대학교 대학원정보공학(공학 박사)
 1979년 11월~1982년 2월 한국과학기술연구소(KIST)연구원
 1982년 3월~1989년 2월 한화종합금융(주) 전산팀장
 1989년 3월~현재 서일대학 인터넷정보전공 부교수
 <관심분야> 데이터통신, 프로그래밍 언어



이 광 형

1998년 광주대학교 전자계산학과 졸업
 2002년 숭실대학교 대학원 컴퓨터학과 졸업(공학석사)
 2005년 숭실대학교 대학원 컴퓨터공학과 졸업(공학박사)
 현재 서일대학 인터넷 정보과 전임강사
 <관심분야> 영상처리, 에이전트시스템, 멀티미디어데이터검색, RFID 응용