

인지과학, 제16권 제4호, 2005, 333~349.

개선된 블록기반 영상처리기법에 의한 실시간 이동물체 추적시스템

Real-Time Moving Object Tracking System using Advanced Block Based Image Processing

김 도 환
(Dohwan Kim)

최 경 주**
(Kyung Joo Cheoi)

이 일 병***
(Yillbyung Lee)

요약 본 논문에서는 인간의 시각처리 원리 및 블록기반 영상처리기법에 바탕을 둔 실시간 이동물체 추적시스템을 소개한다. 제안하는 실시간 이동물체 추적시스템은 인간의 망막이 갖고 있는 생물학적 메커니즘의 장점을 활용하기 위하여 광각렌즈를 장착한 CCD(Charge-Coupled Device) 카메라와 펜-틸트-줌(Pan-Tilt-Zoom) 카메라를 사용하여 줌인(zoom-in)과 줌아웃(zoom-out) 효과를 동시에 발생시킬 수 있도록 구성되었고, 추적의 오차를 줄이기 위하여 입력되는 영상을 개별 화소 단위가 아닌 여러 개의 블록으로 나누어 처리하는 방식을 채택하여 영상처리를 빠르게 함과 동시에 미세한 잡영도 제거하여 이동하는 물체를 실시간으로 효율적이고도 빠르게 추적한다. 제안하는 시스템의 성능을 확인하기 위해 여러가지 형태의 실험을 수행하였으며 분석 결과를 통해 제안하는 시스템이 미세한 잡영에 거의 방해를 받지 않으면서도 움직이는 물체를 빠르게 감지해서 펜-틸트-줌 카메라를 올바르게 제어함을 알 수 있었다.

주제어 이동물체 추적, 블록기반 영상처리, 광각렌즈, 펜-틸트-줌 카메라

Abstract In this paper, we propose a real time moving object tracking system based on block-based image processing technique and human visual processing. The system has two main features. First, to take advantage of the merit of the biological mechanism of human retina, the system has two cameras, a CCD(Charge-Coupled Device) camera equipped with wide angle lens for more wide scope vision and a Pan-Tilt-Zoom camera. Second, the system divides the input image into a numbers of blocks and processes coarsely to reduce the rate of tracking error and the processing time. In an experiment, the system showed satisfactory performances coping with almost every noisy image, detecting moving objects very fast and controlling the Pan-Tilt-Zoom camera precisely.

Keywords Moving Object Tracking, Block-Based Image processing, Wide Angle Lens, Pan-Tilt-Zoom Camera

* University of Florida, Dept. of Computer Information Science Engineering 박사과정

** 교신저자: 충북대학교 전기전자컴퓨터공학부 저임강사, 충북 청주시 흥덕구 개신동 12번지

연구세부분야: 능동시각(Active Vision), 컴퓨터 시각(Computer Vision), 영상처리(Image Processing),

전화: 043-261-3487, E-mail: kicheoi@chungbuk.ac.kr

*** 연세대학교 컴퓨터과학·산업시스템공학과 교수

서 론

눈은 인간의 감각 기관 중 가장 빠르고 정확하게 사물을 인지할 수 있는 센서 체계로 인간은 시각을 통해 보다 구체화된 실세계의 정보를 뇌로 전달할 수 있으며 이는 보다 정교한 정보처리 결과를 놓음으로써 인간의 생활을 보다 창조적으로 만들어 준다. 이렇게 중요한 기능을 하는 인체 기관이기에 인간의 시각 체계는 매우 복잡하고 정교하게 구성되어 있으며 시각체계에 대한 연구가 시작에 불과한 현재의 기술 수준으로서는 인간의 시각 체계를 기계적으로 한 순간에 구성한다는 것이 결코 간단한 일은 아닐 것이다. 그중 하나의 예로 카메라는 줌인 상태에서는 주위를 동시에 살펴볼 수 없지만 반면에 인간의 눈은 멀리 보는 것과 동시에 주위에서 발생하는 상황을 미세하게나마 감지할 수 능력을 보여준다. 한편 CPU의 고성능화와 영상처리 기술의 비약적인 발전에 힘입어 최근의 컴퓨터 영상 처리 능력은 CCD를 컴퓨터의 눈과 같은 존재로 급부상 시켰지만 아직 인간의 뛰어난 시각 체계와 비교될 수 있는 시스템은 개발되어 있지 못한 실정이다.

본 논문에서는 제안하는 실시간 이동물체 추적시스템은 인간의 망막이 갖고 있는 이러한 생물학적 메커니즘에 입각하여 보다 넓은 영역에 걸쳐서 움직임을 감지할 수 있도록 하기 위해 광각렌즈를 장착한 카메라를, 개별추적이 가능하도록 하기 위해 펜-틸트-줌 카메라를 사용한다. 또한 소프트웨어적인 처리에 있어 카메라를 통해 입력되는 전체 영상을 개별 화소 단위가 아닌 수백 개의 블록으로 나누어 블록을 집합으로, 화소를 원소라 가정함으로

써 각 화소를 일정한 크기들의 블록들에 포함시켜 그들의 대표 값으로 나머지를 처리하도록 하였다. 이렇게 하면 수만 개의 화소 대신 그 1/100에 달하는 수백개의 블록만을 처리하므로 보다 빠르고 정확하게 목표물을 추적할 수 있다. 또한 블록방식의 특성을 십분 활용한 일체화 된 잡영제거 효과를 통해 잡영을 제거하기 위한 별도의 영상처리기법을 사용할 필요가 없게 하므로 전체 시스템에 속도와 더불어 잡영에도 매우 강하다는 특성을 가지게 된다.

인간의 시각 원리

제안하는 실시간 이동물체 추적시스템은 하드웨어 구성면에 있어서 인간의 망막의 생체학적인 특징에 있어서 원근 동시감지 능력이라는 특수한 기능에 기반을 두었기 때문에 시스템의 자세한 설명에 들어가기에 앞서 먼저 인간이 사물을 바라볼 때 어떠한 과정이 진행 되는지에 대한 대략적인 설명을 통해 이해를 돋도록 하겠다. 먼저 바라보는 물체와 눈과의 거리 그리고 물체의 크기 등 사물을 선명하게 보기 위한 여러가지 조건을 계산하여 모양체(ciliary body)라는 근육으로 렌즈(lens)를 제어함으로써 중심와(fovea)에 뚜렷한 상이 맷히도록 도와준다. 망막에는 2가지 종류의 광수용체(photoreceptor)인 간상체(rod)와 추상체(cone)가 있는데, 망막의 전반에 걸쳐서 넓게 분포된 간상체는 색채처리를 맡고 있지 않지만 빛에 민감해서 어두운 저녁이나 밤과 같은 상황에서 잘 반응하는 반면, 추상체는 빛에 둔감하지만 추상체에는 광 파장 감도 곡선(SPD:

Spectral Power Distribution)이 각기 다른 세 가지 종류가 있어 이들의 반응을 토대로 색채처리를 한다. 따라서 사물에 대한 구체적인 정보는 추상체가 전달하게 되며 이 추상체가 집중적으로 분포한 지역이 바로 중심와이다[1]. 이렇게 중심와에 맷힌 영상은 빛에 대한 세포의 발화 정도에 따라서 보다 상세한 정보가 시신경으로 전달되게 된다. 사람의 눈이 원근 동시 감지가 가능한 이유는 중심와에 맷힌 이미지에 대해 집중적으로 원하는 사물을 응시하는 순간에도 추상체가 덜 분포된 부위에서도 역시 지속적으로 세포가 빛에 반응하며 주변시야를 담당하고 있기 때문이다. 또한 최근 원숭이를 대상으로 한 시각표상 실험결과, 움직임에 대한 처리속도가 형태나 색의 처리에 비해 빠르다고 보고 되었다[2].

따라서 본 논문에서는 제안하는 시스템을 생물학적인 메커니즘과 유사한 시각시스템으로 구현하기 위하여 2대의 카메라를 사용하여 광각렌즈를 장착한 고정된 카메라는 움직이는 사물에 대한 직접적인 응시는 하지 않는 반면 주변 시야를 담당하게 함으로써 펜-틸트-줌 카메라로 하여금 정확한 응시를 가능하도록 유도하는 역할을 하도록 하였으며, 움직임을 우선 감지한 후 색 지각을 통하여 보다 올바른 추적이 가능하도록 구성하였다.

기존의 이동물체 추적 방법

국내의 대표적인 추적 연구로서는 블록정합 알고리즘과 삼각형 기반 방법의 결점을 해소하기 연구[3]와, 연속적으로 포착되는 두 영상 사이의 미분에 의한 이동 에너지와 시간을 근

거로 하여 원래의 영상 데이터 내부에서 얻은 경계 영상을 함께 적용함으로써 동체 부분을 정확히 추출하는 연구[4] 등이 있으며, 외국의 경우에는 다수의 카메라 및 색상과 움직임 정보를 통한 얼굴 추적 시스템[5], 다수의 비디오 센서를 이용한 감시 시스템[6], 템플릿 매칭을 통한 이동물체 추적 시스템[7], 상호 협동적인 비디오 센서들을 통해 분산된 환경에서 인간과 자동차에 대한 통합 추적정보시스템(VSAM)[8] 등에 관한 연구 등이 수행되고 있다.

이와 같은 연구에서 움직임을 감지하기 위한 방법으로 광류(Optical Flow) 방법, 블록기반(Block-Based) 방법, 화소 반복(Pel-Recursive) 방법, 베이시안 방법 등이 사용되고 있는데, 그 중에서도 광류 방법과 블록기반 방법이 주류를 이루고 있다. 광류를 이용하는 방법은 움직임의 감지와 더불어 속도와 방향을 함께 추적할 수 있지만 구조가 복잡하고 많은 계산이 요구되므로 구현하기가 쉽지 않다는 단점을 가지고 있으며, 블록기반 방법은 H.261과 MPEG 1-2와 같은 비디오 압축 기술부분에 있어 국제표준으로 채택된 방법이므로, 움직임 추정과 보정 분야에서 가장 널리 이용되는 방법이다[9]. 블록기반 방법은 움직임 영역을 검출함과 동시에 방향 성분도 알아낼 수 있으며, 특징점을 추출해야 하는 방법에 비해 빠른 연산시간을 자랑하지만, 화소변화에 매우 민감하고 잡영, 패턴성 무늬, 크기변화 그리고 회전 등에 매우 취약하다는 단점이 있다[10].

블록기반 방법은 탐색 방법에 따라 크게 단계상호관계(Phase-Correlation) 방법, 블록정합(Block-Matching) 방법 그리고 계층적 움직임 추정(Hierarchical Motion Estimation) 방법[9]으로 나

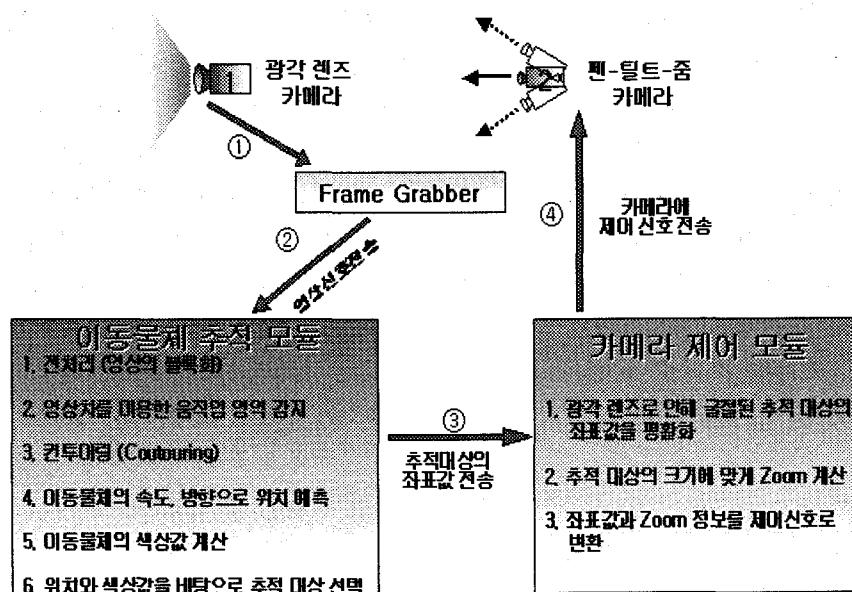
누어 볼 수 있다. 이 중 블록기반 방법의 대표적인 방법이라 할 수 있는 블록정합 방법을 통해 기존 블록기반 방법의 문제점에 대해 분석하였다. 블록정합 방법은 변화가 일어난 블록이 움직임이 일어나기 이전 프레임의 어느 위치에 있는 블록과 가장 일치하는가를 추정하는 방법으로, 움직임을 추정하거나 보상할 때 가장 많이 사용하는 방법[9]이다. 하지만 이 방법은 탐색창(searching window)라는 특정 영역을 정해서 이 영역 내부에서만 2개의 영상을 비교해서 움직임 여부를 알아내기 때문에 물체가 빠르게 이동하는 경우에는 심각한 문제를 발생한다. 예를 들어 이동물체의 형태가 언제나 불변하며 느리게 이동할 경우에는 비교적 올바른 추적이 가능하지만, 물체가 빠르게 움직여서 탐색창을 벗어나게 되면 잘못된 추적을 하게 되는 것이다.

본 논문에서 제안하는 시스템은 블록기반의

특징인 주변의 화소를 하나의 집합으로 포함하는 장점은 그대로 이용하되, 탐색창을 이용해서 추적을 하는 근본적인 한계점은 과감히 버리고 물체의 색상과 운동성을 특징값으로 대신하여 보다 빠르고 광범위하게 움직이는 물체도 올바르게 추적할 수 있도록 구성하였다.

제안하는 실시간 이동물체 추적시스템

제안하는 시스템은 원근 동시감지 기능을 위하여 2대의 카메라를 사용하여 빠르고 불규칙적으로 움직이는 물체를 고해상도로 확대하여 지속적으로 추적하는 것을 가능하게 하였다. 제안하는 시스템의 전체 구성도가 그림 1에 나타나 있다. 제안하는 시스템의 입력모듈은 주변시야를 최대한 확보하기 위하여 화각이 매우 넓은 광각렌즈를 장착한 카메라를 사



(그림 1) 제안하는 시스템의 전체 구성도

용하여 구성하였고, 이를 통해서 들어온 영상 정보는 여러단계에 걸친 추적계산을 통하여 출력모듈인 펜-틸트-줌 카메라로 넘겨진다. 이렇게 넘겨지는 정보는 펜-틸트-줌 카메라로 하여금 매우 빠르게 움직이는 물체를 최상의 해상도로 무리없는 추적을 할 수 있도록 유도해 준다.

광각렌즈를 장착한 카메라를 통한 영상 입력

본 시스템의 최초 입력장치라고 할 수 있는 광각렌즈를 장착한 카메라는 그림 2에서와 같이 화각이 표준렌즈에 비해 2배 이상 넓은 광각렌즈를 장착했기 때문에 넓은 범위에 걸쳐서 주변시야를 확보할 수 있다. 그림 2a는 일반렌즈를 장착한 카메라를 통해서 본 영상이고, 그림 2b는 광각렌즈를 장착한 카메라를 통해서 본 영상이다. 동일한 곳을 촬영한 영상임에도 불구하고 두 렌즈의 화각차이는 마치 줌인 또는 줌아웃을 한 듯한 차각을 할 정도로 큰 차이를 보여준다. 이렇게 광각렌즈의 넓은 화각을 이용하면 보다 넓은 주변시야를

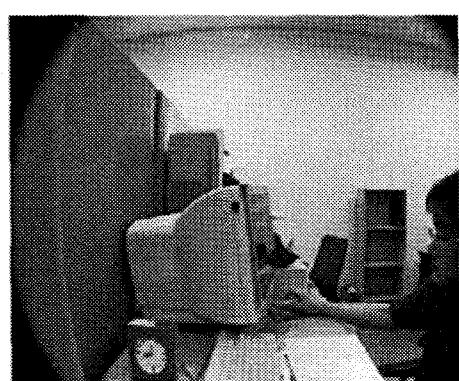
확보할 수 있어 추적 성능을 비약적으로 향상 시킬 수 있다.

영상정보의 전처리(블록화)

광각렌즈를 장착한 카메라를 통해 들어온 영상은 전체 영상을 여러 개의 블록으로 나누는 전처리 과정이 수행되고(그림 3c), 이렇게 전처리 된 결과를 바탕으로 움직임을 추적한다. 제안하는 시스템에서는 320×240 크기의 영상을 입력받았는데, 이를 16×12 크기의 블록으로 나누면 총 400개의 블록이 생성된다. 이렇게 생성된 블록들은 주변의 화소들을 담당하고 있는 하나의 집합으로 볼 수 있는데 이러한 블록을 기반으로 움직임을 감지하면 각 블록내의 임계치(threshold) 값을 조절함으로써 아주 미세한 움직임 변화는 무시할 수 있고 이러한 방법으로 판별이 불가능할 정도로 미세한 영상정보나 배경 전반에 발생하는 잡영 등에 적절하게 대처할 수 있다. 결국 이는 덜 세밀한 영상처리를 가능하게 함으로써 잡영의 영향을 덜 받는 상황에서 움직임을 빠르



a. 일반렌즈로 촬영한 영상



b. 광각렌즈로 촬영한 영상

(그림 2) 일반렌즈와 광각렌즈로 입력된 영상의 차이

게 감지하며 복잡한 연산을 요구하는 후반의 컨투어링(contouring) 작업을 신속하게 처리하는 효과를 얻을 수 있다.

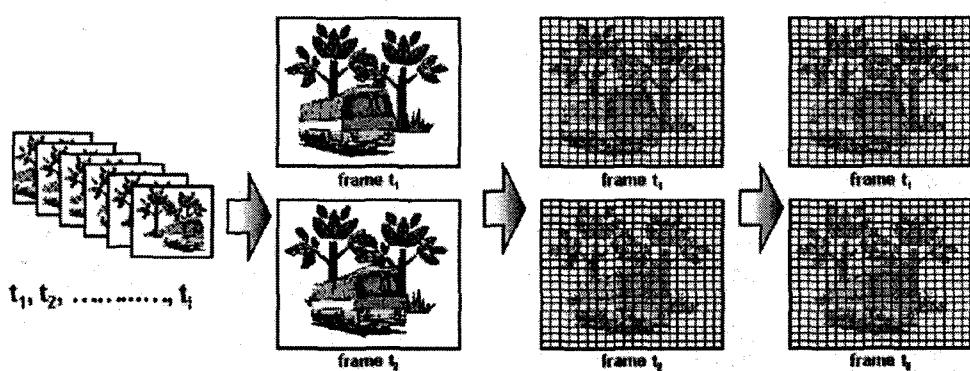
전처리 과정을 포함한 움직임 영역 감지까지의 전체적인 영상처리과정이 그림 3에 나타나 있다. 광각렌즈를 장착한 카메라를 통해 프레임 그레이버(frame grabber)로 입력된 영상 시퀀스(sequence)는 시간순으로 입력되게 되고(그림 3a), 시간적으로 인접한 영상들끼리 서로 비교될 준비가 되어진다(그림 3b). 각각의 영상 시퀀스에 블록화 전처리 과정이 수행되어지고(그림 3c), 이 블록을 기반으로 영상차 연산을 수행하여 움직임 영역을 감지한다(그림 3d).

블록기반 영상처리에 의한 움직임 영역 감지

비교할 2개의 영상을 시간순대로 전처리(블록화) 한 후에는 움직임 영역을 알아내기 위해 필요한 본격적인 작업에 들어간다. 움직임 영역을 알아내기 위하여, 먼저 전처리를 통해 블록화가 이루어진 두 영상에서 같은 위치에 존재하는 블록을 선택하여 비교하고, 그 다음

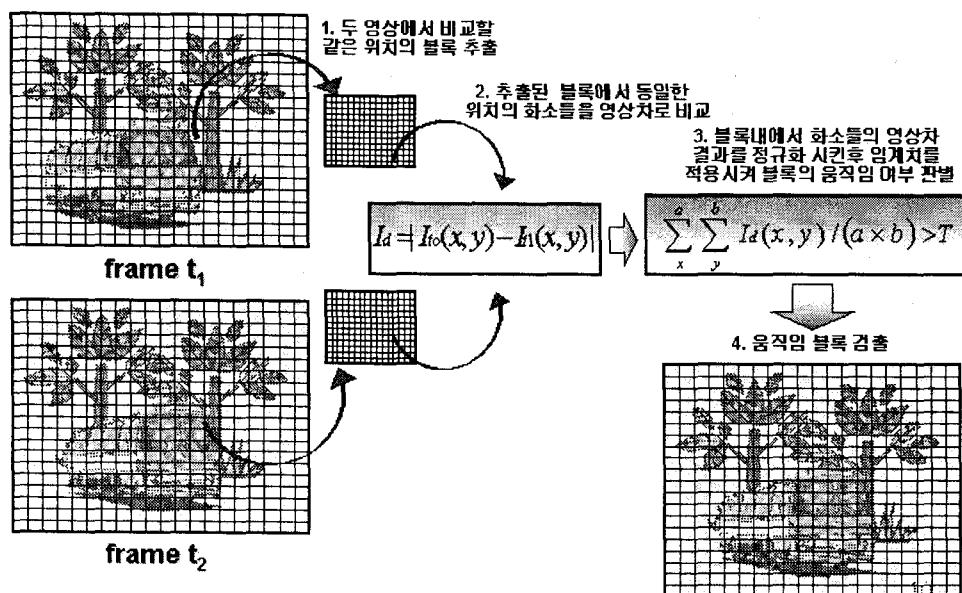
으로 선택된 블록 내에 존재하는 모든 화소들을 영상차 연산으로 비교하게 되며 그 계산 결과를 임으로 주어진 임계치와 비교함으로써 블록의 움직임 여부를 최종 판별하게 된다(그림 4).

본 시스템에서 움직임 영역을 감지하기 위해 먼저, 추출된 2개의 블록내의 화소들을 식 (1)과 같은 영상차 연산을 통해서 순서대로 비교하게 된다. 식 (1)에서 $I_{t_0}(x, y)$ 는 현재 프레임의 (x, y) 에 위치한 현재 프레임의 화소 값이고 $I_{t_1}(x, y)$ 는 같은 위치의 다음 프레임의 화소 값이며 I_u 는 그 차이 값이다. 블록내의 모든 화소들에 대한 영상차 비교 연산이 끝나면 식 (2)와 같이 모든 화소들의 영상차 결과를 합한 후 블록의 크기에 맞게 정규화시키기 위해 블록 사이즈가 $a \times b$ 일 경우 $a \times b$ 로 나누게 된다. 마지막으로 식 (2)의 결과가 임으로 정한 임계치 값보다 크면 움직임이 발견된 것으로 판단하게 된다. 본 시스템에서 임계치 값은 최소 0에서 최대 255까지 설정할 수 있었으며 이 중 잡영에 가장 잘 대처함과 동시에 높은 추적 성능을 보여주는 값인 5로 설정하였다.



a. 연속된 입력영상 b. 비교영상 선택 c. 전처리(블록화) d. 움직임 영역 감지

(그림 3) 움직임 영역 과정



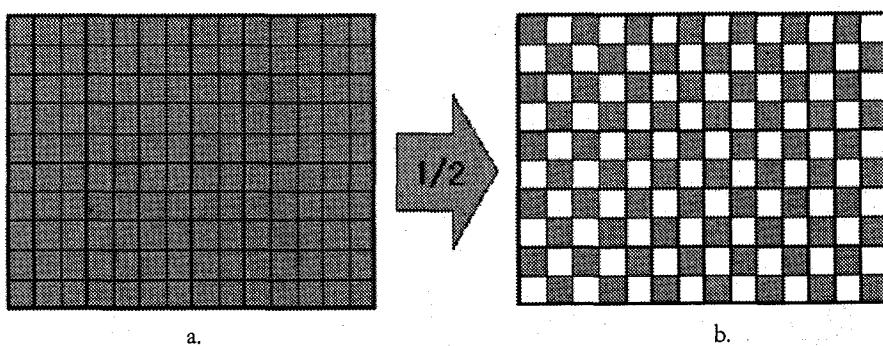
(그림 4) 영상차에 의한 움직임 판별

$$I_d = |I_{t_2}(x, y) - I_{t_1}(x, y)| \quad (1)$$

$$X = \sum_{x=0}^a \sum_{y=0}^b \frac{I_d(x, y)}{(a \times b)} \quad (2)$$

가장 이상적인 실시간 추적시스템은 입력과 추적이 동시에 같은 시간에 이루어지는 것이다. 하지만 시스템에는 당연히 계산처리 시간이 소요되므로 이는 현실적으로 불가능하다. 계산시간이 조금만 지체되어도 실시간 추적시스템은 언제나 지나간 흔적만 추적하게 되므로 실시간 추적시스템에 있어서 시스템의 반응속도를 향상시키는 작업은 무엇보다 중요하다고 할 수 있다. 본 시스템은 영상처리에 소요되는 시간을 최대한 단축시키기 위해 그림 5와 같이 모든 화소를 전부 사용하는 것이 아닌 홀수나 짹수 번에 해당되는 화소만을 이용하는 테크닉을 사용하였다. 그림 5a의 전체 큰 사각형은 16×12 크기의 하나의 블록을 상징하

며 내부에 존재하는 각각의 작은 사각형들은 화소들을 나타낸다. 그림 5b와 같이 좌로부터 홀수 또는 짹수에 해당되는 화소만 입력받아도 움직임 감지에 성능저하를 가져오지 않으면서 동시에 영상의 입력과 처리에 2배 이상의 연산을 절약할 수 있었다. 즉 영상차 비교는 모든 화소를 다 비교할 수도 있지만 본 논문에서는 블록내의 비교할 화소 수나 비교 영역을 줄여 제곱단위비율로 연산량이 줄어들게 함으로서 속도를 많이 높일 수 있다. 또한 이 때 미세하게 손실된 정보는 최종단계에서 MAR(Moving Area Rectangle)를 생성함으로써 하나의 움직임 물체로 묶어주는 단계에서 다시 보충되므로 손실되는 정보는 결과적으로 없게 된다.



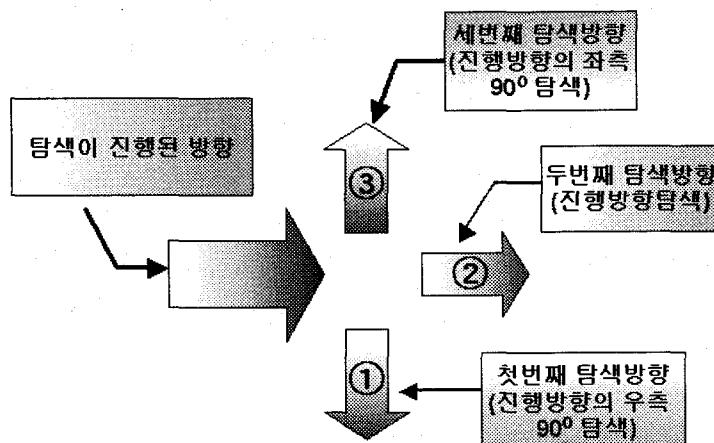
(그림 5) 비교 화소 감소에 의한 블록 내 비교영역 절감

움직임 영역 컨투어링(Contouring)

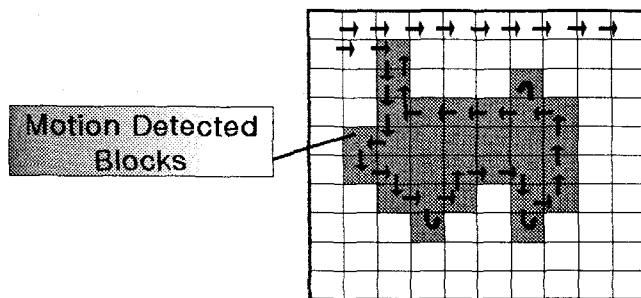
움직임 영역을 감지했으면 전체 움직임을 물체별로 분류해서 최종 추적좌표를 산출해야 한다. 물체별 분류는 이동물체의 외곽을 찾는 컨투어링 작업을 통해서 이루어지게 되는데, 이 작업은 추적을 위한 전체 연산 중 가장 많은 시간을 소요하며 또한 오류도 많이 발생하기 때문에 매우 중요한 작업이라고 할 수 있다. 기존의 컨투어링 방식은 화소기반처리 방식을 사용한 결과 너무 많은 계산 시간과 오

류가 발생하였다. 하지만 제안하는 시스템은 여러개의 화소를 블록으로 묶어주는 작업을 통해 이러한 문제점을 해결할 수 있었다. 본 논문에서는 최대 76,800(320×240)개의 화소들로 처리해야 할 컨투어링을 400(20×20)개의 블록들만으로 처리하기 때문에 이 부분에 있어서의 많은 계산 절감 효과를 볼 수 있었다.

제안한 시스템에서 사용된 움직임 블록 탐색 알고리즘이 그림 6에 나타나 있다. 으로서 상·하·좌·우로 인접한 움직임 블록들만을 하나의 움직임 영역으로 판별한다. 처음 탐색



(그림 6) 컨투어링 알고리즘



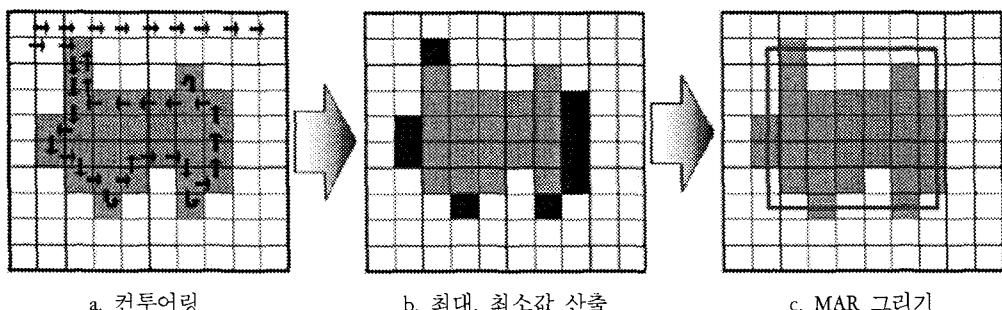
(그림 8) 움직임 영역에 대한 컨투어링 결과

이 시작될 때는 일단 움직임 블록이 감지가 되면 지나온 방향의 +90도인 방향을 제1순위로 탐색하게 되고 그곳에 움직임 블록이 없으면 지나온 방향과 일치한 방향인 0도 방향을 탐색하게 된다. 만일 그곳에도 움직임 블록이 없으면 -90도 방향을 탐색하게 되며 그곳에도 없으면 오던 곳으로 되돌아가게 된다. 연뜻 보면 ‘연결요소(connected component)’와 유사하지만 화소값이 아닌 블록값으로 처리함으로써 속도가 매우 빠르다는 점과 대각방향 탐색을 하지 않음으로써 불필요한 연산이 없고 또한 이해하기 쉬워서 적용이 용이하다는 점이 특징이다.

앞에 설명한 컨투어링의 연산결과가 그림 7에 나타나 있다. 움직임 영역들이 영상차 계

산을 통해서 검출되게 되면 입력된 영상 (320x240)의 맨 위의 좌측에서부터 우측으로 스캔을 하게 되며 우측으로 다 갈 때까지 움직임 영역을 못 찾았으면 한 블록 내려와서 다시 좌측으로부터 탐색이 시작된다.

이러한 컨투어링 작업을 하게 되면 움직임 영역의 외곽에 위치한 블록들의 좌표가 나오게 되는데 이들의 최대, 최소 좌표들의 평균을 구하면 물체의 중심좌표가 얻어지게 된다. 이는 시스템의 추적에 필요한 최종결과이기도 하다. 또한 최대, 최소 좌표들 연결시키면 MAR를 그릴 수 있는데 이러한 MAR는 관찰자가 시스템이 올바른 추적을 했는지의 여부를 확인하는데 도움을 준다. 그림 8은 컨투어링 후 움직임 영역의 중심좌표를 구하고 MAR를



(그림 7) 컨투어링 후 최대·최소 좌표값을 이용하여 MAR 그리기

그리는 작업을 설명한 것이다.

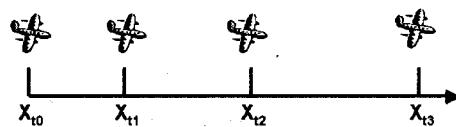
이동지점 예측 및 색상값 계산을 통한 최종 추적 보정

영상입력에서 컨투어링까지의 과정을 통해 이동물체의 중심좌표를 알아냈다고 해서 추적이 끝난 것은 아니다. 지금까지 수행된 작업들은 움직임 감지일 뿐 이동벡터와 특징값에 의한 추적기능은 아직 부여되지 않았기 때문이다. 영상에서 추적물체가 하나인 경우에는 움직임 감지만으로 올바른 추적을 할 수 있지만 실세계 영상에서는 단일 이동물체보다는 다수의 이동물체가 존재하는 경우가 대부분이며 하나의 물체에도 각 부위에 따라서 여러 개의 독립적인 움직임을 보여줄 수 있기 때문에 이동물체 감지는 추적에 도움을 주는 단계일 뿐 그 자체로 추적이 끝난 것으로 볼 수는 없다.

제안하는 시스템에서는 이동물체의 속도와 방향을 토대로 다음 프레임의 위치를 예측해서 그 예측 지점에 가까이 위치하는 물체를 동일 추적대상으로 선택한 후, 영역 간 색상값을 이용한 상관관계를 계산하여 유사도가 높은 영역을 이전 프레임의 영역과 동일한 물체로 선정하여 추적하는 방법을 사용하였다.

이동지점 예측

영상 프레임은 거의 일정한 시간 간격(초당 약 25프레임)으로 입력되므로 프레임 간에 이동한 거리는 상대적인 속도가 된다. 따라서 현재 지점까지의 속도와 가속도를 계산하면 앞으로 이동할 거리를 예측할 수 있다. 그림 9를 예를 들어 이동지점 예측법에 대해 설명하



(그림 9) X축 위에서 이동지점 예측

도록 하겠다. 그림 9는 X축 위에서 비행기가 X_{t_0} , X_{t_1} , X_{t_2} 지점을 차례로 이동하고 있는 것을 나타내고 있으며, X_{t_3} 는 앞으로 비행기의 예상 이동지점이다. 여기서 현재 시점에서 비행기가 있는 지점은 X_{t_2} 라 가정한다.

X_{t_0} 에서 X_{t_1} 로 이동한 거리는 식 (3)에서와 같이 ' $X_{t_1} - X_{t_0}$ '가 되며 바로 상대적인 속도가 된다. 같은 방법으로 X_{t_1} 에서 X_{t_2} 로 이동하는 테 필요한 속도는 식 (4)와 같이 ' $X_{t_2} - X_{t_1}$ '가 된다. 이제 X_{t_2} 지점에서 속도의 변화율을 구하면 X_{t_3} 의 위치를 예측할 수 있게 되는데 가속도는 과거 지점에서의 속도의 차를 구하면 되므로 식 (5)와 같은 결과를 얻을 수 있다. 또한 현재 지점인 X_{t_2} 에서의 속도와 가속도를 알고 있기 때문에 식 (6)과 같이 X_{t_3} 의 위치를 예상할 수 있다. 절대값을 사용하지 않고 마이너스 값이 나올 수 있는 가능성을 남겨둔 이유는 그 기호가 바로 방향과 관계되기 때문이다.

$$X_{t_1} \text{에서의 속도} = X_{t_1} - X_{t_0} \quad (3)$$

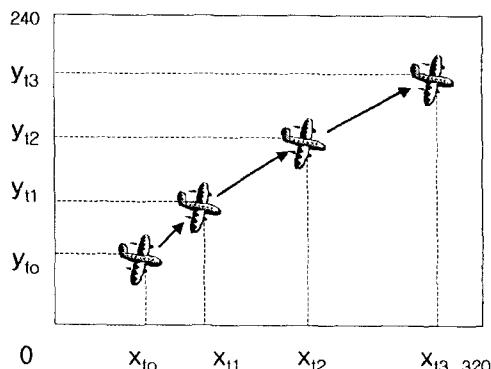
$$X_{t_2} \text{에서의 속도} = X_{t_2} - X_{t_1} \quad (4)$$

$$X_{t_2} \text{에서의 가속도}$$

$$\begin{aligned} &= X_{t_2} \text{에서의 속도} - X_{t_1} \text{에서의 속도} \\ &= X_{t_2} - 2X_{t_1} + X_{t_0} \end{aligned} \quad (5)$$

$$X_{t_3} \text{의 예상위치}$$

$$\begin{aligned} &= X_{t_2} \text{에서의 속도} + X_{t_2} \text{에서의 가속도} \\ &= 2X_{t_2} - 3X_{t_1} + X_{t_0} \end{aligned} \quad (6)$$



(그림 10) XY 평면 위에서 이동지점 예측

이 예는 단지 X축 위에서만의 움직임만을 고려한 것이지만, 그림 10처럼 XY 축으로 구성된 2차원 공간에 있어서도 같은 방법으로 예상 이동 지점을 알아낼 수 있다. 더불어 이와 같은 방법은 각 차원의 스칼라 값의 조합에 의해 위치를 알아내므로 물체의 이동 각도의 변화율을 따로 계산하지 않고도 위치를 예측할 수 있다.

색상 값 계산

카메라 영상을 통해 움직임을 감지하고 추적하는 작업은 실세계를 있는 그대로 받아들이기 때문에 수많은 변수와 여러 가능성에 노출되어 있다. 때문에 움직임에만 전적으로 의존하는 시스템은 물체가 갑자기 멈추거나 또는 비슷한 움직임을 보이는 물체가 있을 경우에 시스템은 오류를 일으킬 수밖에 없다. 따라서 움직임 외에 색상 값이나 물체의 형태를 특정 값으로 계산하여 추적을 보정하는 작업이 필요하다.

본 시스템에서는 물체의 회전이나 변형에도 쉽게 적용할 수 있게 물체의 색상을 특징으로 사용하였다. 색으로 물체를 판별하되 색의 비

율을 계산함으로써 보다 정교한 추적이 가능하게 하였다. 색 분석을 위해 삼원색인 R(Red, 빨강), G(Green, 녹색), B(Blue, 파랑)로 구성된 RGB 컬러공간(color space)을 사용하였는데, CIE (the Commission Internationale de l'Eclairage)가 규정한 색상분류 공식에 따르면 RGB는 식 (7)로 표현될 수 있다. 이러한 값을 조합하면 명암과 상관없이 색상만을 추출해 낼 수 있으므로 본 논문에서는 물체의 색상을 추출하는데 명암의 영향을 받지 않고 색상만으로 특징을 추출하기 위해 이와 같은 색상 규정을 사용하였다.

$$\begin{aligned} r &= \frac{R}{R+G+B}, \quad g = \frac{G}{R+G+B}, \\ b &= \frac{B}{R+G+B} \end{aligned} \quad (7)$$

제안한 시스템은 무한대에 가까운 색상의 종류를 표 1과 같이 특정화 된 영역으로 분류함으로써 보다 용이한 특징추출이 가능하도록 하였다. 색상이 3가지 요소인 것을 감안하여 각각 33%를 기본 단위 r , g , b 의 상대적인 비율을 계산해서 색상을 임의의 총 9가지 색상으로 분류했다. 이런 방식으로 각 블록내의 화소들의 색상을 각각 분류하고 가장 많은 수의 색상 값을 블록의 대표 색상 값으로 정했다. 그리고 움직임 영역 내에서 블록 색상 값들의 비율을 계산한다. 예를 들어 총 10개의 블록으로 이루어진 움직임 영역이 있을 경우 5개의 블록의 색상이 hue.1이고 3개의 블록이 색상이 hue.2이고 나머지 2개의 블록이 색상이 hue.3일 경우 이 움직임 영역의 색상비는 hue.1이 50%, hue.2가 30% 그리고 hue.3가 20%라는 결론이 나오게 되고 물체의 제 1색상은 hue.1이 되고 제 2색상은 hue.2이 되며 제 3색상은 hue.3이 된다. 또한 이들 색상간의 분포비율

<표 1> 전체 색상 분류

색상	색비율 $\frac{R}{R+G+B}$	$\frac{G}{R+G+B}$	$\frac{B}{R+G+B}$
hue.1	1	0	0
hue.2	0.66	0.33	0
hue.3	0.33	0.66	0
hue.4	0	1	0
hue.5	0	0.66	0.33
hue.6	0	0.33	0.66
hue.7	0	0	1
hue.8	0.33	0	0.66
hue.9	0.66	0	0.33

'5 : 3 : 2'는 물체의 색상코드가 된다. 이는 서로 비슷한 색상을 가진 물체들 간에도 색이 차지하고 있는 비율로써 물체를 좀 더 정교하게 구분하는 역할을 해준다. 이러한 색상비에 의한 특징을 사용하게 되면 물체의 모양이 변하고 회전하거나 확대에 많은 약점을 보이는 기존의 블록 매칭 방법과는 다르게 물체의 색상비를 서로 비교해서 유사정도를 판별하게 되므로 보다 강건한 추적을 가능하게 한다.

실험결과 및 분석

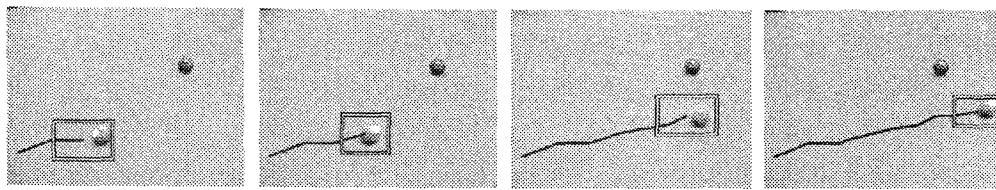
실험장비로 2대의 SONY EVI-D30 카메라와 View Cast Osprey-100 영상캡쳐보드를 사용하였으며 실험영상은 약 1/20초의 프레임 속도로 320×240 크기로 받아들였다. 실험은 크게 2부류로 나누어 수행되었는데, 그 첫 번째가 팬-틸트-줌 카메라를 장착하지 않은 상태에서 단순한 배경에서의 이동물체 추적성능 실험을

통해 기본적으로 시스템이 올바른 추적을 수행하는지 알아보았고, 두 번째로 팬-틸트-줌 카메라를 장착한 상태에서 복잡한 배경이 존재하는 실세계에서의 이동물체 추적실험을 통해 복잡한 실세계에서도 제대로 추적을 수행하는지 알아보았다.

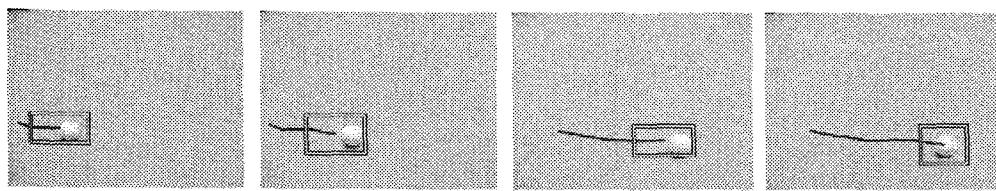
단순한 배경하의 이동물체에 대한 추적실험

먼저 팬-틸트-줌 카메라를 장착하지 않은 상태에서 시스템이 올바른 추적을 수행하는지 알아보기 위해 잡영이 비교적 적은 균일한 배경에서 빠르게 이동하는 물체에 대한 추적실험을 수행하였다. 이를 위해 당구대 위에서 빠르게 움직이는 단일 당구공 추적실험과, 서로 독립적인 방향으로 이동하는 다수의 당구공 추적실험을 수행하였다.

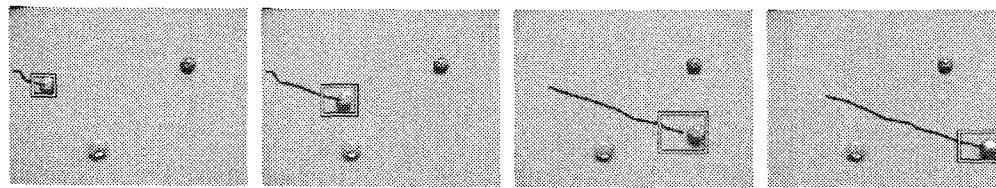
이동하는 물체 1개를 추적한 실험결과가 그림 11~그림 13에 나타나있다. 그림 11은 좌에서 우로 빠르게 이동하는 하나의 당구공을 추



(그림 11) 유사한 형태의 정지물체가 1개 있을 경우 1개의 이동물체 추적



(그림 12) 1개의 이동물체 추적

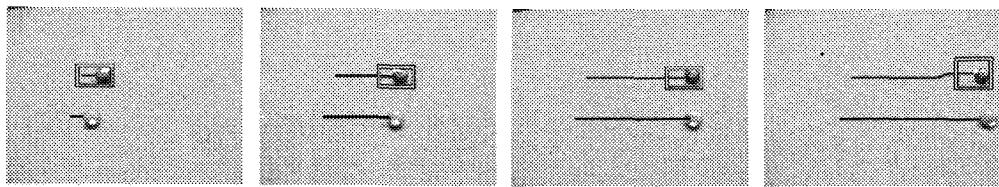


(그림 13) 유사한 형태의 정지물체가 2개 있을 경우 1개의 이동물체 추적

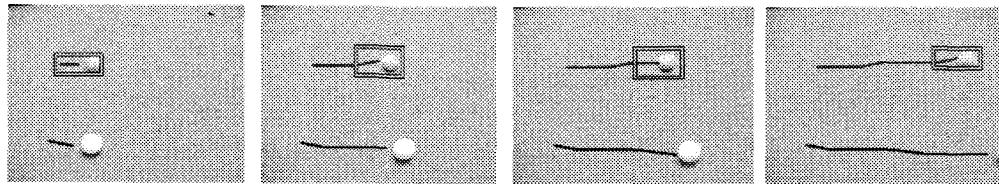
적한 것으로, 붉은색 라인은 공의 이동경로를 나타낸 것이다. 공 주위의 사각형은 움직임을 영역을 표시하는 MAR로써 움직임 영역을 나타내며 완성된 시스템에서 펜-틸트-카메라의 제어를 유도시키는 지점이다. 실험결과 단일 이동물체에 대한 본 시스템은 오류 없이 정확한 추적이 가능함을 알 수 있었다. 그림 12는 움직이지 않는 공을 하나 추가함으로써 모양이 비슷한 물체가 이동경로에 있을 때 추적에 방해받는지 여부를 테스트 한 것으로써 결과에서 보는 것처럼 단일 이동물체에 대한 추적과 같이 올바른 추적을 수행함을 알 수 있다. 또한 그림 13은 움직이지 않는 공 하나를 더 추가해서 방해물을 2개 배치함으로써 좀 더

복잡한 상황을 만들어서 실험한 결과이다. 본 시스템은 일차적으로 움직임을 체크한 후 추적을 시행하기 때문에 방해물의 수와는 상관없이 움직이지 않는 물체는 일단 추적 대상으로 인식하지 않음을 알 수 있다.

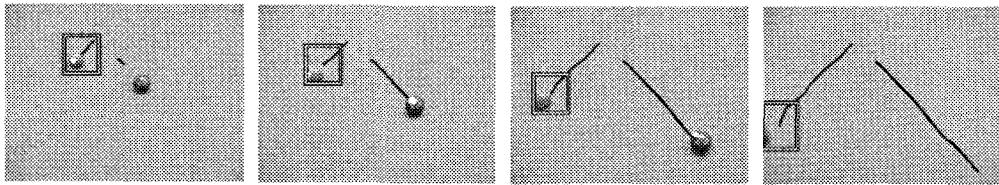
다수의 움직이는 물체를 추적한 실험결과는 그림 14~그림 16에 나타나있다. 그림 14는 같은 방향으로 빠르게 이동하는 2개의 공을 추적한 것으로써 붉은색 추적선을 그리는 공은 시스템이 제일 먼저 발견한 이동물체이며 파란색 추적선을 그리고 있는 공은 2번째로 발견한 이동물체이다. 이동물체마다 다른색의 추적선을 부여하는 이유는 다수의 이동물체가 존재할 경우 각 물체별로 올바른 추적 여부를



(그림 14) 같은 방향으로 이동하는 물체 2개 추적



(그림 15) 유사한 방향으로 이동하는 물체 2개 추적



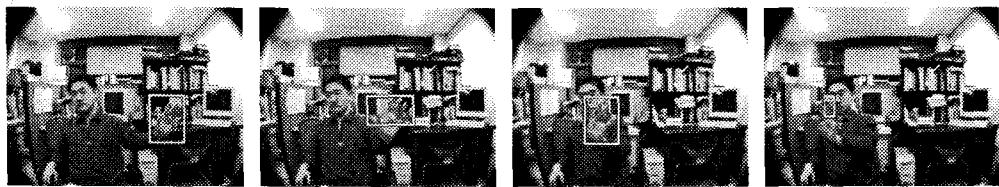
(그림 16) 서로 다른 방향으로 이동하는 물체 2개 추적

확인하기 위해서이다. 붉은색 사각형의 MAR가 녹색공에만 존재하는 이유는 본 시스템은 디폴트 상황에서 제일 먼저 발견된 하나의 이동물체만을 팬-틸트-줌 카메라로 하여금 추적시키기 때문으로써 이는 시스템의 사용목적에 맞추어 재조정이 가능하다. 그림 14와 유사하지만 이동경로와 속도 등을 약간 다르게 해서 추적한 것이다. 결과에서 알 수 있듯이 정확한 추적을 시행함을 알 수 있다. 그림 16은 속도와 방향이 전혀 다른 2개의 공을 동시에 추적하게 하는 것으로써 실세계와 가장 가까운 복잡한 움직임에 대한 실험이다. 결과에서 알 수 있듯이 예측하기 어려운 상황에서도 정확한 추적 결과를 보여주고 있으며

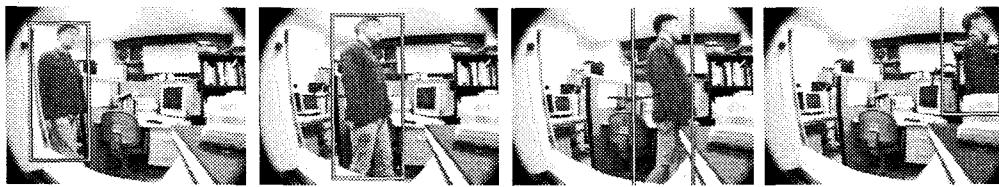
동시에 각도의 변화율을 따로 계산하지 않고도 이동 거리에 대한 스칼라 값만으로 충분히 추적이 가능함을 보여주었다.

복잡한 배경이 존재하는 실세계에서의 추적실험

팬-틸트-줌 카메라를 부착한 상태에서 복잡한 배경이 존재하는 실세계에서 좌측에서 우측으로 이동하는 보행자를 추적한 결과가 그림 17과 그림 18에 나타나 있다. 그림 17은 광각렌즈를 장착한 카메라로부터 전송된 영상을 추적한 결과로 보행자 주위의 흰색 사각형은 MAR를 나타내며 붉은색 선은 이동경로를 나타낸다. 그림 18은 그림 17의 영상정보를 바



(그림 17) 광각렌즈 장착 카메라에 의한 손 추적

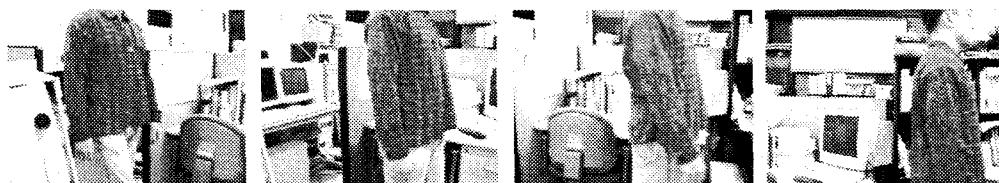


(그림 18) 광각 렌즈 장착 카메라의 보행자 추적

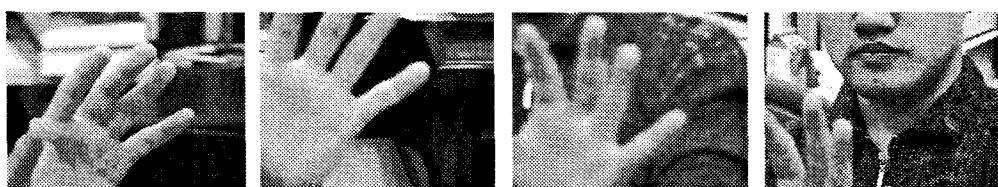
탕으로 시스템의 펜-틸트-줌 카메라를 제어시킨 영상이다. 그림 18의 영상은 좌측부터 우측까지 시간상으로 그림 17과 일치하며, 그림 17의 영상에 비해 보행자가 비교적 화면 중앙에 위치하는 것을 알 수 있는데, 이것은 시스템이 움직이는 물체의 중심좌표를 알아낸 후 이 곳으로 펜-틸트-줌 카메라의 방향을 일치시켰기 때문이다. 결과에서 알 수 있듯이 복잡한

배경에서도 올바른 추적 결과를 보여주었다.

그림 19와 그림 20은 동일한 시스템으로 움직이는 손을 추적한 결과이다. 그림 20은 보행자를 추적한 그림 18에 비해 줌 효과가 강하게 나타나는 것을 알 수 있는데 이것은 제안하는 시스템이 움직이는 대상의 크기 맞추어 줌인을 함으로써 목표대상에 대한 정보를 가능한 많이 얻으려 하기 때문이다. 보는 바



(그림 19) 펜-틸트-줌 카메라에 의한 보행자 추적



(그림 20) 펜-틸트-줌 카메라에 의한 손 추적

와 같이 손의 움직임에 따라 정확한 추적을 하는 것을 알 수 있다.

결 론

본 논문에서는 인간의 생물학적 메커니즘과 유사한 이동물체 추적시스템을 만들기 위해 광각렌즈를 장착한 카메라와 펜-틸트-줌 카메라를 이용한 추적시스템을 제안하였다.

추적시스템의 특성상 아무리 느리게 움직이는 물체라 할지라도 카메라로부터 수 만개에 달하는 화소를 입력받아 복잡한 연산과정 후에 그 물체가 움직인 위치까지 펜-틸트-줌 카메라를 향하게 만드는 데까지는 많은 시간이 걸리기 마련이다. 그 결과 물체가 이미 지나 갔음에도 불구하고 이미 지나버린 곳을 가리키고 있는 경우가 발생할 수 있다. 따라서 시스템의 속도는 그 무엇보다 중요하기 때문에 연산을 절감할 수 있는 부분을 최대한 생략해 주는 것이 시스템 설계의 관건이라고 하겠다.

기존에 연구되고 있는 추적시스템에 대한 여러 방면에서의 고찰을 통해 기존의 시스템들은 지나치게 많은 영상정보를 실시간으로 처리하려고 함으로 인해 시스템의 성능이 저하되고, 더불어 많은 오류를 가지고 있다는 문제점을 발견할 수 있었다. 그 대표적인 예로써 현재 영상압축과 물체 추적분야에서 가장 널리 사용되고 있는 블록정합 방법은 부분 영상을 단순히 비교함으로써 물체의 동일여부를 알아내는 방법이기 때문에 빠르고 변칙적으로 이동하는 물체와 변형되는 물체에 대해서는 많은 오류를 보여 온 것이 사실이다.

본 연구에서는 이러한 블록정합 방법에 대

한 깊이 있는 연구를 통해서 이 방식에 대한 근본적인 문제점을 파악하였고 그와 동시에 많은 부분에서 연산을 절감할 수 있음을 수 있는 부분을 알아낼 수 있었다. 따라서 제안하는 시스템은 이에 대한 대응책으로 전체 영상을 블록화 시켜서 수많은 화소를 한꺼번에 처리하였으며 물체의 가속도와 색상을 이용하여 부족한 추적성능을 보완하였다. 이는 가장 많이 사용되는 추적 방법인 블록정합 방법의 단점인 빠르게 이동하는 물체나 변형되는 물체를 찾아내기 어려운 문제점을 해결한 것으로써 물체 추적방법에 있어서 새로운 접근 방법이라고 할 수 있으며 실험을 통해 속도와 정확도에 있어서 빠르게 이동하는 물체를 빠른 시간 내 올바르게 추적함을 확인할 수 있었다.

참고문헌

- Lindsay, P., and Norman, D. (1972), Appendix B: Operating characteristics, Human Information Processing, New York: Academic Press, 715-732.
- Munakata, Y., Santos, L., Spelke, E., Hauser, M., and O'Reilly, R. (2001), Visual representation in the wild: How rhesus monkeys parse objects, Journal of Cognitive Neuroscience, 13, 44-58.
- 김학수, 임원택, 이재철, 이원규, 박규택 (1997), “다중 선형 회귀모델을 이용한 움직임 추정방법”, 대한전자공학회 논문지, 34-10, 98-103
- 이원규, 김영호, 이제구, 박규택 (1995), 무인감

- 시장치 구현을 위한 단일 이동 물체 추적 알고리즘, 대한전자공학회 논문지, 32-11, 1405-1416
- Hongo, H., Ohya, M., Yasumoto, M., Niwa, Y., and Yamamoto, K. (2000), Focus of attention for face and hand gesture recognition using multiple cameras, Fourth IEEE International Conference on Automatic Face and Gesture Recognition, 156-161
- Kanade, T., Collins, R., Lipton, A., Anandan, Burt, and Wikson (1997), Cooperative Multi-Sensor Video Surveillance, DARPA Image Understanding Workshop(IUW), 3-10
- Lipton, A., Fujiyoshi, H., and Patil (1998), Moving target classification and tracking from real-time video, IEEE Workshop on Application of Computer vision(WACV), pp 8-14
- Collins, R., Lipton, A., Kanade, T., Fujiyoshi, H., Duggins, D., Tsin, Y., Tolliver, D., Enomoto, N., and Hasegawa, O. (2000), A System for Video Surveillance and Monitoring (VSAM), tech. report CMU-RI-TR-00-12, Robotics Institute, Carnegie Mellon University
- Murat, A. (1995), Digital Video processing, Prentice-Hall
- Gharavi, H. and Mills, M. (1990), Block-matching motion estimation algorithms : New results, IEEE Transactions on Circuit and System, 37, 649-651

1 차원고접수: 2005. 6. 5

2 차원고접수: 2005. 11. 27

최종재승인: 2005. 11. 28