실시간영상에서 가변탐색영역을 이용한 객체추적알고리즘

민병묵* , 이광형**, 오해석***
*숭실대학교 컴퓨터공학과,**서일대학 인터넷정보과
***경원대학교 소프트웨어 대학

Object Tracking using variable Search Block on Realtime Image

Byoung-Muk Min* , Kwang-Hyoung Lee**, Hae-Seok Oh**
*Dept. of Computing, Soongsil University
**Dept. of Internet Information, Seoil College
***College of Software, KyungWon University

요 약

카메라를 통하여 실시간으로 입력되는 객체의 움직임은 잡음이나 조명의 변화에 따라 정확하게 추출하고 추적하는 것이 어렵다. 따라서 실시간으로 입력되는 영상에서 객체를 추출하고 움직임을 추적하기 위해서는 고속탐색 알고리즘이 필요하다. 본 논문은 실시간영상에서 객체의 움직임을 추출하고 추적을 위하여 배경영상의 변화에 강인한 배경영상 갱신 방법과 가변적인 탐색영역을 이용한 객체추적의 빠른 알고리즘을 제안한다. 배경영상 갱신 방법은 임계값이 실험적 기준치 보다 작은 경우에는 배경영상을 갱신하고, 큰 경우에는 객체가 유입된 시점으로 판단하여 픽셀검사를 통해 객체의 윤곽점을 추출한다. 추출된 윤곽점은 객체 영역블록의 생성과 일정한 거리를 유지하는 탐색블록을 생성하여 정확하고 빠른 객체의 움직임을 추적한다. 실험결과, 제안한 방법은 95% 이상의 높은 정확도를 보였다.

1. 서 론

인터넷 시대에 접어들면서 웹 카메라를 이용한 보안 시스템의 개발이 활발하다. 웹 멀티미디어 보안시스템은 활용영역이 확대되면서 영상의 해상도와 전송속도, 그리고 보안시스템의 핵심인 객체영역 인식, 영상 정보의 처리, 저장, 검색 기술, 추적 등의 연구가 추가적으로 요구 되고 있다[1].

기존의 실시간 객체 추적 시스템의 구현에는 많은 어려움이 존재한다. 우선 강인한 실시간 객체 추적 시스템을 구현하기 위하여 고가의 장비를 요구한다. 또한 객체 추출이나 추적에 많은 연산을 필요로 하는 알고리즘 등을 적용하여 객체 추출에는 뛰어난 성능을 보이지만 실시간 객체 추적에는 부적합하다. 또한 픽셀 변화만을 사용하기 때문에 객체의 움직임이 없어도 조명의 변화에 따라 쉽게 객체로 오인하기 쉽다.

* 본 연구는 서울시 산학연 협력사업으로 구축된 서울 미래형 콘텐츠 컨버전스 클러스터 지원으로 수행되었습니다.

동영상 압축 방식에서 비디오 신호의 시간상 중복성을 제거하고 동영상을 압축하기 위해 움직임 보상 방식이 요구되고 있다. 움직임 보상방식(Motion Compensation)은 비디오 영상에서 배경과 객체를 구분하여 전송하기 위한 방법으로 움직임 벡터(Motion Vector)를 이용하여 객체의 움직임이나 추적을 예측한다. 움직임 벡터는 실시간으로 처리해야 하기 때문에 계산량을 현저히 줄일 수 있는 알고리즘을 필요로 한다[2].

실시간 영상에서 배경영상과 입력영상을 구분하여 움직인 객체를 추출하거나 추적하기 위해 차영상을 이용한 방법, 블록정합기법, 배경영상을 이용한 방법 등을 이용한 연구가 이루어지고 있다[3].

블록정합기법은 현재프레임 탐색영역 안에서 이전 프레임의 지정된 블록과 가장 유사한 블록을 찾는 방법으로 블록영역에서 객체의 추적이나 객체의 영역을 지정할 수 있다. 그러나 블록영역 밖에서 유입되는 객체는 측정이 불가능하다.

본 논문에서는 실시간 영상에서 초기의 배경영상을 기준으로 입력영상과의 차를 구하고 시간에 따라

변화하는 배경영상을 N×M 픽셀 마스크만큼 교체하여 갱신 한다. 이미지 픽셀 검사는 모든 픽셀을 연산에 참여시키지 않고 일정한 간격을 두고 이미지의 픽셀을 검색하여 효과적으로 객체의 윤곽점을 검출한다. 객체의 윤곽점 좌표는 가변적인 탐색블록을 생성하여 정확하고 빠르게 객체를 추출하고 추적 할 수 있도록 하였다.

본 논문의구성은 2장에서 기존의 움직임 검출방법을 분석하고, 3장에서는 배경영상 갱신과 객체의 윤곽점 추출방법 및 객체의 최소블록 및 탐색블록 이용한 객체의 움직임 추적방법을 제안한다. 4장에서는 제안한 방법으로 실험한 결과를 기술하고, 마지막으로 5장에서는 결론을 기술한다.

2. 객체 추출 및 추적 방법

객체의 움직임 정보는 프레임 단위의 움직임과 각 화소에서의 움직임으로 구분된다. 프레임 움직임은 그 움직임을 추정하여 움직임 벡터를 전송한 다음, 프레임의 움직임 보상을 행한다. 그러나 각 화소에서의 움직임 정보는 프레임 움직임만큼 보상된 이전 프레임과 현재 프레임과의 차이가 큰지 작은지를 나타내는 움직임 검출 정보로서 표현된다.

2.1 차영상 기법

객체 추출을 위한 후보영역을 검출하기 위해 연속된 두 프레임간의 차영상 분석 기법을 사용한다. 그림 1의 차영상 분석 기법은 연속된 두 프레임간의 밝기 차이를 구한 후, 임계값을 사용하여 임계값 보다 낮은 밝기 차이를 가진 부분은 객체가 없는 배경으로 구별하고 임계값 보다 큰 밝기 차이를 가진 부분은 객체의 유입이 있다고 판단한다[4].

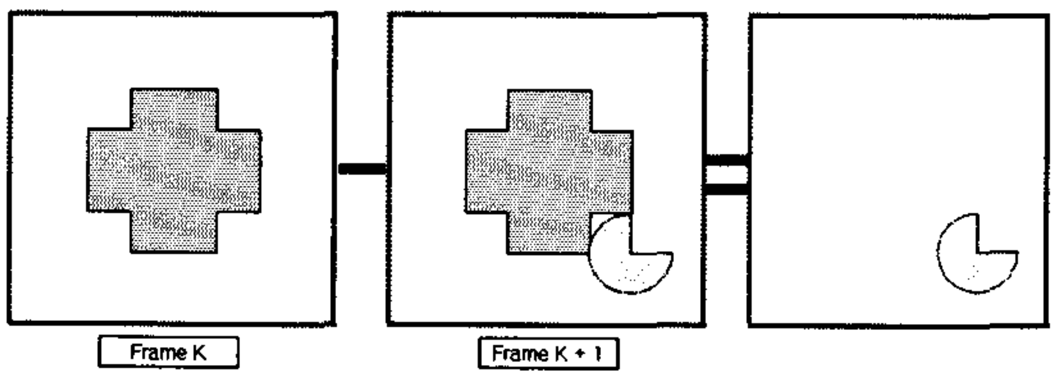

그림 1. 차영상기법

2.2 블록정합 기법

블록정합 기법은 현재 프레임 탐색영역 안에서 이전 프레임의 지정된 블록과 가장 유사한 블록을 찾는 방법으로 그림 2와 같이 객체가 움직이지 않다가 다시 움직이는 경우에도 추적이 가능 하고 블록의 크기와 추적할 객체를 지정할 수 있다[5,6].

블록정합기법에는 전역탐색 알고리즘과 계층적 블록알고리즘이 흔히 사용된다. 전역탐색 알고리즘은 영상의 밝기값 분포가 비교적 균일한 영역이 없는 곳에서 사용된다. 밝기값 분포가 균일한 영역에서는 부정확한 적합가능성이 발생될 수 있다.

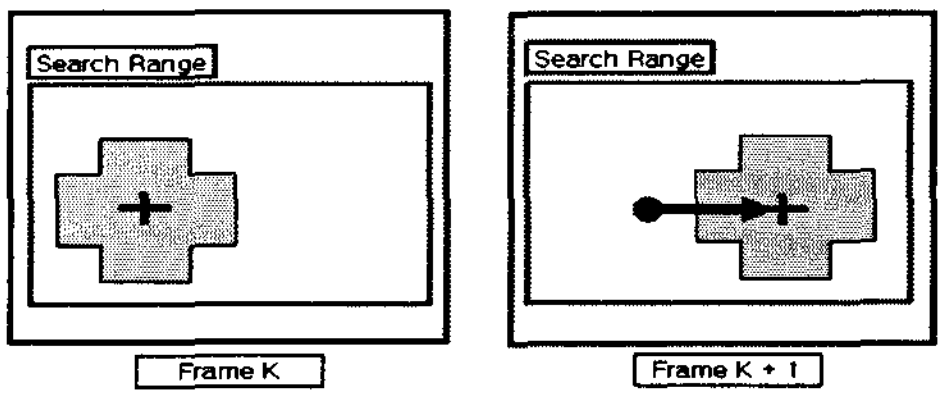

그림 2. 블록정합기법

2.3 배경영상 교체기법

배경영상 교체기법은 현재 프레임과 기준이 되는 배경 영상의 차이를 구하는 방법이다. 차영상기법과 같이 인접한 두 프레임을 비교하는 것이 아니라 (그림 3)과 같이 이전 프레임들로부터 배경이 되는 영상을 추출하고 이 영상과 현재 프레임을 비교하여 모션을 추출하게 된다[6,7,8]. 매번 프레임을 검사하면서 기준이 되는 배경 영상은 오래 전 프레임의 영향을 줄이고 현재 프레임의 영향을 추가시키는 방법으로 특정한 방법에 따라 계속 수정된다. 이 방법에서 많이 사용되는 것으로 시간적 평활법과 시간적 중간치법으로 들 수 있다.

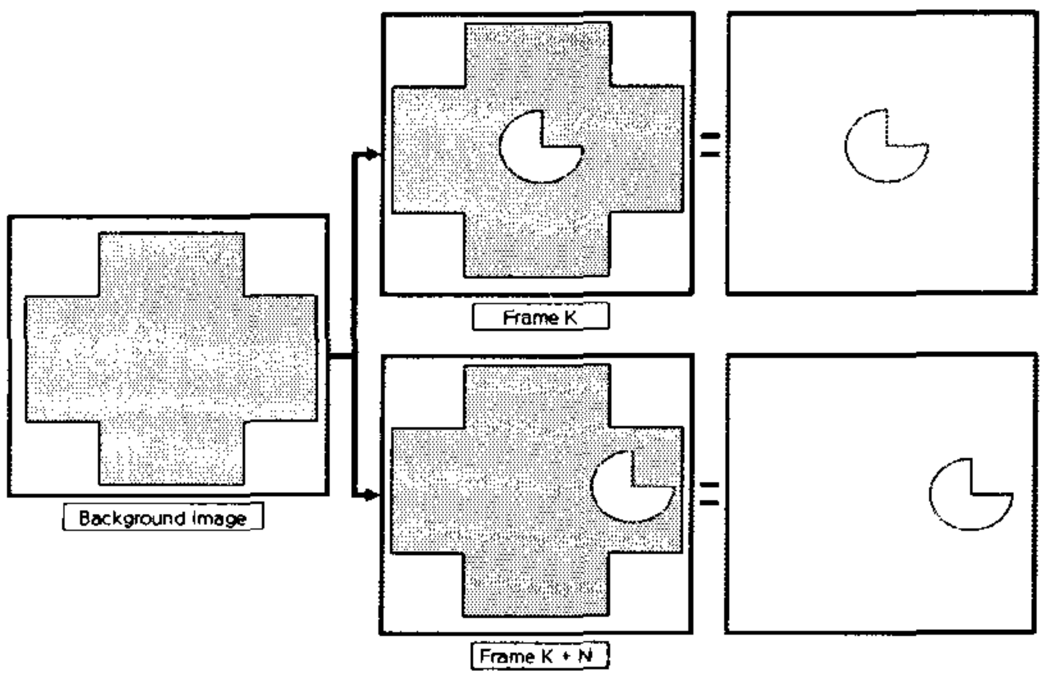

그림 3. 배경영상 교체기법

3. 실시간 객체 추출 및 추적

3.1 제안 방법

제안하는 객체 추출 및 추적 방법은 카메라로부터 획득되는 초기 영상 이미지를 배경영상 이미지로 선택하며 이 이미지는 객체의 유입이 없는 상태에서 획득된 순수한 배경이미지이다. 배경영상과 카메라로부터 입력되는 영상과의 차를 이용하여, 임계값이 존재하면 이미지 검사를 통하여 움직인 객체의 윤곽선에 해당하는 부분을 찾고 움직임 검출여부를 판단하여 추적하게 된다. 초기의 배경영상은 시간이 지남에 따라 조명효과나 잡음에 의해서 이미지 자체가 변화하게 된다. 따라서 초기의 배경영상만으로 입력영상과의 차를 구하게 되면 객체의 유입이나 움직임이 없어도 움직임이 있는 것처럼 인식하게 되므로

정확한 객체 추적을 위해서 배경영상을 갱신한다. 빠른 객체 추적을 위해서 객체의 영역을 추출하고 움직임을 측정하기 위한 탐색영역을 생성하여 매번 전체 이미지의 픽셀을 스캔하지 않고도 정확한 추적을 할 수 있도록 하였다.

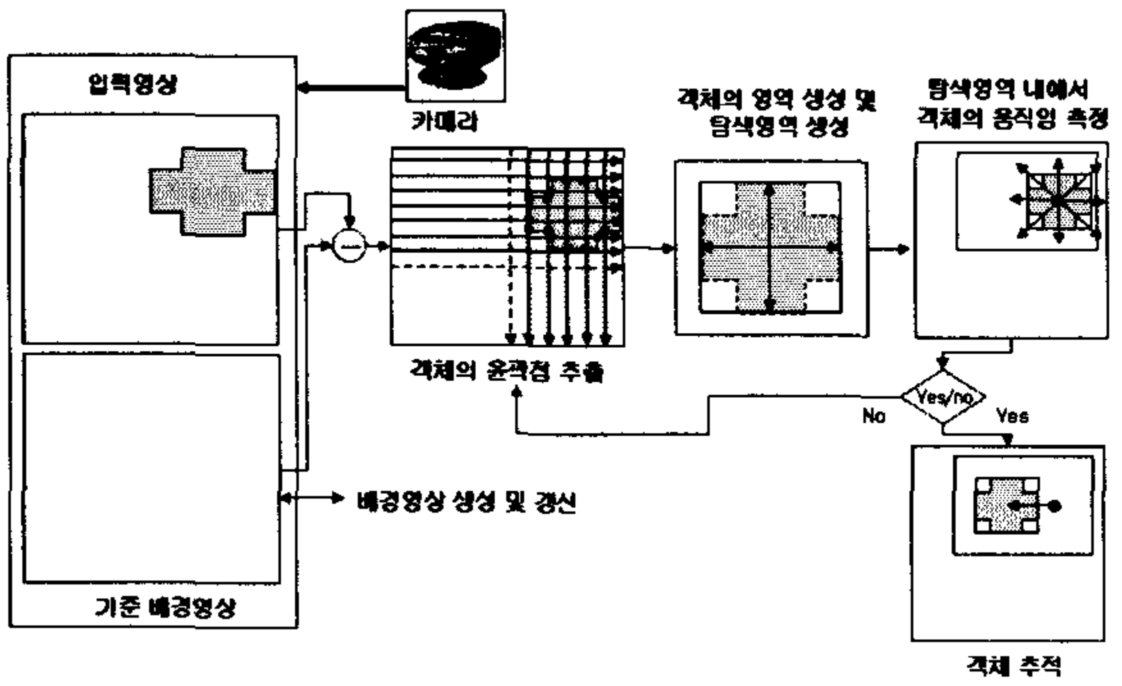

그림 4. 제안하는 객체 추출 및 추적 시스템

그림 4는 제안하는 객체추출 및 추적시스템을 나타내고 있으며, 그림 5에서는 본 논문에서 제안하는 움직임 검출 과정을 나타내고 있다.

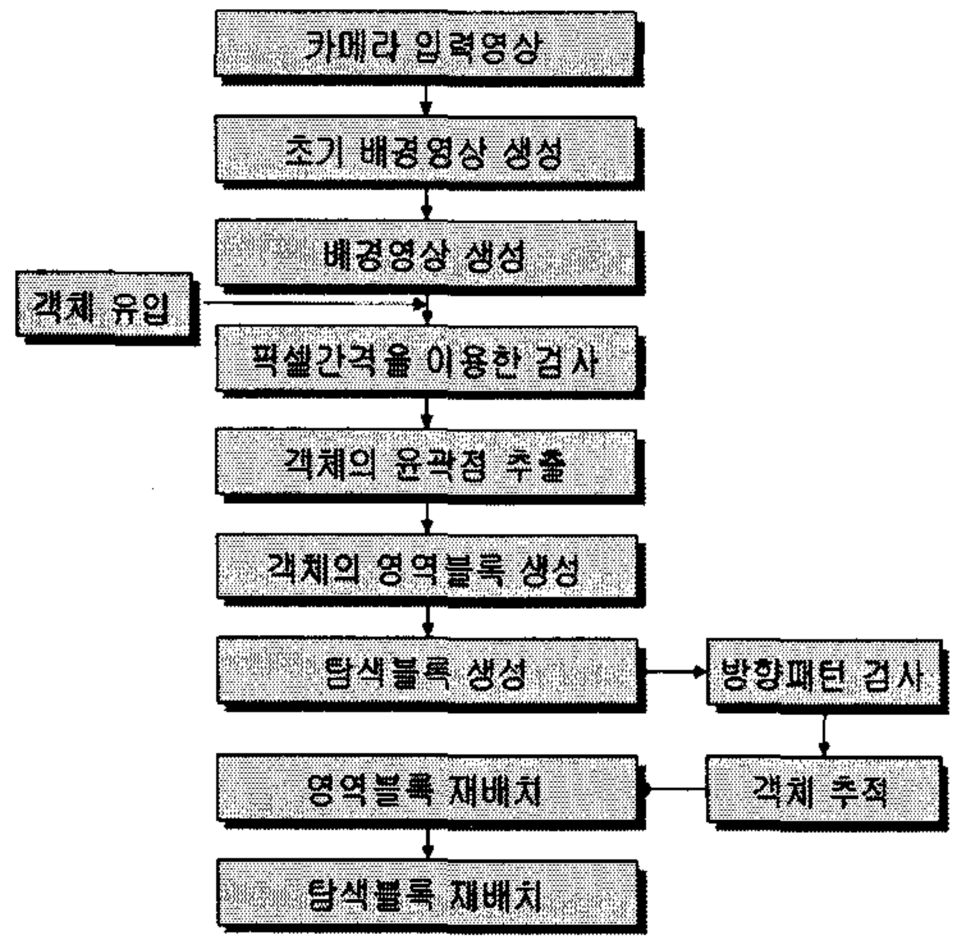

그림 5. 객체 추출 및 추적 과정

3.2 배경영상 획득

카메라의 입력 영상에 의해 얻어지는 이미지는 초기에 배경영상을 얻기 위해 물체의 변화나 이미지의 변화가 없는 것으로 간주한다. 배경영상은 카메라로부터 입력받은 후부터 시간이 지남에 따라 객체가 검출되지 않은 상황이더라도 배경영상 자체가 변하게 된다. 본 논문에서는 배경영상의 잡음을 제거하여, 보다 정확한 움직임 검출을 위해서 배경영상을 갱신하는 방법을 사용한다. 전체적인 배경영상의 갱신은 많은 연산량을 필요로 하고 객체의 움직임을 검출하는데 시간이 많이 소요되므로 N × M 마스크를 사용하여 계속적으로 배경영상을 갱신하면서 객체의 유입을 검사한다.

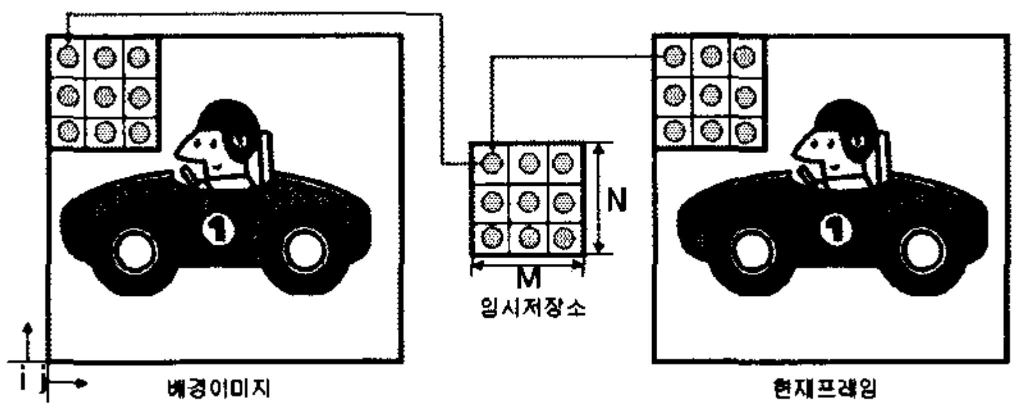

그림 6. 배경이미지 갱신

그림 6은 입력영상 "Frame t"의 N × M 마스크 내에서 RGB 채널의 값과 배경영상 "Background Image"와의 차이값이 임계값 α 보다 작으면 배경영상이 변경된 것으로 간주하여 해당 픽셀을 입력영상의 픽셀로 변경한다.

3.3. 객체의 윤곽점 추출

카메라를 통하여 입력되는 영상이미지는 배경영상의 RGB 채널과 비교하여 수시로 배경영상을 갱신하거나 객체의 유입을 검사한다. 즉, 배경영상 갱신 중 임계값 α 보다 큰 경우가 객체가 유입된 시점이다. 객체가 유입된 영상이미지는 픽셀검사를 통하여 객체의 위치를 검출하고 윤곽점을 추출하게 된다. 영상이미지 전체를 픽셀단위로 검색을 한다면 실시간으로 추출되는 이미지를 모두 검사하게 되므로 많은 연산량을 요구하게 된다.

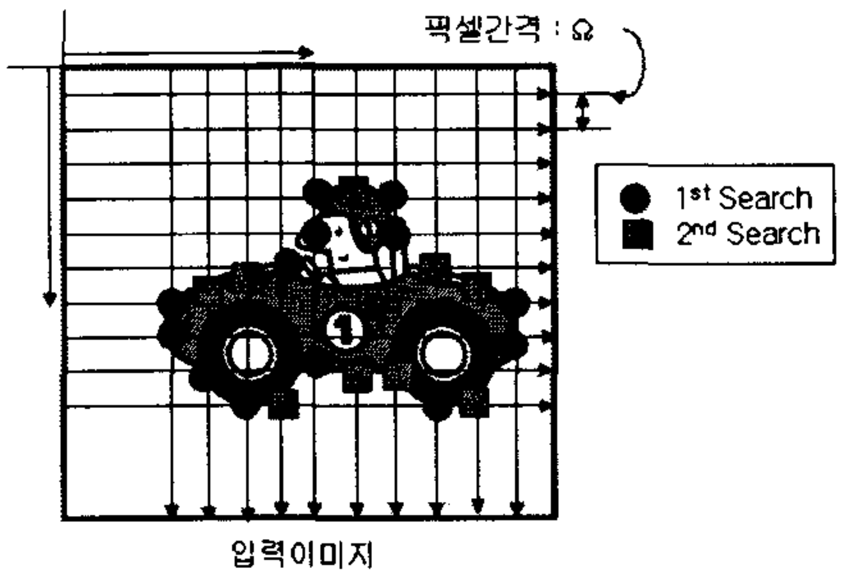

그림 7. 객체의 윤곽점 추출

3.4. 객체의 영역블록 생성

픽셀간격을 통하여 추출한 객체의 위치 좌표를 이용하여 객체의 영역블록을 설정한다. (그림 9)는 객체의 영역을 설정하는 방법이다. 1차 검사 후 추출된 객체의 윤곽점 $x_1, x_2, \ldots, x_n$ 각각의 좌표값을 저장하고, 동일한 스캔라인에 위치한 좌표점과의 거리를 측정하여 객체의 최소영역에 대한 최대 가로길이 값을 추출한다. 그리고 2차 검사 후 추출된 윤곽점인 $y_1, y_2, \ldots, y_n$ 의 좌표점을 계산하여 최대 세로길이 값을 추출한 다음 객체의 최소블록을 생성한다.

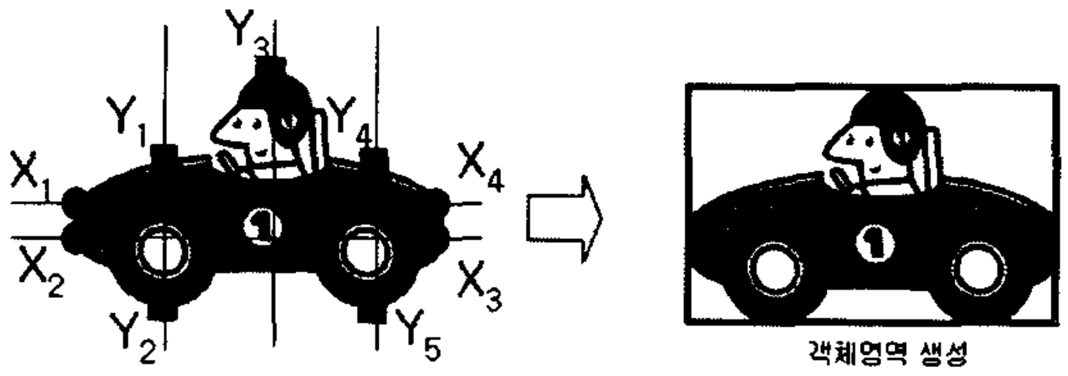

그림 8. 객체의 영역블록 생성

3.5. 탐색블록 생성 및 객체 추적

객체의 윤곽점 추출을 통하여 생성된 영역블록은 카메라 입력영상 이미지와 현재 설정된 최소블록 내의 객체 움직임을 빠르게 추적하기 위해서 탐색블록을 설정한다.

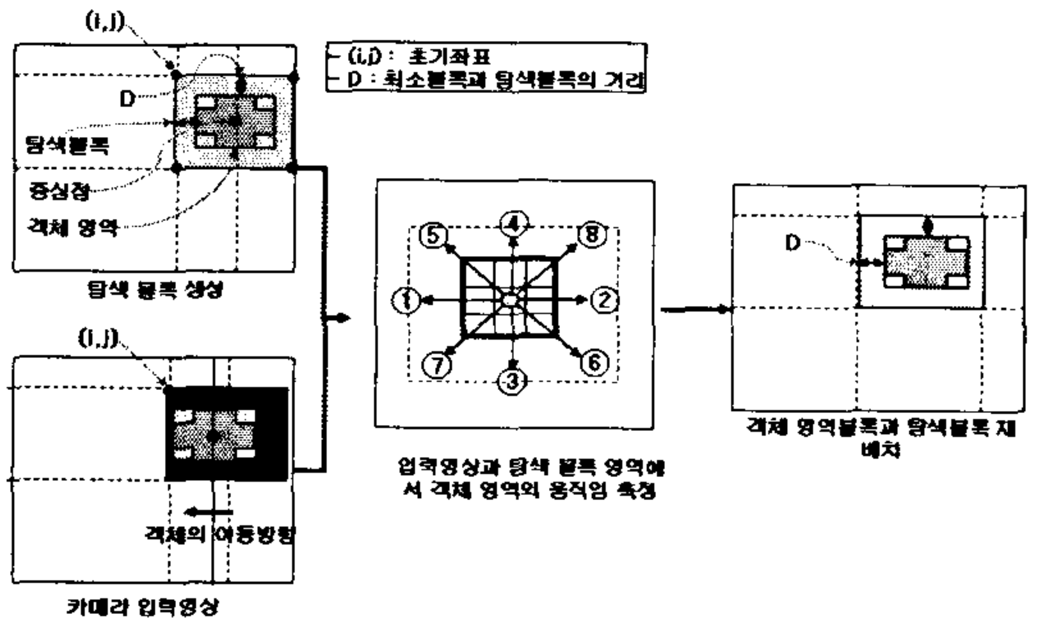

그림 9. 탐색블록의 생성과 객체 추적

4. 실험 결과 및 분석

제안하는 객체 추적 방법을 사용하여 구현한 시스템은 그림 10 과 같으며, 초기 카메라에서 입력되는 영상은 초당 15프레임으로 설정하여 사용하였다.

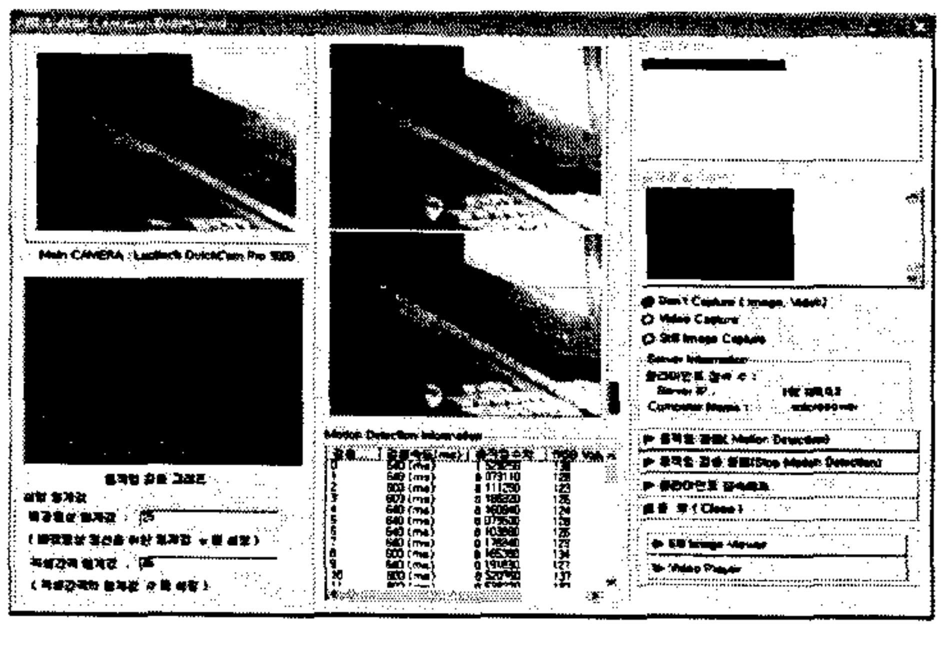

그림 10. 제안 시스템의 메인화면

<표 1>은 제안한 방법에 의한 임계값들과 변수에 대한 최적화 실험 결과이다. 배경영상 생성 임계값은 실험값이 5인 0.210일 때, 배경영상 이미지내의 잡음을 제거 할 수 있었다. 그리고 픽셀 간격 임계값은 조밀할수록 움직임 추출시 객체 인식이 잘 이루어 졌지만, 많은 연산량을 요구하기 때문에 보다 빠른 탐색을 위해 실험값 '5'를 선택하여 실험하였다.

표 1. 임계값과 변수에 대한 최적화

임계값 \ 실험값	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	…
Ω	0.951	0.903	…	0.851	…	0.595	…	0.496	…

탐색블록과 객체의 영역블록을 설정하기 위한 임계값 D 의 실험적 최적값은 <표 2>에서와 같이 "15 pixel"의 중간치 값을 사용하여 실험 하였다.

표 2. 탐색블록의 위치 거리(D)

실험값 (pixel)	검출속도(ms)	오류검출
5	210	0.358
10	251	0.310
15	273	0.215
20	350	0.198
25	399	0.181

5. 결 론

본 논문에서는 초기의 배경영상을 기준으로 입력영상과의 차를 구하고 시간에 따라 변화하는 배경영상을 N×M 픽셀 마스크만큼 교체하여 배경영상을 갱신하였다. 실험결과, 기존의 배경영상기법에서 배경영상 자체의 변화 때문에 생기는 문제점을 해결하였고, 픽셀 검사는 모든 픽셀을 연산에 참여시키지 않고 일정한 간격을 두고 이미지의 픽셀을 검색하여 실시간으로 처리되는 실제적인 연산량을 줄여 객체의 윤곽점을 효율적으로 추출할 수 있도록 하였다.

제안한 방법은 기존방법보다 오류빈도가 적은 정확률과 빠른 속도 보상을 받을 수 있으며, 실시간 객체 추적 시스템으로 적합함을 알 수 있다.

참고문헌

[1] H.Zhang, A.Kankanhalli, S.W.Smoliar, "Automatic Partitioning of Full-motion Video", Multimedia System, Vol.1, No.1, pp.10-28, 1993.

[2] 이완범, "저 전송률 무선 시스템에 적합한 새로운 움직임 추정 알고리즘 및 VLSI 설계에 관한 연구," 원광대학교 박사학위 논문, 2003.

[3] Andreas Koschan, Sangkyu Kang, Joonki Paik, Besma Abidi, Mongi Abidi, "Color active shape models for tracking non-rigid objects", Pattern Recognition Letters 24, pp.1751-1765, 2003.

[4] J.L. Starck,F. Murtagh, E.J. Candes,D.L. Donoho, "Gray and color image contrast enhancement by the curvelet transform", IEEE Transactions on Image Processing, VOL.12 NO.06 pp.0706~0717, JUNE 2003.

[5] 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, pp.42-53, 2001.

[6] A.Hanjalic, R.L.Lagendijk, J.Biemond. "A New Key-Frame Allocation Method for Representing Stored Video-Streams", Proc of the First International Workshop on Image Databases and Multimedia Search, Armsterdam of The Netherlands, pp.67-74, 1996.

[7] Y. Wu, D.Suter, "A Comparison of Methods for Scene Change Detection in Noisy Image Sequence.", Proc of the Fist International conference on Visual Information Systems, Melbourne, Australia, pp.459-468, 1996.

[8] S.Y.Wan, W.E.Higgins, "Symmetric region growing", International Conference on Image Processing 2000, VOL.12 NO.09 pp.1007~1015, SEPTEMBER 2003.