

차영상의 히스토그램 기반의 PAN/TILT 카메라 객체추적 시스템 이 만 석¹ · 최 진 구²

PAN/TILT Camera Object Tracking System based on Histogram of differences of frames

Man Seok Lee¹ · Jin Ku Choi²

Key Words : Tracking, 객체추적, Histogram, DVR

요 약

최근 영상 데이터를 사용하여 서비스를 하는 분야가 늘어감에 따라 그 중 보안과 관련하여 영상데이터 내에 존재하는 객체를 추출하여 추적하는 방법에 대한 연구가 지속적으로 발전하고 있다. 최근에는 프레임버퍼링을 사용하여 초당 30프레임 이상의 영상을 처리하는 경우가 많다. 하지만 영상을 통한 객체 추적의 경우 빠르게 객체를 인지하여 움직임을 추적할 수 있는 기술이 드물다. 본 논문에서는 지속 처리되는 문제를 극복하기 위하여 프레임 버퍼링을 사용하여 객체를 추적할 때 차영상의 히스토그램을 사용할 것을 제안한다. 이에 따라 히스토그램을 사용하는 객체추적 시스템을 설계 및 구현하고, 본 추적능력을 검증한 결과 2배에 가까운 성능이 향상됨을 얻었다.

1. 서론

최근 컴퓨터의 하드웨어의 성능이 크게 발전하면서 과거와는 달리 영상 및 음성 등의 대용량의 복잡한 멀티미디어 데이터를 처리하는 분야의 연구가 발전하고 있다.

하지만 영상처리 쪽에서는 이런 하드웨어적 성능을 따르지 못하는 소프트웨어 기술을 사용하여 해상도가 VGA이하의 영상, 30프레임 미만으로 개발하고 있다.

15프레임 이하의 재생속도로 객체를 추적하는 시스템의 경우 빠르게 이동하는 객체를 많이 놓치는 모습을 보게 된다. 이렇게 적은 횟수로 캡처된 시스템을 보안이나 고품질의 동영상을 녹화하는 서비스분야에서 좋은 결과를 기대하기는 어렵다.

본 연구에서는 프레임 버퍼링을 사용하여 영상 데이터를 얻어 처리하였다. 프레임 버퍼링을 사용하게 되면 VGA급 이상의 영상을 초당 30프

레이드로 처리가 가능하다. 이럴 때에 프레임 간격인 약 33ms 이내에 해상도가 VGA이상인 영상을 처리할 수 있는 기술이 필요하다.[1] 그렇기 때문에 이 짧은 시간 내에 시스템이 객체를 인식 추적할 수 있도록 하기위해 본 논문에서는 히스토그램을 사용하여 다양한 방향과 크기를 가지는 영상 내의 모션벡터에서 정확한 모션 벡터를 추출하여 객체를 추적하는 방법을 제안한다.

본 논문의 구성은 다음의 제 2절에서 시스템의 구성을 3절에서는 설계 및 구현 내용을 설명하고 4장에서는 실험결과, 마지막으로 5장에서 결론을 맺는다.

2. 시스템의 구성

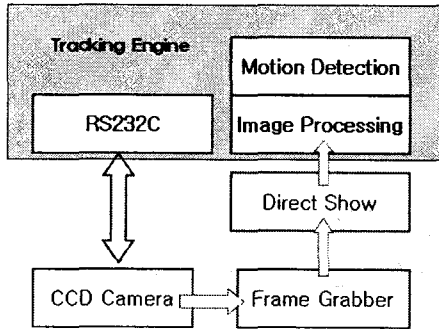
본 시스템은 영상을 직접 획득하기 위한 CCD 카메라와 아날로그 영상 신호를 디지털 신호로 변환 시켜주는 프레임 그래버로 구성된다.

객체를 추적하기 위해 상하좌우로 움직일 수 있는 PAN/TILT가 가능한 CCD카메라를 사용하고, 컴퓨터와 컴포지트 영상선 이외에 RS232C로 연결되어 있다. 이 RS232C로는 카메라 상태와 움직임에 관한 명령 정보가 전송되며, 카메라

1 한국산업기술대학교 컴퓨터공학과
e-mail : zeem0324@hotmail.com

2 한국산업기술대학교 컴퓨터공학과
e-mail : jkchey@kpu.ac.kr

Tel : 031-4968-281 Fax : 031-4968-299



[그림 1] 시스템 구성

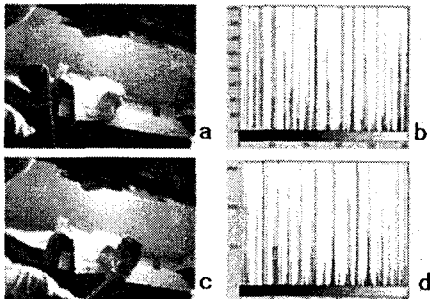
에서 발생한 영상신호는 콤포지트 영상선을 통해 프레임 그래버에 전달되고 프레임 그래버는 이 신호를 디지털 신호로 변환하여 응용프로그램으로 전달한다.

3. 설계 및 구현

영상내의 객체를 추적하기 위해 가장 먼저 해야 할 일은 영상의 변화 여부를 아는 일이다. 영상이 변했다는 것을 알게 되면 영상의 객체를 추출하는 알고리즘을 통해 객체를 추출하고 그 객체의 패턴을 알아내는 것이다.

영상의 변화 여부를 아는 데에는 많은 방법이 존재하지만 여기서는 히스토그램의 이용하는 방법을 제안한다.

히스토그램은 영상의 명암화소의 분포도를 의미한다. 히스토그램의 경우 1픽셀 당 8비트(256색)를 가지는 320x240의 영상에서 614400가지의 서로 다른 형태를 가진다[2].



[그림 2] a. 이전프레임의 영상 b. a영상의 히스토그램
c. 현재프레임의 영상 d. c영상의 히스토그램

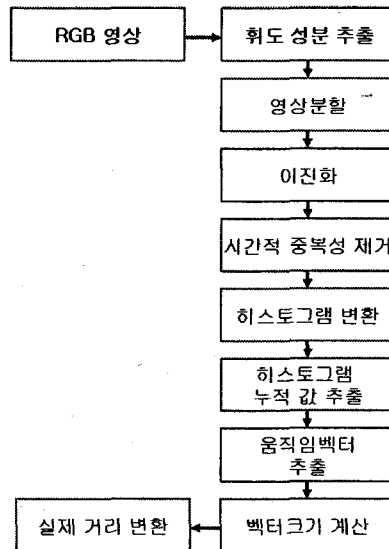
같은 장소에서 같은 사람이 찍은 영상이라도 히스토그램은 다르게 나올 수 있으며 영상의 크기가 커질수록 픽셀마다 가지는 색의 범위가 클

수록 상이한 히스토그램의 경우의 수는 커지게 된다. 단순 영상의 변화 여부를 알아내는 것이라면 이러한 히스토그램을 사용하는 방법도 나쁜 방법은 아니다.[3] 영상에 노이즈나 영상의 조그만 변화가 생기는 상황은 무시하고 사람이 움직이는 것과 같이 큰 변화를 알아내기 위해서 히스토그램이 적절하게 사용된다.

마스크를 쓰지 않고 외곽선을 추출하기 위해 두 연속된 영상간의 시간적 중복성을 제거한다. 이 방법의 결과는 두 영상간의 에너지 차이를 나타내기 때문에 영상 압축 과정에서 반드시 수행하는 작업이기도 하다[4].

영상의 중복되는 부분은 제거하고 변화된 부분은 변화 값으로 나타나기 때문에 영상크기를 n 이라고 할 때 $O(n)$ 만으로 객체추출과 기타 연산 등을 수행 할 수가 있는 좋은 방법이 된다.

이 두 가지를 결합하여 영상내의 객체를 추출하고 객체의 움직임벡터를 구하여 실제 추적거리를 구할 수 있다.



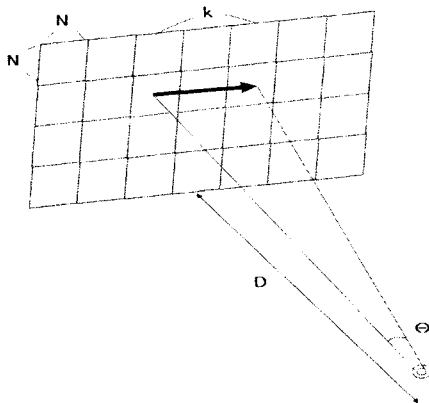
[그림 3] 제안하는 방법의 구조

제안하고자 하는 방법은 전처리 단계로 영상을 간결하게 처리하기 위해 YUV영상으로 변환을 하여 휘도 성분만을 추출한다. 그리고 추출된 영상에 섞여 있는 노이즈를 제거함으로 전처리가 끝난다.

얻어진 영상을 N개의 영상으로 분할을 한다. 그리고 이전 프레임의 같은 위치의 블록 간에

시간적 중복성을 제거한다. 중복성을 제거하게 되면 이전 프레임에서 변화된 부분만이 영상에 남게 된다.

움직임 벡터를 구하기 위해서 각 블록에 대해 히스토그램을 구한다. 히스토그램의 양자화 시에 변화 부분에 최댓값을 가지게 했다면 각 블록의 히스토그램 누적 값 중에서 최댓값을 가지는 블록을 알아낸다. 전 프레임에서 최댓값을 가졌던 블록의 위치를 시점으로 하고 현재 최댓값을 가지는 블록을 종점으로 하는 벡터를 구한다.



[그림 4] 영상 내의 움직임 벡터

이 벡터는 영상내의 상대적인 벡터이므로 실제 객체의 움직임에 대응되는 값으로 변환되어야 한다. 영상의 내용이 되는 객체와 카메라 사이의 거리 D에서 1블록인 N 만큼 화면을 이동시키는 각도를 θ 라 할 때, k 크기로 움직인 객체를 추적하기 위해 움직여야 하는 각도는 $k \cdot \theta$ 이다.

$$\theta = \frac{\pi \cdot N}{\beta \cdot D} \quad (\beta \text{는 절대 거리와 상대거리를 조절하는 가중치})$$

4. 실험 결과

실험은 VFW와 다이렉트 쇼를 사용하여 카메라를 움직이고 그 결과를 분석하였다. 영상은 VGA의 RGB24포맷을 사용하였다. VFW를 사용할 경우 15프레임을 지원하며 다이렉트 쇼를 사용할 경우 30프레임으로 캡처가 가능하다.

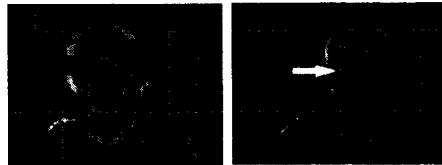
첫 번째 테스트는 15프레임에서 움직이는 사람을 대상으로 하였다. 15프레임이기 때문에 모션 벡터를 추출하는 방법을 전 픽셀을 대상으로 하였다.[5]

두 번째 실험 상황은 30프레임에서 본 알고리즘을 적용하여 앞서 명세한 상황과 비교하였다. 영상을 가로세로 80픽셀의 크기로 8x6크기로 분할하고, 이 때 카메라의 1블록에 해당하는 각도는 13도였다.

첫 번째 실험에서 사람이 한 방향으로 움직이다가 다른 방향으로 빠르게 움직이게 해보았다. 정확히 60ms 이내로 영상 내에서 빠져나가는 것이다. 이 때 카메라는 사람을 추적할 수 없었다. 이 이유는 60ms이내에 효과적인 모션벡터 추출이 불가능했기 때문이다.

하지만 30프레임에서는 움직임 벡터가 보다 세밀하게 작은 값으로 추출되어 카메라는 작은 각도로 세심하게 움직일 수 있게 되어 객체의 속도가 빨라져도 놓치는 일이 없게 된다.

따라서 제안하는 알고리즘을 통해 60m/s이하의 속도를 가지는 물체만을 추적할 수 있던 시스템에서 30ms/s이상의 속도를 가지는 물체를 추적할 수 있게 되어 약 2배의 성능개선을 이루었다.



[그림 5] 분할 블록을 통한 객체 움직임 추정

5. 결론 및 향후 연구

본 논문에서는 프레임 버퍼링을 사용하는 다이렉트 쇼에서 영상을 획득하여 객체를 추적할 수 있는 시스템을 설계하고 구현하였다.

프레임 버퍼링을 통해 초당 30프레임에 VGA급의 영상을 얻은 뒤에 전 영상과 현재 영상의 차영상을 얻은 뒤 그 영상의 히스토그램을 통해 가장 움직임이 많이 일어난 블록을 찾아내고 전 블록과 현재 블록의 위치를 통해 모션 벡터의 크기와 방향만큼 카메라를 움직이면 뛰어난 객체추적을 이룰 수 있고, 이는 30프레임에서 프레임마다 주어지는 시간 30ms이라는 시간안에 처리가 가능함으로 인해서 15프레임에서의 추적에 비해 2배에 가까운 성능을 얻을 수 있었다.

하지만 본 연구를 시행하면서 완벽하게 동작하는지 행해지는 테스트의 정확한 단위와 명세가 필요하다. 아직까지 이 부분에 대해서 테스트 방법의 표준이나 명세화가 되어 있지 않아 결과를

객관적으로 분석하기가 힘들었다.

본 내용을 계속 보완해 나간다면 앞으로 DVR
과 VOD 콘텐츠 제작 등 특정 사람이나 물체를
주시해야 하는 분야에 있어 좀 더 지능적인 서
비스가 가능할 것으로 생각된다.

향후 연구에서는 객체의 움직임과 방향에 맞춰
동작하고 객체를 놓쳤을 경우 다시 추적할 수
있는 지능적인 객체추적 알고리즘을 개발할 예
정이다.

참 고 문 헌

- [1]. Wikipedia, "Frame Buffer",
http://en.wikipedia.org/wiki/Frame_buffer
- [2]. Tao Zhang... Reliable detection based
on the difference image histogram, IEEE,
2003
- [3] 이승철, 이귀상, 최덕재, 김수형, 2005, "자
동보정 카메라에서 HSV를 이용한 이동객체 검
출", 정보과학회 2005 추계학술대회, Vol.32
No.02 pp0910~0912, 2005
- [4] Randy Crane. "Simplified approach to
Image processing", Prentice Hall, 1997
- [5] 김영재, "내용 기반 동영상 검색을 위한 컬러
및 모션 특징 추출 알고리즘", 한국방송공학
회, 1999
- [6] 정석재, "움직임 벡터를 이용한 이동 물체
추적 시스템의 설계 및 구현", 한국정보과학회
학술발표 논문집 p533-536, 1997