

가상 공간의 셀 단위 처리를 이용한 움직임 감지 시스템 성능 향상 기법

홍석용⁰ 박영환
한성대학교 컴퓨터 공학과
{phantom⁰, yhpark}@hansung.ac.kr

A technique to increase the performance in movement detecting systems by processing the cells in a virtual zone

Seok Yong Hong⁰ Young-Hwan Park
Dept. of Computer System Engineering, Hansung University

요 약

영상 내에 존재하는 사물에 대해서 변화를 감지하는 것은 일반적으로 카메라 영상에서 입력되는 두 개의 순차적인 영상에 대한 연산으로 구현된다. 그런데 카메라 영상을 이용하여 사물을 감지하는 경우 보통 이미지 전체가 아니라 일부에 대해서 관심을 갖게 된다. 따라서 전체 이미지에 대해서 움직임 감지 방법을 적용하는 것은 메모리 낭비와 처리 시간의 증가라는 비효율성을 야기시킨다. 또, 영상 내에서 여러 개의 움직인 물체를 발견하는 것은 임의적인 물체의 수와 모양 때문에 처리 방법이 복잡하다. 따라서 본 논문에서는 전체 이미지에서 일부분에 대하여 움직임을 감지할 수 있도록 사용자가 처리 구역을 작은 셀들로 나누어 가상 공간을 설정하고, 나누어진 셀 당 한 물체의 움직임을 감지하는 기법으로 기존 움직임 감지 시스템에 효율성을 제고하고자 한다.

1. 서 론

일반적으로 연속적인 영상에서 움직임을 감지하는 방법은 두 순차적인 이미지의 차를 계산하여 변화를 감지하는 것이다[1][2]. 이는 영상의 크기나 포맷에 따라 적지 않은 시스템 자원을 필요로 한다.

보통 한 영상에서 움직이는 물체의 크기는 국부적이다. 따라서 부분 윤곽선 추출 기법 등을 이용하면 어느 정도의 효율성을 높일 수 있다[1][3].

하지만 일반적으로 카메라 이미지 중에 사용자가 관심을 갖는 부분은 전체 이미지가 아니라 이미지 내의 일부분이다. 결국 기존의 부분 윤곽선 추출 방법은 여전히 불필요한 부분에 대해서 처리를 하는 결과를 초래한다.

변화가 감지된 이미지에서 변화된 물체를 발견하는 것은 쉽지 않은 문제이다[4][5][6][7]. 물체의 개수가 여러 개일 경우나 물체의 모양이 단순하지 않을 경우 그것을 추출하기 위해서 복잡한 처리를 필요로 하게 된다.

본 논문에서는 이런 문제점들을 개선하고자 사용자가 여러 개의 셀을 포함하는 가상 공간을 설정하여 각 셀을 기본 단위로 움직임을 감지하는 방법을 제시한다.

2. 본 론

2.1 부분 윤곽선 추출법

영상 정보에서 사물의 움직임에 의한 변화는 일반적으로

국부적이다[3]. 그러나 일반적인 윤곽선 추출 기법은 가장 처리 시간이 많이 소요되는 윤곽선 추출을 화면의 전체에 대하여 수행한다. 결국 국부적인 정보의 추출을 위하여 많은 자원을 소모하는 결과를 초래한다[1][3].

부분 윤곽선 추출 기법은 변화된 부분이 포함되어 있는 부분만을 찾아내어 윤곽선 추출을 수행한다. 따라서 변화된 부분이 작다면 처리 속도를 훨씬 증가시킬 수 있다.

부분적인 변화 정보는 시간 미분 결과를 사용하여 얻을 수 있다. 시간 미분을 하여 변화 전후의 국부적인 정보를 원 영상에서 추출하여 윤곽선을 발견하고, 그 부분에 대해서만 시간 미분 결과에 곱셈 연산을 수행하면 시간 미분법 중 하나인 Murray 기법과 동일한 결과를 얻을 수 있다(그림. 1)[3].

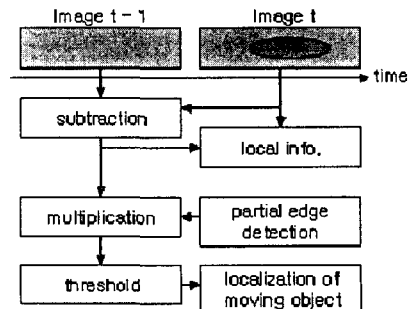


그림 1. 부분 윤곽선 추출 기법의 처리 과정

이 기법은 변화된 부분에 대해서만 윤곽선 추출을 위한 처리를 수행하므로 일반적인 방법보다 효율은 좋지만 초기에 두 전체 영상의 차를 구하여 변화 정보를 얻어내기 때문에 여전히 비효율성을 내포한다.

또 사용자가 영상의 일부에 관심을 갖고 있다면 그 부분에 대한 정보를 동적으로 지정할 수 없기 때문에 전체 영상에 대하여 부분 윤곽선 추출 과정을 수행해야 한다.

2.2 제안하는 방법

본 논문에서 제안하는 방법은 크게 두 가지 절차를 거치게 된다. 카메라와의 거리를 고려하여 물체 하나를 포함할 수 있는 충분한 크기로 셀의 크기를 정하고 관심 지역에 가상 공간을 설정한다. 그 다음 셀 단위의 이미지에 대하여 부분 윤곽선 추출 기법을 적용하여 변화를 감지한다.

2.2.1 가상 공간의 설정

움직임 감지의 대상이 되는 물체의 크기는 물체의 종류에 따라 다르다. 또, 카메라와의 거리에 따라 영상에서 물체의 크기가 변화한다. 그렇게 때문에 원하는 위치에서 관심의 대상이 되는 한 개의 물체를 충분히 포함할 수 있는 크기로 셀을 지정한다. 그림 2는 주차 구역 감시를 위하여 차량 하나를 충분히 포함할 수 있는 크기로 셀을 설정한 가상 공간을 보여준다.

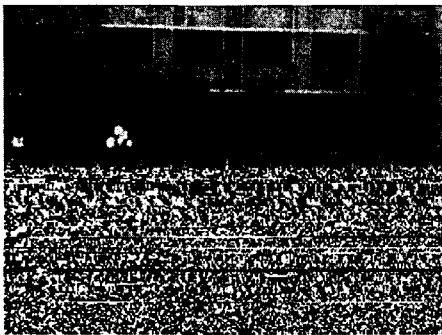


그림 2. 가상 공간의 설정

전체 이미지 상에서 감지된 물체의 좌표를 알아야 할 경우를 위해 셀의 시작 좌표 $\langle top_left_x, top_left_y \rangle$ 를 저장할 수 있다. 전체 영상에서 물체의 좌표는 셀의 시작 좌표와 셀 내에서 물체의 좌표를 합한 값이 된다.

2.2.2 셀 단위 부분 윤곽선 추출

가상 공간을 설정하여 움직임 감지를 위한 관심 영역이 셀 단위로 정의되면 각각에 대해서 그림 3과 같은 움직임 감지를 위한 부분 윤곽선 추출 기법을 적용할 수 있다. 만약 그레이 레벨의 이미지로 처리 가능한 응용의 경우 하드웨어나 소프트웨어 변환기를 사용하여 이미지의 크기와 처리 속도를 줄일 수 있다.

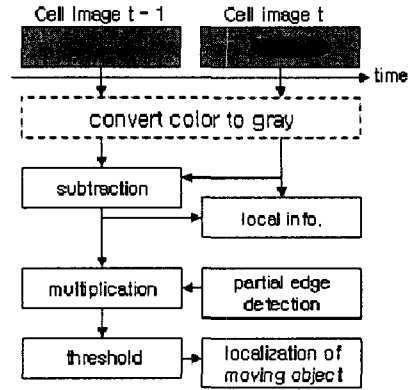


그림 3. 부분 윤곽선 추출 절차

일정 시간 간격의 두 영상이 주어지면 두 영상의 차를 구하여 위치 정보를 얻어낼 수 있다. 만약 영상에 변화가 발생했다면 그 변화된 부분은 위치 정보상에서 0이 아닌 다른 값을 갖게 되고, 그 값이 특정 임계치 이상이 될 때 두 영상간에 변화가 발생했다고 판단할 수 있다.

각 셀에 대하여 입력 영상에 변화가 있다고 판단되면 부분 윤곽선 추출 기법을 두 영상에 적용하여 물체의 정확한 좌표를 얻어낸다. 셀의 부분 영상에 변화가 없다면 부분 윤곽선 추출 기법을 적용하지 않음으로써 처리 시간을 향상시키게 된다.

2.3 성능 분석

전체 영상에 대하여 부분 윤곽선 추출 기법을 적용하는 방법과 제안한 방법의 처리 시간을 비교하기 위해 실험을 하였다. 본 측정에서는 크기가 640×480 픽셀이고, 픽셀당 1바이트의 영상을 사용하였다. 가상 공간의 크기는 전체 영상의 75%, 50%, 25%로 각각 설정하여 측정을 하였다. 또 모든 셀에 변화가 발생한 경우에 대하여 고려하였다. 가상 공간의 설정 크기를 100%로 설정한 경우가 전체 영상에 대하여 부분 윤곽선 추출 기법을 적용한 경우이다.

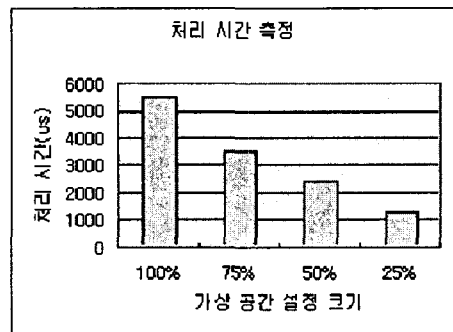


그림 4. 처리 시간의 비교

그림 4를 보면 가상 공간의 크기에 따라 처리 시간이

반비례함을 알 수 있다. 결국, 전체 영상에 대하여 관심 있는 부분을 최소한으로 명확하게 설정할 수 있다면 처리 효율은 높아지게 된다.

다음으로 변화가 있는 셀의 개수에 따라 성능이 어떤 차이를 보이는지 실험을 하였다. 가상 공간을 전체 영상 크기의 50%로 고정을 하고 20개의 셀 중 변화된 셀의 개수를 변경하면서 실행시간을 측정하였다.

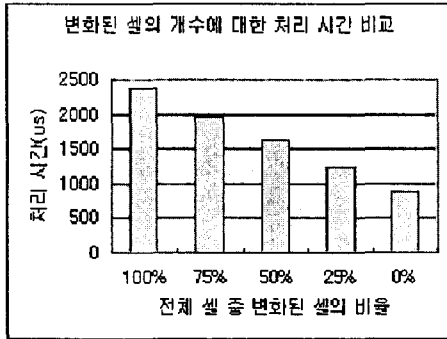


그림 5. 변화된 셀의 개수에 따른 처리 시간 비교

실험을 통하여 변화된 셀의 개수가 적을수록 실행 속도가 빨라지는 것을 확인할 수 있었다(그림 5). 결국 위치 정보를 이용하여 변화가 없는 셀에 대한 윤곽선 추출 과정을 생략함으로써 처리 시간을 단축시킬 수 있었다.

가상 공간내의 셀 단위 처리는 물체의 좌표 값을 계산하는데 쉽고 효율적인 방법을 가능하게 한다. 만약 전체 영상을 대상으로 처리했을 경우 움직인 물체가 다수 개 존재하면 각각의 물체에 대한 좌표를 구하는 것은 쉽지 않다. 물체의 모양이 복잡한 경우에 한 물체를 구분하는 방법 또한 구현이 쉽지 않다. 하지만 물체 하나와 비슷한 크기의 셀을 설정하고 이 셀 단위로 처리를 수행하면 움직인 물체의 좌표를 찾는 계산은 각 셀에 대해 한 개의 물체에 대해서만 수행하면 되므로 간단한 알고리즘을 적용할 수 있다.

3. 결론 및 향후 연구 과제

본 논문에서 제안한 방법을 움직임 감지 시스템에 적용할 경우 가상 공간의 크기가 줄어들고 변화된 셀의 개수가 적을수록 처리 시간이 향상됨을 알 수 있었다. 또한 움직임이 감지되었을 때 물체의 좌표를 계산하는 간단한 방법을 적용할 수 있게 하였다.

본 기법은 무인 주차 감시 시스템과 같은 실시간 움직임 감지 시스템에 적용할 수 있고, 처리 오버헤드를 줄일 수 있기 때문에 임베디드 시스템으로 효과적으로 적용할 수 있다.

처리 단위를 셀 단위로 분할하였기 때문에 병렬 처리를 사용한 처리 시간 단축을 고려해 볼 수도 있다. 병렬 컴퓨팅에서 마스터-워커 패러다임을 적용하여 태스크 서버는 셀 단위로 작업을 분할하고 각 워커들은 할당 받은 일을 병렬적으로 처리하여 전체 시스템의 성능을 향상시킬 수 있을 것이다[8].

참고 문헌

- [1] Murray, D. and Basu, A., "Motion Tracking with an Active Camera", IEEE Trans. Patt. Anal. Mach. Intell., 16(5), p449-459, 1994
- [2] Li, K., Park, Y. H. and Hou, K. M., "Intelligent Camera Dedicated for a Multi-Sensor Perception: Real time Motion and Object Tracking", International Conference on Advances in Vehicles Control and Safety(AVCS' 98), p76-86, 1998
- [3] 박영환, "영상 변화 추적 시스템", 물리탐사학회, Vol. 2, No. 3, p154-158, 1999
- [4] Nichol, D. and Fiebig, M., "Tracking multiple moving objects by binary object forest segmentation", Image Vision Computing, 9(6), p362-371, 1991
- [5] Molineros, J. and Sharma, R., "Real-Time Tracking of Multiple Objects Using Fiducials for Augmented Reality", Real-Time Imaging, Vol. 7, No. 6, p495-506, 2001
- [6] Iketani, A., Nagai, A., Kuno, Y. and Shirai, Y., "Real-Time Surveillance System Detecting Persons in Complex Scenes", Real-Time Imaging, Vol. 7, No. 5, p433-446, 2001
- [7] Ziliani, F. and Cavallaro, A., "Image Analysis for Video Surveillance Based on Spatial Regularization of a Statistical Model-Based Change Detection", Real-Time Imaging, Vol. 7, No. 5, p389-400, 2001
- [8] Nicholas, C. and David, G., "How to Write Parallel Programs: A Guide to the Perplexed", ACM Computing Surveys, Vol. 21, No. 3, 1989